1990
ANNUAL REPORT

International Computer Science Institute

February 1991

Contents

1 Introduction

The **International Computer Science Institute** is an independent, non–profit basic research institute located near the University of California campus in Berkeley, California. The ICSI (pronounced **EYE** c. s. **EYE**) was started in 1986 as a joint project of the Computer Science Division of UC Berkeley and the German National Research Center for Computer Science (GMD). German support of the Institute is now provided by a government-industry consortium. Switzerland joined the sponsorship of ICSI in 1989 and Italy joined in 1990. ICSI also receives support from a variety of U.S. sources and participation by other countries is anticipated. The German initiative was motivated by a recognition of the lack of an international facility for fundamental research in the field of computer science. The supporting industrial and government sponsors expect the participation of foreign scientists to do for fundamental computer science what such international collaboration has done for the field of physics. They also expect to benefit from the experience that young scientists from Germany, Switzerland, Italy and other countries will gain from participation in the work of the Institute and ultimately enhance international recognition of the work which goes on in their own countries. The selection of Berkeley for the Institute was based on the outstanding reputation of the UCB Computer Science Faculty and on their enthusiasm for the project.

The core of the Institute's program is the intramural research effort. The International Computer Science Institute strives to maintain ongoing basic research projects of the highest standard in selected areas of computer science and engineering. Only by maintaining research projects at the forefront of technology can ICSI have the strength to meet its other goals in international cooperation. The particular areas of concentration are chosen for their fundamental importance and their compatibility with the strengths of the Institute and UC Berkeley staff. The current emphasis is on distributed and parallel computation with particular attention to massive parallelism.

In addition to its intramural research programs, ICSI maintains a number of other programs in support of international cooperation in advanced computer science and engineering. These include a post–doctoral program, exchange visits, summer jobs for American graduate students in the sponsor countries and partial support of selected working conferences. One of the principal advantages of ICSI is its ability to act rapidly and flexibly when opportunities arise and thus gain maximum leverage from its resources. As a basic research organization, ICSI does not engage in product development. The Institute does help in the practical realization of its research results and grants non–exclusive, royalty–free licenses to supporting institutions.

All of the ICSI programs place special attention on cooperation with sponsor nations. The post–doctoral program is fairly conventional. Committees of sponsor-nation and ICSI scientists select among applicants on the basis of overall scientific potential and the availability of an appropriate project at ICSI or UCB. Scholarships are competitive and all expenses are paid by ICSI. Exchange visits by more senior investigators can be of any length and are arranged at the mutual convenience of the host and visitor. These are often supported by other funds, but ICSI can provide full support where this is warranted. Several of these visits have given rise to new or strengthened international collaborations.

The summer program for graduate students has concentrated on placing advanced United States students in sponsor–nation laboratories. ICSI advertises in the U.S. for interested candidates, who are required to be beyond the qualifying exam and are asked to provide references. In parallel with this, sponsor–nation committees identify laboratories which might be interested in U.S. students. The ICSI staff reviews the applicants and makes tentative matches between openings and applicants. The appropriate files are forwarded to host laboratories and all further negotiations are between the employing lab and the student applicant. The salary is paid by the employer. ICSI pays for the travel costs to the host laboratory and offers assistance with arrangements. The idea, again, is that a small expenditure of Institute funds and effort can have a significant effect.

The ICSI intramural research program is designed to concentrate international and multi–faceted research teams on computing problems of the greatest scientific and practical potential. The focus is on the middle range future, five to twenty years out. The narrow gap between theory and practice in computer science makes it an exceptionally interesting field, but could create conflict–of–interest problems for an international research institute. By concentrating on pre–competitive but result–oriented projects, ICSI can contribute to the technology base as well as the general scientific knowledge of its sponsors.

The initial research plan of ICSI centered on distributed and parallel computation with a special concern for massively parallel systems. It has been clear for some years that parallel computation is the key to many desirable applications. Distributed and parallel systems of moderate scale are now fairly well understood and are widely employed. But the problems involved with millions or billions of independent computations appear to be different and to require new techniques. ICSI is currently addressing issues in four key areas: theory of computation, realization of massively parallel systems, applications of such systems, and very large distributed networks.

The year 1990 saw the fruition of several efforts ICSI began in past years. The RAP computer came into daily use and has been made available to ICSI sponsors. The Sather programming language is now being employed in many Institute projects. A preliminary announcement of its availability drew interest from several hundred installations. The ICSIM connectionist simulator has seen pilot use and is being converted to Sather. The Networks and Distributed Systems Group started distributing the REAL network simulator, which allows the realistic simulation of packet switching networks. REAL is already in use in a large and growing number of sites distributed over three continents.

There were three international working conferences organized at ICSI in 1990. A June workshop on the early (front-end) components of speech recognition systems is summarized in TR-90-045. A workshop on temporal and real-time specification was held in August and is described in TR-90-060. The first international workshop on network operating system support for digital audio and video was held at ICSI in November. It is summarized in TR-90-062. The Institute continues to organize and sponsor working meetings of interest to its scientists and sponsors.

Section 2 of this paper outlines our efforts and plans in the area of applications of massively parallel systems. The most promising applications are to problems in artificial intelligence including vision, language and speech, and knowledge representation and inference. Section 3 discusses the Institute's research in the theory of computation and its relation with other topics. The realization of massively parallel systems in hardware and in software is the subject of Section 4. Section 5 outlines our work on the supporting systems theory and software for the very large international networks of the near future.

2 Applications of Massive Parallelism

The applications group continues to focus on Artificial Intelligence tasks in the areas of vision, speech, learning, knowledge representation and inference. There is close collaboration with the ICSI realization group; some joint projects are covered in this section of this report and some in Section 4. This year has seen more than the usual amount of effort on tool building and this has led to several transferable systems.

2.1 Computer Vision

We made several advances in the vision research effort during 1990. Subutai Ahmad has been studying biologically plausible neural architectures which implement aspects of visual attention. An architecture which solves a touchstone problem requiring visual attention is the subject of [Ahmad, Omohundro, 1990a]. The general aspects of this kind of network are explored in [Ahmad, Omohundro, 1990b].

There is a large class of vision tasks that cannot be solved efficiently in parallel by strictly feed–forward networks. A model of visual attention is one way to solve some of these problems, by retaining as much of the inherent parallelism as possible. We have developed a connectionist mechanism for visual attention. It has been used to solve the equilateral triangle problem: given three clumps of points, determine whether they form an equilateral triangle [Ahmad, Omohundro, 1990a, Ahmad, Omohundro, 1990b]. The complete network integrates an efficient focus of attention mechanism and a cluster detection scheme to dynamically extract the locations of the vertices.

The attentional mechanism has also been used as part of a network for performing visual search. Top-down information about the target object and bottom-up information about the current image are used to prioritize possible target locations in parallel. The network then visits these locations in order of priority until the target object is detected. The combination of top-down and bottom-up information allows the network to locate the target object quickly, often in one step. The network models human reaction times and makes predictions about how people might perform the task. Finally, we have also studied how our attentional mechanism might be learned.

Work has also continued on general connectionist models of vision and space. In a collaborative effort with the University of Rochester, we ran a number of psychophysical experiments on predictions of previous models. We found the first positive evidence for low-level integration of visual information over saccades. Another collaboration, with Nigel Goddard of Carnegie-Mellon, is developing an ambitious connectionist model for recognizing animal motions from moving light displays. There are now efforts in several other labs pursuing these ideas.

References

[Ahmad, Omohundro, 1990a] Ahmad, S., and Omohundro, S. (1990). "Equilateral Triangles: A Challenge for Connectionist Vision," in *Proceedings of the 12th Annual Meeting of the Cognitive Science Society*, MIT.

[Ahmad, Omohundro, 1990b] Ahmad, S., and Omohundro, S. (1990). "A Network for Extracting the Locations of Point Clusters Using Selective Attention." Technical Report TR-90-011, International Computer Science Institute, Berkeley, CA.

[Feldman, 1990a] Feldman, J. A. (1990). "Apparent Motion and Other Mysteries," *Parallel Processing In Neural Systems and Computers*, R. Eckmiller, G. Hartmann, G. Hauske (eds.), Elsevier Science Publishers B.V., pp. 479-486.

[Feldman, 1990b] Feldman, J.A. (1990). "Neural Networks, Artificial Intelligence and Computational Reality," Memorial Edition for Joseph Hatvany, in *Computers in Industry* 14:145-148, Elsevier Science Publishers.

[Feldman et al., 1989-90] Feldman, J.A. (with others) (1989-90). "Connectionist Systems," *Annual Review of Computer Science* 4:369-383.

[Hayhoe, Feldman et al., 1989] Hayhoe, M., Feldman, J.A., and Lachter, J., (1989) "Integration of Form Across Saccadic Eye Movements," May 1990, *Perception*, in Press.

2.2 Geometric Learning

We made significant progress in the effort to develop new algorithms for "geometric learning." These provide extremely efficient implementations of many of the geometric representation and induction tasks required in vision and robotics. Some general strategies and approaches were layed out in [Omohundro, 1990a]. Stephen Omohundro presented many of these algorithms in a 5–lecture advanced course on "Learning and Recognition" at the VII Escola de Computaçao in São Paulo, Brazil in July 1990.

A formal analysis of an approach with good theoretical properties is presented in [Omohundro, 1990b]. Rusell Pflughaupt is implementing this approach under the joint supervision of Carlo Sequin and Stephen Omohundro to study its practical use in a task of recognizing engine failure symptoms from sensor outputs.

A powerful new data structure called the "bumptree" was introduced. It generalizes several hierarchical geometric data structures including oct-trees, k-d trees, balltrees, and boxtrees. Bumptrees were introduced in [Omohundro, 1991] and were shown to significantly outperform the popular approach of radial basis functions in a typical robot arm learning task. While the learning time for radial basis functions increases as the cube of the samples, the balltree construction approach empirically has a learning time which is only slightly longer than linear in the number of samples. The retrieval time for radial basis functions is linear in the number of samples, while the bumptree approach empirically is almost constant as the number of samples increases. Radial basis functions required half an hour to construct a representation with 300 elements in the basis, while the bumptree construction time for this size sample was only 4 seconds. Retrieval time for radial basis functions increases linearly from 0 up to .03 seconds for 300 elements in the basis, while bumptrees require about .02 seconds all the way up to 10,000 samples. Applications to density estimation, constraint learning and retrieval, function learning, and classification were also outlined and are currently under study.

Bartlett Mel and Stephen Omohundro presented "Coarse Coding, Receptive Fields, Learning, and Biology" at the April, 1990 Neural Networks for Computing meeting at Snowbird, Utah. This analyzes the theoretical constraints on receptive field size and shape for biological systems. An extended analysis with a new dynamic algorithm for placing receptive fields was presented in [Mel, Omohundro, 1991].

References

[Mel, Omohundro, 1991] Mel, B. and Omohundro, S. (1991). "How Receptive Field Parameters

Affect Neural Learning," Technical Report TR-91-010, International Computer Science Institute, Berkeley, CA; also in *Proceedings of the 1990 IEEE Conference on Neural Information Processing Systems – Natural and Synthetic*, in Press.

[Omohundro, 1990a] Omohundro, S. (1990). "Geometric Learning Algorithms," in *Physica D* 42:307-321; also in Forrest, S. (ed.), *Emergent Computation*, North–Holland, 1990.

[Omohundro, 1990b] Omohundro, S. (1990). "The Delaunay Triangulation and Function Learning," Technical Report TR-90-001, International Computer Science Institute, Berkeley, CA.

[Omohundro, 1991] Omohundro, S. (1991). "Bumptrees for Efficient Function, Constraint, and Classification Learning," Technical Report TR-91-009, International Computer Science Institute, Berkeley, CA; also in *Proceedings of the 1990 IEEE Conference on Neural Information Processing Systems – Natural and Synthetic*, in Press.

2.3 The Sather Language and Libraries

During 1990 we designed and implemented a new object–oriented computer language called Sather. Sather was designed by Stephen Omohundro and was heavily influenced by the language Eiffel designed by Bertrand Meyer. Sather focuses more on efficiency and less on some of the formal and theoretical issues addressed by Eiffel. Doing our own implementation has allowed us to maintain our own quality control and requirements for efficiency and allows us to further develop a parallel version of the language. Our aim has been to keep the language small but to develop an extensive class library. We would like to establish a repository for efficient, reusable, well written, publicly available classes for most of the important algorithms in computer science. There are currently about 120 classes in the library covering basic algorithms and data structures, numerical tasks, geometric tasks, connectionist architectures, graphics, statistics and other areas. We recently made a preliminary announcement of the language and libraries and over 350 sites have expressed interest. Many outside institutions have offered to contribute class libraries in their areas of expertise.

Sather code is compiled into portable C and efficiently links with existing C code. The Sather compiler was written in Sather by Chu-Cheow Lim. It has been operational for several months, though it is still being improved. Preliminary benchmarks show a performance improvement over Eiffel of between a factor of 4 and 50 on basic dispatching and function calls. On the Stanford Self benchmarks (including 8 queens, towers of Hanoi, bubblesort, etc), Sather is slightly faster than C++, though this is probably due to the C compiler's better ability to optimize for a Sparcstation than the C++ compiler.

Jeff Bilmes is working on a Sather debugger based on GDB from the Free Software Foundation. Heinz Schmidt has been developing user interface classes for the Open Look XView toolkit that runs under X windows. Stephen Omohundro wrote a Sather editing mode for GNU Emacs which Heinz Schmidt and others are extending to an entire programming environment. Class development is proceeding in many different areas. We expect to make a Beta test version of the compiler and classes available sometime in the next several months. An effort to design a parallel version for shared memory machines has been led by Jerry Feldman and the implementation effort for this version is beginning.

2.4 Learning and Knowledge Representation

Work during 1990 was mostly connected with a research project in natural language acquisition (informally called L_0) involving several ICSI and UCB researchers and students (Jerome Feldman, George Lakoff, Susan Weber, Terry Regier, Andreas Stolcke).

One line of approach to L_0 is concerned with the application of standard PDP connectionist network architectures to the task of mapping surface language to semantics. Within this framework, a series of studies with simple recurrent networks (SRNs) was carried out, investigating both the possibilities and the limitations of these architectures. Results showed that up to a certain degree of syntactic complexity, SRNs can act as trainable and robust syntax-to-semantics transducers. However, for truly context-free (recursively embedded) structures, both at the syntactic and the semantic level, more powerful representations and learning algorithms have to be found.

