Columbia University in the City of New York
Department of Computer Science
New York, NY 10027

(212) 854-2736
tech-reports@cs.columbia.edu

TECHNICAL REPORTS LIST

The Computer Science Department at Columbia University is pleased to announce the enclosed list of technical reports and abstracts for 1990.

If you wish to order any of these reports, please complete the form below, fold so that the return address is properly positioned, and affix postage.

A NOTE TO INTERNET USERS: many Columbia Computer Science technical reports are available by anonymous FTP from host `cs.columbia.edu` directory `pub/reports`. All reports are stored in postscript format.

Technical Report Numbers

1. ________	6. ________
2. ________	7. ________
3. ________	8. ________
4. ________	9. ________
5. ________	10. ________

Name ________________________________

Address ______________________________

City ____________ State ____________ Zip ______

Attn:Technical Reports Librarian
Department of Computer Science
Columbia University
New York, NY 10027

[CUCS-001-90] Incremental algorithms for minimal length paths.

G. Ausiello, G. F. Italiano, A. Marchetti Spaccamela, and U. Nanni.

We consider the problem of maintaining on-line a solution to the All Pairs Shortest Paths Problem in a directed graph $G = (V, E)$ where edges may be dynamically inserted or have their cost decreased. For the case of integer edge costs in a given range $[1 \ldots C]$, we introduce a new data structure which is able to answer queries concerning the length of the shortest path between any two vertices in constant time, and to trace out the shortest path between any two vertices in time linear in the number of edges reported. The total time required to maintain the data structure under a sequence of at most $O(n^2)$ edge insertions and at most $O(Cn^2)$ edge cost decreases is $O(Cn^3 \log(nC))$ in the worst case, where n is the total number of vertices in G. For the case of unit edge costs, the total time required to maintain the data structure under a sequence of at most $O(n^2)$ insertions of edges becomes $O(n^3 \log n)$ in the worst case. Both data structures can be adapted to solve the problem of maintaining on-line maximal length paths in directed acyclic graphs. All our algorithms improve on previously known algorithms and are only a logarithmic factor away from the best possible bounds.

[CUCS-002-90] Semantics for safe concurrency.

Wenwey Hseush, Timothy S. Balraj, Gail E. Kaiser, and Krish Ponamgi.

We present two systems for modelling safe concurrency. The first is an automata-based system, called *predecessor automata* (PAs). The second is a language-based system, called *Data Path Expressions* (DPEs). In a previous paper [DPE], we presented a hierarchy of DPEs and described PAs as a mechanism for implementing the safe DPEs from the viewpoint of parallel debugging. In this paper, we formalize these two models and investigate the power of PAs and DPEs.

[CUCS-003-90] Constraint-based text generation: Using local constraints and argumentation to generate a turn in conversation.

Michael Elhadad.

This report describes work on the problem of generating a turn within an ongoing conversation. Three main points are advanced: (1) Previous discourse determines the form and content of a new turn; this work proposes techniques to take the influence of previous discourse into account when generating text. (2) The connection between previous discourse and a new turn can be described as the interaction between five local constraints; local constraints are relations between one discourse segment of the previous discourse and the new turn. This multi-dimensional description of discursive relations allows a flexible definition of complex rhetorical relations. (3) Argumentation is one class of local constraints which has important effects on the form of the language produced and interacts closely with the other types of local constraints used in this work. The problem of generating a turn is then posed as the problem of finding a set of local constraints compatible with each other, with the previous discourse, and with the information to convey. The application of the local constraints determines the value of pragmatic features of the new turn. A sophisticated grammar can then translate these decisions into appropriate lexical and syntactic decisions. The approach is applied to the implementation of an explanation module for the ADVISOR expert system. The implementation is based on an extended form of functional unification; this extended formalism is a good choice for handling the problem of constraint interaction.

[CUCS-004-90] An object-oriented approach to content planning for text generation.

Ursula Wolz.

This paper describes Genie, an object-oriented architecture that generates text with the intent of extending user expertise in interactive environments. Instead of generating text based solely on either discourse goals, intentions, or the domain, we found a need to combine techniques from each. We have developed an object- oriented architecture in which the concepts about which we talk (domain entities), the goals that may be accomplished with them (intentions), and the rhetorical acts through which we express them (discourse goals) are represented as objects with localized knowledge and methods. A three stage process of content selection, content structuring and content filtering is presented. Content selection and structuring allow us to produce text that is within the context of the task at hand for the user. Content filtering allows us to revise and restructure the utterance to achieve clarity and conciseness.

[CUCS-005-90] Detecting conflicts in multi–agent rule-based development environments.

Naser S. Barghouti and Gail E. Kaiser.

