A RESEARCH FACILITY FOR COOPERATIVE DISTRIBUTED COMPUTING: INTEGRATING RESEARCH IN AI AND DISTRIBUTED COMPUTING

A Coordinated Experimental Research Proposal

by

Victor R. Lesser
Krithivasan Ramamritham
Edward M. Riseman

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

Page 1 of 2

UNIVERSITY OF MASSACHUSETTS AT AMHERST
INTERNAL PROCESSING FORM FOR SPONSORED PROJECTS
OFFICE OF GRANT AND CONTRACT ADMINISTRATION

Tel. 545-0698

PROPOSAL NO.: OGCA ____________

RECEIVED: ___/___/___
Month Day Year

PLEASE READ INSTRUCTIONS ON THE REVERSE

PRINCIPAL INVESTIGATOR (OR FACULTY SPONSOR): Victor R. Lesser

Dept./Unit; Campus Address: Computer and Information Science, LGRES, Rm.A305 Phone: 545-1322

voluntary information; requested by federal agencies

SEX: □ F ☒ M

□ BLACK
☒ CAUCASIAN
□ AMERICAN INDIAN/ ALASKAN NATIVE
□ HISPANIC
□ ASIAN OR PACIFIC ISLANDER
□ HANDICAPPED
□ VIETNAM ERA VETERAN

PROPOSED SPONSOR: National Science Foundation DUE DATE: □ Postmark
□ Receipt 9 / 17 / 84
Month Day Year

1. PROJECT TITLE: A Research Facility for Cooperative Distributed Computing:
(60 spaces max.) Integrating Research in AI and Distributed Computing

2. PROJECT CHARACTERISTICS: ☒ NEW
□ RENEWAL OF ACCOUNT NO.: ____________
□ SUPPLEMENT TO ACCOUNT NO.: ____________
☒ RESEARCH
□ INSTRUCTION
□ OTHER

3. PLEASE ATTACH ABSTRACT OR SUMMARY (OR REFERENCE PAGE IN PROPOSAL).

4. PROPOSED START DATE: July 1, 1985 DURATION: Five years

5. DOES THE PROPOSED ACTIVITY INVOLVE:
a. Use of human subjects? □ Yes ☒ No
b. Use of vertebrate animals? □ Yes ☒ No
c. Use of computing services? □ Yes ☒ No — If yes, is it budgeted in full in the proposal? □ Yes □ No
d. Recombinant DNA? □ Yes ☒ No — Biohazardous Substances? □ Yes ☒ No
Radioactive Materials? □ Yes ☒ No
e. UMass equipment matching funds? ☒ Yes □ No Approved: Dean, Graduate Studies and Research ____________
f. Additional faculty/professional staff to be employed by the University? ☒ Yes □ No If yes, number? ____________

6. THE FOLLOWING FACULTY OR PROFESSIONALS OTHER THAN THE PRINCIPAL INVESTIGATOR WILL PARTICIPATE: (Indicate institution if other than University of Massachusetts.)

NAME	NATURE OF PARTICIPATION	RANK AND UNIT
Krithivasan Ramamritham	Co-Principal Investigator	Assistant Professor, COINS
Edward M. Riseman	Co-Principal Investigator	Professor, COINS

7. UNIVERSITY FACILITIES REQUIRED:
a. Will adequate space be available for the period proposed? ☒ Yes □ No
If yes, building: Lederle Graduate Research Center Room: A25 Approved: E. Riseman
If no, space required: ________ sq. ft. Source: ____________ Approved: ____________
b. Will installation, space renovation or building modification be required? ☒ Yes □ No
If yes, Cost: $ 120,000 UM Account No.: ____________ Approved: ____________

FOR OGCA USE

SPECIAL REVIEWS —

TO BE REVIEWED BY	CONDITION	SIGNATURE

(rev 2/82)

(Instructions: back of page 2)

FIVE-YEAR CUMMULATIVE BUDGET

PROPOSAL NO.: OGCA ______

☒ Original Budget Request ☐ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (Include Research Assistants)	$464,500	$1,124,600	$1,589,100		
b. Fringe Benefits	43,359	244,009	287,368		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		15,000	15,000		
g. Supplies/Materials		50,000	50,000		
h. Equipment*-Purchased	150,000	1,780,000	1,930,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (Itemize under notes)	120,000	730,900	850,900		
l. Total Direct Costs (TDC)	$777,859	$3,944,509	$4,722,368		
m. Indirect Costs (IC)	243,769	1,038,964	1,282,733		

Federal: 48 % MTDC (l. minus: h. and j. in excess of $25K each subcontract)

Nonfederal ______% TDC

TOTAL COSTS (TC) $1,021,628 $4,983,473 $6,005,101

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

* Refer to Instruction 8h. back of page 1.

8.k.

Maintenance	$713,900
Telephone	$12,000
Publication Costs	5,000

2 2 2 1 2
3 874,620.
1 312415
875866
937153
983409
4,983,463

Fabricated equipment:

Salaries & Wages ______

Fringe Benefits ______

Materials ______

**TOTAL ______

______ 9/14/84
Submitted by Principal Investigator(s) (Date)

______ ______
(Date)

______ 9-17-84
Approved for Department or Unit Head(s) (Date)

Fiscal Administrator: ______ ______
(Date)

______ 9/17/84
Approved for School/College by: Dean(s) (Date)

Director OGCA: ______ ______
(Date)

(rev 2/82)

(Instructions: back of page 1)

YEAR 5

PROPOSAL NO.: OGCA ________

☒ Original Budget Request ☐ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (Include Research Assistants)	$116,000	$239,700	$355,700		
b. Fringe Benefits	9,716	54,617	64,333		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		3,000	3,000		
g. Supplies/Materials		10,000	10,000		
h. Equipment*-Purchased	30,000	220,000	250,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (Itemize under notes)		208,500	208,500		
l. Total Direct Costs (TDC)	155,716	735,817	891,533		
m. Indirect Costs (IC)	60,343	247,592	307,935		

Federal: 48 % MTDC (l. minus: h. and j. In excess of $25K each subcontract)

Nonfederal ____ % TDC

	UM SOURCES	SPONSOR	TOTAL
TOTAL COSTS (TC)	$216,059	$983,409	$1,199,468

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620.	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

8.k.

Maintenance	$205,100
Telephone	$2,400
Publication Costs	$1,000

* Refer to Instruction 8h. back of page 1.

Fabricated equipment:

Salaries & Wages ________

Fringe Benefits ________

Materials ________

**TOTAL ________

Submitted by: Principal Investigator(s) 9/14/84 (Date)

Approved for Department or Unit Head(s) 9-17-84 (Date)

Approved for School/College by: Dean(s) 9/17/84 (Date)

Fiscal Administrator: ________ (Date)

Director OGCA: ________ (Date)

________ (Date)

(rev 2/82)

(Instructions: back of page 1)

YEAR 4

PROPOSAL NO.: OGCA ______

☒ Original Budget Request ☐ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (Include Research Assistants)	$97,600	$238,200	$335,800		
b. Fringe Benefits	9,163	51,563	60,726		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		3,000	3,000		
g. Supplies/Materials		10,000	10,000		
h. Equipment*-Purchased	30,000	220,000	250,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (Itemize under notes)		181,800	181,800		
l. Total Direct Costs (TDC)	136,763	704,563	841,326		
m. Indirect Costs (IC)	51,246	232,590	283,836		

Federal: 48 % MTDC (l. minus: h. and j. in excess of $25K each subcontract)

Nonfederal ____% TDC

	UM SOURCES	SPONSOR	TOTAL
TOTAL COSTS (TC)	$188,009	$937,153	$1,125,162

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

8.k.

Maintenance	$178,400
Telephone	$2,400
Publication Costs	$1,000

* Refer to Instruction 8h. back of page 1.

Fabricated equipment:

Salaries & Wages ______

Fringe Benefits ______

Materials ______

**TOTAL ______

Submitted by: Principal Investigator(s) ______ 9/14/84 (Date)

Approved for Department or Unit Head(s) ______ 9-17-84 (Date)

Approved for School/College by: Dean(s) ______ 9/17/84 (Date)

Fiscal Administrator: ______ ______ (Date)

Director OGCA: ______ ______ (Date)

(rev 2/82)

(Instructions: back of page 1)

YEAR 3

PROPOSAL NO.: OGCA ______

☒ Original Budget Request ☐ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (Include Research Assistants)	$92,000	$224,700	$316,700		
b. Fringe Benefits	8,633	48,653	57,286		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		3,000	3,000		
g. Supplies/Materials		10,000	10,000		
h. Equipment*-Purchased	30,000	220,000	250,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (Itemize under notes)		156,800	156,800		
l. Total Direct Costs (TDC)	130,633	663,153	793,786		
m. Indirect Costs (IC)	48,303	212,713	261,016		

Federal: 48 % MTDC (l. minus: h. and j. in excess of $25K each subcontract)

Nonfederal ____% TDC

	UM SOURCES	SPONSOR	TOTAL
TOTAL COSTS (TC)	$178,936	$875,866	$1,054,802

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

8.k.

Maintenance	$153,400
Telephone	$2,400
Publication Costs	$1,000

* Refer to Instruction 8h. back of page 1.

Fabricated equipment:

Salaries & Wages ______

Fringe Benefits ______

Materials ______

**TOTAL ______

Submitted by Principal Investigator(s) ______ 9/14/84 (Date)

Approved for Department or Unit Head(s) ______ 9-17-84 (Date)

Approved for School/College by: Dean(s) ______ 5/17/84 (Date)

______ (Date)

Fiscal Administrator: ______ (Date)

Director OGCA: ______ (Date)

(rev 2/82)

(Instructions: back of page 1)

YEAR 2

PROPOSAL NO.: OGCA ________

☒ Original Budget Request ☐ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (Include Research Assistants)	$86,900	$212,000	$298,900		
b. Fringe Benefits	8,152	45,887	54,039		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		3,000	3,000		
g. Supplies/Materials		10,000	10,000		
h. Equipment*-Purchased	30,000	720,000	750,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (Itemize under notes)		129,400	129,400		
l. Total Direct Costs (TDC)	$125,052	$1,120,287	$1,245,339		
m. Indirect Costs (IC)	45,624	192,138	237,762		

Federal: 48 % MTDC (l. minus: h. and j. in excess of $25K each subcontract)

Nonfederal ____% TDC

	UM SOURCES	SPONSOR	TOTAL
TOTAL COSTS (TC)	$170,676	$1,312,415	$1,483,101

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

8.k.

Maintenance	$126,000
Telephone	$2,400
Publications	$1,000

* Refer to Instruction 8h. back of page 1.

Fabricated equipment:

Salaries & Wages ________

Fringe Benefits ________

Materials ________

**TOTAL ________

Submitted by Principal Investigator(s) 9/14/84 (Date)

E. Riseman

Approved for Department or Unit Head(s) 9·17·84 (Date)

Approved for School/College by: Dean(s) 9/17/84 (Date)

(rev 2/82)

________ (Date)

Fiscal Administrator: ________ (Date)

Director OGCA: ________ (Date)

(Instructions: back of page 1)

YEAR 1

PROPOSAL NO.: OGCA ____________

☒ Original Budget Request □ Revised Budget Request

8. SUMMARY OF PROPOSED BUDGET (INITIAL AWARD PERIOD):

	UM SOURCES	SPONSOR	TOTAL	ACCOUNT NO. FOR UM SOURCES	APPROVED BY:
a. Salaries & Wages (include Research Assistants)	$ 72,000	$210,000	$282,000		
b. Fringe Benefits	7,695	43,289	50,984		
c. Fellowships					
d. Tuition					
e. Consultants					
f. Travel		3,000	3,000		
g. Supplies/Materials		10,000	10,000		
h. Equipment*-Purchased	30,000	400,000	430,000		
-Fabricated**					
i. Computer Costs					
j. Subcontracts					
k. Other (itemize under notes)	120,000	54,400	174,400		
l. Total Direct Costs (TDC)	229,695	720,689	950,384		
m. Indirect Costs (IC)	38,253	153,931	192,184		

Federal: 48 % MTDC (l. minus: h. and j. in excess of $25K each subcontract)

Nonfederal ____% TDC

	UM SOURCES	SPONSOR	TOTAL
TOTAL COSTS (TC)	$267,948	$874,620	$1,142,568

9. TOTAL COSTS REQUESTED FROM SPONSOR, BY YEAR:

FIRST YEAR	SECOND YEAR	THIRD YEAR	FOURTH YEAR	FIFTH YEAR
$874,620	$1,312,415	$875,866	$937,153	$983,409

10. NOTES:

8.k.

Maintenance	$51,000
Telephone	$2,400
Publication Costs	$1,000

* Refer to Instruction 8h. back of page 1.

Fabricated equipment:

Salaries & Wages ________

Fringe Benefits ________

Materials ________

**TOTAL ________

Submitted by: Principal Investigator(s) ________ 9/14/84 (Date)

Approved for Department or Unit Head(s) ________ 9-17-84 (Date)

Approved for School/College by: Dean(s) ________ 9/17/84 (Date)

________ (Date)

Fiscal Administrator: ________ (Date)

Director OGCA: ________ (Date)

(rev 2/82)

(instructions: back of page 1)

PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

Cover Page

FOR CONSIDERATION BY NSF ORGANIZATIONAL UNIT (Indicate the most specific unit known, i.e. program, division, etc.)
The Coordinated Experimental Research Prog.
Computer Science Section

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? Yes ___ No X : IF YES, LIST ACRONYM(S):

PROGRAM ANNOUNCEMENT/SOLICITATION NO.:

CLOSING DATE (IF ANY):

NAME OF SUBMITTING ORGANIZATION TO WHICH AWARD SHOULD BE MADE (INCLUDE BRANCH/CAMPUS/OTHER COMPONENTS)
University of Massachusetts/Amherst

ADDRESS OF ORGANIZATION (INCLUDE ZIP CODE)
University of Massachusetts
Department of Computer and Information Science
Lederle Graduate Research Center, Rm.A305
Amherst, Massachusetts 01003

TITLE OF PROPOSED PROJECT
A Research Facility for Cooperative Distributed Computing

REQUESTED AMOUNT	PROPOSED DURATION	DESIRED STARTING DATE
$4,983,473	Five years	July 1, 1985

PI/PD DEPARTMENT	PI/PD ORGANIZATION	PI/PD PHONE NO. AREA CODE: (413)
Computer and Information Science	University of Massachusetts	545-1322

PI/PD NAME	SOCIAL SECURITY NO.*	DATE OF HIGHEST DEGREE ACHIEVED	MALE*	FEMALE*
Victor R. Lesser	061-36-8038	1972, Ph.D.	X	
ADDITIONAL PI/PD & SIGNATURE Krithivasan Ramamritham	528-25-8928	1981, Ph.D.	X	
ADDITIONAL PI/PD & SIGNATURE Edward M. Riseman	113-34-3547	1969, Ph.D.	X	
ADDITIONAL PI/PD & SIGNATURE				
ADDITIONAL PI/PD & SIGNATURE				

FOR RENEWAL OR CONTINUING AWARD REQUEST, LIST PREVIOUS AWARD NO.:

IF SUBMITTING ORGANIZATION IS A SMALL BUSINESS CONCERN, CHECK HERE □ (See CFR Title 13, Part 121 for Definitions)

*Submission of SSN and other personal data is voluntary and will not affect the organization's eligibility for an award. However, they are an integral part of the NSF information system and assist in processing proposals. SSN solicited under NSF Act of 1950, as amended.

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW:

- □ Animal Welfare
- □ Endangered Species
- □ Historical Sites
- □ Human Subjects
- □ Marine Mammal Protection
- □ Pollution Control
- □ National Environmental Policy Act
- □ Research Involving Recombinant DNA Molecules
- □ Proprietary and Privileged Information

PRINCIPAL INVESTIGATOR/ PROJECT DIRECTOR		AUTHORIZED ORGANIZATIONAL REP.		OTHER ENDORSEMENT (optional)	
NAME Victor R. Lesser		NAME		NAME	
SIGNATURE		SIGNATURE		SIGNATURE	
TITLE Professor		TITLE		TITLE	
DATE 9/14/84	Telephone No. Area Code: 413 545-1322	DATE	Telephone No. Area Code:	DATE	Telephone No. Area Code:

NOTICE OF RESEARCH PROJECT
SCIENCE INFORMATION EXCHANGE
SMITHSONIAN INSTITUTION
NATIONAL SCIENCE FOUNDATION
PROJECT SUMMARY

SIE PROJECT NO.

NSF AWARD NO.

FOR NSF USE ONLY

DIRECTORATE/DIVISION	PROGRAM OR SECTION	PROPOSAL NO.	F.Y.

NAME OF INSTITUTION (INCLUDE BRANCH/CAMPUS AND SCHOOL OR DIVISION)

University of Massachusetts

ADDRESS (INCLUDE DEPARTMENT)

University of Massachusetts
Computer and Information Science
Lederle Graduate Research Center
Amherst, Massachusetts 01003

PRINCIPAL INVESTIGATOR(S)

Victor R. Lesser, Krithivasan Ramamritham, and Edward M. Riseman

TITLE OF PROJECT

A Research Facility For Cooperative Distributed Computing:
Integrating Research in AI and Distributed Computing

TECHNICAL ABSTRACT (LIMIT TO 22 PICA OR 18 ELITE TYPEWRITTEN LINES)

The Computer and Information Science department, with colleagues from the Electrical and Computer Engineering Department at the University of Massachusetts, Amherst, is proposing a coordinated research project on *cooperative distributed computing* based on a tightly-coupled multiprocessor containing between 64 and 128 processing elements. A key aspect of this project will be to address issues that arise out of task and problem solving decomposition in which processes cooperate to solve *a single integrated problem* rather than *a set of independent tasks.* We are especially interested in the issues of cooperation and uncertainty in distributed systems with large number of nodes.

This project will coordinate and integrate large active research groups in the areas of AI (computer vision, robotics and distributed problem solving, adaptive learning networks, neural modelling) and Distributed and Parallel Processing (language primitives, synchronization, concurrency control, network communications, real-time scheduling, load balancing, debugging, protection and reliability).

1. Proposal Folder
2. Program Suspense
3. Division of Grants & Contracts
4. Science Information Exchange
5. Principal Investigator
6. Off. of Govt. & Pub. Progs.

A RESEARCH FACILITY FOR COOPERATIVE DISTRIBUTED COMPUTING: INTEGRATING RESEARCH IN AI AND DISTRIBUTED COMPUTING

A Coordinated Experimental Research Proposal

by

Victor R. Lesser
Krithivasan Ramamritham
Edward M. Riseman

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

1. ABSTRACT

The Computer and Information Science department, with colleagues from the Electrical and Computer Engineering Department, at the University of Massachusetts, Amherst, is proposing a coordinated research project on *cooperative distributed computing* based on a tightly-coupled multiprocessor containing between 64 and 128 processing elements. A key aspect of this project will be to address issues that arise out of task and problem solving decomposition in which processes cooperate to solve *a single integrated problem* rather than *a set of independent tasks.* We are especially interested in the issues of cooperation and uncertainty in distributed systems with large number of nodes.

This project will coordinate and integrate large active research groups in the areas of AI (computer vision, robotics and distributed problem solving, adaptive learning networks, neural modelling) and Distributed and Parallel Processing (language primitives, synchronization, concurrency control, network communications, real-time scheduling, load balancing, debugging, protection and reliability).

We believe that the proposed tightly-coupled and large-scale multiprocessor is the most cost-effective means for performing the extensive and computationally intensive experiments needed to investigate issues in cooperative distributed computing. It will facilitate this research by providing:

1) *a resource for the study of large and complex distributed systems including the simulation of both tightly- and loosely-coupled distributed systems,*

2) *a realistic environment for research in hardware and software architectures to support distributed computation*, and

3) *the infrastructure that will facilitate the needed integration between the disciplines of Distributed Computing and AI, especially in the area of knowledge-based systems.*

In order to fully exploit the capabilities of the multiprocessor, we also request funds in two additional categories. The first is for a network of workstations to support the development of, interaction with, and analysis of the multiprocessor experiments. The second is for the support staff needed to develop and maintain the necessary software tools and to be responsible for the operation of the facility.

2. TABLE OF CONTENTS

Appendicies

3. DESCRIPTION OF EQUIPMENT, MAINTENANCE, AND STAFFING

3.1 Equipment

We propose the acquisition of a large scale multiprocessor facility which will be integrated into the existing departmental research computing facility. Table 1 details the proposed equipment purchases over the five-year period. At the heart of the facility will be a large, tightly-coupled multiprocessor containing between 64 and 128 processing elements each with the computing power of at least 1 Mips. We expect the total memory attached to the system to be between 128 to 256 megabytes. In order to effectively use the multiprocessor as an experimental tool, we also propose the acquisition of a set of workstations connected to the multiprocessor. These workstations will support the development, monitoring, and analysis of experiments done on the multiprocessor. As part of the proposed support hardware, we also request the acquisition of an extensive amount of mass storage to hold the data and programs required by experimenters using the new system and a high-speed line printer for hard copy listings of experimental data.

3.1.1 The Multiprocessor System

In order to support the wide range of distributed and parallel architectures that we will explore, to provide for the tremendous amount of computing cycles needed in our simulations, and to be easily usable and maintainable, we require the proposed multiprocessor to satisfy the following requirements:

1. It will provide over 100 mips of processing capability; in addition, it will have facilities to harness this computing power efficiently.

2. It will be built from off-the-shelf components based on one of the generally available microprocessor architectures with a large address space.

3. The support software and hardware will be geared to assist experimentation in loosely-coupled as well as tightly-coupled systems.

4. It will run COMMON LISP and C and will support large AI programs.

5. The basic software and firmware system components will be flexible and modular enough to permit experimentation, via real implementations, of our ideas in operating systems for distributed computing systems.

6. It will have a high bandwidth connection for workstations and mass storage devices.

7. The interconnection structure among processors will permit extensive interaction without significant bottlenecks.

8. It will be hardware and/or software partitionable making it possible to run multiple experiments simultaneously.

9. There will be at least a small number of other research institutions that have the multiprocessor so that we can share experiences and software.

10. The system will be modular with replicated hardware components; this will contribute to ease of maintenance while increasing its availability.

11. It will be well instrumented to support performance evaluation.

There are currently two machines with which we are familiar that meet these specifications: the BBN Butterfly (based on the Motorola MC 68000 processor) and the DEC PPA (based on the MICRO-VAX-II processor); we also suspect that there will be others coming out in the next year that may satisfy our specifications. The BBN and DEC machines are quite different. In the BBN architecture, each processor has local memory plus a very high-speed, parallel interconnection structure to permit access to the local memory on other processors. Through mapping tables contained on each processor, the memory associated with a program address space can be distributed arbitrarily through the processor network. In the DEC architecture, each processor has no local memory, but a large local cache plus a very high bandwidth bus to access global memory. There is special hardware associated with each processor to monitor requests on the bus in order to guarantee that data in the local cache is valid.

The BBN machine is available immediately, while the DEC machine is scheduled to be produced sometime in 1986. We have proposed delaying the acquisition of the multiprocessor until the second year of the grant for several reasons: since DEC has not released complete details concerning its architecture, it is difficult, at this time, to determine definitively whether the BBN or the DEC architecture best satisfies our needs[1] We also expect that there will be additional options available from other manufacturers during the second year of the grant; and, finally, we feel that a significant part of the software development that will be required to use the multiprocessor effectively can and should be carried out prior to the arrival of the machine.

Given the long lead time, an exact price quote for the multiprocessor systems is difficult to obtain from the vendors, but after informal discussions with BBN and DEC, we expect that $750,000 should provide sufficient funds to acquire an appropriate complement of processors and memory and to allow us to make effective use of the multiprocessor.

[1] There is a possibility that the DEC product may be better suited to our needs: It has a faster processor; it should be easier to integrate it with our current facility (through the high-speed cluster bus); it should be easier to transfer existing software to the DEC system; it is expected to have a better software base; and finally, it is highly (hardware) instrumented.

Initially, we expect to acquire at least 64 processors and 128 megabytes of memory. We feel that we need between two and four megabytes of memory for each of these processing elements because of our desire to explore high-level AI tasks which will involve significant amounts of data and program. This memory requirement is especially important if we choose the BBN architecture in which there is a significant performance penalty if a high percentage of accesses are not from the local memory attached to each processor. During Years Four and Five of the grant, as the user base and the set of applications for the multiprocessor facility significantly increases, we propose to augment the multiprocessor with additional processing elements and memory at a cost of approximately $350,000 ($250,000 in Year Four and $100,000 in Year Five). Though it is difficult to predict exactly, this upgrade may also involve the addition of specialized processors or co-processors to facilitate such operations as associative retrieval from a semantic data base, floating-point processing, low-level signal processing, etc.

3.1.2 Support Hardware

We propose the acquisition of three high-powered and five to six medium-powered workstations in the first year of the proposal to provide, respectively, *experimenters' workstations*, and a *development environment* for multiprocessor applications and operating system software. The high-powered workstations will be large SYMBOLICS 3600 class LISP machines with high-resolution color graphics (1024 x 1024) and the medium-powered workstations will be SUN or Micro-VAX workstations.

The LISP machines will be used as the experimenters′ workstation. Creation of the development environment will involve significant software development to provide tools for the experimenters to set up, monitor, debug, and analyze experiments on the multiprocessor. From past experience in our current research environment, the use of high-resolution color graphics is essential to understanding the dynamics of process structures, such as those that are encountered in cooperative problem solving and vision research. We have found that low-resolution color (512 x 512) does not permit the display of sufficient information to adequately represent the state of complex dynamic systems. This will be especially true when we extend our experiments into large and more complex distributed systems.

The software development environment and high-speed computation capabilities provided by the LISP machines will significantly aid in the development of the system software and its subsequent use by experimenters. The language-specific design of the Symbolics 3600 for example, enables it to run at 5 MIPS on operations central to LISP such as stack manipulation and function calls; this machine is on the order of 10 times the speed of a VAX 11/750 running LISP. In the 3600, single-user operation can exploit the bit-mapped displays, utilize the automatic sharing of the entire address space among all processes (which is crucial when interacting systems are being coordinated), and capitalize on the simplicity of the interprocess architecture. In addition, the software base for general program development (e.g. coordination between the editor and the debugger, and the sophistication of the window management facilities) is markedly larger and of a higher quality on the LISP machine than is available on other systems.

Our medium-powered workstations will be chosen to support the same high-level languages as the multiprocessor (e.g., COMMON LISP and C at a minimum). We plan to build facilities, on top of the operating system of the networked workstations, to simulate the multiprocessor environment. This will enable the network to be used for software development for multiprocessor applications, which can thus go on in parallel with large simulation runs on the multiprocessor. We recognize that this network environment will be extremely inefficient for simulating some of the architectures that we want to examine, especially the more tightly-coupled architectures. However, we believe the network of medium-powered workstations will have sufficient processing power to permit the debugging of many of our experiments with limited test cases involving a small number of nodes and small data sets. The network simulation of the multiprocessor will especially be needed during the initial years of the project when the multiprocessor either will not be available or will have a primitive support environment.

We have chosen to acquire both medium- and high-powered workstations for several reasons: The first is that the multiprocessor development environment built on the workstation network should support the languages available on the multiprocessor. This is beacuse not all of our researchers work in LISP. At this point we cannot acquire a LISP machine that supports the langauge C in an integrated way. The second reason is that we needed a reasonable number of workstation nodes for use in simulating a realistic multiprocessor environment. Currently, the LISP machine is too expensive to get a reasonable number of them to be used in the development environment. The third reason is that we expect potential advantages if the processors on the workstations used for the development environment is compatible with the ones used in the multiprocessor.

The market in workstations is highly volatile, but we believe that we can acquire, by the summer of 1985, three high-powered LISP machines with color graphics at a price of $275,000; the remaining $125,000 will be used to acquire five or six medium-powered workstations with large memories. In addition, given the large amount of output data generated by simulation experiments, we will also acquire a high-speed line printer in the first year for $30,000.

Initially, we plan to use the mass storage that accompanies the individual machines on this workstation network to hold the data and programs used by the multiprocessor. In Year Three, when the use of the facility will have significantly increased, i.e., there will be more users and more complex applications, we have budgeted $100,000 for additional mass storage, another $150,000 for upgrading to the existing workstations and acquiring another LISP machine to be used for an experimenters' workstation for $75K. We anticipate that by year three, for $75K, we should be able to acquire a LISP machine with capabilities at least comparable to the ones acquired in the first year. In Year Five of the proposal, we have also budgeted another $150,000 for general upgrades to the host support structure – mass storage and experimenters' workstations.

Table 1

Procurement Schedule for Multiprocessor Facility

Year I – Host Workstation for Multiprocessor

3 3640 class LISP machines at $90-95K/workstation with color graphics	$275,000
5-6 SUN class workstations at $20-25K/workstation	$125,000
1 High-speed Line printer	$ 30,000

NSF : $400,000 University : $30,000 Total: $430,000

Year II – Multiprocessor

At least 64 processors (each 1 Mips) with 128
megabytes of memory (potential vendors: DEC or BBN)

NSF : $720,000 University : $30,000 Total: $750,000

Year III – Upgrade to Host Environment

Mass storage	$100,000
1 LISP machine	$75,000
Upgrade to workstations (purchased in Year I)	$75,000

NSF : $220,000 University : $30,000 Total: $250,000

Year IV – Upgrade to Multiprocessor

Additional processors, additional memory, and
possible acquisition of specialized processors

NSF : $220,000 University : $30,000 Total: $250,000

Year V – General Upgrades

Multiprocessor	$100,000
Mass storage	$50,000
Workstations	$100,000

NSF : $220,000 University : $30,000 Total: $250,000

FIVE-YEAR TOTAL FOR EQUIPMENT

NSF : $1,780,000 University : $150,000 Total: $1,930,000

3.2 Maintenance

Because the multiprocessor will probably not be a commercially available product, a major consideration in choosing a vendor will be the establishment of a maintenance contract that will guarantee effective servicing of the hardware. It is also necessary that the multiprocessor architecture be sufficiently modular so that a faulty processor can be removed and the system will remain operational. Additionally, the system should be composed of off-the-shelf components making replacement of a faulty component a simple task. Both the DEC and BBN multiprocessor designs satisfy these requirements and, in fact, BBN has established a service organization to begin operations in January 1985 to perform such maintenance. We have a significant amount of DEC equipment and over the years have established a close relationship with the local field-service organization. We anticipate no problems in establishing a special contract with them to handle the maintenance of the DEC multiprocessor even though it will not be a commercially available product. We also are geographically close to both BBN and DEC (less than two hours drive), making site visits, based on time-and-materials, not inordinately expensive. The maintenance of other proposed equipment workstations and mass storage devices will be put on standard vendor service agreements.

We are budgeting an average figure of 10% for equipment maintenance on the multiprocessor and 12% on the other hardware. (See Table 2 for maintenance costs per year.) In order to keep maintenance costs within this budget, we have proposed the addition of a technician to our laboratory staff. This technician will be responsible for performing preventive maintenance on the equipment, isolating hardware faults to the board level, and replacement of faulty boards. We feel that this technician will be especially valuable in assisting in the maintenance of the multiprocessor. Furthermore, we expect that, as part of the purchase agreement for the multiprocessor, some training of the technician and some spare parts for on-site repair will be included. The University, as part of its contribution to the maintenance, will be refurbishing the space necessary to house the new equipment. We expect that the cost will be on the order of $120,000.

In summary, given that the Department's Research Computing Facility has built up considerable expertise over the last five years in maintaining a very large and heterogeneous hardware network (consisting of two VAX 11/780s, ten VAX 11/750s, SUN and LISP machines), we anticipate that the maintenance of this multiprocessor facility can be effectively handled by our staff, with the addition of a new technician and an adequate maintenance budget.

Table 2

Maintenance Costs per year
without overhead

Year 1 :	$51 K
Year 2 :	$126 K
Year 3 :	$153 K
Year 4 :	$178 K
Year 5 :	$205 K

TOTAL FIVE-YEAR MAINTENANCE COSTS $714 K

3.3 Management and Support Personnel

In order to use the multiprocessor effectively, we request funding for operation staff and software development staff. The operation staff will consist of an assistant laboratory manager, a technician for hardware maintenance, and two systems programmers. The software development staff will consist of two groups, one for developing and enhancing LISP for the multiprocessor and the other for developing the multiprocessor operating system. We see both of these activities as major development activities over the entire length of the project; they will be vital its success. Both groups will be headed by a post-doctoral research associate in view of the research-oriented nature of these software development activities. In addition to a research associate, each group will contain two research assistants.

We feel confident that we have the expertise needed to carry out the software support necessary to use the multiprocessor facility effectively. We have already built a number of large software systems for the VAX including a PASCAL system superior to that developed by DEC, a full-scale LISP system that is in use at over 100 VMS/VAX sites (we are also a test-site for DEC's implementation of COMMON LISP), and a variety of low-level and high-level graphics support tools, and have developed significant network expertise by building a heterogeneous network of VAXes, SUNs, and LISP machines.

3.3.1 Management and Operations

The management of a project of this size is a major undertaking which we feel cannot be accomplished solely through a committee formed by the principal investigators and the associated faculty. We believe that one of the principal investigators (Lesser, Ramamritham, and Riseman) should serve as the project manager on a yearly rotating basis to be responsible for the overall project management. Hence we have budgeted one-half release time for one of the PI's each year. The University will cover the costs for this faculty member as part of its contribution to this grant. Aiding the project manager in his duties will be an administrative assistant and the Research Computing Facility laboratory manager. The administrative assistant will be responsible for routine administrative duties, bookkeeping, and secretarial tasks. An assistant laboratory manager, who will be responsible for the day-to-day operation of the facility, will report to the laboratory manager. The laboratory manager's salary will not be charged against this grant since the manager's position is already funded by the University. The new position of assistant laboratory manager will also be funded by the University as part of its contribution to the grant.

The operations staff will be responsible for the daily operation and maintenance of the facility, including both hardware and software for the multiprocessor and workstations. The assistant laboratory manager, whose sole responsibility will be this facility, will direct the technician and two systems programmers.

3.3.2 LISP Support

The LISP development group will be responsible for extensions to the language and modifications to the implementation of COMMON LISP to make it suitable for use on the multiprocessor. They will also be responsible for augmenting the LISP environment to provide facilities for simulating a wide range of distributed and parallel problem solving architectures.

The LISP development effort is envisioned to begin with the addition of process structuring and communication primitives to COMMON LISP to exploit the multiprocessor hardware directly. The basic approach will involve the implementation of a few powerful primitives on which higher-level capabilities can be built. An example of such an approach is the queue-based primitives of Gabriel and McCarthy [GABR84].

Building on these basic primitives, language constructs will be developed for easily specifying the creation, interaction, and deletion of LISP processes on multiple processors. The goal is to have the LISP development group provide a framework for the creation of distributed and parallel computational structures, eliminating the need for experimenters to redundantly develop their own process-management capabilities.

We also see the possibility, in later years of the project, that the LISP team will explore modification of the COMMON LISP kernel to exploit parallelism in programs at a level lower than the process level, or the development and implementation of alternative multiprocessor languages suitable for distributed problem solving at lower levels (such as the schemas used in the Laboratory for Perceptual Robotics, Concurrent Prolog, Hewitt's ACTORs, etc). Although the development of multiprocessing LISP systems has recently received some attention [GABR84;HALS84], these efforts address basic architecture and language design issues rather than the multiprocessing needs of large AI systems. We feel that the large-scale, applications-driven requirements of our research groups would provide us with a unique set of goals in the development of a low-level multiprocessing environment that is well matched to their needs.

3.3.3 Operating System Support

The multiprocessor operating-system development group will be responsible for extensions to the multiprocessor operating system that will facilitate the study of solutions to problems in scheduling, reliability, protection and debugging. The group will be involved in modifying the operating system so that researchers can not only experiment with their ideas, but also obtain performance measures. Thus, their responsibilities will include the instrumenting of the operating system to gather appropriate statistics.

We will require the operating system support group to provide facilities for

1. Studying various operating-system primitives needed to implement our algorithms for protection, reliability and concurrency;

2. Exploring different types of tightly-coupled and loosely-coupled processing architectures needed for organizing AI problems;

3. Experimenting with techniques for structuring cooperating processes using concepts such as task forces and clusters;

4. Monitoring low-level and high-level events needed to debug distributed systems;

5. Simulating various policies for resource allocation, scheduling, load balancing, and task allocation; and

6. Experimenting with various process-interconnection structures.

Work by this group will also include the building of the development environment, on top of the operating system of the networked workstations, to simulate the multiprocessor environment. As mentioned earlier, this will enable the network to be used for the testing and debugging of application and system software for the multiprocessor, which can thus go on in parallel with large simulation runs on the multiprocessor. The operating system team will also develop the interface connections between the multiprocessor operating system and the experimenters' workstations.

The University will support one research-assistant position in Year I, two research-assistant positions in Years II through IV, and three research-assistant positions in Year V.

In addition to these specific responsibilities, the systems programmers on the operation staff and the two software-development teams will contribute, where appropriate, to the development of an experimenters' workstation on the LISP machine. This effort will involve the development of an integrated tool set for displaying the dynamics of an experiment through color graphics displays, for storing and retrieving results of experiments, and for setting up, debugging, and monitoring multiprocessor runs.

Table 3 shows the personnel budget on a year-by-year basis. In the calculation of this budget, we have assumed that the entire complement of support personnel discussed previously will be needed throughout the five-year project. We have also included salary inflation increases of 6% for years two through five. The budget also contains funds for travel ($3,000/yr.), telephone ($2,400/yr.), publications

($1,000/yr.), and supplies ($10,000/yr.). The travel funds will be used mainly for site visits and conferences involving hardware acquisition and for one trip per year for each research associate. The supply budget includes normal office supplies (paper and replaceable disks and tapes for backup), plus miscellaneous hardware components such as cabling and connectors, small equipment such as modems and tools for the technician, and software tools.

Table 3

	UMass Year 1	NSF Year 1	UMass Year 2	NSF Year 2	UMass Year 3	NSF Year 3	UMass Year 4	NSF Year 4	UMass Year 5	NSF Year 5
PI	30000		31800		33700		35700		37800	0
Postdoc		74000		78400		83200		88200		93400
Prog+Tech		88000		93300		98900		104800		111000
Grads	10000	30000	21200	21200	22400	22400	23800	23800	37800	12600
Admin Assist		18000		19100		20200		21400		22700
Assist. Mgr	32000		33900		35900		38100		40400	0
Tot. Sal.	72000	210000	86900	212000	92000	224700	97600	238200	116000	239700
Fringe	7695	43289	8152	45887	8633	48653	9163	51563	9716	54617
Sal.+Fringe	79695	253289	95052	257887	100633	273353	106763	289763	125716	294317
Travel		3000		3000		3000		3000		3000
Material		10000		10000		10000		10000		10000
Pubs.		1000		1000		1000		1000		1000

3.4 University Commitment to the Facility

The University is making a major committment to assist in the research costs involved in funding this coordinated project. It is making contributions in three separate categories:

1. $150,000 of matching equipment funds.
2. $120,000 for renovations of space to house the equipment.
3. $750,000 for salaries for support personnel (1/2 release time for the Project Manager (one of the Principal Investigators), assistant laboratory manager, and ten research assistant positions over the five years); this figure includes the University overhead on these salaries.

This totals over $1,000,000 in University direct contributions for this proposal. The University has also made significant contributions to the department over the last three years in terms of increasing the faculty by eight positions, adding a number of secretary and technical support positions, significant increases in faculty salaries, and additional space for the increased size of our department. The University's commitment to the growth of the department is detailed more fully in Appendix 2.

YEAR 1

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

SUMMARY PROPOSAL BUDGET

FOR NSF USE ONLY			
PROPOSAL NO.	DURATION (MONTHS)		
	Proposed	Granted	
AWARD NO.			

ORGANIZATION University of Massachusetts/Amherst

PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR Victor R. Lesser

	NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)				$ -0-	$
1. Project Director - 1/2 release time (UMass: $30,000					
2.					
3.					
4.					
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)				-0-	
6. ( ) TOTAL SENIOR PERSONNEL (1-5)					
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				74,000	
1. (2) POST DOCTORAL ASSOCIATES				88,000	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.) (UMass: $10,000)				30,000	
3. (4) GRADUATE STUDENTS					
4. ( ) UNDERGRADUATE STUDENTS				18,000	
5. (1) SECRETARIAL-CLERICAL - Adminstrative Assistant (UMass: $32,000)				-0-	
6. (1) OTHER - Assistant Laboratory Manager (UMass: $72,000)				210,000	
TOTAL SALARIES AND WAGES (A+B) (UMass: $7,695)				43,289	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) (UMass: $79,695)				253,289	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C)					
D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION)					
Host Workstation for Multiprocessor:					
3 LISP machines with color graphics @92K/workstation				275,000	
5-6 SUN class workstations @$20K-$25K/workstation				125,000	
1 High Speed Line Printer @$30K					
TOTAL PERMANENT EQUIPMENT (UMass: $30,000				400,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				3,000	
2. FOREIGN					
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $					
2. TRAVEL					
3. SUBSISTENCE					
4. OTHER					
TOTAL PARTICIPANT COSTS				-0-	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				10,000	
2. PUBLICATION COSTS/PAGE CHARGES				1,000	
3. CONSULTANT SERVICES					
4. COMPUTER (ADPE) SERVICES					
5. SUBCONTRACTS					
6. OTHER - Maintenance: $51,000 (UMass: $120,000); telephone: $2,400				53,400	
TOTAL OTHER DIRECT COSTS (UMass: $120,000)				64,400	
H. TOTAL DIRECT COSTS (A THROUGH G) (UMass: $229,695)				720,689	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)				153,931	
TOTAL INDIRECT COSTS (UMass: $38,253)				153,931	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I) (UMass: $267,948)				874,620	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)					
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 874,620	$

PI/PD TYPED NAME & SIGNATURE*	DATE 9/14/84	FOR NSF USE ONLY		
		INDIRECT COST RATE VERIFICATION		
INST. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date of Rate Sheet	Initials - DGC
				Program

*SIGNATURES REQUIRED ONLY FOR REVISED **BUDGET (GPM 233)**

NOTE: University contribution is subject to availability of funds.

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

SUMMARY
PROPOSAL BUDGET

YEAR 2

ORGANIZATION University of Massachusetts/Amherst	FOR NSF USE ONLY PROPOSAL NO.	DURATION (MONTHS) Proposed	Granted
PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR Victor R. Lesser	AWARD NO.		

	NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)					
1. Project director - 1/2 release time (UMass: $31,800)				$ -0-	$
2.					
3.					
4.					
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)					
6. ( ) TOTAL SENIOR PERSONNEL (1-5)				-0-	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (2) POST DOCTORAL ASSOCIATES				78,400	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				93,300	
3. (4) GRADUATE STUDENTS (UMass: $21,200)				21,200	
4. ( ) UNDERGRADUATE STUDENTS					
5. (1) SECRETARIAL-CLERICAL - Administrative Assistant				19,100	
6. (1) OTHER - Assistant Laboratory Manager (UMass: $33,900)				-0-	
TOTAL SALARIES AND WAGES (A+B) (UMass: $86,900)				212,000	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) (UMass: $8,152)				45,887	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C) (UMass: $95,052)				257,887	
D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION) Multiprocessor: at least 64 processors of 1 mip speed with 128 megabytes of memory (possible vendors DEC/BBN) $750,000					
TOTAL PERMANENT EQUIPMENT (UMass: $30,000)				720,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				3,000	
2. FOREIGN					
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 2. TRAVEL 3. SUBSISTENCE 4. OTHER					
TOTAL PARTICIPANT COSTS				-0-	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				10,000	
2. PUBLICATION COSTS/PAGE CHARGES				1,000	
3. CONSULTANT SERVICES					
4. COMPUTER (ADPE) SERVICES					
5. SUBCONTRACTS					
6. OTHER - Maintenance: $126,000; telephone: $2,400				128,400	
TOTAL OTHER DIRECT COSTS				139,400	
H. TOTAL DIRECT COSTS (A THROUGH G) (UMass:$125,052)				1,120,287	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)				192,138	
TOTAL INDIRECT COSTS (UMass:$45,624)				192,138	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I) (UMass:$170,676)				1,312,415	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)					
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$1,312,415	$

PI/PD TYPED NAME & SIGNATURE* Victor R. Lesser	DATE 9/14/84	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION Date Checked / Date of Rate Sheet / Initials - DGC
INST. REP. TYPED NAME & SIGNATURE*	DATE	Program

*SIGNATURES REQUIRED ONLY FOR REVISED BUDGET (GPM 233)

NOTE: University contribution is subject to availability of funds.

YEAR 3

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

SUMMARY PROPOSAL BUDGET

ORGANIZATION University of Massachusetts/Amherst	FOR NSF USE ONLY		
	PROPOSAL NO.	DURATION (MONTHS) Proposed	Granted
PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR Victor R. Lesser	AWARD NO.		

