## *TABLE OF CONTENTS*

(FALL, 1999)

(FALL, 1999)

# DIRECTOR’S VISION

(FALL, 1999)

# DOING MORE BY DOING LESS: OXYGEN

By Michael Dertouzos

You are on business in Paris. You excuse yourself from the meeting, go outdoors, pull out your handy and ask it to contact Joe. It "sniffs" the electro-magnetic surround, finds the local GSM cellular net, and calls Joe in New York. Joe's in-the-wall computer answers. You tell it that the call is urgent, and since the machine also recognizes your handy, it forwards the connection to Boston, where Joe, seated in an office of your subsidiary, is chatting with the local VP. The wall machine of that office, sensing that the door is open, and based on an automation script Joe gave a year ago, determines that it can interrupt him. A pre-stored, life-size image of yourself flashes on the wall, as if you had poked your head through the open door. The image clears its throat.

Joe, who could have said, "go away" and you'd never have been the wiser, says, "Hello Jane. How's Paris?" You explain that an attractive site for your French office has materialized, but you have only six hours to grab it. Joe understands and says, "Oxygen, get Juan, Michael and Mary." The machine finds the first two via their home machines. Mary is driving to work in Chicago, and is connected via her car-trunk computer.

Within seconds, the Oxygen network creates a secure "collab region" for these five co-workers. As they confer, they say things like "Oxygen, get me the map that came with Lori's long message about two months ago," or "Get all the Web info you can on this new site." At the end of the meeting, Joe says, "We'll do it, unless I see an obstacle in the next hour." He points his handy to the printer saying "Oxygen, send a copy of all the documents we reviewed." You rush to set up the closing.

This scenario gives a partial glimpse of what people will be able to do with Oxygen. The new system is a five-year, multi-million research project, pursued in collaboration with our sister Artificial Intelligence Laboratory and sponsored by DARPA. It was launched at the Laboratory's 35th anniversary celebration, in April of 1998. Oxygen's overarching purpose is to let people do more by doing less. It does so through eight hardware-software technologies that enable natural use and increased human productivity.

The fun begins with the Handy21, a portable universal device that Oxygen users carry. It looks like a cell phone but also has a small screen, a camera, a GPS module, an infrared detector, and a powerful computer. Except for a tiny analog part connected to its antenna, the entire innards of the handy are software controllable: Like a chameleon, the unit can change at the flip of a bit, from a cellphone, to a two-way radio, a TV, a beeper, a handheld computer, a pointing device and more.

Next is the Enviro21 device, which unlike the person-centered handheld is space-centered -- in the office wall, the car trunk, or the home basement. Enviros bear the same relationship to handys as do power sockets to batteries -- they mimic the handys, but with greater storage capacity, processing power and communication speed. Many enviros are also connected to sensors and actuators, and can therefore raise the room temperature, operate a fax machine and tell if the door is open.

The third technology, the N21 network, links all Oxygen devices to each other and to the world's networks, and creates secure collaborative regions that rise and collapse as needed. The fourth technology is spoken-dialog software, built deep within Oxygen rather than attached as a mere interface or a fashionable afterthought. Together with some additional visual resources that observe Oxygen users, it is responsible for making Oxygen natural and easy to use.

Besides these four core technologies, Oxygen also provides four user technologies. The knowledge-access technology helps you find the information you need, in your own familiar way, among your own data, the data that friends share with you, and the vast info-terrain of the Web. The automation technology lets you tell the machine what routine human work it should offload from your brain and eyeballs -- for example, when to interrupt you. The collaboration technology helps a group work together by tracing the discussion and keeping an accessible trail of issues, documents, and conversation fragments. The customization technology adapts Oxygen to the needs of individual users and it ensures that all software is downloaded automatically to all devices when new versions have become available, errors have been detected, or users have asked for new capabilities.

With Oxygen, we introduce a new concept: the machine identity of the physical objects we care about. With this capability, the handys can detect the physical objects to which they are pointed, can direct them to do useful tasks on our behalf, and can provide users with "X-ray" vision, by overlaying machine models related to these devices upon the physical images that we see...for example a list of the activities that take place behind an office door.

Oxygen's power lies not in these technologies individually, but rather in their synergistic use over real applications. Moreover, we believe that the technologies of Oxygen are the basic new technologies needed for achieving the myriad of different things we envision in the new world of information. If our dreams come true, Oxygen will mark as major a departure from today's desk-top metaphor as the latter did, when first introduced. By capturing the human utility of new technology, Oxygen should encourage application developers and users to bend machines toward human needs, rather than the other way around.

**CONTACT**

Michael L. Dertouzos, dertouzos@mit.edu

(FALL, 1999)

# HARDWARE & SOFTWARE TOOLS

## A Framework for Automation Using Networked Information Appliances

*Leader:*
Srinivas Devadas

*Members:*
Sandeep Chatterjee, Ning Cheng Cheng,
Fumi Huang, Thanisara Kiatbaramee, Shiraishi, Mark Ye

*Sponsors:*
Nippon Telegraph and Telephone

### Background

We propose to automate information transfer, exchange and management tasks through the synergistic use of computers, intelligent software and sensors or appliances. There are many varieties of information automation problems and new ones arise every day. For example, an intelligent navigation system tells a driver the fastest way of getting from point A to point B, which takes into account current traffic and weather conditions, as well as accommodating driver preferences. A smart thermostat regulates temperature within a home predicting human presence or activity so as to minimize energy consumption.

We are building a framework of computers, sensors and intelligent software which allows for the rapid deployment of information automation problems. We believe that the deployment of a solution should take no more than a week, including customizing hardware and writing software applications.

### Why Is This Interesting?

In general, any appliance can be made intelligent and networked using our approach. We envision interconnecting these appliances using wireless, ethernet or local-area networks, and also providing local computation for each appliance to reduce bandwidth requirements. This strategy can be used to connect appliances within a home or office, as well as to connect up various homes, or businesses.

### Approach

Our approach is based on the (idealistic) notion of composable computing. In composable computing, we have a single interface for all of the black boxes which could be computers or peripherals, and the user can plug-and-play with these devices to suit the needs of his application. Ideally, this is independent of the processor or the operating system or the network type. As an example, the user may put together a computer block, a network block, and a camera block to create a networked digital camera. If he adds image recognition software, downloads it via the network and runs it on the computer, he can create an intelligence surveillance system. Another important feature of a composable computing platform is modularity. Modularity allows the user to upgrade any component of the system, be it hardware or software or appliance-specific components, independent of other components.

(FALL, 1999)

We are implementing a composable computing platform within Oxygen and this framework is comprised of hardware building blocks that consist of credit-card-sized computer cards, and PCMCIA peripherals. The computer cards and the PCMCIA peripherals have exactly the same Cardbus interface. This symmetry is the key for composability and modularity. The computer and peripheral cards can be interconnected together using backplanes that come in two varieties. A stand-alone backplane can be used to connect together up to 8 cards. The embeddable backplane is a two-socket backplane that can be embedded into any kind of appliance.

We are implementing a software development environment that will allow a developer to rapidly code intelligent applications that will run on the computer cards. The paradigm we follow is to layer evolving control over distributed applications. Evolving control consists of a watch-reason-automate loop. In the Watch step, data is monitored using different kinds of hardware and software probes. In the Reason step data is analyzed and correlated with past data. The Automate step actually makes control decisions. This watch-reason-automate loop is implemented by a software developer, and is rapidly deployable because we use intelligence mechanisms that are common across different domains, for instance, pattern recognition algorithms, correlation algorithms, and data mining algorithms, so we can obtain reusability at the algorithmic level. Different appliances, such as an intelligent surveillance system or a route-prediction system can reuse the same algorithm-level software. The user is allowed to guide the automation process and to override decisions, and this is accomplished by providing appropriate hooks in the evolving control module.

## Progress

Prototype hardware corresponding to computer card, stand-alone backplane and embeddable backplane has been built and debugged. A Linux port to computer card has been completed, as well as the middleware. Simple information appliances such as a smart remote control that predicts what channels a human may wish to watch based on past history have been built.

## Future

A version of the computer card whose dimensions are exactly those of a PCMCIA peripheral is being built. A large number of networked information appliances and computer-embedded toys will be built in the near future including an intelligent GPS system, and a smart networked audio player. We are exploring the integration of more complex sensors into the framework, as well as the use of networked speech servers for to easily incorporate speech interfaces to information appliances.

## Contact

Srinivas Devadas, devadas@mit.edu

# Automated Architecture Exploration for Application-Specific Processors

*Leader:*
Srinivas Devadas

*Members:*
Dan Engel, George Hadjiyiannis

*Sponsors:*
National Science Foundation

## Background

Processors can be general-purpose, or application-specific. General-purpose processors can be used to perform all computing tasks, but their performance may be poor for certain applications. In addition the power dissipated by conventional processors may be too large for use in portable communication applications.

## Why Is This Interesting?

Application-specific processors can run particular applications orders of magnitude faster than conventional processors. For any given application, a particular architecture and instruction set will provide the best tradeoff between area, performance, power dissipation, and cost. Since architectural exploration is an art rather than a science, to automate architecture exploration, it is important to first develop formalisms that can result in rapid, yet exhaustive exploration of the space of possible architectures. These formalisms and associated automation tools will enable the design of optimal architectures for a particular application.

## Approach

Efficient design space exploration of application-specific instruction processor architectures requires the development of a mixed hardware/software design environment. The cornerstones of the design environment are a machine description language ISDL, retargetable tools, and a hardware synthesis tool. The machine description language is used to describe the micro-architecture of the application-specific instruction processor. A retargetable assembler and compiler which are driven by the machine description are used to compile an application onto the described architecture. The performance of the architecture on the application code can be analyzed using a retargetable simulator. Finally, a hardware synthesis tool can translate the machine description into a logic implementation. The logic implementation can be analyzed for timing, or laid out in silicon using commercial CAD tools. Given this framework, an architect can use a single description language, namely ISDL, to explore different architectures, by iteratively modifying the description, and running the tools in the manner described above, till all design constraints are met.

## Progress

We have finished the specification of the ISDL language, and have finished writing automated generators for the assembler, disassembler and simulator. The retargetable code generator framework has been completed; code optimization techniques are being implemented. The automatic translation of ISDL descriptions into the Verilog hardware description language has been shown to be possible; this step has to be implemented in a synthesis tool.

## Future

We plan to finish the implementation of this system this year, and spend the early part of 2000 running different applications through the system to automatically produce synthesized architectures for audio and video processing applications that exhibit significant power or performance improvements over general-purpose processors.

## Contact

Srinivas Devadas, devadas@mit.edu

(FALL, 1999)

## Curl: An Integrated Authoring Environment for the Web

*Leader:*
Steve Ward

*Members*:
Mark Herschberg, Matt Hostetter, Art Housingher, David Kranz,
Pat LoPresti, Chris Terman, Marty Wagner

### Background

Web content is currently created using a growing array of mutually incompatible languages, standards, and tools, serving various needs ranging from formatted text to performance-sensitive programmed content. Curl is an attempt to serve this broad range of requirements within a coherent linguistic framework.

### Why Is This Interesting?

A uniform semantic basis for Web content potentially
(a) eliminates semantic discontinuities, improving opportunities for integration of different aspects of content;
(b) minimizes the distinction between "authors" and "programmers"; and
(c) provides an evolutionary path toward a new generation of user interfaces to informational services based on the Web interaction model rather than that of conventional operating systems.

### Approach

Curl offers a single, coherent linguistic basis for expression of Web content at levels ranging from simple formatted text to contemporary object-oriented programming. Curl is part of a research effort aimed at eliminating discontinuities from the function/sophistication curve. This yields an authoring environment in which (1) incremental functionality requires incremental skill acquisition, and (2) a consistent semantics avoids communication obstacles between separately encapsulated fragments of content. We characterize Curl as a "gentle slope system" because it makes it easy to transition from one point to another in the function/sophistication spectrum.

### Progress

Prototype Curl implementations are available for Windows 95, Windows NT, and UNIX (x86 and SPARC) platforms from our web site, along with on-line documentation.

### Future

A major technology transfer effort is underway, involving a spin-off company dedicated to furthering and promoting both the goals and the technology of the Curl project.

(FALL, 1999)

## Contact

Steve Ward, ward@mit.edu
http //cag-www.lcs.mit.edu/curl

## The Flex Project: Program Analysis and Compilation Technology for Active Distributed Systems

*Leader:*
Martin Rinard

*Members:*
C. Scott Ananian, Andy Berkheimer, Duncan Bryce, Brian Demsky, Catalin Francu, Ovidiu Gheorghioiu, Felix Klock, Darko Marinov, Maria Cristina Marinescu, Radu Rugina, Alex Salcianu, Karen Zee

### Background

A key challenge facing computer science is developing technology that allows organizations to use distributed computing systems effectively. Examples of distributed computing systems include: groups of sensors; the information management system in an airplane, ship, or building; and enterprise information systems consisting of networks of clients and servers. Such systems require software that operates robustly in the presence of failures and malicious attacks, is easily upgradable during continuous system operation, and shares information quickly and easily.

Current development methods require programmers to directly implement all of the above functionality. The complexity of doing so results in brittle software that is vulnerable to attack, has unpredictable behavior in the face of component failures, and is difficult to develop and maintain.

### Why Is This Interesting?

Distributed computing systems, especially active distributed computing that interacts with the physical world via sensors and embedded devices, plays a central role in the information acquisition and management needs of every large organization. The hardware required to deploy these systems is available at low cost. The primary obstacle is developing software that can integrate the hardware components into a unified system. We propose to develop a set of tools that can revolutionize the development of this class of software.

### Approach

The basic idea is to give the Flex compiler a global, high-level view of the application, either by coding the application as a single program written in an object-oriented language, or by building the application out of components delivered in a high-level, portable format such as Java byte codes. The Flex compiler then automatically partitions the application into segments, distributes the segments appropriately across the devices and machines, generates the communication required to make the segments work together properly, and produces application-specific monitoring software that reacts to failures by redirecting application activities to available machines and devices. The resulting implementation operates gracefully in the face of partial failures, comes with isolation guarantees that describe how specific failures can affect the operation of the system, supports software upgrades during on-going system operation, and appropriately manages the safety and consistency of the system as devices disconnect from and reconnect to the system.

Advanced program analysis and compilation technology is the key to the success of the Flex project. This technology allows the Flex compiler to recognize related groups of objects, called *teams*, that work together to accomplish a specific task. Examples of teams include *interactors*, which control the interaction of a specific device or machine with the physical environment or user, *transducers*, which process raw data to deliver usable processed information*, services*, which interact with multiple devices or machines to implement components such as a name service or distributed database, and *coordinators*, which combine interactors, services, and transducers to deliver complete applications. In the source program, objects interact using a simple, uniform mechanism from object-oriented programming: method invocation. In the generated code, the compiler automatically implements the interaction using whatever mechanism is available and appropriate. Objects on the same machine or device share an address space and use direct method invocation for efficiency. Objects on different machines or devices use network abstractions such as message passing or remote method invocation.

The Flex compiler generates code that specializes each team for its intended use. Interactors for embedded devices must often be compiled to low-level languages such as C or native machine code. Services are usually compiled with automatic replication and sophisticated protocols that ensure reliable operation in the face of partial failures. Coordinators are compiled to use the specific communication protocols that the compiler has chosen for the teams that they interact with.

Because the Flex compiler has a global view of the application, it can provide useful information about its behavior. It can, for example, analyze the application to determine that a failure in one hardware component of the system will not affect the correct operation of another component. It can also characterize how the interoperation of the teams affects reliability and use this information to determine which specific combinations of hardware or software failures can cause the overall system to fail. Further examples include the identification of potential sources of inconsistency in reliable code that continues to operate while disconnected from part of the system, and location of performance bottlenecks such as insufficient network bandwidth between hardware components.

### Progress

We have developed the initial compiler infrastructure. This infrastructure reads Java bytecodes and converts them into a format suitable for analysis and transformation. The development of the basic analyses is underway. In the near future we expect to be able to automatically partition standard Java programs for distributed execution on multiple machines.

### Future

In the future we will focus on partitioning programs for disconnected execution, automatically applying transformations that make software execute reliably in the face of partial failure, and on compiling for networks of sensors and embedded devices. We will also develop program analysis tools that give the developer hard information about how specific failures will affect the overall execution of the system.

Finally, we will develop software that will monitor the system to detect malicious attacks and take action to ensure that it will continue to operate in the face of the failures that these attacks are likely to cause.

## Contact

Martin Rinard, rinard@lcs.mit.edu
http://www.flex-compiler.lcs.mit.edu/

## The RAW Project

*Leaders:*
Anant Agarwal, Saman Amarasinghe

*Members*:
Jason Kim, Sam Larson, Walter Lee, Albert Ma,
Frank Matthew, Csaba Andras Mortiz, Devabhaktuni Srikrishna,
Mark Stephenson, Michael Taylor

### Background

As the VLSI technology allows a billion transistors on a chip, building a fast and efficient microprocessor without significantly increasing the design and verification costs becomes a challenging problem. However, microprocessor design can be substantially simplified by off-loading many tasks done in hardware to software with the help of whole program analysis and other advanced compiler techniques. The microprocessor design is also influenced by the changes in next generation application workloads, which are centered around streaming and multimedia applications.

### Why Is This Interesting?

The Raw team is investigating novel approaches to breaking the performance and complexity barriers that plague existing microprocessors by exploiting these trends in VLSI technology, compiler techniques and application workloads. The Raw team is in the process of designing and building a microprocessor and a compiler system that address these issues.

### Approach

The Raw approach is to rely on a simple, highly parallel VLSI architecture that fully exposes the hardware architecture's low-level details to the compiler. This allows the compiler to determine and implement the best resource allocation for each application. The Raw architecture is scalable, is much simpler to design than today's superscalars, and supports efficient pipeline parallelism for multimedia and streaming applications.

The Raw design features a two-dimensional mesh of identical tiles, with each tile having its own instruction stream, register file, memory, ALU, and switch. Each processor on a tile is a simple, RISC pipeline. The switches implement both a static network and a dynamic network. The static network is under the control of the instruction streams on the switches, while the dynamic network routes messages conventionally by reading the headers of messages.

The Raw compiler exploits all the hardware resources of the Raw architecture to carefully orchestrate the execution of applications. The Raw compiler accepts sequential C or FORTRAN programs and automatically parallelizes them for a Raw machine. The compiler consists of two main phases, the compiler-managed memory system and the space-time scheduler. The compiler-managed memory uses program information to determine the best memory allocation and customize memory accesses on the static and dynamic networks. The space-time scheduler parallelizes each basic block of the program across the processors, and performs efficient control flow across basic blocks.

### Progress

The initial results, obtained by compiling a few sequential C and FORTRAN benchmarks on the Raw compiler and evaluating them on the Raw simulator, show that the Raw compiler is able to exploit ILP profitably across the Raw tiles. For eight selected benchmarks from the Raw benchmark Suite and the Spec92fp suite, an average speedup of 19.7 on 32 tiles was obtained.

### Future

The RAW team is currently working on a VLSI implementation of the Raw processor as well as many extensions and improvements to the compiler system.

### Contact

http://cag.lcs.mit.edu/raw

(FALL, 1999)

## Active Trust Management for Autonomous Adaptive Survivable Systems

*Leaders:*
Jon Doyle, Howard Shrobe

*Members:*
Isaac Kohane, Bill Long, Peter Szolovits

*Sponsor:*
Defense Advanced Research Projects Agency

### Background

The traditional approaches to building survivable systems assume a framework of absolute trust requiring a provably impenetrable and incorruptible Trusted Computing Base (TCB). Unfortunately, we don't have TCB's, and experience suggests that we never will.

We, therefore, must instead concentrate on architecting software systems to provide useful services in an imperfect environment in which any resource may have been compromised to some extent.

The Active Trust Management (ATM) project explores the hypothesis that such systems can be built by restructuring the ways in which systems organize and perform computations. In particular,

1. Such systems will estimate to what degree and for what purposes a computer (or other computational resource) may be trusted, as this influences decisions about what tasks should be assigned to them, what contingencies should be provided for, and how much effort to spend watching over them.

2. Making this estimate will in turn depend on having a model of the possible ways in which a computational resource may be compromised.

3. This in turn will depend on having in place a system for long term monitoring and analysis of the computational infrastructure which can detect patterns of activity indicative of successful attacks leading to compromise. Such a system will be capable of assimilating information from a variety of sources including both self-checking observation points within the application itself and intrusion detection systems.

4. he application systems will be capable of self-monitoring and diagnosis and capable of adaptation to best achieve its purposes with the available infrastructure.
5. This, in turn, depends on the ability of the application, monitoring, and control system to engage in rational decision making about what resources they should use in order to achieve the best ratio of expected benefit to risk.

(FALL, 1999)

Our claim is simple but revolutionary: “Survivable systems must make careful judgments about the trustworthiness of their computational environment and make rational resource allocation decisions accordingly.”

### Contact

Jon Doyle, doyle@mit.edu
Howard Shrobe, hes@ai.mit.edu
http://www.medg.lcs.mit.edu/projects/atm

(FALL, 1999)

## Automated Negotiation

*Leaders:*
Jon Doyle, Howard Shrobe

*Members:*
Yu-Han Chang, Robert Laddaga

*Sponsors:*
Defense Advanced Research Projects Agency

### Background

The Automated Negotiation project seeks to provide mechanisms for automated resource allocation tasks in a variety of application domains ranging from rapid reallocation of task and monitoring effort in information security and survivability to efficient management of distributed logistics tasks.

The project studies problems including:

1. Development of negotiation protocols appropriate to adaptive agents

2. Representation of preference information and the evolution of preferences and beliefs in the course of negotiation

3. Efficient mechanisms for approximating rational decisions when limited by time, memory, communications band width, or other critical resources, and

4. Automatic construction, deployment, and control of artificial negotiating agents.

### Contact

Howard Shrobe, hes@ai.mit.edu
Jon Doyle, doyle@mit.edu
http://www.medg.lcs.mit.edu/projects/ants

(FALL, 1999)

## Digitizing Cities: MIT City Scanning Project

*Leader:*
Seth Teller

*Members*:
Eric Amram, Matt Antone, Mike Bosse, Johann Burgert, George Chou, Eric Cohen, Satyan Coorg, Barbara Cutler, Douglas De Couto, Manish Jethwa, Adam Kropp, Neel Master, J.P Mellor, Jesus Orihuela, Tara Schenkel, Laughton Stanley, Oixiang Sun, Stefano Totaro, and others

Sponsors:
Defense Advanced Research Projects Agency, Intel Corporation,
Lincoln Laboratories' Advanced Concepts Committee, Office of Naval Research-MURI, Interval Research Corporation

### Why Is This Interesting?

The "computer vision" or "machine vision" problem is a long-standing, difficult problem. Fundamentally, it is: how can a computer algorithm, alone or in concert with a human, produce a useful computational representation of a scene, given one or more images of that scene? Fully hundreds of researchers have developed machine vision algorithms over several decades. However, they have employed widely different definitions of the problem, leading to an enormous variety of problem statements and proposed solutions, and even degrees of automation of proposed solutions.

Researchers have developed algorithms to "recognize" common patterns (e.g., road networks) or objects (e.g., vehicles) in images, or identify spatial or other relationships among such entities. Others have sought to deduce physical structure from images; for example, the topography of some imaged land area. Still others have used machine vision techniques to monitor automated processes (e.g., robotic manufacturing systems, assembly lines) and human activities (e.g., immigration areas, mall parking lots). Machine vision systems have been incorporated into autonomous and semi-autonomous vehicles, such as cars and helicopters.

In all of these contexts, the machine vision problem has proved difficult from both theoretical and practical standpoints. Images are fundamentally ambiguous; they flatten into two dimensions, freeze in time, then clip to fit into a frame, a fully 3D, frameless, temporally continuous world. Cameras, whether film or digital, have low resolution, a small field of view, and limited dynamic range and lightness and color discrimination, compared to the complexity of surface detail, lighting variation, and dynamic range of reflected illumination found in the physical world. Lighting conditions in the real world are enormously complex, as are the surface appearances of even common objects. The usual difficulties of implementing robust numerical/geometric algorithms (ill-conditioned and/or degenerate data) only compound the hardness of realizing working machine vision algorithms.