A second line of research in L_0 involved the design and implementation of a testbed system for experimentation with natural language grammars, representational schemes and interfaces between these two. The scope of the learning task can be narrowed to certain designated sub-problems with an appropriate system infrastructure. The target architecture for an eventual learning system was developed and implemented as a testbed in which to develop learning components. A structured connectionist implementation of the semantic component of the system is currently under development. Such a system was implemented in Prolog, resulting in valuable insights into the problems of the L_0 domain, as well as a flexible software system for ongoing experimentation.

The first task of 1990 was the design and implementation of a system to learn the lexical semantics of closed-class terms denoting spatial relations. Given scenes containing two objects, and an indication of which is the landmark and which the trajector (i.e., the reference object and the object located relative to it, respectively), the system is trained to name the spatial relations depicted. This effort is a part of the L_0 miniature language acquisition project, providing perceptual grounding for the semantics required in the L_0 project. The current version of the system learns such terms for the case of a trajector consisting of a single point.

The second task was the extension of the basic design to allow the system to learn in the absence of explicit negative evidence. The motivation behind this was the desire to provide an account of how children, who do not receive overt negative evidence, might learn such spatial concepts without grossly overgeneralizing. The approach of taking a positive instance for one concept to be an implicit negative instance for all others was used, with limited initial success. Problems arose because of concepts whose extensions overlap, e.g. "above" and "outside". This issue was addressed by attenuating the error caused by implicit negative instances. Current work is partially directed at determining whether it is possible, while learning the concepts themselves, to also learn which concepts provide reliable negative evidence for which others.

Recent work is concerned with neurally inspired, but more structured and informed algorithms for learning syntactic structures from purely distributional surface language data. For this purpose a formal generalization of phrase-structure rules to continuous metric spaces has been devised, which can be subjected to adaptive learning methods (such as back-propagation or competitive learning). Preliminary experimental studies have shown that the learning algorithm can extract rules that approximate context-free grammar representations from exposure to positive and negative instances from a sample language.

Another project proposes a structured connectionist approach to learning a lexical representation of nouns with a context dependent semantics. In the model developed, the (micro-)features comprising the noun's semantics in a given context, once established by metaphoric interpretation, are incorporated into *aspects* of the lexeme's literal semantics by one-shot learning. This reversal of the traditional two-stage model of metaphor interpretation proposes figurative meaning as the basis for acquiring literal semantics. Recent efforts include adding a simple recruitment learning mechanism for incorporating new word senses.

Another project in representation and inference is concerned with connectionist realization of formal theorem proving techniques. The central point was the implementation of a distributed unification algorithm as the core of a connectionist inference system, developed previously by Steffen Hölldobler [Hölldobler, 1990]. The implementation is based on the object-oriented net simulator ICSIM, which is implemented in Eiffel [Schmidt, 1990]. (See Section 4.5) Whereas the implementation showed the feasibility of unification with connectionist implementation techniques, the performance of this first prototype is not very impressive. This is mainly due to the fact that the number of units required by the algorithm is quadratic with respect to the size of the formula, leading to a huge number of connections between units. Most of these units and connections, however, are never really used: the algorithm constructs all possible units and connections for a unification problem of a particular size, independent of the actual structure of the terms involved. There are a number of improvements under consideration: a) design of a distributed algorithm with logarithmic or linear space requirements; b) an improved implementation to neglect units which cannot contribute to the solution of a particular unification problem; c) the introduction of virtual units in ICSIM. In addition, the design of the overall inference system is under review; this involves implementation-oriented issues as well as possible extensions of its functionality.

References

[Feldman et al., 1990a] Feldman, J.A., Lakoff, G., Stolcke, A., and Weber, S.H. (1990). "Miniature language acquisition: A touchstone for cognitive science." Technical Report TR-90-009, International Computer Science Institute, Berkeley, CA; also appeared in the *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pp. 686–693.

[Feldman et al., 1990b] Feldman, J.A., Weber, S.H., and Stolcke, A. (1990). "A testbed for the miniature language L_0." *Proceedings of the 5th Rocky Mountain Conference on Artificial Intelligence*, pp. 25–30, New Mexico State University, Las Cruces, NM.

[Hölldobler, 1990] Hölldobler, S. (1990). "CHCL — A connectionist inference system for Horn logic based on the connection method and using limited resources." Technical Report TR-90-042, International Computer Science Institute, Berkeley, CA.

[Schmidt, 1990] Schmidt, H. (1990). "ICSIM: Initial design of an object–oriented net simulator." Technical Report TR-90-055, International Computer Science Institute, Berkeley, CA.

[Stolcke, 1990] Stolcke, A. (1990). "Learning feature-based semantics with simple recurrent networks." Technical Report TR-90-015, International Computer Science Institute, Berkeley, Ca.

[Stolcke, 1991] Stolcke, A. (1991). "Vector space grammars and the acquisition of syntactic categories." *Working notes of the AAAI Spring Symposium on Connectionist Natural Language Processing*, Stanford University, Stanford, Ca. To appear.

[Weber 1990a] Weber, S.H. (1990). "Acquiring categorical aspects: a connectionist account of figurative noun semantics." *Proceedings of the 5th Rocky Mountain Conference on Artificial Intelligence*, pp. 295–300, New Mexico State University, Las Cruces, NM.

[Weber 1990b] Weber, S.H. (1990). "Modelling semantic flexibility with a structured connectionist implementation of functional category organization." *Proceedings of the 8th Canadian Conference on Artificial Intelligence, CSCSI-90*, pp. 128–133, University of Ottawa, Ottawa, Ontario, Canada.

[Weber, Stolcke, 1990] Weber, S.H. and Stolcke, A. (1990). "L_0: A testbed for miniature language acquisition." Technical Report TR-90-010, International Computer Science Institute, Berkeley, CA.

3 Theory of Computation

3.1 Introduction

During 1990 the research of the Theory of Computation group emphasized the following topics: computational complexity, analysis of algorithms, randomized computation, on-line algorithms, computational learning theory, theory of neural networks, and models of parallel computation. Although the topics are diverse, a number of common threads run through the work. All of us are concerned with classifying problems according to their computational complexity and determining the influence of the model of computation on the difficulty of solving a problem. The study of parallel computation is a central theme. Another recurrent theme is the study of computational systems that organize themselves through a process of learning from examples, rather than being programmed to execute specific algorithms. Finally, much of our research focuses on the power of randomization to simplify algorithms and reduce their time and space requirements.

3.2 Computational Complexity

3.2.1 Breaking the Complexity Barrier with Waffling Programs

Coin-flipping is one way to break through the complexity barrier. It enables one to compute certain functions faster probabilistically than is otherwise possible deterministically. The speedup obtained through the use of randomness, ie. coin-flipping, can be by as much as an exponential for certain functions (and no more than an exponential for any function). Of course, coin-flipping has a cost: the cost of generating the needed randomness, the cost to the user of the uncertainty, etc.

Parallelism is another way to get speedup. It has the potential to speed up some but not necessarily all computations by a factor equal to the number of processors. This too has a cost: the cost of the extra processors, the cost of the added programming complexity, etc.

Is there any other way to get speedup? If so, how much? And at what cost? This subsection discusses a new idea for getting speedup called *waffling.*

A waffling program, intuitively, is one that is permitted to change its mind about the answer, ie. to waffle, a finite number of times. The computation time of a waffling program, called its *waffle time*, is the time that the program takes to converge (on its final answer). The waffle time of a function can be defined in the usual way in terms of the waffle times of its programs. The waffle time is a fundamental parameter of any algorithm or function. It never exceeds and is often less than the ordinary computation time of that algorithm or function. In a great number of cases, waffling enables one to obtain an answer from an algorithm well before it halts in the ordinary sense.

An ordinary program may be able to guess (ie. output) what its final answer is likely to be long before it can certify that answer as correct.

Optimization programs provide good examples: A traveling salesperson program seeking an optimal tour in a weighted graph may find such a tour early in its computation but not be able to certify it as optimal until it has investigated all other tours. The increasingly better tours that an optimization program finds in its search for the optimal tour, and especially the optimal tour itself, may be useful even before the optimal tour can be certified. For this reason, a good suggestion is that such a program should be designed to output its current best guess at the optimal tour when it finds it, and then to change its mind – waffle – whenever it finds a better one.

Probabilistic programs provide additional excellent examples of waffling programs that should output their answer before they can certify it: A probabilistic prime-testing program has a good likelihood to decide correctly whether or not its input is prime after completing its first pass through a basic [Solovay-Strassen, Miller-Rabin] monte-carlo routine. Typically, such a program is given both an input integer to be tested and an integer parameter k that specifies the maximum allowable probability of error to be 2^{-k}. The program then does k passes through its basic monte-carlo routine before announcing its answer. The correct answer, however, is almost certainly known by the program after the first one or two passes. Ron Rivest reported that in a study of a thousand or so 500-decimal-digit numbers, the Miller-Rabin primality test gave the correct answer after just one pass for each and every number tested.

In [B] Manuel Blum gives a model of computation, the so-called waffling machine, that can output its best guess for the correct answer and then change its mind (waffle) as it goes along. A waffling machine is required to converge eventually on its answer (never again to change its mind), and to certify the correctness of its final answer by halting. However, a waffling machine may be able to converge on (i.e. guess) the correct answer long before it can halt, and it is this time to converge (to output its final guess) rather than the time to halt that is taken to be the running time (the so-called waffle time) of a waffling machine.

A waffling machine is defined in terms of multi-tape Turing machines. The time it takes a waffling machine to converge is always bounded above by the time it takes it to halt, which in turn is never greater than the time it takes an ordinary Turing machine to compute the answer.

Waffling machines do exist and are used in practice though there appears to be no theory of them. For example, chess-playing machines are generally designed to output the best move they have found so far in their search of the game tree. Since the game tree for chess is finite (assuming appropriate rules of chess), such machines may be viewed as looking for the best move, and certifying it once the entire game tree has been searched. These machines do waffle, displaying their (possibly changing) current best move on their screen, and their output guesses are useful even though in practice they never get certified. In general, game-playing machines for nearly all complex games (this excludes tic-tac-toe) are good examples of waffling machines.

Blum defines the concept of a waffling machine, gives examples, and finally proves the existence of a waffling 2-work-tape Turing machine that converges (makes its final guess) in a time that is at least a square root faster than any non-waffling (ordinary) 2-work-tape Turing machine. The particular problem that this waffling machine solves is totally uninteresting, except that it suggests where to look for more interesting problems. Definitions and results like those given here for Turing machines can be obtained for waffling machines that are defined in terms of any other reasonable model of computation, such as random access machines.

References

[B] Blum, M. (1991). "Breaking the Complexity Barrier with Waffling Programs," to appear as an ICSI Technical Report.

3.2.2 Theory of Computation Over the Reals

Work has continued in the area of computation over the reals. A major project undertaken by Lenore Blum, Michael Shub (visiting ICSI this year from IBM Yorktown) and Steve Smale (of UC Berkeley) is the writing of a book to lay down the foundations of this theory as well as delve deeply into important areas of complexity-based numerical analysis such as: numerical linear algebra, linear programming, equation solving and Newton's Method.

Ongoing directions of research include:

1. Pursuing interconnections between computation/complexity theory and dynamical systems.

The Blum-Shub-Smale (BSS) theory enables one to formulate (and sometimes resolve) questions of decidability of sets that arise naturally in the theory of complex analytic dynamical systems. Since the classical theory of computation only deals with countable sets, such questions could not be posed in any natural sense previously. For example, Roger Penrose [P] acknowledges the difficulty of formulating his question: Is the Mandelbrot set decidable? This question is naturally formulated in the BSS theory and, using recent work in complex dynamical systems theory, we are able to answer this question "no." (See [Bl] and [BS]).

In this setting it is also natural to apply ideas of degree theory of classical recursive function theory to classify the complexity (i.e. the relative undecidability) of sets over the reals or complex numbers. Roughly we say a set A is decidable relative to set B if a machine with an additional node for deciding B (i.e. an "oracle" for B) can be used to decide A. The question then is: what is the resulting hierarchy? Classically, it was an open problem for a number of years (Post's problem) to find two semi-decidable sets of integers that were incomparable with respect to relative decidability. In earlier work we have shown that most Julia sets are undecidable while their complements are semi-decidable (see also [Bl]) and posed a Post type problem. Using methods from dynamical systems, [C] has shown that the situation here is quite the opposite: indeed most Julia sets of quadratic maps are incomparable. Thus we ask: Are there two comparable undecidable Julia sets? Alternatively, is there a natural way to increase the power of machines so that the resulting hierarchy is meaningful?

In the opposite direction, we have exploited the analogy between computing machines and dynamical systems for a new proof of Godel's Theorem [Bl2]. Here the computing endomorphism of a machine over R is our key conceptual and technical tool.

2. Increasing our understanding of NP-Completeness

Here we wish to better understand our theory of NP-completeness over the reals as well as relate it to the classical theory. One of the reasons the classical theory is so compelling is the large number of seemingly unrelated problems that are NP-complete (and thus mutually reducible). Over the reals, until recently, essentially our only known NP-complete problem was the feasibilty problem for real algebraic varieties (the effective Hilbert Nullstellensatz is the corresponding NP-complete problem over the complex numbers). A number of new NP-complete problems have been discovered. For example, [CR] have shown that the following problems are co-NP complete over the reals: the problem of whether a semi-algebraic system (of polynomial equalities and inequalities) of degree d ($d \geq 1$) is convex (i.e. defines a convex set), of whether a semi-algebraic system of degree d ($d \geq 1$) has at most k points (every k), of whether a polynomial of degree d ($d \geq 3$) is always positive.

One approach to understanding NP-completeness over the reals or an arbitrary ring in the BSS model of computation is to consider variants. Here one might vary the measures of size (logarithmic/bit or dimension), of cost (logarithmic/bit or algebraic), the allowed computations (e.g. linear, polynomial or rational maps) and branching conditions (branching on $=$ or $\leq$). Some progress has been made here [see S] using the main idea (where applicable) that if a machine is in P then the polynomials it computes must have degrees bounded by a polynomial in the size of the problem instance.

References

[Bl] Blum, L. (1990). "Lectures on a Theory of Computation and Complexity over the Reals (or an Arbitrary Ring)," in Lectures in the Sciences of Complexity, Addison Wesley, pp. 1–47.

[BS] Blum, L., and Smale, S. (1990) "The Gödel Incompleteness Theorem and Decidability over a Ring," *Proceedings of the Smalefest*, Berkeley, CA; also Technical Report TR-90-036, International Computer Science Institute, Berkeley, CA.

[Bl2] Blum, L. (1990). "A Theory of Computation and Complexity over the Real Numbers," *Proceedings of the International Congress of Mathematicians*, Kyoto, Japan; also Technical Report TR-90-058, International Computer Science Institute, Berkeley, CA.

[C] Chong, T.C. (1990). "Reducibilities of Julia Sets in Complex Analytic Dynamics," preprint.