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)	NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
1. Project director - 1/2 release time (UMass: $33,700)				$ -0-	$
2.					
3.					
4.					
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)					
6. ( ) TOTAL SENIOR PERSONNEL (1-5)				-0-	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (2) POST DOCTORAL ASSOCIATES				83,200	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				98,900	
3. (4) GRADUATE STUDENTS (UMass: $22,400				22,400	
4. ( ) UNDERGRADUATE STUDENTS					
5. (1) SECRETARIAL-CLERICAL - Administrative Assistant				20,200	
6. (1) OTHER - Assistant Laboratory Manager (UMass: $35,900)				-0-	
TOTAL SALARIES AND WAGES (A+B) (UMass: $92,000)				224,700	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) (UMass: $8,633)				48,653	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C) (UMass: $100,633)				273,353	
D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION) Upgrade to Host Environment: Mass storage $100,000 1 LISP machine $75,000 Upgrade to workstations (purchased in Year I) $75,000					
TOTAL PERMANENT EQUIPMENT (UMass: $30,000)				220,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				3,000	
2. FOREIGN					
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 2. TRAVEL 3. SUBSISTENCE 4. OTHER					
TOTAL PARTICIPANT COSTS				-0-	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				10,000	
2. PUBLICATION COSTS/PAGE CHARGES				1,000	
3. CONSULTANT SERVICES					
4. COMPUTER (ADPE) SERVICES					
5. SUBCONTRACTS					
6. OTHER - Maintenance: $153,400; telephone: $2,400				155,800	
TOTAL OTHER DIRECT COSTS				166,800	
H. TOTAL DIRECT COSTS (A THROUGH G) (UMass: $130,633)				663,153	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)				212,713	
TOTAL INDIRECT COSTS (UMass: $48,303)				212,713	
J. TOTAL DIRECT AND INDIRECT COSTS (H+I) (UMass: $178,936)				875,866	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)					
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 875,866	$

PI/PD TYPED NAME & SIGNATURE* Victor R. Lesser	DATE 9/14/84	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION
INST. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date of Rate Sheet / Initials - DGC / Program

*SIGNATURES REQUIRED ONLY FOR REVISED **BUDGET (GPM 233)**

NOTE: University contribution subject to availability of funds

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

YEAR 4

SUMMARY PROPOSAL BUDGET

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (MONTHS) Proposed	Granted
AWARD NO.		

ORGANIZATION University of Massachusetts/Amherst

PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR Victor R. Lesser

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)		NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
1. Project Director – 1/2 release time	(UMass: $35,700)				$ -0-	$
2.						
3.						
4.						
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)						
6. ( ) TOTAL SENIOR PERSONNEL (1-5)					-0-	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)						
1. (2) POST DOCTORAL ASSOCIATES					88,200	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)					104,800	
3. (4) GRADUATE STUDENTS	(UMass: $23,800)				23,800	
4. ( ) UNDERGRADUATE STUDENTS						
5. (1) SECRETARIAL-CLERICAL – Administrative Assistant					21,400	
6. (1) OTHER – Assistant Laboratory Manager	(UMass: $38,100)				-0-	
TOTAL SALARIES AND WAGES (A+B)	(UMass: $97,600)				238,200	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)	(UMass: $9,163)				51,563	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C)	(UMass: $106,763)				289,763	
D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION) Upgrade to Multiprocessor: Additional processors, memory, and possibly additional specialized processors $250,000						
TOTAL PERMANENT EQUIPMENT	(UMass: $30,000)				220,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)					3,000	
2. FOREIGN						
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 2. TRAVEL 3. SUBSISTENCE 4. OTHER						
TOTAL PARTICIPANT COSTS					-0-	
G. OTHER DIRECT COSTS						
1. MATERIALS AND SUPPLIES					10,000	
2. PUBLICATION COSTS/PAGE CHARGES					1,000	
3. CONSULTANT SERVICES						
4. COMPUTER (ADPE) SERVICES						
5. SUBCONTRACTS						
6. OTHER – Maintenance: $178,400; Telephone: $2,400					180,800	
TOTAL OTHER DIRECT COSTS					191,800	
H. TOTAL DIRECT COSTS (A THROUGH G)	(UMass: $136,763)				704,563	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)					232,590	
TOTAL INDIRECT COSTS	(UMass: $51,246)				232,590	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)	(UMass: $188,009)				937,153	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)						
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)					$ 937,153	$

PI/PD TYPED NAME & SIGNATURE* Victor R. Lesser	DATE 9/14/84	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION
INST. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date of Rate Sheet / Initials - DGC / Program

*SIGNATURES REQUIRED ONLY FOR REVISED **BUDGET (GPM 233)**

NOTE: University contribution is subject to availability of funds.

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

SUMMARY PROPOSAL BUDGET

YEAR 5

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (MONTHS) Proposed	Granted
AWARD NO.		

ORGANIZATION: University of Massachusetts/Amherst

PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR: Victor R. Lesser

	NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)				$ -0-	$
1. Project Director - 1/2 release time (UMass: $37,800					
2.					
3.					
4.					
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)					
6. ( ) TOTAL SENIOR PERSONNEL (1-5)				-0-	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (2) POST DOCTORAL ASSOCIATES				93,400	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				111,000	
3. (4) GRADUATE STUDENTS (UMass: $37,800)				12,600	
4. ( ) UNDERGRADUATE STUDENTS					
5. (1) SECRETARIAL-CLERICAL - Administrative Assistant				22,700	
6. (1) OTHER - Assistant Laboratory Manager (UMass: $40,400)				-0-	
TOTAL SALARIES AND WAGES (A+B) (UMass: $116,000)				239,700	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) (UMass: $9,716)				54,617	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C) (UMass: $125,716)				294,317	

D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION)

General Upgrades

Multiprocessor	$100,000
Mass storage	$50,000
Workstations	$100,000

	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
TOTAL PERMANENT EQUIPMENT (UMass: $30,000)	220,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)		
2. FOREIGN		
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 2. TRAVEL 3. SUBSISTENCE 4. OTHER		
TOTAL PARTICIPANT COSTS	-0-	
G. OTHER DIRECT COSTS		
1. MATERIALS AND SUPPLIES	10,000	
2. PUBLICATION COSTS/PAGE CHARGES	1,000	
3. CONSULTANT SERVICES		
4. COMPUTER (ADPE) SERVICES		
5. SUBCONTRACTS		
6. OTHER - Maintenance: $205,100; Telephone: $2,400	207,500	
TOTAL OTHER DIRECT COSTS	218,500	
H. TOTAL DIRECT COSTS (A THROUGH G) (UMass: $155,716)	735,817	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)	247,592	
TOTAL INDIRECT COSTS (UMass: $60,343)	247,592	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I) (Umass: $216,059)	983,409	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)		
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)	$ 983,409	$

PI/PD TYPED NAME & SIGNATURE* Victor R. Lesser	DATE 9/14/84	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION
INST. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date of Rate Sheet / Initials - DGC / Program

*SIGNATURES REQUIRED ONLY FOR REVISED **BUDGET (GPM 233)**

NOTE: University contribution is subject to availability of funds.

TOTAL FIVE-YEAR BUDGET

(SEE INSTRUCTIONS ON REVERSE BEFORE COMPLETING)

SUMMARY PROPOSAL BUDGET

ORGANIZATION University of Massachusetts/Amherst	FOR NSF USE ONLY PROPOSAL NO.	DURATION (MONTHS) Proposed	Granted
PRINCIPAL INVESTIGATOR/PROJECT DIRECTOR Victor R. Lesser	AWARD NO.		

	NSF FUNDED PERSON-MOS CAL.	ACAD	SUMR	FUNDS REQUESTED BY PROPOSER	FUNDS GRANTED BY NSF (IF DIFFERENT)
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title; A.6. show number in brackets)				$ -0-	$
1. Project Director - 1/2 release time (UMass: $169,000)					
2.					
3.					
4.					
5. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET EXPLANATION PAGE)				-0-	
6. ( ) TOTAL SENIOR PERSONNEL (1-5)					
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				417,200	
1. (2) POST DOCTORAL ASSOCIATES				496,000	
2. (3) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.) (UMass: $115,200)				110,000	
3. (4) GRADUATE STUDENTS					
4. ( ) UNDERGRADUATE STUDENTS				101,400	
5. (1) SECRETARIAL-CLERICAL – Administrative Assistant (UMass: $180,300)				-0-	
6. (1) OTHER – Assistant Laboratory Manager (UMass: $464,500)				1,124,600	
TOTAL SALARIES AND WAGES (A+B) (UMass: $43,359)				244,009	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) (UMass: $507,859)				1,368,609	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C)					
D. PERMANENT EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $1,000; ITEMS OVER $10,000 REQUIRE CERTIFICATION)					
TOTAL PERMANENT EQUIPMENT (UMass: $150,000)				1,780,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				15,000	
2. FOREIGN					
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ _____ 2. TRAVEL _____ 3. SUBSISTENCE _____ 4. OTHER _____					
TOTAL PARTICIPANT COSTS				-0-	
G. OTHER DIRECT COSTS				50,000	
1. MATERIALS AND SUPPLIES				5,000	
2. PUBLICATION COSTS/PAGE CHARGES					
3. CONSULTANT SERVICES					
4. COMPUTER (ADPE) SERVICES					
5. SUBCONTRACTS					
6. OTHER – Maintenance: $713,900; (UMass: $120,000); telephone: $12,000				725,900	
TOTAL OTHER DIRECT COSTS (UMass: $120,000)				780,900	
H. TOTAL DIRECT COSTS (A THROUGH G) (UMass: $777,859)				3,944,509	
I. INDIRECT COSTS (SPECIFY) 48% modified total direct costs (equipment not included)				1,038,964	
TOTAL INDIRECT COSTS (UMass: $243,769)				1,038,964	
J. TOTAL DIRECT AND INDIRECT COSTS (H+I) (UMass: $1,021,628)				4,983,473	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS GPM 252 AND 253)					
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$4,983,473	$

PI/PD TYPED NAME & SIGNATURE* Victor R. Lesser	DATE 9/14/84	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
INST. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date of Rate Sheet	Initials • DGC Program

*SIGNATURES REQUIRED ONLY FOR REVISED BUDGET (GPM 233)

NOTE: University contribution is subject to availability of funds.

5. LIST OF FACULTY

Michael A. Arbib

Andrew G. Barto

Daniel D. Corkill *

Janice E. Cuny

Allen R. Hanson

Leslie J. Kitchen *

James F. Kurose

Victor R. Lesser

John Eliot B. Moss

Krithivasan Ramamritham

Edward M. Riseman

David W. Stemple

Jack C. Wileden

Walter H. Kohler (ECE)

John A. Stankovic (ECE)

Donald F. Towsley (ECE)

* refers to Research Associates

6. RESEARCH IN COOPERATIVE DISTRIBUTED COMPUTING

6.1 Overview of Proposed Research

The Computer and Information Science Department, with colleagues from the Electrical and Computer Engineering Department, at the University of Massachusetts, Amherst, proposes a coordinated research project on *cooperative distributed computing* based on a tightly-coupled multiprocessor containing between 64 and 128 processing elements. We believe that the proposed *tightly-coupled*, large-scale multiprocessor is the most *cost-effective* means for performing the simulations of the *tightly-coupled* and *loosely-coupled* distributed software and hardware architectures needed to solve problems in cooperative distributed computing. In order to fully exploit the capabilities of the multiprocessor, we also request funds in two additional categories. The first is for a network of workstations to support the development, monitoring, and analysis of the multiprocessor experiments. The second is for the support staff needed to develop and maintain the necessary software tools and to be responsible for the operation of the facility.

This project will coordinate and integrate large active research groups in the areas of AI (computer vision, robotics and distributed problem solving, adaptive learning networks, neural modelling) and of Distributed and Parallel Processing (language primitives, synchronization, concurrency control, network communications, real-time scheduling, load balancing, debugging, protection and reliability). The integration of these apparently diverse, but already strongly interacting, research areas, together with the use of the term *cooperative distributed computing*, rather than *distributed processing* indicate what we feel is the unique focus of this coordinated research project. A key aspect of this project will be to address issues that arise out of task and problem solving decomposition in which processes cooperate to solve *a single integrated problem* rather than *a set of independent tasks.* In this type of task decomposition, the major reason for process interaction is *cooperation*, which occurs through the exchange of partial results, rather than *contention*, which occurs mainly in accessing resources. (Henceforth, in this proposal, we use the term *distributed computing* to encompass *parallel as well as distributed computing.*)

The phenomenon of cooperation occurs at many levels and we have already begun investigating cooperative distributed computing architectures at a number of different levels:

- Application Level: distributed sensor networks [Section 6.4.1], distributed planning [Section 6.4.3], distributed inference networks for controlling problem solving in high-level vision [Section 6.4.2], distributed planning and control for perceptual robots [Section 6.4.3], and parallel blackboard architectures [Section 6.4.1].

For example, in the distributed interpretation task for sensor networks, we have been examining a problem solving architecture composed of a network of semi-autonomous Hearsay-II-like system nodes. Each node processes an incomplete and possibly inaccurate view of the environment and they cooperate to converge to a *globally consistent* interpretation. This cooperation occurs via an iterative, *coroutine-like* exchange of high-level results which may be tentative and partial.

- Operating System Level: scheduling and distributed control [Section 6.5.1], protection [Section 6.5.4], synchronization and concurrency control [Section 6.5.2], debugging [Section 6.5.6], failure tolerance, detection, and recovery [Section 6.5.2], and modelling performance evaluation [Section 6.5.7].

For example, in our scheduling algorithm for real-time systems, the scheduler components on individual nodes cooperate in order to determine the node on which a newly arriving task is to be executed. Such cooperation is needed because no node has complete knowledge of the whole system: in particular, no node has complete knowledge of the load on individual nodes. In our algorithm the interaction among multiple nodes is based on a bidding approach.

- Network Communication Level: resource allocation in computer networks, [Section 6.5.3], and network protocols [Section 6.6].

For example, in our approach to decentralized resource allocation mechanisms in computer networks, network agents act as selfish, utility-maximizing entities and their cooperation via a resource pricing mechanism serves as a decentralized computational device for determining an optimal distribution of network resources. The algorithm itself is an iterative process of local (and hence parallel) optimizations followed by a partial information exchange via the pricing mechanism.

- Low-Level: low-level vision [Section 6.4.2], adaptive learning networks [Section 6.4.4], and neural modelling [Section 6.4.5].

For example, the nodes in adaptive learning neural networks are "self-interested" agents that possess means for furthering their interests (defined by a preference structure on their inputs) when interacting with a wide variety of environments. When such nodes interact as members of a network they form interconnections among themselves, and with externally supplied signals, that permit each node to perform better with respect to its own preferences than it would by acting independently of other nodes. The nodes cooperate to form stable "coalitions" to the extent that such collective behavior furthers the interests of the constituents.

Our investigations of cooperative distributed computing will also span the spectrum of levels from loosely-coupled, large-scale process decompositions to tightly-coupled, fine-grain process decompositions (see Section 6.2).

The coordination of our research efforts in AI and distributed processing, which has already begun on a number of different projects, naturally arises out of common themes in both areas. From the AI viewpoint, the distributed processing perspective underlies many of our approaches to organizing AI problem solving systems. This occurs because a distributed architecture is a natural way to organize the complex interactions necessary for effective cooperation among diverse sources of knowledge. A distributed implementation is also important for problem solving in such areas as vision, robotics and sensor networks, where there is a natural spatial distribution of information or effectors. In addition to capitalizing on this natural distribution, AI has also begun to use the distributed perspective as a means to decompose, structure, organize, and implement complex and computationally intensive knowledge-based systems. Thus, distributed processing issues such as synchronization, scheduling, load balancing, and operating system primitives, are also pertinent for AI.

In turn, the distributed processing community has become interested in AI research since heuristic techniques have been found to play a vital role in the design and efficient use of distributed processing systems. The use of heuristic techniques in distributed processing has arisen out of a need to deal with uncertainty caused by distribution. In many distributed situations, it is impossible to efficiently design distributed algorithms which assume that each process has a consistent, up-to-date and complete view of the state of other processes in the system. AI tasks also serve as interesting applications for distributed processing because they involve large process structures designed to solve a single problem and, in these tasks, cooperation rather than contention among processes plays the predominant role.

In the applications that we propose to study, cooperation is necessitated by the need to function in spite of the uncertainty that prevails in the system. Some of the uncertainties are inherent in the sensory information; others are introduced by the lack of complete and accurate state information in any distributed system and by the delays involved in communication across a network. Currently, the phenomena that occur in such systems with large number of cooperating nodes are not well understood. In addition, in the small number of instances where solutions have been proposed, they have not been evaluated owing to the lack of computational resources.

The proposed coordinated research project centered on a large-scale multiprocessor is motivated by our desire to address issues of cooperation and uncertainty in distributed systems with large numbers of nodes. The multiprocessor will facilitate this research by providing:

1) *A resource for the study of large and complex tightly-coupled and loosely-coupled distributed systems,*

2) *A realistic environment for research in hardware and software architectures to support distributed computation,* and

3) *The infrastructure that will facilitate the needed integration between the disciplines of Distributed Computing and AI, especially in the area of knowledge-based systems.*

Let us briefly examine each of these points:

Only in large distributed systems do the problems of cooperation and coherence and the effects of organizational structure dominate. Whereas most of our current work on cooperative distributed computing is based on empirical studies using testbeds or simulation models, lack of computer power for these empirical studies has significantly limited the range of experiments, especially with large networks. For instance, the lack of computational resources places artificial restrictions on the number of nodes in our experiments in distributed problem solving experiment, on the size of the networks simulated in our work on distributed resource allocation, on the size of the network and the nature of tasks in our simulation of distributed algorithms for scheduling and concurrency control, and on the study of parallel architectures and algorithms for vision and motion computation.

The quantitative increase in computing cycles available for these experiments through the proposed facility will change dramatically the size and complexity of the systems that can be explored experimentally. Currently, the maximum effective computing cycles that can be obtained from our loosely-coupled network of VAXes (i.e., ten VAX 11/750s and two VAX 11/780s) to perform a distributed simulation is less than 10 Mips. In fact, the maximum is only appropriate for simulating loosely-coupled and large-grained systems. In more tightly-coupled distributed architectures, the maximum is less than 1 Mips. This is in sharp contrast to the requirements of our experiments which are at least of the order of 100 Mips (see Section 6.2). We believe that the increase in computing power will in turn lead to a qualitative change in the nature of the phenomena that can be studied and the statistical validity of our results.

Rarely have the ideas in distributed computing been tested by large applications. In many cases, distributed databases have been the sole application area. We expect that the proposed emphasis on *cooperative distributed computing*, the challenge provided by large AI applications, and the presence of the proposed multiprocessor will provide the environment for evaluating effectively our ideas in distributed computing as well as in distributed hardware and software architectures. This will occur through extensions and modifications to the multiprocessor operating system to explore ideas in scheduling, protection, debugging, high-level process structuring, language constructs for synchronization and reliability, and performance analysis. This work will result in the development of a kernel appropriate for the implementation of cooperative distributed computing systems.

This proposed facility will also encourage and permit us to undertake new coordinated projects that are aimed at studying long-term and fundamental issues that arise as the disciplines of AI and distributed systems are more closely integrated. For example, we anticipate an integrated effort in the areas of distributed problem solving and distributed load balancing and scheduling. This arises not only because load balancing and task scheduling are issues that need to be resolved in the context of distributed problem solving but also because the nature of interactions between nodes in such systems are similar to those found in distributed load balancing and scheduling. In addition, both have to deal with the uncertainties that are inherent in a dynamic environment and hence have to adopt heuristic knowledge-based techniques. We also expect that there will be similar interactions with recently initiated projects on distributed planning for robotics, information fusion of high-level vision data through a parallel inference-net structure, and parallel processing architectures for blackboard problem solving systems as in Hearsay-II-like architectures.

In summary, we believe that the proposed multiprocessor facility

- *Will provide the infrastructure needed to experiment fully with our current ideas in cooperative distributed computing.*
- *Will extend our current research initiatives and open up new research directions that become apparent as we study large and complex distributed systems.*
- *Will facilitate the needed integration between the disciplines of AI and distributed computing.*

The remainder of this section is structured as follows: In Sections 6.2 and 6.3, we summarize the scientific aspects of this proposal, explain why our current facilities are inadequate for addressing these issues, and discuss why the proposed facility is the most cost-effective way to address these issues. In Sections 6.4 and 6.5, we describe the work of our researchers in AI and distributed computing and show how they will benefit by the acquisition of the large-scale multiprocessing system. In Section 6.6, we briefly describe our work on more formal and conceptual issues that relate to the theme of this proposal. Detailed descriptions of the hardware acquisition and support personnel are contained in Section 3.

6.2 Scientific Issues

The common theme underlying the research projects that will be utilizing the multiprocessor facility is *cooperative distributed computing*. Cooperative distributed computing is an important research area for several reasons. First, hardware technology has advanced to the point where the construction of large distributed and parallel computing networks is not only possible, but economically feasible. While the first networks may consist of only a small number of nodes (on the order of ten to one hundred), networks may eventually contain hundreds or thousands of individual nodes. We are nearing a situation of exciting hardware possibilities unaccompanied by the software technology required for their effective utilization. Second, there are many applications that are inherently spatially distributed. A distributed architecture that matches their spatial distribution offers many advantages over a centralized approach. Third, understanding the process of cooperative problem solving is an important goal in its own right. Whether the underlying system is societal, managerial, biological, or mechanical, we seem to understand competition far better than cooperation. It is possible that the development of cooperative distributed computing networks may serve the same validating role to theories in sociology, management, organizational theory, and biology as the development of AI systems have served to theories of problem solving and intelligence in psychology and philosophy.

Cooperative distributed computing systems differ from distributed processing systems in both the style of distribution and the type of problems addressed [SMIT81]. These differences are most apparent when we study the interactions among nodes in each of the networks. A distributed-processing network typically has multiple, disparate tasks executing concurrently in the network. Shared access to physical or informational resources is the main reason for interaction among tasks. The goal is to preserve the illusion that each task is executing alone on a dedicated system by having the network operating system hide the resource-sharing interactions and conflicts among tasks in the network. In contrast, the nodes in a cooperative distributed computing. system work together to solve a single, integrated problem and are often aware of the distribution of network components and can make informed interaction decisions based on this information. This difference in emphasis is, in part, due to the characteristics of the applications being tackled by conventional distributed processing methodologies. These applications have permitted task decompositions in which a node rarely needs the assistance of another node in carrying out its problem solving function. Thus, most of the research as well as the paradigms of distributed processing do not directly address the issues of cooperative interactions of tasks to solve a single problem. Nevertheless, even in such systems, to execute system tasks such as load balancing and scheduling, the system components on individual nodes must to cooperate in order to perform these tasks intelligently.

For many problems that seem naturally suited to a distributed network, highly cooperative task interaction is a requirement. In these applications, communication delay makes it impractical for the network to be structured so that each node has the relevant up-to-date information needed for its local computations and control decisions. Another way of viewing this problem is that the spatial decomposition of information among the nodes is ill-suited to a functionally distributed solution. Each node may possess the information necessary to perform a portion of each function, but insufficent information to perform any function completely. In this situation, cooperation among nodes is imperative.

In order to realize the benefits of cooperative distributed computing, the following scientific issues need to be addressed:

1. *Techniques for partitioning, assigning, and reassigning tasks* to maximize parallelism, to minimize communication, and to permit graceful degradation in the face of individual component failure. We are also studying ways to redistribute tasks in a distributed system dynamically so as to maintain high performance and reliability as computational demands, processing resources, and communication resources vary.

2. *Strategies for effective cooperation among nodes increase coherence* in the presence of partial, incoherent, and incomplete information. We are pursuing the development of strategies that permit nodes to cooperate in order to stop the spread of incorrect results generated by nodes with inconsistent or incomplete information We are also exploring whether these same strategies for cooperation can handle incorrect results caused by faulty processing nodes or communication channels.

3. *Mechanisms for obtaining organized behavior using decentralized control.* Inherent in the control of a distributed system is the problem of maintaining an accurate model of the entire system when there is no global mechanism for gathering and acting on the current state of system performance. We are developing control algorithms that operate satisfactorily (e.g., are stable and sufficiently accurate) even when the available information about the system is inconsistent or incomplete. We are also studying ways to address reliability questions

as part of the general issue of decentralized control.

4. *Approaches for distributed operating systems to accomodate cooperative distributed computing.* The operating systems will need to employ distributed, cooperative algorithms and will require decentralized system-wide control of resources for the cooperative execution of application programs. Moreover, operating system components must operate with missing, incomplete, delayed, or erroneous state information.

5. *Techniques for high-level process decomposition aid in the solution of large AI tasks.* We are trying to understand the nature of search in these tasks to develop distributed problem solving architectures that will exploit the effective parallelism. We are exploring whether there are high-level AI tasks that can be structured at the process level to exhibit high degrees of effective parallelism (30 to 100 processes) despite the need for cooperative interaction. We are looking at how efficient the search is, the nature of information shared among different parallel search paths, and the nature of coordination among paths.

6. *Language and operating system primitives needed to support cooperative distributed and parallel computing.* In our study of programming languages, our goal is not to design a specific programming environment but rather to investigate the necessary programming abstractions and compilation techniques needed for efficient parallel computing. Our study of operating system primitives includes issues raised by requirements of efficient interprocess communication, shared resource protection, and system reliability.

In order to take new approaches to addressing these issues, problems need to be solved at different levels in the system; at the highest level in terms of cooperating application processes to a much finer and closely coupled interaction at the low-level. Below we describe our work at each of the levels. The descriptions outline the major thrust of the different research projects. The specifics of these projects are available in the indicated sections.

1. The development of techniques for organizing *loosely-coupled* and large-grained distributed computing structures.

   – We are exploring distributed interpretation architectures in which each node is a sophisticated problem solving system that can modify its behavior as circumstances change and plan its own communication and cooperation strategies with other nodes. A major aspect of this research is the use of meta-level control for network coordination. We use the Distributed Vehicle Monitoring Testbed (DVMT) to evaluate alternative approaches empirically. (See Section 6.1.1.)

- We are developing distributed scheduling algorithms for real-time systems, where cooperation occurs among the scheduler components on individual nodes. Application tasks are scheduled based on their deadlines as well as their computational and resource requirements. Determination of the node to which a task is allocated is dependent on the bids sent by the scheduler components on each node. Crucial questions in this regard relate to the nature and amount of state information that needs to be exchanged among nodes, the policy for selecting the best bidder, stability of the system, and the inclusion of system overheads. (See Section 6.5.1.)

- We are developing protection techniques applicable to distributed systems. It is aimed at developing schemes with reduced overheads; the overheads are necessarily incurred when accesses to protected shared resources are checked. We are examining the protocols needed for kernels on individual nodes to cooperate in order to ensure that a process accessing a resource possesses the necessary privileges. We are evaluating the efficiency of a *port-based protection scheme*, whereby a port, i.e., a communication channel, can be created between two processes only if one process has privileges to access a resource managed by the other. (See Section 6.5.4.)

- We are developing decentralized resource allocation mechanisms for computer networks based on concepts from mathematical economics. Here, network agents cooperate, via a pricing mechanism, to ensure optional distribution of network resources. This method, which has already been applied to several channel sharing policies in single-hop multiple access networks, is being extended to multi-hop packet radio networks. Several alternative "pricing" mechanisms are also under investigation. (See Section 6.5.3.)

2. The development of techniques for *tightly-coupled* and *medium-grained* distributed computing structures.

   - We are studying the knowledge and data partitioning provided by the cooperating knowledge source style of problem solving pioneered by Hearsay-II which were developed, in part, with an eye towards multiprocessor implementation. Although there is considerable potential in a multiprocessor "blackboard" machine, there has been little research in that direction. We will be exploring alternative tightly-coupled architectures as possible implementations for such systems using a modified version of Distributed Vehicle Monitoring Testbed. (See Section 6.4.1.)

- We are developing a knowledge-based image interpretation system using a collection of intercommunicating structured processes, called schemas, which run in a parallel fashion. As part of the inference mechanism to control this system, there is a network structure (called the inference net) in which each node in the net represents a proposition. The computations of belief in these propositions are designed so that they can be performed in parallel throughout the network as new information is asynchronously obtained. (See Section 6.4.2.)

- We are developing algorithms for distributed planning for robot control. We are studying the possible patterns of competition and cooperation between subprocesses, which will yield an appropriate pattern of action -- without an overall plan necessarily being encoded at any one node within the system. (See Section 6.4.3.)

3. The development of techniques for organizing *tightly-coupled* and *fine-grained* distributed computing structures.

   - We are studying fine-grained networks consisting of units that are relatively simple in their input/output behavior but implement relatively sophisticated stochastic learning algorithms. (See Section 6.4.4.)

   - We are developing cooperative computation models of neural development and plasticity. To accomplish this, we are studying the frog brain as a focus for analyzing neural mechanisms in visuomotor coordination at a level of detail that allows design and simulation of detailed experiments in neurophysiology, neuroanatomy and animal behavior. (See Section 6.4.5.)

   - We are developing parallel algorithms for low-level vision processing. Information is extracted from the image (in both the static and motion cases) by means of a hierarchically organized parallel structure called the processing cone and linked to the semantic structures resident in the knowledge base. (See Section 6.4.1.)

In addition to the work described above, our research on language primitives for distributed and parallel systems (Section 6.5.5), synchronization and concurrency control (Section 6.5.2), and debugging (Section 6.5.6) are relevant to all these levels and apply to both coarse-grained and fine-grained computations.

The approaches to these research problems vary greatly among the individual researchers. However, as the next section shows, all these approaches have a substantial empirical component to them and involve computationally intensive experimentation.

6.3 Computational Issues

In this section, we first discuss the tremendous computing requirements of our research studies in cooperative distributed computing. These computing requirements come not only from the computational needs of running a single experiment, but also from the type of interactive experimentation demanded by our research methodology. This methodology involves the generation and analysis of a large set of experimental runs (obtained by varying parameter settings and problem solving situations) to explore a particular idea. An important part of this experimental methodolody is the ability to quickly explore alternative parameter settings based on the results of previous experiments.

Following our discussion of our computing requirements, we examine our current departmental research computing facility and show how it is inadequate for meeting the computational requirements of our research. Finally, we argue that a large-scale, tightly-coupled multiprocessor system is the most cost-effective means for meeting these computational requirements.

As shown in the previous section and discussed in further detail in the sections on specific research projects, we are exploring fundamental scientific issues in cooperative distributed computing 1) by building large and complex application systems in a simulated cooperative distributed environment (in the areas of vision, robotics, sensor networks, adaptive learning, and neural networks), and 2) by developing testbeds in which cooperative distributed algorithms for implementing distributed operating system functions (scheduling, concurrency control, and network communications) can be evaluated. Both approaches to studying cooperative distributed systems are, by their very nature, computationally intensive. Each requires a fine grained simulation in which the exact details of what information is exchanged among nodes and how the node reacts to the information is preserved. This level of detail (where a change in a single piece of information can significantly affect the system) is crucial to understanding the performance of cooperative distributed algorithms. Additionally, there is need to simulate these systems and algorithms for long time periods in environments which contain large numbers of nodes. This type of environment is both more realistic (a 100-node sensor system instead of a 5-node system or an adaptive network of hundreds of elements instead of ten elements) but also intrinsically more interesting. Investigations of systems of this size will potentially expose new issues in cooperative distributed computing and will demand understanding what forms of coordination are needed to obtain effective cooperation among a large number of nodes operating with uncertain state information.

Examples of the extensive computational requirements of our research can be seen in the research in distributed problem solving and in the work in vision and motion analysis on parallel processing architectures. The Distributed Vehicle Monitoring Testbed (DVMT) requires 0.5-4 hours of elasped time using 4 dedicated 750s to simulate a 5-10 node network observing 10 time frames of sensory data with little noise. The computational requirements of our simulations result from modelling the detailed activities that each node would perform in an actual distributed problem solving network. Each simulated node generates and communicates with other nodes individual hypotheses and problem solving goals with specific attributes (including belief values and ratings). Our intuition at the beginning of this research was that a significant change in a single hypothesis or goal could lead to significantly different performance results. Abstract simulation of these distributed problem solving networks would not capture such detailed, yet significant, phenomena. Experiments with the Testbed have confirmed our intuition.

Experiments where approximately 100 nodes observe significantly more time frames of noisy sensor data are required to explore many of the cooperation and global coherence problems associated with plausible-sized distributed problem solving networks. Even if all 12 of the department's research VAXes were dedicated to DVMT simulations it would still be impossible to extensively simulate realistic sized networks (*i.e.*, at least 15 hours of network time would be required for a *single* experiment).

Similar computational requirements characterize the research in cooperative distributed architectures for low-level vision processing and for high-level image interpretation tasks. Developing a complete interpretation of an image involves region segmentation and straight line extraction, construction of the intermediate representation composed of the regions, lines, and their features, followed by schema-driven invocation of interpretation rules and execution of the inference net. All of these algorithms are executed on a simulated cooperative distributed architecture. The following table indicates representative execution times for various componenets of the system. The actual CPU time was obtained from current experiments and then projected to higher resolution imagery. Although the schema system and inference net are not pixel-based, higher resolution images produce a larger number of primitives in the intermediate representation and hence a larger number of execution cycles.

		Estimated
	CPU/Hours	CPU/Hours
	(Pixels)	(512x512)
Low Level:		
Motion Analysis	1/2 (128x128)	4
Line Analysis	1/3 (256x256)	1-1/3
Region Analysis	4 (256x256)	16
High-Level:		
Inference Network	8 (256x256)	16*
Schema System	9 (256x256)	18*

* These high level algorithms are not directly affected by the number of pixels but rather by the number of regions and the level of detail desired in the interpretation.

We estimate that an integrated system that performs the entire interpreation of a single 512x512 image would take 50-60 hours of VAX 11/750 CPU time. Processing of motion data would require an order of magnitude greater computing power. Given this turnaround, it is difficult to conduct extensive investigations of how the various components of the interpretation system work together to effectively interpret a particular scene. This lack of computational power is felt across the entire range of interpretation activities (from the simplest image operations to complex scene analysis).

Similar computational requirements exist in the other projects:

1. The study of scheduling algorithms for real-time systems, both hard and soft, require from two to three cpu hours for a very small network of five nodes for one given parameter setting.

2. In the study of adaptive networks, a statistically valid experiment on a small network of five elements involves a hundred simulation runs which requires forty CPU hours.

3. In the study of neural networks of the retina, a simulation involving only a sixteen degree by sixteen degree field (100,000 neurons) requires two to four CPU hours per run.

These computational requirements are just for a single experimental run. These experiments generally involve a large number of parameters whose effects are interdependent. A key part of our research is understanding these interactions and the performance relationship between task characteristics and parameter settings. Thus, *almost nothing definitive can be learned from a single or small number of runs.* These runs often point to interesting phenomenon which then must be studied based on extensive numbers of runs in which different tasks and system parameters are varied. Effective experimental research based on this methodology requires the ability for experimenters to have fast turnaround. Fast turnaround is necessary not only for providing experimental results in reasonable time, but also so that experimenters can quickly determine whether his or her intuitions, developed from a few experiments, are valid in general. Fast turnaround is also the key to effective system implementation and investigation of unexpected actions. If it takes 30 hours to run a simulation to the point where a bug or unexpected behavior occurred, interactive exploration of the internal state of the simulation is severely hampered. We also have learned that parameter settings for experiments, especially where there are large number of dependent parameters, are often set incorrectly. Thus, the ability to interact with the experiment in real-time, monitoring intermediate states to see whether or not the experiment is set up as desired is crucial to efficient experimentation. All of these factors contribute to our desire to have at least 100 mips of computing power to run our experiments. With this level of computing, most of our experiments can be run in less than an hour. For example, large node experiments in the DVMT testbed which have one of our most stringent computational requirements could be done in less than two hours (i.e., 200 hours of 750 time translates in 150 mip/hours which is 1 1/2 on a 100 mip processor).

Our current facility, which is composed of a network of 12 VAXes (2 11/780s and 10 11/750s), is inadequate from a number of perspectives:

1) It is already saturated serving the research needs of existing users: a community of 23 faculty, 15 post-doctoral researchers and 200 graduate students the majority of which are engaged in well-supported research which is highly experimental and requires extensive computational support.

2) There are insufficient computing cycles even if all the resources were dedicated to cooperative distributed computing. Even if all 12 processors on the network could be dedicated to a single simulation (an unrealistic scenario given their current saturated use for general research computing) the combined computing capacity would be less than 10 mips. As discussed above, typical experiments of the type and scale discussed here require on the order of 100 mips for effective experimentation.

3) The computing facility available can not be appropriately organized for studying the range of architectures that we are interested in. In particular, tightly-coupled parallel systems cannot be effectively simulated on our loosely-coupled network because it introduces an inordinate amount of communication overhead during the simulations. For these types of simulations, the most computing power we can bring to bear is 3/4 mip (a dedicated 11/750). Furthermore, our existing hardware is not always appropriate even for the simulation of loosely-coupled distributed processes. For instance, process migration (needed for load balancing) is impaired by the lack of needed

hardware and software support and by the costs of communicating the large amount of process state information among machines.

4) It is does not provide the appropriate hardware support for effective experimental interaction. There are only two medium resolution color graphics workstations and there are currently no high resolution B/W stations available for our use (the two Symbolics 3600s and the three SUN workstations are dedicated to specific research groups). Availability of window-based, high-resolution color workstations would greatly assist the experimenter in monitoring and understanding the many concurrent activities being modeled in his simulation as well as in analyzing the large amount of data that is generated during an experiment.

In summary, the existing facility is already insufficient for our computing needs and hence grossly inadequate for the ongoing activities and for their proposed expansion in the area of cooperative distributed computing.

We believe that a tightly-coupled, large scale multiprocessor, made of general purpose processing elements (e.g., 68000s, MicroVAX-IIs, etc.), is the most cost effective way to achieve the desired 100 plus mips computing rate. BBN's Butterfly multiprocessor (which can be expanded to 128 processing elements) has been operational for the last two years with reasonable operating and language system support. The existence of the Butterfly makes us confident that the multiprocessor approach to achieving our computing needs is realistic. A tightly-coupled multiprocessor, in contrast to a loosely-coupled processor network, permits us to simulate with reasonable efficiency a wide range of different distributed and parallel computing architectures (e.g., from a tightly-coupled and fine-grained architecture to a loosely-coupled and large-grained architecture). This efficiency is achieved because memory sharing among processors is very fast due to a high speed processor interconnect structure; this speed permits the effective simulation of a wide range of processor communication/coordination strategies. We believe that the distributed/parallel character of our simulations makes it more able to fully and in a straightforward manner exploit the parallelism available in the multiprocessor. In fact, some of the larger applications (DVMT and the distributed database testbed, CARAT), have already been structured to run on our loosely-coupled network and their transfer to the multiprocessor should be very quick.

Alternative ways of achieving a 100 mip computing power, such as a Cray-Supercomputer, are at least ten times as expensive as the BBN multiprocessor and are not as easy to expand incrementally. Such an alternative would also not provide the realistic multiprocessor environment in which our work in distributed operating systems can be evaluated. We feel other alternative architectures for achieving this processing power, such as the specialized architecture of the HILLIS connectionist machine, are neither currently available nor are they sufficiently general to effectively implement the wide range of cooperative algorithms that our researchers want to investigate.

Section 3.1 discusses in more detail the criteria that we are using to evaluate potential multiprocessor architectures that serve our needs. This section also discusses and justifies the proposed workstation support environment that will surround the multiprocessor system. We see these workstations as crucial to the effective exploration of the multiprocessor by our researchers.

In summary, we feel that a multiprocessor system with supporting workstation hardware can provide in a cost-effective manner the computing needs to effectively explore research issues in cooperative distributed computing.

6.4 Relevant Research in AI

The Artificial Intelligence group has embarked on a number of major research projects, mainly aimed at the organization of large AI problem solving systems based on cooperative distributed computing. This work, which is experimental in character, involves the building and analysis of sophisticated and computationally intensive systems. Broadly, our work can be categorized along the following major topics:

1. Distributed Problem Solving
2. Computer Vision
3. Robotics
4. Adaptive Neural Networks
5. Visuomotor Coordination

We now describe our work in each of these areas in detail.

6.4.1 Distributed Problem Solving

Personnel: Professor Lesser, 1 Research Associate, and 9 graduate students

Funding Agencies: NSF and DARPA

A cooperative distributed problem solving system is a distributed network of semi-autonomous processing *nodes* that work together to solve a *single* problem. Each node is a sophisticated *problem solving system* that can modify its behavior as circumstances change and plan its own communication and cooperation strategies with other nodes. Research under the direction of Professor Victor Lesser and Dr. Daniel Corkill emphasizes AI applications for which there is a natural spatial distribution of information and processing requirements among nodes, but insufficient local information for any node to make completely accurate processing and control decisions, without interacting with other nodes.

Our approach for implementing these AI applications on a distributed network is to have nodes cooperate among themselves so that the network as a whole can function effectively even though the nodes have inconsistent and incomplete views of the information used in their computations. We call this type of distributed problem solving *functionally accurate, cooperative* (FA/C). In the FA/C approach, the distributed network is structured so that each node can perform useful processing with incomplete input data, while simultaneously exchanging partial, tentative, high-level results of its processing with other nodes to construct cooperatively a complete solution [LESS81]. The intent is that the amount of communication required to exchange these results is much less than the amount of communicated raw data and results that would be required by a conventional distributed processing approach. In addition, the synchronization required among nodes can also be reduced, resulting in increased node parallelism and network robustness.

A key problem in cooperative distributed problem solving is to develop network coordination policies that provide sufficient global coherence for effective cooperation [CORK80; CORK84; SMIT81]. This problem is difficult because the use of a global "controller" node is not an option. Such a node is precluded by two considerations:

- Internode communication is limited, restricting the view of each node (including the proposed controller) of network problem solving activities. A global controller node would become a severe communication and computational bottleneck.

- Network reliability criteria require that the network's performance degrades gracefully if a portion of the network fails. However, if the proposed controller node fails, the resulting network collapse would be anything but graceful.

In the absence of a global controller node, each node must be able to direct its own activities in concert with other nodes, based on incomplete, inaccurate, and inconsistent information. This requires a node to make sophisticated local decisions that balance its own perceptions of appropriate problem solving activity with activities deemed important by other nodes.

We have developed and are continuing to enhance a node architecture capable of such sophisticated local decisionmaking [CORK82; LESS83b]. This architecture has been implemented as part of the Distributed Vehicle Monitoring Testbed (DVMT): a flexible and fully instrumented research tool for the empirical evaluation of distributed network designs and coordination policies. The DVMT simulates a network of problem solving nodes attempting to identify, locate, and track patterns of vehicles moving through a two-dimensional space using signals detected by acoustic sensors [LESS83]. By varying parameters in the testbed a wide range of cooperative distributed problem solving situations can be modeled. These parameters specify the accuracy and range of the acoustic sensors, the acoustic signals that are to be grouped together to form patterns of vehicles, the power and distribution of knowledge among the nodes in the network, and the node and communication topology.

Each problem solving node is an architecturally complete Hearsay-II system [ERMA80] with knowledge sources appropriate for the task of vehicle monitoring. The basic Hearsay-II architecture has been extended to include more sophisticated local control, and the capability of communicating hypotheses and goals among nodes [CORK82b; CORK83]. Goals indicate the node's intention to abstract and extend hypotheses on the data blackboard. Each node has a planner that determines the local problem solving activities of the node based on its potential processing activities (represented by goals created from local problem solving activity) and on externally directed requests from other nodes (communicated goals).

Each node is guided in its local control decisions by a high-level strategic plan for cooperation that is developed and exchanged among the nodes in the network. This strategic plan, which is a form of meta-level control, is represented as a network organizational structure that specifies in a general way the information and control relationships among the nodes. Included in the organizational structure are control decisions that are not quickly outdated and that pertain to a large number of nodes. The local control component of each node is responsible for elaborating these relationships into precise activities to be performed by the node. In this way we have split the network coordination problem into two concurrent activities [CORK83]:

1. Construction and maintenance of a network-wide organizational structure;

2. Continuous local elaboration of this structure into precise activities using the local control capabilities of each node.

The sophistication of each problem solving node in our research is significantly different from Hewitt's work on the actor formalism, Kornfeld's ETHER language, Lenat's BEINGS system, and the augmented Petri nets of Zisman [HEWI77; KORN79; LENA75; ZISM78]. In these latter systems, knowledge is compartmentalized so that each actor or "expert" is a specialist in one particular aspect of the overall problem solving task. Each expert has little or no knowledge of the problem solving task as a whole or of general techniques for communication and cooperation. As a result, the expert cannot function outside the context of the other experts in the system nor outside specific communication and cooperation protocols specified in advance by the system designer.

In our research, each node possesses sufficient overall problem solving knowledge that its particular expertise (resulting from a unique perspective of the problem solving situation) can be applied and communicated without assistance from other nodes in the network. This does not imply that a node functions as well alone as when cooperating with other nodes--internode cooperation is often the only way of developing an acceptable solution--but every node can at least formulate a solution using only its own knowledge. Each node in the distributed network also possesses significant expertise in communication and control. This knowledge frees the network from the bounds of designed protocols and places its nodes in the situation of developing their own communication and cooperation strategies.

We are presently using the testbed to explore, through implementation and empirical studies, a variety of issues in the use of meta-level control for network coordination. These include:

- **Knowledge-Based Fault Diagnosis:**
  How to detect and locate inappropriate system behavior. Inappropriate behavior includes problems caused by hardware errors, as well as inappropriate settings of the problem solving parameters that specify strategic and tactical network coordination.