To combat noise and compartmentalize the problem statement, machine vision systems often include humans in the loop, to provide initial guesses to optimization routines, or crucial identification of or matches between features in images, for example. This inclusion of human capabilities can improve the system, but it makes it harder for researchers to draw conclusions about what the algorithmic components of the system are capable of, in contrast to the system considered as a whole. Thus there is difficulty inherent not only in defining the problem, but solving it, and evaluating the proposed solution.

### Contact

Seth Teller, seth@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/city/city.html

## Weathering and Surface Appearance

*Leader:*
Julie Dorsey

*Members*:
Henrik Jensen, Justin Legakis, Hans Pedersen

(Pat Hanrahan, CANON USA Professor, of the Computer Graphics Laboratory in the Computer Science and Electrical Engineering Departments in the School of Engineering at Stanford University is also involved with this project.)

*Sponsors*:
National Science Foundation, Nippon Telegraph and Telephone,
Alfred P. Sloan Foundation Research Fellowship.

### Background

Existing models of materials used in computer graphics assume the materials are in pristine condition. In contrast, real world objects show the effect of aging and weathering. For example, surfaces become bleached, tarnished, painted, scuffed, corroded, knicked, scratched, stained, and otherwise modified under the influence of the environment. These changes in appearance often enhance the value of the object since they emphasize its longevity and history.

The goal of this project is to develop new computer models of materials that can sustain a broad range of appearances.

### Progress

*"Modeling and Rendering of Metallic Patinas"*
*Julie Dorsey and Pat Hanrahan*
*Proceedings of ACM SIGGRAPH '96. Computer Graphics Proceedings*

*"Flow and Changes in Appearance"*
*Julie Dorsey, Hans Pedersen, and Pat Hanrahan*
*Proceedings of ACM SIGGRAPH '96. Computer Graphics Proceedings*

*"Rendering of Wet Materials"*
*Henrik Wann Jensen, Justin Legakis, Julie Dorsey.*
*Proc. of the Tenth Eurographics Workshop on Rendering, 1999*

"Modeling and Rendering of Weathered Stone"
*Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Pedersen, to appear in Proc. SIGGRAPH '99. Computer Graphics Proceedings*

(FALL, 1999)

## Contact

Julie Dorsey, dorsey@graphics.lcs.mit.edu
Henrik Wann, henrik@graphics.stanford.edu
Pat Hanrahan, hanrahan@cs.stanford.edu
Justin Legakis, legakis@graphics.lcs.mit.edu
Hans Pedersen, hkp@paraform.com
http://graphics.lcs.mit.edu/~dorsey/weathering

(FALL, 1999)

## A Simple Distributed Security Infrastructure (SDSI)

*Leaders:*
Ronald L. Rivest, Butler Lampson

*Members:*
Victor Boyko, Dwaine Clarke, Helen Honf, Stanislaw Jarecki, Joseph Kanapka, Moses Listov, Anna Lysyanskay, Tal Malkin, Andrew Maywah, Zulfikar Ramzan, Leo Reyzin, Amit Sahai, Abhi Shelat

*Sponsors:*
Defense Advanced Research Projects Agency, NASA

### Background

SDSI 1.1
A redesign of SDSI, to yield SDSI version 1.1 is underway. The documentation (see references below) lags behind the actual design work considerably; stay tuned for more details. The goals are to simplify the design still further, and to merge the design with Carl Ellison's SPKI work.

SDSI 1.0
Has been designed (see references below). A prototype implementation by Matt Fredette and Gillian Elcock is underway. An initial version of a SDSI server is operational. The user interface has been designed and a protoype has been implemented, which is described in the thesis "A Web-Based User Interface," for SDSI. Another SDSI 1.0 implementation by Wei Dai has been almost entirely completed.

SDSI 2.0
This design represents the merger of SDSI and SPKI. It has a unified treatment of certificates, a coherent treatment of names (both for individuals and for groups), an algebra of "tags" for describing permissions and attributes, and a flexible means of denoting cryptographic keys.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Butler Lampson, blampson@microsoft.com

(FALL, 1999)

## Anonymity in Cryptographic Protocols

*Leaders*:
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Anna Lysyanskaya, Zulfikar Ramzan, Amit Sahai

### Background

Maintaining user anonymity is desirable in a variety of electronic commerce applications. For example, if you were to vote electronically, you probably would not want anyone to know the candidate for whom you voted; or if you were to use electronic cash to purchase a product, you may not want your identity to be known since this information could be used to trace your spending patterns, and perhaps spam you with junk mail. Although achieving anonymity can be an important design criterion in cryptographic systems, it comes at a cost. If the systems are not carefully designed, the overall security of the system could be compromised. Our goal is to develop mathematical techniques that enable anonymity in cryptographic systems without compromising the security. Recent results include the design of Pseudonym Systems and the construction of Group Blind Digital Signatures.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

## Basic Cryptographic Primitives

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Mihir Bellare, Sahai Halevi, Amit Sahai, Salil Vadhan

### Background

Trapdoor Functions and Public-Key Cryptosystems (to appear in Crypto "98), by Mihir Bellare, Shai Halevi, Amit Sahai, and Salil Vadhan. The heart of the task of building public key cryptosystems is viewed as that of "making trapdoors;" in fact, public key cryptosystems and trapdoor functions are often discussed as synonymous. How accurate is this view?

In this paper we endeavor to get a better understanding of the nature of "trapdoorness" and its relation to public key cryptosytems, by broadening the scope of the investigation: we look at general trapdoor functions (i.e. ones that are not necessarily injective). Our first result is somewhat surprising: we show that (non-injective) trapdoor functions can be constructed from any one-way function (and hence it is unlikely that they suffice for public key encryption). On the other hand, we demonstrate that the injectivity condition can be relaxed, by showing that trapdoor functions with polynomial size pre-images are still sufficient for public key encryption.

We then turn our attention to the converse, and provide some evidence that injective one-way functions are necessary for public key encryption by proving that in the random-oracle model one can construct injective trapdoor functions from public key encryption. However, we demonstrate that proving the same without the random-oracle may be difficult, by showing a failure of a "natural" extension of the proof.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Shafi Goldwasser, shafi@theory.lcs.mit.edu

# Incremental Cryptography

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members*:
Victor Boyko, Dwaine Clarke, Helen Honf, Stanislaw Jarecki, Joseph Kanapka,
Moses Listov, Anna Lysyanskay, Tal Malkin, Andrew Maywah, Zulfikar Ramzan,
Leo Reyzin, Amit Sahai, Abhi Shelat

## Background

The goal of incremental cryptography is the design of cryptographic algorithms with the property that having applied the algorithm to a document, it is possible to quickly update the result of the algorithm for a modified document, rather than having to re-compute it from scratch. Incremental cryptography offers considerable advantages in all settings where similar documents are processed by the same cryptographic transformation. For example if you want to send signed copies of the same letter to several people just changing in each letter the name of the recipient, instead of signing each copy from scratch, you can sign only one letter and then obtain the other signatures by fast update operations. Incrementally can be defined for all cryptographic primitives: digital signatures, encryption, hashing, and authentication.

## Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Shafi Goldwasser, shafi@theory.lcs.mit.edu

(FALL, 1999)

## Lattice-Based Cryptography

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Oded Goldreich, S.Halevi, Daniele Micciancio

### Background

Identifying hard computational problems which are amenable for cryptographic use is a very important task. Although hard computational problems seem to be all around us, only very few of those problems were found to be useful for cryptography. In fact, after two decades of research in cryptography, the vast majority of the public-key cryptosystems still depend on either the hardness of integer factorization or the hardness of extracting discrete logarithms. Moreover, often it has been the case that algorithmic advance in one of these problems was then applied to the other one as well.

### Contact

Shafi Goldwasser, shafi@theory.lcs.mit.edu
Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

# Micropayment Research

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Jeffrey Burstein, A. Shamir

## Background

This page presents various research results relating to micropayments that have been discovered in the Cryptography and Information Security Group of MIT's Lab for Computer Science.

## Progress

- PayWord and MicroMint, Two Simple Micropayment Schemes by R.L. Rivest and A. Shamir. (To appear).
- PowerPoint slides for RSA "96 conference.
- An Implementation of MicroMint by Jeffrey Burstein. MIT EECS Master"s Thesis, May 1998.
- Electronic Lottery Tickets as Micropayments (rump session talk given at the Financial Cryptography "97 conference, and to appear in the proceedings of that conference).

## Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

## Pseudo-Randomness in Cryptographic Applications

*Leader:*
Shafi Goldwasser

*Members:*
Mihir Bellare, Daniele Micciancio

### Background

Randomness is a key ingredient for cryptography. Random bits are necessary not only for generating cryptographic keys, but are also often an integral part of steps of cryptographic algorithms. In practice, the random bits will be generated by a pseudo random number generation process. When this is done, the security of the scheme of course depends in a crucial way on the quality of the random bits produced by the generator. Thus, an evaluation of the overall security of a cryptographic algorithm should consider and take into account the choice of the pseudorandom generator. We started a combined study of pseudo-random number generators and cryptographic applications. The intent is to illustrate the extreme care with which one should choose a pseudo random number generator to use within a particular cryptographic algorithm.

Specifically, in [BGM97] Mihir Bellare from UCSD and CIS members Shafi Goldwasser and Daniele Micciancio consider a concrete algorithm, the Digital Signature Standard, and a concrete pseudo random number generator, the linear congruential generator (or truncated linear congruential pseudo random generators) and show that if a LCG or truncated LCG is used to produce the pseudo random choices called for in DSS, then DSS becomes completely breakable. Slides on this work are available online.

### Contact

Shafi Goldwasser, shafi@theory.lcs.mit.edu

(FALL, 1999)

## The Random Oracle Model

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Ran Canetti, Oded Goldreich,
Shai Halevi, Daniele Micciancio, Omer Reingold

### Background

A popular methodology for designing cryptographic protocols consists of the following two steps. One first designs an ideal system in which all parties (including the adversary) have oracle access to a truly random function, and proves the security of this ideal system. Next, one replaces the random oracle by a "good cryptographic hashing function" (such as MD5 or SHA), providing all parties (including the adversary) with the succinct description of this function. Thus, one obtains an implementation of the ideal system in a "real-world" where random oracles do not exist. This methodology, explicitly formulated by Bellare and Rogaway, and hereafter referred to as the random oracle methodology, has been used in many works.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Shafi Goldwasser, shafi@theory.lcs.mit.edu

(FALL, 1999)

## Threshold Cryptology

*Leaders:*
Shafi  Goldwasser, Ronald L. Rivest

*Members:*
Victor Boyko, Dwaine Clarke, Helen Honf, Stanislaw Jarecki, Joseph Kanapka,
Moses Listov, Anna Lysyanskay, Tal Malkin, Andrew Maywah, Zulfikar Ramzan,
Leo Reyzin, Amit Sahai, Abhi Shelat

### Background

The idea of theshold cryptography is to protect information (or computation) by fault-tolerantly distributing among a cluster of cooperating computers. "Secret sharing" is an example of a threshold cryptography application, in which a valuable information is protected by being distributed among a cluster of computers in such a way so that even if some threshold of these computers are faulty the others can still reconstruct it, and on the other hand, the adversary trying to learn the information needs to break into some threshold of these computers to learn anything. Hence, secret sharing maintains both availability and secrecy of information against attacks by adversaries who can break into (disable, spy on, modify the information of, etc...) some threshold (say, up to a half) of the available computers.

Another interesting example of threshold security is "function sharing", i.e. protecting the availability and secrecy of a computer which performs some highly sensitive operation. A good example is a Network Certification Authority: a computer which (hopefully some day soon in the future) will sign public key credentials of every user on the internet.  Instead of trusting a single computer to do this, it would be much more secure if we could fault-tolerantly distribute this operation among a cluster (say, five or seven) of computers, so that an adversary who somehow breaks in (spies on, disables, controls, modifies, whatever!) to some threshold of these computers, still cannot either make proper signatures on his own, or stop the rest of the computers from creating proper signatures on our demand.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

# A New High-level Hardware Design Flow Based on Term Rewriting Systems

*Leaders:*
Arvind, Srinivas Devadas,Martin Rinard

*Members:*
James Hoe, Mieszko Lis,
Daniel L. Rosenband, Xiaowei Shen, Michael Sung

*Sponsors*:
Intel Corporation, National Science Foundation

## Background

We propose a novel high-level hardware design flow that will facilitate architectural exploration and dramatically reduce hardware design time. The design flow is based on a formalism known as Term Rewriting System (TRS). The central idea is to raise the level of hardware design abstractions and to make greater use of automated synthesis and verification tools. We seek not only to reduce the amount of tedium experts face in designing systems, but also to provide tools to assist novice designers.

## Why Is This Interesting?

In our proposed TRS-based high-level architectural design flow, a hardware designer would spend the majority of his other time and effort in producing and debugging a high-level architectural specification. The debugging will be interactive and at a high level of abstraction using automatically generated simulators and computer-aided proof systems. From a known-to-be-correct specification, multiple revisions of TRS descriptions containing various performance optimizations can be quickly generated both automatically and semi automatically under human direction. The correctness of the optimized descriptions may follow by construction or may be checked against the initial architectural specification using computer aided verification. This will avoid a lengthy "hit-and-miss" validation process. The designer may iterate the process using tools designed to provide feedback regarding power, area and other metrics of interest. Finally the designer may select a design for more detailed synthesis. The potential reduction in time, effort and risk will enable hardware solutions to become competitive in many embedded applications, where currently software-on-DSP's is used as an engineering compromise.

## Approach

We are developing the tool set necessary to realize this design flow. The major areas of work in our proposed design flow include: Design capture in a TRS-based language; computer-assisted design methodology; source-to-source architectural transformation; formal design verification of source transformations; automatically generated simulators with instrumentation; feedback-driven architecture exploration; and automatic hardware synthesis.

(FALL, 1999)

## Progress

During the initial exploration of this research, we have developed and studied a number of TRS hardware descriptions at different levels of abstraction. We have had success in describing both complicated modern out-of-order superscalar processors as well as cache-coherence protocols that can be combined with processor models to yield complete system descriptions. These complex models all have been formally verified against their abstract functional specification. We have also studied these models using mechanically hand-translated simulators. During these studies, we have paid particular attention to the synthesizability of various aspects of the TRS descriptions. Along the way, we have formulated the basic synthesis strategy that we are currently implementing in a compiler. In addition, we are working on source-to-source transformations that can automatically introduce performance-enhancing architectural features such as pipelining, superscalar and speculative execution.

## Future

In the coming year, we will focus on hardware synthesis algorithms, pipelining, superscalar and VLIW transformations, and on testing the suitability of our concrete syntax for TRS language. We expect to have an initial system that can generate simulators and RTL hardware descriptions from TRS within a year. In the second and third years, we will continue to incorporate the more advanced features and increase the automation in the process. Ultimately, we will release the tools for testing in actual hardware development project.

## Contact

Arvind, arvind@lcs.mit.edu
http //www.csg.lcs.mit.edu/~arvind

(FALL, 1999)

## Concerning Malleable Cache for Data-Intensive Computing

*Leaders:*
Arvind, Larry Rudolph

*Members*:
Jan-Willem Maessen, Jacob Schwartz, Xiaowei Shen

*Sponsors*:
Defense Advanced Research Projects Agency

### Background

Caches have become an ubiquitous feature of general-purpose microprocessors, however, their organization is not always suitable for data-intensive applications. In particular, the locality of reference assumption and the LRU replacement strategy of traditional caches are wrong for the stream-based and real-time components of data-intensive applications. Moreover, future microprocessors are likely to have even larger caches which will consume a large fraction of chip area and power. For many application domains, however, the marginal utility of larger on-chip caches is decreasing.

Making caches act just like a fast, on-chip memory is not a workable solution, since it is not backward compatible and forces compilers to be specially tailored to a particular processor generation. Making caches act like a large register file is also not a general solution. Traditional cache control instructions for flushing and prefetching help, but do not provide enough control.

### Why Is This Interesting?

Caching has been around for a long time and has been well studied. However, as computers and their use evolve, so should the cache. It is always interesting to re-examine and improve long standing technology.

Mechanisms that adapt to the existing workload are usually more interesting than purely static ones, especially when they perform better. We begin by examining various uses of the valuable caching resource with the expectation that we will move-on to examining malleable processors as well.

### Approach

We propose novel ways of controlling caches to dramatically improve their utility and power consumption. Such *malleable caches* are suitable for a wide range of data-intensive applications. Small performance-critical code sequences of stream-processing and real-time applications can achieve several orders of magnitude improvement with malleable caches. Applications with low information entropy can make use of the cache resources for data compaction or memoized functions to dramatically improve execution time. In addition, malleable caches permit many architectural optimizations, such as power management and adaptive prefetching.

(FALL, 1999)

In particular, cache performance can be improved by partitioning the cache so that competiting data accesses do not cause thrashing. Since the traditional LRU replacement strategy only looks at the near term past behavior, the wrong elements can be replaced. Cache partitioning can isolate elements so that they do not interfere. This idea is not new. Caches have been partitioned for data and instructions for a long time, and recently they have been partitioned for temporal and spacial data. Page coloring is a technique that indirectly partitions a cache, especially one that is direct mapped. This work proposes a mechanism to dynamically partition caches, that has low overhead both in terms of implementation and in the cost of repartitioning. This mechanism, referred to as "column caching," is explained in the following section.

## Progress

First-level caches and some second-level caches are both direct-mapped and associative; the most significant bits of the address specify a set of entries, while other bits of the address are used to select one of the entries by comparing tags. One can view a cache with a set associative size $k$ as consisting of $k$ columns, one for each element of the cache set, and $r$ rows, one for each direct mapped set. In a traditional cache, a new cache line replaces (or evicts) the least recently used (LRU) entry in a set. Although some systems use an approximation to LRU because of implementation constraints, we shall still refer to this as an LRU replacement strategy.

We have extended the functionality of the cache by limiting the potential destination for a new entry; a bit vector specifies the columns an entry can occupy. If several columns are specified, a new entry replaces the least recently used entry among just those potential columns. Cache lookup behaves in exactly the same way as traditional caches in that all columns are searched; the only change occurs when there is a cache miss. Specifying disjoint sets of columns for different address ranges allows the cache to be partitioned. Accesses to an address in one partition will not pollute entries in a different partition.

We specify the address ranges on a page by page basis. That is, each page of memory has a set of $k$ bits, one for each column of the cache. If the $i$-th bit is set, then data items in this page may be placed in column $i$. These bits are included in the TLB. Thus, during a cache miss, the replacement bits of the containing page are taken from the TLB and used to identify the permissible columns. For backward compatibility, a page would specify all the cache columns. To ensure that no other thread, process, or job will evict a job's cache values, it can be given exclusive access to a set of columns.

There are many application scenarios in which a small task is periodically executed. Suppose the task updates a small amount of memory each time it executes. Typically, the data will need to be brought into the cache, modified, and then later written back to memory. These types of tasks are very damaging for traditional caches, since each time they execute, the cache is "cold." There is a large startup cost in bringing entries into the L2 cache and then the L1 cache, and evicting entries from the L1 cache. Modern microprocessors are not built for such activity, they assume that data brought into the L1 cache will be accessed many times.

We have built a simulation framework that allows us to simulate large traces on column caches with varying sizes. We have run a compression (gzip) benchmark, and a ray-tracing benchmark among others. Initial results indicate that column caching can substantially improve performance.

## Future

Modern "snoopy" caches react to bus accesses in addition to processor accesses. When the address of a bus action is contained in the cache, the cache entry may be invalidated in the case of a bus write operation, or may be used to satisfy the bus request in the case of a bus read of a modified cache entry. We propose to extend functionality when there is a bus access to an address that is not contained in the cache, but is contained within a specified region of memory for which the cache is curious. That is, associated with the cache is a set of *curious* address ranges.

A bus write operation to a curious address will cause the cache to be updated with the data on the bus. It is possible that the fetch operation will cause another cache value to be evicted. To avoid deadlocks, cache elements that have just been updated from the bus will not be candidates for eviction until they have been referenced by the attached processor. By declaring a region of memory to be curious, the cache is stating that its processor will likely reference that data in the near future. Note that this is similar in spirit to the ability to "pin"' virtual pages in physical memory so that an application does not suffer page faults.

The resulting cache state for curious address ranges can also be specified. For standard MESI cache protocols, either modified, exclusive, or shared states can be specified. If exclusive or shared is specified, then the cache simply latches the bus data and updates its cache. If modified is specified, then the cache controller may also prevent the memory from updating its value for that location.

## Contact

Arvind, arvind@lcs.mit.edu
http: //www.csg.lcs.mit.edu/

(FALL, 1999)

## Memory Models for Architects and Compiler Writers

*Leaders*:
Arvind, Larry Rudolph

*Members:*
Jan-Willem Maessen, Xiaowei Shen, Jacob Schwartz

*Sponsors*:
National Science Foundation

### Background

The memory model of a microprocessor is the semantics of load and store instructions. Recently, many "weak memory models" have arisen as a consequence architectural optimization, rather than from some grand high-level design. Many of these optimizations are transparent for uniprocessors but can be observed in multiprocessor because they break the Sequential Consistency model in one way or another. As a partial remedy all modern microprocessors now provide memory fences to control instruction reordering and overload their semantics to imply something vague about store completion.

CRF, our recently proposed mechanism-oriented memory model, can be used as the universal interface between the compiler writer and the underlying architecture. It also enables the development of sophisticated cache coherence protocols.

### Why Is This Interesting?

Many people have difficulty understanding why and how a memory model can affect them. Does it have impact on the way they write programs? Does it improve the performance? Does it affect the compiler or microarchitecture optimizations? The simple answer is that if we stick to uniprocessor systems and sequential programming the memory model is irrelevant. But, we rarely do! Even sequential programmers may initiate input/output tasks that run concurrently with the main program, and uniprocessors may have operating systems that are expressed as "cooperating sequential tasks." The fact is that the memory model affects everything but often in insidious and surprising ways.

Many architectural mechanisms that are transparent in uniprocessors suddenly become visible in multiprocessors. A difference in implementation of memory related instructions could cause a subtle difference in observable behavior, giving rise to a different memory model. That is, the range of permissible behaviors is affected by just about every uniprocessor optimization. The fact that these range of behaviors have a lot of commonality or that the simple sequential execution of instructions is always a permitted behavior is hardly a source of comfort, because it is the additional behaviors that make parallel programs go wrong in surprising ways. Therefore, there is a need to pin down the exact definitions of memory instructions for multiprocessors. Since the same microprocessors and operating systems are used in uniprocessors and multiprocessors, a microprocessor designer has no option but to pay close attention to these subtle differences.

The design of cache coherence protocols plays an important role for shared memory systems and is intimately tied to the memory model. There have been many attempts to develop adaptive cache coherence protocols, but it is very difficult to verify the correctness of all but the simplest ones. A formal memory model that captures the semantics of a cache makes the verification task easier.

## Approach

We investigate a new mechanism-oriented memory model called Commit-Reconcile & Fences (CRF), which exposes both data replication and instruction reordering at the ISA level. There are good reasons to be skeptical of yet another memory model. It is essential to have upward compatibility, i.e. the ability to run existing programs correctly on a machine with the CRF model. Furthermore, if compiler writers adopt CRF as the target model then it is important to have downward compatibility, i.e. the ability to run CRF programs well on existing machines. Indeed, CRF has both of these properties vis-a-vis most other existing memory models.

Our model covers the spectrum from uniprocessor to SMP to DSM. CRF semantics permit most, if not all, current architectural optimizations. The semantics are clear and precise so that there are no fuzzy cases. By issuing memory access instructions out-of-order, architectural mechanisms, such as register renaming and speculative execution, can affect the observed program behavior in a multiprocessor setting. To date such properties of instruction issue have not been specified precisely. However, it is often possible to reason about program behaviors by making conservative assumptions, e.g. no speculation about instruction issue.