[CR] Cucker, F. and Rossello, F. (1990). "On the Complexity of some Problems for the Blum, Shub and Smale Model," preprint.

[P] Penrose, R. (1989). The Emperor's New Mind, Oxford University Press.

[S] Shub, M. (1990). "Some Remarks on Bezout's Theorem and Complexity Theory," *Proceedings of the Smalefest*, Berkeley, CA.

[S2] Shub, M. (1990). "The work of Steve Smale on the Theory of Computation," *Proceedings of the Smalefest*, Berkeley, CA.

3.2.3 Algebraic Complexity Theory

Peter Bürgisser, Marek Karpinski and Thomas Lickteig showed that certain computational problems in linear algebra are as hard as matrix multiplication. They define the complexity of a computational problem using the model of computation trees together with the Ostrowski complexity measure. Natural examples from linear algebra are:

- KER_n: Compute a basis of the kernel for a given $n \times n$ matrix;
- OGB_n: Find an invertible matrix that transforms a given symmetric $n \times n$ matrix to diagonal form;
- SPR_n: Find a sparse representation of a given $n \times n$ matrix.

To such a sequence of problems they assign an exponent similar to the one for matrix multiplication. They show that the exponents for all these problems are the same as the exponent for matrix multiplication.

References

[BKL] Bürgisser, P., Karpinski, M., and Lickteig, T. (1991). "Some computational problems in linear algebra as hard as matrix multiplication," Technical Report TR-91-005, International Computer Science Institute, Berkeley, CA. Submitted to *Computational Complexity*, Birkhäuser.

[BL] Bürgisser, P., and Lickteig, T., "Additive complexity of some computational problems in linear algebra," in preparation.

[BL2] Bürgisser, P. and Lickteig, T. "On additive complexity of bilinear problems," in preparation.

[BL3] Bürgisser, P. and Lickteig, T. " On verification complexity in higher codimensions," to be submitted to *Algebra and Computation*.

[BLS] Bürgisser, P., Lickteig, T. and Shub, M., "Test complexity of generic polynomials," in preparation.

3.2.4 Self-Testing/Correcting Programs

Suppose someone gives us an extremely fast program P that we can call as a black box to compute a function f. Should we trust that P works correctly? A *self-testing/correcting pair* for f allows us to: (1) estimate the probability that $P(x) \neq f(x)$ when x is randomly chosen; (2) on ***any*** input x, compute $f(x)$ correctly as long as P is not too faulty on average. Furthermore, both (1) and (2) take only slightly more time than the original running time of P.

In [BLR] we present general techniques for constructing simple to program self-testing/correcting pairs for a variety of numerical functions, including integer multiplication, modular multiplication, matrix multiplication, inverting matrices, computing the determinant of a matrix, computing the rank of a matrix, integer division, modular exponentiation and polynomial multiplication. In [CL] we apply similar techniques to construct a self-testing/correcting pair for the problem of computing the sin and cosin functions.

References

[BLR] Blum, M., Luby, M., and Rubinfeld, R. (1990). "Self-Testing/Correcting with Applications to Numerical Problems," STOC 90 (May 1990), pp. 73-83.

[CL] Luby, M., and Cleve, R. (1990). "A Note on Self-Testing/Correcting Methods for Trigonometric Functions, " Technical Report TR-90-032, International Computer Science Institute, Berkeley, CA.

3.3 Analysis of Algorithms

3.3.1 Probabilistic Recurrence Relations

The paper [K] is concerned with recurrence relations that arise frequently in the analysis of divide-and-conquer algorithms. In order to solve a problem instance of size x, such an algorithm invests an amount of work $a(x)$ to break the problem into subproblems of sizes $h_1(x), h_2(x), \ldots, h_k(x)$, and then proceeds to solve the subproblems. Our particular interest is in the case where the sizes $h_i(x)$ are random variables; this may occur either because of randomization within the algorithm or because the instances to be solved are assumed to be drawn from a probability distribution. When the h_i are random variables the running time of the algorithm on instances of size x is also a random variable $T(x)$. We give a convenient general method for obtaining a fairly tight bound on the upper tail of the probability distribution of $T(x)$. The proof of the bound requires an interesting analysis of optimal strategies in certain gambling games. The method has a wide range of applications, and it is our hope that it will take its place alongside Chernoff bounds and martingale tail inequalities as a useful tool for the analysis of algorithms.

References

[K] Karp, R. (1991). "Probabilistic recurrence relations," to appear in *Proc. ACM Symp. on Theory of Computation.*

3.3.2 Automatic Worst-Case Analysis of Parallel Programs

This work [Z2] is an extension of previous work on the automatic complexity analysis of functional programs done at the University of Karlsruhe [Z]. Parallel algorithms are expressed in first order parallel functional language which allows the definition of indexed data types and parallel execution of indexed terms. The machine model is a parallel reduction system based on eager evaluation. It is shown how the basic design principles of parallel programs can be analyzed automatically. The programs are designed for EREW-PRAMS. However, the analysis technique for programs having concurrent reads is the same.

The design principles covered are the ***balanced binary tree method***, the ***divide-and-conquer technique***, the ***compression technique***, and the ***pointer jumping technique***. In a functional language, the compression technique and the balanced binary tree method lead often to the same program.

The analysis technique is based on the method of recurrences. It first derives a system of recurrences, describing the worst case complexity of the program, and then solves this system of recurrences. The analysis result includes constant factors. It is therefore possible to compare parallel programs, and compare them also with the corresponding sequential algorithms. The analysis techniques are applied to some basic algorithms for the EREW-PRAM such as list-ranking, computing prefix sums, polynomial evaluation etc. In future, the method is applied to more difficult algorithms (e.g. graph algorithms).

References

[Z] Zimmermann, W. (1990). "Automatische Komplexitätsanalyse funktionaler Programme," *Informatik-Fachberichte 261*, Springer-Verlag.

[Z2] Zimmermann, W., (1990). "Automatic worst case complexity analysis of parallel programs." Technical Report TR-90-066, International Computer Science Institute, Berkeley, CA.

3.3.3 Average Case Analysis of Parallel Algorithms

Most complexity results of parallel algorithms consider the worst case complexity of algorithms. However little work is done in the average case complexity. Geppino Pucci and Wolf Zimmermann started their work by considering the sublist computation based on pointer jumping [PZ]. In its worst case the standard algorithm for this problem needs $\log_2 n$ iterations. They show that on a CRCW-PRAM the algorithm needs approximately $\log_2 \log_2 n$ iterations. It has therefore a significantly better average perfomance than in the worst case. It should be mentioned that the algorithm can be performed on a EREW-PRAM with precisely one CRCW-register. The study will be continued on more complicated algorithms (such as for example algorithms for finding the connected components of a graph).

References

[PZ] Pucci, G., and Zimmermann, W. "On the average case complexity of parallel sublist computation," submitted to WADS 91.

3.4 Randomized Computation

3.4.1 Randomized Algorithms for Approximate Enumeration

Michael Luby and Marek Karpinski have developed a polynomial time Monte-Carlo algorithm for estimating the number of solutions to a multivariate polynomial over *GF[2]* [KL]. This work gives the first efficient method for estimating the number of points on algebraic varieties over *GF[2]*, which has been recently proven to be *#P*-complete even for cubic polynomials. There are a variety of applications of our result.

Recently, Michael Luby, Marek Karpinski, Barbara Lhotzky, and David Zuckerman (a graduate student at U.C. Berkeley) have extended these results by developing algorithms to approximate the number of solutions to a multivariate polynomial over *GF[q]* for an arbitrary value of q. The work to date suggests that there is an algorithm for approximating the number solutions to multilinear multivariate polynomials with running time polynomial in q and the number of terms. We are trying to develop an algorithm that removes the multilinear restriction and thus works for any multivariate polynomial.

Luby and Boban Veličković from the *Mathematical Sciences Research Institute* and also the *Department of Mathematics, U.C. Berkeley* have made substantial progress on the problem of approximating the number of truth assignments that satisfy a formula in disjunctive normal form (a *DNF* formula) [LV]. Previously, it was known that the problem of computing exactly the number of satisfying truth assignments is *#P*-complete, and that there is a probabilistic approximation algorithm that runs in polynomial time. Our work improves dramatically over previous results, showing that there is a simple *deterministic* approximation algorithm with running time that is "almost" polynomial in the input size of the formula. However, we leave open the beautiful question of a whether or not there is a deterministic approximation algorithm with running time truly polynomial in the formula size.

References

[KL] Karpinski, M. and Luby, M. (1990). "Approximating the Number of Solutions of a *GF[2]*Polynomial," *Proc. ACM-SIAM Symp. on Discrete Algorithms.*

[LV] Luby, M. and Veličković, B., (1991). "On Deterministic Approximation of DNF," to appear in *Proc. ACM Symp. on Theory of Computing.*

[KKLLL] Karmarkar, N., Karp, R., Lipton, R., Lovász, L., and Luby, M., (1990). "A Monte-Carlo Algorithm for Estimating the Permanent," Technical Report TR-90-063, International Computer Science Institute, Berkeley, CA.

[K2] Karp, R., "An introduction to randomized algorithms," to appear in *Discrete Applied Mathematics.*

3.4.2 Pseudorandom Generators

We have shown how to construct a pseudo-random number generator from any one–way function [HILL]. This final version of the paper improves substantially on the original version by Impagliazzo, Levin and Luby which was presented at STOC 90.

References

[HILL] Håstad, J., Impagliazzo, R., Levin, L. and Luby, M., "Pseudo-random number generators from one-way functions," to appear.

3.5 On-Line Algorithms

An *on-line algorithm* is one that is presented with its input incrementally and must produce part of the output with limited information about the input. The question is then what kind of solution quality can be obtained. To answer this question, one compares the performance of an on-line algorithm with that of the optimal algorithm that can see the entire input in advance. More specifically, there is a cost function that measures the quality of the solution produced by an algorithm on a specific input. The *competitive ratio* of an on-line algorithm A is then defined to be the worst case over all inputs of the ratio of A's cost to that of the optimal off-line algorithm.

Competitive analysis gives us the ability to make strong theoretical statements about the performance of algorithms without making probabilistic assumptions about the input. However, because of its worst-case nature, competitive analysis can yield results that are more pessimistic than what is observed in practice.

For example, the paging algorithm Least-Recently-Used(LRU) has a competitive ratio of k, where k is the number of slots in the fast memory, but on most programs its cost is within a small constant factor of optimal. The reason for this inconsistency is that competitive analysis does not take into account that request sequences generated by programs exhibit locality of reference. In [BIRS] we address this problem by modeling locality of reference with an *access graph* which constrains the request sequences. This gives a more refined comparison between algorithms and, in many cases, establishes the superiority of LRU over competing algorithms.

Many traditional problems in computer science have natural on-line formulations. In on-line graph coloring, for example, nodes are presented one at a time. When each node is presented, its edges to previously presented nodes are also given. Each node must be assigned a color different from the colors of its neighbors. In [I] Irani gives an optimal on-line algorithm for coloring a special class of graphs, the *d-inductive graphs*, and determines the influence of look-ahead on the quality of the colorings that can be obtained.

Randomization is a powerful tool for designing on-line algorithms. In [IRSW] we consider the List Update Problem, which models the processing of a sequence of access requests to a linked list. Sleator and Tarjan have given a deterministic algorithm that achieves a competitive ratio of 2, and this is best possible for deterministic algorithms. Randomization allows the competitive ratio to be reduced to $\sqrt{3}$.

In [KVV] R. Karp, U. Vazirani and V. Vazirani obtain an optimal randomized on-line algorithm for the construction of a matching in a bipartite graph. Each edge in the graph connects a *boy* with a *girl*. It is assumed that the boys arrive one at a time, and, as each boy arrives, the algorithm must decide which adjacent girl, if any, to match him with. The expected size of the matching produced by the optimal randomized algorithm is guaranteed to be at least $1 - \frac{1}{e}$ of the size of a maximum matching, where e is the base of natural logarithms.

Much of our work on on-line algorithms is concerned with the famous *k-server problem*. The problem is set in a metric space. Requests occur at points in the space, and each request must be served by sending one of k servers to the request point. The cost of processing the request is the distance traveled by the selected server.

The paper [BBKTW] classifies the possible adversaries that may create request sequences to foil a randomized algorithm for the k-server problem. The main distinction is between oblivious adversaries, which do not observe the behavior of the on-line algorithm, and adaptive adversaries, which observe the on-line algorithm's behavior step by step. It is shown that any randomized algorithm that is c-competitive against adaptive adversaries can be converted to a c^2-competitive deterministic algorithm.

The paper [AKW] investigates a zero-sum game related to choosing a spanning tree T in a graph G such that the distance function of T is a good approximation to the distance function for G. The game arises in connection with the k-server problem on a road network. Optimal strategies for the game yield randomized algorithms for the k-server problem on such road networks.

S. Irani, R. Karp, M. Luby and M. Kearns have been studying the k-server problem under the constraint that the request sequence is generated by a fixed but unknown and arbitrary probability distribution. This is in contrast to the usual worst-case assumption, where the sequence is generated by an adversary. Initial results demonstrate that considerable improvements in the competitive ratio may be obtained by moving from the worst-case assumption to the probabilistic model.

R. Karp and A. Fiat have been considering on-line algorithms for certain economic problems. For example, they determine the best competitive ratio achievable by a currency conversion algorithm that must make decisions about converting (say) Deutschmarks to dollars without knowing future exchange rates.

References

[AKW] Alon, N., Karp, R., and West, D. "A graph-theoretic game and its application to the k-server problem," to appear.

[BBKTW] Ben-David, B., Borodin, A., Karp, R., Tardos, G., and Wigderson, A. (1990). "On the power of randomization in on-line algorithms," *Proc. ACM Symp. on Theory of Computing.*

[BIRS] Borodin, A., Irani, S., Raghavan, P., and Schieber, B. (1991). "Competitive paging with locality of reference," to appear in *Proc. ACM Symp. on Theory of Computing.*

[I] Irani, S. (1990). "Coloring inductive graphs on-line," *Proc. 31st Symp. on Foundations of Computer Science*, pp. 470-479. Invited to appear in a special issue of *Algorithmica.*

[IRSW] Irani, S., Reingold, N., Sleator, D., and Westbrook, J. (1991). "Randomized competitive algorithms for the list update problem," *Proc. ACM-SIAM Symp. on Discrete Algorithms.*

[KVV] Karp, R., Vazirani, U., and Vazirani, V., (1990). "An optimal algorithm for on-line bipartite matching," *Proc. ACM Symp. on Theory of Computing.*

3.6 Computational Learning Theory

Since arriving at ICSI in September, Michael Kearns has continued his research in computational learning theory, a branch of theoretical computer science that explores the possibilities for efficient machine learning.