- **Distributed Load Balancing:**
  What specific data, processing results, and processing goals should be transmitted among the nodes? Given a high-level strategic plan for the allocation of activities and control responsibilities among nodes, there is still a need for the nodes to negotiate more localized, tactical decisions. These

tactical decisions, made within the framework of the overall strategic plan, balance network activity based on the dynamics of the current problem solving situation. This work is being conducted jointly with Ramamritham and Stankovic and involves applying their real-time scheduling techniques to the DVMT.

- **Organizational Self-Design:** What is an appropriate initial organizational structure for a particular problem solving situation? We are working on the construction of an organizational structuring module that will select an appropriate structure based on situational parameters. This module will also modify the organizational structure to reduce the effect of hardware errors or an inappropriate organizational structure as they arise during the course of problem solving. When a hardware error is detected by fault diagnosis, the network coordination policy must be modified so that problem solving is not disrupted in other parts of the network. Alternative paths must be established to generate more accurate versions of necessary information wherever possible. When the organizational structure becomes inappropriate, due to changes in the internal or external environment of the distributed problem solving network, plausible alternative structures must be determined and evaluated as potential candidates for network reorganization. A significant aspect of this research is understanding what particular organizational structures perform better than others in particular distributed problem solving situations. Extensive experimentation using the DVMT will be required to develop organizational structuring expertise.

- **Local Planning of Internode Communication Strategies:** Since limited communication among nodes is a major characteristic of cooperative distributed problem solving networks, it is important to use this scarce resource wisely. We are currently developing and implementing communication strategies which temporarily delay the communication of information to another node if it is likely that this information will soon be elaborated or expanded by the local node. Similarly, we are exploring strategies in which a node does not communicate information if it appears that the receiving node should be able to infer the information from earlier messages. These communication strategies have to balance the reduction in communication against the increased uncertainty introduced by delayed or suppressed communication.

The goal of this research is to develop concrete information about the nature of coordination required for the effective performance of a distributed problem solving system performance where there are large numbers of nodes operating in a dyamically changing environment.

Understanding the construction of networks containing hundreds of sophisticated nodes is paramount to the eventual construction of large scale distributed problem solving networks. Such large-scale networks are particularly interesting to us, since it is only in networks of such scale that the problems of cooperation and coherence and the effects of organizational structure dominate. Much of the basic design of the DVMT node architecture was based on its anticipated use in large networks, and we are eager to evaluate this architecture's capabilities empirically in large scale experiments. These experiments would cover a wide range of different problem solving architectures, control parameter settings, and environmental scenarios. In our 4-5 node network experiments, nearly 100 individual simulations were required to explore the effect of varying a few major control parameters.

Unfortunately, the computational requirements of even a small experiment severely tax our available computing resources. Each simulated node in the DVMT is a sophisticated AI system in its own right and requires significant resources to simulate its activity. We are modelling an underlying network architecture in which each machine has the computational power and local memory comparable to a LISP Machine. Detailed simulations of even a small network of such power on a single, general-purpose VAX requires significant computational time. For example, simulating a network of 4-5 processing nodes observing 8 time frames of sensory data containing little noise requires 0.5-2 hours of CPU time on a dedicated VAX 11/750. **A realistic simulation might contain 100 nodes observing noisy sensor data over a large set of time frames and could easily require at least 200 hours of CPU time.**

The computational requirements of our simulations result from modelling the detailed activities that each node would perform in an actual distributed problem solving network. Each simulated node generates and communicates with other nodes individual hypotheses and problem solving goals with specific attributes (including belief values and ratings). Our intuition at the beginning of this research was that a significant change in a single hypothesis or goal could lead to significantly different performance results. Abstract simulation of these distributed problem solving networks would not capture such detailed, yet significant, phenomena. Experiments with the Testbed have confirmed our intuition.

We have obtained the above simulation performance figures only after considerable tailoring of the DVMT simulator and the underlying LISP environment. The Testbed is implemented on a locally developed LISP dialect (CLISP) which has a large and efficient kernel written directly in assembly language. The compiler for this dialect produces very efficient code (at the expense of runtime error checking) and this code is then combined with a minimal LISP runtime system to produce an efficient "production" version of the Testbed. These tailoring efforts continue to divert significant effort away from basic research and experimentation.

We have already taken a step to improve our experimental environment by distributing the DVMT simulator to run on a network of VAX 11/750s connected by DECNet/Ethernet [DURF84]. The Testbed was restructured as a distributed simulation system in which one or more nodes can be simulated on each machine involved in the simulation. Because this decomposition of the simulator is patterned after the loosely-coupled network under simulation, the amount of information communicated between machines is relatively small, and the time required to communicate among machines tends to be absorbed by simulated internode communication delays. However, the distributed simulator cannot provide more computational power than is available in our existing facilities. Even if all 12 of the department's research VAXes were dedicated to our simulations (an unrealistic scenario given their current saturated use for general research computing) it would still be impossible to extensively simulate networks containing 100 or more nodes operating on realistic data (i.e., at least 15 hours of network time would be required for a *single* experiment).

Access to the proposed multiprocessor facility would make an immediate and dramatic improvement in both the speed and scope of our Testbed experiments. Because the distributed Testbed simulator is currently operational on our loosely-coupled VAX network, we are already prepared to use the proposed facility to gather realistic performance data from large network simulations. The facility would remove the computational roadblock that has been the major impediment to our research efforts.

We are in the process of translating the DVMT into VAX LISP (DEC's implementation of COMMONLISP for the VAX). In addition to providing us with a better development environment, COMMONLISP will allow the Testbed to be used outside our local LISP environment. The decision to move to CommonLISP, however, will significantly increase the computational requirements of a simulation. The computational power of the proposed multiprocessor would make this decision less burdensome.

The proposed multiprocessor would also allow us to investigate the implementation of multiprocessor blackboard architectures. The knowledge and data partitioning provided by the cooperating knowledge source style of problem solving pioneered in Hearsay-II were developed, in part, with an eye towards a multiprocessor implementation. Although there is considerable potential in a multiprocessor blackboard "machine", there has been little research in that direction.

We would like to explore the implementation of the following different blackboard architectures:

1. A parallel processing architecture in which the n processing nodes are tightly-coupled through shared memory;

2. A centralized architecture in which there is a single very high speed processor that is n times the speed of the above cases;

3. A hybrid architecture in which there is a combination of loosely and tightly coupled nodes (loosely-coupled clusters where each cluster is a parallel processing architecture).

We plan to use the generalized Hearsay-II architecture developed for the DVMT and the task of vehicle monitoring as the example AI interpretation application. The Testbed is capable (with little modification) of simulating all the above architectures. Because we would be using the same knowledge in the parallel architecture simulations, we would be able to directly compare the results with our loosely-coupled, distributed network experiments.

The issues we plan to address include understanding:

- how much effective parallelism there is in high-level AI tasks such as vehicle monitoring;

- the degree that the synchronization required in a tightly-coupled architecture degrades performance;

- how robust are parallel architectures in the event of hardware failure;

- what are important task characteristics that influence the effectiveness of particular architectures.

We have been eager to use the Testbed to simulate these multiprocessor blackboard architectures, but given the computational limitations facing our current distributed problem solving Testbed experiments, we have been forced to put this research "on hold". Access to the proposed multiprocessor would allow us to experiment quickly with our ideas, with the potential for developing an actual (versus simulated) multiprocessing blackboard architecture using the proposed facility.

6.4.2 Computer Vision

Personnel: Professors Riseman and Hanson, 3 Research Associates,
15 graduate students, 6 undergraduate programmers,
1 lab manager, 1/2 technician

Funding Agencies: AFOSR, DARPA, NASA, and RADC

Industrial Sponsors: DEC and General Electric

The Vision Group in COINS, directed by Professors Hanson and Riseman, conducts an extensive and well established program of research in computer vision, described at greater length in [HANS83]. The emphasis of our work is represented by three distinct research groups working in static image interpretation, motion processing, and parallel architectures for computer vision. Research in each of these three areas is summarized independently, but it must be recognized that the groups exist within a highly synergistic research environment. Each group builds on the expertise of the other groups and there is a significant degree of cross fertilization. The groups share the same systems environment, address similar scientific issues, and share a common set of problems.

In addition to these three major efforts, we have collaborated extensively with other research groups in COINS, most closely with the Laboratory for Perceptual Robotics. They have adapted a number of vision techniques to the more constrained robotics domain. Also, the output from their tactile sensors is image-like, and suitable for properly adapted image-analysis procedures. We also share common research problems with the distributed problem solving group.

KNOWLEDGE-BASED APPROACHES TO STATIC IMAGE INTERPRETATION

The major emphasis of our work has been on the automatic interpretation of images of natural outdoor scenes by the application of knowledge about the properties and organization of objects in the world, and about the image-formation process. The analysis of such images presents a challenging problem because of the rich variety of objects that can appear in the images, the lack of strong constraints on their appearance and positioning, and the ambiguity in local interpretation of any image part unless use is made of expectations derived from context.

As a foundation for robust static interpretation systems, we have developed representations for general world-knowledge about the domain of outdoor scenes, as well as inference mechanisms and control strategies for applying this world-knowledge effectively [WEYM83, WEYM84, WESL84, REYN84, HANS83]. In support of this research in "high-level" processing for knowledge-directed interpretation, we have also developed "low-level" techniques for image segmentation, perceptual grouping, feature extraction, and multi-resolution image matching (for binocular stereo and motion analysis) [NAGI82, KOHL83, BURN84, GLAZ83, ANAN84].

In order to test the general utility of our techniques, we have applied many of them to other visual problems, such as the interpretation of aerial imagery for automatic cartography, biomedical image processing, face recognition, and in industrial automation applications3

MOTION PROCESSING

The goal of our motion research is the recovery of surface properties of the physical environment, such as distance and orientation, the motion parameters of the observer, and the parameters of any independently moving objects in the field of view. The extraction of such temporally changing information is essential for systems that must navigate through complex environments.

Our primary technique for depth inference was derived in Lawton's doctoral dissertation [LAWT84]. He showed that in cases of restricted sensor motion (pure translation, pure rotation, and motion constrained to a plane) one can bypass, or at least simplify, the correspondence problem by combining the computation of the motion parameters with the determination of image displacements.

In other algorithms that we have developed for dealing with general motion and independently moving objects, computation of an initial optic flow field is necessary. Hierarchical algorithms for feature matching for motion and stereo have been developed by Glazer, Reynolds, and Anandan [GLAZ83]. Anandan has been extending this work to detect typical errors (such as occur at occlusion boundaries) using a confidence measure.

A very interesting algorithm for dealing with *arbitrary motion* (any combination of translational and rotational motion), devised by Rieger and Lawton [RIE83], utilizes information at occlusion boundaries where significant depth differences exist. It allows the rotational components of the motion vectors to be subtracted out, leaving the translational displacement component. This again leads to motion parameters and then the depth map.

Finally, in a second doctoral thesis (now nearing completion), Mr. Gilad Adiv [AVI84] is working on the analysis of optic flow fields produced by unconstrained sensor motion in an environment containing *multiple, independently moving objects*. Thus, in road scenes his algorithms will be able to determine the location and motion of other moving vehicles and people while the sensor is also in motion.

We plan to integrate motion analysis into the framework of the object interpretation system. This will enable a robot vehicle in motion through an environment to understand what is in the environment and to navigate safely through it.

PARALLEL PROCESSING AND PARALLEL ARCHITECTURES FOR VISION

Our interest in parallel computation has led us, in collaboration with other members of the Department, to devise hardware designs for parallel computers [LEV84a,b, WEE83, WEE84]. From this has evolved the design of the *Content-Addressable Array Parallel Processor* (CAAPP), which is a 512x512 array of processing elements, interconnected in a square grid, each with a small local memory (133 bits per processor in the current design). A central controller broadcasts instructions to the processing elements, and queries keyed on the contents of the processors' memories. Several modes are provided in which the processors may respond to these queries. Thus, the architecture combines *local celular array* processing with *global associative* communication and processing.

The CAAPP has all the capabilities of SIMD (single instruction / multiple data) image array computers, being able to perform operations over an entire 512x512 image with a 100ns cycle time. However, the possibility of making fast global queries of its entire memory adds a totally new dimension to its capabilities beyond those of similar existing machines. This aspect of the CAAPP we have only begun to explore, and already it has permitted the development of a number of new algorithms for shape and motion analysis [WEEM84, LEVI84a, LAWT83]. One of the chief merits of the CAAPP is that it can be readily fabricated using today's VLSI technology.

While the CAAPP has impressive capabilities, it represents only a single, fixed architecture whose construction happens to be feasible at present. Its hardware supports only two levels of computation and data representation: the very local level of the processor array and the very global level of the broadcast query/response system. A truly powerful machine for image analysis would need to maintain numerous levels of data abstraction. By judicious propagation of data through the processor array, and by use of the global query/response capability, the CAAPP can be made to simulate these levels to some extent, and its raw processing speed permits this to be done reasonably quickly. However, a machine that properly embodied these multiple levels could ultimately realize an even greater processing capacity and consequently move us closer to real-time (or near to real-time) vision.

INADEQUACIES OF CURRENT COMPUTING ENVIRONMENT FOR VISION RESEARCH

At present the Vision Group has exclusive use of two DEC VAX 11/750s, and shared use of a VAX 11/780. Our research efforts are continually hampered by lack of computational power. The few machines available must be shared by many active researchers (approximately 30 faculty, research associates, graduate students, and programmers), not only for direct operations on image data, but also for such support operations as program preparation (editing, compiling, linking), document preparation, and system maintenance.

On a shared machine, such experiments can quite literally take weeks. With this turnaround, it is difficult to conduct experiments as extensive as would be desirable in order to increase our understanding of how such interpretation systems function. These are extreme examples, but the lack of computing power is felt across the entire range of our research activities, from the simplest image operation to the most complex scene analysis. (See table in Section 6.3.)

A constant theme of our research has been the importance of *parallel computation* for visual processing. Vision requires vast amounts of computation, both because of the quantity of input data in images, and because of the complexity of the operations that must be performed on this input. In many practical and important domains, such as robotics and navigation, these computations must be carried out under extreme time constraints. No forseeable improvements in hardware technology would enable computers of conventional architecture to meet these computational demands. Only by the use of multiple processors cooperating in parallel can the requisite computational throughput be achieved. Even more importantly, the proposed multiprocessor will facilitate experimentation with novel computational structures, experimentation which must be performed if our understanding is to be significantly advanced. We need to utilize architectures which can be "scaled-up" significantly in the future in ways that current super-computers will not.

For many vision tasks there exists specialized hardware (such as frame-rate convolvers and array processors). Such hardware has an important role to play, especially in practical applications, but our experience has shown it to have rather limited utility in a research setting. A special processor may perform its designed function very effectively, but that task will make up only a small part of the totality of operations that must be performed in image interpretation. The gain in speed on its specialized task afforded by such a processor will usually be lost in the time it takes to transfer the data to and from the processors that must perform the other stages of the entire computation. The solutions to different parts of a vision problem must be conceived of in artificially different ways, which can distract the researcher from the fundamental issues involved.

Another problem with specialized processors, especially in a research setting, is their inherent inflexibility. A particular algorithm may run quite effectively on such a processor, but a conceptually slight change in the algorithm may render it totally unsuitable for the processor, so the algorithm must be entirely recoded for a more general-purpose machine. This "fragility" can deter researchers from even using a specialized processor in the first place, or if it is used can discourage full explorations of variations.

What is needed is a machine flexible enough to examine different computational, structural organizations with enough memory to perform all stages of vision processing in a unified manner, and with enough computational power to support timely exploration of the range of processing alternatives.

IMPACT OF PROPOSED MULTIPROCESSOR ON VISION RESEARCH

A tightly-coupled multiprocessor, as requested in this proposal, would overcome almost all of the computational deficiencies described above. By reducing turn-around time, it would allow proper and extensive experimentation with vision processes. It would also stimulate the development of new computational structures for machine vision, especially since its organization is not biased towards any particular structure. At this point in our understanding, there is no consensus on the form of the computational structure that a vision interpretation system should take. Thus, what is needed now is the flexibility that the proposed multiprocessor would provide.

The benefits of a tightly-coupled multiprocessor would be felt across the entire range of vision research conducted here:

- low-level processing of pixels and local neighborhoods in a SIMD format;

- the intermediate levels of processing where local areas of the image are processed in a MIMD form;

- the high-level symbolic processing where the computation is distributed in an entirely different manner;

- the multiprocessor system could be organized to run an integrated set of processes for examining real-time issues for robot navigation.

Let us examine each of these in more detail.

Multiprocessor For Low-Level Vision.

The fundamental operations of image segmentation and feature extraction [NAGI82, KOHL83, BURN84] are based on local computations performed at every pixel, or a neighborhood about every pixel, of an image. These operations can readily be decomposed for implementation on a multiprocessor machine by spatially partitioning the image and assigning a processor to each partition. Interprocessor communication is needed only for those few pixels along the boundaries of the pieces, so the parallelism of the multiprocessor can be almost fully exploited to speed up such processing. The computational cost of static vision, where a 512x512 pixel gray scale image involves 1/4 megabyte of data, is increased by 1 to 2 orders of magnitude in motion processing which might involve processing rates approaching 30 frames per second [LAWT84, ADI84].

Our entire software environment has been designed to encourage this style of processing within the VISIONS Image Operating System. Programs for low-level image processing and feature extraction are written as if the operations were to be locally applied in parallel to every pixel of the image; the software system then sequentially simulates these parallel operations. We have investigated ways in which these processes can be decomposed in order to exploit parallel computation, using multiresolution hierarchies [HANS80, GLAZ83, REYN84], but much research remains to be done in this area. This speed up will have a double impact on the development and refinement both of new low-level techniques and of high-level techniques, as described below, which use these low-level operations as computational building blocks.

Multiprocessor For Intermediate Vision

The intermediate level of representation provides an interface between the low- and high-levels of representation, that is, between pixel-based representation and symbolic elements representing visual knowledge stored in a database. In the UMASS VISIONS system [HANS78a,b, HANS83], the intermediate level consists of a symbolic description of the two dimensional image in terms of regions and line segments (that are still in registration with the raw image data) as well as their associated attributes which can be used in the interpretation process. In some systems this level would consist of representations of surfaces, or more generally, "intrinsic" features of the physical environment [MAR82, BAR78].

We believe that the key to vision processing is a flow of communication and control both up and down through all representation levels. Communication between these levels is by no means unidirectional. In most cases, recognition of an object or part of a scene at the high-level will establish a strategy for further processing which usually results in further probing of the low and intermediate levels, in order to pull out additional features under the guidance of a partial interpretation. This might involve joining together region, line, and surface information to form a symbolic representation would more easily and naturally match a stored object description. Thus, the intermediate organization on a multiprocessor would be MIMD, again partitioned spatially, with each processor performing different activities in different areas of the image.

Multiprocessor For High-Level Vision

There are two major approaches that we are investigating that would distribute symbolic processing across the multiprocessor. The first involves parallel, cooperating object schema structures being developed by Weymouth [WEYM83] for image interpretation tasks. Each object schema is a data structure defining an expected collection of objects and their relations, as well as control strategies for detecting and verifying the presence of that object class in an image. Weymouth's doctoral thesis [WEYM84] is beginning to explore sequential and parallel interpretation strategies and intercommunication processes as we begin to examine how to enable these processes to cooperate in the interpretation task.

The second involves inference networks using a Shafer-Dempster formalism for evidential reasoning [LOW82, WESL84]. Here, the knowledge network is represented as a set of propositions (including relational propositions) which is distributed across the set of processors at a finer level of "granularity". The computations of belief in these propositions are designed so that they can be performed in parallel all over the network, even though they now must be executed sequentially. The goal of this work will be to allow many pieces of partial and uncertain evidence to propagate through the network and automatically activate procedures to hypothesize or verify objects in particular areas of the image. We need to begin to understand how a network of thousands of simple nodes can run cooperatively. Wesley's doctoral thesis on control is a design for such an inference network.

The computation for both these symbolic processing networks will be distributed across the multiprocessor along semantic divisions rather than spatial divisions. Here, we distribute object schemas or nodes in a semantic network across the set of processors.

System Organization

The most challenging problems that we face in using such a multiprocessor is that sometimes the individual processors will be used in the spatial distribution of low and intermediate levels of processing; and at other times or at the same time be required to do high-level processing. Of particular interest are structures that facilitate two-way interchange between the low-level, iconic representation of image data and the high-level, symbolic description of a scene. While the structure of many of these computations is regular enough that they may be most effectively implemented in special hardware with dedicated communication paths, extensive simulations of alternative structures must be conducted to study the actual data traffic in such computations. Only a distributed machine would have the structural flexibility and processing power to run such simulations in a natural and timely fashion.

At the most general level of experimentation various modular building blocks for constructing an integrated vision system can be examined. We are interested in the development of a mobile robot system that can navigate through a laboratory environment. This would involve a static vision system for object interpretation integrated with a dynamic vision system for computing the depth of visible surfaces and objects, and a planning system for navigation. We have also been talking with the large natural language processing group in our department about a natural language user interface. The distribution of processors for this type of real-time system would naturally be somewhat different and organized along more pragmatic system constraints.

6.4.3 Robotics

Personnel: Professors Arbib and Riseman
2 Research Associated and 10 Graduate Students

Funding Agencies: NSF and ONR.

Corporate Sponsors: DEC, General Electric, and Martin Marietta

The Laboratory for Perceptual Robotics is currently involved in studying the usage of a variety of sensory information in controlling dexterous robot manipulators. We are attempting to remedy a major deficit of those approaches to high-level planning and control which lack on-line sensory input. Since robots must function in a dynamic, usually unpredictable, world, our approach will use vision and tactile sensation to allow the machines to adapt their behavior to the environment. Our research will yield a *distributed, schema-based, architecture* for specifying tasks which use dynamic sensing within such an architecture.

A robot system consists of an articulated mechanical device capable of interacting with the environment, sensors of various sorts, and a network of logical units given the responsibility of controlling the system. Sensory information is typically integrated at two very different levels depending upon the type of information involved. At the servo-control level, velocity and position information for a given actuator are used by the control algorithms in order to achieve satisfactory performance. Force feedback information is used at a level involving the interactions of several joint-link pairs. The other extreme is exemplified by what might be called the "snapshot" paradigm for the use of visual information in which the actions of the robot are halted, processing of visual input is requested of another unit, this other unit collects and processes the data and returns a coded form of the information. The robot system then proceeds "blindly" based upon this new glimpse of the world. One problem with this approach is the need for the environment to be relatively static and well specified. Moreover, in this approach the functioning of the robot tends to be viewed as a serial progression of events with only a single task or subtask implemented at any given instant. This implies discrete switching to new subtasks and an inability to change attention smoothly from one set of activities to another.

A major portion of the research being done in the Laboratory of Perceptual Robotics is geared toward the study of the use of sensory information in a "Real-Time" manner [ARBI83a]. Use of sensory information in such a manner will allow robots to be continually aware of their environment, which will then allow them to function in a "non-static" environment.

To achieve this aim, we are keying on two sensory areas, Vision and Touch. Overton (now at General Electric) and Begej have developed various tactile sensors and we have begun to analyze possibilities for their use in an active touch paradigm [OVER84; BEGE84; ELLI84] In the area of Vision, we are exploring ways in which a number of useful routines that have been developed by the members of the department's Vision Laboratory can be modified to function in the real-time domain. It is our belief that the combination of information from both types of sensors will help us in achieving our goal of real time control.

In an *explicit* centralized planning/control process, a master process analyzes the assemblage of subprocesses to extract a *motor plan* as a static high-level program which will then be used to generate control signals for all the actuators. By contrast, much of our research will seek patterns of competition and cooperation between subprocesses, which will yield an appropriate pattern of action – without an overall plan necessarily being encoded at any one node within the system. We shall explore the extent to which coordinated control programs (plans) may be decentralized. For example, it is possible to distribute the knowledge without distributing the control, e.g., a master schema could place a set of tasks on a blackboard, all the appropriate subschemas could compete for tasks and decide how to accomplish them, the master could then decide how to synchronize activities and set things going. We can still view this as a distributed system – the "master" is just an expert in one aspect of the task (namely, breaking it up into subtasks and doing the final synchronization). The examination of such alternatives is at the heart of the work in organizational structure conducted by Lesser and Corkill [LESS83] as part of our work in distributed problem solving.

In our description of "distributed control", the subprocesses are known as Schemas, of which there are two main types, Motor and Perceptual [ARBI81; ARBI83a; IBER84]. Each schema monitors the state of its local environment, which includes sensory information as well as information concerning the state of the neighboring schemas. Although schemas may be explicitly "turned on" by programs encoded in other schemas, each will also have an activation level, which when surpassed will cause the schema to active. This activation level can be viewed as a measurement of the effect the schema has on other local schemas and the controlled system. The activation level of a schema can be changed either directly by perceptual input or indirectly by other schemas. Simultaneous activation of schemas which can produce patterns of cooperation and competition can, in many cases, replace explicit control by a centralized plan.

Some of the issues we are addressing with regard to schemas are: the means by which they will communicate with each other; the allocation of schemas to different processors, including both static and dynamic allocation; a distributed operating system for this distributed control structure; and the associated hardware issues. Another long range issue that will need to be addressed is the problem of node failure. We will need to determine what is meant by failure in this type of control structure, how such a failure can be detected, and how the system may be designed to recover gracefully.

Current plans within the Laboratory for Perceptual Robotics call for the design and installation of a two-arm robot system. This system will consist of a series of small distributed networks whose functions will be coordinated by a larger "Control/Planning" network. It is with regards to this network that our plans are not yet formalized.

Several members of the Laboratory for Perceptual Robotics are beginning to implement a Schema enviroment on one of the department's VAX 11/780's [ARBI84]. It is from this work that we are now realizing the deficencies in using this type of hardware. The major reason for these seen deficiencies is due to the inherent concurrent nature of schemas. In addition, a large number of schemas need to be activated to accomplish even very simple tasks.

For example, in picking up a coffee mug, we initially activate some perceptual schemas to first find the mug and then to determine its orientation. Then several motor schemas will be activated to preshape the hand. This preshaping will be based upon information provided by the perceptual schemas and will determine how the mug is to be grasped. While this is occuring, other motor schemas will be activated to move the hand, by the arm, over to the mug. On the approach to the mug, perceptual schemas will constantly have to monitor the scene so as to prevent the hand from smashing into the mug. Once at the mug, which will be determined by the perceptual schemas using both visual and tactile data, one or more motor schemas will be activated to actually grasp and lift the mug.

So as we can see from this small example, a number of schemas need to be activated in order to accomplish a simple task. As we work with more complex scenarios, such as two-arm coordination, the numbers of schemas which need to be concurrently activated will rapidly grow. In addition, the schemas themselves will grow more complex and will require more CPU time. Thus it is easy to see that the present computing environment is not sufficient to develop this method of control.

The current proposed multiprocessor system will, on the other hand, allow us to design and implement a large complex schema enviroment. Its large number of processors should easily be able to accomodate such an environment and allow us to use it to control a robot system in a real-time fashion.

In constructing a schema environment, the multiprocessor system will be first used as a simulation testbed. This testbed will allow us to design and implement the individual schemas, and to observe their interactions in controlling various simulated robot manipulators.

It is during this design and implementation stage that access to several graphic work stations will be needed. It is with these workstations that we will be able to simulate the different robotic environments that the schemas will control. In addition, the workstations will be attached to various nodes within the network so as to allow us to observe the functioning and interaction of any individual schema.

The goal of this phase of work will be to produce a schema control system which will be able to control a graphics representation of a real, complex robotic environment. It is desirable that the output from this network not only be able to control such a simulation but also a real robotic system. Thus the need for a workstation which will be able to drive a simulation based on the output of the network as well as provide sensory information to the schemas running in the network.

Thus the second stage is to use the developed network to control several manipulators in a coordinated fashion. The network will then be connected to the smaller control and sensory networks already in place in the Laboratory for Perceptual Robotics. New and possible specially designed pieces of equipment will be neded to fully interconnect the networks.

Throughout the design and implementation stages, we will be drawing on the expertice of other groups within the department, especially the VISIONS and Distributed Systems groups. Their expertice will be of great help in solving many of the problems that we face in building a schema network.

It is believed that such a "Control/Planning" network will provide an excellant testbed with which we can better understand the need and use of sensory information in controlling robot manipulators. In addition, it will be an important tool in futhering the development of the Schema theory in both robotic and brain research.

6.4.4 Adaptive Neural Networks

Personnel: Professor Barto
1 Research Associate and 3 Graduate Students

Funding Agencies: AFOSR

Barto's research group is studying fine-grained networks consisting of units that are relatively simple in their input/output behavior but that implement relatively sophisticated stochastic learning algorithms. This research, which is partially motivated by issues in neuroscience and the evolution of learning in animals, has implications for the implementation of the "connectionistic" AI systems that are currently being proposed [ANDE82; BART81a; BART81b; BART83; SUTT81; SUTT84].

This project addresses the study of adaptive problem solving and learning, both in animals and machines, via networks of relatively simple, using neuron-even on a dedicated VAX 750. like units. Interest in neural-network approaches to computation is undergoing a revival as numerous groups of researchers argue that networks of interacting neuronlike processing units provide useful alternatives to conventional computational architectures and programming techniques. (for example, Ballard, 1981; Feldman, 1981; Hinton & Anderson, 1981; Hinton and Sejnowski, 1983). This use of massive parallelism is being called by some a *connectionist* paradigm. Not only have advances in microelectronics made the physical realization of massively parallel, and more brain-like, hardware a possibility, but advances in our understanding of certain types of problems suggest that the use of such hardware is necessary for real-time performance (for example, in tasks involving the satisfaction of large numbers of relatively weak constraints).

Barto's research project involves a specific approach to learning in the connectionist paradigm. Learning methods can augment heuristic, knowledge-based techniques in handling uncertainty by extracting regularities from experience in order to reduce uncertainty when similar experience recurs. Unlike earlier adaptive network studies, this approach begins with adaptive units that are much more complex than units studied in the past. We have developed adaptive units that are capable of learning to solve, by themselves, nontrivial problems involving the control of unstable dynamical systems. These learning problems are difficult due to the low specificity and high uncertainty of the feedback available to evaluate system performance. Learning under these conditions requires sophisticated algorithms for generating trials and assigning credit. Thus, this type of distributed system is more finely-grained than the others described in this proposal since the nodes implement simple input/output functions and communicate by means of excitatory and inhibitory signals rather than by symbolic messages. However, the algorithms used by the elements to adjust the parameters of their input/output functions are more complex than those generally considered in these types of networks. The early adaptive network studies greatly underestimated the complexity of the tasks that individual adaptive elements face as components of networks.

In a very literal sense, the nodes in these networks are "self-interested" agents that possess means for furthering their interests (defined by a preference structure on their inputs) when interacting with a wide variety of environments. When such nodes interact as members of a network they form interconnections among themselves, and with externally supplied signals, that permit each node to perform better with respect to its own preferences than it would by acting independently of other nodes. The nodes cooperate to form stable "coalitions" to the extent that such collective behavior furthers the interests of the constituents. Simulations show that networks can learn to solve problems that are beyond the capabilities of any individual node [ANDE82; Barto and Anderson, in progress]. The study of networks composed of this type of unit involves many of the issues that arise in the study of the collective behavior of self-interested agents in the context of game-theory, economics, and evolutionary biology. We think that these issues are also relavant to distributed computing.

There are several reasons why experimentation with these types of systems is computationally intensive. First, in common with all the projects discussed in this proposal, comparative studies must be performed in order to establish progress, and parameter searches must to be conducted in each case to ensure "fair" comparison. Second, the algorithms embodied by the neuronlike units in this study are stochastic. Randomness is used to help avoid limitations associated with "false peaks," and is an essential feature of the learning algorithms. Reliable indications of performance can only be obtained by averaging over many simulation runs. Additionally, for development purposes, we sometimes find it useful to compare algorithms according to a performance measure that requires much more computation than does the simulation of the network itself. For example, a network of just five units was extensively studied. Performance was averaged over 100 simulation runs, each of 80,000 time steps. At each time step of each run, an instantaneous performance measure was computed that was a complicated function of the network's connection weights. This took approximately 40 cpu hours on our VAX 11/780 with a floating point accelerator. With the proposed multiprocessor facility, more accurate estimates of the performance of larger stochastic networks can be obtained.

Experimentation with larger networks is essential since this approach to distributed computation can only be eventually vindicated if it can be shown that it scales up in a reasonable way. This is a classical question asked of these biologically-inspired network approaches to computation, and has yet to be answered (although one might regard the architecture of animal brains as encouraging evidence). To become a practical means of problem solving, these methods will undoubtedly require special-purpose hardware that is distributed as finely as the networks themselves. In order to justify the fabrication of such hardware, and to determine how it should be designed, convincing demonstrations are required of what large nets can do. Although the proposed multiprocessing facility will not be capable of achieving the computational speeds of such special-purpose hardware, it can make possible more convincing demonstrations of the utility of such hardware.

The proposed multiprocessor facility can be used in several ways for this project. Since we simulate stochastic processes and average over many realizations, it is straightforward to obtain considerable speedup by letting each processor produce one of the realizations and then average over the realizations obtained by the processors. Doing this using 100 processors for the example mentioned above would reduce the 40 cpu hours to 15 minutes. Absolutely minimal effort is required to distribute the computation. Another method of distributing the simulations is to let each processor simulate a separate group of network units. For the first method, considerable memory for each processor is required, but a tightly-coupled architecture is not. For the second method, each processor requires much less memory, but the tightly-coupled architecture is necessary. Thus, the flexibility of the proposed facility is desirable for this project.

Although the preceding discussion focussed on the long-term goal of fabrication of finely-grained, massively-parallel hardware, the learning algorithms being developed can be useful in the short-term when embedded in more conventional distributed systems such as those under study by other research groups in the COINS and ECE dapartments. Algorithms of this kind are designed to construct effective control rules when there is little a priori knowledge about the process to be controlled and when there is a lack of high quality instructive information. These conditions arise not only within finely-distributed systems but also within systems whose nodes implement, for example, functionally accurate, high-level interpretation processes as in Lesser's project. One example is provided by a preliminary study of stochastic learning algorithms to facilitate load-balancing in a network of conventional computers [MIRC84]. When a node transfers a process to another (target) node, feedback as to the state of the target node is used to update the probabilistic decision process used to select that target. The decision process uses information about the state of the network that is unreliable owing to its being incomplete and out-of-date (because of propagation delay). Encouraging initial results were obtained, but the extensive experimentation required to establish its utility has not been feasible with our present facility.

6.4.5 Visuomotor Coordination

Personnel: Professor Michael Arbib
3 Graduate Students

Funding Agencies: N.I.H.

Professor Michael Arbib's group has built many computer models of neural networks underlying vision and the control of movement, including models of stereopsis, the development of retino-tectal connections, detailed models of retina, control of hand movement and the role of the cerebellum in the modulation of movement. Models constructed over the last few years address the following: local circuitry in the frog tectum that can mediate facilitation on repeated presentation of a prey-like stimulus [LARA82]; circuitry underlying retinal "feature detectors" [LEE83; LEE84]; distributed processing in the tectum in interaction with the pretectum that can subserve pattern recognition, including the discrimination of prey from predator [CERV83] the interaction of accommodation and disparity cues in the depth perception system of frog and toad (a one-eyed frog can snap fairly accurately at its prey; yet experiments with prisms show that the toad will prefer to use binocular cues in areas visible with both eyes); detour behavior, the highly complex integration of depth cues for worms and barriers that determine whether the animal will snap through a fence at its prey, or detour around it in a complex sequence of movements including side-step, orienting, and snapping [HOUS82; House, 1984] and studies of habituation, in which the relative efficacy of different stimuli changes over time on the basis of the order of presentation. Other work tries to relate these specific neural models to our general theory of schemas (see below) [IBER84; ARBI83b] and to relate the models specific to the amphibian to other models appropriate to mammals in general, and to primates and humans in particular.

Current neural modelling research in COINS department involves visuomotor coordination aspects of frog's brain as its model system. Our approach to analysing mechanisms of visuomotor coordination is two-fold: "top-down", to offer a coordinated control program of interacting schemas to achieve behavior noted by neuroethologists; and "bottom-up", to provide detailed models of neural networks which are consistent with known anatomy and physiology, but which involve additional assumptions, amenable to experimental test, to yield a network capable of exhibiting appropriate behavior. Both approaches of our research address the topics of inherent parallelism of brain as opposed to the serial nature of a conventional digital computer, where in the former schemas serve as hierarchically organized concurrent processes while in the latter individual neurons or a subassembly such as a layer or a column serves serves similar purpose. Even the "simplest" vertebrate brain is a structure of awesome complexity, and modelling of these neural network simulations to date, because of computational limits both in terms of speed and power, are made up of an unrealistically small number of elements and/or of unrealistically simple elements. We feel a multiprocessor system benefits our research area in that

- it increase the total computing power available for simulations of larger neural networks, and

- it provides a genuine distributed computing environment which resembles the underlying structure of neural network in some degrees.

The current layered model of the retina involves approximately 100,000 neurons for the visual field of 16 degrees by 16 degrees with very dense interconnections between the neurons in adjacent layers, which mathematically is represented as convolution operations, resulting in a typical simulation time on a VAX11/780 of two to four hours CPU-time [LEE84]. Increasing the size of the visual field for the current model appears to be not practical. Also, the current two-dimensional model of tectum and pretectum involves approximately 500 neurons for visual field of 32 degrees by 32 degrees [CERV83]. We want to simulate a visual field of at least 100 degrees by 100 degrees, with approximately 6000 neurons involved, to explain more behavioral data besides simple pattern recognition capabilities. Moreover, coupling a various models into a super-model, such as the coupling of separate retina model as a preprocessor for the tectum and pretectum models, still demands more computing power [ARBI82b].

The current model of coordinated control for hand movement utilizes several levels of concurrent perceptual and motor schemas to postulate hand movements involved in reaching to grasp an object, reaching to lift a mug, and putting down an object [ARBI83a]. Currently, it takes about an hour of CPU-time on VAX11/780 and the proposed extension of the model to include individual dynamics for every moving parts involved as well as the real-time display of the simulation will substantially increase the computational requirements for the group [IBER84].

But, using a supercomputer on remote site such as CYBER-205, while it reasonably reduces the computation time for a bigger model, involves an extreme increase of turn-around time for simulation data to be interpreted through the graphics device at the department.

The immense quantity of data that our models produce even now necessitates the interpretation of simulation results through a high resolution graphics device. Further, the layered or columnar structure of our models facilitate the interpretation process substantially if proper mapping is done for the different layers/columns with various color combinations.

First, multiprocessing environment can lend itself for the simulation to be performed by logical partitions based on the underlying structures for the neural networks being modelled. For example, different cell types can be computed by a subset of processors with interconnections between these cell types to be represented by temporal intercommunications between different partitions.

Second, as a future extension, we might be able to perform behavior simulation of the whole animal by mapping each concurrent processes to individual partitions.

Third, there is a chance the we can model more properties of the modelled system such as in modelling chromatic as opposed to achromatic visual pathways.

Past neural modelling efforts has provided a seedbed for our current work in Robotics as an outgrowth of a concern with sensorimotor coordination, and in processing dynamic images as an outgrowth of studies of human and animal motion perception.

Currently, our algorithm on depth perception is being investigated to be used in a robotics environment, and some of VISION group's digital image processing techniques are being sought to improve the computational aspects of the simulation for the neural models.

6.5 Relevant Research in Distributed Computing Systems

The *Distributed Computing Systems* group has embarked on a number of major research projects, all aimed at understanding control and structuring issues in large distributed processing networks. This work, which is experimental in character, involves empirical studies using sophisticated and computationally intensive software testbeds. Broadly, our work can be categorized along the following major topics:

1. Scheduling and Distributed Control
2. Concurrency Control and Synchronization
3. Distributed Resource Sharing
4. Protection in Distributed Systems
5. Programming Facilities for Parallel and Distributed Computing
6. Distributed Debugging
7. Modelling and Performance Evaluation of Distributed Systems

We now describe our work in each of these areas in detail.

6.5.1 Scheduling and Distributed control

Personnel: Professors Ramamritham, Stankovic, and Towsley
7 Graduate Students

Funding Agencies: NSF, the US Army CECOM/CENCOMS, and the
Naval Underwater Systems Center.

Multicomputer systems, including loosely-coupled networks and multiprocessors, that are going to realize their full potential for resource sharing will need to employ distributed, cooperative algorithms. These systems are characterized by decentralized system-wide control of resources for the cooperative execution of application programs [CASE77, RAMA84d, STAN80, 83a, 84a, OUST80]. Moreover, these systems must operate with missing, incomplete, delayed, or erroneous state information. Specifically, our primary research effort is to develop, compare, and evaluate process scheduling and load balancing algorithms [BOKH79, CHOW79, STAN83a, STON77, STON78a, TANT84a, 84b] for such a demanding environment. Algorithms that use heuristics based on statistical decision theory [STAN83b] and stochastic learning automata [MIRC84] and schemes based on bidding [RAMA84d, STAN84a] are being employed. Important research questions, all related to the cooperation between multiple scheduling components, that need to be answered include: How efficient is the cooperation among multiple scheduling components? What type and amount of state information is needed by the schedulers to perform their task? Can the schedulers predict future behavior, using out of date and uncertain state information, with enough accuracy to make prediction worthwhile? Will the distributed algorithms operate in a stable and robust manner [STAN85]?

Currently, the research covers loosely-coupled distributed systems where response time, throughput and delay are the important metrics. This research is significant because it is one of the few attempts at systematically determining the importance of various state information, it accounts for realistic requirements such as the need for (distributed) objects including files, as well as attempting to derive functional dependencies among various parameters important to operating systems. In this work, heuristics for controlling multiple cooperating scheduling entities are developed and the resultant performance of these heuristics is determined to be good [STAN83b, 83c,84a,85, MIRC84].

Our research also covers loosely-coupled distributed systems where tasks to be scheduled have real-time constraints such as deadlines [RAMA84d, ZHAO84]. In this research, the percentage of tasks making their deadline and other metrics relating to the real-time constraints are important. In this area, we have developed a flexible scheduling approach which requires little priori information [RAMA84d]. This is a significant contribution to the state of the art in this field in which almost all previous work requires complete a priori information. The approach includes the development of special guarantee routines [ZHAO84] that can determine dynamically if a task with a deadline can be guaranteed to meet a real-time constraint, and the use of a bidding algorithm to get distributed hosts to cooperate with each other in a system-wide attempt to guarantee the real-time constraint of a task. Extensions of the work are now factoring in such issues as precedence constraints and failures. Also we are exploring channel access protocols for guaranteeing that messages will be delivered within bounded time [RAMA84e].

In total, there now exists five major simulation models implemented and running. The computer time required for solving the analytical models and running the simulations for a 5 node system is large (2-3 hours of cpu time on a VAX 750 per run). Longer runs are expected as soon as new extensions to our algorithm are implemented.

With the proposed multiprocessor in place the simulation itself can be distributed in a number of ways: each node in the multiprocessor system can be allocated a subset of the nodes in the simulated system. Simulation of these subsets can proceed in parallel with sharing of state information accomplished via the shared global memory of the multiprocessor. Alternatively, multiple runs of the simulation (with different random number generator seeds) can be executing in parallel. These runs can then provide good confidence intervals. In addition, it will be possible to test a greater range of parameter values and even attempt to develop an understanding of the functional dependencies among the parameters. To date, such an understanding has not been well developed in distributed scheduling research, due, in part, to lack of sufficient processing power being applied to the problem.

Work on distributed task allocation and load balancing is currently being applied to distributed problem solving systems. Specifically, we are exploring the scientific issue of balancing the load among the nodes cooperating to solve a single problem. In addition, we are studying techniques for distributing the tasks that arise in the Vehicle Monitoring Testbed among the nodes on which the testbed is distributed. Thus, the testbed will also permit it to experiment with possible solutions to the problems of distributed task allocation and load balancing.

In future work, we would like to implement some of the algorithms found to give good performance on an actual or testbed system such as the multiprocessor. These algorithms, although they were meant for a loosely-coupled distributed system, can be evaluated on a tightly-coupled distributed system. Validation of the analytical and/or simulation models with a testbed system could then be attempted providing more evidence of the worth of the algorithms and models, and providing insight into the implementation issues that must be accounted for in the models.

We also envision an expansion of our research to include the development of scheduling algorithms for multiprocessors, and other improvements to multiprocessor operating systems. This requires access to a multiprocessor with a simple kernel. We would then implement our own scheduling algorithms and other higher level functions of the operating system. We believe that this research version of the operating system (would run at dedicated times only) would contribute to the state of the art in multiprocessor operation systems. The AI systems discussed in section 6.2.2 will serve as applications for this new operating system and its distributed control algorithms.

In summary, the new equipment being proposed would significantly enhance our ability to generate more and better results associated with our current work on distributed control, and provides a vehicle for many new projects on multiprocessor operating systems in the near future. The potential advantages of this research are to reduce the amount of centralized control required in a distributed system, thus eliminating critical entities (resources), and provide better overall performance. Furthermore, due to the elimination of critical resources and the development of algorithms that can operate in the presence of missing or erroneous data we hypothesize that improved reliability is also possible.