## Progress

The CRF model exposes data replication via a notion of semantic cache (sache). Each processor is associated with a sache, on which memory access instructions are performed. Each sache cell has an associated state, which can be either Clean or Dirty. The Clean state indicates that the data has not been modified since it was cached or last written back. The Dirty state indicates that the data has been modified and has not been written back to the memory since then.

In CRF, conventional load and store operations are decomposed into finer-grain instructions. There are five memory-related instructions, Loadl (load-local), Storel (store-local), Commit, Reconcile and Fence. The CRF model can be specified by two sets of rules, the CR rules and the reordering rules.

An adaptive cache coherence protocol changes its actions to address changing program behaviors. We developed an adaptive protocol called "Cachet" for distributed shared-memory systems. Cachet is a seamless integration of several micro-protocols, each of which has been optimized for a particular memory access pattern. Cachet embodies both intra-protocol and inter-protocol adaptivity, and exploits adaptivity to achieve high performance under changing memory access patterns. A protocol to implement CRF is automatically a correct implementation of any memory model whose programs can be expressed as CRF programs.

The rules of each micro-protocol and Cachet are classified into two sets: mandatory rules and voluntary rules. The main distinction between mandatory and voluntary rules is that mandatory rules often require certain fairness to ensure the liveness of the system, while voluntary rules have no such requirement and are provided purely for adaptivity and performance reason. A voluntary action is usually enabled when the cache or memory cell is in an appropriate state. It can be initiated at either the cache or memory side. Exact details of voluntary actions may vary for different protocols. The existence of voluntary rules provides enormous intra-protocol and inter-protocol adaptivity.

Heuristic messages and soft states can be used as hints to invoke desired adaptive actions. A heuristic message is a suggestion that some voluntary action or protocol switch be invoked at a remote site. Soft states can be used later as hints to invoke local voluntary actions or choose between different micro-protocols.

## Future

The success of CRF in developing verifiable, adaptive, cache coherent protocols has given us confidence that attack more difficult problems. In particular, we are looking to use CRF to precisely capture the Java memory model. We also wish to extend our methodology to fully model a modern microprocessor, such as Intel P6. Our approach to protocol development and verification that classifies rules as either voluntary or mandatory appears to have wide applicability. We expect to apply it to a wide range of protocols. Finally, we hope to integrate this work with adaptive, malleable cache hardware such as column and curious caching.

## Contact

Arvind, arvind@lcs.mit.edu
http://www.csg.lcs.mit.edu/

(FALL, 1999)

## SFS, the Secure File System

*Leaders:*
David Mazières, Frans Kaashoek

*Members:*
Michael Kaminsky, Kevin Fu, Emily Sit

*Sponsors:*
Intel Corporation, National Science Foundation

### Background

SFS, the Secure File System, creates a global, secure filesystem with a single global namespace and no centralized control. No secure network file system has ever grown to span the Internet. Existing systems all lack adequate key management for security at a global scale. Given the diversity of the Internet, any particular mechanism a file system employs to manage keys will fail to support many types of use.

We propose separating key management from file system security, letting the world share a single global file system no matter how individuals manage keys. We present SFS, a secure file system that avoids internal key management. While other file systems need key management to map file names to encryption keys, SFS file names effectively contain public keys, making them *self-certifying pathnames*. Key management in SFS occurs outside of the file system, in whatever procedure users choose to generate file names.

Self-certifying pathnames free SFS clients from any notion of administrative realm, making inter-realm file sharing trivial. They let users authenticate servers through a number of different techniques. The file namespace doubles as a key certification namespace, so that people can realize many key management schemes using only standard file utilities. Finally, with self-certifying pathnames, people can bootstrap one key management mechanism using another. These properties make SFS more versatile than any file system with built-in key management.

### Contact

Frans Kaashoek, Kaashoek@lcs.mit.edu

(FALL, 1999)

## The  Exokernel

*Leader:*
Frans Kaashoek

*Members:*
Hector M. Briceno, Dawson Engler, Gregory Ganger, Robert Grim, Russell Hunt,
John Jannotti, Kenneth Mackenzie, David Mazieres, Thomas Pinckeny

*Sponsors:*
Defense Advanced Research Projects Agency,
Intel Corporation, National Science Foundation

### Background

The exokernel, a novel operating system architecture, is the frame work of much of our recent research. An exokernel eliminates the high-level abstractions most of today's operating systems are built on, instead concentrating on securely and efficiently multiplexing the raw hardware. A more conventional operating system interface is provided by application-level libraries. This allows untrusted applications to safely override default OS functionality for extensibility, flexibility, or performance.

We have built an exokernel system that allows specialized applications to achieve high performance without sacrificing the performance of unmodified UNIX programs.  We evaluated the exokernel architecture by measuring end-to-end application performance on Xok, an exokernel for Intel x86-based computers, and by comparing Xok's performance to the performance of two widely-used 4.4BSD UNIX systems (Free BSD and Open BSD).  The results show that common unmodified UNIX application scan enjoy the benefits of exokernels  applications either perform comparably on Xok/ExOS and the BSD UNIXes, or perform significantly better. In addition, the results show that customized applications can benefit substantially from control over their resources (e.g., a factor of eight for a Web server).

### Contact

Frans Kaashoek, kaashoek@lcs.mit.edu

(FALL, 1999)

## HAC: Hybrid Adaptive Caching for Persistent Object Systems

*Leader*:
Barbara Liskov

*Sponsors:*
Defense Advanced Research Projects Agency, Office of Naval Research,
Praxis XX1 Fellowship (Mr. Miguel Castro)

### Background

Object-oriented programming environments are widely used and persistent object systems provide a convenient interface for manipulating persistent objects in these environments. Effective caching of small persistent object systems in main memory is crucial to achieve good performance in persistent object systems. Hybrid Adaptive Caching (HAC) is a new caching technique which addresses this issue.

### Why Is This Interesting?

Persistent object systems provide a generic and convenient implementation of persistency. They relieve programmers from the burden of implementing ad hoc versions of persistency, which is time consuming, error prone, and prevents widespread sharing of persistent objects. However, previous persistent object systems were inefficient due to the lack of effective techniques for caching small persistent objects in main memory. HAC solves the problem; it outperforms all previous techniques by more than an order of magnitude.

### Approach

Previous caching techniques for persistent object systems can be classified as either page caching or object caching. Page caching fetches fixed-size pages into the cache and discards full pages to make room for fetched pages. It has a low cache management overhead but it can waste cache space when spatial locality is poor; i.e., when only a small fraction of the objects in each page is used. Existing studies show that it is hard or impossible to cluster objects into pages to achieve good spatial locality. Therefore, object caching was designed to allow discarding the objects in a page that are not used while retaining the ones in use. The problem with object caching is that it incurs high space and time overhead to manage the cache. HAC is a hybrid between page and object caching that combines the virtues of both while avoiding their disadvantages. Like page caching, HAC has low cache management overhead, and like object caching, it can discard pages while retaining their objects of interest.

HAC's good performance is mostly due to its adaptive cache replacement technique when spatial locality is good, it behaves like page caching, avoiding the overheads of object caching where it has low benefit. If spatial locality is poor, it behaves like object caching, taking advantage of the low miss rates to offset the increased overheads. HAC partitions the client cache into page frames and fetches entire pages into the cache. To make room for an incoming page, HAC selects some page frames for compaction, discards the cold objects in these frames, and compacts the hot objects to free one of the frames. The policy that drives this mechanism strives both to achieve low overhead and to discard objects that are unlikely to be reused. This policy is implemented using hierarchical usage statistics. HAC maintains usage statistics on a per-object basis and the usage statistic of a page is a function of the objects it contains. Per-object usage statistics are maintained using a novel technique with low space and time overheads. The technique combines both recency and frequency information to make good replacement decisions.

The high performance of HAC is furthermore achieved by adherence to three important design principles think small, be lazy, and work while waiting. We were careful about the sizes of all data structures, but particularly about objects; we designed our object format parsimoniously, and the result is better performance in all parts of the system, from the server disk to the client cache. Furthermore, we do work only when absolutely necessary, since this allows us to avoid it entirely in many cases. Finally, we perform most of the work while waiting for fetches to complete.

## Progress

HAC has been fully implemented in the Thor client/server object-oriented database. We have performed a detailed performance analysis of this implementation including a comparison with the best previously proposed caching techniques. This comparison showed HAC performs more than an order of magnitude better. These results were published by Miguel Castro, Atul Adya, Barbara Liskov, and Andrew C. Myers,

## Future

The existing implementation of HAC provides support for the object-oriented programming language Theta, which was designed by our group. Work is currently under way to support Java and provide an efficient implementation of persistency for Java objects on the Internet. Work is also underway to integrate HAC with a cooperative caching technique developed by our group. Cooperative caching will further improve performance by allowing different machines to share their caches.

## Contact

Barbara Liskov, liskov@lcs.mit.edu
Miguel Castro, castro@lcs.mit.edu
http //www.pmg.lcs.mit.edu

(FALL, 1999)

## Practical Byzantine Fault Tolerance

*Leader:*
Barbara Liskov

*Sponsors*:
Defense Advanced Research Projects Agency, NEC, Praxis XX1 Fellowship (Mr. Miguel Castro), United States Air Force

### Background

Fault-tolerance techniques enable systems to tolerate faults in some of their componenets. Byzantine fault tolerance makes no assumptions about the behavior of faulty components. For example, it allows systems to tolerate components that exhibit malicious behavior. In contrast, most fault-tolerance techniques assume that components fail benignly by not executing steps.

### Why Is This Interesting?

The growing reliance of industry and government on online information services makes malicious attacks more and more likely to occur. Furthermore, the number of software errors is increasing due to the growth in size and complexity of software systems. Both malicious attacks and software errors can cause faulty nodes to exhibit arbitrary behavior, that is, to behave in a byzantine manner. Therefore, the assumption that most faults are benign, which underlies conventionally fault-tolerant techniques, will soon be inadequate, and byzantine-fault-tolerant algorithms will be needed instead.

### Approach

Previous byzantine-fault-tolerance algorithms were not practical because they assumed a synchronous system, were too slow to be used in practice, or required an excessive amount of resources. Our research focuses on practicality. Its goal is to design, and implement new practical byzantine-fault-tolerance algorithms that can be widely used and to evaluate their performance in real systems.

We have combined cryptography with techniques from asynchronous environments like the Internet and incorporates several important optimizations that improve the response time of previous algorithms by one to two orders of magnitude.

We have also implemented the algorithm in a replication library that can be used to replicate existing services in a byzantine-fault-tolerant way with minimal changes to code that implements those services. We used the library to implement a replicated version of the NFS distributed file system and evaluated its performance. Our results demonstrate that the algorithm is practical; the response time of the replicated NFS in the Andrew benchmark is within 20% of the response time of a standard unreplicated NFS.

(FALL, 1999)

## Progress

We have designed the first practical Byzantine-fault-tolerant replication algorithm and implemented it. We have also used the algorithm to implement a widely used service and we evaluated its performance. These results were published in

## Contact

Barbara Liskov, liskov@lcs.mit.edu
Miquel Castro, castro@lcs.mit.edu
http //www.pmg.lcs.mit.edu

(FALL, 1999)

# Weak Consistency Models and Mechanisms for Scalable Transactional Systems

*Leader*:
Barbara Liskov

## Background

Large distributed systems are expected to be common in the future, in office environments and over wide-area networks. Providing data consistency in such environments can be difficult since there is a higher likelihood of conflicts and the cost of handling these conflicts can be high. Thus, we need mechanisms that provide data consistency efficiently while providing application programmers with a simple model of computation.

## Why Is This Interesting?

Some applications may not need strong consistency guarantees and running them at consistency levels such as serializability may unnecessarily penalize performance. To address this problem, existing database systems provide different consistency levels

## Approach

Our first contribution is to redefine the existing degrees of isolation to allow a range of concurrency control mechanisms including optimism and locking. Furthermore, unlike previous definitions, our specifications allow different guarantees to be provided for running and committed transactions, permitting more efficient consistency schemes.

We have also defined two new consistency levels that fill the gap between degree 2, which does not provide consistency, and degree 3, which provides full serializability. Degree 2L supports legacy applications by capturing certain guarantees provided by locking. Degree 2+ provides consistency, so that application transactions can depend on database invariants; we believe it is the weakest definition that supports consistency.

Our second contribution is in the area of efficient optimistic implementations. We have designed and implemented new optimistic schemes that provide degree 2+ and other isolation degrees in a client-server distributed system. These schemes use multipart timestamps or multistamps to capture the consistency information. However, in general, multistamps do not scale well in large systems. To solve this problem, we have developed a novel multistamp compression technique that takes advantage of loosely synchronized clocks for removing old information in multistamps.

## Progress

We have redefined the existing degrees of isolation for committed transactions, and we have defined two new isolation degrees, 2L and 2+. We have implemented various isolation degrees in a simulator and our initial performance studies show that our multistamp truncation technique allows us to use small multistamps (e.g., 60-70 bytes) in a large distributed systems to capture the consistency information efficiently.

## Future

We plan to continue our work on defining degrees of isolation. Two areas are of interest: extending the definitions to take account of interactions among uncommitted transactions, and developing definitions that work with abstract operations instead of simply reads and writes.

We also plan to continue our implementation studies. Our ultimate goal is to develop a thorough understanding of the performance benefits that are possible through the use of lower isolation levels. We are developing new, high-conflict workloads that ought to provide an advantage to lower levels; it remains to be seen what their impact on performance will be. Additionally, we are continuing to develop new implementation techniques; we plan to include the best of them in the Thor object-oriented database system, so that users of Thor can choose the levels that are right for their applications.

## Contact

Atul Adya, adya@lcs.mit.edu
Barbara Liskov, liskov@lcs.mit.edu
http://www.pmg.lcs.mit.edu

(FALL, 1999)

# Protecting Privacy with Mostly Static Analysis

*Leaders:*
Barbara Liskov, Andrew Myers

## Background

Privacy and secrecy are becoming increasingly important to protect as sensitive information is stored on network-accessible computers, and untrusted code is used more frequently (e.g., downloaded Java applets). New security mechanisms are needed to protect against leaks of private information.

## Why Is This Interesting?

Existing security mechanisms do not provide adequate support for protection of privacy or secrecy. Common mechanisms such as sand boxing and access control (discretionary or mandatory) are either too restrictive -- they prevent applications from sharing data usefully -- or too weak. We have developed a new technique for analyzing programs statically (at compile time) to determine whether they follow certain simple rules. Programs that follow these rules cannot leak information through covert storage channels. Since enforcement of the rules is almost entirely through static analysis, the resulting programs execute efficiently, with little compile-time or run-time overhead.

## Approach

We define user-supplied program annotations, called _labels_, that describe the allowed flow of information in a program. Annotated programs can be checked at compile time, in a manner similar to type checking, to ensure that they do not violate information flow rules. Compile-time checks have no run-time overhead in space or time, and unlike run-time checks, when they fail, they do not leak information about the data the program is using. Our label model improves on existing models by allowing individuals to declassify data they own, rather than requiring a central authority to do it. It can support the privacy concerns of multiple principals simultaneously, even in the presence of mutual distrust.

The new language _JFlow_ (an extension to Java) allows programs to be statically checked for information leaks by an extended Java compiler. In JFlow, variables and objects are annotated with statically-checked dissemination labels. These labels often can be automatically inferred, so annotating programs is not onerous. An explicit form of declassification provides a safe escape hatch when the amount of information leaked is acceptable to the programmer. Safe dynamic checks also may be used when static checks are insufficient. There is little code space, data space, or run time overhead, because most checking is performed statically.

JFlow extends previous static checking models in three ways. First, it introduces an implicit form of parametric polymorphism, called _label polymorphism_, to express procedures that are generic with respect to the security labels of their arguments, or with respect to the principal on whose behalf the procedure executes. Second, since purely static analysis would be too limiting for structures like file systems

(FALL, 1999)

(where information flow cannot be verified purely statically), JFlow includes new secure run-time escape hatch for these structures, with explicit run-time label checks. Uses of the run-time information flow mechanism are still partially verified statically, to ensure that they do not leak information. Finally, despite these features, the labels of local variables in JFlow programs can be inferred automatically, easing the job of adding flow annotations to a program.

## Progress

We define user-supplied program annotations, called _labels_, that describe the allowed flow of information in a program. Annotated programs can be checked at compile time, in a manner similar to type checking, to ensure that they do not violate information flow rules. Compile-time checks have no run-time overhead in space or time, and unlike run-time checks, when they fail, they do not leak information about the data the program is using. Our label model improves on existing models by allowing individuals to declassify data they own, rather than requiring a central authority to do it. It can support the privacy concerns of multiple principals simultaneously, even in the presence of mutual distrust.

The new language _JFlow_ (an extension to Java) allows programs to be statically checked for information leaks by an extended Java compiler. In JFlow, variables and objects are annotated with statically-checked dissemination labels. These labels often can be automatically inferred, so annotating programs is not onerous. An explicit form of declassification provides a safe escape hatch when the amount of information leaked is acceptable to the programmer. Safe dynamic checks also may be used when static checks are insufficient. There is little code space, data space, or run time overhead, because most checking is performed statically.

JFlow extends previous static checking models in three ways. First, it introduces an implicit form of parametric polymorphism, called _label polymorphism_, to express procedures that are generic with respect to the security labels of their arguments, or with respect to the principal on whose behalf the procedure executes. Second, since purely static analysis would be too limiting for structures like file systems (where information flow cannot be verified purely statically), JFlow includes new secure run-time escape hatch for these structures, with explicit run-time label checks. Uses of the run-time information flow mechanism are still partially verified statically, to ensure that they do not leak information. Finally, despite these features, the labels of local variables in JFlow programs can be inferred automatically, easing the job of adding flow annotations to a program.

## Future

The existing work protects against covert storage channels in untrusted programs, assuming a trusted execution environment. Interesting extensions that we are investigating include integration of this model with support for additional features: access control, integrity, covert timing channels, and distributed systems that include untrusted computers.

## Contact

Andrew Myers, andru@lcs.mit.edu
Barbara Liskov, liskov@lcs.mit.edu
http://www.pmg.lcs.mit.edu

(FALL, 1999)

## Support for Widely-Distributed Inter-Enterprise Applications

*Leader*:
Barbara Liskov

### Background

This project will define and implement a new platform that enables development and execution of robust global applications. We assume an environment in which applications running at client machines make use of persistent information; e.g., files or databases, stored at servers. Typically an application has code running at many clients; e.g., at different geographic locations. The application programs at different clients share information stored at the servers. Many applications fit naturally into this model, including banking and medical information systems. We are particularly interested in applications that span multiple organizations, such as electronic commerce. Also, we want to support geographically distributed environments, in which system components may be physically far away from one another, and very large systems; e.g., with tens of thousands of servers and hundreds of thousands of clients.

### Why Is This Interesting?

The growth of the Internet and the Web is leading to global applications with geographically distributed components that span multiple organizations. Support for such applications is inadequate at present. For example, the Web provides no consistency guarantees when applications require access to mutable online information stored at multiple locations, nor is there adequate support for applications that must continue to provide service in the presence of failed components misbehaving in arbitrary ways. To address these inadequacies, fundamental improvements are needed. Our new platform will provide the needed support.

### Approach

Our goal is to make it as easy as possible for global, inter-enterprise applications to be implemented. We plan to accomplish this by (1) defining as the platform interface a simple programming model that hides the underlying complexity from programmers yet provides sufficient power; and (2) providing a high performance implementation of the platform. The high performance implementation is essential since otherwise programmers will not be willing to implement applications on top of the platform. Additionally, the platform will provide robust support so that applications can survive both failures and malicious attacks.

The programming model will provide transactional access to the shared persistent store plus a new mechanism that allows convenient and efficient transactional communication between components in different organizations. Transactions simplify implementing applications because they hide details of concurrency control and atomicity, allowing programmers to focus on the service the application provides to its users.

Our model will augment the conventional transactional model in two ways. First is support for global computations that visit many geographically distributed components spanning multiple organizations. Such computations occur as a linked sequence of transactions. We are investigating the best support for this model, which might take the form of atomic queues or atomic agents. Second is support for weaker transactional semantics so that application developers can choose the support that matches their needs and avoid the expense of stronger semantics when it is unnecessary.

## Progress

Earlier Darpa-supported research led to an object-oriented database system called Thor that will provide the basis for our work. Thor provides a transactional persistent store, and it has a distributed, client/server implementation that provides excellent performance. Its implementation is based on three important innovations: AOCC, a very efficient optimistic concurrency control mechanism; HACS, a new hybrid and adaptive system for managing the content of the client cache; and the MOB architecture for managing storage at the server. These techniques together enable Thor to achieve performance at least an order of magnitude better than previous systems.

## Future

The new research will extend the Thor model of computation, and the Thor implementation, to provide efficient support for global applications. The research has four major components. First is a new model of computation that provides explicit support for Inter-enterprise applications via atomic queues or atomic agents, and that also allows programmers to make use of weaker consistency levels when appropriate. Second is an extension of HACS to provide support for multi-client caching in which a group of clients can cooperate to share their joint cache. Third is extended support for replication in the presence of fail stop failures, and fourth is development of new replication protocols that survive byzantine failures, thus allowing the system to survive malicious attacks.

## Contact

Barbara Liskov, liskov@lcs.mit.edu

(FALL, 1999)

# Alcoa/Alloy: New Technology for Analyzing Software Designs **Design with Object Models**

*Leader*:
Daniel Jackson

*Members:*
Ian Schechter, Benjamin Self, Ilya Shlyakhter, Mandana Vaziri, Allison Waingold

*Sponsors:*
CIPD, National Science Foundation

## Background

The object model is the key design representation in most object-oriented methods. It shows what objects there are and how they are connected. In the requirements and specification phases, object models are essential for ensuring that the right domain notions have been identified, and that the complexities of their relationships have been captured correctly. In the design phase, object models are essential for ensuring that the system hangs together, that good abstractions have been chosen, and that invariants that relate objects are correctly established.

## Why Is This Interesting?

Rationals Unified Modeling Language (UML) is the de facto standard for object modelling, and has been widely adopted. Alloy, our new object modelling notation, was designed to be as compatible with UML as possible, while overcoming its flaws. UML is heavily influenced by C++, and UML encourages the designer to employ its programming language-specific features. Alloy is programming-language-independent, so an Alloy design can be implemented in any language. UML does not have a precise semantics, so the meaning of a UML model may be ambiguous and obscure, and rigorous analysis is impossible. Alloy has a mathematical semantics that is exploited by Alcoa, its fully automatic checker, and which results in a simpler and more coherent set of features. UMLs graphical and textual notations do not fit together smoothly, and every UML model must include a graphical component. Alloys graphical and textual notations were designed together. The graphical notation is a strict subset of the textual notation, making it easy to exchange graphical models in textual form, and to integrate both notations in analysis. UMLs textual notation, the Object Constraint Language (OCL), has an unconventional syntax and many of the complexities of a programming language. Rules involving iterated navigations are hard to express (eg, that in a file system every file is reachable from the root). Alloy uses a standard logical syntax, is much simpler than a programming language, and has transitive closure built-in.

*Alcoa Alloy Constraint Analyzer*
Alcoa is a new tool for software design that offers, for the first time, fully automatic analysis of object models. Current commercial tools offer only shallow, syntactic analysis -- primarily that names are used consistently. Alcoa, on the other hand, can perform a deep semantic analysis of models that incorporate complex textual constraints. It can check the consistency of constraints, generate sample

configurations, simulate execution of operations, and check that operations preserve constraints.  An object model usually describes a huge set of possible configurations. Alcoa analyzes every configuration within some bounded size exhaustively. It routinely analyzes 10^30 configurations in seconds. In the construction of an object model, Alcoa can give interactive response as the designer experiments with different constraints.

Alcoa works by translating the constraints to be analyzed into a huge boolean formula that may then be presented to a variety of backend solvers. The current Alcoa prototype incorporates SATO, a Davis Putnam solver and WalkSAT, a new stochastic algorithm developed at Bell Labs. As boolean satisfaction technology continues to evolve, Alcoa will be able to exploit it for the analysis of object models.