With David Haussler of U.C. Santa Cruz and Rob Schapire of Harvard [HKS], Kearns has developed an information-theoretic interpretation of several aspects of the "probably approximately correct" model of learning first introduced by Les Valiant. This view yields sharpened bounds on the number of random examples required to accurately learn an unknown target function chosen from a known class of functions. It also provides a unifying explanation for several previous results in this learning model.

Kearns has also continued the investigation of learning probabilistic concepts begun with Schapire [KS]. Here the goal is to model the fact that many concepts are inherently uncertain, but still seem to be learned by humans in a rather strong sense. Kearns has been developing algorithms for probabilistic concept classes intended to partially capture some realistic form of uncertainty.

Recently Kearns has begun investigating extensions of existing theoretical learning models in which the learner must do more than simple concept learning. These models are partly based on literature on categorization from psychology, and ask that the learner not only infer basic concepts well, but also construct an accurate hierarchy of inclusion among concepts.

References

[HKS] Haussler, D., Kearns, M., Schapire, R., "Learning and information theory," in preparation.

[KS] Kearns, M., and Schapire, R., (1990). "Efficient distribution-free learning of probabilistic concepts," *Proc. 31st IEEE Symp. on the Foundations of Computer Science.*

3.7 Theory of Neural Networks

M.W. Hirsch and B. Baird collaborated on the design and simulation of neural networks. They concentrated on the theory of such nets, and also ran some tests on prototypes in collaboration with F. Eekman of Lawrence Livermore National Laboratory.

A specific task for which the nets are designed is the learning and reconstruction of handwritten digits. Preliminary results are encouraging: learning appears to be fast and flexible, and reasonably reliable.

These are recurrent nets, running in continuous time, with continuous state spaces. The aim is to investigate the usefulness of periodic or even chaotic attractors for storing, recognizing, classifying and reconstructing data. A novel feature is that the underlying dynamical system (a system of ordinary differential equations) is chosen in advance, independent of the data. Any of a number of learning rules linearly transforms data vectors into the coordinates of the standard dynamical system. Thus learning rules are simply coordinate changes, and (unlike most nets), they do not influence the dynamics. Further work will refine and study such nets and learning rules.

References

[BE] Baird, B., and Eekman, F., (1991). "CAM storage of analog patterns and continuous sequences with $3N^2$ weights," to appear in *Advances in Neural Information Processing Systems 3,* D.S. Touretzky and R. Lippman (eds.).

3.8 Models of Parallel Computation

In [PP] G. Pucci and M. Pinotti introduce the Parallel Priority Queue abstract data type. A PPQ stores a set of integer-valued items and provides for the concurrent insertion or deletion of sets of items. The algorithms for realizing PPQ operations on a CREW PRAM are based on two new data structures, the *n-Bandwidth-Heap* $(n-H)$ and the *n-Bandwidth-Leftist-Heap* $(n-L)$. Using these structures, efficient algorithms are given for concurrent insertion, concurrent deletion and for melding two PPQs.

In [P] Pucci surveys the existing models of asynchronous PRAMs. This bibliographic investigation has revealed that all the asynchronous models in the literature assume a completely unpredictable interleaving of the processors' steps. As a consequence, algorithms for these models have to be designed to be correct regardless of any variation in processor speeds. This assumption is far too restrictive. Pucci is considering less restrictive models, within which it may be possible to find efficient ways of balancing the computational load for a number of parallel programming techniques, such as tree-based computations, pointer jumping or divide-and-conquer.

References

[PP] Pinotti, M., and Pucci, G. "Parallel priority queues," to appear.

[P] Pucci, G. "Making the PRAM feasible. An overview of more realistic models of shared memory computation," to appear.

4 Realization of Massively Parallel Systems

In 1990, we have continued our research in the area of the design of parallel systems. The work can be viewed as consisting of five major areas:

1. Architectures for connectionist computation
2. Connectionist recognition of continuous speech
3. System Design - in particular, focused on the RAP machine
4. Digital VLSI for connectionist applications
5. Simulation tools for connectionist design

4.1 Architectures for Connectionist Computation

We have continued our evaluation of existing computer architectures for use in connectionist applications. The goal of this study has been to identify the architectural requirements that connectionist models impose on massively parallel digital hardware, including the effects of algorithms for learning. This research encompasses hardware and software design considerations for large scale connectionist networks and the detailed analysis of the design space. Different implementation techniques require unique data structures and impose different requirements on the underlying architecture. The study not only deals with the global architectural structure of such massively parallel systems, but also with the design of the individual processing elements to optimally support Artificial Neural Networks (ANNs).

The design of the connectionist simulator in an object-oriented language (described in Section 4.5) has led us to investigate architectural requirements and synchronization issues for parallel object-oriented systems. Object oriented systems can conceptually be regarded as message passing systems. However, efficient implementations utilize data sharing in a shared memory model. The compiler for the language Sather, currently under development at ICSI, also assumes a global shared memory to optimize code generation. In the last year we have been studying shared memory architectures and their synchronization primitives. Closely related to synchronization issues is the memory organization in a shared memory architecture. If all processors in the system see the memory in the same state at all times we speak of a coherent memory model. However, to guarantee memory coherency requires a significant implementation effort and might decrease system performance. The advantage of memory coherent systems, on the other hand, is that they considerably ease inter-processor synchronization. In view of our requirements we have investigated different memory coherency models. Current standardization efforts to define a high-speed scalable coherent interface (SCI) protocol look very promising; however, the lack of hardware chips which support the protocol makes it premature to asses the viability of SCI [1]. It might also not be necessary to enforce strict coherency, and much of the hardware support for coherent memory systems could potentially be implemented in software, thereby considerably reducing the hardware complexity of the system.

Another important issue for massively parallel architectures is fault tolerance. We have examined this issue for the special case of feed-forward ANNs [2]. The errors resulting from defective units and faulty weights in layered feed-forward ANN's have been analyzed, and techniques to make these networks more robust against such failures have been explored. For instance, using some simple examples of pattern classification tasks and of analog function approximation, it has been demonstrated that standard architectures subjected to normal backpropagation training techniques do not lead to any noteworthy fault tolerance. Additional, redundant hardware coupled with suitable new training techniques is necessary to achieve that goal. A simple and general procedure was introduced to develop fault tolerance in neural networks: the type of failures that one might expect to occur during operation are introduced at random during the training of the network, and the resulting output errors are used in a standard way for backpropagation and weight adjustment. The result of this training method is a modified internal representation that is not only more robust to the type of failures encountered in training, but which is also more tolerant of faults for which the network has not been explicitly trained. For systems that perform analog function approximation, complete fault tolerance is difficult to achieve. The contributions of the individual basis functions from which the output function is composed have to be carefully balanced, so that neither the loss of any single basis function nor an extreme, saturated signal from any unit can produce an excessive output error. The domain of the individual basis functions should be chosen just wide enough for adequate generalization, but otherwise kept as small as possible, in order to limit the extent of any saturated weight errors. To obtain more general fault tolerance to within a higher accuracy for analog data approximation tasks, a hierarchical approach seems preferable. Several redundant subnets have been used to perform the same approximation task in parallel, and a supervisory circuit combines their outputs by eliminating the signals that fall outside some margin and by averaging the other subnet outputs.

References

[1] Alnaes, K., et al. (1990). "Scalable Coherent Interface," SLAC-PUB-519, Stanford Linear Accelerator Laboratory.

[2] Sequin, C., and Clay, R. (1990). "Fault Tolerance in Feed-forward Artificial Neural Networks," Technical Report TR-90-031, International Computer Science Institute, Berkeley, CA.

4.2 Connectionist Recognition of Continuous Speech

Speech recognition continues to be a focus within the group. Recent results from this work showed the utility of Multilayer Perceptron (MLP) algorithms for estimating joint probabilities required by Hidden Markov Model (HMM) speech recognition systems [1, 2, 3]. The algorithms that we have developed for this purpose have performed better than a conventional probabilistic estimator, as shown in tables showing word recognition performance for speaker-dependent 1000-word continuous speech recognition tasks. Table I show results for 11 speakers from the English language Resource Management task. These results are somewhat higher than the ones reported in [2], due to improvements in training that were implemented late in the year.

Much of the work in 1990 was centered on preparation for coming studies in feature extraction and selection to further improve the estimation system. In particular, a recent workshop held at ICSI explored plausible feature sets for such a system [4]. As a result of these studies, we have begun to implement our hybrid HMM/MLP system using Perceptual Linear Prediction (PLP) features [5], along with their temporal derivatives, as inputs to the MLP [6]. Table II shows the advantage of using these new continuous input features over the older vector-quantized mel cepstrum. Another experiment suggested that PLP features were at least as good, and probably more succinct, than their continuous mel cepstral counterparts. Finally, some preliminary work suggested that simple modifications of the PLP analysis provided some immunity to spectral deformations of the channel for speech recognition over the telephone.

We also have further developed our Multilayer Perceptron software to increase its flexibility and in preparation for running on the RAP machine this summer (described below). Finally, we have begun to extend the capabilities of the Hidden Markov Models we are using to include the influence of phonetic context, for instance for commonly occurring words in the training. This latter work has just begun, but for the cases of the best and worst speakers of Table I, these approaches appear to raise recognition scores significantly.

References

[1] Bourlard, H., and Morgan, N. (1990). "Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition," in *Artificial Neural Networks: Advances and Applications*, North–Holland Press, E. Gelenbe (ed.).

[2] Morgan, N., and Bourlard, H. (1990). "Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico.

[3] Morgan, N., Wooters, C., Bourlard, H., and Cohen, M. (1990). "Continuous Speech Recognition on the Resource Management Database using Connectionist Probability Estimation," Technical Report TR-90-044, International Computer Science Institute, Berkeley, CA; also in *Proceedings of ICSLP-90*, 1337-1340, Kobe, Japan.

[4] Morgan, N., Hermansky, H., and Wooters, C. (1990). "SPOONS '90: The SPeech recOgnition frOnt eNd workShop," Technical Report TR-90-045, International Computer Science Institute, Berkeley, CA.

[5] Hermansky, H. (1990). "Perceptual Linear Prediction of Speech," J. Acoust. Soc. Am. 87 (4), April, 1990.

[6] Morgan, N., Hermansky, H., Bourlard, H., Kohn, P., and Wooters, C. (1991). *Continuous Speech Recognition Using PLP Analysis with Multilayer Perceptrons*, ICASSP '91, in Press.

4.3 System Design - currently focused on the RAP Machine

In 1990 we further developed our prototype Ring Array Processor (RAP), a multiprocessor system to speed up MLP calculations, particularly for the speech project [6, 7, 1, 3, 4, 5]. This system is based on powerful floating-point programmable Digital Signal Processor (DSP) chips (the TI TMS320C30) connected by a distribution ring which can shift data between processors at high speed with minimal processor intervention. Ring communication architectures can deliver high bandwidth with minimal hardware overhead by using simple hardware semaphores to implement a pipelined data flow. The RAP's pipeline communication ring has been implemented with Programmable Gate Arrays. These single 175-pin devices contain most of the inter-node wiring complexity of a communication pipeline stage, yet they are fast enough to implement short hand-shake protocols in a single RAP cycle. In addition, because of the availability of 640 reconfigurable logic blocks within the chip, other modifications to the data transfer protocol can be added at a later time. Programmable gate arrays have also been used to interface the processors to their local memories. One gate array per processor node multiplexes the address lines and generates the address strobes as required by the DRAMs, and controls access and provides data buffering for the VME system to the node memory. Through the use of these circuits it has been possible to fit 4 processors, 16 MBytes of dynamic memory (upgradeable to 64 MBytes), 1 MByte static memory, and the pipeline communication logic on one board. Our prototype design has a peak performance of 128 MFlops/board, and expected communication overhead of no more than 10-20% for algorithms of current interest. Several six-layer printed circuit boards have been completed and have been successfully programmed to implement a number of functions including common matrix-vector library routines, and the forward and backward phases of back-propagation (Table III). The first row of Table III shows the performance of a single board for a routine which has been hand-tuned. For a large enough dimension, this routine approaches 100% efficiency, and the single RAP board is roughly 50 times the speed of a Sun SparcStation 1 running the same benchmark. For smaller problems, such as the 32x32 case, the more generally useful "parameterized" routine (for which the dimension is a passed parameter) gives at most a 25% efficiency loss. Similarly, for the larger networks, the forward propagation performance becomes almost identical to the matrix-vector multiply (which is $O(N^2)$), since the sigmoid calculation (which is $O(N)$) becomes inconsequential in comparison. We anticipate that the system should scale up slightly, but not beyond about eight boards (32 processors). The current system has run large speech recognition training problems on a 3-board RAP, and performance for these tasks has scaled roughly linearly for the large problems currently of interest, which compute roughly 200,000 to 400,000 connections for each speech input frame. A new fabrication run is about to begin, which will yield 10 more boards.

RAP Software development is aided by a Sun-workstation-based command interpreter, tools from the standard C environment and a library of matrix and vector routines. The latter routines are written in TI assembly language to optimize performance, but still are parameterized routines so that matrices and vectors of varying sizes can be accommodated. A user programming the RAP produces either C source or TI assembler code using a text editor. This code is then compiled (and/or assembled) by the TI software tools and is then linked with the RAP library, all running on a Sun workstation. This produces an executable TMS320C30 file ready for loading. Using the command interpreter (RAPMC), a user can load executable files to the RAP and control the execution and debugging either globally or on a per-processor basis. Multiple RAPMCs may be used to direct text output from particular RAP nodes to separate windows. The board is now in daily use for our computationally-intensive problems in network training for speech recognition.

Finally, the general power of pipelined architectures, of which the RAP is a special instance, has led us to investigate software techniques to pipeline code in multiprocessor systems [2]. This technique employs the concept of software pipelining and an architecture to support this technique. Rather than attempting to pipeline a sequence of individual instructions, this technique tries to pipeline entire blocks of code, i.e. the units to be pipelined are chunks of code. In this model blocks of code are identified which can be executed in a pipelined fashion. Neighboring blocks of code do not need to be data independent; pipeline stages can feed results and/or synchronization markers on to the next pipeline stage. The architecture can be seen as an attempt to use classical pipelining techniques in a multiprocessor system. The architecture consists of a circular pipeline of ordinary microprocessors and is very similar to the RAP architecture. The advantage of this technique of software pipelining is that it does not require a static task allocation and data distribution. This techniques is especially suitable to pipeline loop iterations or recursive function calls. The execution model has been simulated on our architectural simulator. We have also simulated the effect of various cache organization on the system performance. For the pipelining of regular loop iterations the simulation results are very encouraging.