Researchers have been exploring the possibility of applying production systems to the domain of software development. One resulting approach is rule-based development environments (RBDEs), which provide expert assistance to developers working on large-scale projects. RBDEs model the development process in terms of rules, and then enact this model by automatically firing rules at the appropriate time to carry out chores that the developer would have done manually. In order to realistically model the domain of software development, RBDEs must support the cooperation among multiple developers, each of whom selects commands, causing the firing of multiple rules (either directly or via chaining) that concurrently access shared information. In this domain, rule parallelism is *intrinsic* and not *opportunistic* (i.e., not for the purpose of speeding up the production system). One of the problems resulting from executing rule chains concurrently causes is how to detect conflicts that occur between the chains. This paper presents the multi-agent RBDE model, explores the conflict detection problem, and suggests an approach to solving it. The problem of resolving conflicts without unnecessarily obstructing cooperation among developers is only briefly discussed.

[CUCS-010-90] An incremental priority-based solution to the multiple asynchronous edit problem.

Simon M. Kaplan and Gail E. Kaiser.

We present a new solution to the problem of incrementally updating attributes in the context of multiple asynchronous edits to an attributed parse tree. The major difficulty that arises is that of *merging* the attribute propagations from the various edit points so as to ensure that each attribute is reevaluated the least possible number of times. Our solution uses priority numbers to capture topological order of dependency information among attributes, which allows efficient merging. The worst-case time complexity is $O(k \cdot (\mid \mathit{Influenced} \mid \cdot \log(\mid \mathit{Influenced} \mid)))$, where k is the number of disjoint times at which overlapping edits occur and *Influenced* is the smallest possible set of attribute instances that must be reevaluated during the attribute propagation process in order to reestablish attribute consistency. Best-case complexity is $O(\mid \mathit{Influenced} \mid \cdot \log(\mid \mathit{Influenced} \mid))$, which arises when propagations from edits do not overlap, or all edits occur simultaneously, where *Influenced* is the same as the conventional *Affected* metric.

[CUCS-011-90] Parallel and distributed processing of rules in data intensive applications.

Ouri Wolfson and Aya Ozeri.

This paper is concerned with the parallel evaluation of datalog rule programs, mainly by processors that are interconnected by a communication network. We introduce a paradigm, called data-reduction, for the parallel evaluation of a general datalog program. Several parallelization strategies discussed previously in [CW, GST, W, WS] are special cases of this paradigm. The paradigm parallelizes the evaluation by partitioning among the processors the instantiations of the rules. After presenting the paradigm, we discuss the following issues, that we see fundamental for parallelization strategies derived from the paradigm: properties of the strategies that enable a reduction in the communication overhead, decomposability, load balancing, and application to programs with negation. We prove that decomposability, a concept introduced previously in [WS, CW], is undecidable.

[CUCS-012-90] Decomposability and its role in parallel logic-program evaluation.

Ouri Wolfson and Avi Silberschatz.

This paper is concerned with the issue of parallel evaluation of logic programs. We define the concept of "program decomposability", which means that the load of evaluation can be partitioned among a number of processors, without a need for communication among them. This in turn results in a very significant speed-up of the evaluation process. Some programs are decomposable, whereas others are not. We completely syntactically characterize three classes of single rule programs with respect to decomposability: nonrecursive, simple linear, and simple chain programs. We also establish two sufficient conditions for decomposability.

[CUCS-013-90] Network management with consistently managed objects.

Shyhtsun F. Wu and Gail E. Kaiser.

A *consistency constraint* exists between two objects in a network *management information base (MIB)* if a change in value of one object will cause a change in another. Based on the definition of *common knowledge* by Halpern, Fagin, and Moses, such constraints can be classified by the time needed to maintain them and the probability of correctly maintaining them. In this paper, we study the practical issues of how to build an MIB with a set of objects and consistency constraints. We will go through an experimental MIB example showing that distributed network entities can be consistently managed in an elegant way, even when the network environment is very dynamic.

[CUCS-014-90] Redundancy management in a symmetric main memory database.

Calton Pu, Avraham Leff, Frederick Korz, and Shu-Wie Chen.

We describe the architecture of a symmetric distributed main memory database. The high performance networks in many large distributed systems enable a machine to reach the main memory of other nodes faster than local disks. Thus we introduce *remote memory* as an additional layer in the memory hierarchy between local memory and disks. Exploiting the remote memory (every node's cache) improves performance and increases availability. We have created a simulation program that shows this significant advantage over a wide range of cache sizes and compares the effects of several object replacement policies in the symmetric distributed main memory database.

[CUCS-015-90] Why a single parallelization strategy is not enough in knowledge bases.

Simona Rabinovici Cohen and Ouri Wolfson.

We address the problem of parallelizing the evaluation of logic programs in data intensive applications. We argue that the appropriate parallelization strategy for logic-program evaluation depends on the program being evaluated. Therefore, this paper is concerned with the issues of program-classification, and parallelization-strategies. We propose several parallelization strategies based on the concept of data-reduction – the original logic-program is evaluated by several processors working in parallel, each using only a subset of the database. The strategies differ on the evaluation cost, the overhead of communication and synchronization among processors, and the programs to which they are applicable. In particular, we start our study with pure-parallelization, i.e., parallelization without overhead. An interesting class-structure of logic programs is demonstrated, when considering amenability to pure-parallelization. The relationship to the NC complexity class is demonstrated. Then we propose strategies that do incur an overhead, but are optimal in a sense that will be precisely defined. This paper makes the initial steps towards a theory of parallel logic-programming.