## Progress

Alcoa, and its predecessor Nitpick, has been used to analyze a variety of practical designs including the Coda File System, the Department of Defense's High-Level Architecture, Mobile IPv6, the aggregation mechanism of Microsoft's COM, a paint colour database system, and an air-traffic control handoff protocol. In many of these, serious flaws were exposed.

The graphical sub-language of Alloy has been taught to undergraduate and graduate students in software engineering courses at MIT.

## Contact

Daniel Jackson, dnj@lcs.mit.edu

(FALL, 1999)

## Self Updating Software

*Leader:*
Daniel Jackson

*Sponsors:*
Nippon Telegraph and Telephone

### Background

Most research in software has focused on its execution, but increasingly the biggest costs are in its installation, upgrading, and removal. Current methods for installing software are inefficient and unsafe. They consume vast resources, not only in network bandwidth and unnecessary local storage, but, worse, in the attentions of the user.

As software production grows, and as small computing devices with limited memory become more pervasive, the pattern of updates will change from occasional, large updates to frequent, small ones. Moreover, our environment will become filled with software-enabled sensors and actuators whose interfaces allow no direct human control, and thus cannot be updated by current methods at all.

### Why Is This Interesting?

Our infrastructure will make it easy to build software that updates itself. In response to changes in the environment, to new requests from users, or to publication of new versions, the software will autonomously detect the need for an update; locate the right server; download the new software; certify that the update will have no bad effects; and integrate the new code. All this will occur without major interruptions in service, and will scale to a worldwide network. Updates will be extremely fine-grained, often at the level of an individual object. The infrastructure will provide simple, reliable, and flexible mechanisms, on top of which application developers can implement different updating policies.

### Approach

We separate two key problems.

First is the object management problem: how is the need for an update perceived, and how are the updates performed? How is the storage of objects organized? Where do objects reside in the network, and how are they named? To address these issues, we are working initially on a phased update protocol. When a server decides to update a class of objects, for example, it will use directory and caching schemes (as in Thor) to find out where affected objects reside. Individual sites then identify the affected objects locally, and modify them temporarily so that subsequent uses are trapped.

Second is the certification problem: how are objects checked in advance of their integration into the running client system, in order to ensure that the update is safe? What kinds of properties can be checked, and how are they specified? Our certification approach combines offline and online activities. Performing checking offline reduces the runtime cost, but is less safe. An economical balance is called for, in which compute-intensive analysis is performed offline for establishing more subtle properties, and simpler, but still critical checks are performed online. We plan also to accommodate analyses that cannot be automated at all; these might be carried out manually offline, and then certified cryptographically.

### Progress

The project has just begun. We are building an experimental telephone client that supports dynamic addition of new features, exploiting Thor, our object-oriented database, and the Windows TAPI interface.

### Future

Here are some examples of the kinds of application that might benefit from self-updating software:

* Adaptive Front End for Centralized Service. The user buys a service such as a filtered newsfeed, home banking, or airline reservation; as the service is expanded by its provider, the front-end application adapts automatically and invisibly, so the user sees new features as they become available.

* New Features for Embedded Devices. An elevator manufacturer improves its scheduling algorithm; new software is automatically installed at local plants worldwide. A new telephone switch feature requires additional computation at the handset; when the user requests the feature, the relevant software is downloaded. To optimize traffic flow in a city, the transport authority decides to price roads dynamically, so that the charge for travelling on a particular stretch depends on how congested it is at that time; car navigation software then adapts invisibly to a new price information source. A cable TV company offers a new way to choose programs according to user profiles; the user's VCR downloads new software spontaneously.

* Embedded Devices Adapt to Changing Environment. An elevator detects heavy load patterns and downloads software to plan better. A climate-control system adapts to a heat wave by downloading a special energy-saving regimen. A PABX detects the addition to a local network of a new device for teleconferencing, and fetches appropriate software.

* Gentle-Slope Application. The initial installation of a program by a user gives a minimal system with extensive help facilities; as the user matures, the application automatically fetches new features from a server, removes basic help functions, and morphs the user interface.

### Contact

Daniel Jackson, dnj@lcs.mit.edu

# IOA: A Language and Set of Tools for the Design, Analysis, and Construction of Distributed Systems

*Leaders:*
Anna E. Chefter, Stephen J. Garland, Nancy A. Lynch,
Joshua A. Taube, Mandana Vaziri

## Background

IOA is a new language, accompanied by a set of tools that supports the production of high-quality distributed software. The IOA development process begins with a high-level specification, refines that specification via successively more detailed designs, and ends by automatically generating efficient distributed programs. The IOA language and tools encourage system decomposition, which helps make distributed programs understandable and easy to modify. Most importantly, the tools provide a variety of validation methods (theorem proving, model checking, and simulation), which can be used to ensure that the generated programs are correct, subject to stated assumptions about externally-provided system services (e.g., communication services).

IOA is based on the I/O automaton model. This model has been used to describe and verify many distributed algorithms and systems, and also to express many impossibility results. The model's features make it especially suitable for such tasks: its fundamental concepts are mathematical (rather than linguistic); it is simple; it provides notions of external behavior (based on linear traces), composition (based on synchronized external actions) and abstraction (based on trace-set inclusion); and it supports a rich set of compositional proof methods, including invariant assertions and simulation relations.

## Why Is This Interesting?

The I/O automaton model was developed originally to reason about theoretical algorithms. Increasingly, it has been applied to practical system services such as distributed shared memory, group communication, and communication services such as TCP. Its use has helped resolve ambiguities and repair logical errors in these services. Most of this work has been done by hand; however, much is sufficiently stylized to admit computer assistance. For example, we have proved the correctness of the Dolev-Shavit Bounded Concurrent Timestamp protocol using the Larch Prover.

Practical successes have convinced us that the I/O automaton model can play an important role in developing real distributed systems. However, until now, there have been serious barriers to its use. The main problem, which applies to all prior formal validation frameworks, is the lack of a clear, formal connection between verified designs and the corresponding final code. Although it is feasible to verify the correctness of an abstract distributed system design using a theorem prover, there is no convenient way to extend this proof to actual code. Rather, the verified design must be recoded in a real programming language like C++ or Java before it can be run in a real distributed system. This coding step involves costly duplication of effort and can introduce errors.

(FALL, 1999)

A related problem is the lack of a programming language for distributed systems that is suitable for both verification and code generation. The features that make a language suitable for verification (axiomatic style, simplicity, non-determinism) are different from those that make it suitable for code generation (operational style, expressive power, determinism). Another problem for I/O automata has been the unavailability of lightweight validation tools such as simulators and model checkers. Such tools can yield quick feedback to help in debugging, prior to attempting a time-consuming formal proof.

## Approach

The design of the IOA language strikes a balance between the competing requirements for verification and code generation. The semantics of IOA are based firmly on mathematics, thereby providing a sound basis for verification. IOA allows non-determinism, so that users can express designs in as general a form as possible. IOA allows both axiomatic and operational descriptions for transitions, either separately or in combination, since neither alone is sufficient for all purposes. When the particular expressions in a program present problems for tools such as code generators, other tools enable the user to refine the program in a way that both eliminates the problematic expressions and preserves correctness.

IOA tools allow the user to subject their designs to a range of validation methods, including complete proof using an interactive theorem prover, study of selected executions using a simulator, and exhaustive study of small instances of the design using a model checker. The IOA tools assist the programmer in decomposing the design into separable interacting components, based on I/O automaton composition, and in refining it using levels of abstraction, based on trace inclusion and simulation relations. Novel features of the IOA toolset include its support for levels of abstraction (e.g., user-specified step correspondences), certain aspects of the simulation method (e.g., paired simulations, for testing whether one automaton refines another), and certain aspects of the code-generation method (e.g., abstract channels, for incorporating standard communication services into distributed systems).

## Progress

The IOA language has a working parser and static semantic checker, which produce an internal representation suitable for use by other tools. There is also a prettyprinter. A composition tool converts the description of a composite automaton into primitive form by explicitly representing its actions, states, transitions, and tasks.

Interfaces to the Larch Prover and to the SPIN model checker are currently under construction, as are a simulator and code generator.

## Future

We expect both to extend the IOA language and to enhance the IOA tools in response to the experience we gain from their use. One likely extension to the language will be based on the timed I/O automaton model, which enables users to express and establish performance bounds for distributed systems.

(FALL, 1999)

### Contact

Dr. Stephen J. Garland, garland@lcs.mit.edu
http //www.sds.lcs.mit.edu/~garland/ioaLanguage.html

(FALL, 1999)

## Cilk: A Multithreaded Programming System for Metacomputer

*Leader:*
Charles E. Leiserson

*Members:*
Don Dailey, Matteo Frigo,
Philip Lisiecki, Harald Prokop,
Sridhar Ramachandran, Bin Song, Svetoslav Tzvetkov

*Sponsors:*
Defense Advanced Research Projects Agency

### Background

Cilk is an algorithmic multithreaded language inlaid into C. The philosophy behind Cilk is that a programmer should concentrate on structuring his program to expose concurrency and exploit locality, leaving the run time system with the responsibility of scheduling the computation to run efficiently on a given platform. Cilk's run time system takes care of details like load balancing and communication protocols. Unlike other multithreaded languages, however, Cilk is algorithmic in that the runtime system's scheduler guarantees provably efficient and predictable quality of service.

### Why Is This Interesting ?

Parallel processors are now commodity products. One can now buy a 4-processor server for under $15,000. In the next few years, parallel computers will appear as ordinary desktop and laptop machines. But, programming applications is expensive and hard, requiring elaborate communication protocols to be programmed. Cilk offers a simple but efficient programming environment for these platforms that naturally extends the existing serial programming environment, obviating the need for applications programmers to support two separate implementations for serial and parallel platforms. Moreover, since Cilk programs are adaptively parallel, this environment can be lifted into the more demanding metacomputer environments, such as distributed clusters of multiprocessors.

### Approach

We have taken the approach of "inlaying" functionality into the existing serial programming paradigm, rather than "layering" the functionality on top. Consequently, Cilk avoids the "bloating" problem common to layered abstractions, providing a lean and mean "translucent" abstraction of parallelism. Cilk extends C with only 5 new keywords and no new data structures. Spawning a parallel function takes only a few machine cycles longer than a C function call. Extensively throughout the Cilk implementation, we have backed up our algorithms with mathematical proofs of their effectiveness. Even in the heuristic area of debugging, Cilk provides a race-detection debugger that offers provable guarantees of bug detection.

(FALL, 1999)

### Progress

Cilk-5.2, including its Nondeterminator-2 debugger, was released in June 1998. Cilk is now being used by scores of researchers, it is used for teaching at several universities (MIT, CMU, Berkeley, U. Texas, etc.), and industry (Sun, Intel, SGI, Microsoft, etc.) has started to show active interest in its ideas. Researchers at Dartmouth have modified Cilk to implement the world's fastest simulator of wide-area networks. A prototype Distributed Cilk has been released which runs on clusters of multiprocessors. An MIT team using Cilk won the ICFP Programming Contest, defeating 48 other teams from the U.S. and around the world.

### Future

We are currently developing the programming environment surrounding Cilk. We are developing support for parallel file and stream I/O. We are also developing operating-system mechanisms for scheduling adaptively parallel Cilk jobs. These new capabilities will allow Cilk to address the needs of real-time applications involving streaming data.

### Contact

Charles E. Leiserson, cel@mit.edu
http //supertech.lcs.mit.edu/cilk

(FALL, 1999)

## FFTW (The Fastest Fourier Transform in the West)

*Leader*:
Charles E. Leiserson

*Members:*
Matteo Frigo, Steven G. Johnson

*Sponsors*:
Defense Advanced Research Projects Agency

### Background

FFTW is a fast portable C code implementing the fast Fourier transform, one of the most important computational problems. FFTW's goal to achieve high performance automatically. The performance of modern computers depends on so many factors that high performance is often difficult to achieve. Moreover, processor architecture changes continuously, and old high-performance programs do not perform well anymore on new machines. FFTW is an adaptive code that automatically solves these problems.

### Why Is This Interesting?

From a real-world standpoint, the FFTW project already produced a production-quality implementation of the fast Fourier transform that is extremely fast, and it is currently used by thousands of users worldwide. From a scientific point of view, FFTW has automatically produced algorithms that were not known before. Moreover, it led to a better understanding of the capabilities of current compilers, and how they can be exploited effectively.

### Approach

At compile time, a special-purpose compiler (called the FFTW codelet generator) produces optimized code automatically. The codelet generator resembles both a computer algebra system and a compiler. The generator is able to derive concrete algorithms from high-level abstract descriptions by systematic application of algebraic rewriting rules. The back-end of the generator performs scheduling of the algorithm, so as to minimize the lifetime of variables, and produces C code that can be compiled efficiently on most machines.

At runtime, the FFTW system automatically selects the fastest algorithms among those produced by the codelet generator. This choice is machine-specific. In this way, the FFTW system adapts itself to the hardware, maximizing its performance.

Progress Version 2.0 was released in September 1998. It computes one-and multi-dimensional Fourier transforms both of real and complex data. FFTW enjoys thousands of users worldwide, including government agencies and commercial companies.

(FALL, 1999)

A comparison of FFTW versus more than 40 other packages has been performed on several platforms, demonstrating the excellent performance of FFTW even with respect to vendor-provided libraries.

### Future

We envision a "metacompiler" that should generalize the capabilities of the codelet generator to other problems. Basically, a user should be able to write a program that expresses an algorithm abstractly, and to direct the compiler (through a metalanguage) on how to compile the program. The meta program can comprise optimizations that would not be safe for a general-purpose compiler to apply. The FFTW experience demonstrates that special-purpose compilation rules can be extremely effective for achieving both performance and correctness.

We plan to extend the current system to handle mathematical tools related to the Fourier transform. In particular, the two-dimensional discrete cosine transform is used in the JPEG standard for image compression.

### Contact

Matteo Frigo, athena@fftw.org
http //www.fftw.org

# Building Blocks for High Performance and Fault Tolerant Distributed Systems

*Leaders:*
Roberto De Prisco, Idit Keidar, Roger Khazan, Victor Luchangco, Nancy Lynch

*Collaborators:*
Ken Birman, Alan Fekete (U. Sydney), Jason Hickey (Cornell),
Alex Shvartsman, Robbert Van Renesse (U. Conn)

## Project Overview

Designs for high performance, fault tolerant (HPFT) distributed systems are often extremely complex. A popular approach to managing this complexity (used, for example, in systems like Orca, Isis,Transis, Horus, and Ensemble) is to construct the systems using communication and memory "building blocks." Such building blocks may represent global or local services. Building blocks may be combined in parallel, or may describe all or part of a system at different levels of abstraction. In our view, a building block is not just a program, but also has a mathematically well defined external specification.

In this project, we are working with system developers (including Kaashoek, Clark, Lampson, Birman, van Renesse, Malkhi, Arvind, Leiserson, and others) to define Input/Output Automaton specifications for key communication and memory building blocks. We use our specifications to argue the correctness of implementations of these building blocks, and we model and analyze HPFT systems that use these building blocks. Many of our specifications include performance and/or fault tolerance properties as well as safety properties, so we can also reason about these aspects of the system.

## Why Is This Interesting?

Decomposing complex systems into building blocks using parallel composition and levels of abstraction makes them easier to build, to understand, and to analyze. Using this approach, programmers can modify pieces of a system without destroying needed properties of the system as a whole. The specifications serve as high quality documentation, for both users and system developers. A mathematically rigorous "building blocks" approach enables specifications to be used by both validation and code-generation tools.

This approach can be used to reduce occurrence of errors, a consideration that is especially important for safety-critical systems (e.g., air-traffic management systems, strategic information systems) and other settings where failure is expensive (e.g., banking systems, stock market systems). Including performance information in the specifications can permit predictions of runtime system performance. Including fault tolerance information in the specifications allows analysis of safety and performance properties even in the presence of failures of some system components.

(FALL, 1999)

## Approach

Our approach is both mathematically rigorous and closely tied to real systems. In previous work, our group has developed a significant general theory, based on I/O automata, that enables rigorous reasoning about distributed algorithms and systems. The usefulness of this framework has been demonstrated by many case studies (see, for example, Lynch's books on Distributed Algorithms and Atomic Transactions, as well as our group's many papers on system modeling and verification). Working with system developers, we are extending this theory and applying it to HPFT systems.

We illustrate our approach with our recent work on view synchrony. In this work, we give a clean and simple specification (VS) of a view-synchronous group communication service, similar to the virtually synchronous group communication services of Isis, Transis, Totem and Ensemble. Many attempts have been made to specify such services, but these were complicated, and in some cases had inconsistencies. Our VS service integrates multicast communication with a group membership service, informing distributed clients of changes in the "view" of the group of clients, and ensuring that the sequences of messages received for each view are consistent across different clients. Our specification includes a performance and fault tolerance property that says that, once the underlying physical system stabilizes, the views stabilize soon thereafter, and messages are delivered rapidly.

In order to arrive at a good specification for a service, we typically "work backwards" from applications of the service, defining the service to be exactly what is needed to make the applications work correctly. Working in this way tends to lead to specifications that are mathematically much simpler than working exclusively from service implementations. For example, the VS specification was developed with the motivation of giving a formal algorithm that implements a totally-ordered broadcast service on top of it, and being able to prove it correct and analyze the performance of the resulting system.

In analyzing a system, our goal is to expose the structure of the system, breaking it down into conceptual pieces using global specifications where appropriate, rather than along system component lines. Our work on view synchrony shows how to decompose a system providing totally-ordered broadcast into two conceptually simpler pieces, one which implements VS, and the other which uses VS to implement the broadcast.

Of course, what the "conceptual pieces" are depends on the level at which we are looking at the system. Our theory allows a layered or hierarchical view of the system, so we can analyze it at a variety of levels. Each layer can be used as the basis for building higher level systems, and also as the specification to be implemented by lower level components. It provides an abstraction barrier that allows the components used to implement a particular service to be hidden from the application using the service specification. In designing a system to provide totally-ordered broadcast using VS, the designer does not need to worry about how VS is implemented. And a designer using the totally ordered broadcast does not need to know about VS at all.
Because our framework is mathematical, it supports formal proofs of the safety and performance claims of the systems we analyze. It also provides a solid foundation for the construction of computer tools to aid in the design and verification of these

systems, including the possibility of generating code directly from our specifications through a series of refinements. Many of the proofs of invariants used to establish correctness of the broadcast algorithm using VS have been checked mechanically using the PVS theorem prover.

Because of the elegance and mathematical rigor of our specifications, they provide a framework that can be used to define similar services, and applications that use these services. For example, Chockler uses the VS specification to define similar services provided by Transis, and we use it as the basis of a load balancing replicated data service algorithm. We also extended it to maintain information about dynamic primary views, which is used by some algorithms.

## Progress

In addition to the view synchrony work described above, we have applied our theory successfully to a wide variety of domains, including connection-management protocols such as TCP and T/TCP, and higher level group communication primitives. These domains were provided by such systems as Isis and Transis, data service and consensus service specifications, and weakly coherent memory models being implemented by modern architectures. Despite the diversity of domains, we have achieved significant successes in applying our theory to these systems. We mention a few particular examples to give a flavor of our work.

We have produced a model and proof of the Orca systems distributed implementation of coherent shared memory using the Amoeba communication primitives. We defined the implementation in two layers, using an intermediate service, called context multicast, to expose the structure of the implementation. In doing this, we found a logical error in the existing Orca implementation, which was subsequently repaired in our model and in the real implementation.

We have given a formal description of a fault-tolerant consensus service, shown how to use this to manage replicate data, and given an implementation based on Lamport's Paxos algorithm. We also defined an eventually serializable data service, and gave an algorithm based on one by Ladin, Liskov, et al., which uses lazy replication. For each of these, we give a formal proof of the safety, performance and fault tolerance. We have also provided possibility and impossibility results for the $k$-set consensus service, a generalization of the consensus service.

Finally, we are also examining relaxed consistency models for multiprocessor shared memory, and have developed a simple and clean framework to describe a class of models called precedence-based memory models. This is an outgrowth of our earlier work on eventually serializable data services, and we expect that it will be useful for shared memory models both at the architecture and the systems level. We use this to prove the correctness of an algorithm used by the Cilk multithreaded language system.

## Future

We intend to continue our current direction of research, formally specifying other useful building blocks. In particular, we plan to work closely with the Ensemble group at Cornell in modeling building blocks that arise in that system, and giving a formal analysis of their system implementation, as well as possibly proving some lower bound results on the performance of certain group communication primitives, such as totally-ordered broadcast. And we plan to look at other group communication services and applications that use them.

We also intend to use our specifications to make formal comparisons of different system implementations. For example, we want to model and analyze an implementation of VS described in the literature, and compare distributed implementations of shared memory based on VS and based on Paxos, including their performance and fault tolerance properties. We also want to consider alternative definitions of view synchrony and dynamic view synchrony, and how algorithms built using the original definitions need to be modified to work on these alternative service specifications.

One application area deserves particular mention, that of computer-supported cooperative work (CSCW), which include multi-media and desktop conferencing, distance learning, interactive games, and collaborative computing.

In the area of relaxed consistency models, we would like to understand how to incorporate locks into the framework of precedence-based memory models, and to model formally some architectures for machines such as the PowerPC, or the DEC Alpha.

## Contact

Joanne Talbot, joanne@theory.lcs.mit.edu
http //theory.lcs.mit.edu/tds/applications.html

## Developing a Model-Based Validation Framework for Hybrid Systems Through Case Studies in Automated Transportation

*Leaders:*
Carl Livadas, Nancy A. Lynch

*Collaborators:*
John Lygeros, Roberto Segala, Frits Vaandrager

### Background

The recent trend towards system integration and computer automation has lead to the emergence of very complex hybrid systems -- dynamical systems that involve both discrete and continuous behavior and/or components. To date, the validation of hybrid systems has predominantly been accomplished by simulation testing. However, as systems get more complex, the exhaustive testing of all possible system behaviors becomes unrealistic and, subsequently, the confidence in such validation schemes is diminished.

The safety-critical nature of numerous hybrid systems, such as automated transportation systems, has encouraged the formal modeling and validation through deductive reasoning. This approach not only forces the system designers to produce formal specifications of the system's behavior, but provides irrefutable proof that a system satisfies its safety requirements. When generating the formal specifications of a system's behavior, it is helpful to take advantage of modular decomposition and abstraction. The generation of specifications in a modular fashion emphasizes the component structure of the system and simplifies the modeling process by focusing attention to a single component at a time. The use of abstraction allows the system designers to ignore inessential system behavior, to use nondeterminism in a system's description, and to allow reasoning about the system behavior at a high level.

### Why Is This Interesting?

The advantages of a formal approach to modeling, analyzing, and validating hybrid systems -- in particular, safety-critical automated transportation systems -- are numerous. First, this approach produces a clear and precise model of the system that is both abstract (that is, inessential system details are omitted) and complete (all components of the system, including environment models, are modeled). Second, the deductive reasoning techniques and well-established proof techniques from CS and control theory may be used to validate system properties. It is important to note that this validation process is formal, in the sense that, under prespecified assumptions, any behavior of the system satisfies the properties that are verified.

### Approach

In order to model hybrid systems, the TDS group has been developing a mathematical framework denoted the hybrid I/O automaton (HIOA) model. In this model, a hybrid system is specified as a set of automata that share variables and discrete transitions. The discrete behavior of a system is described by a set of discrete state transitions; discrete transitions shared among automata synchronize based on transition labels. The continuous behavior of a system is described by a set of trajectories which specify the behavior of the variables of an automaton with time.

The HIOA formal model has evolved to its latest form through the modeling of various automated transportation systems. The modeling efforts of the TDS group have been guided by the actual designers of the various systems, who have often assisted so as to guarantee that the system models are truthful descriptions of the behavior of the actual systems.