References

[1] Beck, J. (1990). "The Ring Array Processor (RAP): Hardware," Technical Report TR-90-048, International Computer Science Institute, Berkeley, CA.

[2] Beer, J. (1990). "A Pipelining Model Which Pipelines Blocks of Code," Technical Report TR-90-053, International Computer Science Institute, Berkeley, CA.

Table I. Resource Management Word Recognition Results
(No Grammar; Perplexity = 1000)

speaker name	ML	MLP(9)
jws04	48.2	62.3
bef03	39.3	56.7
cmr02	59.5	70.9
dtb03	49.8	61.2
das12	63.8	76.5
ers07	45.4	58.3
dms04	58.0	69.1
tab07	60.8	70.5
hxs06	60.9	76.3
rkm05	37.9	53.8
pgh01	50.4	63.6
mean	52.2	65.4

Table II. Pilot Feature Study, Speaker DTD
(No Grammar; Perplexity = 1000)

		mel cep VQ	PLP	PLP	PLP+slope
TYPE	INPUT	(4 codebooks)		+slope	+curvature
Frame ASR	single-frame	51.2	51.5	66.3	69.1
Frame ASR	multi-frame	67.9	69.8	72.4	72.6
Word ASR	single-frame	47.6	50.1	63.0	67.1
Word ASR	multi-frame	63.5	65.2	70.7	70.1

Table III. Measured performance, 1 RAP board, 16 MHz clock

operation	32x32	256x256
	internal RAM	external static RAM
optimized matrix × vector (MMACS)	40.6	61.4
parameterized matrix × vector (MMACS)	29.1	59.3
parameterized forward propagation (MCPS)	23.5	56.9
parameterized forward plus learning (MCUPS)	8.3	13.2

MMACs = Millions of Multiply/Accumulates Per Second

MCPS = Millions of Connections Per Second

MCUPS = Millions of Connection Updates Per Second

[3] Bilmes, J., and Kohn, P. (19909). "Ring Array Processor (RAP): Software Architecture," Technical Report TR-90-050, International Computer Science Institute, Berkeley, CA.

[4] Kohn, P., and Bilmes, J. (1990). "Ring Array Processor (RAP): Software User's Manual Version 1.0," Technical Report TR-90-049, International Computer Science Institute, Berkeley, CA.

[5] Morgan, M. (1990) "The RAP Machine: Algorithms and Architecture," Technical Report TR-90-047, International Computer Science Institute, Berkeley, CA.

[6] Morgan, N., Beck, J., Allman, E. and Beer, J. (1990). "RAP: A Ring Array Processor for Multilayer Perceptron Applications," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Albuquerque, NM.

[7] Morgan, N., Beck, J., Kohn, P., Bilmes, J., Allman, E., and Beer, J. (1990). "The RAP: a Ring Array Processor for Layered Network Calculations," *Proc. of Intl. Conf. on Application Specific Array Processors*, pp. 296-308, *IEEE Computer Society Press*, Princeton, NJ.

4.4 Digital VLSI for Connectionist Applications

We have previously established a limited capability for application–specific integrated circuit (ASIC) design. Semi–automatic IC design and layout tools have been imported from UC Berkeley, and silicon foundries (via MOSIS) provide the manufacturing capability. Realization group members are using the CAD tools to design small test chips, and working closely with EECS faculty (in particular Prof. John Wawrzynek of the Computer Science Division) to improve these tools while generating standard blocks of interest to us. While most experimental ANN circuits have used analog designs [3, 4], the digital approach we are investigating appears to have potential advantages in terms of ease of design, circuit time-sharing, flexibility to do high resolution or logical operations where required, and speed (particularly for scaled sub–micron circuits) [5, 6, 7].

We have laid out library cells using BOSS, an experimental object-oriented system designed at UC Berkeley. These cells are building blocks for the saturating finite-precision arithmetic we require for our algorithms. We are also developing a successor system, OctC++ [1]. While BOSS was designed as a single module to speed implementation, OctC++ is being split into more manageable chunks. The lowest layer of OctC++ consists of C++ wrappers around the Berkeley Octtools database. This code will help isolate the higher layers from future changes in the design database. Higher level layers will implement the abstractions present in BOSS, such as cell arrays and wiring pragmas. BOSS is fairly conservative in its use of the novel features of C++, while OctC++ attempts to further capitalize on inheritance, dynamic typing, and operator overloading to ease the library design process.

Cells have been fitted together into larger blocks for which we also expect repeated use, such as fast static RAM, saturating data paths, and nonlinear operators. Several prototype chips have been designed to test these larger cells. For instance, we have recently implemented a 2 micron version of a saturating data path, and successfully tested it at 20 MHz. Software simulations of limited-precision versions of our network algorithms have also been run, suggesting precision as low as 6 bits for most of the variables used in our speech-related research, such as connection strength increments. We have also done a preliminary design of a fully pipelined layered network circuit which we believe will be an extremely area-efficient implementation [2].

References

[1] Asanovic, K. "OctC++: A C++ Interface to the Oct Database," Technical Report, International Computer Science Institute, Berkeley, CA, in prep.

[2] Asanovic, K., Kingsbury, B., Morgan, N., and Wawrzynek, J., "A Highly Pipelined Architecture for Neural Network Training," *IFIP Workshop on Silicon Architectures for Neural Nets*, in Press.

[3] Mead, C. (1989). "Analog VLSI and Neural Systems," Addison Wesley, Reading, MA.

[4] Morgan, N. (ed.) (1990). "Artificial Neural Networks: Electronic Implementations," *Computer Society Press Technology Series Computer Society Press of the IEEE*, Washington, DC.

[5] Morgan, N. (1990). "Considerations for the Electronic Implementation of Artificial Neural Networks," Technical Report TR-90-003, International Computer Science Institute, Berkeley, CA.

[6] Morgan, N. (1990). "Digital Implementations of Artificial Neural Networks," *Proceedings of the 1st International Workshop on Microelectronics for Neural Networks*, pp. 187-195, University of Dortmund.

[7] Morgan, N., Asanovic, K., Kingsbury, B., and Wawrzynek, J. (1990). "Developments in Digital VLSI Design for Artificial Neural Networks," Technical Report TR-90-065, International Computer Science Institute, Berkeley, CA.

4.5 Simulation Tools for Connectionist Design

A prototype of a new algorithmic simulator (ICSIM) for connectionist networks is currently being designed and tested. The focus of our initial design is on the structure of the class libraries. The implementation is in the object-oriented Eiffel 2 language.

An object-oriented language was chosen because the object-oriented approach facilitates flexibility, extensibility, modularization, and efficiency. These criteria are very important for a simulator in a highly exploratory field of research such as the design of artificial neural nets. Flexibility is important due to the different mathematical models underlying neural nets and the different network architectures and applications. Efficiency is equally important due to the conceptually massively parallel computation. For instance, the simulation of real-time speech recognition methods on sequential machines may take hours. Ultimately efficient simulation must be achieved by parallelizing it. This goal requires consideration in the design of ICSIM right from the beginning in order to allow simulation objects to be distributed on different processors with ease. The implementation realizes modular composition of nets. The approach is partly synthetic in that pre-defined incomplete library classes can be combined and instantiated to form a complete working model to be simulated. Fast prototyping can also be accomplished by modifying complete and working library classes to derive new net behavior.

Shared structures such as shared subnets, shared connections or shared weights are supported both to achieve compact representations of models and to allow experiments with learning mechanisms in which different subnets interfere with each other.

The kernel classes, such as 'unit', 'link', etc., have been designed and implemented. A number of examples have been used to test how connectionist nets can be specified and implemented using ICSIM. A tutorial describing ICSIM's features and their usage has been written [1]. This will allow other researchers to experiment with the initial prototype of ICSIM and to provide valuable feedback about the design of ICSIM and its functionality.

Some experimental classes for a net objectbase and for graphic presentation (net views) have been implemented independently of ICSIM. These will be integrated in a revised form later on.

A number of user interface library classes were implemented in Sather (see Section 2.3) and first steps towards porting ICSIM to Sather were taken. The user interface classes build on the Xview library written in C. The goal is to rewrite the simulator in Sather in order to obtain an implementation that is more efficient than the Eiffel implementation and to utilize a well-maintained graphic user interface rather than implementing one from scratch.

Beside the Sather compiler that is written in Sather the implementation was one of the first major applications of Sather with a moderate number of interdependent classes and an extensive use of the foreign function capabilities of Sather which allows to integrate Sather programs with C packages. Various improvements and suggestions for the Sather language and its implementation were worked out. In particular the type system of Sather was studied in detail and could be improved. Some minor tools were developed in support of editing and managing Sather programs such as editing and compilation functions under GNU Emacs.

Two papers on the semantics and analysis of parallel systems [2, 3] that were submitted at the end of 89, were revised and one of them was presented at the IEEE COMPSAC 90 Conf. in Chicago. This work relates the theory of abstract data types with that of Petri nets and studies to which extent analysis methods of either side can be extended to a specification method integrating both. It is shown that net analysis methods for deadlock detection, liveness and safeness, can be generalized to work on the extended specification and that and how rewrite methods can be effectively used in net simulations and analysis. The GRASPIN symbolic execution environment is described in which small models of intended parallel systems can be prototyped early by generating executable code from their specifications rather than needing to program them in full detail.

References

[1] Schmidt, H. (1990) "ICSIM: Initial design of an object–oriented net simulator," Technical Report TR-90-055, International Computer Science Institute, Berkeley, CA.

[2] Schmidt, H. (1990). "Prototyping and Analysis of Non-Sequential Systems Using Predicate-Event Nets," Technical Report TR-90-020, International Computer Science Institute, Berkeley, CA. Revised version to appear in: *JSS 91*, special issue on GRASPIN.

[3] Kraemer, B., and Schmidt, H. (1990). "Architecture and Functionality of a Specification Environment for Distributed Software," *Proc. 14th Intl. IEEE COMPSAC 90 Conf.*

5 Networks and Distributed Systems

5.1 The TENET Group

5.1.1 Introduction

The Tenet Group, under joint University of California and ICSI sponsorships, has done research in the field of very high speed wide-area networking. Tenet research has been conducted within the framework of the XUNET Project, which is led and supported by AT&T Bell Laboratories and involves, besides the Tenet Group, networking research groups at Bell Laboratories in Murray Hill and at the Universities of Wisconsin-Madison and Illinois at Urbana-Champaign, as well as scientific visualization research groups at the National Center for Supercomputer Applications at the University of Illinois, the Space Science and Engineering Center at the University of Wisconsin, and the Lawrence Berkeley Laboratory. During 1990, the XUNET Project has started its participation in the Gigabit Testbed Project launched by the Corporation for National Research Initiatives under NSF and DARPA support. The testbed, called BLANCA, is one of the five testbeds in the Gigabit Testbed Initiative. Tenet research has been supported also by the University of California under the MICRO Program and by Hitachi America, Ltd.

Topics in the following areas have been investigated by the Tenet Group: real–time communication, congestion control, multiplexing policies, network traffic modeling, and digital video communication technologies.

5.1.2 Real-Time Communication

The real-time communication project has completed the design of a scheme to provide deterministic and statistical delay, throughput, and loss bounds in a packet-switching network by attacking and solving the buffer space allocation problem; proposed a scheme for the establishment of channels with bounded delay jitter in a packet-switching network; devised algorithms for providing performance and reliability guarantees in the XUNET II testbed, which is an ATM network with hierarchical round-robin (HRR) scheduling in the switches; investigated in more general terms the question of offering bounded-delay, bounded-reliability services on ATM networks; designed protocols for the implementation of real-time communication services in an internetwork, which will be used to perform admission control experiments on XUNET II when this 45 Mbps testbed will be installed in early 1991; and begun a study of pricing policies and charging mechanisms suitable for networks offering a variety of services, including real-time ones.

Buffer Allocation for Real Time Communication (Dinesh Verma and Domenico Ferrari) A real time communication service allows a client to transport information with performance guarantees. The schemes in [Ferr89] and [FeVe90a] describe how such a service can be supported when the main performance attribute of interest is the delay of packets in a conversation. Those schemes, however, assume infinite buffer space at each node in the communication network. In this project, we have extended the scheme to provide guarantees about loss rates and designed a scheme to allocate buffer space to real-time channels. Thus, this work completes the real-time channel algorithm proposed in earlier studies. The details of the buffer allocation algorithms and simulation results are described in [FeVe90b].

Delay Jitter Control For Real Time Communication (Dinesh Verma, Hui Zhang, and Domenico Ferrari) Guaranteeing a bound on delay jitter is important in continuous media applications. In this project, we have extended the scheme presented in [Ferr89], [FeVe89a], and [FeVe89b] to provide deterministic jitter bounds for real time channels. A new rate control mechanism has been proposed to reduce the jitter along a channel's path. Compared to the solution of eliminating jitter by buffering in the receiver, our solution has the advantage of reducing buffer requirements significantly both at the receiver and within the network. Another advantage of our solution is that the provided jitter bounds are independent of the length of a channel's path; i.e., jitter does not accumulate along the path. We have given the proof of the correctness of our scheme and have shown the effectiveness of our solution by simulations [VeZF91].

Real–Time Channel Establishment in XUNET II (Anindo Banerjea and Dinesh Verma) Future high speed networks should be capable of providing performance guarantees for applications such as video, audio, and other multi-media communications. We are considering ways in which to provide such guarantees in a particular network, XUNET II. XUNET II is an experimental wide-area network under development at AT&T for Broadband ISDN. The key concepts upon which XUNET II is based are Asynchronous Transfer Mode (ATM) and Virtual Circuit Switching [AT&T89]. The underlying scheduling at the switches is Hierarchical Round Robin (HRR) [KaKK90].

Guarantees can be provided by reserving resources at each node along the path of a real-time channel during the establishment phase [FeVe90a]. We have designed a protocol which allows us to establish channels with low setup time and cost. The protocol requires a round trip of an establishment message, during which the necessary calculations are performed at each node and tentatively resources are reserved. Variations on this basic strategy have been proposed and are being tested by simulation.

Work has been done on the possibility of dynamically managing channels which have already been established, with the objective of relocating resources without violating guarantees, so that more channels can be established [BaGu90]. We have come up with some possible mechanisms, which we now need to analyze.

We have also worked on the HRR scheduling mechanism itself, and more specifically on the choice of the HRR structure, i.e., of the number of slots per level. We have come up with a mechanism to design the structure given the expected traffic, with the objective of obtaining higher utilizations.

Quality of Service in ATM Networks (Dinesh Verma and Domenico Ferrari) The ATM (the Asynchronous Transfer Mode) is the recommended transmission mode for Broadband Integrated Services Data Networks. The provision of quality–of–service guarantees to individual connections in such a network is recognized as an open problem.