6.5.2 Concurrency Control and Synchronization

Personnel: Professors Kohler, Moss, Stankovic, and Towsley
4 Graduate Students

Funding Agencies: NSF and the Naval Underwater Systems Center.

Distributed database systems are extensions of conventional single-site database systems which have become widely accepted as important information management tools. There are many advantages to a distributed database system, but lack of centralized information, site or communication failures, and the synchronization of multiple autonomous sites, make it more complicated to design efficient control algorithms. Specifically, the concurrency control algorithms, and commit and recovery protocols for distributed databases are all more complicated. We are developing algorithms in this environment and studying their performance in a testbed. In the field, little or no quantitative results have been published about the performance of these types of algorithms and protocols for distributed databases. Such experimental data is essential to the development of practical concurrency control algorithms, commit and recovery protocols.

The research at UMASS addresses the design, evaluation and modelling of distributed database systems. We have invested more than 5 man years of effort in implementing, CARAT, a distributed database testbed, on a loosely-coupled network [KOHL83a, 83c]. Using this testbed we are undertaking empirical, simulation and analytical studies. These studies involve the implementation and measurement of algorithm and protocol alternatives on the testbed and in simulation models, the evaluation and comparison of the performance of each of these alternatives, and the development and validation of analytic models which describe the observed behavior [KOHL83b]. For example, we have developed a protocol for dealing with network partitioning that makes use of semantic information about the data and transactions of the database system [STAN84b]. This protocol is decentralized and must integrate with other decentralized protocols. Getting multiple, decentralized protocols to cooperate effectively is major research problem. We are also developing a distributed deadlock detection algorithm for a real-time database system. Developing or maintaining large wait-for-graphs or sending many messages are not solutions the deadlock problem for real-time systems. The resultant deadlock resolution scheme interacts with the concurrency control protocols and the overall system performance needs to be better understood and evaluated. Experiments on the tesbed can advance the state of the art in this area.

The main problem we have with performing all of this experimentation on the current testbed is that the current testbed is small (2-4 nodes) and the subnet on which these nodes reside is part of a larger network which cannot be dedicated to our use. Large amounts or a burst of unexpected traffic on the subnet can invalidate a given test. This frustrates our experimental efforts causing a significant amount of lost time. Using a dedicated portion of the multiprocessor would enable us to test more realistic network sizes (64 processors) and avoid resource contention with other users. Of course, CARAT would have to be converted to run on the multiprocessor. This should not be a major problem given the experience we already have in this area. The multiprocessor would also facilitate the simulation and analytical studies portions of this research in the same manner as described above for the distributed scheduling algorithms research.

The major strengths of this research are that the testbed is already implemented and operational (a very difficult task as witnessed by the limited number of such testbeds in existence), and that analytical models have also been developed providing a combination of empirical and analytic studies so sorely needed in this area of research. We now have the ability to validate these analytical models, but we do not yet have the proper equipment to perform the task correctly.

In the future we would like to better integrate the distributed operating system work with the database work, e.g., in the transaction concept and deadlock resolution areas. Integrating operating systems and databases based on a multiprocessor environment rather than a loosely-coupled network would contribute to state of the art in supercomputers; not the hardware but the controlling software.

We are also studying the value and limitations of traditional transaction oriented consistency management, and the extensions required for cooperative distributed environments. Extensions to transaction semantics will be necessary because traditional techniques are designed strictly for competing entities rather than cooperating ones. Among the extensions currently conceived are nested transactions [MOSS82], and the provision of transaction based concurrency and recovery control at multiple levels of abstraction simultaneously. Future work must examine how to provide recovery, and discover types and levels of consistency required in distributed problem solving applications.

With a distributed database testbed, we will investigate transaction methods, which gain consistency via concurrency restrictions, negative feedback and other error correction schemes, and cooperation, as means for maintaining consistency. More importantly, we hope to gain a deeper understanding of the role of consistency and the degree in which it is required and exhibited by cooperative systems. It could be that problem solutions can be developed at several levels of consistency, but with different performance and failure robustness properties. We will try to reveal the relationships among consistency, performance, and robustness.

In summary, the new equipment being proposed would enable us to complete our current database research on a realistic size network, develop proper validation results, and also serve as a means for addressing some of the software control issues that will need to be solved before the full processing power of the new supercomputers can be exploited.

6.5.3 Distributed Resource Sharing

Personnel: Professor Kurose

We have been exploring an approach to decentralized resource allocation mechanisms in computer networks based on models and methods from mathematical economics. In this approach, network agents act as selfish, utility-maximizing entities and their cooperation via a resource pricing mechanism serves as a decentralized computational device for computing an optimal distribution of network resources. The algorithm itself is an iterative process of local (and hence parallel) optimizations followed by a partial information exchange via the pricing mechanism. This approach has already been successfully applied to the optimization of several channel sharing policies in single-hop multiple access networks (Kurose et. al. 1984a, Kurose et. al. 1984b). Interestingly, several networks mechanisms such as flow control and priorities were found to naturally emerge from this approach.

Currently, these ideas are being extended to other network environments, such as multi-hop packet radio networks. The "microeconomic" approach should be particularly well-suited to this type of environment, since one of its properties is that as the (coupled) resources themselves attain a degree of spatial locality (as in multihop networks), so too does the interaction involving their distribution. Several alternative "pricing" mechanisms are also under investigation.

Although this work addresses fundamental theoretical issues in the design of cooperative decentralized resource sharing mechanisms, there is a very strong experimental aspect to this work as well. We have found that the sufficient (but not necessary) mathematical constraints to guarantee the convergence of the iterative process to be overly restrictive and seldom satisified by resource allocation problems occurring in "real-world" networks. Thus, the converge properties and performance of the resource allocation algorithms can only be studied empirically. Until now, constraints on the available computing resources have limited our studies to simulations of a small (four node) network (Kurose 1984c). While these studies indicate the feasibility of the basic approach and have provided much insight into the issues involved, these approaches must still be examined in a much larger network environment in order to determine their suitability for implementation in real existing networks. The proposed multiprocessor facility would thus provide an ideal setting for pursuing this experimental aspect of our work: the additional computational power would provide the means for not only simulating these algorithm in large scale networks, but also provide the opportunity to implement the algorithms themselves in their distributed form and examine their behavior in this setting as well.

6.5.4 Protection in Distributed Systems

Personnel: Professors Stemple and Ramamritham
2 Graduate Students

Gutenberg is an object-oriented operating system being designed to study protection issues in distributed systems [STEM83a, 83b, 83c, 84a, 84b, RAMA83b, 84a]. The Gutenberg system represents a departure from traditional capability based systems in that it adopts a capability scheme for controlling process interactions, but not for local procedure calls and other access within a process. This approach we have called a *nonuniform object model of protection*: Data local to a process *are not* protected by the kernel; resources shared by multiple processes are structured as objects and *are* protected by the kernel.

In Gutenberg, a process may access a shared object only if it has access to a communication channel (termed *port*) that is specifically associated with the object's type, and connected to the object's manager. Port creation is linked to access authorization. The principle underlying the Gutenberg approach to access authorization can be summarized as follows: *Permit a user process to create a port to the process managing a remote, protected object only if the user process has the right to access the object.* This right manifests itself as a capability to access the object with a particular operation. Since access authorization is performed at port creation time (by checking the capability) and the port can only be used to perform a specific operation on the object, the only check that needs to be made when a process requests access to a remote object is whether the port belongs to that process. In Gutenberg, this check is done at the node in which the user process resides; it is a local check. The adoption of non-uniform object-orientation, along with the use of ports for all access to user-defined, shared objects, distinguishes Gutenberg from other capability-based systems. We believe our approach reduces the overheads introduced by the need for total mediation of object access, a requirement of every protection system.

The control of capabilities used to create ports to access protected objects is a central theme in Gutenberg; our approach to this control can be summarized as follows: *All capabilities for accessing shared, protected objects are maintained in a physically distributed but logically unified structure to ensure that they are accessed in a controlled and reliable manner.* This structure is called the *capability directory*, and is similar in structure to the UNIX file directory *[RITC74]*. The deployment of capabilities in this structure is ideal for implementation in a distributed environment: distribution of the capability directory results in a reliable, decentralized access authorization facility. The configuration and maintenance of the directory necessitates the cooperative interaction of kernels on individual nodes. This cooperation poses an interesting research problem, for instance, since a distributed system introduces the need for partitioning and distribution of the capability directory.

The factors that affect the design of the distributed Gutenberg kernel include the distribution of and synchronization of access to the capability directory, and the reliability of kernel and user objects and their managers. In addition, naming of objects, inter-node communication and object location pose problems in any distributed environment and we are attempting to address them in the context of the Gutenberg approach to protection.

The thrust of our current research is to investigate the system support needed for the Gutenberg protection mechanism when implemented in a distributed environment. Whereas efficient implementations of protection-oriented systems has been a problem even in centralized systems, the problem is exacerbated in distributed systems by the need to avoid the use of centralized information and by the need for validating requests for access to distributed objects from processes in distributed sites. Thus, we are interested in examining the performance of a mechanism based on our approach to protection.

Since there are many decisions involved in the design and development of our kernel, we are designing a testbed geared to examining the efficacy of different design choices. However, the current VAX network, due to the limited possibilities of communication among the processors, makes the experimentation with port-based communication extremely difficult and our initial estimates are that the simulation will also be highly inefficient. Hence, due to the lack of appropriate system support, we are currently designing the testbed to provide only logical parallelism.

Though the testbed being designed will help us in the study of some of the conceptual issues relating to protection in distributed systems, it is not appropriate for adequately addressing questions relating to its actual performance. We believe that the proposed multiprocessor and its associated software support will provide the necessary infrastructure for the implementation of a testbed to carry out a complete evaluation of the efficacy of the Gutenberg approach to protection. The tightly-coupled multiprocessor facility with the common shared memory will make the testbed easy to design and will have considerable performance benefits. Overheads of our approach will be measured in terms of the type and amount of messages that need to exchanged, for example to share a remote object and to maintain the distributed capability directory. We will also be examining the performance of the system under the effect of various design choices mentioned above.

6.5.5 Programming Facilities for Parallel Computing

Personnel: Professor Cuny
2 Graduate Students

Funding Agencies: ONR (and an IBM award)

The routine use of massively parallel computers will be possible only after the development of appropriate high-level, programming facilities and it is these facilities that form the focus of our current research. Our goal is not to design a specific programming environment but rather to investigate the necessary programming abstractions and compilation techniques.

As in the case of sequential programming languages, the development of high-level parallel programming languages will require the identification of unifying programming abstractions; an identification that can come only after we have accumulated sufficient programming experience. Unlike the case of sequential machines, however, the requirements of proposed parallel algorithms vary dramatically as do the capabilities of proposed architectures. For these reasons, we intend initially to concentrate on specific classes of algorithms, programming them for a general purpose, MIMD-type machine. We are particularly interested in identifying common patterns of communication and synchronization.

The development of appropriate compilation techniques for high-level, parallel programming languages will naturally depend on the ultimate choice of programming abstractions but some initial requirements can be determined. In particular, the mapping of interconnected, logical processes and their data structures onto existing machines and the translation of their synchronization specifications pose problems not encountered in sequential machines. Since the mapping of arbitrary graphs to machine structures is quite difficult, we would like to design high-level operators that produce "structured" patterns of processor interconnections more ammenable to mapping. (Results here may also be relevant to the automatic decomposition of existing sequential programs for parallel execution.) The translation of high-level synchronization into target code will be complicated either if source and target protocols differ significantly or if there is a possibility of optimization when full generality of the protocols are not needed.

Currently, we are looking at these issues of execution mode; in particular, the compilation of asynchronous code for synchronous machines. (Asynchronous machines often provide a more natural environment from the programmers point of view but can result in inefficient code if the full generality of the handshaking protocols is not needed for a particular program.) Several algorithms have been designed for this conversion

[CUNY84] and we are currently implementing them as part of the Poker Parallel Programming Environment [SNYD84]. Once the algorithms have been incorporated into Poker we will to experimentally evaluate them by simulation with the expectation of identifying a larger class of programs for conversion and developing measures of performance improvement.

The availability of the multiprocessor facility as proposed would be beneficial to the above work for two reasons:

1. it would provide access to a community of users of a highly parallel system whose experience and feedback would be key in the development of any high-level facilities; and

2. it would provide sufficient processing capacity to enable us to simulate a variety of proposed architectures so that programming constructs and compilation techniques could be experimentally evaluated. (Currently Poker provides a backend emmulator of the Pringle architecture which we are adapting to a more general simmulator.)

6.5.6 Distributed Debugging

Personnel: Professors Wileden and Lesser

Funding Agencies: NSF, DARPA, and ONR

Debugging is especially crucial to the development of distributed software systems, whose concurrency, time dependencies, physical distribution and local autonomy make them susceptible to extremely complex and subtle bugs. Yet in recent years, while distributed systems researchers have focused much attention on improving specification techniques and programming methodology, relatively little effort has been expended on developing techniques and tools that will aid in the debugging process for distributed systems.

The result is that debugging complex distributed and real-time programs is a frustrating and difficult task. This is due primarily to the low-level, computation-unit view of systems which is the predominant perspective provided by existing debugging tools and techniques. This perspective is necessarily detail intensive and offers little aid in dealing with the higher level operational characteristics of a system. While this is a drawback for debugging any realistically large and complex software system, it is an especially serious problem for debugging distributed or real-time programs, whose possible behaviors are more numerous and complex and do not possess the time invariant property of sequential programs.

Our work in debugging has emphasized a high-level approach which focuses on patterns of system behavior ([BATE83b, BATE83c]). Debugging is viewed as a process of creating *models* of actual behavior from the activity of the system and comparing these models to the models of expected behavior held by implementors and users of the system. When the models of actual and expected behavior are accurate, the differences between them characterize the errorful behavior of the system. Our goal is to facilitate this modelling process by providing tools and techniques for describing abstractions of system behavior and recognizing the occurrence of these abstractions in the system's activity ([BATE82, BATE83a]). It is not intended that our approach provide the means to create complete formal descriptions of the expected or actual behavior of a system. Instead, the approach encourages users to selectively model aspects of a system which are related to the sources of errorful behavior, involving only a level of detail necessary to understand those aspects. This selective modelling of behavior is termed *behavioral abstraction*.

Within the behavioral abstraction framework, a system's activity is viewed as consisting of a stream of significant, distinguishable behaviors, termed *events*. By selectively clustering sequences of events into higher level events, more abstract views of the system can be created. This approach permits a user to deal with only the level of detail that is appropriate for the currently relevant model of system activity. Using the event based behavioral view for debugging distributed systems allows us to directly investigate complex activity, without appealing to the low, procedure-level details of a computation.

Automated support for debugging has two interrelated aspects, *monitoring* and *intervention*. Monitoring is observation of the system's behavior. Its use is in gathering information to be incorporated into models of actual behavior. Intervention involves altering the system's behavior as a means of testing hypotheses about sources of erroneous behavior. Straightforward extension of monitoring and intervention to distributed software systems leads to variations on what we term *remote* debugging. Monitoring of the system is accomplished by having each node of the system send local information to some central debugging node. Intervention is carried out at nodes participating in the distributed computation (remote nodes), but decisions about when to intervene and what elements to alter originate at the central monitoring site.

The behavioral abstraction approach admits a wide range of distrubuted debugging toolset implementations. The debugging tools become increasingly effective as the toolset implementation becomes increasingly distributed. The possibilities range from remote debugging, which is a minimally distributed realization, up to a maximally distributed implementation that results in a simple form of cooperative debugging. In this maximally distributed configuration many nodes may be active participants in debugging actions, which effectively causes network-wide patterns of debugging activity to occur.

Remote debugging is attractive primarily because of its simplicity. However, when the distributed system has complex interacting components residing at individual nodes or the network architecture becomes more complex (e.g. distributed sensor nets), remote debugging becomes less effective.

Remote debugging does not fully exploit the high-level nature of behavioral abstraction. While model building and evaluation may be conducted from a high-level perspective at the site containing the toolset, intervention must still be carried out from a low-level (primitive event level) perspective. The lack of distributed high-level recognition also limits the ability to intervene based on the context contained in higher level models. High-level debugging, due to the abstraction inherent in the approach, offers the potential for a substantial improvement in the information content of debug-related interprocessor communications. With remote debugging, however it is entirely posssible to introduce unacceptable communication overhead since only low-level (and necessarily voluminous) event information is traversing the system. Greater distribution of the behavioral abstraction-based debuggging toolset to the remote nodes of the system can help to overcome these limitations imposed by remote debugging.

Moving toward a cooperative distributed debugging toolset increases the complexity of the implementation, but it yields a significant payoff in enhanced capabilities. Distribution of the behavioral abstraction-based monitoring and intervention facilities requires each remote node to participate more fully in the distributed debugging task. No longer do the remote nodes simply report and compare primitive events. Instead, remote nodes interpret locally generated primitives in terms of higher level models of behavior. Communication overhead required for operation of the debugging tools will decrease through a change toward communication of high-level event information. Many low-level events will be filtered as unnecessary to abstractions and others incorporated into the high-level abstractions recognized locally and sent to cooperating nodes.

With cooperative distribution of the debugging tools, the tool user gains the capability to deal in system-wide high-level modelling. System monitoring is in many ways simplified because the task is distributed and is higher level as a result of the communication and use of high-level events in abstraction. Intervention is enhanced because the user can now specify actions to be based on higher level event occurrences detected at the remote nodes. Because intervention can now be constrained to begin only in the context of other, related events, fewer unnecessary intervention activities will be performed. Those that are performed are also more accurately performed. Tool transparency is increased through obvious lessening of interruption of local processing for debugging actions and reduction in communication overhead. More general network organizations will be more easily handled. This should be particularly true for mesh organizations such as might be found in a distributed sensor net or a network architecture that has grown to meet the needs of a large organization.

Our research to date has been limited to studying the behavioral abstraction approach implemented as remote debugging from a central site. There are a number of issues that must be explored in assessing the applicability and extendability of our approach to debugging very complex systems, but which are not easily addressed in our current computing environment. First are those related to the complexity of the system being debugged. It is not easy to create extremely complex and subtle patterns of interaction with the limited number of processing sites now available. Approaches to real timing problems due to congestion of communication channels or scheduling of large numbers of processes cannot be fully explored since it is only possible to artifically create the conditions under which this occurs. Similarly, problems due to loss of information or limited access to information because of processor autonomy are issues that are difficult to explore since there are few alternative channels through which to acquire information.

For our debugging tools to be distributed across the network processors, we require an adequate set of high-level interprocess communication facilities. Currently, interprocess communication is only possible through arduous programming with low-level system procedures. Creating a truly distributed implementation of our toolset would be facilitated by a rich set of state-of-the-art facilities such as remote procedure call or connection-based virtual circuits.

We are currently using the distributed vehicle monitoring testbed as a target distributed system to which our behavioral abstraction approach to distributed debugging can be applied. The approach, however, is quite independent of the system that it is being used to examine. A description of the target system's primitive activity and an ability to access that activity is all that is required to permit the use of our debugging toolset. Debugging is an important and necessary phase in the creation of any significant software system. We therefore anticipate that with increased emphasis on creating distributed programs, which the proposed multiprocessor will help to generate, will come increased use of the succession of debugging toolset implementations that we will be developing. Not only will this provide important software development assistance to our colleagues who will be creating the distributed programs, but our understanding of what it takes to debug distributed programs will also be greatly enhanced.

As mentioned, our current debugging toolset resides on a central node and only receives primitive event traffic. The set of interprocess communication and process structuring facilities of the proposed multiprocessor will enable us to distribute the debugging toolset throughout the system in many ways. In addition, the applications developed for the proposed mp will give us a rich set of examples to work out difficulties with our approach.

6.5.7 Modelling and Performance Evaluation of Distributed Systems

Personnel: Professors Towsley and Kurose
3 Graduate Students

Funding Agencies: NSF and Naval Underwater Systems Center

As the use of distributed computer and multiple processor systems become more widespread, it will become increasingly important to model their performance. At this point in time performance modelling and evaluation techniques are well understood for centralized and single processor systems but not for distributed systems. Recent approaches to modelling distributed systems have relied on aggregation and decomposition techniques. Under aggregation, a large portion of the system is replaced by a black box whose performance characteristics are understood. The resulting model then consists of this box in addition to the remainder of the system. A simple model is constructed from these two components that include their interactions. Under decomposition, the system is divided into many parts (each part perhaps containing one processor). Each component is modeled separately. Typically independence assumptions are invoked in order to facilitate the construction of the component models.

Although numerous studies of distributed system performance have relied on such techniques, very little work has been undertaken with the goal of validating these approaches. We see such a validation as an important research problem which can only be attacked through experimental methods. The availability of a testbed such as the multiprocessor is a necessity in order to approach this problem. There are currently several independent efforts in the areas of scheduling algorithms (Professors Ramamritham, Stankovic, and Towsley) database systems (Professors Kohler, Stankovic, and Towsley), networking protocols (Professors Kurose and Towsley), and fault tolerant architectures (Professors Pradhan, Singh, and Towsley of the ECE Department). Each of these efforts contains an evaluation component. If a multiprocessor becomes available, we hope to coordinate these evaluation efforts and determine the applicability and accuracy of models based on aggregation and decomposition. The experience gained through this effort would be of tremendous value to the performance evaluation community at large.

To complement this work in modelling and analysis, we are developing modelling tools for constructing and analyzing (either analytically or through simulation) models of resource contention systems such as computer communication networks. Previous work [SAUE84] has shown that even when presented with high-level modelling constructs and a sophisicated langauge/editor for manipulating these constructs, a system modeler will typically first develop a pictorial or graphical specification of a model, and only then translate this specification into a modelling language such as RESQ. A primary focus of our current work is thus the design, development and implementation of a highly interactive, graphically-oriented modelling tool, with particular emphasis placed on providing the modelling constructs and associated graphics support for developing hierarchical, well-structured performance models. An important part of the projected graphics support supplied to the user will be the use of color in the model development and specification phases. The graphics support and computational facilities currently available are not adequate for either the design/development phase of the modelling tool itself or for for the anticipated subsequent use of this tool by members of the research facility.

The close interaction among the developers of this tool and the modelers within our research community can have an important synergistic effect. The tool itself provides a means by which researchers associated with the multiprocessor facility can rapidly specify, evaluate and modify high-level performance models and thus efficiently estimate the impact of architectural tradeoffs and alternate design considerations. For those researchers directly involved in the field of modelling and performance evaluation (e.g., Professors Towsley and Kurose), it would provide the means for quickly validating analytic models and evaluating various analytic approximation techniques. Researchers using this tool would, in turn, provide valuable feedback, evaluations and suggestions regarding the modelling tool itself.

In summary, the new multiprocessor facility being proposed would significantly enhance our ability to study and refine the various tools and techniques used in the performance evaluation of distributed systems.

We have been exploring an approach to decentralized resource allocation mechanisms in computer networks based on models and methods from mathematical economics. In this approach, network agents act as selfish, utility-maximizing entities and their cooperation via a resource pricing mechanism serves as a decentralized computational device for computing an optimal distribution of network resources. The algorithm itself is an iterative process of local (and hence parallel) optimizations followed by a partial information exchange via the pricing mechanism. This approach has already been successfully applied to the optimization of several channel sharing policies in single-hop multiple access networks (Kurose et. al. 1984a, Kurose et. al. 1984b). Interestingly, several networks mechanisms such as flow control and priorities were found to naturally emerge from this approach.

Currently, these ideas are being extended to other network environments, such as multi-hop packet radio networks. The "microeconomic" approach should be particularly well-suited to this type of environment, since one of its properties is that as the (coupled) resources themselves attain a degree of spatial locality (as in multihop networks), so too does the interaction involving their distribution. Several alternative "pricing" mechanisms are also under investigation.

Although this work addresses fundamental theoretical issues in the design of cooperative decentralized resource sharing mechanisms, there is a very strong experimental aspect to this work as well. We have found that the sufficient (but not necessary) mathematical constraints to guarantee the convergence of the iterative process to be overly restrictive and seldom satisified by resource allocation problems occurring in "real-world" networks. Thus, the converge properties and performance of the resource allocation algorithms can only be studied empirically. Until now, constraints on the available computing resources have limited our studies to simulations of a small (four node) network (Kurose 1984c). While these studies indicate the feasibility of the basic approach and have provided much insight into the issues involved, these approaches must still be examined in a much larger network environment in order to determine their suitability for implementation in real existing networks. The proposed multiprocessor facility would thus provide an ideal setting for pursuing this experimental aspect of our work: the additional computational power would provide the means for not only simulating these algorithm in large scale networks, but also provide the opportunity to implement the algorithms themselves in their distributed form and examine their behavior in this setting as well.

6.6 Related Research in other areas

As outlined in the previous sections, the proposed multiprocessor system will have a great impact on the experimental research. We also expect our work on some of the conceptaul problems dealing with distributed computing to be influenced by the presence of the multiprocessor system. Here we briefly discuss some of this research.

Natural Language Processing

Professors Lehnert and McDonald research in the area of natural language processing ("NLP") is focused on fundamental problems in comprehension ("parsing"), knowledge representation and memory, and generation, and is being applied in projects involving, among other things, intelligent man-machine interfaces, tutoring, and summarizing narratives. NLP is one area where parallel processing is not automatically regarded as a necessary or even a good thing: the essentially stream-like character of natural language texts and the extremely rich system of temporally distributed dependencies that govern their grammatical and rhetorical structure conspire to make the core processes of NLP (i.e. parsing into and generation out of a memory representation) essentially sequential. However, while the *linguistic* processing of text appears to be best suited to conventional sequential architectures, the *semantic and conceptual* processing that accompanies it are potentially another matter entirely. The size and complexity of the knowledge base that the text-manipulating NLP processes draws on has traditionally been the rate-limiting resource of NLP research, and is here that the availability of a large "grain size" multiprocessor system would open up research opportunities that we find very intriguing.

The manipulation of memory representations is by its nature a distributed, multi-focus activity where alternatives and multiple lines of reasoning can, in principle, be carried out simultaneously. Each line of reasoning would involve executing its own large LISP process, making it the kind of application of parallelism that calls for relatively moderate numbers of quite large processors. We are interested in exploring the use of the multiprocessor facility for this problem because as we increase the sophiscation of our NLP processing we find that a high percentage of this new activity involves this semantic processing.

Protocol Verification

One of our areas of concern in computer networks is the unification of the functional (correctness) and performance analysis of communication protocols. Our work is motivated by the belief that protocols are, among other things, real-time software and as such, their design should be guided by a mixture of functional and performance objectives. Therefore, the classical software paradigm of "correctness first, then performance" is no longer valid. A new theory must capture this mixture of design considerations rather than draw artificial boundaries.

Previous work (Yemiini & Kurose 1982) has demonstrated important instances in which existing protocol verification techniques are not sufficient to capture some critical aspects of protocol correctness. Our current efforts are focusing on the unification of functional and performance considerations through the extension of the semantics for serially executing probabilistic programs proposed by Kozen (Kozen 1979).

Design and Analysis of Distributed Systems

Professor Jack Wileden is developing techniques for describing and analyzing designs of distributed systems. This work is in collaboration with Professor George Avrunin of Mathematics. The research differs from other work on distributed system design techniques largely because it emphasizes the role of analysis in design. A design notation has been defined that is suitable for describing a wide range of distributed systems, including dynamically structured distributed systems [WILE82]. Analysis techniques based on elementary algebra have been developed for designs specified in the design notation [AVRU83]. Extensions to the design notation and analysis techniques are currently being explored, as are automated tools supporting both the notation and the analysis techniques.

In conjunction with this work, Wileden and Avrunin are developing a formal framework for studying distributed systems and their behavior. The constrained expression formalism allows one to derive a closed form representation for the potentially infinite set of behaviors corresponding to a distributed computational system. This representation then provides a basis for analyzing the system's behavior. To date, this work has been applied to the distributed system design notation, to Petri nets, and to a subset of Hoare's Communicating Sequential Processes (CSP) notation. Currently, the analysis methods for the constrained expression framework are being strengthened, extended, and automated.

Specification and Verification of Resource Controller Processes

Professor Ramamritham is developing specification, design, and implementation techniques for distributed system software by focussing on the processes that manage shared resources. Two important issues in distributed system design are interactions among processes and the distribution and management of shared resources. A distributed system is viewed as a set of processes and a set of resources shared by these processes. Committing a process to manage a shared resource limits interactions and thus has the benefit of enabling modular designs for distributed systems.

Our investigations into the specification, design and implementation of resource controllers through grants from the National Science Foundation has led to (1) a language for specifying the properties of resources and their controllers [RAMA83d], (2) schemes for verifying extant resource controllers - both when control is exercised in a centralized manner [RAMA82a] as well as in a distributed manner [RAMA83e], and (3) the design of a system for automatically synthesizing code for resource controllers given the specifications for the resources and their controllers [RAMA83a, 83c]. Whereas the synthesis technique can be used to construct correct resource controllers, the verification technique can be applied to extant resource controllers to guarantee their correctness. Thus the results obtained so far should aid in developing better specification techniques and in implementing reliable distributed system software.

Software Development environments

In an effort to overcome the shortcomings in current approaches to software development, researchers in the Software Development Laboratory, led by Professors Clarke and Wileden, have been pursuing an approach that we call *feedback-directed development* [WILE84]. An SDE supporting this approach would employ *consistent abstractions* and provide *analysis tools* (i.e., tools providing feedback about the quality of the system being produced) applicable throughout the entire software development process.

We view appropriate and consistent sets of abstractions as the key to integration in the software development process. Having appropriate sets of abstract concepts for describing various aspects of a computational system is of crucial importance to software developers. Even the most appropriate abstractions will be of limited value, however, unless the same set, or closely related sets, of abstractions can be used to describe a software system during each phase of the development process. Such consistent sets of abstractions are conspicuously lacking in current approaches to software development, where at each phase of development (e.g., specification, design, program) a software system is described in terms of vastly different sets of concepts. Appropriate and consistent sets of abstractions would provide a basis for thorough integration of the languages and tools constituting an SDE. In particular, they would facilitate the application of nearly identical analysis tools to descriptions at various stages of development and various levels of detail. By providing analysis capabilities that are available throughout the software development process, the environment would support continual reasoning about the properties of an evolving software system. This ongoing reasoning would guide the SDE user in evaluating decisions, exploring alternatives and smoothly progressing toward a completed software system. In some cases, this reasoning might be performed by the environment itself. In other cases, the environment would simply augment the reasoning abilities of its human user by providing insightful information. In any event, the environment's consistent abstractions and its continually available analysis capabilities would provide the basis for that reasoning, and hence for an exploratory, feedback-directed development process.

We are presently constructing and evaluating a prototype SDE supporting the feedback-directed approach to software development. This prototype is based upon three aspects of our current research:

1. language constructs and analysis tools supporting precise description of modularity and interfaces within a software system [CLAR83].

2. the constrained expression framework for description and analysis of behavior in concurrent/distributed systems [AVRU83, WILE82].

3. rigorous and systematic testing methods applied throughout the software development process [RICH81, 82].

Distributed knowledge acquisition in intelligent interfaces

A major problem with carrying out complex tasks on current distributed computer systems is that the user/system interface is at the level of the system tools rather than the user's tasks. This means that, in a typical environment, the users must keep track of the activities involved in a set of concurrent, distributed tasks as well as deciding which system tools are needed for these activities. A system that does not understand a user's intent cannot provide assistance at the task level and cannot explain or summarize a user's actions to other people who may be involved in the same tasks. Both of these capabilities are essential for a truly effective interface. The aim of the proposed research is to develop a system that, based on a representation of user tasks, can provide the following facilities;

1. Assistance: This includes functions such as providing task agendas, error-checking, providing default values, carrying out simple tasks and coordinating complex distributed tasks.

2. Knowledge acquisition: The system should acquire knowledge about new tasks and the reasons for particular user actions. These tasks can involve many people using many different workstations.

3. Explanation: To demonstrate its understanding of the user's activities, the interface should be able to describe the status of current tasks and explain how previous tasks were carried out.

This research is currently being applied to the office system and software development areas [CROF82]. An interesting feature of the system is that it requires the integration of a number of techniques from different areas of AI such as planning, natural language interfaces, expert systems and knowledge representation.

The formalism being used to represent the tasks is an extension of the EDL formalism used by Bates and Wileden in their work on distributed debugging [BATE82]. It combines a procedure-based and a goal-based representation. This combination allows flexibility in choosing interpretation or planning strategies so that tasks can either be passively monitored or actively invoked by the system. A major research issue that arises in this project is how to capture and represent the reasons for a user choosing a particular action during a task. This knowledge will enable the system to explain a user's activities to other people and provide better assistance to inexperienced users carrying out the same tasks.

References

[ADIV83] Adiv, G., "Recovering Motion Parameters in Scenes Containing Multiple Moving Objects," *Proceedings of CVPR*, Washington, D.C., June 1983, pp.399-400.

[ADIV84] Adiv, G., "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," Department of Computer and Information Science, Technical Report 84-07, University of Massachusetts at Amherst, April 1984.

[ANAN84] Anandan, P., "Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion," Department of Computer and Information Science, Technical Report, University of Massachusetts at Amherst, forthcoming.

[ANDE82] Anderson, C. W.: Feature generation and selection by a layered network of reinforcement learning elements: Some initial experiments, Department of Computer and Information Science, Technical Report 82-12, University of Massachusetts, Amherst, 1982.

[ARBI81] Arbib, M., "Perceptual Structures and Distributed Motor Control," (V.B. Brooks, Ed.), *Handbook of Physiology - The Nervous System III, Motor Control*, Bethesda, Maryland, American Physiological Society, 1981, pp.1449-1480.

[ARBI82a] Arbib, M.A., (Ed.), *Proceedings Of The Workshop On Visuomotor Coordination In Frog And Toad: Models And Experiments*, Department of Computer and Information Science, Technical Report 82-16, 1982.

[ARBI82b] Arbib, M.A., "Rana Computatrix: An Evolving Model Of Visuomotor Coordination In Frog And Toad," *Machine Intelligence*, 10, 1982, pp.501-517.

[ARBI83a] Arbib, M., Lawton, D., and Overton, K., "Perceptual Systems for Robots," Department of Computer and Information Science, Technical Report 83-24, University of Massachusetts, 1983.

[ARBI83b] Arbib, M.A., Iberall, T., and Lyons, D., "Coordinated Control Programs For Movements Of The Hand," Exp. Brain Res. Suppl. (in press), (also Department of Computer and Information Science, Technical Report 83-25, University of Massachusetts, Amherst), 1983.

[ARBI84] Arbib, M. and Lyons, D., "A Greaphics System for Hand Simulation", in preparation, 1984.

[AVRU83] Avrunin, G.S. and Wileden, J.C., "Describing and Analyzing Distributed

Software System Designs," University of Massachusetts, Department of Computer and Information Science, Technical Report 83-28, August 1983, (submitted for publication).

[BARR78] Barrow, H. and Tenebaum, J.M., *Computer Vision Systems*, (A. Hanson and E. Riseman, eds.), Academic Press, 1978.

[BART81a] Barto, A. G., Sutton, R. S., Brouwer, P.: Associative search network: A reinforcement learning associative memory. Biol. Cybernetics, *40*, 201-211 (1981)

[BART81b] Barto, A. G., Sutton, R. S.: Landmark learning: An illustration of associative search. Biol. Cybernetics, *42*, 1-8 (1981)

[BART83 Barto, A. G., Sutton, R. S., Anderson, C. W.: Neuron-like adaptive elements that can solve difficult learning control problems. IEEE Trans. Syst., Man, and Cybern., vol. SMC-13, pp.834–846, 1983.

[BART84] Barto, A. G., Anderson, C. W.: On the collective behavior of pattern-recognizing stochastic learning automata. In progress.

[BATE82] Bates P. and Wileden, J.C., "Event Definition Language: An Aid To Monitoring And Debugging Of Complex Software Systems," *Proceedings of the Fifteenth Hawaii International Conference on System Sciences*, January 1982, pp.86-93.

[BATE83a] Bates, P.C., Wileden, J.C., and Lesser, V.R., "A Debugging Tool for Distributed Systems," *Proceedings of the Phoenix Conference on Computers and Communications*, March 1983.

[BATE83b] Bates P. and Wileden J., "An Approach to High-Level Debugging of Distributed Systems," *Proceedings of the Software Engineering Symposium on High-Level Debugging*, Asilomar, California, March 1983.

[BATE83c] Bates and Wileden J., "High Level Debugging of Distributed Systems," *Journal of Systems and Software*, 3, 4, December 1983, pp.255-264.

[BEGE84] Begej, S., "An Optical Array Sensor", *Proceedings of Intelligent Robots and Computer Vision*, Cambridge, Massachusetts, 1984.

[BOKH79] Bokhari, S.H., "Dual Processor Scheduling with Dynamic Reassignment," *IEEE Transactions on Software Engineering*, SE-5, 4, (July 1979).

[BURN84] Burns, J.B., Hanson, A.R., and Riseman, E.M., "Extracting Linear Features," *Proceedings of the Seventh International Conference on Pattern Recognition*, Montreal, Canada, July 30 - August 2, 1984.

[CASE77] Casey, L. and Shelness, N., "A Domain Structure for Distributed

Computer Systems," *Proceedings of the Sixth ACM Symposium on Operating Systems Principles*, (November 1977), pp.101-108.

[CERV83] Cervantes, F. and Arbib, M., "Models of a Neural Net Subserving Worm-Antiworm Discrimination in Frogs and Toads," *Proceedings of the Thirteenth Annual Meeting of the Society for Neurosciences*, Vol.9, Part I, 1983, p.536.

[CHOW79] Chow, Y.-C. and Kohler, W.H., "Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System," *IEEE Transactions in Computers*, C-28, 5, (May 1979).

[CLAR83] Clarke, L.A., Wileden, J.C. and Wolf, A.L., "Precise Interface Control: System Structure, Language Constructs, and Support Environment," University of Massachusetts, Department of Computer and Information Science, Technical Report 83-26, August 1983.

[CORK80] Corkill, D.D., "An Organizational Approach to Planning in Distributed Problem Solving Systems," Department of Computer and Information Science, Technical Report 80-13, University of Massachusetts, Amherst, May 1980.

[CORK82a] Corkill, D.D., Lesser, V.R., and Hudlicka, E., "Unifying Data-Directed and Goal-Directed Control: An Example and Experiments," *Proceedings of the Second National Conference on Artificial Intelligence*, August 1982, pp.143-147.

[CORK82b] Corkill, D.D., "A Framework for Organizational Self-Design in Distributed Problem Solving Networks," Ph.D. Thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, September 1982.

[CROF82] Croft, W.B. and Lefkowitz, L., "POISE - An Intelligent Assistant for Profession-Based Systems," Department of Computer and Information Science, Technical Report, University of Massachusetts, Amherst, (1982)

[CORK83] Corkill, D.D. and Lesser V.R., "The Use Of Meta-Level Control For Coordination In A Distributed Problem Solving Network," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, August 1983, pp.748-756.

[CUNY83] CUNY, J.E., "Compilation of Data-Driven Programs for Synchronous Execution," with Lawrence Snyder, *Proceedings of the 1983 Annual Symposium on Principles of Programming Languages*, January 1983, pp.197-202.

[CUNY84] Cuny, J.E. and Snyder, L., "Testing the Coordination Predicate", IEEE Trans. on Computers, March, 1984.

[DILL84] Dillon, L.K., "Analysis of Distributed Systems Using Constrained Expressions," Ph.D. Dissertation and Technical Report, Department of Computer and Information Science, University of Massachusetts, August 1984.

[DURF84] Durfee, E.H., Corkill, D.D., and Lesser, V.R., "Distributing A Distributed Problem Solving Network Simulator," To appear in *Proceedings of the Fifth Real-Time Systems Symposium*, December 1984.

[ELLI84] Ellis, R., "Extraction of Tactile Features by Passive and Active Sensing", *Proceedings of Intellegent Robots and Computer Vision*, Cambridge, Massachusetts, 1984.

[ERMA80] Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, D.R., "The Hearsay-II Speech Understanding System: Integrating Knowledge To Resolve Uncertainty," *Computing Surveys*, 12(2), June 1980, pp.213-253.

[FELD81] Feldman, J. A.: A connectionist model of visual memory. In: Parallel models of associative memory. Hinton, G., Anderson, J. (eds.). Hillsdale, N. J: Erlbaum 1981

[GABR84] Gabriel, Richard P., and McCarthy, John, "Queue-based multi-processing LISP," *Proceedings of the 1984 ACM Conference on LISP and Functional Programming*, pages 25-44, August 1984.

[GLAZ83] Glazer, F., Reynolds, G. and Anandan, P., "Scene Matching by Hierarchical Correlation," *Proceedings of Computer Vision and Pattern Recognition Conference*, Arlington, Virginia, June 1983.

[HANS78a] Hanson, A. and Riseman, E., "Segmentation of Natural Scenes," A. Hanson and E. Riseman (Eds.), *Computer Vision Systems*, New York, Academic Press, 1978, pp.129-163.

[HANS78b] Hanson, A. and Riseman, E., "VISIONS: A Computer System For Interpreting Scenes," A. Hanson and E. Riseman (Eds.), *Computer Vision Systems*, New York, Academic Press, 1978, pp.303-333. 1983.

[HANS80] Hanson, A.R. and Riseman, E.M., "Processing Cones: A Computational Structure of Image Analysis," in *Structured Computed Vision*, (S. Tanimoto, ed.), Academic Press, 1980.

[HANS83] Hanson, A.R. and Riseman, E.M., "A Summary of Image Understanding Research at the University of Massachusetts," Department of Computer and Information Science, Technical Report 83-35, University of Massachusetts at Amherst, October 1983.

[HALS84] Halstead Jr., Robert H., "Implementation of MultiLISP: LISP on a multiprocessor," *Proceedings of the 1984 ACM Conference on LISP and*

Functional Programming, pages 9-17, August 1984.

[HEWI77] Hewitt, C., "Viewing Control Structures As Patterns Of Passing Messages," *Artificial Intelligence* 8(3), Fall 1977, pp.323-364.

[HINT81] Hinton, G. E., Anderson, J. A. (eds.): Parallel models of associative memory. Hillsdale, N. J.: Erlbaum 1981

[HINT83] Hinton, G. E., Sejnowski, T. J.: Analyzing Cooperative Computation. Proceedings of the Fifth annual conference of the Cognitive Science Society, Rochester N.Y., May 1983.

[HOUS82] House, D., "The Frog/Toad Depth Perception System: A Coorporative/Competitive Model," *Proceedings of the Workshop on Visuomotor Coordination in Frog and Toad: Models and Experiments*, Department of Computer and Information Science, Technical Report 82-16, University of Massachusetts, Amherst, 1982.

[HUDL84] Hudlicka, E. and Lesser, V., "Meta-Level Control Through Fault Detection And Diagnosis," *Proceedings of The Fourth National Conference on Artificial Intelligence*, August 1984, pp.153-161.

[IBER84] Iberall, T. and Lyons, D., "Towards Perceptual Robotics," Department of Computer and Information Science, Technical Report 84-17, University of Massachusetts, 1984.

[IBER84] Iberall, T. and Lyons, D., "Towards Perceptual Robots," presented at *IEEE International Conference on Systems, Man, and Cybernetics*, Technical University of Nova Scotia, Nova Scotia, October 9-12, 1984.

[KAPA84] Kapauan, A., Wang, K., Cuny, J., Gannon, D., and Snyder, L., "The Pringle: An Experimental System For Parallel Algorithm Testing and Software Development," *Proceedings of The 1984 International Conference on Parallel Processing*, 1984, pp.1-6.

[KOHL81] Kohler, W., "A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems," *ACM Computing Surveys*, Vol.13, No.2, June 1981.

[KOHL83] Kohler, R.R., "Integrating Non-Semantic Knowledge into Image Segmentation Processes," Ph.D. Dissertation and Department of Computer and Information Science, Technical Report 84-04, University of Massachusetts at Amherst, September 1983.

[KOHL83a] Kohler, W., Garcia-Molina, H., and Germano, F., "Architectural Overview of a Distributed Software Testbed," *Proceedings of the 16th Annual Hawaii International Conference on System Sciences*, January 1983.

[KOHL83b] Kohler, W., Wilner, K., and Stankovic, J., "An Experimental Comparison of Locking Policies in a Testbed Database System," *Proceedings of SIGMOD*, May 1983.

[KOHL83c] Kohler, W. and Jenq, B., "CARAT: A Distributed Software Testbed," University of Massachusetts/Amherst, Technical Report, December 1983.

[KORN79] Kornfeld, W.A., "ETHER: A Parallel Problem Solving System," *Proceedings of the Sixth International Joint Conference on Artifical Intelligence*, August 1979, pp.490-492.

[KOZE79] Kozen, D., "Semantics of Probabilistic Programs", *Proceedings of the 20th Sym. on the Foundations of Computer Science*, 1979.

[KURO84a] Kurose, J., Schwartz, M., and Yemini, Y., "Multiple Access Protocols and Time-Constrained Communication", *Computing Surveys*, March 1984.

[KURO84b] J. Kurose, M. Schwartz, Y. Yemini, "A Microeconomic Approach to Optimizing Channel Access in Multiaccess Networks", Technical Report, Department of Computer Science, Columbia University, New York, New York, 1984. (submitted for publication)

[KURO84c] J. Kurose, "Time-Constrained Communication in Mutliple Access Networks", PhD Thesis, Department of Computer Science, Columbia University, New York, New York, 1984.