The proof methods that are used are assertional. They involve either proofs that the system satisfies a particular invariant, or proofs that a refined model of the system actually implements a more abstract model (simulation relations). It is important to note that assertions can involve timing information; that is, predicates on continuous variables of the system. Since the behavior of HIOA involves both discrete transitions and trajectories, proof techniques from control theory must be used to reason about the continuous system behavior. However, through multiple discrete and hybrid system case studies, the TDS group has become proficient in determining when and how to use proof techniques from CS and control theory. The key is to reason about discrete and continuous system behavior independently and, subsequently, to combine the individual results so as to prove a particular property of the hybrid behavior.

Furthermore, it is important to note that due to modular system decomposition and abstraction, the HIOA modeling and validation proofs can scale to larger and more complex hybrid systems. This is achieved by reasoning about the system's behavior at a high level of abstraction and showing that more refined system models are actual implementations of their abstract counterparts.

### Progress

To date, the TDS group has successfully modeled and verified several hybrid systems in the area of automated transit. One such system is the automated highway system of the California PATH project. Recently, Dolginova and Lynch have used HIOA to model and verify the safety of the platoon join maneuver -- a maneuver in which two adjacent vehicle platoons join to form a single platoon. In the realm of personal rapid transit, the TDS group has modeled the PRT 2000TM system under development at Raytheon Corporation. For various track topologies, it was verified that the vehicles comprising the system neither exceed a prespecified speed limit, nor collide among themselves. While modeling this system, Livadas and Lynch developed an abstract model of a protector -- a protection subsystem that guarantees a particular safety property under a set of assumptions. The advantage of this model is that the proof of correctness of particular protector implementations get reduced to simple simulation proofs from the implementation to the abstract model.

Recently, Lygeros, Livadas, and Lynch have been modeling and analyzing the traffic alert and collision avoidance system (TCAS II version 7) that is being used by aircraft to avoid midair collisions. The modeling approach involves specifying all components of the system which in this case not only involves the system hardware but also the aircraft pilots. Currently, Lygeros et al. are in the process of proving correctness of the TCAS algorithm for an ideal system (a system with no delays, no uncertainties, and exact sensors). The goal is to be able to prove that under particular assumptions the TCAS system will prevent aircraft collisions. Since the validation techniques to be used scale to more complex systems, once the correctness of TCAS is shown for the ideal system, the reintroduction of delays and uncertainties should be straightforward.

### Future

The hybrid system research goal of TDS is to finalize the HIOA model and to explore its modeling and validation advantages and limitations. It is important to show that a formal approach to validating complex hybrid systems is not only viable but relatively simple and computationally tractable.

As future research, the TDS group will attempt to tackle increasingly complex behaviors of TCAS and other automated transportation systems and determine how such a validation approach can help in error detection and system redesign. As with all research conducted in the TDS group, such case studies will also be used to extend the HIOA modeling and validation techniques applicable to the area of hybrid systems; that is, whether the HIOA model is expressive enough, which proof techniques can and should be used, etc. Furthermore, the TDS group is planning to develop a specification language for HIOA so as to interface with existing software packages for reasoning about continuous and discrete system behavior, such as theorem provers, model checkers, and symbolic mathematics manipulators. The goal is to develop automated tools that can assist in the validation process and make such an approach more attractive to the mainstream system designers.

Finally, taking advantage of HIOA model refinement, it is in principle possible to reason about a system's behavior down to the level of running code. The TDS group will tackle the validation of a hybrid system's behavior to such a degree either by automatically generating code for the discrete part of the system's HIOA model, or by proving that the code of the actual system implements the discrete part of the system's HIOA model. The former approach would entail extending the I/O Automaton model (a model developed by the TDS group to reason about and validate discrete event systems) code generation tool so as to allow the expressiveness of the HIOA model. The latter approach would entail showing behavior inclusion through simulation relations; that is, showing that each step of the implementation corresponds to a step, or sequence of steps, of the HIOA model. This research will enable hybrid system designers to provide irrefutable proof that a system's code is a correct implementation of the system's HIOA model. Subsequently, any properties shown to be true for the system's HIOA model would also be true for the actual system implementation.

### Contact

Nancy A. Lynch, lynch@theory.lcs.mit.edu
http //www.theory.lcs.mit.edu/tds/hs.html

(FALL, 1999)

# Approximation Algorithms for Hard Optimization Problems

*Leader:*
Michel X. Goemans

## Background

Since the early seventies, it is known that solving exactly and efficiently many optimization problems is unlikely since, through the theory of NP-completeness, the existence of any efficient algorithm for any such problem would imply the existence of efficient algorithms for all of them. The search for approximation algorithms, i.e. polynomial-time algorithms that are guaranteed to deliver a solution within a certain factor of optimal, then started, but the early results were sparse and often based on very elementary arguments. We are interested in obtaining general and sophisticated techniques to design approximation algorithms and to improve upon the best known performance guarantee for many fundamental combinatorial optimization problems.

## Why Is This Interesting?

First, there are many large-scale combinatorial optimization problems that arise in avariety of settings, such as for example routing in, or design of telecommunication networks. Techniques used in improved approximation algorithms often lead to practical improved codes for these problems. The advances in computing power also allows software designers to be much more sophisticated in the design of efficient algorithms.

Secondly, what can and cannot be done is a fundamental question in computing. In recent years, through major developments in the area of probabilistically checkable proofs, it was discovered that for many problems, there is a limit to the extent one can approximate a given problem. Closing as much as possible the gap between the approximability and the non-approximability results is therefore a major challenge in obtaining a better understanding of computability for optimization problems.

## Approach

Our goal is to design general techniques for approximation algorithms and to investigate their power, limitations and applicability. Fundamental problems, such as the traveling salesman problem, the bisection problem, the satisfiability problem, many network design problems, are being considered since advances for these problems often lead to advances for many other problems.

## Progress

Over the last few years, we have proposed and investigated several general techniques to design approximation algorithms semi-definite programming, the primal-dual method, and new randomized rounding procedures. This has led to improved (and often with the currently best bound) approximation algorithms for a variety of problems, ranging from the maximum cut problem to network design problems.

## Future

We are planning to continue to push the limits of how well important optimization problems can be approximated.

## Contact

Michel X. Goemans, goemans@theory.lcs.mit.edu

(FALL, 1999)

## Haystack: Adaptive Personalized Information Retrieval

*Leaders:*
David Karger, Lynn Andrea Stein

*Sponsors:*
Merrill Lynch, Nippon Telegraph and Telephone

### Background

The Haystack Project aims to create a community of individual but interacting "haystacks": desktop-integrated personal information repositories, which archive not only base content but also user-specific meta-information, enabling them to adapt to the particular needs of their users. Unlike current IR systems, which treat users anonymously, Haystack learns about its user's knowledge, vocabulary, and habits and uses this information to provide retrieval that matches the needs of its user.

### Why Is This Interesting?

Web search engines now play an integral role in our daily activities, showing how important the retrieval of information is to effective work. These web search engines fill the role traditionally held by libraries. They organize vast masses of useful and useless information, probably containing (somewhere) the information we want, and make it accessible to all according to a single organizing principle that all must follow. Everyone who accesses them is treated in the same anonymous fashion.

Nobody begins their search at the library. Instead, we always start by searching for information in our bookshelves. We find these small repositories better than libraries for many reasons. We vette the information entering our bookshelves, and thus trust that it is (mostly) of high quality and understandable by us. We organize the bookshelf idiosyncratically, using subject and usage-based rules that might make no sense to anyone else but that make it easy for us to find information. When our bookshelves fail, we still avoid the library, turning first to those colleagues whose knowledge and judgment we trust to steer us towards useful information.

Haystack is an attempt to carry this "bookshelf metaphor" into digital information systems. We index users personal information holdings (mail, files, web pages, and so on). We let users organize and annotate this information to make it easily accessible. And the system adapts over time as it learns about the query and retrieval habits of its user. In the future, we plan to add the capability to seek out and query other trusted colleagues information collections as well.

Besides the obvious benefits of the system, an important secondary benefit is the strong motivation it provides to make more information accessible to more people. Putting one's work on the web takes work and provides no direct reward, so a great deal of valuable information never appears on web search engines. Organizing your own information is a goal everyone has; with Haystack a side effect of this undertaking will be the (controlled) exposure of the information to others who can benefit from it.

## Approach

Haystack is a middleware system that integrates into the user interface on one side and into a collection of (off the shelf) search and database tools on the other. Its goal is to collect all possible data about the user's information and is accesses to it, to store it in a powerful data model that lets us represent it all, to feed that data into the underlying search tools, and to mediate between the user and the search tools (as well as other colleagues' Haystacks) in order to adapt their search behavior to the user's habits.

Haystack is designed for tight integration into the user's desktop. Haystack can watch web browsing, mail reading and sending, shell commands (file operations), search activity, editing, and file operations. It will record this information (for example, what files a user edited while visiting a particular web page) and use it to enhance retrieval (if the given web page is relevant to a search, perhaps the files are too).

Haystack's data model allows us to store all of this information. Haystack is easily extensible to handle arbitrary data types; at present it understands assorted mail formats, postscript, and HTML. But in addition to storing the raw data, Haystack is a general metadata system in which any element can be related to any other element. The relationships are themselves first class objects that can be part of other relations. For example, a user might attribute authorship of a document to someone (linking document to author). Later, another uses might comment that the attribution is wrong (linking a comment to the attribution link).

The information Haystack stores can be used to bias the behavior of off the shelf search tools to better meet the needs of the user. For example, Haystack keeps track of what documents a user actually views after performing a query. If a similar query is made in the future, Haystack knows that certain documents are the preferred answers for it. For many search engines, this change in response can be arranged by annotating the (text of the) document with a copy of the query. This lets us continue to use the power of the search engine while changing its behavior to match what the user expects. Since Haystack acts as a middleman between the user and the search engines, it will be easy to extend it to talk to other search engines -- in particular, other user's Haystacks.

## Progress

We have a prototype version of Haystack that exhibits basic examples of the behavior described above. The system is integrated into a web browser, a mail reader, and a command shell. Three text search engines (two free, one commercial) and one relational database (LORE) have been integrated as back ends. Haystack recognizes and extracts information from HTML, postscript, and ASCII documents. It adapts to queries in the fashion described above. We concluded that this version of Haystack would not adapt well to the demands we were placing upon it, and have spent the year writing from scratch a new version with the extensibility that we will need to continue our work.

## Future

In the future, we aim to expand Haystack's integration with the user interface and its ability to learn from user behavior, and aim to add the ability to interact with colleagues' Haystacks. Future user interface extensions include a window-server level observer of user activity (so we can see what window he is watching) and augmented facilities for navigating the metadata in the Haystack (this metadata takes the form of links, and may be effectively navigated using hypertext methods). On the learning front, we are exploring technology from the machine learning community. Much of this technology has been applied to so called "content rating" services (such as firefly) which rate "good music" by gathering information from many users. Our goal is to use this same technology to identify material that is useful for a particular information need.

### Contact

David Karger, karger@mit.edu
Lynn Andrea Stein, las@mit.edu
http://www.ai.mit.edu/projects/haystack

## Real-time Clustering and Ranking for the Web

*Leaders:*
Ravi Kannan (Yale), Santosh Vempala

### Background

The first question addressed here is to identify the 10 (or any user specified number of) most important documents from among those returned by a search engine on a keyword(s). The second more difficult problem is to group the documents into clusters, where each cluster contains similar documents, and return the most important documents in each cluster.

### Approach

Recently, we presented a solution to both problems by using a very fast randomized algorithm to find the Singular Value Decomposition of a suitable "document-term" matrix.

### Progress

Theoretical analysis of the real-time aspect is solid. Analysis of the clustering quality is still at a very early stage. Empirical evaluations of these algorithms are highly favorable.

### Future

There will be more empirical testing.  A good model for clustering, and some performance guarantees for the algorithm.

### Contact

Santosh Vempala, vempala@theory.lcs.mit.edu

(FALL, 1999)

## Speeding up Learning Algorithms via Random Projection

*Leader:*
Santosh Vempala

### Background

Many algorithms take time that increases with the dimensionality of the input data. This can result in a severe slowdown in situations where each input point inherently has a large number of dimensions (attributes). The technique of random projection allows on to reduce the dimensionality of the input while preserving several nice properties. Can this be used to speed up algorithms in general, and learning algorithms in particular?

### Approach

Recently, Rosa Arriaga (Harvard) and I presented a robust concept model for which the number of examples required to learn a concept depends only on the robustness of the target concept and not on its dimensionality. This suggests the following generic algorithm:
(i) Project the data to a random low-dimensional space
(ii) Learn a target concept in the low-dimensional space
(iii) To classify a new example, first project it down and then classify it according to the learned concept.

### Progress

For robust concepts (roughly speaking, robustness is resistance to attribute noise), the approach has some nice theoretical guarantees.

### Future

Exactly when is the approach applicable? How well does it do empirically on typical learning problems?

### Contact

Santosh Vempala, vempala@theory.lcs.mit.edu

(FALL, 1999)

# INTERFACES & APPLICATIONS

(FALL, 1999)

# Project LCS-Marine: A New Paradigm for Human Computer Interactions

*Leaders:*
Michael Dertouzos, John Wroclawski, Victor Zue

*Sponsors:*
Defense Advanced Research Projects Agency

## Background

LCS-Marine is a project aimed at serving people's future information needs. It sets forth a radically new human-machine interaction paradigm (Listen, Communicate, and Show) in which the computer can interact with the user naturally via spoken language instead of keyboard. The interaction promotes mobility, since compute and knowledge servers can be connected to the user via phone or a computer network. A key component of the project is the development of the lightweight, handheld units. It is intended to be a concept demonstration for use by the Marines.

## Why Is This Interesting?

This project is motivated by several factors. First, people are demanding connectivity in order to access information, even when they are on the move. Second, the computer is getting smaller; pretty soon its form factor will prevent one from typing to it. Last but not least, speech remains the most natural, flexible, and efficient means of communication for humans. The project is interesting because it investigates the next generation of PDAs, devices that are almost completely driven by speech.

## Approach

Our approach is based on the premise that, instead of designing wearable computers, we would like to make computing available through mobile communication networks. The system has four key components. First, the handheld devices enable natural human/machine communication using spoken dialogue, thus eliminating the need for bulky keyboards. It is lightweight and mobile in that it requires only limited local computing and display capabilities, and instead relies on wireless telephones or computer networks for connectivity. Second, the underlying system consists of many such devices for all members of a group, interconnected with a set of server clusters by a wide or local-area network. Third, a collection of server PC's, also wired to the network, handles spoken dialogue with the user, and fetches and processes the desired information. The entire scenario is enabled by a spoken language interface that permits a user to access information and solve problems. Uses may include queries about events and weather, navigation through unfamiliar territory (talking to your map), surfing the web, spatial location of fellow group members, composing and reading short messages, etc.

## Progress

The Spoken Language Systems (SLS) Group at LCS has designed, implemented and distributed the GALAXY-II architecture for dialogue interaction, and has developed initial versions of three of the four applications. The Advanced Network

Architecture (ANA) group has developed several network connectivity alternatives, and demonstrated the system's ability to provide network access in a mobile environment. For example, we are now able to achieve near ISDN bandwidth network connectivity in a Hummer, so that a person can access the Web in a totally mobile environment. A subcontractor (MicroDisplay Corporation) has been selected, and we have been working closely with them in the design and implementation of the handheld device; a first generation prototype was demonstrated at the 35th anniversary celebration of LCS in April 1999. A second-generation prototype is available in the Fall of 1999. Members of the Spoken Language Systems Group have been working closely with Lockheed Martin on technology transfer, resulting in the successful demonstration of spoken dialogue interaction in the logistics domain in January 1999.

### Future

We will be refining our design of the handheld device, which will have a smaller 600x800 color display and camera. Furthermore, spoken language applications continue to be developed and improved.

### Contact

Victoria Palay, palay@mit.edu
http://www.sls.lcs.mit.edu

(FALL, 1999)

# The MAITA Project: Knowledge-Based Support for Monitoring Tasks

*Leaders*:
Jon Doyle, Peter Szolovits

*Members*:
Yu-Han Chang, Mary Desouza, Isaac Kohane, William Long,
Mojdeh Mohtashemi, Delin Shen, Christine Tsien

*Sponsors:*
Defense Advanced Research Projects Agency

## Background

The MAITA system provides knowledge-based support for monitoring tasks. The acronym stands for the Monitoring, Analysis, and Interpretation Tool Arsenal. The name reflects the provision of tools rather than monitoring systems themselves. The hypothesis underlying this work is that one can significantly reduce the costs -- in time, effort and expertise -- of constructing a monitoring system through use of a rich library of monitoring systems and tools that enable easy composition, modification, testing and abstraction of these library elements. The arsenal thus includes tools for solving monitoring problems, tools which permit composition and correlation of separate signals into informed alerts, and tools which permit rapid addition or tailoring of monitor behavior.

This work was undertaken to construct a uniform monitoring infrastructure of use to multiple investigations conducted by the Clinical Decision Making group at the Massachusetts Institute of Technology Laboratory for Computer Science and the Informatics Program at Childrens Hospital of Boston. We aim to use the many monitoring needs of these multiple investigations as a source of tasks through which to test the hypothesis.

## Why Is This Interesting?

We designed the MAITA system to help address two difficult problems: monitoring in context, and monitoring of special or temporary concerns. Hospital intensive care units (ICUs) provide good examples of the problem of monitoring in context. Each ICU has dozens of monitoring devices attached to the patient, and each device signals alerts if the one or several signals it measures oversteps certain (usually narrow) bounds. Since even simple movements of the patient can cause momentary loss of signal, this herd of dissociated monitors typically sounds an alarm once or twice every minute. Investigating and disposing of each alert takes a trained nurse or physician several minutes, so attendants frequently disable most or all of the alarms to allow themselves to attend to the patient rather than to the machines. Though disabling the alarms defeats the purpose of the alarms, it does not always have the terrible consequences one might imagine because most of the alerts are false alerts, ones which can be seen to be false right away with some knowledge of medicine. For example, a heart rate of zero in the presence of steady blood pressure, blood oxygen, and brain activity signals a detached heart rate probe, not a stopped heart. The problem here is to find a way to use medical knowledge and common sense to combine all the hundreds of signals into considered alerts that represent the true conditions requiring attention of the attendants.

Computer security provides good examples of the problem of special or temporary concerns. Miscreants attempting to break into one site often repeat their attacks at other sites. A successful or partially successful attack against one site may call for alerting similar or related sites of the special characteristics of the attack, since the attack may have involved several events which mean little in themselves but which in retrospect appeared essential elements of the successful attack. For example, a random request from an obscure Lilliputian site might immediately precede a seemingly unrelated request from a Blefescucian site in the course of the attack on a particular service. If the succumbing site can alert its neighbors to this fact, the neighbors may repel similar attacks by watching for this specific combination and taking appropriate ameliatory actions pending a longer term solution to the vulnerability. The problem here is how to rapidly modify monitoring systems in place to recognize additional or specialized conditions and how to easily remove these specialized additions as windows of opportunity close or shift.

## Approach

The central concept in the MAITA architecture is that of a network of distributed monitoring processes. The metaphor we use for thinking of the operation of these monitoring networks is that of electrical networks, in which we "wire together" various components and network fragments by connecting their terminals together. In the computational context, individual monitoring processes take the place of electrical components, and transmitting streams of reports takes the place of electrical conduction. The set of monitoring processes form the nodes of the network, and the communication paths form the edges or links of the network. Each process in the network may have a number of "terminals," each of which receives or emits streams of reports. The network may exhibit a hierarchical structure, as some monitoring processes may consist of a subnetwork of subprocesses.

The distributed processes may degenerate into chaotic interference without some means for structuring the interactions. To provide this structure, MAITA provides a "monitor of monitors" or "MOM" to construct, maintain, inspect, and modify the monitoring network and its operation. We achieve a degree of uniformity in the control process by organizing MOMs as special types of monitoring processes.

The MOM is designed to provide for resilient and persistent networks of monitoring processes. Toward this end, the command and control system monitors all the other monitoring processes, correcting and restarting them as needed. The control system itself is monitored by a subsidiary monitor which corrects and restarts the control system as needed. The architecture employs a persistent database to aid in providing this level of stability, and monitors the functioning of the database system as well. The control system also works to ensure the accuracy of the database records, both by updating them as changes are made and by checking them as needed.

The architecture is designed to provide an open platform for system development and interconnection. Command operations are transmitted using hypertext transport protocol (HTTP), allowing for basic system operation from any web browser, using commands entered by hand or through multiple specialized web pages or applets. Such web-based control minimizes requirements for installing specialized software on local machines. Supporting this open operation further, we provide reference implementations of the MAITA-specific communications mechanisms, in the form of Java and Common Lisp classes that provide monitoring process wrappers for use in legacy systems written in these languages.

Data are transmitted by an expandable set of common protocols, permitting direct interconnection with many legacy and separately-developed systems. The design permits information to flow through the network by several different protocols, including socket-based ASCII character streams, HTTP (Hypertext Transport Protocol, used by the World Wide Web), SMTP (the Simple Mail Transport Protocol, used by email systems), Java RMI (Java Remote Method Invocation), ODBC (Open Database Connectivity), and OKBC (Open Knowledge Base Connectivity, a protocol for transmitting logical and frame-structured knowledge to and from knowledge bases). The system developer or user chooses the protocol appropriate to the volume, regularity, and type of the information being transmitted. Regular and high-frequency transmissions typically go through persistent stream, ODBC, or OKBC connections. Intermittent and low-frequency transmissions probably go on temporary HTTP, SMTP, Java RMI, ODBC, or OKBC connections. Records of information transmitted to input or from output terminals are structured in protocol-dependent formats.

One can call any monitoring system knowledge-based, since its designers employ knowledge in the course of its construction. The MAITA architecture is intended to support additional roles for explicitly-represented knowledge, in the operation of monitoring systems.

The first role is that of monitoring processes which explicitly reason in the course of their analysis. MAITA supports these by offering an ontology of monitoring concepts and a knowledge base of monitoring methods. We annotate the structure of information flow with knowledge-level descriptors, distinguishing the reports being transmitted and received from the computational representations of these reports, and distinguishing these computational representations from the protocol-specific encodings used for transmission.

The second role is that of monitoring networks, in which the structure of the network explicitly reflects knowledge about the conditions being monitored. This network structure should identify the conditions of interest and the dependencies among them. For example, the structure of a monitoring network should revolve around the conditions on which attention should be focussed, and provide checking of expectations (both positive and negative) related to these conditions.

The third role is that of alerting models, in which knowledge about the likelihood of different classes of alerts or reports, time and other costs of transmission, and utility to different recipients or recipient classes is used to make rational choices about who to tell what, and when and how. Sensible engineering design calls for components that may be reused or adapted in subsequent designs.
The MAITA libraries provide means for abstracting, recording, and sharing monitoring networks developed for one purpose with developers of monitors for other purposes. These libraries aim to provide a broad and deep base of abstract and concrete monitoring methods, event descriptions, and alerting models.

## Progress

Prototypes of the MAITA system have existed since June 1998, and have been applied to demonstrating monitoring systems in battlefield information fusion and tracking, neonatal intensive care unit monitoring, and intrusion detection and response.

## Contact

Jon Doyle, doyle@mit.edu
Peter Szolovits, psz@mit.edu
http://www.medg.lcs.mit.edu/projects/maita

(FALL, 1999)

# Computer Graphics for Capture, Exploration, Design, and Simulation of Human-Scale Environments

*Leaders:*
Julie Dorsey, Leonard McMillan, Seth Teller

*Members:*
Matthew Antone,Ziv Bar-Joseph, Jaroslav (Yaroslav) Blagojevic, Mike Bosse, Hector Briceno, Eric Brittain, Britton Bradley, Chris Buehler, Max Chen, George Chou, Sterling Crockett, Barb Cutler, Fredo Durand, Steven Gortler, Adel Hanna, Aaron Isaksen, Robert Jagnow, Manish Jethwa, Kari Anne Kjolaas, Bhuvana Kulkarni, Justin Legakis, Peter Luka, Neel Master, Wojciech Matusik, JP Mellor, Byong-Mok Oh, Victor Ostromoukhov, Hans Kohling Pedersen, Gernot Schaufler, Robert W. Sumner, Osama Tolba, Stefano Totaro, Winston Wang, Rebecca Xiong, Jason Yang

*Sponsors:*
National Science Foundation

## Background

Our work focuses on several fundamental questions of Computer Graphics. First, how do we capture simulation data from the real world, efficiently? Second, once a large amount of data has been captured, how can we maintain responsiveness of interaction? Third, how can such data be used for effective design, analysis, and simulation? Fourth, how can graphics techniques such as interactive inspection, simulation, and visualization be used for enhanced pedagogy?