In this project, we have tried to apply within the scope of an ATM network the ideas we devised to provide guaranteed performance communication in conventional store–and–forward packet switching networks. The main difference between the environment to which our previous work referred and the one of an ATM network is due to an ATM network's need for very simple and fast scheduling policies. This results in a very different architecture for switches to be used in ATM networks. On the other hand, the problem of providing performance guarantees is simplified somewhat by the fact that all data is transferred in small constant sized cells. The contribution of our work is a simple and practical mechanism to provide performance guarantees in ATM networks. This work is described in [VeFe90].

Real–Time Communication Protocols (Carlyn Lowery) We have investigated a methodology for supporting real-time communications, and proposed a real-time delivery system with channel control, as well as network protocols that may be used as a basis for experimentation [Lowe91].

Our real-time delivery system is composed of a new protocol for the administration of real-time connections combined with modifications to the Internet Protocol (IP) to support such connections. The methodology for providing real-time performance guarantees is based upon connection establishment with resource reservation, as described in [FeVe90a] and [FeVe90b]. The environment for which our real-time delivery system has been designed is an internetwork whose gateways implement the system itself, and whose subnets have performance characteristics that can be bounded. The experimental environment includes hosts connected via FDDI rings to the routers at the periphery of an ATM network (XUNET II). This network participates in the delivery system as a subnet that provides performance guarantees.

The real-time delivery system has been designed with multiple goals in mind. First, interoperability with existing Internet protocols was deemed to be desirable. Many applications will not have real-time requirements, and will continue to desire the services provided by TCP/IP and UDP/IP. Also, hosts that do not provide real-time services should be able to transmit and receive ordinary IP packets. Second, an iterative approach to development is desirable; for early implementations, minimal essential features will be included, and existing Internet services will provide valuable support.

Separate protocols are used for channel administration and data delivery. This clean separation is expected to make the enhancement of the control protocol more straightforward, to reduce the overhead processing for data packets, and to minimize the changes to be made to the standard IP header.

The control protocol is called the Real-Time Channel Administration Protocol (RCAP). It provides management services, such as channel establishment, teardown, and modification. Control packets are delivered by the network protocol, the Real-Time Internet Protocol (RTIP), and are identified by a unique protocol number in the RTIP header. RTIP is based on IP, but has the additional features of fixed routing, deadline-based scheduling, and rate control. The RTIP header is nearly identical to the IP header.

We have also briefly examined transport protocol issues, even though many of the transport layer functions are provided by real-time channels. Three categories of real-time clients with different transport-level requirements have been identified, and ways to satisfy their needs within the framework of our delivery system have been discussed. The three categories are those of clients who want: (1) real-time connections such as those provided by RCAP; (2) real-time connections with end-to-end flow control and retransmission; (3) express datagrams.

A Pricing Policy for Real-Time Channels (Colin J. Parris) We are investigating pricing policies for real-time channels in our real-time channel establishment scheme [FeVe90a]. The requirements of a useful pricing policy have been determined and used to analyze conventional pricing policies that are applied to similar utilities (e.g., power and telephone). The pricing policy is based on charges for resource reservations and usages. The reservation charge depends on the Type Of Service (TOS) requested for the channel, the duration of the channel's lifetime, and the period of the day during which the channel exists. The usage charge is based on the number of packets sent over the channel. The TOS parameter is a function of the node utilization, buffer utilization, and the value of the delay requested by the client. We are currently in the process of determining this function and verifying the validity of our pricing policy with the aid of simulations and analysis.

Client Requirements for Real–Time Communication Services (Domenico Ferrari) A real–time communication service provides its clients with the ability to specify their performance requirements and to obtain guarantees about the satisfaction of those requirements. We proposed a set of performance specifications that seem appropriate for such services; they include various types of delay bounds, and reliability bounds [Ferr90]. We also described other requirements and desirable properties from a client's viewpoint, and the ways in which each requirement is to be translated to make it suitable for lower levels in the protocol heirarchy. Furthermore, we presented some examples of requirements specification, to show how adequate the proposed characterizations of client wishes would be in practical solutions.

References

[AT&T89] "XUNET II Overview," October 26, 1989.

[BaGu90] Banerjea A., and Gupta, A. (1990). "Real Time Channels in ATM-switched HRR-scheduled Networks," CS292V Class Project Report, University of California, Berkeley.

[Ferr89] Ferrari, D. (1989). "Real-Time Communication in Packet Switching Wide-Area Networks," Technical Report TR-89-022, International Computer Science Institute, Berkeley, CA.

[Ferr90] Ferrari, D. (1990). "Client Requirements for Real–Time Communication Services," *Proc. International Conf. Info. Processing Soc. of Japan to Commemmorate the 30th Anniversary*, Tokyo, Japan, pp. 95–102; also *IEEE Communications Magazine* 28(11):65–72.

[FeVe90a] Ferrari, D., and Verma, D.C. (1990). "A Scheme for Real-Time Channel Establishment in Wide-Area Networks, *IEEE J. Selected Areas Comm.*, SAC-8(3):368-379.

[FeVe90b] Ferrari, D., and Verma, D.C. (1990) "Buffer Allocation for Real-Time Channels in a Packet-Switching Network," Technical Report TR-90-022, International Computer Science Institute, Berkeley, CA.

[KaKK90] Kalmanek, C.R., Kanakia, H., and Keshav, S. (1990). "Rate Controlled Servers for Very High-Speed Networks," to appear in *Proc. GlobeComm*, San Diego, CA.

[Lowe91] Lowery, C.M. (1991). "Protocols for Providing Performance Guarantees in a Packet Switching Internet," Technical Report TR-91-002, International Computer Science Institute, Berkeley, CA.

[VeFe90] Verma, D.C. and Ferrari, D. (1990). "Quality of Service in ATM Networks," Technical Report TR-90-062, International Computer Science Institute, Berkeley, CA.

[VeZF91] Verma, D.C., Zhang, H., and Ferrari, D. (1991). "Delay Jitter Control for Real Time Communication in a Packet Switching Network," to appear in *Proc. TriComm'91.*

5.1.3 Congestion Control

During 1990, we explored three ideas in depth: the Hierarchical Round Robin (HRR) service discipline, the Packet Pair (2P) flow control protocol, and a control-theoretic approach to flow control.

The HRR service discipline is a non-work-conserving queue service discipline that can provide bandwidth, delay, and jitter guarantees as well as best-effort service to virtual circuits. It is simple and can be efficiently implemented in hardware. Our analysis and simulations show that networks built using such a discipline will have many advantages over current networks [KaKK90].

The 2P protocol uses a simple passive state probe to determine the service rate in networks of virtual Time Division Multiplexed servers. This information is used to respond intelligently to changes in network state. Analysis and simulations indicate that this protocol will waste less bandwidth and drop less packets due to congestion than the other state-of-the-art protocols [KeAS90], [SiAK90].

The work on 2P led us to consider the question of optimal flow control. We believe that a control-theoretic approach can adequately capture the dynamics of virtual circuit flow, and hence is an appropriate approach to flow control. We have designed models of virtual circuits based on fluid-flow and are using techniques from minimum-variance control and linear-quadratic-gaussian control to perform optimal flow control.

In other work, we added a number of features to the REAL network simulator [Kesh88] and released REAL version 3.0. This software is being used at sites around the world.

References

[KaKK90] Kalmanek, C.R., Kanakia, H., and Keshav, S. (1990). "Rate-Controlled Servers for Very High-Speed Networks," *Proc. GLOBECOM '90*, San Diego, December 1990; also submitted to *J-SAC*, Special Issue on Congestion Control.

[KeAS90] Keshav, S., Agrawala, A., and Singh, S. (1990). "Design and Analysis of a Flow Control Algorithm for a Network of Rate Allocating Servers," *Proc. Second IFIP TC6/WG 6.1/6.4 International Workshop on Protocols for High Speed Networks*, Palo Alto, CA; also to appear in a book published by North–Holland.

[Kesh88] Keshav, S. (1988). "REAL: A Network Simulator," Technical Report UCB/CSD 88/472, University of California, Berkeley, Computer Science Division.

[SiAK90] Singh, S., Agrawala, A., and Keshav, S. (1990). "Deterministic Analysis of Flow and Congestion Control Policies in Virtual Circuits," Technical Report TR 2490, University of Maryland; also submitted to *J-SAC, Special Issue on Congestion Control.*

5.1.4 Multiplexing Traffic at the Entrance to a Wide Area Network

In a hierarchical wide-area network, many user conversations must be multiplexed at the points where local-area networks meet the long-haul portion of the network. These conversations are composed of different types of traffic, such as remote file transfer, remote terminal, and electronic mail data. The different traffic types offer different workloads to the network and demand different qualities of service from the network. It is important to provide each conversation with good performance while making efficient use of network resources.

We are evaluating traffic multiplexing schemes that can be used at these junction points to provide the desired functionality and performance. Alternatives range from a simple first-in first-out service that regards all incoming traffic as a single stream, to a round-robin service that cleanly separates user conversations. Intermediate solutions such as a round-robin service that separates different traffic types, but not necessarily individual conversations, will also be considered. For each multiplexing scheme, the delay and throughput observed by individual user conversations will be compared to the number of network level connections opened during a suitable time period. Network utilization will also be a metric of interest. This research is described in more detail in [Cace90].

During 1990, we continued the wide-area network traffic analysis introduced in [Cace89]. New wide-area Internet traffic traces were gathered at the University of California at Berkeley, and similar traces were obtained from Bellcore in Morristown, New Jersey. The traces contain raw packet headers and microsecond-resolution timestamps for all traffic flowing between these organizations and the rest of the Internet during a one-week period. Our analysis aims at characterizing the packet length and packet interarrival time behavior of the different types of traffic currently populating the Internet. We plan to produce realistic models of individual traffic sources for each of these types, as well as a realistic model of the mix of traffic types.

In 1990, we also constructed a wide-area network simulator to carry out the traffic multiplexing studies described above. The simulator incorporates realistic models of wide-area network components, including hosts, routers, switching elements, and communication lines. It will be driven by the traffic source models described above. We will concentrate our simulation studies on the routers that connect the local-area networks to the long-haul portion of the network. As we see them, these routers will play an important role in the effective multiplexing of different traffic types at the entrance to future wide-area networks.

References

[Cace89] Caceres, R. (1989). "Measurements of Wide Area Internet Traffic," Technical Report UCB/CSD 89/550, University of California, Berkeley, Computer Science Division.

[Cace90] Caceres, R. (1990). "Multiplexing Traffic at the Entrance to a Wide Area Network," Ph.D. Qualifying Examination, University of California, Berkeley.

5.1.5 Models of Variability of Packet Arrival Processes

We have modeled packet arrival processes produced by single-user workstation in a local-area network [Guse90b]. These models characterize interarrival time variability based on what we call a "generalized two-state semi-Markov model": a semi-Markov model in which intervals in each state are correlated, though intervals belonging to different states are independent. Variability is expressed in terms of indices of dispersion [Guse91].

The distribution of packet lengths of measured packet arrival processes shows a sharp distinction between packets shorter than 200 bytes (short packets) on one side and packets longer than 500 bytes (long packets) on the other side; there are virtually no occurrences of lengths between 200 and 500 bytes [Guse90a]. To build the models, we first show that the lengths of short packet sequences and the lengths of long packet sequences can be produced by a two-state Markov chain: one state for short packets, the other for long packets. Second, we associate to the states of the Markov chain the probability density functions (component functions) of the interarrival times of short and long packets. Third, we derive the index of dispersion for intervals for the generalized semi-Markov model in terms of the autocovariances of the interarrival densities of the two states.

By comparing the model's index–of–dispersion curves with those obtained from simulation runs with controllable parameters, we discover that the model does not fit nonstationary data and that in order to match measured packet arrival processes well, we have to limit its range of applicability to lags up to 30 ms and to interarrival times up to about 500 ms. However, these are the intervals over which the variability of arrival processes, produced and consumed by fast, interactive queuing systems, is relevant.

References

[Guse90a] Gusella, R. (1990). "Measurement Study of Diskless Workstation Traffic on an Ethernet," *IEEE Transactions on Communications*, 38(9):1557–1568.

[Guse90b] Gusella, R. (1990). "A Characterization of the Variability of Packet Arrival Processes in Workstation Networks," Technical Report UCB/CSD 90/612, University of California, Berkeley, Computer Science Division.

[Guse91] Gusella, R. (1991). "Characterizing the Variability of Arrival Processes with Indices of Dispersion," *IEEE Journal on Selected Areas in Communications*; to be published.

5.1.6 Digital Video Communication Technologies

In the area of digital video communication, a project has investigated the possibility and limitations of compressing/decompressing video streams using purely software methods. Another study has examined various techniques that can be used to support video conferencing traffic effectively in an SMDS (Switched Multimegabit Data Service) network as well as in a more traditional packet-switching network.

Real–Time Transmission and Software Decompression of Digital Video in a Workstation (Akihiro Okazaki and Kyoji Umemura) We have performed an experiment in which compressed motion video data was transferred via an Ethernet to a workstation, uncompressed by software, and displayed on the workstation. The workstation had no special compression/decompression hardware. The motion video stream had 30 frames per second, 192x114 pixels gray scale per frame, and represented humans speaking with a static background. The frames were displayed on a monochrome display with halftoning. The display area was 768x576 pixels; the video data were magnified by a factor of 4. The quality of output was suitable for applications such as video conferencing, video mail, etc. We chose the block truncation algorithm [HeMi81] for intra-frame compression, and the difference method [HaMC72] for inter-frame compression.

We experimented with two video streams, both of which show a person talking. One, called "Miss America" has a uniform background, while the other, called "Salesman," has a background with many objects. The maximum frame rates and data rates for the two streams are shown in the table below.

Sequence	Speed (frames/s)	Date rate (KB/s)
Miss America	72.0	225
Salesman	50.4	221

Networking Support for Video Conferencing (Mark Moran and Russell Pflug- haupt) Since Fall 1990 we have been working on a project to study the network services that best support real-time video applications. As we expect to perform video conferencing across an internetwork, we have looked at network-level and transport-level services. To simplify the study, we have concentrated on video conferencing as a representative and more easily modelled application. Our preliminary work involved attempting to map the needs of a restrictive video conferencing application unto two representative network services: SMDS, a connectionless network service developed by Bellcore [SMDS89], and the connection-oriented real-time network services developed by the Tenet group at ICSI [FeVe90a]. This work entailed selecting a suitable algorithm for video compression and mechanisms for avoiding and tolerating packet losses. Our major conclusions from the study are that connections are necessary for efficient transmission of video streams and that, for low bit-rate video conferencing (< 750Mb/s each way), forward error correction and prioritization of packets are too costly or difficult to implement. We have also suggested a transport service that will allow incomplete packets to be delivered and used [PfMo90]. These conclusions will be checked by simulation and possibly actual experimentation on the XUNET II testbed.