[LARA82] Lara, R., Cervantes, F., and Arbib, M., "Two-Dimensional Model of Retino-Tectal-Pretectal Interactions for the Control of Prey-Predator Recognition and Size Preference in Amphibia," *Competition and Cooperation in Neural Nets*, (S. Amari and M. Arbib, Eds.), Lecture Notes in Biomathematics 45, Springer-Verlag, 1982, pp.371-393.

[LAWT83] Lawton, D. and Rieger, J., "The Use of Difference Fields in Processing Sensor Motion," *Proceedings of the DARPA Image Understanding Workshop*, Arlington, Virginia, June 1983.

[LAWT84] Lawton, D., "Processing Dynamic Image Sequences from a Moving Sensor," Ph.D. Dissertation and Department of Computer and Information Science, Technical Report 84-05, University of Massachusetts at Amherst, February 1984.

[LAWT83] Lawton, D. Processing translational motion sequences. *Computer Vision Graphics and Image Processing*, April 1983.

[LEE84] Lee, Y., "Population Model of Frog's Retina," *NIH.. Symposium of Mathematics and Computer Science in Biomedical Applications*, N.I.H., Maryland, 1984.

[LEE83] Lee, Y. and Arbib, M., "Computer Model of Cone Pathways in Frog's Retina," *Proceedings of the Thirteenth Annual Meeting of the Society for Neurosciences*, Vol.9, Part I, 1983, p.686.

[LENA75] Lenat, D.B., "Beings: Knowledge As Interacting Experts," *Proceedings of the Fourth International Joint Conference on Artifical Intelligence*, August 1975, pp.126-133.

[LESS81] Lesser V.R. and Corkill, D.D., "Functionally-Accurate, Cooperative Distributed Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1), January 1981, pp.81-96.

[LESS83] Lesser ,V.R. and Corkill ,D.D., "The Distributed Vehicle Monitoring Testbed: A Tool For Investigating Distributed Problem Solving Networks," *AI Magazine*, 4(3), Fall 1983, pp.15-33.

[LESS83] Lesser, V. and Corkill, D., "The Distributed Vehicle Monitoring Testbed", The AI Magazine, 4(3):15-33, 1983.

[LEVI84a] Levitan, S., "Parallel Algorithms and Architectures: A Programmer's Perspective," Ph.D. Dissertation and Department of Computer and Information Science, Technical Report 84-05, University of Massachusetts at Amherst, May 1984.

[LEVI84b] Levitan, S., Weems, C., and Riseman, E., "Signal to Symbols: Unblocking the Vision Communications/Control Bottleneck," *Proceedings of the 1984 IEEE Workshop on VLSI Signal Processing*, November 27-29 1984, University of Southern California, Los Angeles, California.

[LOW82] Lowrance, J., "Dependency-Graph Models of Evidential Support," Ph.D. Dissertation and Technical Report 82-26, Department of Computer and Information Science, University of Massachusetts at Amherst, September 1982.

[MARR82] Marr, D., *Vision*, W.H. FReeman, San Francisco, 1982.

[MIRC84] Mirchandaney, Ravi, "Decentralized Job Scheduling Using a Network of Stochastic Learning Automat," Masters Thesis, University of Massachusetts/Amherst, August 1984.

[MOSS82] Moss, J.E.B., "Nested Transactions and Reliable Distributed Computing", *Symp. on Reliability in DIstributed Software and Database Systems*, (July) 1982.

[NAGI82] Nagin, P.A., Hanson, A.R., and Riseman, E.M., "Studies in Global and Local Histogram-Guided Relaxation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-4, No.3, May 1982, pp.263-277.

[OUST80] Ousterhout, J.K., et al., "Medusa: An Experiment in Distributed Operating System Structure," *Communications of the ACM*, 17, 7, (February 1980), pp.92-105.

[OVER84] Overton, K., "The Acquitition, Processing, and Use of Tactile Sensor Data in Robot Control", Ph.D. dissertation, Department of Computer and Information Science, University of Massachusetts, 1984.

[PARM80] Parma, C., Hanson, A., & Riseman, E. Experiments in schema-driven interpretation of a natural scene. Department of Computer and Information Science, Technical Report 80-10, 1980. Also in R. Haralick & J. C. Simon (Eds.), *Nato advanced study institute on digital image processing*, Bonas, France, 1980.

[RAMA82a] Ramamritham, K. "Proof Techniques for Resource Controller Processes," submitted to *Acta Informatica*, June 1982, (revised June 1984).

[RAMA83a] Ramamritham, K., "Synthesizing code for Resource Controllers," submitted to *IEEE Transactions on Software Engineering*, March 1983.

[RAMA83b] Ramamritham, K., Vinter, S., and Stemple, D., "Primitives for accessing protected objects," *Third Symposium on Reliability in Distributed Software and Database Systems*, October 1983.

[RAMA83c] Ramamritham, K. and Sunderrajan, P., "Automatic Generation of code for Resource Controller Tasks in Ada," *Symposium on the Application and Assessment of Automated Tools for Software Engineering*, San Fransisco, November 1983.

[RAMA83d] Ramamritham, K. and Keller, R.M., "Specification of Synchronizing Processes," *IEEE Transactions on Software Engineering*, November 1983.

[RAMA83e] Ramamritham, K. "Correctness of A Distributed Transaction System," *Information Systems*, Vol.8, No.4, 1983.

[RAMA84a] Ramamritham,K., Briggs, D., Stemple, D., and Vinter, S., "Privilege Transfer and Revocation in a Port-Based System," submitted to *IEEE Transactions on Software Engineering*, January 1984.

[RAMA84b] Ramamritham, K. "Resource Controller Tasks in Ada: their Structure and Semantics" *Seventh International Conference on Software Engineering*, March 1984.

[RAMA84c] Ramamritham, K., "Enabling Local Actions by Global Consensus", submitted to *Information Systems*, March 1984.

[RAMA84d] Ramamritham, K. and Stankovic, J.A., "Dynamic Task Scheduling in Hard Real-Time Distributed Systems," *IEEE Software*, July 1984.

[RAMA84e] Ramamritham, K., "Channel Access Delays in Local Area Hard Real-time Systems", submitted to *Fifth International Conference on Distributed Computing Systems*, September 1984.

[REYN84] Reynolds, G., Irwin, N., Hanson, A., Riseman, E., "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," *1984 Proceedings of the Workshop on Computer Vision Representation and Control*, Annapolis, Maryland, April 30 - May 2, 1984, pp.238-247.

[RICH81] Richardson, D.J. and Clarke, L.A., "A Partition Analysis Method to Increase Program Reliability", *Proceedings of the Fifth International Conference on Software Engineering*, March 1981.

[RICH82] Richardson, D.J. and Clarke, L.A., "On the Effectiveness of the Partition Analysis Method," *Proceedings of the IEEE Sixth International Computer Software and Applications Conference*, November 1982.

[RIEG83] Rieger, J., & Lawton, D. Sensor motion and relative depth from difference fields of optic flows. *Proceedings of IJCAI-83*, Karlsruhe, W. Germany, August 1983, 1027-1031.

[RITC74] Ritchie, D. and Thompson, K., "The UNIX Time-Sharing System," *Communications of the ACM*, vol. 17, no. 7, July, 1974.

[SAUE84] C. Sauer, E. MacNair, J. Kurose, "Queueing Network Simulation of Computer Communication", *IEEE Journal on Selected Topics in Communications*, Vol. SAC-2, No. 1, Jan. 1984, pp.203-220.

[SMIT81] Smith, R.G. and Davis, R., "Frameworks For Cooperation In Distributed Problem Solving, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1), January 1981, pp.61-70.

[SNYD84] Snyder, Lawrence, "Parallel programming and the Poker Programming Environment," *Computer*, 17(7), 1984, pp.27-37.

[STAN79] Stankovic, J.A. and van Dam, A., "Research Directions in (Cooperative) Distributed Processing," *Research Directions in Software Technology*, MIT Press, Cambridge, Massachusetts, 1979.

[STAN80] Stankovic, J.A., "A Comprehensive Framework for Evaluating Decentralized Control," *Proceedings of the 1980 International Conference on Parallel Processing*, (August 1980).

[STAN81] Stankovic, J.A., "ADCOS - An Adaptive, System-Wide, Decentralized Controlled Operating System," University of Massachusetts/Amherst, Technical Report, ECE-CS-81-2, 1981.

[STAN83a] Stankovic, J.A., Ramamritham, K., and Kohler, W.H., "Current Research and Critical Issues in Distributed Software Systems," presented at the NSF sponsored Information Technology Workshop, January 1983.

[STAN83b] Stankovic, J.A., "Simulations of Three Adaptive Decentralized Controlled, Job Scheduling Algorithms," to appear, *Computer Networks*, 1983.

[STAN83c] Stankovic, J.A., "Bayesian Decision Theory and Its Application to Decentralized Control of Job Scheduling," to appear, *IEEE Transactions on Computers*, 1983.

[STAN84a] Stankovic, J.A. and Sidhu, I.S., "An Adaptive Bidding Algorithm for Processes, Clusters, and Distributed Groups," *Proceedings of the 4th International Conference on Distributed Computing*, May 1984.

[STAN84b] Stankovic, J.A., "On Detecting and Recovering From Database Partitioning," University of Massachusetts/Amherst, Technical Report, in preparation, September 1984.

[STAN84c] Stankovic, J.A., "Perspectives on Distributed Computer Systems," invited paper, Special Issue in Celebration of IEEE Centennial, *IEEE Transactions on Computers*, to appear, December 1984.

[STAN85] Stankovic, J.A., Stability and Distributed Scheduling Algorithms," invited paper, *Proceedings of the ACM Natural Conference*, to appear, March 1985.

[STEM83a] Stemple, D., Ramamritham, K., Vinter, S., and Sheard,T., "Operating System Support for Abstract Database Types," chapter in ICOD-2, *Proceedings Second International Conference on Databases*, S.M.Deen and P.Hammersley, eds., Wiley Heyden Ltd., August-September 1983, pp.179-195.

STEM83b] Stemple, D., Ramamritham, K., and Vinter, S., "Interprocess Communication Without Process Identification," submitted to *IEEE Transactions on Software Engineering*, January 1984.

[STEM83c] Stemple, D., Ramamritham, K., and Vinter, S., "Operating System Support for Abstract Database Types," *Proceedings of the 2nd International Conference on Databases*, September, 1983.

[STEM84a] Stemple, D., Vinter, S. and Ramamritham,K., "Types and Cooperation Classes: Interprocess Communication Without Process Identifiers", submitted to *IEEE Transactions on Software Engineering*, January 1984.

[STEM84b] Stemple, D., Vinter, S., and Ramamritham, K., "Dynamic Control of Module Interconnections", submitted to *IEEE Software*, August 1984.

[STON77] Stone, H.S., "Multiprocessor Scheduling with the Aid of Network Flow

Algorithms," *IEEE Transactions on Software Engineering*, SE-3, (January 1977).

[STON78] Stone, H.S., "Critical Load Factors in Distributed Computer Systems," *IEEE Transactions on Software Engineering*, SE-4, (May 1978), pp.254-258.

[SUTT81] Sutton, R. S., Barto, A. G.: Toward a modern theory of adaptive networks: Expectation and prediction. Psychol. Rev., *88*, 135-171 (1981)

[SUTT84] Sutton, R. S: Temporal aspects of credit assignment in reinforcement learning, Univ. of Massachusetts Ph.D. Dissertation, 1984.

[TANT84a] Tantawi, A. and Towsley, D., "Optional Load Balancing in Distributed Computer Systems," IBM Technical Report, RC 10346, (_ 46163), January 1984.

[TANT84b] Tantawi, A. and Towsley, D., "A General Model For Optional Load Balancing in Star Network Configurations," submitted to *Performance 84*, May 1984.

[WEEM83] Weems, C., Levitan, S., Foster, C., et al, "Titanic Information Kit," Department of Computer and Information Science, Technical Report 83-32, University of Massachusetts at Amherst, 1984.

[WEEM84] Weems, C., "Image Processing with a Content Addressable Array Parallel Processor," Ph.D. Dissertation and Technical Report 84-14, Department of Computer and Information Science, University of Massachusetts at Amherst, September 1984.

[WESL82] Wesley, L. and Hanson, A., "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning About Control," *Proceedings of the IEEE Workshop on Computer Vision*, Rindge, New Hampshire, August 1982, pp.14-25.

[WESL84] Wesley, L., "Reasoning about Control: An Evidential Approach," SRI Technical Note 324, June 1984.

[WEYM83] Weymouth, T., Griffith, J., Hanson, A. and Riseman, E., "Rule Based Strategies for Image Interpretation," *Proceedings of the DARPA Image Understanding Workshop*, June 1983, Arlington, VA. A shortened version also appeared in *Proceedings AAAI*, August 1983, Washington, D.C.

[WEYM84] Weymouth, T., "Using Object Descriptions in a Schema Network for Machine Vision," Ph.D. Dissertation, Department of Computer and Information Science, University of Massachusetts at Amherst, in progress.

[WILE82] Wileden, J.C., "Constrained Expressions and the Analysis of Designs for Dynamically-Structured Distributed Systems," *Proceedings of the*

International Conference on Parallel Processing, August 1982.

[WILE84] Wileden, J.C. and Clarke, L.A. "Feedback-Directed Development of Complex Software Systems," *Proceedings of the Software Process Workshop*, January 1984.

[WOLF83] Wolf, A.L., Clarke, L.A. and Wileden, J.C., "A Formalism for Describing and Evaluating Visibility Control Mechanisms," University of Massachusetts, Department of Computer and Information Science, Technical Report 83-34, October 1983.

[YEMI82] Y. Yemini and J.F. Kurose, "Can Current Protocol Verification Techniques Guarantee Correctness?", *Computer Networks*, Vol. 6, Dec. 1982, pp.377-381.

[ZHAO84] Zhao, W., Ramamritham, K., and Stankovic, J.A., "Scheduling Tasks with Resource Requirements in Hard Real-Time Systems", submitted to *IEEE Transactions on Software Engineering*, May 1984.

[ZISM78] Zisman, M.D., "Use Of Production Systems For Modeling Asynchronous, Concurrent Processes," (D.A. Waterman and Frederick Hayes-Roth, eds.), *Pattern-Directed Inference Systems*, Academic Press, 1978, pp.53-68.

CURRICULUM VITA OF VICTOR R. LESSER

July 1984

Name: Victor R. Lesser

Soc. Sec. No.: 061-36-8038

Business Address:
Computer and Information
Science Department
University of Massachusetts
Amherst, MA 01003
(413) 545-1322 or 2744

Born:
November 21, 1944,
New York, New York

Citizenship: U.S.A.

Marital Status: Married

Education:

University	Dates Attended	Degrees
Stanford University Stanford, California	1966 to 1972	M.S. & Ph.D. in Computer Science
Cornell University	1962 to 1966	A.B. in Mathematics

Current Employment:

Professor in Computer and Information Science Department, University of Massachusetts (1982-). (On Sabbatical during 1983, Visiting Scientist at MIT Artificial Intelligence Laboratory.)

Previous Employment:

Associate Professor in Computer and Information Science Department, University of Massachusetts (1978-1982).

Assistant Professor in Computer and Information Science Department, University of Massachusetts (1977-1978).

Research Computer Scientist, Computer Science Department, Carnegie-Mellon University (1974-1977).

Consultant: (1982-)

Consultant, GTE Laboratories (1984-)

Scientific Advisory Board, Scientific Leasing Inc. (1984-).

Board of Directors, Visual Intelligence Corp. (1983-).

Consultant, Lincoln Laboratories (1983-).

Consultant, ALPHATECH (1983-).

Consultant, TEKNOWLEDGE (1982-).

Consultant, SYMANTEC (1982).

Honors: (1982-)

Invited lecturer, Instituto di Scienze Dell'Informazione, Universita' di Pisa, Pisa, Italy, on "Distributed Problem Solving", 1982. Presented a two-day lecture on my research in this area.

Research Administrative Responsibilities: (1982-)

Principal Investigator - "Cooperative Distributed Problem-Solving", Joint-NSF/DARPA Funding (1978-1985). $1,648,000.

Principal Investigator, "Intelligent User Interfaces", Digital Equipment Corporation, (1982-1984), $150,000.

Laboratory Director for the COINS departmental research computing facility (1978-1983). As part of this responsibility, I was actively involved in acquiring funds to build this laboratory into a modern state-of-the-art facility. In conjunction with Professor Edward Riseman, I secured for the department three National Science Foundation (NSF) equipment grants and three Digital Equipment Corporation corporate gifts totaling approximately $2,000,000 for the years 1977-1982.

Research Profile:

I have made contributions to the areas of artificial intelligence, computer architecture, distributed processing, microprogramming, and parallel processing. The theme that underlies these diverse interests is the organization and control of complex computational structures. My approach to this research theme has included a strong experimental component. I have either built or simulated in detail the systems architectures that I constructed in each of the different areas so that an in-depth performance analysis was possible. Recent work has also included some first steps to developing theories and models that permit the analysis and prediction of the effectiveness of alternative task decompositions.

Journals and Refereed Proceedings: (1982-)

Durfee, E.H., Corkill, D.D., and Lesser, V.R. (1984). "Distributing a Distributed Problem Solving Network Simulator", accepted for publication, 1984 IEEE Fifth Real-Time Systems Symposium, December.

Hudlicka, E. and Lesser, V.R. (1984). "Meta-Level Control Through Fault Detection and Diagnosis", accepted for publication, 1984 National Conference on Artificial Intelligence.

Carver, N.F., Lesser, V.R., and McCue, D.L. (1984). "Focusing in Plan Recognition", accepted for publication, 1984 National Conference on Artificial Intelligence.

Hudlicka, E. and Lesser, V.R., (1984). "Design of a Knowledge-Based Fault Detection and Diagnosis System", *International Conference on System Sciences*, Hawaii.

Lesser, V.R. and Corkill, D.D., (1983). "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks", *AI Magazine*, 4(3), (Fall), 15-33.

Corkill, D.D. and Lesser, V.R. (1983). "The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network", (long paper), *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, (August), 748-756.

Lesser, V.R., et al. (1983). "A High-Level Simulation Testbed for Cooperative Distributed Problem-Solving", *Third International Conference on Distributed Computer Systems*, (October), 341-349.

Corkill, D.D., Lesser, V.R., and Hudlicka, E. (1982). "Unifying Data-Directed and Goal-Directed Control: An Example and Experiments", in *Proceedings of the 1982 National Conference on Artificial Intelligence*, (August), 143-147.

Reports and Papers Presented: (1982-)

McCue, D.L. and Lesser, V.R. (1983). "Focusing and Constraint Management in Intelligent Interface Design", University of Massachusetts, Amherst, Department of Computer and Information Science Technical Report 83-36.

Pang, Yun-Jie, House, D.H., and Lesser, V.R. (1983). "A Procedure Based Chinese-Style Landscape Painting System, Computer Hua-Jia (Picture Master)", submitted to Transactions on Graphics.

Corkill, D.D. and Lesser, V.R. (1983). "Coordination in a Distributed Problem Solving Networks", Proceedings of the Conference on Artificial Intelligence, Oakland University, Rochester, Michigan, (April).

Hudlicka, E. and Lesser, V.R. (1983). "Knowledge-Based Approach to Fault-Tolerance", IEEE Workshop on Applications of Artificial Intelligence to Fault Diagnosis and Fault-Tolerance, San Juan, Puerto Rico, (January).

Abstract on Distributed Interpretation Testbed appeared in SIGART Newsletter, (April 1982), 16-17.

Editor of review on Artificial Intelligence/Brain Theory at University of Massachusetts, Winter issue of AI Magazine, (1982).

Bates, P.C., Wileden, J.C., and Lesser, V.R. (1983). "A Debugging Tool for Distributed Systems", Proceedings of the Second Annual Phoenix Conference on Computers and Communications, (March 1983), 311-315.

Lesser, V.R. (1983) "A Meta-Level Architecture for Distributed Problem Solving Systems", appeared as an abstract in Proceedings of the 1983 National Bureau of Standards Conference on Software Trends, Washington, D.C.

Croft, W.B., Lefkowitz, L.S., Lesser, V.R., and Huff, K.E. (1983). "POISE: An Intelligent Interface for Profession-Based Systems", Proceedings of the Conference on Artificial Intelligence, Oakland University, Rochester, Michigan, (April).

Huff, K.E. and Lesser, V.R. (1982). "Knowledge-Based Command Understanding: An Example for the Software Development Enviroment", Technical Report 82-6, Department of Computer and Information Science, Univeristy of Massachusetts, Amherst, Massachusetts, (July).

KRITHIVASAN RAMAMRITHAM

ADDRESS

Department of
Computer and Information Science
Graduate Research Center
University of Masachusetts
Amherst Massachusetts 01003
(413) 545-0196

21 Riverglade Apts.
East Hadley Road
Amherst Massachusetts 01002
(413) 253-7634

EDUCATION

Doctor of Philosophy — August 1981
University of Utah — Computer Science
Salt Lake City Utah

Master of Technology — July 1978
Indian Institute of Technology — Computer Science
Madras India

Bachelor of Technology — June 1976
Indian Institute of Technology — Electrical Engineering
Madras India

INTERESTS

General: Computer Science and Engineering.

Specific: Distributed Systems. Software Engineering.

EXPERIENCE

Sep 1981 - present
Assistant Professor. Department of Computer and Information Science, University of Massachusetts.

Sep 1979 - Aug 1981
Graduate Research Assistant, Department of Computer Science, University of Utah, Salt Lake City, Utah.

Ph.D. Thesis: Application of Temporal Logic to the synchronization of Concurrent Processes. Designed a high-level language to specify invariant and temporal properties affecting synchronization. Developed an algorithm for the automatic synthesis of synchronizers from such specifications. Devised a technique for the verification of synchronizers.

Sep 1978- May 1979
Graduate Teaching Assistant, Department of Computer Science,
University of Utah, Salt Lake City, Utah.

Assisted in teaching various undergraduate programming courses. Have experience with the Network Database Management System DBMS 20, on the DECsystem 20. Assisted in teaching a course on databases. Designed and implemented a database for preparing departmental class schedules.

June 1979 - Aug 1979
Programmer, VLSI project, Department of Computer Science,
University of Utah, Salt Lake City, Utah.

Enhanced a graphic editor to create and edit Storage Logic Array Descriptions using an Evans and Sutherland Picture System and a PDP 11/34.

Aug 1976 - Jul 1978
Graduate Student, Indian Institute of Technology,
Madras, India.

Designed and implemented software to augment the file handling capabilities of PL/1(F) compiler to create and maintain Virtual Storage Access Method files. The system served as an interface between compiled code and VSAM. This work was carried out on an IBM 370/155 operating under OS/VS1. Built an interpreter for an experimental block structured language. Designed and implemented a statistics generator for PL/1 programs.

PROFESSIONAL AFFILIATIONS

Member of the ACM and the IEEE Computer Society.

HONORS

University of Utah Graduate Research Fellow for 1980-81.

RESEARCH CONTRACTS AND GRANTS

1. "Resource Controllers in Distributed systems," Faculty Research Grant, University of Massachusetts, Nov 1981 to Oct 1982, Amount $2,500.

2. "Resource Controllers in Distributed systems," National Science Foundation, April 1982 to Oct 1984, Amount $61,383.

3. "Protection of Resources in Distributed Systems," Faculty Research Grant, University of Massachusetts, June 1982 to May 1983, Amount $2,200.

4. "Synthesis of Resource Controllers in Distributed systems," National Science Foundation, July 1984 to June 1985, Amount $35,043.

PUBLICATIONS

1. "Dynamic Task Scheduling in Hard Real-Time Distributed Systems," (with J. A. Stankovic) *IEEE Software*, July 1984 (also appeared in *4th International Conference on Distributed Computing Systems*, May 1984).

2. "Resource Controller Tasks in Ada: their Structure and Semantics" *Seventh International Conference on Software Engineering*, March 1984.

3. "Specification of Synchronizing Processes," (with R.M. Keller) *IEEE Transactions on Software Engineering*, Nov 1983.

4. "Correctness of A Distributed Transaction System," *Information Systems*, Vol 8, No 4, 1983.

5. "Automatic Generation of code for Resource Controller Tasks in Ada," (with P. Sunderrajan) *Symposium on the Application and Assessment of Automated Tools for Software Engineering*, San Fransisco, Nov 1983.

6. "Primitives for accessing protected objects," (with S. Vinter and D.Stemple) *Third Symposium on Reliability in Distributed Software and Database Systems*, Oct 1983.

7. "Operating System Support for Abstract Database Types," (with D.Stemple, S. Vinter and T. Sheard), chapter in ICOD-2, *Second International Conference on Databases*, S.M.Deen and P.Hammersley, editors, Wiley Heyden Ltd., August-September 1983, pp 179-195.

8. "Protecting Objects through the use of Ports," (with S. Vinter and D.Stemple) Proceedings of the *Second Phoenix Conference on Computers and Communication*, March 1983.

9. "Degrees of Fairness and their Realization," *Proc. Hawaii International Conference on System Sciences*, January 1983.

10. "On the Termination of Transactions in the Delta System," *Proc. 3rd International Conference on Distributed Computing Systems*, October 1982.

11. "Specification and Synthesis of Synchronizers," *Ph. D. Thesis*, The University of Utah, Aug, 1981.

12. "On Synchronization and its Specification," (with R.M. Keller) *Springer Lecture Notes in Computer Science 111*, Springer-Verlag, June 1981, 271-282.

13. "Specifying and Proving Properties of Sentinel Processes," (with R.M. Keller) *Proc. 5th International Conference on Software Engineering*, March, 1981.

14. "Specification and Synthesis of Synchronizers," (with R.M. Keller) *Proc. 1980 International Conference on Parallel Processing*, Aug, 1980, 311-321.

REPORTS

1. "Channel Access Delays in Local Area Hard Real-time Systems", submitted to *Fifth International Conference on Distributed Computing Systems*, Sep 1984.

2. "Dynamic Control of Module Interconnections", (with Stemple, D. and Vinter, S.) submitted to *IEEE Software*, Sep 1984.

3. "Scheduling Tasks with Resource Requirements in Hard Real-Time Systems", (with Zhao, W., and J. A. Stankovic) submitted to *IEEE Transactions on Software Engineering*, May 1984.

4. "Enabling Local Actions by Global Consensus", submitted to *Information Systems*, March 1984.

5. "Current Research and Critical Issues in Distributed Software Systems," (with J. A. Stankovic and W. H. Kohler) presented at the NSF sponsored Information Technology Workshop, also submitted to *Computing Surveys*, Jan 1984.

6. "Privilege transfer and revocation in a port-based system" (with D. Briggs, D.Stemple, and S.Vinter), submitted to *IEEE Transactions on Software Engineering*, Jan 1984.

7. "Interprocess Communication without process identifiers" (with D.Stemple, and S.Vinter), submitted to *IEEE Transactions on Software Engineering*, Jan 1984.

8. "Synthesizing code for Resource Controllers," submitted to *IEEE Transactions on Software Engineering*, March 1983.

9. "Proof Techniques for Resource Controller Processes," submitted to *Acta Informatica*, June 1982 (revised June 1984).

PROFESSIONAL SERVICE

Referee for
ACM Transactions on Programming Languages and Systems
IEEE Transactions on Computers
The National Science Foundation
International Symposium on Reliability in
Distributed Software and Database Systems
International Conference on Software Engineering
Real-time Symposium
International Conference on Parallel Processing
International Conference on Distributed Computing Systems

PROFESSIONAL CONFERENCE PARTICIPATION

4th International Conference on Distributed Computing
Systems, San Fransisco, California, May 1984, speaker.
Seventh International Conference on Software Engineering
Orlando, Florida, March 1984, speaker.
Third Symposium on Reliability in Distributed Software and
Database Systems, Clearwater Beach, Florida, Oct 1983, speaker.
IEEE Workshop on Real-time Operating systems,
Niagra Falls, New York, Aug 1983, speaker.
Sixteenth Hawaii International Conference on System Sciences,
Honolulu, Hawaii, Jan 1983, speaker.
Third International Conference on Distributed Computer Systems,
Miami, Florida, Oct 1982, speaker.
Conference on Functional Programming Languages and Computer
Architecture, Portsmouth, New Hampshire, Oct 1981, attendee.
CONPAR 81, Nurenberg, West Germany, speaker.
Fifth International Conference on Software Engineering
San Diego, California, March 1981, speaker.
1980 International Conference on Parallel Processing,
Boyne Highlands, Michigan, August 1980, speaker.
Symposium on the Principles of Programming Languages,
Las Vages, Nevada, Jan 1980, attendee.

TALKS AND COLLOQUIA

Indian Institute of Technology, Madras, India, June 1980.
Ohio State University, Columbus, Ohio, Jan 1981.
Wayne State University, Detroit, Michigan, Jan 1981.
Northwestern University, Evanston, Illinois, Feb 1981.
Rensselaer Polytechnic, Troy, New York, Feb 1981.
University of Waterloo, Ontario, Canada, Feb 1981.
University of Iowa, Iowa City, Iowa, Feb 1981.
University of British Columbia, Vancouver, Canada, Mar 1981.
University of Massachusetts, Amherst, Mass., Mar 1981.
Italian Committee for Nuclear Research, Rome, Italy, May, 1981.
GMD - Gesellschaft fur Mathematic und Datenverarbeitung,
MBH Bonn, West Germany, June 1981.
HMI - Hahn-Meitner Institute fur Kernforschung, Berlin, June 1981.
University of North Carolina, Chapel Hill, North Carolina, Jan 1982.
INRIA - Rocquencourt, Paris, France, June 1982.
IRISA - Rennes, France, Jan 1984.
INRIA - Rocquencourt, Paris, France, Jan 1984.
GTE Laboratories, Waltham, Mass., June 1984.

PERSONAL DATA

Born: 28 Feb 1955, Madras, India.
Marital Status: Married.
Social Security Number: 528-25-8928.
Permanent resident of the United States.

VITA

Personal

Name: Edward M. Riseman   Title: Professor   Birthdate: 8/15/42

Place of Birth: Washington, D.C.   Nationality: U.S.A   Sex: Male

Education

Ph.D.	Electrical Engineering	Cornell University	1969
M.S.	Electrical Engineering	Cornell University	1966
B.S.	Electrical Engineering	Clarkson College of Technology	1964

Professional Employment

March 1981 - Present	Chairman and Professor of Computer and Information Science, University of Massachusetts.
September 1978 - March 1981	Professor of Computer and Information Science, University of Massachusetts.
February 1978 - December 1978	on sabbatic leave; Institute Observer and Consultant at SRI, INT. (formerly Stanford Research Institute).
February 1973 - September 1978	Associate Professor of Computer and Information Science, University of Massachusetts.
September 1969 - February 1973	Assistant Professor of Computer and Information Science, University of Massachusetts.

Summary of Research and Teaching

Graduate courses taught include artificial intelligence, pattern recognition, cybernetics, computer vision, natural language processing, and AI applications in computer assisted instruction.

Major research areas are computer vision, including image segmentation, texture, parallel architectures, image interpretation, knowledge representation, inference and control structures. Application domains of our image analysis research include industrial automation, chip inspection, biomedical domains, remote sensing images, and natural outdoor scenes.

Previous research areas include techniques for learning, inference, and common-sense reasoning by computer; feature selection and the use of context in character recognition; educational and medical applications of artificial intelligence; limits of program efficiency via parallel processing; microcomputer communication systems for non-vocal severely handicapped people.

With Dr. Allen Hanson, and acting as Director of the Laboratory for Computer Vision Research, I have built a large research group. There is now a research community of approximately 20 people, including 2 visiting faculty, 1 postdoctoral researcher, and 8 doctoral candidates who are performing research in a variety of computer vision research efforts.

With Dr. Victor Lesser (Director of the COINS Research Computer Facility), I have led the development of the department computer facility over the last four years. It now includes a network of 11 VAX computers, two image processing systems, and a variety of color display devices, with a total value of over $2 million.

With Dr. Michael Arbib, I am in the process of building the foundation for a major robotics laboratory. We have acquired a GE cartesian coordinate arm with 4 degrees of freedom, and we have added a camera and a tactile sensor so that coordinated visual and tactile processing algorithms can be developed. We are in the process of seeking a second arm and additional research funding beyond the current NSF funding, so that cooperative algorithms for two armed manipulation can be conducted.

With Dr. Stanley Kulikowski II, I have been developing a research program in Microcomputers for the Severely Handicapped. The SpeakEasy Communication system has been developed as an intelligent communication aid for non-vocal almost paralyzed individuals such as those with cerebral palsy. We have received some funding from industry and the Massachusetts Rehabilitation Commission and we are attempting to build a program jointly supported by state, federal, and industrial sponsors.

Professional Societies

Member of IEEE, ACM, Pattern Recognition Society, American Association of Artificial Intelligence, Eta Kappu Nu, and Tau Beta Pi.

Ph.D. Students

Edward Fisher, "The Use of Context in Character Recognition," March 1976. Current Position: Digital Equipment Corporation.

Elliot Soloway, "'Learning = Interpretation + Generalization:' A Case Study in Knowledge-Directed Learning," September 1978. Current Position: Assistant Professor, Computer Science Department, Yale University.

Paul Nagin, "Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation," September 1979. Current Position: Assistant

Professor of Ophthalmology, Tufts New England Medical Center.

Thomas Williams, "Computer Interpretation of a Dynamic Image from a Moving Vehicle," May 1981. Current Position: Manager of Manufacturing Automation in Corporate Research, Digital Equipment Corporation.

Bryant York, "Shape Representation in Computer Vision," May 1981. Current Position: IBM Research Laboratories.

John Lowrance, "Dependency-Graph Models of Evidential Support," September 1982. Current Position: Research Scientist, SRI, International.

Ralf Kohler, "Integrating Non-Semantic Knowledge Into Image Segmentation Processes," September 1983.

Daryl Lawton, "Processing Dynamic Image Sequences from a Moving Sensor," February 1984.

Current Advanced Doctoral Students

Charles Kohl, Hierarchical Planning
Terry Weymouth, Image Interpretation

Major COINS Department Administrative Activities

January 1978 - present	Director of Laboratory for Computer Vision Research
September 1979 - June 1980	Chairman of COINS Personnel Committee
June 1974 - January 1978	Director of COINS laboratory research computer (PDP-15)
June 1974 - May 1976	Graduate Admissions Director

Professional Activities

Workshop on Computer Vision: Representation and Control, Rindge, New Hampshire, August, 1982 (a presentation)

Workshop on Multiresolution Image Processing and Analysis, Leesburg, Virginia, July, 1982 (a presentation)

International Joint Conference on Artificial Intelligence, Vancouver, August, 1981 (member of program committee)

5th International Conference on Pattern Recognition, Miami, December, 1980 (member of program committee; session chairman and member of panel session)

Workshop on Picture Data Description and Management, Asilomar, August, 1980 (member of program committee)

Nato Advanced Study Institute on Digital Image Processing, Bonas, France, June, 1980 (invited speaker)

NSF sponsored U.S.-Japan Seminar on Real-Time Parallel Image Processing, Tokyo, Japan, November, 1978 (one of six invited U.S. representatives; two presentations)

Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, November 1978 (member of panel session)

Electrotechnical Laboratory, Seminar on Scene Analysis, Tokyo, Japan, November 1978 (a presentation)

Engineering Foundation Conference on Algorithms for Image Processing, Pacific Grove, California, February 1978 (two presentations and chairman of session)

U.S.-Japan Seminar on Machine Vision, Boston, Massachusetts, August 1977 (a presentation)

IEEE Conference on Pattern Recognition and Image Processing, Rensselaer Polytechnic Institute, Troy, New York, June 1977 (session chairman)

Workshop on Computer Vision Systems, sponsored by the Office of Naval Research, held at the University of Massachusetts, Amherst, June 1-3, 1977 (co-organizer with A. Hanson)

IEEE Workshop on Picture Data Description and Management, Chicago, Illinois, April 1977 (member of panel session)

<u>Recent Research Grants and Contracts</u> (Principal or Co-Principal Investigator of each)

Air Force Office of Scientific Research, 4/83 - 9/85
 Representation and Control in the Interpretation of Complex Scenes
 $802,500
 (with A. Hanson)

Office of Naval Research, 7/84 - 7/86
 Dynamic Tactile Processing
 $70,000
 (with A. Hanson and M. Arbib)

Defense Advanced Research Projects Agency, 6/82 - 5/84
 Processing Dynamic Images from Camera Motion
 $420,000
 (with A. Hanson)

National Science Foundation, 1/82 - 1/85
 Visuo-Tactile Coordination for Robot Control
 $317,759
 (with M. Arbib)

Office of Naval Research, 1/75 - 3/84
 Semantically Directed Vision Processing
 $354,778

(with A. Hanson)

National Science Foundation, 9/79 - 8/82
A Computer System for Visual Interpretation of Natural Scenes
$373,800
(with A. Hanson)

Army Research Institute, 7/80 - 6/82
The Development and Evaluation of Instructional Strategies for an ICAI System
$156,776
(with E. Soloway)

Digital Equipment Corporation, 9/80 - 7/81
Computer Vision Applications for Industrial Automation
$109,173

Office of Naval Research, 1/75 - 3/80
Semantically Directed Vision Processing
$214,778
(with A. Hanson)

Equipment Grants for Department Research Computer Facility

(Principal or Co-Principal Investigator of each)

National Science Foundation, 6/82 - 9/83
Computer Science Research Equipment
$175,000
University Matching Funds: $35,000
(with V. Lesser)

National Science Foundation, 6/78 - 11/79
Computer Science Research Equipment
$28,470
(with V. Lesser)

Digital Equipment Corporation, 12/77
Grant towards COINS Department Computer Research Facility
$110,914
(with V. Lesser)

National Science Foundation, 6/77 - 11/78
Departmental Equipment Grant
$90,000
(with V. Lesser)

Recent Publications

Books and Book Chapters

"Region Relaxation in a Parallel Hierarchical Architecture," (with P.A. Nagin and A.R. Hanson) Real-Time Parallel Computing Image Analysis (M.

Onoe, K. Preston, and A. Rosenfeld, eds.), Plenum Press, 1981, pp. 37-61.

"Experiments in Schema-Driven Interpretation of a Natural Scene," (with C.C. Parma and A.R. Hanson) Digital Image Processing (J.C. Simon and R.M. Haralick, Eds.), D. Reidel Publishing Company, pp. 449-509, 1981; (also COINS Technical Report 80-10).

Journal Articles

"Reply to Image Segmentation: A Comment on 'Studies in Global and Local Histogram-Guided Relaxation Algorithms'," (with A.R. Hanson and P.A. Nagin) IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume PAMI-6, Number 2, March 1984, p. 249.

"Studies in Global and Local Histogram-Guided Relaxation Algorithms," (with P.A. Nagin and A.R. Hanson) IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume PAMI-4, Number 3, May 1982, pp. 263-277.

Conference Proceedings

"A Methodology for the Development of General Knowledge-Based Vision Systems," (with A. Hanson) Proc. of the IEEE Workshop on Principles of Knowledge-Based Systems (to appear), Denver, Colorado, December 1984.

"Extracting Linear Features," (with J. Brian Burns and A. Hanson) Proc. of the Seventh International Conference on Pattern Recognition, Montreal, Canada, July 30 - August 2, 1984.

"Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," (with G. Reynolds, N. Irwin and A. Hanson) Proc. of the 1984 Computer Vision Conference, Annapolis, MD, April 30 - May 2, 1984, pp. 238-247.

"Rule Based Strategies for Image Interpretation," (with T. Weymouth, J. Griffith and A. Hanson) Proc. of the National Conference on Artificial Intelligence, Washington, D.C., August 1983, pp. 429-432. A longer version of this paper appears in Proc. of the DARPA Image Understanding Workshop, Arlington, VA, June 1983, pp. 193-202.

"Image Understanding Research at the University of Massachusetts," (with A. Hanson) Proc. of the DARPA Image Understanding Workshop, Arlington, VA, June 1983, pp. 37-42.

Technical Reports

"Extracting Straight Lines," (with J. Brian Burns and A.R. Hanson) COINS Technical Report, University of Massachusetts, September 1984.

"A Summary of Image Understanding Research at the University of Massachusetts," (with A.R. Hanson) COINS Technical Report 83-35, University of Massachusetts, October 1983.

CURRICULUM VITAE

Michael Anthony Arbib

Place of Birth: Eastbourne, Sussex, England

Date of Birth: 28th May, 1940

Citizenship: United Kingdom (U.S. Immigrant Visa)

Education:

1. University of Sydney, N.S.W., Australia (February 1957 to December 1960). B.Sc., April 1961, with First Class Honours in Pure Mathematics; University Medal for Pure Mathematics; Prize for Best Honours Student in Faculty of Science.

2. Massachusetts Institute of Technology, Cambridge, U.S.A. (February 1961 to September 1963): Ph.D., September 1963, in Mathematics; Thesis Title: "Hitting and Martingale Characterization of One-Dimensional Diffusions" (Thesis Advisor: H. P. McKean, Jr.).

Record of Employment:

1. Computer Programmer, IBM (Australia) Pty., Ltd. Australian Summers 1958-59 and 1959-60.

2. Teaching Assistant, Research Assistant, or DSR Staff Member, M.I.T., February 1960 to September 1963, except for Summer of 1962.

3. Research Associate, Department of Engineering Mechanics, Stanford University, June 1965 to February 1966.

4. Assistant Professor, Department of Electrical Engineering, Stanford University, February 1966 to January 1969; Associate Professor, February 1969 to August 1970.

5. Professor of Computer and Information Science (Chairman, September 1970 to August 1975), and Adjunct Professor of Psychology, University of Massachusetts at Amherst, September 1970 to present. Director, Center for Systems Neuroscience, January 1974 to present.

Authored and Co-Authored Books Michael A. Arbib

1. "Brains, Machines, and Mathematics," McGraw-Hill: New York (1964), 152 + xiv pages. Also available in Polish, Russian, Italian, Japanese, and Spanish.

2. R.E. Kalman, P.L. Falb and M.A. Arbib: "Topics in Mathematical System Theory," McGraw-Hill: New York (1969), 358 + xiv pages. Also available in Russian and Rumanian.

3. "Theories of Abstract Automata," Prentice-Hall: Englewood Cliffs, N.J. (1969), 412 + xiii pages. Also available in Japanese.

4. "The Metaphorical Brain," Wiley-Interscience: New York (1972), 243 + xii pages. (Recipient of the American Society for Information Science Award for Best Information Sciences Book, 1973.) Also available in Russian, Polish and Japanese.

5. L. Padulo and M.A. Arbib: "System Theory: A Unified State-Space Approach to Continuous and Discrete Systems," W.B. Saunders: Philadelphia (1974), 779 + xvii pages.

6. L.S. Bobrow and M.A. Arbib: "Discrete Mathematics: Applied Algebra for Computer and Information Science," W.B. Saunders: Philadelphia (1974), 719 + xii pages.

7. J. Szentágothai and M.A. Arbib: "Conceptual Models of Neural Organization," (published as Neurosciences Research Program Bulletin Volume 12, No. 3, October 1974, 307-510). Republished in hardcover by the MIT Press: Cambridge, Mass. (1975). Also available in Russian.

8. M.A. Arbib and E.G. Manes: "Arrows, Structures, and Functors: The Categorical Imperative," Academic Press: New York (1975), 185 + xii pages.

9. "Computers and the Cybernetic Society," Academic Press: New York (1977), 494 + xxi pages. Also available in Dutch and Spanish.

10. S. Alagić and M.A. Arbib: "The Design of Well-Structured and Correct Programs," New York: Springer-Verlag (1978), 292 pages. Also available in Japanese, Polish and Russian.

11. M.A. Arbib, A.J. Kfoury and R.N. Moll: "A Basis for Theoretical Computer Science," New York: Springer-Verlag (1981), 220 + viii pages.

12. A.J. Kfoury, R.N. Moll and M.A. Arbib: "A Programming Approach to Computability," Springer-Verlag: New York, Heidelberg, Berlin (1982), viii + 251 pages.

13. Michael A. Arbib: "Computers and the Cybernetic Society," (Second Edition) Academic Press (1984), 491 + xix pages.

Edited Books

1. M.A. Arbib (Editor): "The Algebraic Theory of Machines, Languages, and Semigroups," Academic Press: New York (1968), 359 + xiv pages. Also available in German and Russian.

2. M.A. Arbib, D. Caplan and J.C. Marshall (Editors): "Neural Models of Language Processes," Academic Press: New York (1982), 571 + xvi pages.

3. S. Amari and M.A. Arbib (Editors): "Competition and Cooperation in Neural Nets," Lecture Notes in Biomathematics 45, Springer-Verlag: New York (1982), 441 + xiv pages.

4. O.G. Selfridge, E.L. Rissland and M.A. Arbib (Editors): "Adaptive Control of Ill-Defined Systems," Plenum Press (1984) x + 349 pages.

Membership of Editorial Boards

Cognition and Brain Theory
Cybernetics and Systems
International Journal of Man-Machine Studies
Journal of Computer and System Sciences
Journal of Mathematical Biology
Journal of Semantics
Journal of Social and Biological Structures
Biomathematics Series and Lecture Notes in Biomathematics, Springer-Verlag

90. (with E.G. Manes) "Foundations of System Theory: The Hankel Matrix," J. Comp. Syst. Sci. 20 (1980) 330-378.