We have divided our work into four thematic areas:

*Capture*
This involves research into the automatic acquisition of 3D urban environments and botanical structures and computational video techniques for organizing enormous amounts of data.

*Exploration*
We are developing data structures, algorithms, and hardware for managing and interacting with extremely complex image, video, and geometric data-sets.

*Design and Simulation*
We are developing novel techniques to facilitate the interactive design and simulation of 3D environments, ranging from user-interface techniques to inverse techniques in which target performance is specified, and the system optimizes the underlying model to best meet the target.

*Education*
We are pursuing several educational initiatives, including a new computer graphics curriculum, a program to involve undergraduates in research, and techniques to facilitate interactive, exploratory learning.

## Why Is This Interesting?

Our work addresses several fundamental bottlenecks in modeling and simulation. We are pursuing semi-automated and fully automated methods for acquiring CAD data representing urban environments, which will reduce or eliminate the human time required for such capture operations. Data-sets routinely grow huge, and today's systems become unresponsive while processing them. We are developing novel data structures, algorithms, and systems which organize image, video, and geometry data to maintain responsiveness. One typically captures data in order to aid design and/or simulation processes. We are developing new design and simulation methods that incorporate text, still images, video, and geometry, and produce models of data and meta-data, for example about the source, creator, or use of the model. Finally, the human visual system is capable of absorbing large amounts of information over short times; we are exploiting this capability in developing new techniques for developing algorithms, and for teaching and exposition of our ideas.

## Approach

The graphics group is tackling several large, driving problems. How can we capture a CAD model of an entire campus, or small city? Can large numbers of images be used to produce a realistic, immersive experience of the imaged scene? How can simulation data be processed to include convincing signs of aging such as weathering and dirt? How can the human designer and simulation engine synergize to produce designs that optimize performance in the face of enormous number of constraints and design spaces with very many degrees of freedom?

## Progress

We have captured a CAD model of our immediate surroundings (Technology Square) from about 4,000 geo-located digital images. We have prototype image-based rendering techniques, and a prototype modeler -- something like a 3D photoshop -- that works directly on images. We are developing acoustic simulation engines that optimize concert hall design, and can predict and correct problems in existing concert halls. We have developed a platform-independent, network-transparent dataflow architecture for teaching and development of algorithmic concepts. We have a prototype, interactive renderer that uses only general-purpose computation, and no special-purpose graphics hardware.

## Future

There are many applications for these techniques. Visual simulation of a captured environment is one of our first, proof-of-concept applications. But these techniques are useful for training and rehearsal, for emergency response planning, for urban design and planning, for virtual tourism, for helping visually disabled people, and for education.

(FALL, 1999)

### Contact

Julie Dorsey, dorsey@graphics.lcs.mit.edu
Leonard McMillan, mcmillan@graphics.lcs.mit.edu
Seth Teller, seth@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/

(FALL, 1999)

## Computer-Assisted Sketching

*Leaders:*
Julie Dorsey, Leonard McMillan

*Members:*
Osama Tolba

*Sponsors*:
National Science Foundation, Sloan

### Background

Freehand sketching has long had appeal as an artistic medium for conceptual design because of its immediacy in capturing and communicating design intent and visual experience. We present a sketching paradigm that supports the early stages of design by preserving the fluidity of traditional freehand drawings. In addition, it attempts to fill the gap between 2D drawing programs, which have fixed views, and 3D modeling programs that allow arbitrary views. We implement our application as a two-dimensional drawing program that utilizes a projective representation of points -- i.e. points that lie on the surface of a unit sphere centered at the viewpoint. This representation facilitates the production of novel re-projections generated from an initial perspective sketch and gives the user the impression of being immersed in the drawing or space. We describe a method for aligning a sketch drawn outside the system using its vanishing points, allowing the integration of computer sketching and freehand sketching on paper in an iterative manner. The user interface provides a virtual camera, projective grids to guide in the construction of proportionate scenes, and the ability to underlay sketches with other drawings or photographic panoramas.

### Progress

“Sketching with Projective 2D Strokes.” O.Tolba, J. Dorsey, and L. McMillan. To appear in Proceedings of ACM UIST '99.

### Contact

Osama Tolba, tolba@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/~tolba/sketch

(FALL, 1999)

## Dynamically Reparameterized Light Fields

*Leader:*
Leonard McMillan

*Members:*
Steven Gortler, Aaron Isaksen

*Sponsors:*
Nippon Telegraph and Telephone

### Background

An exciting new area in computer graphics is the synthesis of novel images with photographic effect from an initial database of reference images. This is the primary theme of image-based rendering algorithms, which typically can create life-like novel images from actual photographs. This research extends the light field and lumigraph image-based rendering methods and greatly extends their utility, especially in scenes with much depth variation. First, we have added the ability to vary the apparent focus within a light field using intuitive camera-like controls such as a variable aperture and focus ring. As with lumigraphs, we allow for more general and flexible focal surfaces than a typical focal plane. However, this parameterization works independently of scene geometry; we do not need to recover actual or approximate geometry of the scene for focusing. In addition, we present a method for using multiple focal surfaces in a single image rendering process.

Images, usually described as a collection of pixels, can also be thought of as a database of rays. Each pixel corresponds to a ray of light that enters the focal point of a camera from a particular direction. By taking different rays from different cameras, depending on some parameterization, we can create new images and animations from places that our reference images were not taken from.

### Progress

Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. “Dynamically Reparameterized Light Fields.” Technical Report MIT-LCS-TR-778. May 1999.

### Contact

Aaron Isaksen, aisaksen@graphics.lcs.mit.edu
Leonard McMillan, mcmillan@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/~aisaksen/projects/drlf/index.html

(FALL, 1999)

## High Performance Visualization of Urban Scenes

*Members:*
Xavier Decoret (Project Members at iMAGIA), Julie Dorsey, Gernot Schaufler,
Francois Sillion (Project Members at iMAGIS)

### Background

This project is being researched in conjunction with iMAGIS in France and the Computer Graphics Group here at MIT.

### Approach

This project aims at the development of new visualization techniques allowing the interactive manipulation of urban data. Efficient visualization of 3D urban scenes is important for a number of applications such as: the evaluation of construction and renovation projects (site planning and visual impact studies); civil and military simulators (flight, drive, combat); navigation helpers for automobiles; virtual tourism and education; climate and environmental studies (plant growth in urban areas, detailed visualization of a number of simulations such as the diffusion of pollutants etc.); and city development planning.

The visualization of urban scenery is a very challenging problem because of the great complexity of these environments. Typical views of urban scenes contain very rich visual details at a fairly small scale, while the extent of the model is often very large (at least several square kilometers). Therefore the geometric and visual complexity of the models easily exceeds the memory and processing capacity of most visualization systems.

However, we observe that while urban scenes are extremely complex, they are also heavily structured. This structure is a consequence of the artificial nature of the built environments and reflects the spatial, social, and historical organization of these scenes (for instance in the form of the network of streets). The goal of this project is to exploit the structure of urban environments to offer interactive visualization techniques as well as efficient simulation tools tailored to urban scenery. To achieve this goal, efficient image caching and interpolation techniques will be combined with traditional 3D techniques. In particular, specific level of details will be associated with urban objects based on the underlying structure, and appropriate criteria will be devised for the generation and usage of cached images.

### Progress

"Multi-layered Impostors for Accelerated Rendering"
*Xavier Decoret, Gernot Schaufler, François X. Sillion and Julie Dorsey*
*Proceedings of Eurographics'99, Milan, Italy, September 1999.*

*"Efficient Impostor Manipulation for Real-Time Visualization of Urban Scenery"*
*François X. Sillion, George Drettakis and Benoit Bodelet*

(FALL, 1999)

## Contact

***iMAGIS***
Franois Sillion, Francois.Sillion@imag.fr
Xavier Decoret, Xavier.Decoret@imag.fr
http://www-imagis.imag.fr/VILLE/index.html

***MIT***
Julie Dorsey, dorsey@lcs.mit.edu
Gernot Schaufler, gs@lcs.mit.edu

## LiveWeb

*Leaders:*
Eric Brittain, Rebecca Xiong

*Sponsors:*
Intel Corporation

### Background

The LiveWeb Project seeks to enrich Web users' experience by visualizing the real-time activities of other users and enabling the user to interact with others. Currently, Web users have little knowledge about the activities of fellow users. They cannot see the flow of on-line crowds or identify centers of on-line activity.

The Web contains much information for discussion and is not limited by physical constraints on the number of users or their location. By allowing users to exchange information easily, the Web can become an ideal interaction environment for education, work, or entertainment.

### Progress

Visualization I: WebMap
WebMap overlays user activities over a simple 2D site map. For example, a research group Web site can be represented using a site map created from the floor plan. Each office contains one or more names that represent individual member's homepages. As Web visitors move from page to page, the corresponding icons will also move from room to room.

Visualization II: WebFan
WebFan visualizes user activity at WebBoards, which are Web-based message boards. It uses the thread structure of the WebBoard to lay out the messages in a fan-like fashion. It then overlays instantaneous and cumulative message accesses. Using this system, users can perceive broad activity patterns and individual behaviors of other Web users.

**Contact**

Rebecca Xiong, becca@graphics.lcs.mit.edu
Eric Brittain, ericb@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/liveweb

(FALL, 1999)

## Electronic Voting

*Leader:*
Ronald L. Rivest

*Members:*
Benjamin M. Adida, Brandon DuRette, Kevin Mcdonald

*Sponsors:*
Defense Advanced Research Projects Agency

### Background

The design of the voting scheme is based on the paper, "A practical secret voting scheme for large scale elections," by Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta (Proceedings AUSCRYPT "92, 1993, 244-251).

### Progress

May 25th, 1999 -- A second branch of the EVOX system has been created which uses multiple administrators for vote signing. This should improve the security by preventing the administrator from forging votes. This work was done as a Bachelors thesis by Brandon DuRette.

March 13th, 1999 -- EVOX has recently reached what we're calling version 2.0 beta. EVOX now has a new interface, built entirely in HTML. This interface now links to the EVOX Java applet via a few simple Javascript procedures, using Netscape Live Connect. The code for the servers has been cleaned up, and a generalized Object Store interface is now being used for the Anonymizer, so that the system can easily be switched over to a database-backed system whenever we get to using JDBC-to-Oracle. The system is now running the UA Elections, from March 10th to March 14th.

### Future

In no specific order, we plan to do the following over the next few weeks and months:

- Use JDBC to store votes in a database.
- Add ability to have multiple counters.
- Explore other protocols (modularize out the protocol?)
- Post full documentation of protocol, changes made to it for practical reasons, and Java implementation.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

# Private Information Retrieval

*Leaders:*
Shafi Goldwasser, Silvio Micali, Ronald L. Rivest

*Members:*
Amos Beimal, Yael Gertner,
Yuval Ishai, Eyal Kushilevitz, Tal Malkin

## Background

Private Information Retrieval (PIR) schemes allow users to retrieve information from a database while keeping their query private. Motivating examples for this problem include databases with sensitive information, such as stocks, patents or medical databases, in which users are likely to be highly motivated to hide which record they are trying to retrieve. PIR schemes aim at achieving this goal efficiently, where the main cost measure has traditionally been the communication complexity. PIR is a strong primitive, which may also be useful as a building block within other protocols.

## Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Shafi Goldwasser, shafi@theory.lcs.mit.edu

(FALL, 1999)

## Secure Multicast

*Leaders:*
Shafi Goldwasser, Ronald L. Rivest

*Members:*
Ran Canetti, Tal Malkin, Knobbi Nissim

### Background

Multicast communication is an attractive method for delivery of data to multiple recipients, minimizing consumption of both sender and network resources. Multicast is supported, for example, on the Internet, or via satellite communication. Some applications that benefit from multicast include real-time information update, multi-party conferencing, and pay T.V. Securing multicast communication poses several important challenges. In the paper, *Efficient Communication-Storage Tradeoffs for Multicast Encryption by Ran Canetti, Tal Malkin, and Kobbi Nissim. Eurocrypt 99,* we focus on providing access control, namely ensuring that only legitimate members of the multicast group have access to the group communication. This is done by maintaining a secret session key known to all legitimate members, and encrypting all group communication using this key. We consider a dynamic group, where users join or leave the group in an arbitrary fashion, and a center which is in charge of performing the re-keying associated with group updates. We require strong security, where the session key is secure against any coalition of non-members.

There is a variety of different scenarios using multicast, presenting a range of efficiency requirements with respect to several parameters. We give an upper bound on the tradeoff between storage and communication parameters. In particular, we suggest an improvement of the schemes by Wallner et al. and Wong et al. with sub-linear center storage, without a significant loss in other parameters. Correctly selecting the parameters of our scheme we can efficiently accommodate a wide range of scenarios. This is demonstrated by applying the protocol to some known benchmark scenarios.

We also show lower bounds on the tradeoff between communication and user storage, and show that our scheme is almost optimal with respect to these lower bounds.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu
Shafi Goldwasser, shafi@theory.lcs.mit.edu

(FALL, 1999)

## Self-Delegation with Controlled Propagation

*Leader:*
Ronald L. Rivest

*Members:*
Oded Goldreich, Birgit Pfitzmann

### Background

Self-Delegation with Controlled Propagation, by Oded Goldreich, Birgit Pfitzmann, and Ronald L. Rivest, introduce delegation schemes wherein a user may delegate rights to himself, i.e., to other public keys he owns, but may not safely delegate those rights to others, i.e., to their public keys. In the motivating application, a user has a primary (long-term) key that receives rights, such as access privileges, that may not be delegated to others, yet the user may reasonably wish to delegate these rights to new secondary (short-term) keys he creates to use on his laptop when traveling, to avoid having to store his primary secret key on the vulnerable laptop. We propose several cryptographic schemes, both generic and practical, that allow such self-delegation while providing strong motivation for the user not to delegate rights that he only obtained for personal use to other parties.

### Contact

Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

# GALAXY: Enabling Mobile and Affordable Access of On-Line Information using Spoken Dialogue

*Members:*
Jim Glass, T.J. Hazen, Lee Hetherington, Raymond Lau
Joe Polifroni, Stephanie Seneff, Nikko Ström, and Victor Zue

## Background

Since the late 1980s, the Spoken Language Systems has been conducting research leading to the development of conversational systems, systems that can converse with users in a spoken dialogue in order to fulfill their needs. A conversational system embodies several human language technologies. The spoken input is first processed through the speech recognition component. The natural language component, working in concert with the recognizer, produces a meaning representation. For information retrieval applications, for example, the meaning representation can be used to retrieve the appropriate information in the form of text, tables and graphics. If the information in the utterance is insufficient, the system may choose to initiate a dialogue with the user for clarification. Speech output can also be generated by processing the information through natural language generation and text-to-speech synthesis. Throughout the process, discourse information is maintained and fed back to the speech recognition and language understanding components.

## Why Is This Interesting?

The advent of the information age places increasing demands on the notion of universal access. For information to be truly accessible to everyone (especially the technologically naive), anytime, and anywhere, we must seriously address the issue of user interface. An interface based on a user's own language is particularly appealing, because it is the most natural, flexible, and efficient means of communication among humans. Conversational systems are particularly appropriate when the information space is broad and diverse, or when the users' requests contain complex constraints.

## Approach

Our research and development of human language technologies has been embedded in GALAXY, a system that enables universal information access using spoken dialogue. GALAXY has a distributed, client/server architecture that shares compute servers (for speech recognition and natural language understanding) and domain servers among many users, and relies on lightweight clients for input/output. The domain servers each encapsulate some area of expertise, and are each capable of dealing with a certain set of queries. They contain general knowledge about the structure of their domain in addition to the capability of accessing specific databases. The servers interpret user requests, locate required information, and compose a suitable response. The client program provides the interface to the user. It captures audio or typed input from the user, and presents the servers' responses using graphics, text, and synthetic speech. It is our intention to minimize the computational needs of the client program, thus providing information access to the widest user population in the most affordable way.

## Progress

The GALAXY system was first demonstrated in the spring of 1994. Since then, GALAXY has served as the testbed for our research and development of human language technologies, resulting in systems in different domains (e.g., automobile classified ads, restaurant guide and weather information), different languages (e.g., Mandarin Chinese and Spanish), and different access mechanisms (telephone-only or with displays). In 1996, we made our first significant architectural redesign to permit universal access via any web browser. The resulting WEBGALAXY architecture makes use of a "hub" to mediate between a Java GUI client and various compute and domain servers, dispatching messages among the various servers and maintaining a log of server activities and outputs.

In the process of developing dialogue modules for various domains in GALAXY, we came to the realization that it is critical to be able to allow researchers to easily visualize program flow through the dialogue, and to flexibly manipulate the decision-making process at the highest level. To this end, we developed a simple high-level scripting language that permits boolean and arithmetic tests on variables for decisions on the execution of particular functions. We found this mechanism to be very powerful, and were successful in incorporating it into our newest domain servers for weather and flight status information. We then began to contemplate the idea of incorporating an analogous mechanism into the program control of the entire system, which was being maintained by the GALAXY hub. In 1998, a new version of the architecture, called GALAXY-II, has been designed and implemented. In addition to serving our own needs, it has also been designated as the reference architecture for the DARPA Communicator Program, whose goal is partly to promote resource sharing and plug-and-play interoperability across multiple sites for the research and development of dialogue-based systems.

## Future

In the coming year, we will continue to refine and improve human language technology components in all areas, and to apply this technology in both existing and new application domains. One example of a new technology is in the area of dynamic vocabulary and language modeling, which would allow greater range and flexibility of coverage of our conversational systems. We will continue to make infrastructure improvements to the GALAXY architecture to enable faster prototype development in new application domains, such as flight-status information. We will develop the necessary tools and software so that application developers outside of our group will be able to develop their own applications for the DoD.

## Contact

Victoria Palay, palay@sls.ics.mit.edu
http //www.sls.ics.mit.edu

(FALL, 1999)

# Jupiter: A Spoken Language Interface to On-Line Weather Information

*Members:*
Jim Glass, T.J. Hazen, Lee Hetherington, Raymond Lau,
Joe Polifroni, Stephanie Seneff, Nikko Ström and Victor Zue

*Sponsors:*
Defense Advanced Research Projects Agency

## Background

Jupiter is a telephone-only conversational interface for weather information for more than 500 cities worldwide. To obtain weather information, a user simply picks up the phone and conducts a verbal dialogue with the computer. The weather information is obtained from four on-line sources on the Internet, and is updated several times daily.

## Why Is This Interesting?

By using the telephone as a means of delivering the information, we can empower a much larger population to access the wide range of information that is becoming available. Jupiter serves as a platform for investigating several research topics. First, in the scenario that we envision, a user could conduct "virtual browsing" in the information space without ever having to point or click. Second, displayless information access poses new challenges to conversational interfaces. If the information can only be conveyed verbally, the system must rely on the dialogue component to reduce the information to a digestible amount, the language generation component to express the information succinctly, and the text-to-speech component to generate highly natural and intelligible speech. Third, channel distortions place heavy demands on the system to achieve robust speech recognition and understanding. Finally, by applying human language technologies to understanding the "content," in this case the weather forecast, we can manipulate and deliver exactly the information that the user wants, no more and no less.

## Approach

Jupiter adopts a conversational paradigm, allowing the user to ask questions in natural ways and responding in a way a human conversational partner might. It employs Galaxy's client-server architecture, used previously to build conversational interfaces, except the client in this case is simply a telephone. It reuses all human language technology components, with some minor modifications. Finally, we have adopted the approach of building a first version of the system and making it accessible to users via a toll-free number. We were thus able to collect data from real users in real environments to further develop and evaluate the system.

## Progress

The Jupiter system has been available to the general public via a toll-free number (1-888-573-8255) since May, 1997. With only word-of-mouth advertising, we have been able to collect over 185,000 utterances from more than 30,000 callers. Currently, Jupiter understands about 80% of user queries, including those containing out-of-domain, out-of-vocabulary, and disfluent or accented speech. With correction dialogues, users usually can obtain the desired information through persistence. We have also incorporated a confidence measure, so that out-of-domain queries can be rejected.

### Future

While we are continuing to improve Jupiter's capabilities, we have also begun to investigate the issue of portability by moving on to a different language and domain. Currently, Jupiter is being developed for Japanese, Mandarin Chinese, and Spanish. Pegasus, a conversational interface for flight status, will soon be deployed.

### Contact

Victoria Palay, palay@sls.lcs.mit.edu
http://www.sls.lcs.mit.edu/jupiter

(FALL, 1999)

# SCIENCE & IT

# Local Rules Modeling of Virus Shell Assembly

*Leaders:*
Bonnie Berger, Russell Schwartz

## Background

Our work is aimed at developing a simulation tool for studying the formation of icosahedral virus protein coats. There are currently many open questions about how viruses form. One aspect of their assembly that has been particularly difficult to describe is the formation of their protein coats. These coats, which protect the viral genome, typically form from several hundred chemically identical proteins which are capable of spontaneously self-assembling into a complex but regular structure. The local rules model of (Berger et al., 1994) provided a framework for understanding this assembly process; this framework specifies the structure of a shell in terms of local binding interactions between different shapes, or conformations, of the coat protein. In the current work, we are attempting to use this framework to create computationally tractable simulations of the assembly process.

## Why Is This Interesting?

Icosahedral viruses include most human and animal viruses and are responsible for many diseases, including herpes, influenza, and some types of hepatitis. Currently, very few anti-viral medications have been developed. In part, this is because viruses can rapidly evolve their outer surfaces to protect against attacks on the mature virus. This suggests that attacking viruses during growth may be a more effective strategy. The eventual aim of this work is to develop methods to disrupt a particular aspect of viral growth: the assembly of their protein coats. However, the process of assembly is not currently well understood, making it difficult to develop such approaches. Current experimental methods have been unable to study the process in detail because of the small size of viruses and their rapid rate of assembly. Simulations have the potential to assist laboratory work in developing models of the assembly process, studying their behavior, and predicting how their behavior might be altered. However, previous simulation techniques have been unable to look at low-level kinetic processes such as this in a reasonable time for a problem as complicated as virus shell assembly.

## Approach

Our simulator is based on combining the principle of local rules with a molecular dynamics-like model to make computationally feasible simulations of assembly dynamics. When patterns of binding interactions can be determined, local rules provides a simple model for those binding interactions. With the local rules model, we can abstract away the complex combination of forces actually responsible for the binding interactions, allowing us to use a much higher-level model of individual viral coat proteins.

(FALL, 1999)

By using such a model, we can run simulations of the interactions hundreds or thousands of proteins in a reasonable time; this kind of simulation would be far beyond current computational capabilities if done with a more traditional molecular dynamics simulation done at atomic or amino-acid resolution. Furthermore, the simulator is developed to run in parallel using the Cilk multithreading system, allowing it to take advantage of parallel hardware to further extend the scope of systems we can model.

## Progress

We currently have a working simulator tool and have been using it to run simple experiments on virus assembly. The simulator combines a graphical user interface with a parallel implementation of the numerical methods to allow for simulations that are fast and easy to control. We have so far run several kinds of simulation experiments. One class of experiments involves testing how varying different parameters affects rates of growth and incidence of malformation. Another compares two different models of the source of nucleation-limited behavior, an important aspect of virus assembly kinetics. We have also conducted preliminary work modeling the disruption of virus growth through capsid assembly targeted (CAT) antivirals and looking at some non-virus self-assembly systems.