References

[FeVe90a] Ferrari, D., and Verma, D.C. (1990). "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Sel. Areas in Comm.* 8:368-379.

[HaMC72] Haskell, B.G., Mounts, F.W., and Candy, J.C. (1972). "Interframe Coding of Video Telephone Pictures," *Proc. IEEE* 60:792-800.

[HeMi81] Healy, D.J., and Michell, O.R. (1981). "Digital Video Bandwidth Compression Using Block Truncation Coding," *IEEE Trans. on Communications* COM-29(12).

[PfMo90] Pflughaupt R., and Moran, M. (1990). "SMaRTS: a Network Service for Real-Time Traffic," CS292V Class Project Report, University of California, Berkeley.

[SMDS89] "Generic System Requirements in Support of Switched Multi-Megabit Data Service," Bellcore Technical Advisory, TA-TSY-000772, Issue 3, October 1989.

5.2 The DASH Project

The DASH group, under the direction of David P. Anderson, has focused its research on software support for *continuous media* (audio and video) in distributed computer systems. Our target applications include audio/video conferencing, multimedia documents, and audio/video production. Several software levels are involved: network protocols, operating systems, window systems, and application toolkits. We are building prototype software systems, described below, to use as research platforms. We are also working on formal models for describing real-time requirements, scheduling of hardware resources, and application synchronization and ordering constraints. These models are integrated in the experimental systems, and are in turn influenced by our experiences with them.

Our approach is called *integrated digital continuous media* (IDCM). In this approach, audio and video data is handled in the same hardware and software frameworks as other data types. An IDCM system must combine the functions of current distributed systems with the stringent real-time requirements of continuous media. These requirements involve most components of the system: networks, protocols, file system, CPU scheduling, memory management, window systems, application programming models, etc.

A paper summarizing the IDCM philosophy and its implications for computer system design was presented in June 1990 at the 10th International Conference on Distributed Computer Systems [ATWGA, 1990]. The plans of the DASH project involve adding IDCM support to an existing system software environment based on UNIX, TCP/IP, and X11/NeWS. This plan is described in [AGHW, 1990].

In addition to Anderson, the DASH project has included three foreign visitors to ICSI during 1990 (Ralf Herrtwich, Luca Delgrossi, and Kyoji Umemura) as well as several graduate and undergraduate students from UC Berkeley. Together with Domenico Ferrari's Tenet group, we helped organize a Workshop on Operating System and Network Support for Continuous Media, held at ICSI in November 1990.

5.2.1 Window System Extensions for Continuous Media

ACME is a set of abstractions for the input and output of "continuous media" (audio and video) by a network window system. In ACME, continuous media (CM) data is handled by software in both the client and server, and is conveyed on network connections. The ACME design has the goals of network transparency, device independence, ability to support concurrency, and management policy independence. The ACME abstractions include *strands* (streams of audio or video data), *ropes* (combinations of several strands), *logical time systems* (reference frames in which several strands or ropes can be played synchronously), and *logical devices* (representing microphones, speakers, video cameras, and video windows). A paper on ACME was presented by Ramesh Govindan at the 1991 Int. Conf. on Multimedia Information Systems [AGH, 1991].

ACME is intended as an addition to a network window system such as X11 or NeWS. We have explored the design issues in such a server, including 1) the interaction between the base window system and the ACME component; 2) the process structure, real-time techniques, and IPC requirements of the ACME component; 3) the interface to CM I/O devices, and 4) synchronization of parallel CM data streams. These issues are discussed in [HGA, 1990] and [AH, 1991].

We have implemented a prototype ACME server. It is written in C++, and is based on a real-time lightweight process library developed by Anderson and Jeff Bilmes [AB, 1991]. The ACME server runs on Sparcstations at ICSI. It supports multiple concurrent channels of audio input and output at different sampling rates, as well as compressed and uncompressed video output [UO, 1991]. We have integrated the ACME server with Sun's OpenWindows window system (an X11/NeWS hybrid), so that control operations can be done by both NeWS and X11 clients. Pamela Chan has extended the Xlib client library to include support for multiple streams of CM I/O, also using the lightweight process library. We are developing CM applications using an ACME server. These include remote audio playback and record, telephony, conference server, ambient sound generator, and multi-part score playback.

5.2.2 Scheduling Model for IDCM

The most basic problem in supporting IDCM is how to schedule devices in a way that satisfies the real-time requirements of CM. We have developed the *DASH resource model*, a workload and scheduling model that allows communicating peer entities to reserve the resources (such as CPU and network bandwidth) necessary to achieve given delay and throughput objectives. The immediate goal of the model is to support digital audio and video in distributed systems. The model defines a parameterization of client workload, an abstract interface for hardware resources, and an end-to-end algorithm for negotiated resource reservation based on cost minimization.

The DASH resource model was originally presented in [ATWGA, 1990], and later by Ralf Herrtwich in May 1990 at the IEEE Workshop on Real-Time Operating Systems and Software [AH, 1990a]. The most recent and complete description is given in [A, 1990].

5.2.3 Time Capsules

Ralf Herrtwich, a postdoctoral visitor to the DASH group, has been working on a formal model for describing distributed CM applications [H, 1990b]. This model, presented at the 11th IEEE

Real-Time Systems Symposium in December 1990 [H, 1990a] is at a higher level than that of the DASH resource model. It defines an abstraction called *time capsules* that can be used to represent a source or sink of CM data; these can included files, I/O devices, processes, and communication channels. Each time capsule has an associated *clock* that determines its rate, and whose speed may be varied. Applications can be represented as directed acyclic graphs of time capsules. By analyzing these graphs, optimal delay assignments and start times can be determined.

Herrtwich and other members of the DASH group designed a set of extensions to the Office Document Architecture (ODA) for accommodating continuous media elements, and for expressing the temporal relationships between these elements [HD, 1990].

5.2.4 Operating System and Network Support for Continuous Media

In addition to higher-level issues, we are actively researching issues in how operating systems and networks can better support CM. We designed the Session Reservation Protocol (SRP), an IP-based protocol that implements the end-to-end session establishment algorithm of the DASH resource model. The design of SRP is described in [AH, 1990b, AHS, 1990]. Luca Delgrossi implemented SRP in the Mach 2.5 operating system. The issues involved are discussed in [ADH, 1990].

References

[A, 1990] Anderson, D.P. (1990). "Meta-Scheduling for Distributed Continuous Media," Technical Report UCB/CSD 90/599, University of California, Berkeley, Computer Science Division.

[AB, 1991] Anderson, D.P. and Bilmes, J. (1991). "Concurrent Real-Time Music in C++," *USENIX C++ Workshop*, June 1991.

[ADH, 1990] Anderson, D.P., Delgrossi, L. and Herrtwich, R.G. (1990). "Process Structure and Scheduling in Real-Time Protocol Implementations," Technical Report TR-90-021, International Computer Science Institute, Berkeley, CA.

[AGH, 1991] Anderson, D.P., Govindan, R. and Homsy, G. (1991). "Abstractions for Continuous Media in a Network Window System," *International Conference on Multimedia Information Systems*, Singapore, January 1991.

[AGHW, 1990] Anderson, D.P., Govindan, R., Homsy, G., and Wahbe, R. (1990). "Integrated Digital Continuous Media: a Framework Based on Mach, X11, and TCP/IP," Technical Report UCB/CSD 90/566, University of California, Berkeley, Computer Science Division.

[AH, 1990a] Anderson, D.P. and Herrtwich, R.G. (1990). "Resource Management for Digital Audio and Video," *IEEE Workshop on Real-Time Operating Systems and Software*, Charlottesville, May 1990, 99-103.

[AH, 1990b] Anderson, D.P. and Herrtwich, R.G. (1990). "Resource Management in Distributed Multimedia Systems — A Solution for the Internet," *IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, Berlin, October 1990.

[AHS, 1990] Anderson, D.P., Herrtwich, R.G. and Schaefer, C. (1990). "SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet," Technical Report TR-90-006, International Computer Science Institute, Berkeley, CA.

[AH, 1991] Anderson, D.P. and Homsy, G. (1991). "Synchronization Policies and Mechanisms in a Continuous Media I/O Server," Technical Report UCB/CSD 91/617, University of California, Berkeley, Computer Science Division.

[ATWGA, 1990] Anderson, D.P., Tzou, S., Wahbe, R., Govindan, R., and Andrews, M. (1990). "Support for Continuous Media in the DASH System," *Proc. of the 10th International Conference on Distributed Computing Systems*, Paris, May 1990; also Technical Report UCB/CSD 89/537, University of California, Berkeley, Computer Science Division.

[H, 1990a] Herrtwich, R.G. (1990). "Time Capsules: An Abstraction for Access to Continuous-Media Data," *11th Real-Time Systems Symposium*, Orlando, December 1990.

[H, 1990b] Herrtwich, R.G. (1990). "Timed Data Streams in Continuous-Media Systems," Technical Report TR-90-017, International Computer Science Institute, Berkeley, CA.

[HD, 1990] Herrtwich, R.G. and Delgrossi, L. (1990). "ODA-Based Data Modeling in Multimedia Systems," Technical Report TR-90-043, International Computer Science Institute, Berkeley, CA.

[HGA, 1990] Homsy, G., Govindan, R. and Anderson, D.P. (1990). "Implementation Issues for a Network Audio/Video Server," Technical Report UCB/CSD 90/587, University of California, Berkeley, Computer Science Division.

[UO, 1991] Umemura, K. and Okazaki, A. (1991). "Real-Time Transmission and Software Decompression of Digital Video in a Workstation," Technical Report TR-91-004, International Computer Science Institute, Berkeley, CA.

6 Technical Reports 1990

TR-90-001 Stephen M. Omohundro
"The Delaunay Triangulation and Function Learning"

TR-90-002 Chuck Wooters and Nelson Morgan
"Speech Segmentation and Labeling on the NeXT Machine"

TR-90-003 Nelson Morgan
"Considerations for the Electronic Implementation of Artificial Neural Networks"

TR-90-004 Marek Karpinski and Friedhelm Meyer auf der Heide
"On the Complexity of Genuinely Polynomial Computation"

TR-90-005 Dima Yu. Grigoriev, Marek Karpinski, and Michael F. Singer
"Interpolation of Sparse Rational Functions Knowing Bounds on Exponents"

TR-90-006 David P. Anderson, Ralf Guido Herrtwich, and Carl Schaefer
"A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet"

TR-90-007 Domenico Ferrari
"Client Requirements for Real-Time Communication Services"

TR-90-008 Hans W. Guesgen and Peter B. Ladkin
"An Algebraic Approach to General Boolean Constraint Problems"

TR-90-009 Jerome A. Feldman, George Lakoff, Andreas Stolcke & Susan Hollbach Weber
"Miniature Language Acquisition: A Touchstone for Cognitive Science"

TR-90-010 Susan Hollbach Weber
"L_0: A Testbed for Miniature Language Acquisition"

TR-90-011 Subutai Ahmad and Stephen Omohundro
"A Connectionist System for Extracting the Locations of Point Clusters"

TR-90-012 Steffen Hölldobler
"A Connectionist Unification Algorithm"

TR-90-013 Richard Cleve
"Towards Optimal Simulations of Formulas by Bounded-Width Programs"

TR-90-014 Hans Werner Güsgen and Joachim Hertzberg
"Dynamic Constraints"

TR-90-015 Andreas Stolcke
"Learning Feature-based Semantics with Simple Recurrent Networks"

TR-90-016 Christian Freksa
"Temporal Reasoning Based on Semi-Intervals"

TR-90-017 Ralf Guido Herrtwich
"Time Dated Streams in Continuous-Media Systems"

TR-90-018 Hans Werner Güsgen
"A Connectionist Approach to Symbolic Constraint Satisfaction"

TR-90-019 Michael D. Hirsch
"Applications of Topology to Lower Bound Estimates in Computer Science"

TR-90-020 Heinz W. Schmidt
"Prototyping and Analysis of Non-Sequential Systems Using Predicate-Event Nets"

TR-90-021 David P. Anderson, Luca Delgrossi, and Ralf Guido Herrtwich
"Structure and Schedulingin Real-Time Protocol Implementations"

TR-90-022 Domenico Ferrari and Dinesh C. Verma
"Buffer Space Allocation for Real-Time Channels in a Packet-Switching Network"

TR-90-025 Marek Karpinski and Michael Luby
"Approximating the Number of Solutions of a GF[2] Polynomial"

TR-90-026 Ralf Guido Herrtwich
"Audio and Video in Distributed Computer Systems: Why and How?"

TR-90-027 Richard Cleve
"Complexity Theoretic Issues Concerning Block Ciphers Related to D.E.S."

TR-90-028 Peter B. Ladkin, Frank D. Anger, and Rita V. Rodriguez
"Temporal Resoning with Intervals in Branching Time"

TR-90-029 Peter B. Ladkin and Judith S. Crow
"On Location: Points About Regions"

TR-90-030 Paul Dagum
"On the Magnification of Exchange Graphs with Applications to Enumeration Problems"
(Thesis)

TR-90-031 Carlo H. Sequin and Reed D. Clay
"Fault Tolerance in Feed-forward Artificial Neural Networks"

TR-90-032 Richard Cleve and Michael Luby
"A Note on Self-Testing/Correcting Methods for Trigonometric Functions"

TR-90-033 Marek Karpinski and Andrzej Ehrenfeucht
"The Computational Complexity of (XOR, AND)-Counting Problems"

TR-90-034 Pietro Perona
"Finite Representations of Deformable Functions"

TR-90-035 Ralf Guido Herrtwich
"An Introduction to Real-Time Scheduling"

TR-90-036 Lenore Blum
"The Gödel Incompleteness Theorem and Decidability over a Ring"

TR-90-037 Sandy Irani
"Two Results on the List Update Problem"

TR-90-038 J.F. Traub and H. Wozniakowski
"Information-Based Complexity: New Questions for Mathematicians"

TR-90-039 J. F. Traub and H. Wozniakowski
"The Monte Carlo Algorithm with a Pseudo-Random Generator"

TR-90-040 Ronitt Rubinfeld
"Designing Checkers for Programs that Run in Parallel"

TR-90-041 Manuel Blum, Michael Luby, and Ronitt Rubinfeld
"Self-Testing/Correcting with Applications to Numerical Problems"

TR-90-042 Steffen Hölldobler
"CHCL - A Connectionist Inference System for Horn Logic based on the Connection Method and using Limited Resources"

TR-90-043 Ralf Guido Herrtwich and Luca Delgrossi
"ODA-Based Data Modeling in Multimedia Systems"

TR-90-044 N. Morgan, C. Wooters, H. Bourlard, and M. Cohen
"Continuous Speech Recognition on the Resource Management Database Using Connectionist Probability Estimation"

TR-90-045 N. Morgan, H. Hermansky, and C. Wooters (editors)
"SPOONS '90: The SPeech recOgnition frOnt eNd workShop"

TR-90-046 Gabriel Cristobal
"Space-Frequency Distributions in the Analysis and Modeling of Early Vision"

TR-90-047 Nelson Morgan
"The Ring Array Processor (RAP): Algorithms and Architecture"

TR-90-048 James Beck
"The Ring Array Processor (RAP): Hardware"

TR-90-049 Phil Kohn and Jeff Bilmes
"Ring Array Processor (RAP): Software User's Manual Version 1.0"

TR-90-050 Jeff Bilmes and Phil Kohn
"Ring Array Processor (RAP): Software Architecture"

TR-90-051 Riccardo Gusella
"Characterizing the Variability of Arrival Processes with Indices of Dispersion"

TR-90-052 Thomas Lickteig
"On Semi-Algebraic Decision Complexity"

TR-90-053 Joachim Beer
"A Pipelining Model Which Pipelines Blocks of Code"

TR-90-054 Ronitt Rubinfeld
"A Mathematical Theory of Self-Checking, Self-Testing and Self-Correcting Programs" (Thesis)

TR-90-055 Heinz W. Schmidt
"ICSIM: Initial Design of An Object-Oriented Net Simulator"

TR-90-056 Wolfgang Maass and Gyoergy Turan
"How Fast Can A Threshold Gate Learn?"