[CUCS-017-90] An object-oriented framework for modeling cooperation in multi-agent rule-based development environments.

N. S. Barghouti and G. E. Kaiser.

Rule-Based Development Environments (RBDEs) exploit expert systems via rule-based modeling of the software development process. RBDEs store the software artifacts in a project database, and define each software development activity that manipulates these artifacts as a rule. Opportunistic forward and backward chaining on the rules automates some of chores that developers would have otherwise done manually and ensures consistency in the database. Existing RBDEs do not scale up to real software development projects involving large teams of developers, in part because the current expert system paradigm does not handle multiple agents sharing the working memory of the rule system. In this article, we investigate the scaling up of RBDEs through the use of an object-oriented framework that integrates the structural and behavioral aspects of data modeling and rule-based process modeling. We use this framework to support both consistency maintenance and cooperation in multi-agent RBDEs.

[CUCS-018-90] Interfacing cooperative transactions to software development environments.

Gail E. Kaiser.

Many cooperative transaction models are intended to support teams of programmers sharing a software development environment. However, papers on this topic often neglect to describe how the programmers might actually interact with the cooperative transactions in terms of user commands and corresponding responses from the environment. In this paper, we consider a particular cooperative transaction model, called participant transactions, and explain the user model that needs to be implemented by a software development environment.

[CUCS-019-90] Parallel and distributed incremental attribute evaluation algorithms for multi-user software development environments.

Gail E. Kaiser and Simon M. Kaplan.

[CUCS-020-90] Multiple concurrency control policies in an object-oriented programming system.

Gail E. Kaiser, Wenwey Hseush, Steven S. Popovich, and Shyhtsun F. Wu.

Object-oriented systems have been generally recognized as a good choice for approaching a range of parallel and distributed programming problems. But different parallel and distributed applications have different consistency models, so *multiple concurrency control policies* are needed. When objects are shared among applications with different policies, multiple policies must operate simultaneously and compatibly. We have investigated two consistency models, atomic blocks and classical serializable transactions, and the interactions both between these models and with "uncontrolled" concurrency, in the context of the Meld distributed object-oriented programming language. We present the relevant language constructs and discuss certain programming problems and implementation issues.

[CUCS-021-90] A formalization and implementation of topological visual navigation in two dimensions.

John R. Kender, Il-Pyung Park, and David Yang.

In this paper we formalize and implement a model of topological visual navigation in two-dimensional spaces. Unlike much of traditional quantitative visual navigation, the emphasis throughout is on the methods and the efficiency of qualitative visual descriptions of objects and environments, and on the methods and the efficiency of direction-giving by means of visual landmarks. We formalize three domains–the world itself, the map-maker's view of it, and the navigator's experience of it–and the concepts of custom maps and landmarks. We specify, for a simplified navigator (the "level helicopter") the several ways in which visual landmarks can be chosen, depending on which of several costs (sensor, distance, or communication) should be minimized. We show that paths minimizing one measure can make others arbitrarily complex; the algorithm for selecting the path is based on a form of Dijkstra's algorithm, and therefore automatically generates intelligent navigator overshooting and backtracking. We implement, using an arm-held camera, such a navigator, and detail its basic seek-and-adjust behaviors as it follows visual highways (or departs from them) to reach a goal. Seeking is based on topology, and adjusting is based on symmetry; there are essentially no quantitative measures. We describe under what circumstances its environment is visually difficult and perceptively shadowed, and describe how errors in path-following impact landmark selection. Since visual landmark selection and direction-giving are in general NP-complete, and rely on the nearly intractable concept of characteristic views, we suggest some heuristics; one is that the landmark object "itself", rather than its views, may be its most compact encoding. We conclude with speculations about the feasibility of intelligent navigation in very large self-occluding visual worlds.

[CUCS-023-90] Parallel least-squares solution of general and toeplitz-like linear systems.

Victor Pan.

We use $O(\log^2 n)$ parallel arithmetic steps and n^2 processors to compute the least-squares solution $\mathbf{x} = A^+\mathbf{b}$ to a linear system of equations, $A\mathbf{x} = \mathbf{b}$, given a $g \times h$ matrix A and a vector $\mathbf{b}$, both filled with integers or with rational numbers, provided that $g + h \leq 2n$ and that A is given with its displacement generator of length $r = O(1)$ and thus has displacement rank $O(1)$. For a vector $\mathbf{b}$ and for a general $p \times q$ matrix A (with $p + q \leq n$), we compute A^+ and $A^+\mathbf{b}$ by using $O(\log^2 n)$ parallel arithmetic steps and $n^{2.851}$ processors, and we may also compute $A^+\mathbf{b}$ by using $\mathrm{O}(n^{2.376})$ arithmetic operations.

[CUCS-024-90] On the evaluation of the eigenvalues of a banded toeplitz block matrix.

Dario Bini and Victor Pan.