## Future

One major area of future work is improving the simulator itself. This is currently focused on developing numerical methods that are faster or parallelize better than the current methods. In addition, we are attempting to make the simulator easier to use, so it can be made accessible to experimental virologists. We are also working on refining our simulation models and extending them to more complicated viruses; experiments done to date have focused only on the simplest icosahedral viruses. We plan to look in more depth at our prior areas of experimentation, exploring the importance of different assembly parameters to the growth process. We also intend to create more sophisticated simulations of strategies for disrupting growth in the hope that these strategies can be applied to finding new anti-virals. Finally, we plan to extending the local rules model to non-virus systems, exploring other self-assembly systems in nature, as a means of testing and refining our model, and looking at the use of simulations in designing novel self-assembling materials.

## Contact

Bonnie Berger, bab@theory.lcs.mit.edu
http://theory.lcs.mit.edu/~bab/virus.html

(FALL, 1999)

## Heart-at-Home: Heart Failure Assistant

*Leader:*
Bill Long

*Members:*
Hamish Fraser, Shapur Naimi

### Background

Heart Failure is a chronic condition that is a major cause of death and disability, and the incidence of this condition is rising as the population ages. Over 2 million Americans suffer from congestive heart failure (CHF) and an additional 400,000 develop it annually. There are almost 1 million hospitalizations a year, many of which could be avoided if the early signs of recurring failure were recognized and treated at home.

MEDG has had a long-term interest in making medical care patient-centric. The guardian angel projects and heart failure presents an important opportunity to empower patients in the management of their disease in a way that can drastically cut the number of times they need to be hospitalized and ultimately increase their life expectancy and quality of life.

### Why Is This Interesting?

We are proposing to take over functions that have required human analysis and intervention in even the most forward looking of heart failure intervention programs. Typically the people monitoring patients have been specially trained cardiac nurses, limiting the applicability of the best programs to centers where such people are available. We expect that suitable software will make it possible for any heart failure patient to have this kind of care.

### Approach

The most important parameters for heart failure patients to watch are weight and symptoms. When the patient starts to put on fluid weight or experience shortness of breath or swelling of the extremities, the early signs of a relapse are present. We are developing a program that will log the daily weights of the patient, daily vital signs, symptoms, along with some simple questions about diet, exercise, and compliance with medications. With this information the program will look for trends in the weight and vital signs, which correlated with the other indication will give a reasonably specific indication of heart failure. When this is detected, the patient can make dietary corrections and an extra dose of diuretic to head off the relapse.

The system we are designing will consist of a module for the patient, the Heart-at-Home module, a Physician Assist Module, and a server to maintain data and communications. The patient module will monitor the patient, train the patient about their disease, provide information, advise the patient, and recognize situations in which the physician should be alerted. The physician module will keep the physician informed about the patient's status, provide the materials necessary to help the physician follow the appropriate heart failure guidelines, and generally assist the physician to manage the patient.

With this program we will be able to address all of the common treatable causes for patients to have heart failure episodes. One of the main reasons that people have episodes of heart failure is that they forget or stop taking their medications. Interacting with the Heart-at-Home system on a daily basis will also keep the patients engaged in their own care and more likely to maintain compliance. More than that, if the patient stops interacting with the program, the physician will know immediately and be able to contact the patient to correct the program.

## Progress

We are in the process of collecting patient data to analyze the characteristics of daily weights and vital signs. We already have data from a few heart failure patients gathered over the Web from which we have done initial analysis. We also have developed a data generator for generating artificial data of known characteristics to use in initial algorithm development. Finally, we have a Web based mock-up of the kind of Web site that patients will use for entering their data.

## Future

We are currently looking for funding for this project.

## Contact

Bill Long, wjl@mit.edu

(FALL, 1999)

## Acoustic Design and Modeling

*Leader*:
Julie Dorsey

*Members*:
Michael Monks, Byong Mok Oh

*Sponsor:*
National Science Foundation, Equipment Sponsor: Silicon Graphics, Inc.

### Project Synopsis

Acoustic design is a difficult process, because the human perception of sound depends on such things as decibel level, direction of propagation, and attenuation over time, none of which are tangible or visible. The advent of computer simulation and visualization techniques for acoustic design and analysis has yielded a variety of approaches for modeling acoustic performance. While these techniques certainly offer new insights into acoustic design, they fail to enhance the design process itself, which still involves a burdensome iterative process of trial and error.

The objectives of this research are twofold: 1) to refine and enhance the accuracy of acoustic simulation techniques while maintaining or enhancing computational tractability, and 2) to address the inverse problem of determining material and geometric parameters for an environment from a description of the desired acoustic performance.

### Papers In This Area

*"Acoustic simulation and visualization using a new unified beam tracing and image source approach." Michael Monks, Byong Mok Oh, and Julie Dorsey. Conference of the 101st Audio Engineering Society, Los Angeles (1996).*

*"Audioptimization: Goal based acoustic design." Michael Monks, Byong Mok Oh, and Julie Dorsey. Technical Report MIT-LCS-TM-588 (submitted for publication).*

### Contact

Michael Monks, mcm@graphics.lcs.mit.edu
Julie Dorsey, dorsey@graphics.lcs.mit.edu

(FALL, 1999)

## Educational Fusion: Internet Solutions for Computer Science Instruction

*Leader:*
Seth Teller

*Members:*
Aaron Boyd, Randall Graebner,
Kevin Kennedy, Bhuvana Kulkarni, Alexander Rodriguez

### Background

The Educational Fusion project was begun in January of 1996. We set out to study interactive algorithm development through the Web. We looked into methods for interactively developing algorithms including using basic building blocks that could be dragged into the algorithm to define the flow, and interactively editing source code.

### Approach

Educational Fusion is a system for developing and utilizing advanced interactive educational algorithmic visualizations on the World Wide Web. Educational Fusion allows educators to create interactive visualizations which require students to use problem-solving skills to implement solutions. The students can then immediately visualize their algorithm's output, and compare it to that of a hidden reference algorithm. This is an improvement over other educational systems which are limited to semi-interactive visualizations and simple question-and-answer forms.

We have created a framework for rapidly developing these visualizations. We have also created a general system for incorporating these visualizations into hierarchical concept graphs. Associated with these concept graphs is a representation of the human and other resources currently available to help in understanding the visualization, thus facilitating collaborative learning. We believe these techniques are highly applicable to knowledge domains in which students can gain greater understanding and express their competence by implementing a correct algorithm.

### Progress

Papers (Theses)

Aaron Boyd, Spring 99
Aaron's work on Fusion focused primarily on the actual teaching by visualization that is provided to the students by the system, and how these methods of teaching could be improved. This issue is the focal point of his thesis, which also contains details about other algorithm visualization systems in the educational community.

Nick Tornow, Spring 98
While Nick was a member of the Fusion team, he added a number of new features to the system. They included a way to record movies and a way for two users to share the same screen. Nick's thesis primarily covers his additions.

Brad Porter, Fall 97
Brad did a lot of the development behind the visualization panels. He also contributed heavily to the underlying architecture. His thesis, which is less technical than Nate's, contains a case study of Fusion's first use in a classroom environment.

Nate Boyd, Spring 97
Nate developed a good portion of the underlying architecture of the Fusion system. His thesis presents a good technical overview of Fusion in its early development. Although a lot has changed since Nate left the team, much of what he dicusses is still applicable.

Papers (Other Documents)
*The Fusion Editor, Fall 98*
*http://edufuse.lcs.mit.edu/fusion/people/bhuvana&randy/description.html This web page details the current state of the Fusion editor. It contains a brief overview of the architecture, a feature list, and plans for the future.*

*eFuse: A Platform for Collaborative Pedagogy, Spring 98*
*http://edufuse.lcs.mit.edu/fusion/papers/siggraph98/siggraph98.html SIGGRAPH 98 Educator's Program submission. Details the current state of the system and where we hope to be in the near future.*

*Developing Visualization Applets, Fall 97*
*http://graphics.lcs.mit.edu/~bwporter/docs/VizDeveloper/VizDeveloper.html Written for internal use and use by devlopers. This document provides an overview of how to create visualization applets for the Educational Fusion system based on the classes we have developed.*

*Educational Fusion Project, Fall 97*
*http://graphics.lcs.mit.edu/~bwporter/presentations/index.html Presented originally for Silicon Graphics, Inc. Provides an introduction to the motivations and objectives behind the project.*

*Fusion: A Trial Run Fall 97*
*http://edufuse.lcs.mit.edu/fusion/papers/Fall97/sld001.htm*
*Results of the trial run on the Computer Graphics class (6.837) from Fall term, 1997. Details the state of the system in the Fall, and gives a brief organization outline.*

### Contact

Seth Teller, seth@graphics.lcs.mit.edu
Aaron Boyd, aaronbo@graphics.lcs.mit.edu
Randy Graebner, randyg@mit.edu
Kevin Kennedy, kkennedy@graphics.lcs.mit.edu
Bhuvana Kulkarni, bhuvana@mit.edu
Alex Rodriguez, alexrodr@mit.edu
http://edufuse.lcs.mit.edu/fusion/

*Other related groups here at MIT include the Rivet Virtual Machine group (see http://sdg.lcs.mit.edu/rivet.html for more information) and The Scheme Group (see http://www-swiss.ai.mit.edu/projects.scheme for more information).

(FALL, 1999)

# Zero-Knowledge Proofs

*Leaders:*
Shafi Goldwasser, Silvio Micali, Ronald L. Rivest

*Members:*
Cynthia Dwork, Oded Goldreich
Daniele Micciancio, Moni Naor, Tal Rabin, Amit Sahai, Salil Vadhan

## Background

Zero-knowledge proofs are probabilistic and interactive proofs that efficiently demonstrate membership in the language without conveying any additional knowledge. Zero-knowledge proofs were introduced by Goldwasser, Micali and Rackoff in "The Knowledge Complexity of Interactive Proof Systems," (SIAM J. of Computing, January 1989). The wide applicability of zero-knowledge was demonstrated in Proofs that Yield Nothing But their Validity or All Languages in NP have Zero-Knowledge Proofs, co-authored by Goldreich, Micali and Wigderson [JACM, July 1991]. In particular, assuming the existence of one-way functions, they showed that every language in NP has a zero-knowledge proof system. This work is not available on-line, yet most of the material is covered in Foundations of Cryptography - Fragments of a Book [by Oded Goldreich].

## Contact

Shafi Goldwasser, shafi@theory.lcs.mit.edu
Ronald L. Rivest, rivest@theory.lcs.mit.edu

(FALL, 1999)

## Amorphous Computing

*Leader:*
Hal Abelson

*Members*:
Don Allen, Daniel Coore, Chris Hanson, Tom Knight,
Radhika Nagpal, Erik Rauch, Darren Schmidt, Latanya Sweeney, Chris
Terman, Bei Wang, Ron Weiss, Jeremy Zucker

### Background

A colony of cells cooperates to form a multicellular organism under the direction of a genetic program shared by the members of the colony. A swarm of bees cooperates to construct a hive. Humans group together to build towns, cities, and nations. These examples raise fundamental questions for the organization of computing systems: How does one obtain coherent behavior from the cooperation of large numbers of unreliable parts that are interconnected in unknown, irregular, and time-varying ways? What are the methods for instructing myriad's of programmable entities to cooperate to achieve particular goals? These questions have been recognized as fundamental for generations. The objective of this research is: to identify the engineering principles and languages that can be used to observe, control, organize, and exploit the behavior of programmable multitudes. This effort is the study of Amorphous Computing.

### Why Is This Interesting?

Microelectronic components are so inexpensive that we can imagine mixing them into materials that are produced in bulk, such as paints, gels, and concrete. Engineers can exploit the resulting "smart materials" to reduce the need for strength and precision in mechanical and electrical apparatus, through the application of computational intelligence. For example, we can imagine coating a building or a bridge with "smart paint" that reports on traffic loads and wind loads, and monitors the integrity of the structure, or even heals small cracks by shifting material around. A clean room with "active skins" lined with cilia can push dirt and dust into a corner for removal. A wall that could sense vibration (and move slightly on its own) could monitor a premises for intrusion or it could actively cancel noise.

### Approach

We are developing a suite of tools for organizing local geometry, topology, and communication that are an appropriate substrate for the expression of amorphous computing algorithms. They are working on an expressive high-level language for developing software for amorphous computing systems. The language will integrate a geometric and topological toolkit to provide convenient primitive building blocks, appropriate combinators for making compound structures, and means of abstraction for common patterns of usage. They are developing and will report on algorithms for amorphous systems that, by construction, obey the fundamental conservation laws of physics. Such algorithms, based on exchange processes, will be used to implement high-performance programs for the precision integration of partial differential equations representing important physical systems on an amorphous computer.

## Progress

On the language development front, we have demonstrated (in simulation) how amorphous processes can use organizational ideas inspired by plant growth to provide precise topological control over the generation of structures (e.g., as in electrical circuits). We have also demonstrated how amorphous computing systems can control differentiation and growth via a SIMD-type language of marker and message propagation. In addition, we are designing a chip that combines a SPARC processor and a CDMA communications system on a single chip, to serve as a test bed for amorphous computing ideas. Fabrication of the communications module has been successfully completed.

## Future

We will continue the fundamental development of software to support amorphous computing. We are preparing the first draft of a language appropriate for organizing myriad's of nameless computing particles to achieve particular goals. In this target language it should be easy to express, for example, algorithms that could be used to determine the topology of a surface covered with programmable paint. Surface topology is one of the most important properties that smart paint has to discover about the surface it coats. For example, in order to be able to report cracks in a bridge, our amorphous computer must build a description of the global geometry of the bridge; but generating a geometrical description of the bridge requires the knowledge of its topology.

This research project has a long time horizon. We do not expect extensive commercial or military application of these ideas very quickly, but we think that we will uncover principles that will become dominant in the next generation of advanced engineering systems.

We expect that there will be some short-term military and commercial spin-offs, but the major impact will be through the students we graduate. One example of a spin-off result with short-term commercial interest will be the Big Cheap Display currently being developed by Knight and Surati.

## Contact

Hal Abelson, hal@mit.edu
Gerald Jay Sussman, gjs@mit.edu

(FALL, 1999)

# Biological Computation

*Leaders:*
David K. Gifford, Alexander J. Hartemink, Julia Khodor

## Background

Biological computation has attracted a lot of interest because of its potential for high performance parallel computation. In 1994 Professor Gifford proposed a new technique for implementing biological computation called programmed mutagenesis. Previous techniques relied on generating a large set of witness molecules of DNA and then searching among those molecules for the ones that satisfy a given set of constraints. In contrast, programmed mutagenesis is based on a string rewriting model of computation wherein a strand of DNA encodes a string.

DNA strand replication provides the ability to copy molecules as well as the opportunity to introduce sequence specific changes into newly synthesized molecules.

Programmed mutagenesis is an in vitro mutagenesis technique that uses DNA polymerase and DNA ligase to create copies of template molecules where the copies incorporate engineered mutations at sequence specific locations. Each time a programmed mutagenesis reaction is thermal-cycled, a rewriting event occurs. Because the technique relies on sequence specific rewriting, multiple rules can be present in a reaction at once, with only certain rules being active in a given rewriting cycle. The system's ability to accommodate inactive rules allows it to proceed without human intervention between cycles.

## Laboratory Exploration of a Programmed Mutagenic Unary Counter

We are exploring in the laboratory a simple programmed mutagenesis system, the unary counter.

Over the last year we have successfully demonstrated the ability to increment the counter in the first cycle and the ability to increment the counter in the second cycle if and only if the counter was successfully incremented in the first cycle, as desired.

Currently, we are focusing on evaluating the performance of the advanced cycles of the unary counter machine. We have nearly completed our experiments investigating the third cycle, and we have begun investigating the fourth cycle as well.

We are intrigued by the possibility that a living cell might be computing its way through its life cycle. Our future research directions include using programmed mutagenesis in the framework of directed evolution. We are also interested in studying the possible role that programmed mutagenesis-like events might play in evolution and development.

(FALL, 1999)

## Computer-based Tools for Simulation, Analysis, and Design of Biological Computation

Based on our experience implementing computational systems in the laboratory, we have gained insights into some of the fundamental constraints imposed on biological computational models as they are put into practice. This knowledge has helped us develop a number of tools for the simulation, analysis, and design of biological computation.

In order to provide a means for rapidly testing hybridization specificity hypotheses, a simulator called BIND has been developed.

When BIND's predictions are compared with melting temperatures reported in the literature, the distribution of error is roughly Gaussian, with a mean of 0ºC and a standard deviation of 3ºC n=93. To simplify the design of our computational systems, we developed a tool for constraint-based selection of nucleotide sequences. This tool incorporates domain knowledge that has proven to be important in our experimental process. We also formulated a framework for systematically solving a general nucleotide selection problem and produced the program SCAN to assist in the selection process.

SCAN exploits the computational melting temperature primitive contained in BIND to search a "nucleotide space" for sequences satisfying a set of pre-specified design constraint utilized 24 hours of compute time to search a space of over 7.5 billion unary counter designs and found only 9 designs satisfying all of the pre-specified constraints. One of SCAN's designs has been implemented in the laboratory and has shown a marked performance improvement over the products of previous attempts at manual design. This design is the one currently under investigation.

We are continuing to develop tools and algorithms to provide a greater understanding of the systems we are exploring in the laboratory. Ongoing work is in two primary directions: The first is a theoretical exploration of the computational power of various biological computation systems. The second is the development of a more powerful simulation tool, capable of predicting the complete array of DNA strands that are produced during each cycle of a programmed mutagenesis reaction.

### Contact

http://www.psrg.lcs.mit.edu/

(FALL, 1999)

# TRANSPORT

## Automatic Adaptive Network Operation: Delivering Better Performance without Human Intervention

*Leaders:*
David Clark, Karen Sollins, John Wroclawski

### Background

Robust networks must adapt to changing conditions: changes that occur over a range of time-scales. In the very short time-scale, statistical overloads can lead to inappropriately lost or delayed traffic and poor performance. Especially in military networks, externally forced sudden changes in topology or capacity can demand corresponding change in application behavior and priority. Over longer time scales, the network can re-deploy its communications assets in different ways to meet evolving needs. Today, short time-scale performance adaptation occurs more slowly than desirable, especially when individual traffic flows are using a significant part of the total capacity (a condition more often true of military networks). Mid-and long-term network engineering and reconfiguration usually involves human intervention, which implies slower response and chance of human error.

The goal of this project is to develop new strategies for network design that lead to effective and automatic adaptation of the network to its changing environment at many time-scales.

### Why Is This Interesting?

Networks that adapt their performance to changing conditions effectively, rapidly, automatically, and appropriately will reduce the human effort of deployment and operation, broaden the range of conditions over which applications can operate, and increase the reliability of the delivered service.

We identify two major limitations with current technology. First, current algorithms tend to depend on a consistent global view of network conditions, and seek optimal solutions in that context. Second, current algorithms typically operate based on measures of function and performance, but not policy and administrative constraints. By breaking free of these limitations, we expect to demonstrate new classes of network algorithms and protocols that support these highly adaptable networks.

### Approach

Our approach is based on several core concepts:

- Rule-based expression of local administrative and performance metrics.

- Ability to bound rules and heuristics that are applicable only within physical or virtual domains or "regions." One form of this regionalization is expressed in our work on the Metanet.

- Algorithms that make decisions based on partial and local knowledge. These algorithms may sacrifice optimality for operationally simple and effective behavior.

- Use of "underdamped" control algorithms that apply short time-scale active control to an otherwise unstable system, in order to achieve faster adaptation to changing conditions.

- A scalable multicast-based strategy for implementing software agents that automatically locate, monitor and report on events of interest within the network.

- Support of basic network functions through sets of composable lower-level building blocks, as opposed to fixed pre-specified services.

## Progress

Our past work in this area has focused on the controlled allocation of network resources to different applications with different service requirements, and on improving the adaptation of applications and network protocols to changing network conditions. We have developed the definition of the real time service on the Internet, the Controlled Load service. We have proposed an approach called RIO to allocate the bandwidth of a data network to different data transfers in a way that is highly scalable and explicitly controllable. We have proposed extensions to current congestion controls in the Internet, and explored alternatives based on explicit rate-based feedback.

## Future

The successful completion of this project will lead to networks that operate effectively and deliver good performance in the face of rapidly changing conditions and incomplete global knowledge. These networks will configure and monitor themselves without significant human intervention, using algorithms that consider administrative as well as technical requirements. Among the specific examples of this are:

- High-performance applications that learn of specific critical conditions such as persistent congestion, lossy or failing paths, and so on, and reconfigure internal structure or adapt behavior to meet performance goals.

- Agents working on behalf of a high-performance application will operate within the network, converging on critical points and reporting current conditions while they facilitate the traffic of the application.

- Regionalization primitives create local contexts for algorithms or administrative constraints, in order to achieve global goals by piecewise composition.

## Contact

David Clark, ddc@lcs.mit.edu
http://www.ana.lcs.mit.edu

(FALL, 1999)

# Beyond the PC: The Convergence of Networks, Systems, and Applications

*Leaders:*
David Clark, Karen Sollins, John Wroclawski

## Background

This project imagines a world where computing moves from a PC or Client-Server model to an Ensemble model. Much as increasing demand and decreasing cost of computing resources triggered a move from timesharing to the personal computer, we expect current technology trends and application requirements to trigger a move from the personal computer to the networked appliance, the sensor net, and the highly mobile user interface device. We assume that users of tomorrow will interact with many computers. Some of these will continue to support general purpose computing, some will be dedicated to providing human-level interfaces, and some will be embedded in the environment in which we sit -- smart devices, sensors, actuators and so on.

## Why Is This Interesting?

In the Ensemble model, both hardware and software are composites of elements connected to the network. These components dynamically come together in various arrangements to execute applications. This has many implications. Three of these are:

- The user may employ different interface devices depending on his current circumstances and skills. This implies that applications must be able to adjust themselves to different human interface modalitiesñ voice, keyboard, pointer, with output of speech, text, or images.

- Sensors or inputs may be dynamically activated depending on the current need for data gathering.

- The selection of resources used by an application must be determined not only by the physical availability of hardware, software and network resources, but also overall policy constraints constraining their use.

These concepts describe an environment where applications rapidly crystallize as and when required, rather than being statically constructed to meet a preconceived set of circumstances.

## Approach

Our approach centers around three ideas described below: the application catalyst, the scoped region, and next generation heterogeneous networking. We anticipate that a number of other technologies will become equally important as the work progresses.

Construction of applications by means of catalysts. In a decentralized network of interfaces and computing components, the application code does not run in any one place, but on some connected subset of the available devices. The application, instead of being shrink-wrapped object code for some processor, becomes a set of instructions for composing available devices and instantiating the running code on whatever devices it composes. We call these instructions an application catalyst, since they contain the information to construct the running application. To ensure that the application continues to function well, the catalyst will be long-lived and continue to monitor the effectiveness of a particular realization of the application, re-catalyzing the application as needed to meet changing conditions.

Scoping the operation of application construction. A catalyst should only take advantage of those components that it is supposed to use, a definition that includes both technical and non-technical constraints. This set of usable parts must be identified in some way. Manual cataloging has unacceptable overhead. Purely technical metrics, such as network topology, are not adequate to bound the scope of organization in real settings. What is needed instead is some concept that captures the idea of administrative scope, or shared trust. We identify the scoping region as a basic building element of a next generation network architecture. We propose efficient and intuitive ways to introduce new components into a region of trust so that the necessary associations are created naturally.

Next generation heterogeneous networks.  Today's Internet provides a set of conventions that permit seamless end-to-end communication even though multiple sorts of physical-layer network technology are in use. In the post-PC world of tomorrow, there will likely be an increasing range of different network technologies, which will address different needs for mobility, speed, cost and so on. This implies a new generation of router, suited for personal portable or residential use, which provides interworking among the different sorts of next generation local networks, and  connects between these and the wide area Internet of tomorrow. In particular, connecting personal mobile devices to a wide-area wireless infrastructure will require a personal wireless router, which will permit a new  generation of networked personal devices to come into existence.

## Progress

This program is just starting. A key activity at present is our work, in cooperation with the Lab's Spoken Language Systems group, to develop the system architecture, device hardware, and supporting technology for a new generation of cognitive network appliances. Our goal is to allow users of the system to interact with information servers, other, collaborating users, and whatever other networked resources are available in a given environment entirely through human-level interface such as speech understanding and gesture recognition.