TR-90-057 Terry Regier
"Learning Spatial Terms without Explicit Negative Instances"

TR-90-058 Lenore Blum
"A Theory of Computation and Complexity over the Real Numbers"

TR-90-059 Peter B. Ladkin
"Constraint Reasoning With Intervals: A Tutorial, Survey and Bibliography"

TR-90-060 Peter Ladkin and Friedrich Vogt (Editors)
"Proceedings of the Berkeley Workshop on Temporal and Real-Time Specification, August 9-10, 1990"

TR-90-061 Jun Zhang and Nelson Morgan
"Stochastic Model-Based Image Segmentation Using Markov Random Fields and Multi-layer Perceptrons"

TR-90-062 "Proceedings of the First International Workshop on Network Operating System Support for Digital Audio and Video"
Held at the International Computer Science Institute, November 8-9, 1990

TR-90-063 N. Karmarkar, R. Karp, R. Lipton, L. Lovász, and M. Luby
"A Monte-Carlo Algorithm for Estimating the Permanent"

TR-90-064 Domenico Ferrari and Dinesh C. Verma
"Quality of Service in ATM Networks"

TR-90-065 Nelson Morgan, Krste Asanovic, Brian Kingsbury
"Developments in Digital VLSI Design for Artifical Neural Networks"

TR-90-066 Wolf Zimmermann
"Automatic Worst Case Complexity Analysis of Parallel Programs"

7 Talks at ICSI 1990

01/09 Barbara Simons
IBM Almaden Research Center
"A Foundation for Sequentializing Parallel Code"

01/16 Christian Freksa
Technische Universität München
"Representing Fuzziness: Simulation vs. Characterization"

01/25 Fan R. Chung
Bellcore, Morristown, NJ
"Quasi-Random Graphs and Communication Complexity"

01/25 Prabhakar Raghavan
IBM Research, Yorktown Heights, NY
"Random Walks, Electric Networks and Algorithmic Applications"

01/25 Paul Beame
University of Washington
"Low Overhead Parallel Schedules for Task Graphs"

02/09 Oliver Gunther
FAW Ulm, Germany
"Hierarchical Representation Schemes for Curved Objects"

02/13 Hartmut Noltemeier
Universität Würzburg
"Dynamic Partial Orders and Generalized Heaps"

02/13 Simone Rehm and Michael Ranft
GMD-Karlsruhe and Universität Karlsruhe
"The STONE Project"

02/13 Larry Ruzzo
University of Washington
"The Power of Parallel Pointer Manipulation"

02/14 Geoffrey Hinton
University of Toronto
"Representing Spatial Structures in Neural Networks"

02/15 Thomas Lickteig
ICSI and Universität Tübingen
"Approximative Complexity – Numerical Computation – Complexity of Tests"

02/15 Christos Papadimitriou
UC San Diego
"On Local Search, Bimatrix Games and Other Total Functions"

02/20 Michael Kearns
Laboratory for Computer Science, M.I.T.
"Distribution-free Learning of Probabilistic Concepts"

02/22 David Chapman
Massachusetts Institute of Technology
"Instruction Use, Vision and Activity"

03/01 Domenico Parisi
Institute of Psychology, National Research Council, Rome
"ECONETS: Neural Networks that Learn in an Environment"

03/02 Daniel Lehmann
Hebrew University, Jerusalem
"Learning Temporal Sequences by Local Synaptic Changes"

03/06 Fabrizio Luccio
Universitá di Pisa, Italy
"Computational Models Based on a Pipelined Access to Memory"

03/13 Howard P. Katseff
AT&T Bell Laboratories, Holmdel, NJ
"Communications-Intensive Workstations"

03/15 Ingo Beinlich
Medical Information Science, Stanford University
"Probabilistic Networks - Fast Algorithms and Applications"

03/15 Vasant Honovar
University of Wisconsin-Madison
"Generalized Connectionist Network Structures and Processes for Intelligent Systems"

03/16 Danny Soroker
IBM Almaden Research Center
"Synchronization and Fault Tolerance in Massively Parallel Computers"

03/22 Paul M. B. Vitanyi
Universiteit van Amsterdam, Netherlands
"Two Applications of the Universal Distribution"

03/22 Martin Dyer
University of Leeds, UK
"Computing the Volume: A Survey"

03/23 Brigitte Vallee
Université de Caen, France
"Generation of Elements with Small Modular Squares and Provably Fast Integer Factoring Algorithm"

03/26 Stefan Wrobel
GMD, Germany
"Observational Learning and Representation Adjustment in MOBAL-0"

03/27 Joachim von zur Gathen
University of Toronto
"Fast Parallel Exponentiation"

03/27 Dana Angluin
Yale University
"Learning with Queries"

03/29 Risto Miikkulainen
Artificial Intelligence Laboratory, UCLA
"A Neural Network Model of Script Processing and Memory"

03/29 Bart Selman
University of Toronto
"Tractable Commonsense Reasoning Using Defaults"

04/02 G. Dreyfus
École Superieure de Physique et de Chimie Industrielles de la Ville de Paris
"An Overview of Neural Network Activities at ESPCI, Paris"

04/02 Ethem Alpaydin
Lab. de Microinformatique - EPFL, Lausanne, Switzerland
"A Connectionist Model of Optical Character Recognition or Reading Through Time"

04/03 Thomas Laussermair
Siemens and Technische Universität München
"Nonlinear Iterated Transformations: A New Way to Cope with Local Optima in Genetic and Connectionist Search"

04/09 Marcus Frean
Department for Physics and Centre for Cognitive Science,
Edinburgh, Scotland
"Feed-forward Neural Networks: the Upstart Algorithm"

04/16 Gerhard Lakemeyer
University of Toronto
"Decidable Reasoning in First-Order Knowledge Bases with Full Introspection"

04/24 Steve Smale
UC Berkeley
"On the Problem of Cook, 'Is P not equal to NP' "

04/25 Ralf Guido Herrtwich
Technische Universität Berlin and ICSI
"Real-Time Requirements in Distributed Continuous-Media Systems"

05/01 Anatole M. Vershik
Leningrad State University
"A Unified Approach to Combinatorial Optimization and Complexity Theory"

05/01 Burghard Rieger
Universität Trier, Germany
"Reconstructing Meaning from Texts - A Computational View on Natural Language Understanding"

05/02 Dimitris Karagiannis
FAW Ulm, Germany
"Office Automation - Flexible Office Systems (FOS)"

05/04 Craig Partridge
Harvard University and Bolt, Beranek and Newman
"Redesigning Remote Procedure Call (RPC) for Gigabit per Second Networks"

05/08 Amitabha Mukerjee
Texas A&M University
"Qualitative Geometric Models"

05/11 John Lazzaro
California Institute of Technology
"Silicon Models of Early Audition"

05/16 Robert Gabriel
GMD-Karlsruhe, Germany
"A Uniform Approach to Formal Program Development"

05/24 Yann Le Cun
AT&T Bell Laboratories, Holmdel, NJ
"Unconstrained Digit Recognition with Constrained Networks"

05/24 Ramesh Subramonian
UC Davis
"Asynchronous Algorithms for Shared Memory Parallel Computers"

05/29 Mark Fanty
Oregon Graduate Center
"Speaker–Independent Recognition of Spoken English Letters"

05/31 Keith Stenning
Edinburgh University, UK
"Establishing Bindings in Human Working Memory: Some Experimental Analyse"

06/05 Robert Restrick
AT&T Bell Laboratories
"The XUNET–II Queue Module"

06/05 Jürgen Schmidhuber
Technische Universität München
"An On–Line Algorithm for Dynamic Reinforcement Learning and Planning in Non–Stationary Reactive Environments"

06/07 Trent E. Lange
University of California at Los Angeles
"High–Level Inferencing in a Structured Connectionist Network: What ROBIN Can Do Now, and What are the Next Steps"

06/18 Nigel H. Goddard
Robotics Institute, Carnegie Mellon University
"The Interpretation of Visual Motion: Recognizing Moving Light Displays"

06/22 Christof Koch
California Institute of Technology
"Towards a Neurobiological Theory of Consciousness"

07/02 Anupindi Suresh
Cambridge University, United Kingdom
"Supporting Dynamic Multimedia Documents in a Distributed Environment"

07/03 Dana S. Scott
Carnegie Mellon University
"Teaching with Mathematica"

07/09 Michael Swain
"Color Indexing"

07/10 Steven Rudich, Carnegie Mellon University,
Jim Aspnes and Richard Beigel, Merrick Furst
"The Expressive Power of Voting Polynomials"

07/10 Toru Ueda
Oregon Graduate Institute
"A Speaker Dependent Japanese Speech Recognition System and a Japanese Character Recognition System Using Neural Networks"

07/17 Vincenzo Cutello, Courant Institute of Mathematical Sciences, NYU
Domenico Cantone, Archimedes-Catania and University of Catania
"Computable Set Theory"

07/24 Feng Gao
University of British Columbia
"Towards Structured Parallel Computing — On Architecture–Independent Algorithm Design"

07/26 Adi Shamir
The Weizmann Institute of Science, Israel
"Multiple Non-Interactive Zero Knowledge Proofs"

07/31 Adi Shamir
The Weizmann Institute of Science, Israel
"IP=PSPACE"

08/02 Gary Miller
Carnegie Mellon University
"Separators for Graphs that Arise in Computation Geometry and Numerical Analysis"

08/07 Eli Biham
The Weizmann Institute, Israel
"Differential Cryptanalysis of DES-like Cryptosystems"

08/07 Gilles Brassard
Université de Montréal
"Experimental Quantum Cryptography"

08/08 Amos Fiat
Tel Aviv University, Israel
"Competitive k-Server Algorithms"

08/14 Chistopher Cherniak
Institute for Advanced Computer Studies, University of Maryland
"Component Placement Optimization in the Brain"

08/14 Richard Karp
UC Berkeley and ICSI
"A Graph–Theoretic Game and its Application to the k–Server Problem"

08/21 Carlo H. Séquin
UC Berkeley
"Fault Tolerance in Feed–Forward Artificial Neural Networks"

08/22 Gerd Sandweg
Siemens Corporate Research and Development, München, Germany
"Parallel Computation Research and Architecture of Intelligent Systems"

08/22 Joel Friedman
Princeton University
"The Spectra of the Infinite Hypertree"

08/23 Dorit S. Hochbaum
UC Berkeley
"Is P = SP?"

08/24 Hans-Werner Hein
Universität Dortmund, Germany
"Machine Intelligence Research at the University of Dortmund"

08/30 Paul R. Cooper
Northwestern University
"Recognition with Uncertainty: Coupled Segmentation and Matching"

09/05 Azaria Paz
Technion, Haifa, Israel
"The Theory of Graphoids: A Survey"

09/06 Azaria Paz
Technion, Haifa, Israel
"Axioms and Algorithms for Inferences involving Probabilistic Independence"

09/11 Per Stenstrom
Lund University, Sweden
"Hardware Schemes to Preserve Cache Coherency and Access Order in Shared Memory Multiprocessors"

09/13 Wolf Zimmermann
Universität Karlsruhe, Germany
"The Automatic Complexity Analysis of Functional Programs"

09/27 Ernst W. Mayr
Universität Frankfurt, Germany
"Polynomial Ideals and Applications"

10/04 Thomas Bemmerl
Technische Universität München
"TOPSYS: An Integrated Tool Environment for
Programming Parallel Systems"

10/11 David Rumelhart
Stanford University
"Regression, Statistics and Back Propagation"

10/15 Thomas Bräunl
Universität Stuttgart
"Transparent Massively Parallel Programming with Parallaxis"

10/18 Eyal Kushilevitz
Technion, Haifa, Israel
"Private Computations over the Integers"

10/24 David H. Wolpert
Theoretical Division and Center for Non–Linear Studies, Los Alamos, NM
"Cross–Validation, Learning Set Transformations, and Generalization"

10/26 Michael Gilge
Institute for Communication Engineering, RWTH Aachen
"A Generalized Orthogonal Transform for Image Coding"

10/30 Ran Canetti
Technion, Haifa, Israel
"Bounds on Tradeoffs between Randomness and Communication Complexity"

11/06 Eric Baum
NEC Research Center, Princeton, NJ
"Neural Net Algorithms that Learn in Polynomial Time from Examples and Queries"

11/07 Wolfgang Maass
University of Illinois at Chicago
"How Fast can a Threshold Gate Learn?"

11/13 Andreas Weigend
Stanford University
"Connectionist Networks with Weight-Elimination for the Time Series Prediction"

11/15 Laszlo Babai
University of Chicago and Eötvös University, Budapest
"Multiple Provers: Technique and Applications"

11/19 Robert J. Hall
MIT AI Laboratory, Cambridge, MA
"Program Improvement by Automatic Redistribution of Intermediate Results"

11/19 James Renegar
Cornell University
"Condition Numbers"

11/27 Michael Luby
ICSI
"On Deterministic Approximation of DNF"

12/13 Baruch Awerbuch
MIT
"Sparse Partitions"