91. (with E.G. Manes) "Machines in a Category," J. Pure and Applied Algebra 19 (1980) 9-20.

92. "Visuomotor Coordination: From Neural Nets to Schema Theory," Cognition and Brain Theory 4 (1981) 23-39.

93. (with A. Cornelis) "The Role of System Theory in the Social Sciences: An Interview", Journal of Social and Biological Structures 4 (1981) 375-386.

94. "Perceptual Structures and Distributed Motor Control", in Handbook of Physiology, Section 2: The Nervous System, Vol. II, Motor Control, Part 1 (V.B. Brooks, Ed.). American Physiological Society (1981), 1449-1480.

95. "Feature Detectors, Visuomotor Coordination and Efferent Control," Lecture Notes in Biomathematics 44, Recognition of Pattern and Form (D.G. Albrecht, Ed.), Springer-Verlag (1982) pp. 100-110.

96. "Cooperative Computation and the Cybernetic Society," in Progress in Cybernetics and Systems Research, Vol. VIII (R. Trappl, G. Klir and Pichler, Eds.). Hemisphere Publishing Corp.: Washington, D.C. (1982), pp. 3-12 (Vienna): Reprinted in Cybernetics: Theory and Applications (R. Trappl, Ed.) Hemisphere Publishing Corporation, (1983), pp. 361-372.

97. (with D. Caplan and J.C. Marshall) "Neurolinguistics in Historical Perspective," in Neural Models of Language Processes (M.A. Arbib, D. Caplan and J.C. Marshall, Eds.), Academic Press (1982), pp. 5-24.

98. "From Artificial Intelligence to Neurolinguistics," in Neural Models of Language Processes (M.A. Arbib, D. Caplan and J.C. Marshall, Eds.), Academic Press (1982), pp. 77-94.

99. "Perceptual-Motor Processes and the Neural Basis of Language," in Neural Models of Language Processes (M.A. Arbib, D. Caplan and J.C. Marshall, Eds.), Academic Press (1982), pp. 531-551.

100. (with E. Manes) "The Pattern-of-Calls Expansion is the Canonical Fixpoint for Recursive Definitions," J. ACM 29 (1982) 577-602.

101. (with R. Lara and A.S. Cromarty) "The Role of the Tectal Column in Facilitation of Amphibian Prey-Catching Behavior: A Neural Model," J. Neuroscience 2 (1982) 521-530.

102. (with P.M. Lavorel) "Towards a Theory of Language Performance: Neurolinguistics, Perceptual-Motor Processes, and Cooperative Computation," Theoretical Linguistics 8 (1981) 3-28.

103. (with K.J. Overton) "Systems Matching and Topographic Maps: The Branch-Arrow Model (BAM)," in Lecture Notes in Biomathematics 45: Competition and Cooperation in Neural Nets (S. Amari and M.A. Arbib, Eds.), Springer-Verlag (1982) 202-225.

104. "Modelling Neural Mechanisms of Visuomotor Coordination in Frog and Toad," in Lecture Notes in Biomathematics 45: Competition and Cooperation in Neural Nets (S. Amari and M.A. Arbib, Eds.), Springer-Verlag (1982) 342-370.

105. (with R. Lara and F. Cervantes) "Two-Dimensional Model of Retinal-Tectal-Pretectal Interactions for the Control of Prey-Predator Recognition and Size Preference in Amphibia," in Lecture Notes in Biomathematics 45, op. cit. (1982) 371-393.

106. (with R. Lara) "A Neural Model of the Interaction of Tectal Columns in Prey-Catching Behavior," Biological Cybernetics 44 (1982) 185-196.

107. (with R. Lara) "A Neural Model of Interaction between Pretectum and Tectum in Prey Selection," Cognition and Brain Theory 5 (1982) 149-171.

108. "Rana Computatrix: an evolving model of visuo-motor coordination in frog and toad," in Machine Intelligence 10 (J.E. Hayes, D. Michie, and and Y.-H. Pao, Eds.), Chichester: Ellis Horwood Publishers (1982), 501-517.

109. (with K.J. Overton) "The Extended Branch-Arrow Model of the Formation of Retino-Tectal Connections," Biological Cybernetics 45 (1982) 157-175.

110. (with I. Lieblich) "Multiple representations of space underlying behavior," The Behavioral and Brain Sciences, 5 (1982) 627-659. (Target Article, 627-640; Open Peer Commentary, 640-651; Authors' Response, 651-657.)

111. (with E.G. Manes) "Parametrized Data Types Do Not Need Highly Constrained Parameters," Information and Control 52 (1982) 139-158.

112. "Współpraca Informatyczna a Społeczeństwo Cybernetyczne," Projektowanie i Systemy, 4 (1983) 105-125 (Polish translation of "Cooperative Computation and the Cybernetic Society").

113. (with M. Steenstrup and E.G. Manes) "Port Automata and the Algebra of Concurrent Processes," Journal of Computer and System Sciences 27 (1983) 29-50.

114. "From Synergies and Embryos to Motor Schemas", in Human Motor Actions-Bernstein Revisited (H.T.A. Whiting, Ed.) Elsevier Science Publishers B.V. (North-Holland) 1984, 545-562.

115. (with J.M. Prager) "Computing the Optic Flow: The MATCH Algorithm and Prediction", Computer Vision, Graphics, and Image Processing, 24 (1983) 271-304.

116. "Cognitive Science: The View from Brain Theory", in The Study of Information: Interdisciplinary Messages (F. Machlup and U. Mansfield, Eds.) John Wiley and Sons (1983), pp. 81-91.

117. "Cybernetics: The View from Brain Theory", in The Study of Information: Interdisciplinary Messages (F. Machlup and U. Mansfield, Eds.) John Wiley and Sons (1983), pp. 459-465.

118. (with O.G. Selfridge and E.L. Rissland) "A Dialogue on Ill-Defined Control," in Adaptive Control of Ill-Defined Systems (O.G. Selfridge, E.L. Rissland and M.A. Arbib, Eds.), Plenum Press (1984) pp. 1-9.

119. (with K.J. Overton & D.T. Lawton)"Perceptual Systems for Robots," Interisciplinary Science Reviews, 9, No. 1, (1984) 31-46.

VITA

Andrew G. Barto

Biographical Data

Date of Birth: July 3, 1948

Home Address:
124 Chestnut Street
Amherst, MA 01002
(413-549-0616)

Business Address:
Computer and Information Science Department
University of Massachusetts
Amherst, MA 01003
(413-545-2109)

Education

University of Michigan	1970	B.S.	Mathematics
University of Michigan	1972	M.S.	Computer and Communication Sciences
University of Michigan	1975	Ph.D.	Computer and Communication Sciences

Employment

Computer and Communication Sciences University of Michigan	Teaching Assistant	1971-74
Logic of Computers Group University of Michigan	Research Assistant	1971-75
School of Advanced Technology SUNY, Binghamton, NY	Assistant Professor	1975-79
Computer and Information Science University of Massachusetts	Post-doctoral Research Associate	1977-79
	Adjunct Assistant Professor	1979-82
	Associate Professor	1982-present

Professional Memberships

American Association for the Advancement of Science
Society for Neuroscience
Institute of Electrical and Electronics Engineers
American Association for Artificial Intelligence

Academic Awards and Honors

B.S. with Distinction, 1970
William J. Branstrom Freshman Prize, University of Michigan, 1967

Teaching Experience

Univ. of Michigan 1971-74, Teaching Assistant, Teaching Associate
Courses: Introductory Computer Programming, Foundations of Computer and Communication Science, Theory of Natural Language Structure (laboratory), Digital and Analog Systems (laboratory)

SUNY, Binghamton 1975-77, Assistant Professor
Courses: Logic Design and Computer Architecture, Discrete Simulation Techniques, Digital Signal Processing

Univ. of Mass. 1977-present, Post-doctoral Research Associate, Adjunct Assistant Professor, Associate Professor
Courses: Neural Modelling I and II, Assembly Language Programming, Data Structures. Seminars: Memory and Natural Intelligence, Cybernetics of Learning and Memory

Teaching Interests

Neural Modelling, Learning and Memory, Modelling and Simulation

Research Interests

Learning and memory in natural and artificial systems; highly distributed, parallel, and associative knowledge representation; cooperativity in distributed systems; models of sensorimotor learning; models of synaptic plasticity; computational tools for modelling and simulation. Current research involves synthesis of behavioral, neurobiological, and cybernetic studies of learning and memory.

Refereed Publications

A. Barto, ''A Neural Network Simulation Method Using the Fast Fourier Transform," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-5, No. 12, December 1976, pp. 863-867.

A. Barto, "Discrete and Continuous Models," International Journal of General Systems, Vol. 4, 1978, pp. 163-177.

A. Barto, "A Note on Pattern Reproduction in Tesselation Structures," Journal of Computer and Systems Sciences, Vol. 16, No. 3, June 1978, pp. 445-455.

A. Barto, "Invariant Linear Models of Varying Linear Systems," In NATO Conference Series, Series II, Systems Science, Vol. 5, G. Klir (Ed.), Plenum, New York, 1978.

R. Sutton and A. Barto, "Toward a Modern Theory of Adaptive Networks: Expectation and Prediction," Psych. Review, Vol. 88, No. 2, 1981, pp. 135-170.

A. Barto and R. Sutton, "Simulation of Anticipatory Responses in Classical Conditioning by a Neuron-like Adaptive Element," Behavioural Brain Research 4, 1982, pp. 221-235.

A. Barto, R. Sutton and P. Brouwer, "Associative Search Network: A Reinforcement Learning Associative Memory," Biological Cybernetics, Vol. 40, 1981, pp. 201-211.

A. Barto and R. Sutton, "Landmark Learning: An Illustration of Associative Search," Biological Cybernetics, Vol. 42, 1981, pp. 1-8.

R. Sutton and A. Barto, "An Adaptive Network that Contructs and Uses an Internal Model of its Environment," Cognition and Brain Theory, 4, 1981, pp. 217-246.

A. Barto, C. Anderson and R. Sutton, "Synthesis of Nonlinear Control Surfaces by a Layered Associative Network," Biological Cybernetics, Vol. 43, 1982, 175-185.

A. Barto, R. Sutton and C. Anderson, "Spatial Learning Simulation Systems," Proceedings of the 10th IMACS World Congress on Systems Simulation and Scientific Computation, 1982, pp. 204-206.

A. Barto, R. Sutton and C. Anderson, "Neuron-like Adaptive Elements that can Solve Difficult Learning Control Problems," IEEE Trans. on Systems, Man, and Cybernetics, 13, 1983, 834-846.

Unrefereed Publications

A. Barto, "Simulation of Networks using Multidimensional Fast Fourier Transforms," ACM Simuletter, 5, 4, July 1974.

A. Barto, "Cellular Automata as Models of Natural Systems," Ph.D. Thesis, Logic of Computers Group Technical Report, University of Michigan, 1975.

B.P. Ziegler and A. Barto, "Alternative Formalisms for Biosystem and Ecosystem Modelling," in New Directions in the Analysis of Ecological Systems, Part 2, G. Innis (Ed.), Simulation Councils Proceedings Series, Vol. 5, No. 2, 1977, pp. 167-178.

A. Barto and R. Sutton, "Goal Seeking Components for Adaptive Intelligence: An Initial Assessment," Final Technical Report AFWAL-TR-81-1070, Avionics Laboratory (AFWAL/AAAT), Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio 45433 (530 pages).

A. Barto and R. Sutton, "Neural Problem Solving." In: W. B. Levi and J. A. Anderson (eds.), Synaptic Modification, Neuron Selectivity, and Nervous System Organization, Erlbaum, in press.

A. Barto and S. Epstein, "Adaptive Network Representations of Control Surfaces for Motor Systems." Proceedings of the Second Workshop on Visuomotor Coordination in Frog and Toad: Theory and Experiment, November 17-19, 1982, Mexico City, Mexico, COINS Technical Report 83-19.

A. Barto, Review of S. Grossberg's "Studies of Mind and Brain," D. Reidel Publishing Company, 1982. To appear in Mathematical Biosciences.

A. Barto (Editor), "Simulation Experiments with Goal-Seeking Adaptive Elements." Final Technical Report AFWAL-TR-84-1022, Avionics Laboratory (AFWAL/AAAT), Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio 45433 (163 pages).

Papers in Progress

A. Barto, P. Anandan, "Pattern Recognizing Stochastic Learning Automata." Submitted to IEEE Trans. on Systems, Man, and Cybernetics.

A. Barto, C. Anderson, "On the Collective Behavior of Pattern-Recognizing Stochastic Learning Automata."

Conferences and Presentations

Colloquium speaker, The Aerospace Corporation, El Segundo, CA, October, 1982.

Neural Modelling Conference, Arizona State University, April, 1983; paper presented.

Colloquium speaker, Center for Neural Science, Brown University, May, 1983.

Workshop on Vision, Brain, and Cooperative Computation, Amherst, MA, May 9-11, 1983; Discussion leader and speaker.

Review of Air Force Sponsored Basic Research in Biomedical Sciences, July, 1983, Irvine, CA. Presentation with abstract in proceedings (AFOSR, Bolling AFB, DC 20332).

Colloquium speaker, Institute for Cognitive Science, Univ. of California, San Diego, July, 1983.

Workshop on Parallel Distributed Processing, Institute for Cognitive Science, Univ. of California, San Diego, June 24-26, 1984.

Grants and Contracts

Co-Principal Investigator (Principal Investigator - D. N. Spinelli), "Decision-Making by Adaptive Networks of Goal-Seeking Components," 3 years, AFOSR and the Avionics Laboratory (Air Force Wright Aeronautics Laboratory), ended June, 1983.

Principal Investigator, "Adaptive Problem Solving with Networks of Goal-Seeking Components," AFOSR and the Avionics Laboratory (Air Force Wright Aeronautics Laboratory), September 1, 1983 to August, 1986.

Co-Principal Investigator (Principal Investigator - J. W. Moore, Psychology), "Adaptive Element Models of Classical Conditioning", NSF Psychobiology, August, 1983.

Daniel D. Corkill

Department of Computer and
Information Science
University of Massachusetts
Amherst, Massachusetts 01003
(413) 545–0156

Home:
Stage Road
South Deerfield
Massachusetts 01373
(413) 665–4544

Degrees:

Ph.D. January 1983, University of Massachusetts

M.S. 1976, B.S. 1975, University of Nebraska

Research Interests:

Coordination in distributed problem solving networks.

Planning and control in large AI systems.

Design and programming methodologies for contructing large AI systems.

Hardware and software support for AI systems.

Experience:

Research Computer Scientist, Department of Computer and Information Science, University of Massachusetts, since January 1983.

> Performed basic research in cooperative distributed problem solving and meta-level control.

Research assistant to Victor R. Lesser, Cooperative Distributed Problem Solving, Department of Computer and Information Science, University of Massachusetts, November 1978 to December 1982.

> Formulated methodologies for cooperative distributed problem solving, planning, and organizational self-design. Performed major design, implementation, and experimentation on a distributed interpretation testbed system based on an extended Hearsay-II architecture. Performed major design, implementation, and documentation work on a production-level Lisp system (CLisp) for the VAX 11/780.

Instructor, Computer Science Certificate Program, University of Massachusetts, January 1982 to May 1982.

Seminar instructor, Atomic Energy of Canada Limited, Chalk River Nuclear Laboratories, Chalk River, Ontario, Canada, September 11–13, 1979.

Research assistant to Edward M. Riseman and Allen R. Hanson, VISIONS: A Computer System for the Segmentation and Interpretation of Visual Scenes, Department of Computer and Information Science, University of Massachusetts, November 1977 to October 1978.

Research assistant to James H. Burrill, Development and Support of APL 2.0 (Control Data Corporation), University Computing Center, University of Massachusetts, January to September 1977.

Teaching assistant, Department of Computer Science, University of Nebraska, June 1975 to December 1976.

User consultant (part time), University of Nebraska Computer Network, University of Nebraska, September 1974 to May 1975.

Programmer (full time, staff position), University of Nebraska Computer Network, University of Nebraska, October 1973 to May 1974.

Honors:

Upsilon Pi Epsilon (computer science honors society), 1974.

Ph.D. dissertation selected as the Computer and Information Science Department's nominee to the 1983 ACM Doctoral Dissertation Award Program.

Personal Information:

Born 15 June 1952, Lincoln, Nebraska.

Social Security Number: 476-52-9224.

Married.

Publications and Reports:

Edmund H. Durfee, Daniel D. Corkill, and Victor R. Lesser.
Distributing a distributed problem solving network simulator.
To appear in the *Proceedings of the Fifth Real-Time Systems Symposium*, December 1984.

Jasmina Pavlin and Daniel D. Corkill.
Selective abstraction of AI system activity.
Proceedings of The Fourth National Conference on Artificial Intelligence, pages 264–268, August 1984.

Victor R. Lesser and Daniel D. Corkill.
The Distributed Vehicle Monitoring Testbed: A tool for investigating distributed problem solving networks.
AI Magazine (4)3:15–33, Fall 1983.

Daniel D. Corkill and Victor R. Lesser.
The use of meta-level control for coordination in a distributed problem solving network.
Proceedings of the Eighth International Joint Conference on Artificial Intelligence, pages 748–756, August 1983.

Daniel D. Corkill and Victor R. Lesser.
Coordination in a distributed problem solving network.
Proceedings of the Rochester Conference on Artificial Intelligence, pages 637–648, April 1983.

Daniel David Corkill.
A Framework for Organizational Self-Design In Distributed Problem Solving Networks.
Ph.D. thesis, University of Massachusetts, February 1983.

Daniel D. Corkill, Victor R. Lesser, and Eva Hudlicka.
Unifying data-directed and goal-directed control: An example and experiments.
Proceedings of the National Conference on Artificial Intelligence, pages 143–147, August 1982.

Janice E. Cuny

COINS Department, University of Massachusetts

DATE OF BIRTH: September 6, 1951

EDUCATION:

A.B.	Independent Study/Computer Science	Princeton Univ.	1973
M.S.	Computer Science	Univ. of Wisconsin	1974
Ph.D.	Department of Computer and Communication Sciences	Univ. of Michigan	1981

HONORS AND HONOR SOCIETIES:

magna cum laude, Princeton University

Member of Phi Beta Kappa, Eta Kappa Nu, Sigma Xi (Associate member)

IBM Graduate Fellowship, 1979-80

IBM Faculty Development Award, 1984

PROFESSIONAL EXPERIENCE:

Assistant Professor, Department of Computer and Information Science, University of Massachusetts, 1984-.

Assistant Professor, Department of Computer Sciences, Purdue University, 1981-1983.

Teaching Assistant, Department of Computer and Communication Sciences, The University of Michigan, 1974-79, University of Wisconsin, 1973-74.

Research Assistant, Department of Computer and Communication Sciences, The University of Michigan, 1977-78.

Instructor, Washtenaw Community College, 1977-78.

PAPERS/TECHNICAL REPORTS:

A model theoretical approach to software correctness, photosynthesis, neural networks and contextual structure, with Mary Beth Zak, Mark Weiser and Gary Strong. Logic of Computers Group Technical Report 189, The University of Michigan, 1975.

Synchronization; a formal approach. Ph.D. Thesis, The University of Michigan, 1981.

Testing the coordination predicate, with Lawrence Snyder. Technical Report 391, Purdue University, 1982. To appear *IEEE Transactions on Computers*, March 1984.

Conversion from data-flow to synchronous execution in loop programs, with Lawrence Snyder. Technical Report 392, Purdue University, 1982.

"Testing coordination in 'homogeneous' parallel algorithms," with Lawrence Snyder. Proceedings of The 1982 International Conference on Parallel Processing, pp. 265-267.

"A model for analyzing generalized interprocessor communication systems," with Lawrence Snyder. In *Algorithmically-Specialized Computers*, L. Snyder, H. Seigel, L. Seigel and D. Gannon (eds.), to appear. Also available as Technical Report 406, Purdue University, 1982.

"Compilation of data-driven programs for synchronous execution," with Lawrence Snyder. Proceedings of the 1983 Annual Symposium on Principles of Programming Languages, January 1983, pp. 197-202.

"The Pringle: An experimental system for parallel algorithm testing and software development," with Alejandro Kapauan, Ko-Yang Wang, Dennis Gannon, and Lawrence Snyder. Proceedings of The 1984 International Conference on Parallel Processing, pp. 1-6.

VITA

Personal

Name: Allen R. Hanson
Birthdate: August 4, 1942
Place of Birth: Jamaica, New York
Nationality: USA
Marital Status: Married
Children: Two

Title: Associate Professor
Computer and Information
Science Department
University of Massachusetts
Amherst, Massachusetts
01003

Education

Cornell University, Ph.D., Electrical Engineering, 1969

Cornell University, M.S., Electrical Engineering, 1966

Clarkson College of Technology, B.S., Electrical Engineering, 1964

Professional Experience

1981 - Present	Associate Professor, Computer and Information Science, University of Massachusetts
1978 - 1980	Associate Professor, School of Language and Communication, Hampshire College
1978 - 1979	On Leave from Hampshire to Computer and Information Science Department, University of Massachusetts
1975 - 1976	Dean, School of Language and Communication, Hampshire College
1974 - 1975	Associate Dean, School of Language and Communication, Hampshire College
1972 - 1978	Assistant Professor, School of Language and Communication, Hampshire College
1969 - 1972	Assistant Professor, Computer, Information, and Control Sciences, University of Minnesota

Honors

Eta Kappa Nu (Electrical Engineering Honorary Society)
Tau Beta Pi (Engineering Honorary Society)
B.S. with Honors (Clarkson College of Technology)
Atomic Energy Commission Fellowship, 1964 - 1967
New York State Regents Scholarship, 1962 - 1964

Teaching Experience

Undergraduate Courses

Introductory Programming (FORTRAN, Pascal, ALP, LISP, SNOBOL, ALGOL, assembly language)
Data Structures
Structure of Programming Languages
Real Time Programming
Computer Architecture
Computer Graphics
Systems Programming
Pattern Recognition
Computers and Society
Game Theory
Artificial Intelligence
Philosophy of Mind
Cognitive Psychology
Image Processing
Computer Vision

Graduate Courses and Seminars

Image Interpretation
Image Processing
Pattern Recognition
Interactive Raster Graphics
Artificial Intelligence

Research Interests

Current research areas are computer vision/image processing and graphics. Topics of interest in computer vision include manipulation of complex images (segmentation, color and texture, parallel architectures for image processing), knowledge representation and control structures for knowledge application to complex image processing tasks, the qualitative/ quantitative study of sources of image specific information (e.g., surface properties, perspective transformations, object shape, motion, object occlusion) and mechanisms for their extraction from images, and applications of image processing techniques to remotely sensed data (including biomedical images). Topics in graphics are related to the work in machine vision and include parametric surface representations and their use in 3D shape representations, fractals, high speed graphics hardware, distributed graphics systems, and real time display of time varying data.

Previous research interests have included pattern recognition, feature selection, and the use of contextual information in pattern recognition; design of a hardware/software distributed microprocessor system with raster graphics capabilities; use of computer assisted instruction in medical education; computer science education; and applications of pattern recognition techniques to biomedical data (particularly diagnostic and analytic methods in conjunction with Mayo Clinic and the Chronobiology Laboratory at University of Minnesota).

Funding Record

Active Grants and Contracts

(Principal or Co-Principal Investigator on each)

Air Force Office of Scientific Research, 4/83 - 9/85
Representation and Control in the Interpretation of Complex Scenes
(with E. Riseman)

Defense Advanced Research Projects Agency, 6/82 - 5/84
Processing Dynamic Images from Camera Motion
(with E. Riseman)

Office of Naval Research, 7/84 - 7/86
Dynamic Tactile Processing
(with E. Riseman and M. Arbib)

Air Force Office of Scientific Research, 7/82 - 7/83
Equipment in Support of Computer Vision and Adaptive Network Research
(with E. Riseman)

Rome Air Development Center via Syracuse University, 8/83 - 9/84
Applying the VISIONS System to Interpretation of Aerial Images
(with E. Riseman)

A.C. Nielsen, 3/83 - 11/84
PACMS
(with E. Riseman)

Previous Support

Office of Naval Research, 1/74 - 3/84
Semantically Directed Vision Processing

Office of Naval Research, January 1976
Developing and Coordinating a Workshop on Computer Vision Systems which was held June 1977

Office of Naval Research, 1/71 - 1/74
Automated Feature Selection and Use of Contextual Analysis in Pattern Recognition

UMass Remote Sensing, 2/82 - 2/83
Development of Initial Design and Implementation Specifications for the University of Massachusetts Segment of Remote Sensing Research and Development Programs
(with G. Reynolds)

National Science Foundation, 9/79 - 8/82
A Computer System for Visual Interpretation of Natural Scenes

National Science Foundation, 9/77 - 9/78
A Computer Graphics/TV Facility

Selected Publications

Journal Articles and Refereed Conference Proceedings

E. Riseman and A. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems," to appear in *Proc. of the IEEE Workshop on Principles of Knowledge-Based Systems*, Denver, Colorado, December 1984.

J. Brian Burns, A. Hanson and E. Riseman, "Extracting Linear Features," *Proc. of the Seventh International Conference on Pattern Recognition*, Montreal, Canada, July 30 - August 2, 1984.

G. Reynolds, N. Irwin, A. Hanson and E. Riseman, "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," *1984 Workshop on Computer Vision Representation and Control*, Annapolis, MD, April 30 - May 2, 1984, pp. 238-247.

A. Hanson, E. Riseman and P. Nagin, "Reply to Image Segmentation: A Comment on 'Studies in Global and Local Histogram-Guided Relaxation Algorithm'," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume PAMI-6, Number 2, March 1984, p. 249.

T. Weymouth, J. Griffith, A. Hanson and E. Riseman, "Rule Based Strategies for Image Interpretation," *Proc. of the National Conference on Artificial Intelligence*, Washington, D.C., August 1983, pp. 429-432. A longer version of this paper appears in *Proc. of the DARPA Image Understanding Workshop*, Arlington, VA, June 1983, pp. 193-202.

E. Riseman and A. Hanson, "Image Understanding Research at the University of Massachusetts," *Proc. of the DARPA Image Understanding Workshop*, Arlington, VA, June 1983, pp. 37-42.

R.R. Kohler and A.R. Hanson, "The VISIONS Image Operating System," *Proc. of 6th International Conference on Pattern Recognition*, Munich, Germany, October 1982.

L. Wesley and A. Hanson, "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning About Images in the VISIONS System," *Proc. of the Workshop on Computer Vision: Representation and Control*, Rindge, New Hampshire, August 1982.

P.A. Nagin, A.R. Hanson and E.M. Riseman, "Studies in Global and Local Histogram-Guided Relaxation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume PAMI-4, Number 3, May 1982, pp. 263-277.

Technical Reports

J. Brian Burns, A.R. Hanson and E.M. Riseman, "Extracting Straight Lines," COINS Technical Report, University of Massachusetts, September 1984.

E.M. Riseman and A.R. Hanson, "A Summary of Image Understanding Research at the University of Massachusetts," COINS Technical Report 83-35, University of Massachusetts, October 1983.

Professional Activities

Recent Invited Talks and Panel Discussions

1983	"State of the Art in Computer Graphics," Southeastern University, May 1983
1983	"Architectures for Image Processing and Machine Vision," Workshop on Vision, Brain, and Cooperative Computation, Amherst, Massachusetts, May 9-11, 1983
1982	Workshop on Computer Vision: Representation and Control, Rindge, New Hampshire
1982	Sixth International Conference on Pattern Recognition, Munich, Germany (conference committee)
1981	International Joint Conference on Artificial Intelligence, Vancouver, Canada
1979	Pattern Recognition and Image Processing Conference, Chicago, Illinois
1979	Fourth Interdisciplinary Conference, Teton Village, Wyoming

Recent Memberships on Editorial Boards and Conference Organizing Committees

Editorial Board, Computer Graphics and Image Processing.

Program Committee, Fifth National Conference of the Canadian Society for Computational Studies of Intelligence, May 1974, University of Western Ontario, London, Canada.

Program Committee, 1983 Computer Vision and Pattern Recognition Conference, Arlington, VA, June 19-23, 1983.

Program Committee, Workshop on Computer Vision, Annapolis, MD. To be held April 3Ø - May 2, 1984.

Co-Organizer (with Michael Arbib) of the Workshop on Vision, Brain, and Cooperative Computation, University of Massachusetts, Amherst, MA, May 9-11, 1983.

Co-Organizer (with Edward Riseman) of the Workshop on Computer Vision Systems, sponsored by ONR, held at the University of Massachusetts, Amherst, June 1977.

Co-Chairman and Founder (with Jay Leavitt), First International Conference on Computers and the Humanities, University of Minnesota, July 1973.

Reviewer

National Science Foundation

Office of Naval Research
Computing Reviews
Prentice-Hall Publishing Co.
ACM
IEEE
Computer Graphics and Image Processing
Academic Press
West Publishing Co.

Member: IEEE, ACM

Recent Conferences Attended:

Workshop on Computer Vision: Representation and Control, Annapolis, MD, April 30-May 2, 1984.

International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, August 8-12, 1983.

SIGGRAPH 83, Detroit, MI, July 25-29, 1983.

Computer Vision and Pattern Recognition Conference, Arlington, VA, June 1983.

DARPA Image Understanding Workshop, Arlington, VA, June 1983.

Other Activities

Department Service
   COINS Curriculum Committee (Chairman) 1983 - Present
   COINS Personnel Committee, 1981 - 1982
   COINS-ECE Cooperation Committee, 1981 - 1982
   COINS Research Computing Facility Committee, 1981 - Present

Assistant Director, Laboratory for Computer Vision

Member, Sloan Foundation Working Group on Cognitive Science

Consulting, Intelligent Software Systems, Amherst, MA.
   Responsibilities include overseeing the technical development and implementation of a high-performance color graphics system for real-time display of time-varying data.

CURRICULUM VITAE

Leslie John Kitchen

Born

9th April 1954, Sydney, Australia.

Education

Ph.D. in Computer Science, University of Maryland, December 1982.

B.Sc., First Class Honours in Computer Science, University of Sydney, April 1976.

Scholarships and Awards

Samuel Alexander Award (Washington D.C. Chapter of Association for Computing Machinery), 1982.

Australian Commonwealth University Scholarship, 1972.

Research Experience

October 1983 to present: Senior Post-Doctoral Research Associate, Computer and Information Science Department, University of Massachusetts, Amherst.

January 1983 to October 1983: Research Associate, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park.

January 1983 to October 1983: Consultant for Nuclear Engineering and Sciences Group (Department of Energy Information System Project), University Research Foundation, Greenbelt, Maryland.

January to December 1982: Faculty Research Assistant, Computer Vision Laboratory, Computer Science Center, University of Maryland.

January 1978 to January 1982: Graduate Research Assistant, Computer Vision Laboratory, Computer Science Center, University of Maryland.

Teaching Experience

August 1979 to October 1983: Lecturer for the following Computer Science courses: CMSC 430 (Theory of Compiling), CMSC 220 (File Processing), and CMSC 120 (Intermediate Programming), University College, University of Maryland.

March to May 1981: Half-semester lecturer for IFSM 498B (Programming Languages for Commercial Applications), Dept. Information Systems Management, University of Maryland.

August to December 1977: Teaching Assistant for CMSC 103 (Introduction to Programming for Non-Majors), Dept. of Computer Science, University of Maryland.

February to December 1975: Teaching Assistant for 3rd-year course in Numerical Analysis, Grader for 2nd-year course in Assembly Language, Basser Dept. of Computer Science, University of Sydney.

Other Experience

June 1976 to July 1977: Applications Programmer at University Computing Centre, University of Sydney.

December 1974 to February 1975, December 1975 to June 1976: Programmer for Computing and Statistical Operations Pty Ltd, Sydney (Actuarial Consultants).

Refereed Publications

L. J. Kitchen and A. Rosenfeld, "Scene analysis using region-based constraint filtering", *Pattern Recognition*, vol. 17, no. 2, 1984, pp. 189–203.

S. Kasif, L. J. Kitchen and A. Rosenfeld, "A Hough transform technique for subgraph isomorphism", *Pattern Recognition Letters*, vol. 2, 1983, pp. 83–88.

L. J. Kitchen, M. Pietikaïnen, A. Rosenfeld and Cheng-Ye Wang, "Multispectral image smoothing by use of global distribution of pixel values", *IEEE Trans. Systems, Man, & Cybernetics*, vol. SMC-13, no. 4, July/August 1983, pp. 626–631.

L. J. Kitchen and A. Rosenfeld, "Gray-level corner detection", *Pattern Recognition Letters,* vol. 1, no. 2, December 1982, pp. 95–102.

L. J. Kitchen and A. Rosenfeld, "Non-maximum suppression of gradient magnitudes makes them easier to threshold", *Pattern Recognition Letters,* vol. 1, no. 2, December 1982, pp. 93–94.

L. J. Kitchen and E. V. Krishnamurthy, "Fast, parallel, relaxation screening for chemical patent data-base search", *Journal of Chemical Information and Computer Sciences,* vol. 22, 1982, pp. 44–48.

R. L. Hartley, L. J. Kitchen, Chen-Ye Wang and A. Rosenfeld, "Segmentation of FLIR images: A comparative study", *IEEE Trans. Systems, Man, & Cybernetics,* vol. SMC-12, no. 4, July/August 1982, pp. 553–566.

L. J. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence", *IEEE Trans. Systems, Man, & Cybernetics,* vol. SMC-11, no. 9, September 1981, pp. 597–605.

L. J. Kitchen, "Relaxation applied to matching quantitative relational structures", *IEEE Trans. Systems, Man, & Cybernetics,* vol. SMC-10, no. 2, February 1980, pp. 96–101.

L. J. Kitchen and A. Rosenfeld, "Discrete relaxation for matching relational structures", *IEEE Trans. Systems, Man, & Cybernetics,* vol. SMC-9, no. 12, December 1979, pp. 869–874.

Reports and Other Publications

L. S. Davis, L. J. Kitchen, F.-P. Hu and V. Hwang, "Image matching using generalized Hough transforms", Technical Report 27, Center for Automation Research, University of Maryland, October 1983.

S. Kasif, L. J. Kitchen and A. Rosenfeld, "A Hough transform technique for subgraph isomorphism", Computer Science Technical Report 1248, University of Maryland, March 1983.

L. J. Kitchen, "Local consistency and constraint filtering for structure matching and image analysis", Ph.D. thesis, Computer Science Department, University of Maryland, December 1982.

Cheng-Ye Wang and L. J. Kitchen, "Improvements in multispectral smoothing", Computer Science Technical Report 1152, University of Maryland, March 1982.

L. J. Kitchen, "Scene analysis using region-based constraint filtering", Computer Science Technical Report 1150, University of Maryland, Feb. 1982.

R. L. Hartley, L. J. Kitchen, Cheng-Ye Wang and A. Rosenfeld, "A comparative study of segmentation algorithms for FLIR images", Computer Science Technical Report 1104, University of Maryland, September 1981.

L. J. Kitchen and E. V. Krishnamurthy, "Fast, parallel, relaxation screening for chemical patent data-base search", Computer Science Technical Report 1095, University of Maryland, July 1981.

L. J. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence", Computer Science Technical Report 981, University of Maryland, December 1980.

L. J. Kitchen and A. Rosenfeld, "Gray-level corner detection", Computer Science Technical Report 887, University of Maryland, April 1980.

L. J. Kitchen and A. Rosenfeld, "Non-maximum suppression of gradient magnitudes makes them easier to threshold", in Computer Science Technical Report 885, University of Maryland, March 1980.

R. L. Kirby, R. C. Smith, P. A. Dondes, S. Ranade, L. J. Kitchen and F. Blonder, "Interfaces, subroutines, and programs for the Grinnell GMR-27 Display Processor on a PDP-11/45 with the UNIX Operating System", Computer Science Technical Report 810, University of Maryland, October 1979.

L. J. Kitchen, "Relaxation applied to matching quantitative relational structures", Computer Science Technical Report 707, University of Maryland, October 1978.

L. J. Kitchen, "Discrete relaxation for matching relational structures", Computer Science

Technical Report 665, University of Maryland, June 1978.

L. J. Kitchen, "Analysis of cartoons into ellipses", Honours Project Report, Basser Dept. of Computer Science, University of Sydney, December 1975.

James F. Kurose
Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

Education

Doctor of Philosophy, Computer Science, Columbia University, May 1984
Thesis Area: Distributed Computation and Communication

MS, Computer Science, Columbia University, May 1980

IBM Predoctoral Fellowship, Doctoral dissertation nominated for ACM Doctoral Dissertation Award, Eta Kappa Nu

Bachelor of Arts, Physics, Wesleyan University, May 1978

Magna cum laude with high honors in Physics, Phi Beta Kappa, Sigma Xi, Bertman Research Prize, Thorndike award, Eastman-Kodak research fellowship

Research Interests

Computer networks, distributed systems, modeling and performance evaluation. My thesis examined the problem of supporting time-constrained communication applications in multiple access networks. My current research focuses on examining mathematical models of human economic and social interaction as metaphors for achieving cooperation and resource sharing in distributed systems, the unification of performance and correctness considerations in protocol verification, performance analysis of time-constrained communication protocols and the development of modeling and analysis tools for computer communication systems.

Work Experience

Assistant Professor, Univeristy of Massachusetts, 9/84 to present
Department of Computer and Information Science.

Consultant, IBM Thomas J. Watson Research Center, 9/80 to present
Design and implementation of the modeling language and solution components for RESQ, a system for constructing and analyzing models of resource contention systems. Performance Modeling and Methodology group.

Instructor, Graduate Fellow, 9/80 to 5/84
Department of Computer Science, Columbia University.

Systems Supervisor, Columbia University, 9/79 to 5/80
Full time technical, managerial and budgetary responsibility for PDP 11/60 UNIX system in the School of Engineering. System maintenance, systems programming, user consultation.

Systems Programming, IBM Thomas J. Watson Research Center, 5/79 to 8/79
Design and implementation of a programmable i/o diagnostic monitor for an experimental minicomputer.

Refereed Journal Publications

J.F. Kurose, M. Schwartz and Y. Yemini, "Multiple Access Protocols and Real Time Communication", *Computing Surveys,* Vol. 16, No. 1, March, 1984.

Y. Yemini and J.F. Kurose, "Can Current Protocol Verification Techniques Guarantee Correctness?", *Computer Networks,* Vol. 6, Dec. 1982, pp. 377-381.

C.H. Sauer, E.A. MacNair and J.F. Kurose, "Queueing Network Simulation of Computer Communication", *IEEE Journal on Selected Topics in Communications* Vol. SAC-2, No. 1, Jan. 1984, pp.203-220.

Refereed Conference Publications, Contributions to Edited Volumes

J.F. Kurose, M. Schwartz and Y. Yemini, "A Microeconomic Approach Towards Optimizing Channel Sharing in Multiaccess Networks", submitted to *5th Int. Conference on Distributed Computing Principles,* May 1985

J.F. Kurose, M. Schwartz and Y. Yemini, "Controlling Time Window Protocols for Time-Constrained Communication in a Multiple Access Environment",*8th Int. Data Communication Symposium,* October 1983.

J.F. Kurose and M. Schwartz, "A Family of Window Protocols for Time Contrained Communication in a Multiple Access Environment", *IEEE INFOCOM,* April 1983.

J.F. Kurose and Y. Yemini, "The Specification and Verification of a Connection Establishment Protocol Using Temporal Logic", in Protocol Specification, Testing, and Verification, C. Sunshine (ed.), North-Holland, 1982.

Y. Yemini and J.F. Kurose, "Towards the Unification of the Functional and Performance Analysis of Protocols", in Protocol Specification, Testing and Verification, C. Sunshine (ed.), North-Holland, 1982.

C.H. Sauer, E.A. MacNair and J.F. Kurose, "RESQ: Past, Present and Future", *National Computer Conference,* June 1982.

Technical Reports

J. Kurose, "Time-Constrained Communication in Multiple Access Networks", Department of Computer Science, Columbia University, May 1984.

J.F. Kurose and S. Salza, "Structure and Internal Operation of the RESQ Language Translator", IBM proprietary technical report, Aug. 1983.

A. Blum, E. Jaffe, E.A. MacNair, J.F. Kurose, C.H. Sauer, "RESQ: An Introduction and Update", IBM Research Report RC-10016, Yorktown Heights, N.Y., June, 1983.

C.H. Sauer, E.A. MacNair and J.F. Kurose, "RESQ: CMS User's Guide", IBM Research Report RA-139, Yorktown Heights, N.Y., April 1982.

C.H. Sauer, E.A. MacNair and J.F. Kurose, "RESQ: Introduction and Examples", IBM Research Report RA-138, Yorktown Heights, N.Y., April 1982.

C.H. Sauer, E.A. MacNair and J.F. Kurose, "Computer and Communication System Modeling with RESQ", IBM Research Report RA-128, Yorktown Heights, N.Y., November 1981.

Conference and Workshop Participation/Presentations

IEEE Computer Networking Symposium, Laurel Park MD, Dec. 1983, attendee.
8th International Data Communication Symposium, Falmouth MA, Oct. 1983, speaker.
Institute for Mathematics and Its Application, University of Minnesota, Mineappolis MN, Oct. 1983, invited visitor.
5th RESQ Modeling Workshop, Yorktown Heights NY, Oct. 1983, speaker.
IEEE INFOCOM Conference, San Diego CA, Apr. 1983, speaker.
4th RESQ Modeling Workshop, Yorktown Heights NY, Oct. 1982, speaker.
2nd International Workshop on Protocol Specification, Testing and Vertification Idlywild CA, May 1982, speaker.
ACM Computer Networks Symposium, College Park MD, Apr. 1982, attendee.
3rd RESQ Modeling Workshop, Yorktown Heights NY, Oct. 1983, attendee.

Colloquia and Talks

University of Maryland, College Park MD, March 1984.
Harvard University, Cambridge MA., March 1984.
University of Massachusetts, Amherst MA, February 1984.
Princeton University, Princeton NJ, February 1984.
Yale University, New Haven Conn., February 1984.
University of Virginia, Charlottesville VA, February 1984.
Columbia University, Department of Electrical Engineering, New York NY,

March 1983.
IBM Thomas J. Watson Research Center, Yorktown Heights NY, November 1982.
Columbia University, Department of Electrical Engineering, New York NY,
March 1982.

Professional Activities

Referee for: *Computer Networks*
IEEE INFOCOM Conference
Member ACM, IEEE.

Personal Data

Born 9/13/56, Greenwich, Conn.
US citizen
Social Security Number 040-52-9130

Curriculum Vitae of
John Eliot Blakeslee Moss

Education

1981: PhD, Computer Science, MIT
1978: MSEE and EE, Computer Science, MIT
1975: BSEE, Computer Science, MIT

Experience

Research

Personal research in nested transactions and reliable distributed databases, with emphasis on crash recovery algorithms and semantics;

Joint work on recovery semantics in progress with Nancy Griffeth and Marc Graham (Ga. Tech.)

Consultant to TeleSoft (Dr. Gerry Fisher) on compilation of Ada for microprocessors

PhD research on reliable distributed databases and nested transactions, under Barbara Liskov (MIT Lab. for Comp. Sci.)

Distributed Systems Group, MIT Lab. for Comp. Sci. (Liskov)

Masters' thesis research on abstract data types in programming languages, under Barbara Liskov (MIT Lab. for Comp. Sci.)

CLU language design team, MIT Lab. for Comp. Sci. (Liskov)

Consultant to Intermetrics (Cambridge, MA) on RED language (last round of Ada competition) under Ben Brosgol

Graduate and undergraduate lab experience with programming language design and implementation; distributed systems architecture; networks and protocols; formal semantics; document preparation; hardware design and construction; microprocessor hardware, software, and interfacing

Teaching

Undergraduate course on information structures (Dickinson College)

Short course on Ada programming language (Army War College)

Lectures on various computer topics (Army War College)

Several short courses on microprocessors (MIT)

Assistant for courses on formal semantics of programming languages, discrete mathematics, and digital hardware (MIT)

Programming

1981 to present: US Army officer at Army War College; design, code, document, and maintain war games and microcomputer system software

Graduate school: MIT Lab. for Comp. Science; language run-time features for CLU, including garbage collectors; document preparation and network utilities

Areas of interest

Distributed computing: nested transactions; distributed database concurrency and recovery algorithms and semantics; reliability; network protocols; security; distributed operating systems

Computer architecture: processor and system architecture especially instruction set architecture; capability and very large address space systems; garbage collection; reduced instruction sets; language and operating system support features

Programming languages: compiler technology (especially code generation and microprocessors); design of advanced features such as abstract data type support mechanisms; object oriented languages; logic programming languages; run-time system design; Ada

Microprocessors: architecture; operating systems; compilers; etc.

Publications

"Nested Transactions: An Introduction", chapter of a book to appear in 1984, edited by Bharat Bhargava (Univ. of Pitt.)