Our development of system-wide protocols and tools for this new environment builds strongly on our related past work. Examples include the M.I.T. Internet Telephony Consortium, which we lead, and the Information Mesh Project, recently completed by our group. The M.I.T. Internet Telecommunications Convergence Consortium looks at applications that take us beyond the PC to new modes of human interaction over the Internet. One major aspect of the Information Mesh Project has been the development of practical globally unique, long-lived identifiers, and the necessary location resolution infrastructure required to realize them. These persistent identifiers are a key to catalysis and scoping.

## Future

The concepts presented above under Approach represent the starting points of this project. Two system elements we expect to develop are the Abstract Interface Toolkit and the Ensemble Monitoring Architecture.

The Catalyst Support Environment supports a catalyst's ability to locate, compose, invoke, and monitor distributed computations, with focus on evaluation of performance and runtime changes in behavior to improve functionality, or performance, of catalyzed software. The framework implements the structure and mechanisms necessary for catalysts to understand their operating environment in reasonably abstract terms, and provides the tools needed for catalysts to map between functional specification and a particular run-time instantiation within specific programming language and hardware environments.

The Abstract Interface Toolkit provides an abstraction boundary between application and a variety of different human interfaces, such as speech-based, written, keyboard, and so on. This is a dramatic generalization of today's Abstract Windowing Toolkit (Java) and similar ideas, which provide some application independence from the underlying infrastructure but are restricted to a single modality of interface.

## Contact

David Clark, ddc@lcs.mit.edu

(FALL, 1999)

# Network Aware Internet Video Encoding

*Leader:*
Leonard McMillan

*Members:*
Hector Bricenor, Steven Gortler

*Sponsors:*
Defense Advanced Research Projects Agency

## Background

The distribution of digital video content over computer networks has become commonplace. Computer networks are extremely diverse, users receive data via phone, cable, satellite or wireless. These networks are prone to packet loss. Unfortunately, most digital video encoding standards do not degrade gracefully in the face of packet loss, which often occur in a bursty fashion.

## Approach

We propose a new video encoding system that scales well with respect to the network's performance and degrades gracefully under packet loss. Our encoder sends packets that consist of a small random subset of pixels distributed throughout a video frame. The receiver places samples in their proper location (through a previously agreed ordering), and applies a reconstruction algorithm on the received samples to produce an image. Each of the packets is independent, and does not depend on the successful transmission of any other packets. Also, each packet contains information that is distributed over the entire image. We also apply spatial and temporal optimization to achieve better compression.

## Contact

You can find more information on this, in the MIT/LCS Technical Memorandum #591 - Network Aware Internet Video Encoding (NAIVE).

Hector Briceno, hbriceno@lcs.mit.edu
Steven Gortler, sjg@cs.harvard.edu
Leonard McMillan ,mcmillan@lcs.mit.edu
http://graphics.lcs.mit.edu/~hbriceno/naive

(FALL, 1999)

# Computational Video

*Leaders:*
David Gifford, Leonard McMillan

*Members:*
Hector Briceño, Osama Tolba

*Sponsors:*
Defense Advanced Research Projects Agency

## Background

Our perception of "video data" has been significantly influenced by its long history. Video was designed primarily as a broadcast medium with the major processing costs centralized at the point of production in order to minimize the requirements of the consumer.  However, in today's world of low-cost computation resources it seems worthwhile to revisit many of the underlying assumptions that have governed the evolution of video. This is of particular interest when we consider the future development of digital-video standards. Modern digital video requires that a certain amount of processing power be resident at the receiver. We expect that these processing capabilities will migrate toward general-purpose computing models in order to allow for future growth and enhancements. The computational video project summarized here explores how to make the best use of this client-side computing resource and the implications of such a capability on the video medium as a whole.

The fundamental principle of computational video is the encapsulation of video as objects containing data, representing one or more video streams, and methods for operating on video data. This formalism allows common processes associated with video, such as decompression, video editing, and image effects, to be handled in a uniform and maintainable fashion. Using this model we can also readily extend the capabilities of video to include queries, segment switching, information mining, and content playback or rendering. This new form of multimedia will also allow new collaborative applications between multiple viewers.

Unlike conventional digital-video streams, computational video includes computational elements embedded within its data stream. These computational elements, or methods, support dynamic and content-specific user interactions, such as menus, dialogs, conferencing facilities, transaction capabilities, graphic overlays, and indexing frameworks. These methods can be transported to specific clients or broadcast to larger audiences as needed. Multiple video streams can be associated, configured, and controlled using methods attached to one or more video streams. Thus, computational video becomes a natural framework for shared multimedia conferencing applications. In a computational video system, multiple video streams from remote sources can be combined and presented to a number of viewers. The same system can further provide viewers with their own customized view of each session.

Our design partitions computational-video objects into two separate layers. The video component layer provides simple and uniform interfaces to common operations on video-data streams. Video components play the same role in computational video that a standard library plays in a programming language. For example, video components provide for operations such as decompression, spatial filtering and image resizing. Since all components adhere to a common interface, their functionality can be overridden by either substituting one component for another, or cascading multiple components to achieve their aggregate effect. The second layer of a computational-video object is its script layer. The scripting layer is a simple, yet complete, programming language that allows for the declaration of video data and expression of operations upon that data. Each properly formed video script is itself a video component that can be included in other video scripts, or used to override other video components. Our goal is to make this video-scripting language as simple and easy to use as HTML (hypertext markup language). This will allow a novice computational-video user to rapidly construct simple multimedia objects, while also allowing sophisticated users to construct complex applications and rapidly prototype new ideas.

## Contact

Osama Tolba, tolba@graphics.lcs.mit.edu
Hector Briceño, hbriceno@graphics.lcs.mit.edu
Leonard McMillan, mcmillan@graphics.lcs.mit.edu
http://graphics.lcs.mit.edu/cv

(FALL, 1999)

## CLICK – A New Software Architecture for Building Flexible and Configurable Routers

*Leader:*
Frans Kaashoek, Robert Morris

*Members:*
Benjie Chen, Edward Kohler, John Jannotti

*Sponsors:*
Defense Advanced Research Projects Agency,
Intel Corporation, National Science Foundation

### Background

Click is a new software architecture for building flexible and configurable routers. A Click router is assembled from packet processing modules called elements. Individual elements implement simple router functions like packet classification, queueing, scheduling, and interfacing with network devices. Complete configurations are built by connecting elements into a graph; packets flow along the graph's edges. Several features make individual elements more powerful and complex configurations easier to write, including pull processing, which models packet flow driven by transmitting interfaces, and flow-based router context, which helps an element locate other interesting elements.

We have built several working configurations, including an IP router and an Ethernet bridge. These configurations are modular -- the IP router has 16 elements on the forwarding path -- and easy to extend by adding additional elements, which we demonstrate with augmented configurations. On commodity PC hardware running Linux, the Click IP router can forward 64-byte packets at 73,000 packets per second, just 10% slower than Linux alone.

### Contact

Frans Kaashoek, kaashoek@lcs.mit.edu

(FALL, 1999)

## Removing Barriers To Use

*Leaders:*
David Clark, Karen Sollins, John Wroclawski

### Background

Computing and communications are evolving rapidly, but they are still far from mature. For many people, they have not yet proved their value, they are not available at reasonable price and performance, and they are still difficult to install and use. Major barriers must be overcome to accelerate the useful and effective deployment of computers in society. There are issues of policy and economics that bound this effort, but serious technical factors are key in further progress. These apply in common to the military and civilian sector.

### Why Is This Interesting?

The worth of computing is defined by value to the users. In order for technical accomplishments to be translated into successful products, these products must meet users' needs, so that there is motivation to deploy them. By specifically understanding where short-term utility is found, we can shape the development and deployment of technology, leveraging this short-term utility to achieve longer-term benefit in both civilian and military sectors. Further, when the barriers to deployment are largely non-technical, we can often discover specific technical questions to address that can mitigate these barriers.

### Approach

In general, our approach is to stimulate a two-way interaction between the world of technology and the larger context of policy, economics, human factors, and so on. Our work attempts to engage those larger communities, in order to inform them of what the real technical limitations and possibilities are, and in order to learn what the real technical problems are.

A specific case is the slow pace of deployment of broadband communication to the home, small business and other geographically distributed end-points. The real barriers to deployment are economic and political, and this fact raises several important technical questions. First, what is the service that a link should support in order to be called broadband? This question is central in current policy debates. Second, what interfaces should a broadband local loop technology offer in order to support vigorous and fair competition for higher level services, even if competition cannot sustain the loop itself? Third, are there technical alternatives that would reduce the cost of installation, in order to prove the market? Wireless is an interesting alternative where technical and non-technical issues combine to gate progress.

In addition to broadband access, we have identified other specific areas where technology issues intersect strongly with the larger set of factors that gate the future of communications for computers. These include issues of pricing for services, lack of an agreed service model for the network, and difficulties in deployment and use.

### Progress

We have explored economic issues in the design of the Internet by developing general pricing models for network service, models that permit a range of higher-level applications to map their pricing models onto the lower level payment schemes. This has exposed us to some of the larger economic issues, and provided technical context to economists working in this area. This work has contributed to other group projects on bandwidth allocation, as well as specific projects on shared payment plans for the Internet.

### Future

As computer network technology matures, the nature of research must change. The question is no longer "can we build something," but "what should we build that matters. "Our goal is to insure that our research remains relevant by relating it to larger societal issues. We believe that this interdisciplinary component of research will be necessary to success, and the military must particularly come to understand this larger context, so that they can direct the future so as to obtain the maximum value from COTS devices.

### Contact

David Clark, ddc@lcs.mit.edu
http://www.ana.lcs.mit.edu/

(FALL, 1999)

## SpectrumWare

*Leaders:*
Stephen Garland, John Guttag

*Members*:
John Ankcorn, InterSync Corporation, Sunil Rao, Michael Saginaw, Brett Vasconcellos, Matt Welborn

*Sponsors:*
Defense Advanced Research Projects Agency

### Background

There is a rising demand for wireless communications systems that can adapt rapidly to specified changes in their functionality or to dynamic channel conditions, and be easily modified to perform unanticipated functions. The objective of the SpectrumWare project is to demonstrate that both of these needs can be met by moving the hardware/software and analog/digital boundaries close to the antenna, and then doing all of the digital signal processing in software running on commodity workstations.

### Why Is This Interesting?

Present day communication systems have two characteristics that limit their flexibility: 1) they subject raw signals to substantial analog pre-processing in advance of A/D conversion, and 2) they perform digital processing primarily using customized, rather than general purpose, hardware and software.

Recent hardware developments make it possible to extend the software processing domain to encompass a wide range of sampled signals. Digital technology is flexible and can process signals arising from speech, video, ultrasound, radar, and baseband or IF radio.

### Approach

A new portable programming environment, Spectra, supports the modular construction of realtime signal-processing applications. Spectra separates the signal-processing function into three areas: composable elements, data management for intermediate results, and functions for scheduling calculations. This separation enables a high degree of code reuse across varying applications and execution environments.

The elimination of dedicated hardware introduces flexibility into wireless communications systems and transducer-based instruments. We use this flexibility to ensure that the system maintains a balance between resource consumption (specifically of computation, power, and spectrum) and output signal quality, while ensuring that end-to-end performance specifications are met.

## Progress

Spectra has been used to build a number of software radio devices:

- a multi-mode/multi-band receiver (AM/FM radio, television audio, walkie talkie, and cellular telephone),
- a multi-channel receiver (four simultaneous channels)
- a software patch panel for communicating between "incompatible" devices that use differing modulation schemes in separate radio frequency (RF) bands, and
- a black and white NTSC television receiver.

Experience in building these applications has led to new algorithms for signal processing and has provided a deeper understanding of the data flow and computations they require.

## Future

The SpectrumWare project will focus on five different challenges in the future:

- Providing high-bandwidth, continuous I/O streams
- Refining a programming environment for building adaptive real-time signal processing systems
- Developing novel signal processing algorithms that take advantage of the assets available on a general-purpose platform.  The temporal decoupling offered by this environment offers excellent opportunities in areas such as randomized and approximate signal processing
- Taking advantage of the fact that communication can be closely integrated with applications.  This can lead to new functionality as well as improved management of resources.
- Applying advanced compiler technology to combine and target software modules for execution on reconfigurable hardware architectures.

## Contact

John Guttag, guttag@lcs.mit.edu
Stephen Garland, garland@lcs.mit.edu

(FALL, 1999)

## The Congestion Manager (CM)

*Leader*:
Hari Balakrishnan

*Members:*
David Andersen, Deepak Bansal, Dorothy Curtis, N. A. B. Priyantha

*Collaborator*:
Srinivasan Seshan, IBM, T.J. Watson Research Center

*Sponsors*:
IBM, Nippon Telegraph and Telephone

### Background

Congestion management is one of the fundamental problems in Internet research today. While this has always been a problem, it has become a critical challenge now because of the increasing presence of non-adaptive real-time streaming applications and the rapid growth of the Web, whose traffic characteristics make adapting to congestion especially hard. A new approach is needed to overcome these challenges.

### Why Is This Interesting?

Real-time streaming applications run over UDP, and in most cases today do not employ any adaptive congestion control. To obtain the best performance and user-perceived behavior, applications need to learn about and adapt to changing network conditions such as bandwidth, latency and loss rate. Today, there is little system support for this. In addition, many applications are characterized by multiple concurrent, often-short flows between sender and receiver. While TCP (the reliable transport protocol of choice in the Internet) does implement congestion control, most TCP transfers today are short and do not give TCP enough time or information to adapt to the state of the network. This is exacerbated by concurrent connections to the same client that compete with each other for scarce resources. Our system provides an API for applications to adapt and ensures stable network behavior by implementing safe congestion control algorithms.

### Approach

We use theoretical and experimental analyses to demonstrate several shortcomings in current approaches to Internet congestion management. We address these drawbacks by designing and implementing a novel end-to-end congestion management architecture for Internet hosts. Our solution is a transport- and application-independent congestion management sub-layer (the Congestion Manager, CM) that operates primarily at the data sender, between the transport and network layers of the Internet protocol stack. This architecture enables the sharing of path characteristics and congestion state amongst different flows via the CM. In addition, the CM exports a simple API that allows applications (e.g., audio, video, etc.) and transport protocols to use the CM and adapt to network congestion.

Our theoretical analysis studies the trade-offs between flow throughput and packet loss rate in the network for various classes of linear and non-linear control protocols. The end-result is a formal characterization of the necessary and sufficient properties that any stable congestion control protocol must possess.
Increasingly, the trend is for unicast data servers to transmit a wide variety of data, ranging from best-effort streaming content to reliable Web pages and applets. Our design of the CM is motivated by the observation that in the future Internet, many logically different streams using different transport protocols will share the path between server and client. They all have to dynamically probe for bandwidth and adapt to congestion for the Internet to be stable. Rather than have each stream act in isolation and thereby adversely interact with the others, our CM uses "shared learning" to maintain host-specific and domain-specific path information. Thus, path properties are shared between different streams and all transmissions are performed only with the CM's consent. In addition, we design a simple API to allow applications to adapt to congestion, accommodating the heterogeneous nature of streams. Another innovation is the use of hints from the receiver to the sender's CM to flexibly partition the available bandwidth amongst the different streams. One of the challenges the CM has to overcome is the coexistence of rate-based and window-based adaptation by different applications and transports, as will as handle different granularities of receiver feedback. In addition, certain applications can be greedy or malicious and attempt to either bypass the CM or try to fool it into allocating more bandwidth than is safe. We are developing security schemes that will prevent such applications from compromising the stability of the Internet. This is especially interesting because most current network security efforts focus on data access, not on whether data can be transmitted.

The desired end result is an integrated end-to-end congestion management architecture for a wide variety of Internet applications, including potentially buggy, greedy, or malicious ones. We believe that our approach to these problems is an important step in a stable Internet with adaptive applications, free from the danger of congestion collapse.

## Progress

We have completed a prototype design of the CM and have conducted simulations that demonstrate the benefits of our approach. Our results show that the CM API enables streaming applications and Web transfers to adapt well to congestion. A paper based on this work appeared in the 1999 ACM SIGCOMM conference.

A subset of the CM design, which involves no receiver kernel modifications, has recently been submitted as an Internet draft. Based in part on our work, an ECM (End-point congestion management) working group has been chartered in the Internet Engineering Task Force (IETF).

We have also implemented parts of the CM and expect to complete a version with applications soon.

## Future

In the near future, we expect to complete a prototype CM with sample applications including streaming, real-time, and Web transfers. We intend to deploy these and experiment with them on the Internet. We have started investigating the effect of receiver feedback granularity on performance and are developing techniques to identify the presence of differentially serviced paths between hosts, which will affect the granularity of sharing in the CM. We will also investigate new congestion control algorithms, especially for streaming and real-time applications.

(FALL, 1999)

### Contact

Hari Balakrishnan, hari@lcs.mit.edu
http://wind.lcs.mit.edu/

(FALL, 1999)

## WIND: Wireless Networks of Devices

*Leaders:*
Hari Balakrishnan, John Guttag

*Members:*
Anit Chakraborty, Jeremy Lilley,
Wendi Heinzelman, Alex Snoeren, William Adjie-Winoto

*Sponsors*:
IBM, Nippon Telegraph and Telephone

### Background

WIND is a system of middleware and protocols that enable self-configuring networks of devices, sensors and mobile computers and self-organizing network applications in such dynamic environments.

### Why Is This Interesting?

Networks of the future are certain to include appliances and devices in addition to general purpose computers. These devices are rich sources of information (e.g., remote cameras or sensors), can be used to control the physical environment (e.g., actuators), or can be used to monitor equipment over the network (e.g., photocopy machines).

Such environments display a degree of dynamism not usually seen in traditional wired networks due to mobility of nodes and services as well as rapid fluctuations in performance. There is usually no pre-configured support for configuring these heterogeneous, mobile networks, or for describing, locating, and gaining access to available services.

### Approach

We are designing an ad hoc address allocation protocol and topology formation algorithms that efficiently allow nodes to discover neighbors and form neighborhoods. We are designing energy-efficient protocols for disseminating information in networks of devices and sensors.

Our approach to solving the problem of resource discovery in dynamic, mobile environments is to design a new naming system, called the Intentional Naming System (INS). Because applications in these environments often do not know the best network location that satisfies their needs for information or functionality.

## Progress

We have designed and implemented most of INS and are focusing on developing non-trivial applications. Some of the internal INS components are being improved, including its routing protocol and scaling techniques.

## Future

The main work for the near future is to design, deploy and evaluate the self-configuration protocols for the resolvers and develop non-trivial location-based applications. We also expect to complete designing the ad hoc network configuration protocol and topology formation algorithms in the next several months.

## Contact

Hari Balakrishnan, hari@lcs.mit.edu
http://wind.lcs.mit.edu/

## World Wide Web Consortium: Uniquely Positioned to Lead the Evolution of the World Wide Web

Leading the World Wide Web's evolution means staying ahead of a significant wave of applications, services, and social changes. For W3C to effectively lead such dramatic growth at a time when a "Web Year" is equivalent to a few months, it must demonstrate exceptional agility, focus and diplomacy. To this end, the Consortium fulfills a unique combination of roles traditionally ascribed to quite different organizations.

Like its partner standards body, the Internet Engineering Task Force (IETF), W3C is committed to developing open, technically sound specifications backed by running sample code. Like other information technology consortia, W3C represents the power and authority of hundreds of developers, researchers, and users. Hosted by research organizations, the Consortium is able to leverage the most recent advances in information technology.

W3C works on technologies that impact people's daily commercial, cultural and personal activities. For several years W3C has recognized the social, legal, and public policy challenges associated with the development of the Web.

### Host Institutions

W3C was formally launched in October 1994 at the Massachusetts Institute of Technology's Laboratory for Computer Science (MIT LCS). Moving beyond the Americas, the Consortium established a European presence in partnership with France's National Institute for Research in Computer Science and Control (INRIA) in April 1995. As the Web's influence continued to broaden internationally, the resulting growth in W3C Membership created the need for an Asian host. In August 1996, Keio University in Japan became the Consortium's third host institution.

### Members

The Consortium's real strength lies in the broad technical expertise of its Membership. W3C currently has more than 300 commercial, non-profit, and academic Members worldwide, including hardware and software vendors, telecommunications companies, content providers, corporate users, citizen groups, and government and academic entities.

W3C provides a vendor-neutral forum for its Members to address Web-related issues. Working together with its team and the global Web community, the Consortium aims to produce free, interoperable specifications and sample code. Funding from Membership dues, public research funds, and external contracts underwrites these efforts.

The Consortium's Advisory Committee (AC) is composed of one official representative from each Member organization, who serves as the primary liaison between the organization and W3C. The Advisory Committee's role is to offer advice on the overall progress and direction of the Consortium.

### Team

W3C is led by Director Tim Berners-Lee, creator of the World Wide Web, and Chairman Jean-François Abramatic. With more than 30 years combined expertise in a wide array of computer-related fields, including real-time communications, graphics, and text and image processing, Berners-Lee and Abramatic are well prepared to lead the Consortium's efforts in spearheading the global evolution of the Web. The Consortium's technical staff include full-time and part-time employees, visiting engineers from Member organizations, consultants, and students from more than 13 countries worldwide. The W3C Team works with the Advisory Committee, the press, and the broader Web community to promote W3C's objectives.

### Recommendation Process

Specifications developed within the Consortium must be formally reviewed by the Membership. Consensus is reached after a specification has proceeded through the review stages of Working Draft, Proposed Recommendation, and Recommendation. As new issues arise from Members, resources are reallocated to new areas to ensure that W3C remains focused on topics most critical to the Web's interoperability and growth.

### Domains

Leading the evolution of a technology as dramatically in flux as the World Wide Web is a challenging task. W3C is a unique organization, well adapted to today's fast-paced environment. Its mission is to lead the Web to its full potential: as a scalable computer-to-computer system, as an efficient human-to-computer interface, and as an effective human-to-human communications medium. In order to achieve these goals, W3C's team of experts works with its Members to advance the state of the art in each of the four Domains: Architecture, User Interface, Technology and Society and The Web Accessibility Initiative. Each Domain is responsible for investigating and leading development in several Activity areas which are critical to the Web's global evolution and interoperability.

### Activities

When W3C decides to become involved in a new technology or policy, it seeks the approval of its Membership to initiate an Activity in that area. W3C Members are formally invited to review the proposed scope and charter, and subsequently to participate in the work of the Activity. Generally an activity is carried out by one or more groups which may be working groups, interest groups or coordination groups. An Activity may also be carried out by establishing a software project.

### Open Source Software

The natural complement to W3C specifications is running code. Implementation and testing are an essential part of specification development and releasing the code promotes exchange of ideas in the developer community. All W3C software is Open Source. W3C Open Source Software includes:

- Jigsaw, W3C's object-oriented web server, written in Java and supporting HTTP 1.1
- Amaya, W3C's testbed browser/editor
- A CSS Validator which allows the user to validate the CSS style sheets used by HTML and XML pages
- An HTML Validator which allows HTML documents to be validated against the DTDs for HTML, including for HTML 4.0
- Libwww, a general-purpose Web API written in C for Unix and Win32. It is a highly modular extensible API which can be used as the code base for writing Web clients, servers, and proxies.
- HTML Tidy, a free utility for correcting HTML syntax automatically and producing clean markup. Tidy can be used to convert existing HTML content into compliant XML.

The current list of Activities is as follows:

### Architecture Domain

HTTP
HTTP-NG
Jigsaw
TV and the Web
Web Characterization
XML

### User Interface Domain

Amaya
DOM
Graphics
HTML
Internationalization
Math
Mobile Access
Style Sheets
Synchronized Multimedia

### Technology and Society Domain

Digital Signatures
Metadata
Micropayments
Privacy

### Web Accessibility Initiative

WAI International Program Office
WAI Technical Activity
(FALL, 1999)

**Contact**

http://www.w3.org