OREGON STATE UNIVERSITY COMPUTER SCIENCE DEPARTMENT

COMPUTER SCIENCE AND COMPUTER ENGINEERING
RESEARCH EQUIPMENT

Fred M. Tonge

Department of Computer Science
Oregon State University
Corvallis, Oregon 97331

83-4-1

FINAL TECHNICAL REPORT

National Science Foundation

Grant MCS-8112087

Computer Science and Computer Engineering
Research Equipment

Department of Computer Science
Oregon State University

Fred M. Tonge
Principal Investigator

July 8, 1983

SUMMARY

This grant supported acquisition of a minicomputer system for departmental research. The equipment selected is a DEC VAX-11/750 system*, installed in remodeled space in the Computer Science (formerly Farm Crops) Building. Grant funds for equipment acquisition were supplemented by support from the Tektronix Foundation.

After completion of the physical facilities for housing the equipment, installation of the VAX-11/750 began in early November 1982. Professor Douglas Moran took responsibility for coordinating the preparation of physical facilities and the installation itself. A number of researchers were making productive use of the system by January 1983. The license for 4BSD UNIX** was obtained in March 1983. Beginning Spring term of 1983, the VAX system was also used for several graduate courses and graduate student projects.

Research projects involving ten faculty members and over twenty graduate students are currently active on the system. These projects are in the areas of software complexity, computer graphics, database management, database system architecture, database reliability and consistency, expert systems, natural language processing, language implementation, intelligent interfaces, and decision support systems.

The following sections list the equipment configuration as of the end of the grant period, the major software systems installed, and descriptions of ongoing research projects.

---

* DEC and VAX are Trademarks of the Digital Equipment Corporation.
** UNIX is a Trademark of Bell Laboratories.

EQUIPMENT CONFIGURATION (as of May 31, 1983)

VAX-11/750

4 megabytes main memory
16 terminal ports (DZ)
1 RM-80 Winchester disk (124 megabytes) on Massbus controller
1 TU-77 tape drive (800/1600 bpi) on Massbus controller

LP-26 line printer

2 DEC DF03-AC modems (300/1200 baud, auto/manual answer/dial)

1 Racal-Vadic VA212 modem (300/1200 baud, auto/manual answer/dial)

13 VT-100 terminals

5 GIGI terminals

2 Spinwriters, model 7725 (KSR)

<u>Pending</u>

3 RA-81 Winchester disks on UDA-50 Unibus controller
(456 megabytes each)

MAJOR SOFTWARE AVAILABLE (as of May 31, 1983)

Operating System:
4.1 BSD UNIX (upgrade to 4.2 planned)

Programming Languages:
included in 4.1 BSD UNIX:
C
EFL (fortran)
PASCAL
Franzlisp
Interlisp-VAX (from USC-ISI)
ICON (string processing language) (from U. of Arizona)
Modula-2 *
CLU *
PSL (Portable Standard Lisp) (from U.of Utah)

Networks
CSnet
uucp (UNIX-to-UNIX CoPy network)
USENET (UNIX Users' News network, via uucp)
- uses NOTESFILES interface (from U. of Illinois)
BerkNet (local network connection to department's PDP-11/44**
and, via the 11/44, to University Computer Center)

Software Environment
Forms package - permits terminal screen forms to be quickly designed and implemented (from Johns Hopkins U.)
RCS (Revision Control System) - a successor to SCCS (Source Code Control System) - tracks updates to a file so that they can be easily reversed. (from Purdue U.)
EMACS Editor (Gosling and CCA versions)

TCSH - the 4.2BSD extended C-shell -- permits filename completion
VSH - visual shell
WM - window manager (from Naval Research Lab)

Text Processing

BIB - bibliographic system - successor to REFER -- entries are stored in a database and extracted when they are referenced in a paper. Inexact references and keywords are supported. (from U. of Arizona)

---

* currently being obtained.
** PDP is a Trademark of the Digital Equipment Corporation.