"Checkpoint and Restart in Distributed Transaction Systems", 3rd Symp. on Reliability in Distributed Software and Database Systems (October, 1983)

"Nested Transactions and Reliable Distributed Computing", 2nd Symp. on Rel. in Distr. Soft. and Database Systems (August, 1982)

"Nested Transactions: An Approach to Reliable Distributed Computing", PhD Thesis, MIT, 1981

"Abstract Data Types in Stack Based Languages", Master's Thesis, MIT, 1978

"CLU Reference Manual", Springer-Verlag, 1981, co-author

Conferences and presentations

Colloquium speaker, Digital Equipment Corp. (Hudson, MA), November, 1983

Colloquium speaker, Univ. of Mass. (Amherst), November, 1983 (nested transactions)

AdaTEC meeting, October, 1983, Dallas, TX

3rd Symposium on Reliability in Distributed Software and Database Systems, October, 1983, Clearwater Beach, Florida; paper presented

Modeling, Simulation, and Gaming of Warfare, August 1983, Georgia Tech.

Colloquium speaker, Univ. of Pitt., June, 1983 (nested transactions)

Colloquium speaker, Georgia Tech., February, 1983 (nested transactions)

Colloquium speaker, Princeton Univ., November, 1982 (nested transactions)

Colloquium speaker, Computer Corp. of America (Cambridge, MA), November, 1982 (nested transactions)

AdaTEC Conference on Ada, October, 1982, Arlington, VA

2nd Symposium on Reliability in Distributed Software and Database Systems, August, 1982, Pittsburgh, PA; paper presented

Honors, Awards, Certifications

Army Commendation Medal, for programming work April 1982 to July 1982

Army Achievement Medal, for programming work August 1981 to March 1982

Certificate in Computer Programming, Institute for the Certification of Computer Professionals (ICCP)

Certificate in Data Processing, ICCP

National Science Foundation Graduate Fellowship

Army ROTC Four Year Scholarship

National Merit Scholarship

Personal information

Addresses:

CPT Eliot Moss
965 Crain's Gap Rd.
Carlisle, PA 17013
(717) 245-2596

or

CPT Eliot Moss
Box 483, USAWC
Carlisle Barracks, PA 17013
(717) 245-4334, AV 242-4334

ARPAnet: EBM at MIT-XX

Born: January 1, 1954, at Staunton, VA

Married: May 29, 1976 (Hannah Allen Abbott); no children

VITA

David W. Stemple
Associate Professor
Computer and Information Science
University of Massachusetts, Amherst

EDUCATION

Ph.D. - Computer and Information Science, University of Massachusetts, 1977
M.S. - Math, West Virginia University, 1960
A.B. - Math, West Virginia University, 1959

POSITIONS

1981 - 1984 Associate Professor, Computer and Information Science, University of Massachusetts, Amherst

1971 - 1981 Associate Director, Systems Research and Development, University Computing Center, University of Massachusetts, Amherst

Adjunct Assistant Professor of Computer and Information Science, University of Massachusetts, Amherst

1968 - 1971 Vice President, Research, Multicomp, Inc., Wellesley, MA

1966 - 1968 Systems Analyst and Lecturer, Research Computing Center and Computer Science Program, University of Massachusetts, Amherst

1963 - 1965 Systems Analyst, Computer Usage Company, New York

1960 - 1963 Systems Programmer, IBM, New York

1959 - 1960 Instructor of Mathematics, Wheeling College, Wheeling, West Virginia

DEVELOPMENT EXPERIENCE

Worked on the improvements to the original FORTRAN which were involved in the transfer of the compiler from the IBM 704 to the 709 and 7090.

Designed and wrote part of the FORTRAN IV compiler for the IBM 7094.

Designed and wrote part of a JOVIAL compiler for the IBM 360/9020 used by FAA in air traffic control.

Was one of a three-man team which designed an operating system for the RCA Spectra 70/45 and 55. (consultant)

Worked on the design and implementation of a time-sharing system for the CDC 3600. This system was used for eight years by the University of Massachusetts and is still being used by United Computing Services. Some of my specific responsibilities were the execution monitor, the file system, the BASIC compiler and simulation of certain processes.

Designed many databases, from a research database on property transfer and families in sixteenth century Mexico, to administrative databases for space management and scheduling.

Designed and directed the implementation of a relational database system (TABLET) and many other systems, from an editing pseudo-machine to a graphical interface to the Entity-relationship Model.

GRANTS

Control Data Corporation Grant M106A for the Investigation of Independence in Data Manipulation Languages, approximately $21,000.

NSF Grant MCS78-07616 - Effects of procedurality on the learnability of two relational query languages. 1978-1979 ($24,594)

Faculty Research Grant 2-03197 for the establishment of a Research Interest Profile System. 1979 ($1,165)

Faculty Research Grant for the building of a testbed operating system. 1983 ($2,000)

HONORS

Phi Beta Kappa, Sigma Pi Sigma (Physics)
Alpha Pi Mu (Industrial Engineering)

PUBLICATIONS

Journals and refereed conferences

Specification and Verification of Abstract Database Types, with Tim Sheard, presented at the Third Symposium on Principles of Database Systems, April, 1984, Waterloo, Ontario.

Primitives for Accessing Protected Objects, with Krithi Ramamritham, and Stephen Vinter, Third Symposium in Distributed Software and Database Systems, Clearwater Beach, Florida, October, 1983.

Operating System Support for Abstract Database Types, with Krithi Ramamritham, Stephen Vinter, and Tim Sheard, presented at the Second International Conference on Databases, Cambridge, England, August, 1983.

Protecting Objects through the Use of Ports, with Stephen Vinter and Krithi Ramamritham, presented at the Second Annual Phoenix Conference on Computers and Communications, Phoenix, Arizona, March, 1983.

Strategies and Interfaces (abstract) in Proceedings of ACM 82 Conference, Dallas, October, 1982.

Human Factors Comparison of a Procedural and a Nonprocedural Query Language, with Charles Welty, ACM Transactions on Database Systems, Vol.6, No.4, December 1981.

Dissertation: A Database Management Facility and Architecture for the Realization of Data Independence.

A Database Management Facility for Automatic Generation of Database Managers, ACM Transactions on Database Systems, Vol. 1, No. 1, March, 1976.

On the Development of Data Base Editions, Data Base Management, J.W. Klimbie and K.L. Koffeman, eds., North-Holland Pub. Co., Amsterdam, 1974, with R. W. Taylor.

Technical reports and other unrefereed publications

Verifiable Abstract Database Types, with Tim Sheard, COINS TR 83-37, University of Massachusetts.

Preliminary Design of a Port-oriented Operating System, with Krithi Ramamritham and Steve Vinter, COINS TR 82-24, University of Massachusetts.

Generalized Type Specification for Database Systems, COINS TR 82-15, University of Massachusetts.

TABLET, A Relational Query Language, Proceedings of VIM-30, Scottsdale, Arizona, 1979.

TABLET: The Algebra Based Language for Enquiring of Tables, TR78-19, Computer and Information Science Department, University of Massachusetts, 1978.

Programming the IBM System 360 with other authors of the staff of Computer Usage Comapany, edited by Ascher Opler, John Wiley & Sons, Inc., 1966.

UMASS - A Modest Time Sharing System for the CDC 3600, with C. Foster and R. Hambleton, University of Massachusetts Research Computing Center Technical Note.

Papers currently submitted for publication —

Types and Cooperation Classes: Interprocess Communication without Process Identifiers, with Krithi Ramamritham and Stephen Vinter, submitted for journal publication.

Privilege Transfer and Revocation in a Port-based System, with Krithi Ramamritham, David Briggs, and Stephen Vinter, submitted for journal publication.

Dynamic Control of Module Interconnection, with Krithi Ramamritham and Stephen Vinter, submitted for journal publication.

CURRICULUM VITAE

Jack Craig Wileden
Department of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003
(413) 545–0289 or 545–2744

home:
115 Pelham Road
Amherst, MA 01002
(413) 256–8882

Personal Information: Born September 25, 1950, Lansing, Michigan
Social Security Number 373–56–0737
Citizenship USA

Degrees: AB 1972 (Mathematics), University of Michigan
MS 1973, PhD 1978 (Computer and Communication Sciences),
University of Michigan

Current Position: Associate Professor of Computer and Information Science
University of Massachusetts

Memberships: Association for Computing Machinery (ACM)
ACM Special Interest Groups: SIGPLAN, SIGOPS, SIGACT, SIGSOFT
Institute of Electrical and Electronics Engineers (IEEE)
IEEE Computer Society
European Association for Theoretical Computer Science
Sigma Xi

Research Interests: Integrated Software Development Environments
Tools for Pre-Implementation Description and Analysis
Formal Methods for Describing and Analyzing Distributed Systems
Debugging Techniques for Distributed Systems
Issues in Programming Language Design

Research Support: Principal Investigator, "Tools Supporting Pre-Implementation Development of Flight Control Software", National Aeronautics and Space Administration, January 1, 1981 to February 28, 1983, $134,721

Honors and Awards: ACM National Lectureship 1978–1980
University of Michigan, Rackham Predoctoral Fellow
July 1977–December 1977

JACK C. WILEDEN

Recent Publications

Journal Papers:

J. Wileden, J. Sayler, W. Riddle, A. Segal and A. Stavely. Behavior Specification in a Software Design System. *Journal of Systems and Software*, 3, 2, (June 1983), 123-135.

P. Bates and J. Wileden. High Level Debugging of Distributed Systems. *Journal of Systems and Software*, 3, 4, (December 1983), 255-264.

A. Marmor-Squires, W. Riddle, G. Sumrall, J. Wileden. The Support System Task Area. *IEEE Computer*, 16, 11, (Nov. 1983), 97-104.

G. Avrunin and J. Wileden. Techniques for Analyzing Designs of Dynamically Structured Distributed Systems. *ACM Transactions on Programming Languages and Systems*, (to appear).

Refereed Conference Papers:

P. Bates and J. Wileden. EDL: A Basis for Distributed System Debugging Tools. *Proc. 15th Hawaii International Conference on Systems Science*, Honolulu, (January 1982).

J. Wileden. Constrained Expressions and the Analysis of Designs for Dynamically-Structured Distributed Systems. *Proc. 1982 International Conference on Parallel Processing*, Shanty Creek, Michigan, (August 1982).

G. Avrunin and J. Wileden. Algebraic Techniques for the Analysis of Concurrent Systems. *Proceedings of the 16th Hawaii International Conference on Systems Science*, Honolulu, (January 1983).

P. Bates, J. Wileden and V. Lesser. A Debugging Tool for Distributed Systems. *Proceedings of the Phoenix Conference on Computers and Communication*, (March 1983).

P. Bates and J. Wileden, An Approach to High-Level Debugging of Distributed Systems, *Proc. Software Engineering Symposium on High-Level Debugging*, Asilomar, California (March 1983).

J. Wileden and L. Clarke, Feedback-Directed Development of Complex Software Systems. *Proc. Software Process Workshop*, Egham, England (February 1984).

A. Wolf, L. Clarke and J. Wileden. An Ada Environment for Programming-in-the-Large. *Proc. Conference on Ada Applications and Environments*, St. Paul, Minnesota (October 1984).

Unpublished Reports:

"Introduction to a Unified Treatment of Interface Control and Program Structure," (with L. Clarke and A. Wolf), COINS Technical Report 82-32, December 1982.

"Analyzing Distributed Systems Using Constrained Expressions," (with L. Dillon and G. Avrunin), February 1983.

"Precise Interface Control: System Structure, Language Constructs, and Support Environment," (with L. Clarke and A. Wolf), COINS Technical Report 83-26, August 1983.

"Describing and Analyzing Distributed Software System Designs," (with G. Avrunin), COINS Technical Report 83-28, August 1983.

"A Formalism for Describing and Evaluating Visibility Control Mechanisms," (with L. Clarke and A. Wolf), COINS Technical Report 83-34, October 1983.

August 1984

CURRICULUM VITAE

Walter H. Kohler

PERSONAL DATA:

Addresses: (Home)

15 Ash Lane
Amherst, MA 01002
(413) 253-7923

(Office)

Department of Electrical
and Computer Engineering
Univ. of Massachusetts
Amherst, MA 01003
(413) 545-0765/545-2441

Born: June 2, 1945, Dover, New Jersey, U.S.A.

Marital Status: Married, two children.

EDUCATION:

Doctor of Philosophy (Electrical Engineering), Princeton University, Princeton, New Jersey, October 1972.

Master of Arts (Electrical Engineering), Princeton University, Princeton, New Jersey, January 1971.

Master of Science in Engineering (Electrical Engineering), Princeton University, Princeton, New Jersey, June 1968.

Bachelor of Science in Engineering (Electrical Engineering), Princeton University, Princeton, New Jersey, June 1967.

POSITIONS HELD:

Visiting Scientist, Digital Equipment Corporation, New England Research Center, Hudson, MA, July 1984 - present.

Acting Head, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, September 1982 - August 1983.

Associate Professor of Electrical and Computer Engineering, University of Massachusetts, Amherst, September 1978 - present.

Assistant Professor of Electrical and Computer Engineering, University of Massachusetts, Amherst, September 1972 - August 1978.

Research Assistant/Teaching Assistant, Department of Electrical Engineering, Princeton University, September 1969 - August 1972.

Member of Technical Staff, Bell Telephone Laboratories, Government and Special Systems Engineering Center, Holmdel, New Jersey, June

1968 - September 1969.

Summer Associate, Bell Telephone Laboratories, Telephone Studies Department, Holmdel, New Jersey, June 1967 - August 1967.

Systems Engineering Trainee, International Business Machines, Poughkeepsie, New York and Trenton, New Jersey, June 1966 - August 1966.

CONSULTING ACTIVITIES:

Department of the Army, CORADCOM, DRDCO-TCS-BC, Ft. Monmouth, New Jersey, Computer architecture selection, 1978 - 1979.

Digital Equipment Corporation, Corporate Research and Architecture Group, Hudson, Massachusetts, Software research in distributed databases, transaction processing, and object based systems design, 1979 - present.

AREAS OF RESEARCH INTEREST AND EXPERIENCE:

Distributed Database System Design
Concurrency Control and Recovery Protocols
Computer Performance Measurement, Modeling, and Evaluation
Algorithms for Combinatorial Optimization Problems

TEACHING EXPERIENCE:

Undergraduate:

Introduction to Computing (PASCAL, FORTRAN)
Introduction to Computer Engineering
Introduction to Discrete Systems
Introduction to Digital Systems
Digital Circuits Laboratory

Advanced Undergraduate/Graduate:

Computer Organization and Assembly Language Programming
Data Structures and Computer Algorithms
Advanced Switching Theory
Operating Systems
Programming Language Design and Implementation

Graduate:

Distributed Database Systems
Computer Performance Modeling and Evaluation
Design and Analysis of Computer Algorithms
Linear System Theory
Graph Theory and Its Applications

ADMINISTRATIVE EXPERIENCE:

Departmental/School Administration:

Acting Head, Department of Electrical and Computer Engineering: 9/82 - 8/83.
Director, Engineering Computer Station: 9/78 - 1/79, 9/81-8/83.
Director, ECE Graduate Program: 9/80 - 9/81
Director, ECE Undergraduate Program: 9/78 - 8/79.
IEEE Student Branch Counselor: 9/78 - 8/79.

Research Administration:

National Science Foundation Grants: 1973 - present.
Digital Equipment Corporation Project Leader: 6/80 - 8/82.

PROFESSIONAL SOCIETY MEMBERSHIP:

Association for Computing Machinery (SIGOPS, SIGMETRICS, SIGARCH, SIGMOD, SIGPLAN)
Institute of Electrical and Electronic Engineers (Computer Society)

PROFESSIONAL ACTIVITIES:

Reviewer:

Journal of the ACM
Communications of the ACM
Computing Reviews
IEEE Computer
IEEE Transactions on Computers
ACM Transactions on Mathematical Software
Mathematics of Operations Research
ACM Symposium on Operating System Principles
Management Science
Operations Research
IEEE Computer Architecture Symposium
International Symposium on Computer Performance Modeling, Measurement, and Evaluation
National Science Foundation
National Research Council

Technical Conference Organization:

COMPCON 80 (IEEE Computer Society International Conference), Session Chairman, Synchronization, Recovery, and Reliability Issues in the Distributed Database Environment.

Fifth Texas Conference on Computing Systems, Program Committee.

RESEARCH/EDUCATIONAL GRANTS:

"Research Initiation - Branch and Bound Algorithms for Combinatorial Problems," Principal Investigator, NSF GK-37400, $17,000, for the period 4/1/73 - 3/31/75.

"Laboratory Equipment for Graduate Research in Computer Engineering," Co-principal Investigator with H.S. Stone, NSF ENG 75-20070, $24,300 plus $20,000 matching State Funds, for the period 6/1/76 - 11/30/77.

"Instructional Scientific Equipment Program - Computer Systems Laboratory," Principal Investigator, NSF SER 76-12941, $14,200 plus $14,200 matching State Funds, for the period 6/1/76 - 5/31/78.

"Queueing Models for Distributed Computer Systems," Principal Investigator, NSF MCS 76-03667, $45,397, for the period 6/1/76 - 11/30/78.

"Sabbatical Research on Distributed System Design," Principal Investigator, Digital Equipment Corporation, $23,254, for the period 6/1/79 - 5/30/80.

"Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems," Principal Investigator and Project Director, (Professor Donald F. Towsley and John A. Stankovic, Co-Principal Investigators), NSF ECS 81-20931, $134,190 for the period 6/1/82 - 5/30/84.

"Data General's Equipment Donation Program," Project Director, Data General Corporation, $350,000 MV8000 Computer System, October 1982.

"Personal Computers in Engineering and Computer Science Education," Principal Investigator, Digital Equipment Corporation, $787,853 equipment allowance, October 1983.

PUBLICATIONS

"A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems," Computing Surveys, Vol.13, No. 2, June 1981, pages 149-183. (Translated into Japanese and reprinted in Bit, Vol. 1, 1983, pages 41-73.)

"Architectural Overview of a Distributed Software Testbed," (with H. Garcia-Molina and F. Germano), Proceedings 16th. Annual Hawaii International Conference on System Sciences, January 1983.

"An Experimental Comparison of Locking Policies in a Testbed Database System," (with K. Wilner and J.A. Stankovic), SIGMOD'83, San Jose, California, May 1983, pages 108-119.

"Current Research and Critical Issues in Distributed Software Systems," (with J.A. Stankovic and K. Ramamritham), Technical Report, University of Massachusetts/Amherst, March 1983, 66 pages.

(Submitted for publication.)

"CARAT: a Distributed Software Testbed," (with Bao-Chyuan Jenq), Technical Report, University of Massachusetts/Amherst, December 1983, 30 pages.

"Debugging a Distributed Computing System," (with H. Garcia-Molina and F. Germano), IEEE Transactions on Software Engineering, Vol. SE-10, No. 2, March 1984, pages 210-219.

SHORT VITA

John A. Stankovic

BIRTH:	September 15, 1948, Brooklyn, New York
MARITAL STATUS:	Married, two children
ADDRESS:	RR 3, January Hills Road, Amherst, MA 01002
TELEPHONE:	(413) 545-0720 (office) (413) 549-4045 (home)
EDUCATION:	1970: Sc.B. Electrical Engineering, Brown University 1976: Sc.M. Computer Science, Brown University 1979: Ph.D. Computer Science, Brown University

HONORS:

Outstanding Junior Faculty Award (1983) - School of Engineering, University of Massachusetts

PhD thesis published as a book as part of a series of the best PhD theses in Computer Science

Sigma Xi

New York State Regents Scholarship

PROFESSIONAL EXPERIENCE:

June 1970 - November 1974: employed as a software engineer by the Western Electric Company. My work included large real-time software system design and implementation, assembly language programming, unit and integration testing, and continual software modification and maintenance for the Surveillance function of the SAFEGUARD Anti-Ballistic Missile system. I was also responsible for live mission support and analysis. The work was performed at Bell Labs, Whippany, N.J., and the Kwajalein Missile Range, Marshall Islands.

January 1975 - May 1979: Research Assistant, Brown University.

September - December 1976: Teaching Assistant, Brown University.

June 16 - August 15, 1980: Visiting Research Scientist, Computer Sciences Corp., Newport, R.I.

June 1, 1982 - June 20, 1982: Visiting Research Scientist, INRIA, Rocquencourt, France.

August 1979 - August 1984: Assistant Professor, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA.

September 1984 - present: Associate Professor, Department of Electrical and Computer Engineering, University cf Massachusetts, Amherst, MA.

BOOKS (last 2 years):

Structured Systems and Their Performance Improvement Through Vertical Migration, UMI Research Press, Ann Arbor, Michigan, 1982.

"Vertical Migration," with Tom Weidner, chapter in Handbook of Microprogramming and Firmware Engineering, edited by Stanley Habib and Subrata Dasgupta, Van Nostrand Inc., to appear.

TUTORIAL TEXTS and PRESENTATIONS:

Introduction to Computer Networks and Distributed Operating Systems, IEEE Computer Society, Chapter Tutorial Workbook, invited tutorial text, December 1983.

Reliable Distributed System Software, IEEE Press, Tutorial Text, submitted for publication, July 1984.

Tutorial on Distributed Software presented at the Third Symposium on Reliability in Distributed Software and Databases, October 17, 1983.

Three day tutorial on Distributed Processing presented at IBM, Kingston, N.Y., with Harold Stone, July 20-22, 1981.

FORMAL PUBLICATIONS (last 2 years):

"Stability and Distributed Scheduling Algorithms," invited paper, ACM National Conference, March 1985.

"Scheduling Tasks With Resource Requirements in Hard Real-Time Systems," with Wei Zhao and Krithivasan Ramamritham, submitted to IEEE Transactions on Software Engineering, May 1984.

"Perspectives on Distributed Computer Systems," invited paper, Special Issue in Celebration of IEEE Centennial, IEEE Transactions on Computers, accepted for publication.

"An Adaptive Bidding Algorithm For Processes, Clusters and Distributed Groups," with Inderjit Sidhu, Proceedings 4TH International Conference on Distributed Computing Systems, May 1984.

"Dynamic Task Scheduling in Distributed Hard Real-Time Systems," with Krithivasan Ramamritham, IEEE Software, Vol. 1, No. 3, July 1984.

"An Application of Bayesian Decision Theory to Decentralized Control of Job Scheduling," IEEE Transactions on Computers, accepted for publication.

"Current Research and Critical Issues in Distributed Software Systems," with K. Ramamritham, and W. Kohler, submitted to ACM Computing Surveys, March 1983.

"Simulations of Three Adaptive, Decentralized Controlled, Job Scheduling Algorithms," Computer Networks, accepted for publication.

"An Experimental Comparison of Locking Policies in a Testbed Database System," with Walt Kohler and Ken Wilner, Proceedings of SIGMOD, May 1983.

"A Technique To Identify Implicit Information Associated With Modified Code," Proceedings Conference on System Description Methodologies, Kecskemet, Hungary, May 1983.

"A Heuristic For Cooperation Among Decentralized Controllers," Proceedings INFOCOM 83, invited paper, April 1983.

"Achievable Decentralized Control For Functions of a Distributed Processing Operating System," Proceedings of COMPSAC, November 1982.

"An Evaluation of the Applicability of Different Mathematical Approaches to the Analysis of Decentralized Control Algorithms," with Noor Chowdhury, Ravi Mirchandaney, and Inderjit Sidhu, Proceedings of COMPSAC, November 1982.

"Good System Structure Features: Their Complexity and Execution Time Cost," IEEE Transactions on Software Engineering, Vol. SE-8, No. 4, pp. 306-318, July 1982.

"Software Communication Mechanisms: Procedure Calls Versus Messages," IEEE Computer, Vol. 15, No. 4, pp. 19-25, April 1982.

RESEARCH CONTRACTS AND GRANTS (PI unless otherwise noted) (last 2 years):

Robust Algorithms for Real-Time Distributed Systems, submitted as a proposed extension to US Army CECOM Contract DAAB07-82-K-J015, 9 months, $26,000.

Design, Evaluation, and Modeling of Distributed Database Systems, National Science Foundation, PI, Walt Kohler, Co-PI, Don Towsley, 2 years, submitted in July 1984. Amount $268,222.

Scheduling in Real-Time Distributed Systems, National Science Foundation, Co-PI, K. Ramamritham, 2 years, submitted in March 1984. Amount $293,776.

Non-Traditional Architecture for Distributed Real-Time Systems, General Electric, Sept. 1, 1984 - August 31, 1985. Amount $18,000.

A Study of Deadlock Detection in Distributed Databases, Naval Underwater Systems Center, N00140-84-M-WM07, Co-PIs, Walt Kohler and Don Towsley, July 1 - December 31, 1984. Amount $9930.

Load Balancing, Naval Underwater Systems Center, N00140-84M-VG68, Co-PI, Don Towsley. March 1, 1984 - August 31, 1984. Amount $9970.

Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems, National Science Foundation, ECS-8120931, PI, Walter Kohler, Co-PI, Donald Towsley. June 1, 1982 - May 31, 1984. Amount $134,190.

Decentralized Control of Scheduling in Distributed Processing Systems, U.S. Army CECOM, DAAB07-82-K-J015, Dec. 15, 1981 - Feb. 15, 1985. Amount $130,830.

Decentralized Control of Job and Process Scheduling in Distributed Processing Systems, National Science Foundation, MCS-8104203, July 1, 1981 - June 30, 1983. Amount $38,799.

Software Tools For the Support of System Modification, Naval Underwater Systems Center, Newport, RI, N00140-81-M-MY61, August 1, 1981 - January 31, 1982. Amount $6,000.

Design and Development of a Distributed Computer-Based Instructional System, Digital Equipment Corporation, BD-749339, July 1, 1981 - June 30,1982. Amount $47,350.

SPECIAL EDITORSHIPS (last 2 years):

Guest Editor of a Special Issue of IEEE Distributed Processing Newsletter, on Distributed Operating Systems, Vol. 6, No. SI-2, June 1984.

PROFESSIONAL SERVICE POSITIONS (last 2 years):

Speaker, IEEE Chapter Tutorials Program, 1984-85.

Vice Chairman for IEEE Distributed Operating Systems Technical Committee on Distributed Processing, Jan. 1, 1981 to the present.

CSnet Liaison for the Department of Electrical and Computer Engineering, Univ. of Mass., January 1982 - August 1983.

PROFESSIONAL SERVICE (last 2 years):

Program Committee, Real-Time Systems Symposium, Austin, Texas, to be held in December 1984.

Program Committee, Vice Chairman for Distributed Operating Systems and Programming Languages, The Fifth International Conference on Distributed Computing Systems, to be held in May 1985.

Program Committee, The Fourth International Conference on Distributed Computing, San Francisco, April 1984.

Awards Committee, Computer Data Engineering Conference, Los Angeles, April 24-27, 1984.

Program Committee, The Third Symposium on Reliability in Distributed Software and Databases, October 18-19, 1983.

INVITED LECTURES, PANELS AND CONFERENCE PRESENTATIONS (last 2 years):

Invited Speaker on Distributed Systems, ACM National Conference, to be held in New Orleans, March 1985.

Apollo Computer, CS seminar, Chelmsford, Ma., June 28, 1984.

US Army CECOM, 2 paper presentations, Fort Monmouth, NJ, June 12, 1984.

Fourth International Conference on Distributed Computing Systems, paper presentation, San Francisco, May 17, 1984.

University of California, Berkeley, CS Seminar, May 16, 1984.

INRIA, France, Invited Lecture, January 18, 1984.

Panel Member, Critical Issues in Real-Time Operating Systems, Real Time Systems Symposium, Washington, D.C., Dec 7, 1983.

IEEE First Workshop on Real-Time Operating Systems, Paper Presentation, Niagara Falls, NY, August 12, 1983.

Keynote Speaker, Working Conference on Distributed AI, Holyoke, Massachusetts, June 8, 1983.

Working Conference on System Description Methodologies, Paper Presentation, by special invitation, Kecskemet, Hungary, May 27,1983.

Presented results of a working group on "Specifications to Code Mapping", Kecskemet, Hungary, May 27, 1983.

INFOCOM83, Paper Presentation, San Diego, Calif., April 19, 1983.

COMPSAC82, Two Papers Presented, Chicago, November 10, 1982.

U.S. Army, CECOM, Fort Monmouth, NJ, presentation of semi-annual report of a research contract, July 28, 1982.

INRIA, Rocquencourt, France, CS Seminar, June 16, 1982.

U.S. Army, CECOM, Fort Monmouth, NJ, a series of presentations, March 25, 1982.

Boston University, CS Seminar, February 5, 1982.

VITA

Donald F. Towsley

Personal

Date of Birth:	September 30, 1949
Place of Birth:	Timmins, Ontario, Canada
Citizenship:	United States Citizenship
Home Address:	33 Sunset Ave. Amherst, MA 01002
Telephone:	(413) 545-0766 (office) (413) 549-0436 (home)

Degrees

B.A., Physics, University of Texas	1971
Ph.D., Computer Sciences, University of Texas	1975

Professional Experience

Research Assistant, Department of Computer Sciences, University of Texas	1971-1975
Postdoctoral Fellow, Department of Computer Sciences, University of Texas	1975-1976
Assistant Professor,Department of Electrical and and Computer Engineering, University of Massachusetts	1976-1981
Associate Professor, Department of Electrical and and Computer Engineering, University of Massachusetts	1981-
Visiting Scientist, IBM Research division, Yorktown Heights	1982-1983

Professional Affiliations and Activities

Member of 1984 SIGMETRICS Conference Program Committee
Associate Editor, Journal of NETWORKS
Member of: Association of Computing Machinery (ACM)
Institute of Electrical and Electronics Engineers
Operations Research Society of America

Journal Publications

"On the instability of the slotted ALOHA multiaccess algorithm," (with W. A. Rosenkrantz), IEEE Trans. on Automatic Control, Vol. AC-28, 10, pp. 994-996, Oct. 1983.

"Random multiple-access communication and group testing," (with T. Berger, N. Mehravari, and J.K. Wolf), IEEE Trans. on Communications , vol. COM-32, pp. 669-679, July 1984.

"An analysis of a point to multipoint channel using a go back-N error control protocol," to appear in IEEE Trans. on Communications.

"On adaptive tree polling algorithms", (with J. K. Wolf), to appear Dec. 1984 in IEEE Trans. on Communications.

Flow control protocols for statistical multiplexers handling real time data," (with B.G. Kim) accepted for publication IEEE Trans. on Communications.

Conference publications and presentations

"Window and tree protocols for satellite channels," (with T. Liu), Proc. of INFOCOM `83, April 1983.

"Group testing algorthms for random access communications," (with J.K. Wolf), presented at the International Symposium on Information Theory, June 1982.

"An approximate analysis of a multiple processor system," Proc. 1983 SIGMETRICS Conf., Aug. 1983.

"Optimal load balancing in distributed computer systems," (with A. Tantawi), presented at ORSA/TIMS meeting, Nov. 1983.

"A collision resolution algorithm for a multiple access channel," (with S.S. Panwar and J.K. Wolf), presented at the International Symposium on Information Theory, Sept. 1983.

"A general model for optimal load balancing in star network configurations," (with A.N. Tantawi), accepted for PERFORMANCE 84.

Research Contracts and Grants

Error control in data networks, National Science Foundation, ENG-77-33153, (Co-PI J.K. Wolf, Co-PI D. Towsley), March 1, 1978- April 30, 1980. Amount $91,130.

On the analysis and synthesis of computer communication networks, National Science Foundation, ECS-7921140, (Co-PI J.K. Wolf, Co-PI D. Towsley), June 1, 1980-March 31, 1985. Amount $164,944.

Concurrency control and recovery algorithms for distributed transaction processing systems, National Science Foundation, ECS-8120931, (PI W. Kohler, Co-PI J. Stankovic, Co-PI D. Towsley) May 1982-Nov. 1984. Amount $134,190. A two year renewal has been submitted for an amount of $250,000.

On the analysis and synthesis of computer communication networks, National Science Foundation, ECS-8310771, (Co-PI J.K. Wolf, Co-PI D. Towsley), Sept. 1, 1983-March 31, 19865. Amount $154;650.

Load balancing, Naval Underwater Systems Center, Newport, Rhode Island, N00140-84M-VG68, (PI J. Stankovic, Co-PI D. Towsley), March 1, 1984-Aug. 31, 1984. Amount $9970.

Performance and reliability issues in distributed computer systems, submitted to the National Science Foundation, Sept. 1 1984-Aug. 31 1986. Amount requested, $108,134. It has been informally accepted at about $42,000 for the first year.

Load balancing in a multiple processor system, Digital Equipment Corporation, June 1 1984-July 15, 1985. Amount $32,000.

A study on deadlock detection in distributed databases, Naval Underwater Systems Center, Newport, Rhode Island, (PI J. Stankovic, Co-PI W. Kohler, Co-PI D. Towsley) July 1, 1984-December 31. Amount $9929.

Load balancing in shared memory multiple processor systems, submitted to IBM Research Division, Nov. 1 1984-Oct. 31 1984, Amount requested, $50,000.

GRANT FUNDING - SEPTEMBER 1984 FACULTY

Principal Investigators and Associated Faculty

Approved

1. Arbib and Manes, Co-Principal Investigators, NSF, "Algebraic Semantics of Programming Languages," $98,440, 1/1/83 - 12/31/84.

2. Arbib and Riseman, Co-Principal Investigators, NSF, "Visuo-Tactile Coordination for Robot Control," $318,218, 1/15/82 - 1/14/85.

3. Croft and Lesser, Co-Principal Investigators, Digital Equipment Corporation, personnel support for Office Information Systems, $85,000, 10/1/83 - 12/31/84.

4. Cuny, Principal Investigator, IBM Faculty Development Award, $30,000, 7/1/84 - 6/30/85.

5. Cuny, Principal Investigator, ONR, "Execution Mode in Highly Parallel Computation," $149,965, 5/1/84 - 4/30/87 (first yr. = $31,082).

6. Hanson, Principal Investigator, NASA, "NASA Graduate Student Researchers Program," $15,000, 6/1/84 - 5/31/85 (to be renewed annually).

7. Hanson and Riseman, Co-Principal Investigators, AFOSR, "Representation and Control in the Interpretation of Complex Scenes," $548,500, 4/1/83 - 9/30/84. ($254,000, 10/1/84 - 9/30/85; renewal)

8. Kohler (ECE), Principal Investigator, NSF, "Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems," $134,190, 5/1/82 - 10/31/84.

9. Kohler (ECE) and Towsley (ECE), Co-Principal Investigators, Naval Underwater Systems Center, "A Study of Deadlock Detection in Distributed Databases," $9,930, 7/1/84 - 12/31/84.

10a. Lesser, Principal Investigator, DARPA, "Coordination in Distributed Problem-Solving," $379,945, 10/1/82 - 9/30/85.

10b. Lesser, Principal Investigator, NSF, "Coordination in Distributed Problem Solving," (with DARPA), $366,000, 5/15/83 - 5/14/86.

11. Lesser, Principal Investigator, RADC, "Plan Recognition in an Intelligent User Interface Design," $78,000, 2/1/84 --> 8 mos.

12. Ramamritham, Principal Investigator, NSF, "Synthesis of Resource Controllers for Distributed Systems," $35,043, 8/1/84 - 7/31/85.

13. Riseman, Principal Investigator, NSF, "A Group Research Facility for Artificial Intelligence, Distributed Computing, and Software Systems (Computer Research)," $360,000, 5/1/84 - 4/30/87 (first yr. = $160,000).

14. Riseman and Hanson, Co-Principal Investigators, DARPA, "Processing Dynamic Images from Camera Motion," $750,000, 6/1/84 - 5/31/87 (first yr. = $240,000).

15. Riseman and Hanson, Co-Principal Investigators, Rome Air Development Corp. (RADC), "Applying the VISIONS System to Interpretation of Aerial Images," $45,000, 8/1/83 - 9/30/84.

16. Riseman, Hanson, and Arbib, Co-Principal Investigators, ONR, "Dynamic Tactile Processing," $70,000, 5/1/84 - 4/30/86.

17. Riseman and Weiss, Co-Principal Investigators, A.C. Nielsen Co., "Feasibility Study for a Passive Audience-Composition Metering System," $28,320, 5/15/84 - 11/14/84.

18. Stankovic (ECE), Principal Investigator, General Electric, "Non-Traditional Architecture for Distributed Real-Time Systems, "18,000, 9/1/84 - 8/31/85.

19. Stankovic (ECE), Principal Investigator, Naval Underwater Systems Center, "A Study on Deadlock Detection in Distributed Databases," $9,930, 7/1/84 - 12/31/84.

20. Stankovic (ECE), Principal Investigator, Naval Underwater Systems Center, "Load Balancing," $9,970, 3/1/84 - 8/31/84.

21. Stankovic (ECE), Principal Investigator, U.S. Army CECOM, "Decentralized Control of Scheduling in Distributed Processing Systems," $130,830, 12/15/81 - 2/15/85.

22. Towsley (ECE), Principal Investigator, DEC, "Load Balancing in a Multiple Processor System," $32,000, 6/1/84 - 7/15/85.

23. Towsley (ECE), Principal Investigator, NSF, "On the Analysis and Synthesis of Computer Communication Networks," $154,650, 9/1/83 - 2/28/86.

24. Towsley (ECE), Principal Investigator, NSF, "Performance and Reliability Issues in Distributed Computer Systems," $42,000, 9/1/84 - 8/31/86.

Approved Equipment Grants

A. McDonald and Lesser, Co-Principal Investigators, NSF, "Equipment to Upgrade Computer Support of AI Research," $69,716, 5/15/84 - 5/14/85.

Proposed

25. Arbib, Principal Investigator, DARPA, "A Dynamic-Sensing Approach to Distributed Planning and Control for Robots," $1,836,463, 1/1/85 -12/31/88. (also submitted to NASA, NSF, and National Bureau of Standards).

26. Arbib, Principal Investigator, NIH, "Visuomotor Coordination: Neural Networks and Schemas," $1,071,670, 1/1/85 - 12/31/88. (also submitted to NSF)

27. Arbib, Principal Investigator, NSF, "Schemas for Control of Human Robot and Prosthetic Hands," $933,491, 9/1/84 - 8/31/87 (first yr. = $301,297).

28. Clarke and Wileden, Co-Principal Investigators, DARPA, "Toward Feedback Directed Development of Complex Software Systems," 1/1/85 - 12/31/88. $886,000, (first yr. = $290,000)

29. Clarke and Wileden, Co-Principal Investigators, NSF, "A Systematic Treatment of Interface Control," $238,591, 1/1/85 - 12/31/86 (first yr. = $113,100)..

30. Cohen and Lesser, Co-Principal Investigators, DARPA, "Proposal to RADC and DARPA Expert System Technology Program," $1,000,000 (4 yr. total), 10/1/84 (approx. start date)

31. Cuny, Principal Investigator, NSF, "Presidential Young Investigator's Award," $25,000, 5 years.

32. Kohler (ECE), Principal Investigator, NSF, "Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems," $250,000, 2 years.

33. Kohler (ECE) and Towsley (ECE), Co-Principal Investigators, NSF, "Design, Evaluation, and Modeling of Distributed Database Systems," $268,222, submitted in July, 1984, 2 years.

34. Lesser, Principal Investigators, GTE Laboratories, "Meta-Level Control Through Situation Analysis," $25,000, 8/1/84 - 7/31/85.

35. Lesser and Croft, Co-Principal Investigators, RADC, "A Knowledge Acquisition, Assistance and Explanation System," $2,225,001, 11/1/84 - 10/31/89 (first yr. = $431,001).

36. Ramamritham, Principal Investigators, NSF, "Presidential Young Investigator's Award," $25,000, 5 years.

37. Ramamritham and Stankovic (ECE), Co-Principal Investigators, NSF, "Scheduling in Real-Time Distributed Systems," $293,776, submitted in March, 1984, 2 years.

38. Riseman, Principal Investigator, General Electric Foundation, "The Recruitment and Retention of Doctoral Candidates in Computer Science and Electrical Engineering," $475,000, five years.

39. Riseman, Hanson, and Kitchen, Co-Principal Investigators, U.S. Army, "Dynamic Image Interpretation for Autonomous Vehicle Navigation," $2,816,514, 10/1/84 - 9/30/88.

40. Stankovic (ECE), Principal Investigator, submitted as a proposed extension to U.S. Army CECOM Contract DAAB07-82-K-J015, "Robust Algorithms for Real-Time Distributed Systems," $26,000, 9 months.

41. Towsley (ECE), Principal Investigator, IBM, "Load Balancing in Shared Memory Multiple Processor Systems," $50,000, 11/1/84 - 10/31/85.

Proposed Equipment Grants

42. Cohen, Lesser, Riseman, and Hanson, Co-Principal Investigators, DARPA, "Request for LISP Machines," $258,000, 8/1/84 - 7/31/88.

43. Riseman and Hanson, Co-Principal Investigators, U.S. Army, "Equipment Proposal for Autonomous Vehicle Navigation," $302,000, 10/1/84 - 9/30/86.

COINS/ECE GRANT FUNDING - SEPTEMBER 1984

Approved

1. Arbib and Manes, Co-Principal Investigators, NSF, "Algebraic Semantics of Programming Languages," $98,440, 1/1/83 - 12/31/84.

2. Arbib and Riseman, Co-Principal Investigators, NSF, "Visuo-Tactile Coordination for Robot Control," $318,218, 1/15/82 - 1/14/85.

3. Barto, Principal Investigator, AFOSR, "Adaptive Problem-Solving with Networks of Goal-Seeking Components," $539,341, 9/1/83 - 8/31/86.

4. Barto and Moore (Psychology), Co-Principal Investigators, NSF, "Adaptive Element Models of Classical Conditioning," $25,880, 3/1/84 - 2/28/86.

5. Clarke, Richardson and Zeil, Co-Principal Investigators, Control Data Corporation, "Development of a Prototype Testing System," $100,165, 6/1/84 - 5/31/87 (first yr. = $35,045).

6. Clarke, Richardson and Zeil, Co-Principal Investigators, NSF, "The Development of a Scientific Testing Method," $345,762, 6/1/84 - 5/31/87.

7. Cohen, Principal Investigator, NSF, "Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach," $113,908, 7/1/84 - 6/30/86 (first yr. = $55,285).

8. Croft and Cohen, Co-Principal Investigators, RADC, "Knowledge Representation in Task-Oriented Domains," $82,000, 2/1/84 —> 8 months.

9. Croft and Lesser, Co-Principal Investigators, Digital Equipment Corporation, personnel support for Office Information Systems, $85,000, 10/1/83 - 12/31/84.

10. Cuny, Principal Investigator, IBM Faculty Development Award, $30,000, 7/1/84 - 6/30/85.

11. Cuny, Principal Investigator, ONR, "Execution Mode in Highly Parallel Computation," $149,965, 5/1/84 - 4/30/87 (first yr. = $31,082).

12. Hanson, Principal Investigator, NASA, "NASA Graduate Student Researchers Program," $15,000, 6/1/84 - 5/31/85 (to be renewed annually).

13. Hanson and Riseman, Co-Principal Investigators, AFOSR, "Representation and Control in the Interpretation of Complex Scenes," $548,500, 4/1/83 - 9/30/84. ($254,000, 10/1/84 - 9/30/85; renewal)

14. Kohler (ECE), Principal Investigator, NSF, "Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems," $134,190, 5/1/82 - 10/31/84.

15. Kohler (ECE) and Towsley (ECE), Co-Principal Investigators, Naval Underwater Systems Center, "A Study of Deadlock Detection in Distributed Databases," $9,930, 7/1/84 - 12/31/84.

16. Kulikowski, Principal Investigator, DEC, $23,000, 11/1/83 - 10/31/84.

17. Kulikowski, Principal Investigator, NSF, "Self-Organizing Knowledge-Based Production of Small Languages from Few User Imputs," $45,000, 10/1/83 - 9/30/84.

18. Kulikowski, Principal Investigator, Raytheon Corporation, support for Speakeasy Project, $5,000, 7/1/84 - 12/31/84.

19. Lehnert, Principal Investigator, NSF and AMOCO Production Company, "Presidential Young Investigator Award," $45,000, 6/1/84 - 5/31/85.

20. Lehnert, Principal Investigator, NSF, "Information Representations for Text Summarization," $192,983, 3/1/83 - 2/28/85.

21. Lehnert and McDonald, Co-Principal Investigators, RADC, "Natural Language Used for Knowledge Acquisition and Explanation," $70,000, 2/1/84 --> 8 mos.

22a. Lesser, Principal Investigator, DARPA, "Coordination in Distributed Problem-Solving," $379,945, 10/1/82 - 9/30/85.

22b. Lesser, Principal Investigator, NSF, "Coordination in Distributed Problem Solving," (with DARPA), $366,000, 5/15/83 - 5/14/86.

23. Lesser, Principal Investigator, RADC, "Plan Recognition in an Intelligent User Interface Design," $78,000, 2/1/84 --> 8 mos.

24. G.P. Nerbonne, (Comm. Dis.), J. Duffy (Comm. Dis.), and S. Kulikowski, Co-Principal Investigators, U.S. Dept of Education, "Training in Augmentative Communication for Nonspeaking Persons," $252,543, 6/1/83 - 5/30/86.

25. Ramamritham, Principal Investigator, NSF, "Synthesis of Resource Controllers for Distributed Systems," $35,043, 8/1/84 - 7/31/85.

26. Riseman, Principal Investigator, NSF, "A Group Research Facility for Artificial Intelligence, Distributed Computing, and Software Systems (Computer Research)," $360,000, 5/1/84 - 4/30/87 (first yr. = $160,000).