Let $A = (a_{ij})$ be an $n \times n$ Toeplitz matrix with bandwidth $k+1$, $k = r + s$, that is, $a_{ij} = a_{j-i}$, $i, j = 1, \ldots, n$, $a_i = 0$ if $i > s$ and if $i < -r$. We compute $p(\lambda) = \det(A - \lambda I)$, as well as $p(\lambda)/p'(\lambda)$, where $p'(\lambda)$ is the first derivative of $p(\lambda)$, by using $O(k \log k \log n)$ arithmetic operations. Moreover, if a_i are $m \times m$ matrices, so that A is a banded Toeplitz block matrix, then we compute $p(\lambda)$, as well as $p(\lambda)/p'(\lambda)$, by using $O(m^3 k(\log^2 k + \log n) + m^2 k \log k \log n)$ arithmetic operations. The algorithms can be extended to the computation of $\det(A - \lambda B)$ and of its first derivative, where both A and B are banded Toeplitz matrices. The algorithms may be used as a basis for iterative solution of the eigenvalue problem for the matrix A and of the generalized eigenvalue problem for A and B.

[CUCS-025-90] Computing matrix eigenvalues and polynomial zeros where the output is real.

Dario Bini and Victor Pan.

Surprisingly simple corollaries from the Courant-Fischer minimax characterization theorem enable us to devise a very effective algorithm for the evaluation of a set S interleaving the set E of the eigenvalues of a real symmetric tridiagonal matrix T_n (as well as a point that splits E into two subsets of comparable cardinality). This is similar to the more complicated algorithms of [BOT] and [BFKT] that compute such interleaving sets and splitting points for the set of the zeros of a polynomial $p(\lambda)$ having only real zeros. As a result, the record upper estimates for the parallel and sequential complexity of approximating to all the zeros of such a polynomial are obtained in an alternate way and, moreover, are extended to the eigenvalue computation for T_n. Furthermore, we dramatically decreased the previous record upper estimates for the parallel complexity of the latter problem, to the level within polylogarithmic factors from the straightforward lower bounds.

[CUCS-026-90] On a recursive triangular factorization of matrices.

Victor Pan.

We estimate the parallel and sequential complexity of computing a recursive triangular factorization of an $n \times n$ matrix, which may, in particular, replace Choleski factorization as a means of computing the QR-factorization of a matrix and may also have some other applications. The estimates for the storage space, sequential time and the number of processors required for the computation and storage of such a recursive factorization is by almost the factor of n less than the similar estimates for the Choleski factorization in the case of Toeplitz-like and Hankel-like matrices.

[CUCS-027-90] A framework for immibrating existing software into new software development environements.

Michael H. Sokolsky and Gail E. Kaiser.

We have investigated the problem of immigrating software artifacts from one software development environment (SDE) to another for the purpose of upgrading to new SDEs as technology improves, while continuing development or maintenance of existing software systems. We first taxonomize the larger problem of data migration, to establish the scope of immigration. We then classify SDEs in terms of the ease of immigrating software artifacts out of the data repository of the source SDE without knowledge of its internal representation. A framework is presented for constructing automatic immigration tools as utilities provided by destination SDEs. We describe a specific immigration tool, called Marvelizer, that we have implemented as part of the Marvel SDE and discuss our experience using the tool.

[CUCS-029-90] Improved parallel polynomial division.

Dario Bini and Victor Pan.

We compute the first N coefficients of the reciprocal $r(x)$, $r(x)p(x) = 1 \bmod x^N$ (given a natural N and a polynomial $p(x)$), $(p(0) \neq 0)$, by using $O(h \log N)$ arithmetic steps and $O\big((N/h)(1 + 2^{-h} \log^{(h)} N)\big)$ processors, for any h, $h = 1, 2, \ldots, \log^* N$, under the PRAM arithmetic models, provided that $O(\log N)$ steps and N processors suffice to perform DFT at N points and that $\log^{(0)} N = N$, $\log^{(h)} N = \log_2 \log^{(h-1)} N$, $h = 1, \ldots, \log^* N$, $\log^* N = \min\{h : \log^{(h)} N \leq 1\}$. The same estimates apply to several other computations such as the division with a remainder of two polynomials of degrees $O(N)$, the inversion of an $N \times N$ triangular Toeplitz matrix, and the evaluation modulo x^N of the m-root $p(x)^{1/m}$ of $p(x)$ for any fixed natural m. This improves the known estimates.

[CUCS-030-90] Approximate polynomial division with a remainder by means of evaluation and interpolation.

Victor Pan, Elliot Landowne, and Akimou Sadikou.

We apply the evaluation-interpolation techniques of A. Toom to approximate polynomial division with a non-vanishing remainder. Surprisingly, this elemetary technique immediately leads us to the same computations and to the same record parallel complexity estimates for this problem, $O_A(\log m, m)$, as do the far more sophisticated algorithms of D. Bini and A. Schönhage, and our approach enables us to simplify the error analysis too.

[CUCS-031-90] Complexity of computations with matrices and polynomials.

Victor Pan.

We review the complexity of polynomial and matrix computations, as well as their various correlations to each other and some major techniques for the design of algebraic and numerical algorithms.

[CUCS-032-90] Parametrization of newton's iteration for computations with structured matrices and applications.

Victor Pan.