DESCRIPTIONS OF RESEARCH PROJECTS

Reduced Form for Sharing Software Complexity Data

Participants: Curtis R. Cook, Warren A. Harrison

We are currently engaged in the development of a "reduced form" which can capture the characteristics of interest to those studying software complexity, yet ensure that the software cannot be reconstructed. We are hopeful that such a reduced form may make organizations more willing to share data on their software. This may aid us in the evaluation of complexity metrics.

Presently, we are constructing a tool that will automatically generate a reduced form for COBOL programs.

PUBLICATIONS:

"A Reduced Form for Sharing Software Complexity Data" by W. Harrison and C. Cook, OSU Tech Report #83-1-1, also submitted to IEEE Transactions on Software Engineering.

"Comparing the Sensitivity of Topological Complexity Metrics for a Restricted Class of Programs" by W. Harrison, submitted to the Journal of Systems and Software.

Geometric Graphics Package

Participant: Alan J. Coppola

The goal of this project is to provide interactive graphics support for planar graph-theoretic and computational geometry problems. Current local applications for this package are in the creating and analyzing of heuristics for the planar Steiner minimal tree problem (find the shortest length tree containing a given finite set of points). Globally, any problem involving planar graphs and/or finite subsets of the plane can use the geometric graphics package.

Besides the usual geometric operations which occur in a simple graphics package (zoom, translate, rotate, reflect, character operations, etc.), the package includes the geometric operations of:

a) Finding and displaying a minimal length spanning tree(no extra vertices) for finite subsets of the plane.

b) Finding and displaying the Voronoi(Thiessen) polygonal decomposition of the plane, with respect to a finite subset of the plane.

c) Finding and displaying the Delaunay triangulation of a finite subset of the plane.

Operations b) and c) use Shamo's algorithm, which computes the Voronoi polygons and the Delaunay triangulation in time O(nlog(n)), where n is the number of points.

Using the VAX, and VK100 DEC low resolution color raster graphics (GIGI) terminals, Youn-Su Kim completed the first version of the geometric graphics package described above. This work, done in the C programming language, constituted part of her Masters' thesis work. The thesis was completed in March of 1983.

Future work on this project will be done by myself and Jay Nave. Nave's work will be part of his Masters' thesis. The work will involve modifying the current existing graphics primitives, and adding new primitives.

Heuristics for Steiners' Problem

Participant: Alan J. Coppola

The goal of this project is to develop heuristics for approximating solutions to Steiners' problem for finite point sets in the plane. The techniques used in this project are a combination of numerical analysis, non-linear programming, combinatorial optimization, and computational geometry. The geometric graphics package is used as an algorithmic design aid to enhance the creation and experimental testing of new heuristics for the NP-hard problem of finding a Steiner minimal tree.

Youn-Su Kim, as part of her Masters' thesis developed an iterative 'string' model heuristic, based on Shamo's algorithm for finding a minimal spanning tree. The results of this heuristic currently work for n<=5.

The major part of Jay Nave's Masters' thesis will be devoted to extending and modifying the current heuristic, in conjunction with developing a new heuristic.

Device driver for TEK4113 graphics terminal

Participant: Alan J. Coppola

The goal of this project is to take the geometric graphics package developed by Youn-Su Kim, and make it run on the TEK4113 medium resolution color graphics terminal. The TEK4113 terminal has more local capability than the GIGI terminals. It is a complicated device.

Tim Holt, a senior at OSU, completed the transfer of the geometric graphics package during the spring quarter of 1983. Some features which are device dependent were also added to the package, but these were isolated to increase portability possibilities.

Extended Network Database Management System

Faculty participant: Earl F. Ecklund, Jr.
Graduate student participants: Ms. Irene Anderson, Mr. Rajen Shah, Ms. Chattip Swasdichai, Ms. Myra Uy, Mr. Jianhua Zhu.

This project is to provide a vehicle for ongoing information systems architecture as well as an environment for teaching database management system implementation. Ms. Swasdichai and Ms. Uy are currently implementing the basic network DBMS with two features: The object schema schema is specified as a network database, and will be maintained within the DBMS using the usual DML commands, and CALC is defined in terms of extendible hashing tables. Ms. Anderson is designing a Data Dictionary in a DBMS, which will be integrated with the object schema and maintained within the DBMS. Mr. Zhu is working on compilation of schema specifications (DDL) into an object schema representation.