27. Riseman and Hanson, Co-Principal Investigators, DARPA, "Processing Dynamic Images from Camera Motion," $750,000, 6/1/84 - 5/31/87 (first yr. = $240,000).

28. Riseman and Hanson, Co-Principal Investigators, Rome Air Development Corp. (RADC), "Applying the VISIONS System to Interpretation of Aerial Images," $45,000, 8/1/83 - 9/30/84.

29. Riseman, Hanson, and Arbib, Co-Principal Investigators, ONR, "Dynamic Tactile Processing," $70,000, 5/1/84 - 4/30/86.

30. Riseman and Weiss, Co-Principal Investigators, A.C. Nielsen Co., "Feasibility Study for a Passive Audience-Composition Metering System," $28,320, 5/15/84 - 11/14/84.

31. Spinelli, Principal Investigator, AFOSR, "Image Understanding by Adaptive Networks of Goal-Seeking Neurons," $304,959, 5/1/83 - 4/30/86.

32. Stankovic (ECE), Principal Investigator, General Electric, "Non-Traditional Architecture for Distributed Real-Time Systems, "18,000, 9/1/84 - 8/31/85.

33. Stankovic (ECE), Principal Investigator, Naval Underwater Systems Center, "A Study on Deadlock Detection in Distributed Databases," $9,930, 7/1/84 - 12/31/84.

34. Stankovic (ECE), Principal Investigator, Naval Underwater Systems Center, "Load Balancing," $9,970, 3/1/84 - 8/31/84.

35. Stankovic (ECE), . Principal Investigator, U.S. Army CECOM, "Decentralized Control of Scheduling in Distributed Processing Systems," $130,830, 12/15/81 - 2/15/85.

36. Towsley (ECE), Principal Investigator, DEC, "Load Balancing in a Multiple Processor System," $32,000, 6/1/84 - 7/15/85.

37. Towsley (ECE), Principal Investigator, NSF, "On the Analysis and Synthesis of Computer Communication Networks," $154,650, 9/1/83 - 2/28/86.

38. Towsley (ECE), Principal Investigator, NSF, "Performance and Reliability Issues in Distributed Computer Systems," $42,000, 9/1/84 - 8/31/86.

Approved Equipment Grants

A. McDonald and Lesser, Co-Principal Investigators, NSF, "Equipment to Upgrade Computer Support of AI Research," $69,716, 5/15/84 - 5/14/85.

Proposed

39. Arbib, Principal Investigator, DARPA, "A Dynamic-Sensing Approach to Distributed Planning and Control for Robots," $1,836,463, 1/1/85 -12/31/88. (also submitted to NASA, NSF, and National Bureau of Standards).

40. Arbib, Principal Investigator, NIH, "Visuomotor Coordination: Neural Networks and Schemas," $1,071,670, 1/1/85 - 12/31/88. (also submitted to NSF)

41. Arbib, Principal Investigator, NSF, "Schemas for Control of Human Robot and Prosthetic Hands," $933,491, 9/1/84 - 8/31/87 (first yr. = $301,297).

42. Clarke and Wileden, Co-Principal Investigators, DARPA, "Toward Feedback Directed Development of Complex Software Systems," 1/1/85 - 12/31/88. $886,000, (first yr. = $290,000)

43. Clarke and Wileden, Co-Principal Investigators, NSF, "A Systematic Treatment of Interface Control," $238,591, 1/1/85 - 12/31/86 (first yr. = $113,100).

44. Cohen and Lesser, Co-Principal Investigators, DARPA, "Proposal to RADC and DARPA Expert System Technology Program," $1,000,000 (4 yr. total), 10/1/84 (approx. start date)

45. Croft, Principal Investigator, NSF, "An Expert Assistant for a Document Retrieval System," $180,642, 11/1/84 - 10/31/86 (first yr. = $94,330).

46. Cuny, Principal Investigator, NSF, "Presidential Young Investigator's Award," $25,000, 5 years.

47. Kohler (ECE), Principal Investigator, NSF, "Concurrency Control and Recovery Algorithms for Distributed Transaction Processing Systems," $250,000, 2 years.

48. Kohler (ECE) and Towsley (ECE), Co-Principal Investigators, NSF, "Design, Evaluation, and Modeling of Distributed Database Systems," $268,222, submitted in July, 1984, 2 years.

49. Lehnert and McDonald, Co-Principal Investigators, ONR, "English Generation for Narrative Summaries," $92,555, 10/1/85 —> ?

50. Lehnert, McDonald, and Rissland, Co-Principal Investigators, DARPA, "Request for LISP Machines," $172,000, 10/1/84 - 9/30/88.

51. Lehnert, Rissland and McDonald, Co-Principal Investigators, DARPA, "Natural Language Processing in Battle Management," $1,600,000, 10/1/84 - 9/30/88.

52. Lesser, Principal Investigators, GTE Laboratories, "Meta-Level Control Through Situation Analysis," $25,000, 8/1/84 - 7/31/85.

53. Lesser and Croft, Co-Principal Investigators, RADC, "A Knowledge Acquisition, Assistance and Explanation System," $2,225,001, 11/1/84 - 10/31/89 (first yr. = $431,001).

54. Orenstein, Principal Investigator, Army Research Office, "Data Structures and Algorithms for Range Searching," $150,243, 9/1/84 - 8/31/86 (first yr. = $73,961).

55. Orenstein, Principal Investigator, NSF, "Data Structures and Algorithms for Range Searching," $128,139, 6/1/84 - 5/31/86.

56. Ramamritham, Principal Investigators, NSF, "Presidential Young Investigator's Award," $25,000, 5 years.

57. Ramamritham and Stankovic (ECE), Co-Principal Investigators, NSF, "Scheduling in Real-Time Distributed Systems," $293,776, submitted in March, 1984, 2 years.

58. Reynolds, Principal Investigator, Defense Mapping Agency, "The Development of an Expert System for Image Interpretation Tasks, "$374,277, 10/1/84 - 9/30/87.

59. Riseman, Principal Investigator, General Electric Foundation, "The Recruitment and Retention of Doctoral Candidates in Computer Science and Electrical Engineering," $475,000, five years.

60. Riseman, Hanson, and Kitchen, Co-Principal Investigators, U.S. Army, "Dynamic Image Interpretation for Autonomous Vehicle Navigation," $2,816,514, 10/1/84 - 9/30/88.

61. Stankovic (ECE), Principal Investigator, submitted as a proposed extension to U.S. Army CECOM Contract DAAB07-82-K-J015, "Robust Algorithms for Real-Time Distributed Systems," $26,000, 9 months.

62. Towsley (ECE), Principal Investigator, IBM, "Load Balancing in Shared Memory Multiple Processor Systems," $50,000, 11/1/84 - 10/31/85.

63. Woolf, McDonald, and Rissland, Co-Principal Investigators, NSF, "An Intelligent Computer Tutor for Pascal and Plane Geometry," $401,075, 1/1/85 - 12/31/87.

Proposed Equipment Grants

64. Cohen, Lesser, Riseman, and Hanson, Co-Principal Investigators, DARPA, "Request for LISP Machines," $258,000, 8/1/84 - 7/31/88.

65. Riseman and Hanson, Co-Principal Investigators, U.S. Army, "Equipment Proposal for Autonomous Vehicle Navigation," $302,000, 10/1/84 - 9/30/86.

9. RESPONSIBILITIES FOR COMPUTING EQUIPMENT

The proposed multiprocessor facility will be incorporated in to the department Research Computing Facility. The project manager (one of the three principal investigators: Professors Victor Lesser, Krithivasan Ramamritham, and Edward Riseman on a yearly rotating basis) in coordination with RCF will be responsible for the procurement, maintenance, and operation of the equipment. The RCF is directly under the supervision of the department chairman, Professor Edward Riseman. Details of the RCF are discussed in Section X.

CURRENT RESEARCH COMPUTING FACILITY (RCF)

Management

The COINS Department has developed an effective administrative structure in the COINS Research Computing Facility to oversee the common computational needs of all researchers in the environment. It has been in place for four years and encompasses representation and communication between the administrative staff, the faculty, and the user community (which is dominated by research graduate students).

The current management organization includes the chairman of the COINS Department (Professor Edward Riseman) who oversees the entire operations, a faculty laboratory director (Professor David McDonald), a full-time professional laboratory manager (Mr. Arthur Gaylord, who has the equivalent of an M.S. in Computer Science). There is a nine-member faculty/student RCF committee overseeing the lab operation, setting policy, and deciding upon major equipment acquisitions. There is also a graduate student User Council which makes recommendations to the lab manager and the RCF committee.

Staffing

The full-time staff of the RCF currently consists of a laboratory manager, an operations supervisor, two systems programmers, and a secretary/bookkeeper. In addition there are three part-time graduate assistants and three part-time undergraduate assistants. The duties and responsibilities of these staff positions are described below.

The laboratory manager has direct administrative, fiscal, and technical responsibility for the COINS Research Computing Facility. This position reports to the department chairman. The laboratory manager implements the objectives and policy initiatives of the RCF policy committee and is a co-chairperson of that committee. In particular, the laboratory manager is responsible for long-range planning and the graceful evolution of our computing environment in meeting the research needs of our faculty and graduate students. The laboratory manager supervises the operations supervisor on the day-to-day operations of the laboratory, and the systems programmers in the design and implementation of additions and enhancements to the facility's software base. This person also has ultimate responsibility for the research assistants who may be directly supervised by other personnel. The laboratory manager is responsible for purchasing major software and hardware, and the administration of associated maintenance contracts.

The operations supervisor has primary responsibility for the daily operation of the installation's equipment and procedures. This person has responsibility for system libraries, account creation and deletion, accounting procedures and backup. It is this person's responsibility to insure that contracted hardware and software maintenance functions are performed by the vendors and that adequate stocks of consumable supplies are maintained. In addition, this person will develop system software and research tools of a generally applicable nature.

The administrative secretary is responsible for routine administrative duties, bookkeeping, and secretarial duties. A major part of this person's work relates to the effective coordination of resources available to the RCF via the numerous individual faculty research grants and the tracking of equipment purchases through the university administration and the vendors.

The most senior of the system programmers is responsible for general maintenance of the software base and implementation of low-level utilities. This as a non-trivial activity, since there is a non-homogeneous hardware and software base in the network. Thus, there are a number of different operating systems, language processors, and specialized output devices that have to be maintained, along with the interfaces necessary for interconnecting these systems. This maintenance, as well as enhancements such as new device drivers, improved text processors, and additional language support, occupies the time of one of the two systems programmers. While this person is responsible for the overall software base of the RCF, most of this person's actual programming time is spent with the VAX/VMS systems.

The networking of all the various types of processors together in a manner that allows consistent styles of access and functionality is a major task. We are dedicating the second systems programmer and one of the RAs to this area. These people are also responsible for designing and, when necessary, implementing network graphics protocols, and the interfaces to the specialized graphics devices in the network. Of course we attempt to acquire software from other academic units to the degree that it is available and applicable to our needs. The two staff members assigned to networking tasks are assisted with the graphics part of this work by one of the other RAs whose primary responsibilities will be to develop the appropriate graphics software for our workstations and the software for the high- and low-level graphical displays needed by the community as a whole.

The third RA assists in administrative and operational details such as the addition and deletion of user accounts, maintenance of queues, and general assistance to the operations supervisor. A secondary duty of this person is the maintenance of text processing and text formatting software.

Finally, there are half-time undergraduate assistants whose primary responsibilities are to perform routine backup and archiving functions. One of these assistants also helps in the installation of new equipment and communications wiring, and constructs simple electronic equipment such as switches.

Equipment

Currently the major computers which are run by the COINS Research Computer Facility are twelve Digital Equipment Corporation VAX processors, two Symbolics 3600 LISP machines, and several Sun Microsystems workstations; all of which are linked to each other by an Ethernet. In addition there are a number of smaller mini- and microcomputers and special purpose processors.

There are two VAX 11/780s, each of which has four megabytes of main memory and approximately 600 megabytes of online disk storage. These two processors are our primary timesharing computers and have about 100 terminal ports attached to them as well as a magnetic tape drive, a laser graphics printer, a medium speed graphics-capable printer, a digital recording camera, and a digital plotter. In addition there are five specialized graphics or image processors interfaced to these computers. They are a Comtal Vision One/20 image processor, a Grinnell GMR-27 graphics processor, two DEC VS11 image processors, and a VICOM image processor (on loan from DEC).

The other ten VAX computers are VAX 11/750s, each having two megabytes of main memory and, depending upon the type of disk drive attached, either 152 or 484 megabytes of disk storage . The majority of these processors have been dedicated to specific projects of those research groups involved in the proposals through which they were originally obtained.

The two Symbolics 3600s are each equipped with 2.3 megabytes of main memory, a 169MB fixed disk, a high resolution bit-mapped display terminal, and an Ethernet interface.

The SUN Microsystems workstations are Motorola 68010-based processors running a version of Berkeley UNIX. They each have between one and two megabytes of memory, a high resolution bit-mapped graphics display, and an Ethernet interface. There are currently five of these systems installed and two more have been ordered.

The most notable other equipment within the RCF is in the Perceptual Robotics Facility, where there are several small DEC LSI-11 systems dedicated to controlling a General Electric cartesian robot arm equipped with tactile sensors, a Puma arm, and an experimental Salsbury tendon-controlled hand.

Expansion plans for the near future include the addition of significantly more main memory to all the VAX processors, the incorporation of the VAXs into a large VAXcluster, and the purchase of additional LISP machines.

ATTACHMENT 4
OMB 3145-0058
12/31/85

Date of this Report September 14, 1984

METRICS FOR RESEARCH ENVIRONMENT
(Mark unavailable data as N/A)

NOTE: These numbers reflect COINS statistics, only.

DEPARTMENTAL FACTORS

Number of Full Time Faculty 23 (AY84-85)

Number of Technical Support Staff 19 (includes professionals and associated researchers)

Number of Clerical Support Staff 13

Budget

Internal instructional* 23.8 %

Internal research* 3.4 %

External research 72.7 %

Other %

These figures are based on the FY84 budget and 21 COINS Faculty members.

Departmental Budget per Faculty ($/person) $196,431

Capitalization per Faculty** $102,173.91

Space per Faculty

Office 664.86

Laboratory 234.17

Total 899.03 (Fall 1984)

Append faculty curriculum vitae***

Append a list of faculty publications 1977-1982***

Append a list of full time faculty for each year 1972-82, indicating additions and departures.

* As defined by the institution.

** Major equipment only. market value cost basis.

(NOTE: These numbers reflect
COINS statistics, only)

STUDENT QUALITY

Undergraduates - Fall 1984

Number

Lower division 80

Upper division 130

Percentage attending graduate studies 40%

Percentage taking employment 60%

Average salary NA

Average number of offers NA

Graduates - Fall 1984

Number

M.S. (part-time) NA

M.S. (full-time) 46 M.S. Students

Ph.D 59 M.S./Ph.D. Students
76 Ph.D. Students

Average entering test scores(GRE,etc.)

1331 out of 1600

Undergraduate GPA 3.37

Ratio number accepted to number enrolled 1.59

[The GRE, GPA, and ratio figures are based on the incoming class for Sept., 1984, only.]

Number holding Fellowships 5

Research Assistantships
in dept. 77

other dept. 18

Teaching Assistantships 43

Append list of first positions of Ph.D. graduates*

Append list of dissertation titles*

* From beginning of program.

NOTE: All figures on this page represent the entire COINS Dept. (22 during the 12-month period used) and 3 members of the ECE Dept.

EXTERNAL SUPPORT

Append a list of current active grants and contracts

Append a list of pending proposals*

Number of grants 40 Number per faculty 1.60

Number of contracts - Number per faculty -

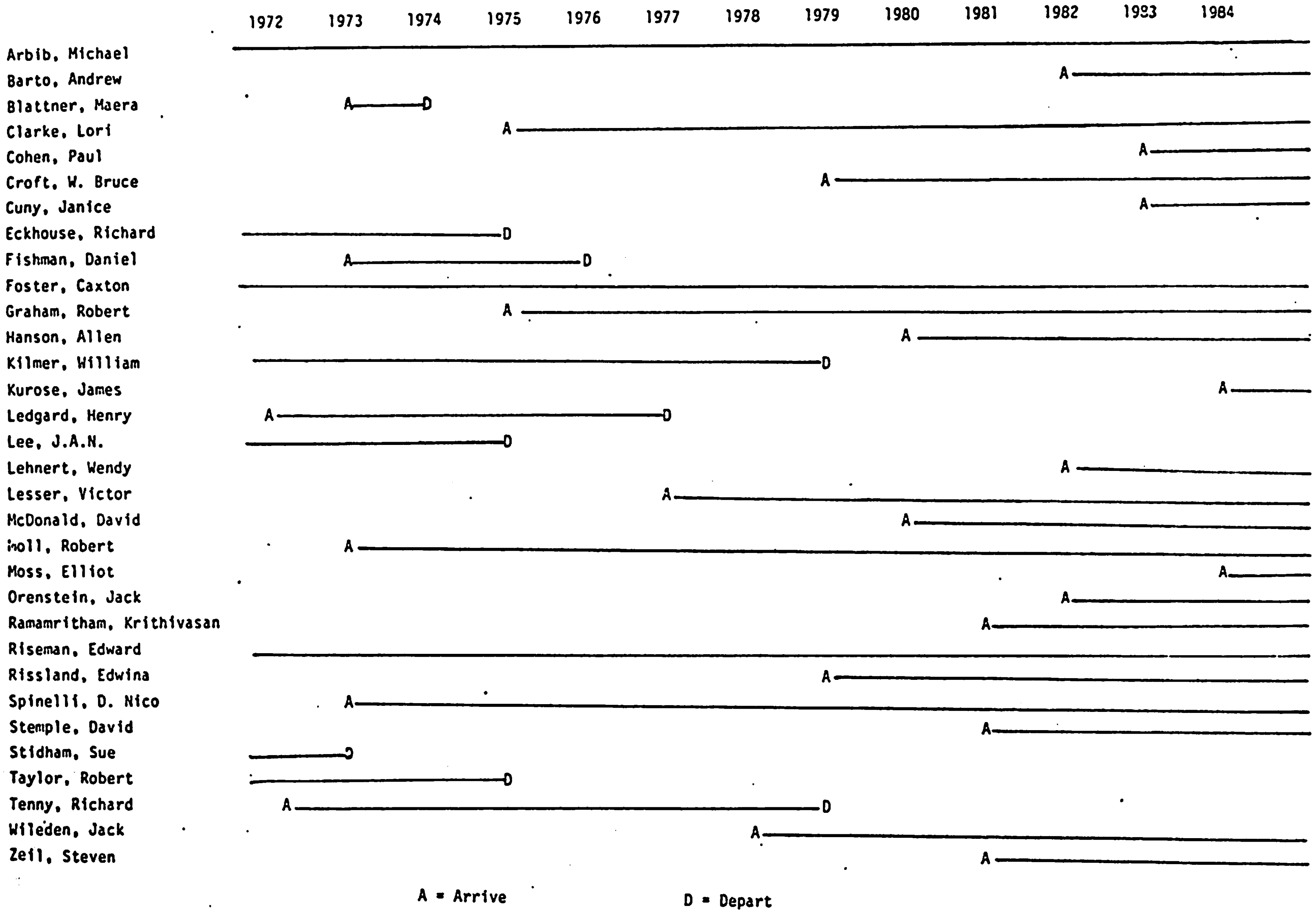
Total annual grant and contract budget $3,090,800

Budget per faculty $123,632

Number of agencies/foundations providing support 9

Number of companies providing support 7

Number of corporate and other gifts NA


1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
Arbib, Michael
Barto, Andrew
Blattner, Maera
Clarke, Lori
Cohen, Paul
Croft, W. Bruce
Cuny, Janice
Eckhouse, Richard
Fishman, Daniel
Foster, Caxton
Graham, Robert
Hanson, Allen
Kilmer, William
Kurose, James
Ledgard, Henry
Lee, J.A.N.
Lehnert, Wendy
Lesser, Victor
McDonald, David
Moll, Robert
Moss, Elliot
Orenstein, Jack
Ramamritham, Krithivasan
Riseman, Edward
Rissland, Edwina
Spinelli, D. Nico
Stemple, David
Stidham, Sue
Taylor, Robert
Tenny, Richard
Wileden, Jack
Zeil, Steven
A
D
A = Arrive
D = Depart

COINS Ph.D. Theses, First Jobs

(2/74) Suad Alagic : Algebraic Aspects of Programming and Formal Languages.

(6/75) Fred Lenherr : Effects of Inter-Ocular Coherence and Audio-Visual Correlations on the Development of Visual Cortex in Kittens.

(6/76) Edward Fisher: The Use of Context in Character Recognition.

(9/76) James Stanley: Network Models of Habituation.

(9/76) Fanya Montalvo: Visual Feature Organization and Interaction as Modelled in Neural Networks.

(9/76) Peter Burt: Stimulus Organizing Processes in Stereopsis and Motion Perception.

(2/77) Stephen Hegner: Applications of Topological Vector Spaces in Linear System Theory.

(2/77) David Stemple: A Database Management Facility and Architecture for the Realization of Data Independence.

(9/77) Frederic Richard: Compiling Technique for Associative Processors.

(9/77) Yuan-Chieh Chow: Queueing Models for Distributed Computer Systems.

(9/78) Elliot Soloway: Mechanisms for Knowledge-Directed Learning.

(9/78) Arthur Karshmer: The Use of Frame-to-Frame Differences in Encoding Computer Graphics Data in a Network Environment.

(5/79) John Prager: Segmentation of Static and Dynamic Scenes.

(5/79) Charles Welty: A Comparison of a Procedural and a Nonprocedural Query Language: Syntactic Metrics and Human Factors.

(9/79) Paul Nagin: Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation.

(9/79) Andrew Singer: Formal Methods and Human Factors in the Design of Interactive Languages.

(2/80) Ruth Maulucci: Kinetics and Optimality in Quadraped Locomotion.

(5/80) John Woods: Path Selection for Symbolic Execution Systems.

(2/81) Thomas Probert: A Model of Renewable Resource Dynamics in Exploited Ecosystems.

(5/81) Alan Morse: Computer Graphics and Modelling: Using Effective Data Displays to Enhance Understanding of Simulation Results.

(5/81) Scott Reed: Using Numerical Search in Managing Ecological Systems: Handling Nonlinearities and Time Delays.

(5/81) Thomas Williams: Computer Interpretation of Dynamic Images from a Vehicle in Motion.

(5/81) Bryant York: Shape Representation in Computer Vision.

(9/81) Balakrishnan Krishnamurthy: Examples of Hard Tautologies and Worst-Case Complexity Results for Propositional Proof Systems.

(9/81) John Tan: An Empirical Approach to Computer Performance Improvement.

(9/81) Debra Richardson: A Partition Analysis Method to Demonstrate Program Reliability.

(2/82) Rolando Lara: Neural Models of the Visuomotor System of Amphibia.

(9/82) Helen Gigley: Neurolinguistically Constrained Simulation of Sentence Comprehension: Integrating Artificial Intelligence and Brain Theory.

(9/82) Jane Hill: A Computational Model of Language Acquisition in the Two-Year-Old.

(9/82) John Lowrance: Dependency-Graph Models of Evidential Support.

(2/83) Dan Corkill: A Framework for Organizational Self-Design in Distributed Problem Solving Networks.

(2/83) Rajendra Wall: The Theory and Use of Scenarios.

(2/83) Richard Brooks: Experiments in Distributed Problem Solving with Iterative Refinement.

(6/83) Jeffrey Conklin: Rapid but Limited Discourse Generation Using Salience.

(9/83) Ralph Kohler: The Utilization of Redundant Information in Image Segmentation.

(2/84) Daryl Lawton: Processing Dynamic Image Sequences from a Moving Sensor.

(2/84) Richard Sutton: Temporal Aspects of Credit-Assignment in Reinforcement Learning.

(6/84) Kenneth Overton: The Acquisition, Processing and Rise of Tactile Sensory Data in Robot Control.

(6/84) Steven Levitan: Parallel Algorithms and Architectures: A programmer's Perspective.

(6/84) Beverly Woolf: Context Sensitive Text Generation in the Implementation of a Machine Tutor.

(9/84) Charles Weems: Image Processing on a Content Addressable Array Parallel Processor.

(9/94) Laura K. Dillon: Analysis of Distributed Systems Using Constrained Expressions.

(9/84) House, Donald: Models of Anuran Depth Perception.

FIRST POSITION OF PH.D STUDENTS

Suad Alagic: Teaching Position, Electro Technology Department, University of Sarajevo, Yugoslavia.

Fred Lenherr: Staff Associate, Center for Systems Neuroscience, University of Massachusetts, Amherst, Masssachusetts.

Edward Fisher: Research and Development Group, Pattern Analysis and Recognition, Inc., Rome, New York.

James Stanley: Post-Doctoral Researcher, Neurobiology and Anatomy Department, University of Texas Health Science, Houston, Texas.

Fanya Montalvo: Industrial Consultant, Computer Science Department, Lawrence Berkeley Lab, University of California, Berkeley, California.

Peter Burt: Post-Doctoral Researcher, Psychology Department, New York University, New York, New York.

Stephen Hegner: Systems Analyst, Lawrence Livermore Lab, University of California, Livermore, California.

David Stemple: Associate Director for Research, Computing Center, University of Massachusetts, Amherst, Massachusetts.

Frederic Richard: French Army.

Yuan-Chieh Chow: Assistant Professor, Computer Science, Wright State University, Dayton, Ohio.

Elliot Soloway: Post-Doctoral Researcher, Computer and Information Science, University of Massachusetts, Amherst, Massachusetts.

Arthur Karshmer: Assistant Professor, Computer Science, New Mexico State University, Las Cruces, New Mexico.

John Prager: IBM Scientific Center, Cambridge, Massachusetts.

Charles Welty: Associate Professor, Department of Computer Science, University of Maine, Portland, Maine.

Paul Nagin: Assistant Professor, Department of Opthamology, Tufts New England Medical Center, Boston, Massachusetts.

Andrew Singer: Vice-President for Research, E _& L Instruments, Derby, Connecticut.

Ruth Maulucci: Post-Doctoral Researcher, Brain Research Lab, American Oncologic Hospital, Philadelphia, Pennsylvania.

John Woods: Consultant, Government Systems, Control Data Corporation, Sunnyvale, California.

Thomas Probert: EDIS/CEAS/MEAD, Washington, D.C.

Alan Morse: Senior Software Engineer, Applicon, Lexington, Massachusetts.

Scott Reed: Owner of a software company, Amherst, Massachusetts.

Thomas Williams: Manager of Manufacturing Automation Project, Digital Equipment Corporation, Maynard, Massachusetts.

Bryant York: Technical Staff Member, IBM Research Laboratory, San Jose, California.

Balakrishnan Krishnamurthy: Research Staff, General Electric, Schenectady, New York.

John Tan: Has not yet accepted a position.

Debra Richardson: Visiting Assistant Professor, Computer and Information Science, University of Massachusetts, Amherst, Massachusetts.

Rolando Lara: Faculty Member, Centro de Investigaciones en Fisiologia Celular, Universidad Nacional Autonoma de Mexico, Mexico City, Mexico.

Helen Gigley: Assistant Professor, University of New Hampshire, Durham, New Hampshire.

Jane Hill: Assistant Professor of Computer Science, Smith College, Northampton, Massachusetts.

John Lowrance: Research Scientist, SRI International, Menlo Park, California.

Daniel Corkill: Post-Doctoral Research Associate, Computer and Information Science Department, University of Massachusetts, Amherst, Massachusetts.

Rajendra Wall: Texas Instruments, Plano, Texas.

Richard Brooks: Teknowledge, Inc. Palo Alto, California.

Jeffrey Conklin: Unit Manager, ATL/STL/ Intelligent Software Camden, New Jersey.

Ralph Kohler: Software Consultant, ISS Corporation, Amherst, Massachusetts.

Daryl Lawton: Post-Doctoral Research Associate, Computer and Information Science Department, University of Massachusetts, at Amherst

Richard Sutton: Post-Doctoral Research Associate, Computer and Information Science Department, University of Massachusetts, at Amherst

Beverly Woolf: Visiting Professor, Computer and Information Science Dept., University of Massachusetts at Amherst.

Steven Levitan: Visiting Professor, Electrical & Computer Engr./COINS University of Massachusetts at Amherst.

Kenneth Overton: Computer Scientist, General Electric Corp., Flexible Automation Systems Program, Computer Rsch. & Devel. Schectady, NY.

Charles Weems: Visiting Professor, Electrical & Computer Engr./COINS University of Massachusetts at Amherst.

Laura Dillon: Visitng Professor, Electrical & Comptuer Engr., University of Massachusetts at Amherst.

Donald House: Assistant Professor, Mathematical Science Dept., Williams College, Williamstown, MA.

Appendicies

1. Overview of Department

Like many of the other leading computer science departments in the United States, COINS has been struggling throughout the last few years to maintain and expand quality research and instructional programs. We have been faced with dramatic enrollment pressures, intense competition (particularly from industry) for advanced graduate students and faculty, and fluctuations in funding due to political uncertainties at both the state and federal levels. Unlike many departments in similar situations, COINS has managed to significantly increase its strength and quality during this period:

Faculty and Students: We have successfully competed for high quality faculty and students;

Research Funding: The scope and funding of our research programs have been expanded substantially and we currently have internationally recognized programs in the areas of AI, Cybernetics, Distributed Computing, Software Development, and Computer Vision;

Research Activities: Collaborative research programs address issues in parallel processing and VLSI, distributed computing, vision, robotics, database systems, integrated software development environments, expert systems, natural language understanding, theory of computation, and computational epistemology.

Computing Facility: A first-rate, although already saturated, research computing facility has been assembled;

Industrial Affiliations: Strong affiliations with industry have been developed;

University Support: Strong university support for the department has solidified with additional faculty positions, space, and a commitment to aid in the development and maintenance of our laboratory.

2.1 Faculty and Students

With the addition of four new faculty positions over the 1983/84 and 1984/84 academic years, each of which we successfully filled despite an intensely competitive job market, COINS has now grown to twenty-two full-time faculty, fourteen research computer scientists and six professional administrative/technical support staff members. The Electrical and Computer Engineering Department has added an additional computer engineering faculty member, bringing their number of computer engineering faculty to eight. We have recently proposed the establishment of the Computer Research Institute (CRI) in order to bring together COINS researchers and those members of the University's Electrical and Computer Engineering (ECE) department who have closely related research interests. The CRI will thus serve to enhance interdepartmental collaborative computer research and will further enrich the COINS research environment. COINS has also established close ties with the Department of Mathematics and all three departments are currently providing coordinating course offerings.

The graduate pipeline has now almost approached steady state. There are presently 181 graduate students in COINS, approximately 60 of whom have already passed their Ph.D. qualifying examinations. This year's new class of 56 graduate students have all been offered full financial support by the department and continues our trend of recent years of attracting very strong graduate candidates. Our recent Ph.D.s have been very successful in the job market, joining the research laboratories of such organizations as DEC, GE, HP, TI, IBM, and SRI as well as several well-known universities and colleges.

The following table outlines COINS graduate degree production since 1979; the figures indicate the number of graduate students enrolled and the number of students receiving advanced degrees as of September of the specified calandar year. The figures for this 1984/85 academic year are projections. Twenty-six graduate students are in the advanced stages of their doctoral research and their advisors believe that it is probable that they will receive their degrees by August 1985. However, our projections in the past have usually slightly underestimated the finishing times for dissertations close to completion, so we have adjusted our projection figures accordingly. Of these 26 advanced PhD candidates, 11 will have certainly completed their dissertation defense and graduated, an additional eight are expected and likely to have done so, and eight additional completions are possible. We have excluded this latter group from the PhD degree statistics below.

	1979	1980	1981	1982	1983	1984	1985
Total Graduate Enrollment	96	96	115	126	143	181	200
M.S. Degrees Granted	24	17	29	12	21	32	42

Ph.D. Degrees Granted	4	2	8	4	5	8	18

It should be noted here that *graduate student researchers account for the majority of the usage of our current research computer facility.* Obviously the presence of a first-rate computing facility will allow us to continue to attract outstanding students and make it possible for them to perform their doctoral research in a reasonable way.

2.2 Summary of Research Funding

The department has steadily expanded both the scope of its research program and its external research funding which comes from a diverse set of agencies summarized in the table below. The 22 faculty members in COINS and 3 ECE faculty members associated with this proposal have 40 grants which are active as of Sept. 1, 1984. The total funding of these grants is almost six million dollars ($5,945,826). We selected the 12 month period beginning Oct. 1, 1983 to compute the pro-rated yearly funding rate which comes to 3.09 million dollars per year. Much of the research support is in the form of relatively small grants obtained through the competitive, peer-review process, including 11 NSF grants. We are also receiving substantial research support through 8 grants via our industrial affiliations as noted in section 2.3. The grant figures below indicate the number of grants held by these faculty members.

Table I. Tabulation of Grant Agencies
(for Grants Active as of 9/1/84)

Agency	*Number*
NSF	15
DOD	15
DARPA	2
ONR	2
ARMY CECOM	1
AFOSR	3
RADC	4
Naval Underwater Systems Center	3
NASA	1
US Dept. of Ed.	1
Industry	8
Total	40

2.3 Research Activities

The most remarkable characteristic of the research programs in the Department of Computer and Information Science (COINS) is that they extend across category boundaries to simultaneously address issues in three areas: Artificial Intelligence and Brain Theory, Systems, and the Theory of Computation. This interaction is augmented by increasing collaboration with the faculty of the Department of Electrical and Computer Engineering (ECE), and with several other departments including those of Mathematics, Linguistics, and Philosophy. It is our hope that the following summary[1] gives an indication not only of the richness of our research, but also of the exciting pattern of interactions between the research groups.

As the sophistication of computers and their interconnections increases so does the need for software to drive these machines. Computer technology is moving toward scheduling access to multiple resources for multiple users, thus raising complex issues in software maintenance and testing, and data base protection. We must also consider the interplay of diverse users as they enter the many nodes of a network, sometimes seeking the same information, and at other times acting discordantly or in competition for scarce resources. Our work on Distributed Computing includes exploration of these issues.

Powerful software development environments are urgently needed if there is to be any hope of fulfilling the increasing demand for complex software systems. Research in "Software Development" and "Software Testing" lies at the very core of our increasing concern with producing reliable software. Our work in these areas provides tools for integrated software development environments, incorporating support for precise specification of interface control and modularity, and sophisticated software testing methodology.

The ever-increasing complexity of computing systems suggests that more and more information will be stored in them. Typical database systems are large and complicated, difficult to design and to implement efficiently. The goal of our research into "Database Systems" is to simplify the building of database systems, to improve their performance, and to expand the range of of their applications. The work on database query and system specification deals with human factors and verification of correctness. Other research is directed to improving the performance of database systems.

[1] This summary is extracted from the Department's research brochure.

The work in "Intelligent User Interfaces and Office Automation" suggests some of the interactions among research topics that characterize our department. We have a network of ten VAX 11/750's that can be used as "workstations for the Office of the Future". Part of our research with such networks explores "Cooperative Distributed Problem Solving," where we ask how a computer network can unify the people in an organization into an effective problem solving team. This work builds upon earlier work on the HEARSAY speech understanding systems, to develop a general methodology for using the interactions between a set of nodes, each with limited and possibly inaccurate information, to yield a cooperative solution to some global problem. In another direction, but in the same task domain, we are building an adaptive information retrieval system in which the retrieval algorithm adapts differentially over time to histories of queries from specific users.

The intelligent interface work also points ahead to two other areas of our expertise in Artificial Intelligence, namely "Natural Language Processing" and "Knowledge-Based Systems". One of the most exciting developments in information management, and one to which the varied expertise in our department is contributing significantly, is the blurring of the boundaries between the three hitherto separate areas of information retrieval, database management, and knowledge representation. The work in intelligent user interfaces provides one test bed for this integration, and for the use of natural language, both to interpret the informal input from the human, and to provide responses in a rhetorically well-structured form. However, our work in natural language processing and in knowledge-based systems goes well beyond such demands. It has involved us in conceptual sentence analysis, treating summarization as a prototype for selective information retrieval; and in developing techniques for producing textual output, which is used both in the intelligent interface project and in research on a machine tutor.

This development of a machine tutor is an exercise in intelligent computer-assisted instruction, an interesting facet of our work in "Knowledge-Based Systems". In related work, we have been concerned with the way in which access to a domain of knowledge can be facilitated by the choice of appropriate examples. To this end we have considered the construction of examples for intelligent on-line help facilities, for understanding legal reasoning, and for learning systems. A third area of research in knowledge-based systems is on "Computer Aids for the Disabled." This work opens up pathways of communication for the non-vocal severely disabled who have experienced an almost complete loss of communication. We have also contributed to research in expert systems, both by studying the extent to which the use of algorithms in place of heuristics can be made to speed up systems, and by exploring issues in representation of knowledge for these systems. One such development is a logic for reasoning about evidence and uncertainty, designed to facilitate explanation and reasoning about control when the available knowledge is incomplete, inconsistent, and inaccurate.

Our Department is distinguished by a research program in "Brain Theory," directed to visuomotor coordination and neural plasticity. This work extends to adaption in networks designed for artificial intelligence applications along the lines of connectionist learning approaches. One of the striking features of the brain is that it does not have the coordinate addressing scheme of the conventional computer, but rather seems to have a content-addressable memory – partial cues can seek out the necessary associations, without knowing where the relevant information is stored. The group working on "Parallel Architectures and Associative Memory" is explicitly designing VLSI circuitry for the production of content-addressable, highly parallel, computers. And, in one of the interactions that is so typical of our research, the efficacy of this design is being demonstrated by showing that techniques for processing sequences of visual images which can take many minutes on a serial computer can be brought up to frame rate if implemented on a practical and realizable Content Addressable Array Parallel Processor.

The research on "Machine Vision and Image Understanding" shares the orientation towards distributed processing that is so prevalent in our department. They use highly parallel arrays containing large numbers of similar components for low-level vision processing, complemented by a limited number of knowledge sources cooperating in high-level scene understanding. This research thus The VISIONS research group is now beginning to integrate the capabilities of identification of objects in static images with the ability to extract depth maps of the environment from a sequence of images.

Our growing concern for parallelism is motivated by the rising technology of VLSI. It is the availability of cheap microprocessors, together with conceptual developments in machine vision and in our analysis of brain mechanisms for visuomotor coordination, that have led to our approach to "Robotics" research. Here, we see how networks of computers for visual and tactile processing may interact with networks of controllers, each with their own dedicated microprocessor. The goal is the control of articulated robot manipulation tasks under the guidance of continual sensory input.

In computer science in general, and at COINS in particular, the theory of computer science is developed both by abstraction from empirical studies and as a prelude to empirical work. A three-departmental theory group, linking COINS faculty with colleagues in the departments of Mathematics and Electrical and Computer Engineering (ECE) are examining a wide variety of computing models.

2.4 Industrial Affiliations

COINS has been developing research relationships with DEC in the areas of industrial robotics, profession-based systems, distributed computing, and graphics. The most significant project involves a $1,250K donation of equipment and $200K of research support by DEC for intelligent user interfaces for profession-based systems. This is a major cooperative research effort between COINS and DEC, integrating approaches from expert systems, natural language processing, data base systems, and office automation.

Several faculty have developed or are in the process of developing research relationship in the area of expert systems and AI applications with the General Electric Corporation, GTE, TASC (The Analytical Sciences Corporation), Mitre Corporation and AMOCO.

The vision and robotics groups have had an ongoing relationship with DEC in the development and use of tactile sensors and vision algorithms for a variety of algorithms with significant credits towards equipment discounts. The General Electric Corporation recently donated two prototype robot arms and peripheral equipment valued at $150K. Support from DEC and GE have supplemented an NSF grant and made possible the evolution of a serious robotics effort in COINS. The robotics group has also established avenues of cooperation with the Martin-Marietta Corporation. In addition, there has been interest expressed by a number of companies (including GE, DEC, Hughes, Martin Meriatta, and RCA) in developing a research relationship in the area of parallel computer architectures. The vision group also has a long-term relationship with Tufts New England Medical Center and Massachusetts General Hospital in biomedical image processing.

Our graduate students have been awarded IBM Fellowships and last year COINS had an unsolicited research development grant from IBM acknowledging the excellence of our research activities. We are also expanding our ties with CDC in the areas of APL systems, database query languages, and software development environments. Finally, Data General has recently donated an Eagle MV-8000 system and peripherals worth $300K to the COINS and ECE Departments for educational use. This is very important since all of the equipment in our Research Computer Facility is restricted to research purposes.

Our continuing and growing industrial relationship is being supplemented by an Industrial Affiliates Program that is intended to be part of the proposed Computer Research Institute for COINS and the Computer Systems faculty of the ECE Department.

2.5 Institutional Commitment

The computer industry is at the heart of the resurgence of the high technology industry in Massachusetts. The Computer and Information Science (COINS) Department is now receiving the strongest support possible from all levels of the University administration. In the face of extreme institutional competition for a limited pool of flexible funds, resources of all kinds are still being reallocated to computer science and computer engineering. Refer to Appendix 2 the formal institutional statement.

COINS and the computer engineering component of the Electrical and Computer Engineering (ECE) Department are jointly planning their expansion as a cooperative effort in both education and research. The computer systems component of ECE (10 faculty) now resides with the COINS Department (22 faculty) in the same building. The previously mentioned proposed Computer Research Institute that will serve as an institutional focus for the cooperative research effort.

Four new COINS faculty were hired in COINS in the 1983/84 and 1984/85 academic years. The long range COINS plans project a growth to the range of 30-40 faculty, which combined with 15 post-doctoral or visiting researchers, will form a community in the range of 50 Ph.D. researchers in the COINS department. ECE has also hired new computer engineering faculty in the last two years and plans to grow to 10 in the computer area.

During the past two years the salaries of COINS faculty have been substantially increased to bring them more into line with comparable salaries in other academic and industrial organizations. At the same time the standard teaching load for COINS faculty members has been reduced to three courses per year in recognition of the time commitment required by their extensive research activities.

Significant amounts of space are being made available to accommodate the ongoing expansion of computer science and engineering faculty and graduate students by moving academic units out of the building. Approximately 3000 square feet of additional space were provided last year and about 8,000 square feet are in the planning stages for this year; about a quarter of this additional space will be allocated for expansion of the current computing laboratory facilities.

During the past six years the University of Massachusetts has provided several matching grants totalling $117,000 which have been combined with major grants from NSF, DARPA, and DEC, to allow the acquisition of approximately $2.35 million of computer equipment for the COINS research facility.

As further evidence of administrative and state support, an additional $300,000 in funds has been added by the university to the department's support base beginning in the Fall of 1984. These funds have been provided to COINS for faculty salaries, increased teaching assistant stipends and administrative support. As part of this support, the COINS Department has recently been allocated three new professional positions. One of these positions is for a laboratory manager of the Research Computer Facility and another for an additional technical support staff; the final position is for an Administrative Assistant to aid the department in its rapid growth and heavy research commitment. In short, the University is prepared to fully support COINS and ECE in developing an outstanding research environment in computer science and engineering and to maintain this environment after completion of the CER grant.

APPENDIX 2

UNIVERSITY OF MASSACHUSETTS

AMHERST

TO: Whom It May Concern

FROM: Dean, Faculty of Natural Sciences and Mathematics

DATE: September 17, 1984

SUBJECT: University Commitment

As a University commitment to this proposal, we are prepared to allocate funding for the Laboratory Manager of this project and for a number of Research Associates (one the first year, two in each of the next three years and three during the fifth year). The total value of this commitment, including fringe benefits and indirect costs, is roughly $500,000. In addition, the University will provide release time from teaching for the various faculty who will assume directorship responsibilities fpr the project each year. This will amount to an additional contribution of roughly $250,000. Finally, the Vice Chancellor for Research and Graduate Studies has commited $30,000 per year in equipment matching funds.

Furthermore, it is our intention, if this grant is awarded, to renovate a 2000-square-foot room in the Lederle Research Center for the Research Computing Facility, which will house the equipment purchased on this grant, along with existing equipment. Since we are a public insitution, the timing of this renovation will depend somewhat on the State construction regulations which are in force at the time the grant is awarded. We estimate that these renovations will cost roughly $120,000.

More generally, the University has demonstrated in a number of ways its continuing commitment to the Department of Computer and Information Sciences (COINS). In order to facilitate the Department's expansion, the University is in the process of renovating space in the Lederle Research Towers which, when completed, will house the Nuclear Physics group. This group is currently located in 8,000 square feet of space next to our COINS Department. When the Nuclear Physics group occupies its renovated space, our computer people will take over this 8,000-square-foot area. This space is in addition to the space which will be renovated for the Research Computing Facility.

Also, over the past four years, we have doubled the number of faculty in our COINS Department, and as of this present academic year we have successfully obtained from the State Legislature a special appropriation of $300,000 which will permanently increase the Department's base budget. This has enabled us to hire support people, increase faculty salaries and teaching assistant stipends, and also add several more faculty. There is good reason to hope that the Legislature will add several more faculty lines for next year.

F. W. Byron

Frederick W. Byron, Jr.
Dean, Faculty of Natural
Sciences and Mathematics