We apply a new parametrized version of Newton's iteration in order to compute (over any field F of constants) the solution or a least-squares solution to a linear system $Bx = v$ with an nxn Toeplitz or Topelitz-like matrix B, as well as the determinant of B and the coefficients of its characteristic polynomial, $det(\lambda I - B)$, dramatically improving the processor efficiency of the know fast parallel algorithms. Our algorithms, together with some previously know and some resect results, as well as with our new techniques for computing polynomial GCDs and LCMs, imply respective improvement of known estimates for parallel arithmetic complexity of several fundamental computations with polynomials and with both structured and general matrices.

[CUCS-033-90] Multi-user semantics-based editors for computer-supported cooperative work.

Josephine Micallef and Gail E. Kaiser.

We propose a novel direction for structure-based environments: *multi-user* semantics-based editors. Multi-user editors enable collaboration among multiple programmers working together in a group. Since semantics-based editors can also be applied to other kinds of structured documents, in addition to programs, multi-user editors seem a practical basis for many areas of computer-supported cooperative work. The focus of this paper is on the algorithms and facilities needed to make multi-user semantics-based editors practical, in particular, to correctly and efficiently carry out incremental semantic processing in response to asynchronous changes made by multiple users. We also describe our preliminary experience with multi-user editors for both programming and non-programming applications.

[CUCS-034-90] On the criteria to be used in composing tools into systems.

David Garlan, Gail E. Kaiser, and David Notkin.

The well-known notion of data abstraction, which separates the interface of a module from its implementation, permits the representation and implementation of abstract data types to evolve without affecting the module's clients. However, many enhancements change the function of a system, not just its implementation. Data abstraction does not directly ease the difficulties and costs of such enhancements. *Tool abstraction* complements data abstraction by supporting evolutionary changes that are made incrementally and independently. This approach structures a system as a pool of shared data structures, viewed by a collection of cooperating tools, each of which provides a piece of the overall function. This article identifies tool abstraction as the common foundation of a variety of existing systems – including spreadsheets, structure-oriented environments, production systems, and object-oriented systems.

[CUCS-035-90] Real-time visual servoing.

Peter Allen, Billibon Yoshimi, and Aleksandar Timcenko.

This paper describes a new real-time tracking algorithm in conjunction with a predictive filter to allow real-time visual servoing of a robotic arm. The system consists of two calibrated cameras that provide images to a real-time, pipelined-parallel optic-flow algorithm that can robustly compute optic-flow and calculate the 3-D position of a moving object at approximately 5 Hz rates. These 3-D positions serve as input to a predictive kinematic control algorithm that uses an $alpha - beta - gamma$ filter to update the position of a robotic arm tracking the moving object. Experimental results are presented for the tracking of a moving model train in a variety of different trajectories.

[CUCS-036-90] Understanding data refinement using procedural refinement.

Steven M. Kearns.

Data refinement is converting a program that uses one set of variables to an equally correct program that uses another set of variables, usually of different types. There have been a number of seemingly different mathematical definitions of data refinement. We present a unifying view of data refinement as a special case of procedural refinement, which is simpler to understand. All the data refinement theories in the literature are shown to be instances of two formulas, but we show that there are actually an infinite number of theories. In addition, we introduce the concepts of nonlocal data refinement, data refinement using semi-invariants, and procedural optimization using data refinement.

[CUCS-037-90] Tlex user manual.

Steven M. Kearns.

The TLex Language, a regular expression-based language for finding and extracting information from text, is described. TLex features a pattern language that includes intersection, complement, and context sensitive operators. In addition, TLex automatically creates a parse tree from a successful match and offers convenient functions for manipulating it.

[CUCS-039-90] Modeling safe and regular concurrent processes.

Wenwey Hseush, Timothy S. Balraj, and Gail E. Kaiser.

The authors have previously described the use of data path expressions and predecessor automata in debugging concurrent systems. In this paper we examine the relationship of these models to two traditional models of concurrent processes: pomset languages and k-safe Petri net systems. We explore the regularity and safety of the concurrent languages described by each of the four models. Our main result is the equivalence of regular safe pomset languages and the languages described by safe data path expressions, safe predecessor automata and k-safe Petri net systems.

[CUCS-040-90] Lexical choice in natural language generation.

Jacques Robin.

In this paper we survey the issue of lexical choice in natural language generation. We first define lexical choice as the choice of *open-class* lexical items (whether phrasal patterns or individual words) appropriate to express the content units of the utterance to generate in a given situation of enunciation. We then distinguish between five major classes of constraints on lexical choice: grammatical, inter-lexical, encyclopedic, interpersonal and discursive. For each class we review how they have been represented and used in existing generators. We conclude by pointing out the common limitations of these generators with respect to lexical choice as well as some directions for future research in the field.

[CUCS-041-90] Testing object-oriented programs by mutually suspicious parties.

Travis Lee Winfrey.

Testing object-oriented programs has been studied primarily in terms of paradigms that apply to all programs, i.e., white-box and black-box testing. We describe a new testing method for object-oriented programs that specifically exploits encapsulation properties. Individual object classes or even individual instances of objects may be instrumented for testing. At the heart of the method is the systematic renaming and duplication of object classes.