Mr. Shah is working on the pedagogical part of the project, a dynamic linking capability which will enable a student to write a module for the system and run it in parallel with the equivalent system module, validating their results with the results of the native system module. This feature has promoted a clean system design with all module interfaces being functional, devoid of side effects.

The contributions of each of the students will be the basis of their Masters theses.

Future enhancements will include a dynamic DDL facility which will allow schema to be defined or modified dynamically. This will use the integrated object schema facility of the current phase. Also we intend to investigate multiple model database system support by adding an external

relational model capability which will coexist on a common physical database with the network system.

Database Operating System.

Faculty participant: Earl F. Ecklund, Jr.
Graduate Student participant: Mr. Larry England.

We are investigating the relationship of operating system structure with the efficiency and throughput of a high-volume online transaction system, where we characterize such a system as processing short database queries or updates at a volume in excess of 1000 transactions per second. We are focusing on elimination of "standard system overhead" wherever possible, and removal of redundant actions performed by many operating systems and database systems. Thus we propose an architecture which integrates database management into the operating system, and structures the components of the system into a network of actions through which the transactions flow as data through a dataflow architecture. Mr. England, for his Masters Thesis, is designing the kernel primitives needed to route each transaction through the appropriate actions in the system.

The next step for this project is to simulate and/or prototype the kernel design to verify the appropriateness of the proposed architecture. The simulation or prototyping will be done on the VAX.

OSIRIS Project

Participant: Michael J. Freiling

OSIRIS is a prototype database system architecture which is being developed to illustrate certain architectural principles which lead to a great deal of flexibility and robustness. The main architectural principles are:

--- Inclusion of a distinct level of data representation which embodies no assumptions about either the interpretation of data, nor its structure.

--- The power to describe complete algorithms during query optimization, so that semantically described queries can be mapped completely into lower level algorithms, removing the need for excessive dialogue between different levels of the architecture.

--- Encapsulation of all access methods as instances of a single abstract data type, to provide for extension to yet undiscovered algorithms or storage technologies.

Implementation of various levels of this architecture are proceeding as follows.

SEMANTIC LEVEL (Shirley Roth)

A semantic data model called SIDUR has been specified and is currently being implemented. The query processor for SIDUR handles queries expressed as manipulations of three principal constructs: objects, actions, and situations. SIDUR queries are mapped into a declarative formalism referred to as the "bag query language", which completely describes the operations to be performed on stored data.

BASIC ACCESS MACHINE (Pat Kalvin, Earl Ecklund)

Once semantic level queries are mapped into the bag query language, it is intended that they be optimized by a bag query processor, which converts the optimized query into a dataflow graph whose constituents call individual access methods. The program code to implement this dataflow graph is produced by a program called the access code expander. The access code expander retrieves access methods from a library and "glues" them together with instructions which process the dataflow stream containing partial query results. All instructions are written for a virtual machine called the "basic access machine". Implementation of the access code expander and basic access machine is now largely complete. The code is being ported to the VAX from Version 6 UNIX. Implementation of the bag query processor itself is awaiting further design of the optimizer.

RELIABLE B-TREE (Marion Hakanson)

To permit experimentation with less than full-scale versions of the bag query processor, a Lisp-based access method has been implemented. This access method is based on the B-tree family of indexing techniques, and uses shadow updating to provide resiliency. This project was performed as a part of Marion's undergraduate course work.

Expert System Tools

Participant: Michael J. Freiling

A number of Lisp-based tools for expert system construction are being produced using the VAX.

LISP UTILITIES (Marion Hakanson)

Small Lisp utilities, such as a structure editor, filing package, and uniform interface to property list functions have been built for Franz Lisp and are being ported to Portable Standard Lisp.

LARGER TOOLS (Marion Hakanson)

Some larger tools, including a generalized property list package and agenda based programming system have been written for ALISP on the CDC Cyber and are being ported to Franz Lisp and Portable Standard Lisp.

HCPRVR (David Montague)

Dan Chester's Horn Clause Prover (HCPRVR), which is functionally equivalent to a subset of Prolog, has been implemented on the VAX, together with axiom sets demonstrating that abstract algebraic specifications for data types such as queue and stack are directly executable by a logic programming system.

YET ANOTHER PORTABLE LISP (Mojy Mirashrafi)

A portable Lisp written in a special macro language called LIL has been implemented on the VAX by writing macros to translate LIL commands to C code. This Lisp contains special extendibility features not found on other Lisps, such as the ability for the user to define new function types as well as new functions, and the ability to use the Lisp internal stack as if it were a double-ended queue, permitting agenda-style processing directly in Lisp. This experimental Lisp, CORVALLISP, is also being implemented on the Intel 86/330.

VAX -- INTEL 86/330 COMMUNICATIONS LINK (Jim Harriger)

To facilitate downloading of the CORVALLISP implementation from the VAX to the Intel 86/330, a communications package between the machines has been implemented. This package supports file transfer in both directions.

Intelligent Tactical Assistant

Participants: Ken Funk, Paul Meagher (both Dept. of Industrial and General Engineering)

With the increase in the tempo and density of the naval tactical environment, the need to make timely yet well-informed decisions is becoming more and more critical. The ITA (Intelligent Tactical Assistant) project is concerned with the development of a prototype knowledge-based system to aid commanders in tactical situation assessment. This prototype will consist of a frame-like associative network in which is encoded a large body of information about surface ships, sensors, and weapons, and a rule-based interpreter which will work interactively with the operator to

identify radar contacts and perform threat evaluation.

The project is sponsored by the Navy Personnel Research and Development Center and is administered by Battelle.

Consistent Distributed Database State Maintenance

Participant: Toshi Minoura

A theory for resiliency control of distributed database systems was developed, and schemes that preserve the consistent global database state in the presence of site crashes and message link failures were devised. The schemes can be paired with any concurrency control scheme that produces "V-serializable" executions, and one of the schemes can handle "partially-replicated data" as well as "partitioned data" and "fully-replicated data".

The research was performed with Profs. Susan Owicki and Gio Wiederhold, both Stanford University. The result of this research has been submitted for possible publication in ACM Transactions on Database Systems.

Reliable and Consistent Database State Maintenance.

Participant: Toshi Minoura

A typical database system stores its data redundantly on secondary storage in order to support them reliably. Then, when a data item is updated, multiple secondary storage write operations are required in order to update its data copies. Thus, updating data copies on secondary storage may become the bottleneck for the transaction processing. We have reviewed several reliable storage update mechanisms for centralized database systems, and are now investigating various buffering schemes for secondary storage update operations. Secondary storage update buffering is expected to improve the performance of reliable storage update mechanisms, since it allows more concurrency and flexible disk scheduling.

We are now developing a simulator in order to confirm our claims. Three undergraduate students have written a disk scheduler and have confirmed the results on various disk scheduling policies as reported by Teorey and Pinkerton [CACM, March 1972]. The simulator that includes a concurrency control mechanism and a buffering mechanism is now being developed by Mr. Ching-Chao Liu. Preliminary results

of our theoretical study have been submitted for a possible conference presentation at 3rd Symp. on Reliability in Distributed Software and Database Systems. This research is also supported by NSF grant ECS-8307478 "Research Initiation: Reliable and Consistent Database State Maintenance."

Multi-Version Concurrency Control Scheme for a Database System

Participant: Toshi Minoura

A multi-version concurrency control scheme for centralized database systems was developed. In developing the scheme, we devised the notion of "V-serializability", which is a serializability condition based on versions. The scheme possesses several novel features.

This research was performed with Dr. Shojiro Muro, Kyoto University, and Prof. Tiko Kameda, Simon Fraser University. The result of this research will be published in JCSS.

Publications

(1) Muro, S., Kameda, T., and Minoura, T. "Multi-Version Concurrency Control Scheme for a Database System." To appear in JCSS. Also, TR 82-2, Dept. of Computing Science, Simon Fraser University, Feb. 1982.