[CUCS-044-90] A methodology for measuring applications over the internet.

Calton Pu, Frederick M. Korz, and Robert C. Lehman.

Measuring the performance of applications running over live wide area networks such as the Internet is difficult. Many of the important variables (path, network load, etc) cannot be controlled. Monitoring the entire Internet to follow individual messages is impractical. We developed the Layered Refinement methodology to meet these challenges. Our method uses generic software tools to collect data simultaneously at important software layers. Data are analyzed and refined to reduce variance, and correlated to give us a good understanding of each layer. Sample data spanning seven days illustrate the method, which narrows the confidence interval by an order of magnitude to show the end-to-end performance of applications that run over Internet.

[CUCS-046-90] An object-based programming model for shared data.

Gail E. Kaiser and Brent Hailpern.

The classical object model supports private data within objects and clean interfaces between objects, and by definition does not permit sharing of data among arbitrary objects. This is a problem for real-world applications — such as advanced financial services and integrated network management — where the same data logically belongs to multiple objects and may be distributed over multiple nodes on the network. Rather than give up the advantages of encapsulated objects in modeling real-world entities, we propose a new object model that supports shared data in a distributed environment. The key is separating distribution of computation units from information hiding concerns. Minimal units of data and control, called *facets*, may be shared among multiple *objects* and are grouped into *processes*. Thus a single object, or information hiding unit, may be distributed among multiple processes, or computation units. In other words, different facets of the same object may reside in different address spaces on different machines. We introduce our new object model, describe a motivating example from the financial domain, and then explain facets, objects and processes, followed by timing and synchronization concerns.

[CUCS-047-90] Scaling up rule-based development environments.

Naser S. Barghouti and Gail E. Kaiser.

Rule-based development environments (RBDEs) model the software development process in terms of rules that encapsulate development activities, and execute forward and backward chaining on the rules to provide assistance in carrying out the development process. We investigate the scaling up of RBDEs along two dimensions. The first dimension covers the nature of the assistance provided by the RBDE, and the second spans the functionality. There is a spectrum of assistance models implemented by RBDEs. We consider three models: pure automation, strict consistency preservation, and a maximalist assistance model that integrates consistency and automation. The choice of assistance model impacts the functionality of the RBDE, more precisely the solutions to three problems that arise in scaling up RBDEs: (1) multiple views; (2) evolution; and (3) concurrency control. We explore the solutions to the three problems with respect to each of the three assistance models. Throughout the paper, we use the Marvel RBDE as an example.

[CUCS-048-90] Dynamic reconfiguration in an object-based programming language with distributed shared data.

Brent Hailpern and Gail E. Kaiser.

On-line distributed applications generally allow reconfiguration while the application is running, but changes are usually limited to adding new client and server processes and changing the bindings among such processes. In some application domains, such as on-line financial services, it is necessary to support finer grained reconfiguration at the level of entities within processes, but for performance reasons it is desirable to avoid conventional approaches such as interpretation and dynamic storage allocation. We present a scheme for special cases of fine grained dynamic reconfiguration sufficient for our application domain and show how it can be used for practical changes. We introduce new language concepts to implement this scheme in the context of an object-based programming language that supports shared data in a distributed environment.

[CUCS-049-90] An information retrieval approach for automatically constructing software libraries.

Yoelle S. Maarek, Daniel M. Berry, and Gail E. Kaiser.

Although software reuse presents clear advantages for programmer productivity and code reliability, it is not practiced enough. One of the reasons for the only moderate success of reuse is the lack of software libraries that facilitate the actual locating and understanding of reusable components. This paper describes a technology for automatically assembling large software libraries that promote software reuse by helping the user locate the components closest to her/his needs. Software libraries are automatically assembled from a set of unorganized components by using information retrieval techniques. The construction of the library is done in two steps. First, attributes are automatically extracted from natural language documentation by using a new indexing scheme based on the notions of lexical affinities and quantity of information. Then, a hierarchy for browsing is automatically generated using a clustering technique that draws only on the information provided by the attributes. Thanks to the free-text indexing scheme, tools following this approach can accept free-style natural language queries. This technology has been implemented in the GURU system, which has been applied to construct an organized library of AIX utilities. An experiment was conducted in order to evaluate the retrieval effectiveness of GURU as compared to INFOEXPLORER a hypertext library system for AIX 3 on the IBM RISC System/6000 series. We followed the usual evaluation procedure used in information retrieval, based upon recall and precision measures, and determined that our system performs 15% better on a random test set, while being much less expensive to build than INFOEXPLORER.

[CUCS-050-90] Detection and exploitation of file working sets.

Carl Tait and Dan Duchamp.

The work habits of most individuals yield file access patterns that are quite pronounced and can be regarded as defining working sets of files used for particular applications. This paper describes a client-side cache management technique for detecting these patterns and then exploiting them to successfully prefetch files from servers. Trace-driven simulations show the technique substantially increases the hit rate of a client file cache in an environment in which a client workstation is dedicated to a single user. Successful file prefetching carries three major advantages: (1) applications run faster, (2) there is less "burst" load placed on the network, and (3) properly-loaded client caches can better survive network outages. Our technique requires little extra code, and DASH because it is simply an augmentation of the standard LRU client cache management algorithm DASH is easily incorporated into existing software.