(2) Minoura, T., Owicki, S., and Wiederhold, G. "Consistent Distributed Database State Maintenance." Submitted to ACM Tr. on Database Systems, being revised.

(3) Liu, C. and Minoura, T. "Reliable Storage Update." Submitted to 3rd Symp. on Reliability in Distributed Software and Database Systems.

Natural Language Processing

Participant: Douglas B. Moran

This project is in the preliminary stage of building a computational environment. Several natural language processing systems have been acquired from other researchers. These systems will serve two basic purposes: (1) they can be used as components in larger systems where the focus is on the development of the other components, (2) they can serve as the basis for research to extend the theory embodied in the system. These systems will also be available to other research projects to use as interfaces to their systems. We

are presently focusing on three acquired systems:

(1) The RUS parser. This is the most comprehensive of the cascaded ATN parsers. It was developed at BBN by Bobrow, Bates, et al and converted from Interlisp-10 to Interlisp-VAX at U. of Delaware by Weischedel et al.

(2) The TEXT language generation system. This system generates sentences in response to questions about the information in a database. This was developed at the U. of Pennsylvania by Kathleen McKeown (now at Columbia U.), Kathy McCoy and Steve Bossie.

(3) Mitchell Marcus' PARSIFAL "deterministic" parser. This parser addresses issues of performance in the human sentence parsing mechanism.

These systems were chosen because they are significant systems that were coded for an computational environment similar to the one we have. We expect to continue to acquire interesting systems that can be reasonably converted to run in our environment.

Prototype Language Implementation

Participant: David W. Sandberg

This project is to implement a programming language that will be used to investigate some ideas for programming language. It will attempt to provide better mechanisms for abstraction than present in existing programming languages, provide a better programming environment than usually provided, and investigate making end of lines and indentation semantically significant to the programming language. Furthermore, the language should be very simple, otherwise the project will quickly become too large.

So far about 1800 lines of code have been written. This includes a screen editor and the parse and semantic phase of the compiler. A code generator, and a run-time environment/interpretor are yet to be written. By the end of the summer some results are expected on the effects of intergrating a editor with a compiler and making end of lines and indentation significant. The abstraction mechanism is still in its formative stages and will take longer to design and develop.

Intelligent Interface for a Text Processor

Participants: Fred Tonge, Doug Eshelman

The goal of this project is to design and implement an intelligent interface for inexperienced users of a text processing system. The initial version of the interface will allow the user to ask "How do I do ... ?" and "Where did I go wrong?". Later versions will include a model of the user, through which the system can adapt to the particular user's experience and can provide training feedback to the user based on discrepancies between him and an ideal user.

Document Preparation with Graphics

Participants: Fred Tonge, Juanita Ewing

As part of a project to design a decision support system for information systems analysis, a document preparation system is being designed and built in which graphical objects can be manipulated and edited by the user at the same level as and intermixed with text. Initial graphical objects will include bar charts, pie charts and scatter diagrams. Later additions will include bubble diagrams, flow charts, and cost and statistical tabulations.

Decision Support System for Strategic Planning

Participants: Fred Tonge, Mike Goul, Barry Shane

An expert system to assist in the preparation of a strategic audit for an organization is being developed. The system is based upon a specific model of the way in which decision makers attend to and use information. The system will direct the user's attention to particular facts, issues, discrepancies in information already received, questions to ask, and other aspects of carrying out an analysis of the organization's strategic position. An initial evaluation of the system's usefulness will be made through a controlled experiment involving its use in a senior business policy class. Mr. Goul's doctoral dissertation will be based in part on this work.

The underlying system will also be adapted to serve as part of a proposed decision support system for information systems analysis. In that context, it will help to guide initial information gathering as to the underlying issues and problems which led to the authorization of an information systems analysis project.

Word Processing Extension to EMACS

Participants: Fred Tonge, Judy Pohl

The purpose of this project is to design and implement a word processor using the screen editor EMACS as its core. Implementation of this extension will involve changes to the commands and internal structures of EMACS. These are intended to make the commands more consistent and easier to use by office staff non-programmers. The package will also include many word processing additions, resulting in a screen editor oriented toward document preparation with the capacity for partially interactive formatting.