[CUCS-051-90] An object-based approach to implementing distributed concurrency control.

Steven S. Popovich, Gail E. Kaiser, and S.F. Wu.

We have added distributed concurrency control to the MELD object system by representing in-progress transactions as simulated objects. *Transaction objects* exploit MELD's normal message passing facilities to support the concurrency control mechanism. We have completed the implementation of an optimistic mechanism using transaction objects and have designed a two-phase locking mechanism based on the same paradigm. We discuss the tradeoffs made and lessons learned, dealing both with transactions *on* objects and with transactions *as* objects.

[CUCS-052-90] Mortgage pool allocation by simulated annealing.

Eugene Pinsky, Paul Fahn, and Yechiam Yemini.

Many optimization problems in finance are known to be computationally complex. In this paper we consider one such problem of Mortgage Pool Allocation. In this problem one needs to allocate a set of mortgage-backed securities to fill forward contracts in real-time. The number of securities and contracts is typically large and there are numerous constraints on allocations in the form of rules published by the Public Securities Association (PSA). The problem is to find the most profitable allocation that satisfies all of the PSA's constraints. Simulated annealing is a recently explored probabilistic algorithm, used most commonly to solve simply-formulated combinatorial optimization problems with a small number of constraints. This paper shows successful application of simulated annealing to the problem of mortgage pool allocation: it finds good (highly profitable) solutions very quickly. This result is significant both for the complexity of the problem solved and because it demonstrates the feasibility of simulated annealing for solving real-world financial problems.

[CUCS-053-90] Applications of epsilon-serializability in federated databases.

Calton Pu and Avraham Leff.

We apply the concept of *epsilon-serializability* (ESR) to data replication. ESR converges to global serializability when the replicated system reaches a quiescent state, i.e., all the inter-site messages arrive their destination. ESR integrates many previous proposals for increasing availability for replicated data and bring new insights.

[CUCS-054-90] Epsilon-serializability.

Calton Pu and Avraham Leff.

We introduce and formalize the concept of espsilon-serializability (ESR). ESR converges to serializability (SR) when the database reachs a quiescent state, i.e., all the update messages are processed. Divergence control methods make ESR practical by restricting the temporary inconsistency in the database. ESR is useful since it allows more concurrency than SR and can be integrated with SR.

[CUCS-057-90] A distributed algorithm for adaptive replication of data.

Ouri Wolfson.

We present a distributed algorithm for replication of a data-item in a set of processors interconnected by a tree network. The algorithm is adaptive in the sense that the replication scheme of the item (i.e. the set of processors, each of which stores a replica of the data-item), changes as the read-write pattern of the processors in the network changes. The algorithm is optimal in the sense that when the replication scheme stabilizes, the total number of messages required for the reads and writes is minimal. The algorithm is extremely simple.

[CUCS-058-90] Incremental evaluation of rules and its relationship to parallelism.

Ouri Wolfson, Hasanat Dewan, Salvatore Stolfo, and Yechiam Yemini.

Rule interpreters usually start with an initial database and perform the inference procedure in cycles, ending with a final database. In a real time environment it is possible to receive updates to the initial database after the inference procedure has started or even after it has ended. We present an algorithm for incremental maintenance of the inference database in the presence of such updates. Interestingly, the same algorithm is useful for parallel and distributed rule processing. When the processors evaluating a program operate asynchronously, then they may have different *views* of the database. The incremental maintenance procedure we present can be used to synchronize these views.

[CUCS-059-90] Active databases for communication network management.

Ouri Wolfson, Soumitra Sengupta, and Yechiam Yemini.

This paper has two purposes. First is to propose new database language features for systems used in real time management. These features enable the specification of change-traces, events and correlation among events, and they do so in a declarative set-oriented fashion. Second is to introduce network management as an important and interesting application of active distributed databases.

[CUCS-060-90] Getting and keeping the center of attention.

Rebecca J. Passonneau.

The present work extends prior investigations into centering and attentional state by investigating the contrastive functions of a definite and a demonstrative pronoun in very similar contexts of use. It thus provides an opportunity to examine the separate contributions to attentional state of two linguistic features—definiteness and demonstrativity—independently of pronominalization per se. The two pronouns, 'it' and 'that', have clearly contrastive contexts of use, explained here in terms of distinct pragmatic functions. Certain uses of 'it' are claimed to perform a distinctive cohesive function, namely to establish a "local center" (which extends rather than replaces the notion of a center). The crucial distinction between a local center and the Cb of the centering framework is that there is only a single potential local center rather than an ordered set of Cfs. The local center is argued to constitute a reference point in the model of the speech situation in a manner analogous to 1st and 2nd person pronouns. In contrast, a deictic function is posited for apparently anaphoric uses of 'that' whereby the attentional status of a discourse entity is changed, or a new discourse entity is constructed from non-referential constituents of the linguistic structure. Since it is impossible to observe attentional processes directly, I present a general empirical methodology for investigating discourse coherence relations. I analyze statistically significant distributional models in terms of three types of transitions in the cognitive states of conversational participants: expected transitions, unexpected transitions, and transitions with no relevant effect.

[CUCS-063-90] Complexity of mortgage pool allocation.

Salvatore J. Stolfo and Xiangdong Yu.

A new class of NP-hard problems are proved and this result is applied to the complexity analysis of the mortgage pool allocation problem in American banks. A boundary on pool sizes is shown to sharply distinguish hard instances from easy ones in this kind of combinational optimazition problems.

[CUCS-064-90] Analyzing skewed symmetries.

Ari D. Gross and Terrance E. Boult.

A bilaterally symmetric figure on an arbitrarily oriented plane viewed under orthographic projection yields a "skew symmetric" figure whose axes of symmetry and skew constrain the orientation of the underlying plane. Given a skewed planar contour and a set of skewed axes, it is well known that the plane is constrained to lie on a hyperbola in the gradient space. Many papers have dealt with methods for finding a specific solution for the gradient of the skewed planar figure from amongst the hyperbola of possible solutions. Few papers have dealt with the practical side of finding the axes of skew. In particular, no direct non-heuristic method has yet been presented for determining the axes of skew, given points along the skewed contour. This paper provides a direct moment-based approach for finding axes of skew symmetry. This method does not rely on the contour being continuous. Examples of the method on both real and synthetic images are presented. The moment-based method assumes that the degree of skew symmetry is not greater than four. This is generally sufficient to handle the skewed symmetric contours that come up in real images. In addition to this global method, a new local method is introduced that uses tangent values at skew pairs to directly determine the skew angle. This method is shown to have certain desirable properties over existing local algortihms. As in the case of the global method, skewed axes are computed for real and synthetic contours. The aforementioned techniques presume that the contour is the projection of a planar symmetric figure. This paper derives a projective quasi-invariant that can be used for quick moment-based detection of skew symmetry.

[CUCS-065-90] Shape from a symmetric universe.

Ari D. Gross.

Much of the shape from contour work currently done in the vision community is model-based. Assuming a certain class of models, an image interpretation is given that comes closest to being consistent with the models. Although a model-based approach has the advantage of trying to find a solution for a relatively well-defined problem, it is constrained to recognition in a limited domain, i.e., scenes composed of objects in the class of models. Other shape from contour methods rely on heuristics, such as that of skew symmetry [kan81], or maximum compactness [brad83]. These methods, while not constrained by a set of models, generally try to solve for the 3D shape from a single contour alone. The problem of recovering shape from contour, however, is generally context-sensitive and cannot be solved for from a single contour, as this paper will show. There are shape from contour algorithms that try and combine heuristics with some degree of conext-sensitivity, but this is done by incorporating models into the recovery process. Perceptual organization techniques are context-sensitive and are not model-based (in a strict sense) but only compute 2D features, such as ribbons, and are very computationally expensive. This paper presents an approach to the shape from contour problem that does not rely on models, except in a very general sense. The underlying heuristic of this paper is that the world exhibits a high degree of symmetry and, wherever possible, a symmetric interpretation is preferred over an asymmetric one. This symmetry heuristic is used to recover shape from images where the correct solution is sensitive to constraints among adjacent contours. A hierarchical approach is used where planar skew symmetric regions are first detected in the image, and then axes of skew are recovered using SYMAN, a symmetry analyzer developed by the author. These skew symmetric regions tend to have several possible solutions for the set of skew axes; this ambiguity is known as axial ambiguity. This axial ambiguity can be resolved using context, determining which of these skew axes is globally consistent with a symmetric shape in the scene. Having recovered a unique set of skew axes for each skew contour, there is still a degree of ambiguity, termed gradient ambiguity, that needs to be resolved since there exists a hyperbola in the gradient space on which the gradient of a symmetric planar region can lie. By using a context-sensitive gradient constraint, the gradient solution of symmetric regions can be constrained. The transformation between viewer- and object- centered coordinate systems is then solved for by finding a set of image axes that form a basis for the object space and then solving for the 3D scene vectors corresponding to this image basis. Having recovered the transformation between viewer space and object space, the gradients at symmetric regions are then solved for uniquely.

[CUCS-066-90] Dynamic programming with convexity, concavity, and sparsity.

Zvi Galil and Kunsoo Park.

Dynamic programming is a general problem-solving technique that has been widely used in various fields such as control theory, operations research, biology, and computer science. In many applications dynamic programming problems satisfy additional conditions of convexity, concavity, and sparsity. This paper presents a classification of dynamic programming problems and surveys efficient algorithms based on the three conditions.

[CUCS-067-90] Inversion of toeplitz matrices with only two standard equations.

George Labahn and Tamir Shalom.

It is shown that the invertibility of a Toeplitz matrix can be determined through the solvability of two standard equations. The inverse matrix is represented by two of its columns, (which are the solutions of the two standard equations) and the entries of the original Toeplitz matrix.

Technical Reports Librarian
Department of Computer Science
Columbia University
New York, NY 10027