This report describes many of the computer related research efforts at the University of Rochester. The Department of Computer Science is involved in research in automatic programming, including very high level languages and data structures; machine perception; and in problem solving using combinations of traditional heuristic methods, artificial intelligence, and utility theory. The research of the Department of Electrical Engineering includes basic computer engineering research in the construction of computer systems and operating systems, research in image processing and in numerical methods, and research in production automation which is concerned with mechanical manufacturing and assembly, and is currently developing mathematical models of parts, raw materials and tools. In conjunction with other departments, Electrical Engineering is also using computers for biomedical applications including ultrasound diagnostic techniques for heart disease, and pattern recognition techniques for detection of cancer from PAP smears.
## Research in Computer Science and Computer Engineering

### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A. Artificial Intelligence</strong></td>
<td></td>
</tr>
<tr>
<td>I. Planning and Acting</td>
<td>2</td>
</tr>
<tr>
<td>II. Machine Perception</td>
<td>3</td>
</tr>
<tr>
<td><strong>B. Programming Languages &amp; Systems</strong></td>
<td></td>
</tr>
<tr>
<td>I. Intelligent Terminal System</td>
<td>5</td>
</tr>
<tr>
<td>II. Automatic Programming</td>
<td>7</td>
</tr>
<tr>
<td><strong>C. Computer Engineering</strong></td>
<td></td>
</tr>
<tr>
<td>I. Computer Engineering Research Facility</td>
<td>10</td>
</tr>
<tr>
<td>II. Computer System Modelling</td>
<td>11</td>
</tr>
<tr>
<td><strong>D. Production Automation Project</strong></td>
<td></td>
</tr>
<tr>
<td><strong>E. Applications</strong></td>
<td></td>
</tr>
<tr>
<td>I. Adaptive Modelling and Data Processing</td>
<td>12</td>
</tr>
<tr>
<td>II. Biomedical Applications</td>
<td>13</td>
</tr>
<tr>
<td>III. Digital Image Processing</td>
<td>13</td>
</tr>
</tbody>
</table>
Rochester is a small University and cannot afford to maintain large departments with many duplicated research areas. The Computer Science and Engineering effort (which is being greatly expanded) is spread over a number of departments in several colleges of the University. The compensating advantage of a small University is that the intellectually close efforts can be, and in this case, are cooperative. The research projects described here are primarily focused in the Departments of Electrical Engineering and Computer Science but involve many other departments. The main omission from this report is in the area of Theoretical Computer Science. A theoretical group, involving at least the Departments of Statistics, Computer Science, and Mathematics is currently being developed. The theoretical efforts plus a number of smaller ones will be described in a subsequent report.

The current activities based in the Computer Science Department can be roughly divided into two major subareas: Artificial Intelligence and Programming Systems. The activities focused in Electrical Engineering can be divided into Computer Engineering, Production Automation and Applications (primarily to Medicine).

A. Artificial Intelligence (A.I.)

The current major topics in A.I. include the representation and use of knowledge, problem solving, heuristic search, language and speech, vision and robotics, information processing psychology, and automatic programming.

These are highly interdependent and anyone who is serious about the field must be concerned with all of them as well as with relevant applications and systems efforts. For the next three to five years, we at Rochester plan to make a coordinated attack on three main problem areas: Planning and
Acting, Automatic Programming, and Machine Perception. It is possible that additions to the staff will enable us to work on Natural Language Understanding as well.

1. Planning and Acting

One of the most important problems in A.I. concerns the formulation and execution of plans in an uncertain environment.

a. General Methods

We have felt, for some years, that the heuristic methods of A.I. could be combined with the computational techniques of operations research to solve problems inaccessible to either alone. Our first application(11) was to semantics-based image analysis and forms the basis for task II.a below. More recently we have begun tackling traditional A.I. problems using these techniques(10). One result of this work is a new insight into the relation between heuristics and utilities which we express as "a heuristic is a utility theorem". That is, a sound heuristic is a computation-reducing rule which has a favorable expected utility (including the computation saved). We have found a number of general results (for example, on where in a plan to place tests) which bear a fascinating resemblance to ideas in code optimization.

b. Applications

While much is yet to be learned about domain-independent techniques, we expect most progress in this area to come from the consideration of particular problems.

1. Robotics -- The origin of many of these ideas was the attempt to make devices that would interact intelligently with the real world. Programmable assembly or repair robots will
necessarily incorporate ways of planning and carrying out sequences of observations, thoughts, and actions. Work is currently underway on abstract\(^{(10)}\) and (in cooperation with Stanford) concrete\(^{(5)}\) robot problems.

2. Planning Aides -- Another tractable problem area is programs which assist in planning everyday activities. Consider the problem of travel planning. From the schedules of various modes of transportation, cost, and time utilities, of the traveler and expectations about difficulties, a program could produce a travel plan as good as many we now use. A better program (and one more in keeping with the theory) would have dynamic access to relevant information and be accessible for consultation during travel. Experimental formulations of such planning aides are currently being developed.

3. Other -- The topics described above are among the simplest real problems we can find. As we progress we hope to tackle much harder problems of plan elaboration and execution. Possible problem domains include automatic programming, medical consultation, and operational robots.

II. Machine Perception

This is the least understood major area of A.I. and among the most important.

a. Intelligent Image Processing

Previous work\(^{(11)}\) has demonstrated the feasibility of incorporating domain-dependent knowledge in a general image analysis program. The properties of individual regions and boundaries and of relations
between them were learned for particular domains and incorporated in a decision-theoretic control program. We intend to extend our work to new domains (including medicine) in order to find the limitations of the techniques and hopefully solve some practical problems. The problems of automating or partially automating the processing of images which are derived from a computer-driven microscope present a rich source of real tasks.

SCAN, a Standard Cell Analysis System, is being developed with the Anatomy Department(22). A vidisector is attached to a light microscope, and interfaced with a Nova Computer. Tissue biopsies are examined by the program and organized according to smallest homogeneous areas. These areas are then given a probable classification according to shape and intensity, within the context of surrounding elements and an a priori knowledge of typical tissue structure. As additional area classifications are posited, previous assumptions are reexamined and modified to take into account new information. The shape, size, position, and intensity spread of each cell are retained. Muscle biopsy analysis will be the first application of this technique, but other kinds of cell-oriented input (such as cortical tissue) will also be considered.

b. Perception and Problem Solving

It has been clear for some time that perception is an active process with much of the perceived structure coming from preconceived models. We have been involved for years in studying how perception and problem solving can be combined to achieve a desired goal. Our current plan is to concentrate on visual perception of simple scenes where there are still many untried ideas. We are
beginning to develop some understanding\(^{(10)}\) of how perception problems fit into the general planning and acting framework discussed briefly in Section I. It is possible that this work will carry over to natural language understanding and speech recognition, where similar problems arise\(^{(19)}\).

B. Programming Languages and Systems

I. Intelligent Terminal System

The Computer Science Department of the University of Rochester is engaging in the development of an intelligent terminal system (ITS) based on a network of mini-computers. The ITS effort is intended to provide basic computing support for the department and be a tool for a variety of research activities in addition to its inherent interest as a research and development project. The ITS project has five principal phases of which we are now in the second.

The first step was to formulate basic design goals and to purchase appropriate equipment. The second phase is to develop a kernel operating system to support future developments. The next step will be to develop a simple system which makes it easy for users to communicate with a wide variety of resources. The fourth, and first moderately intelligent, phase is to have the system software buffer the user from the idiosyncracies of the various accessible resources. Finally, we hope to develop really intelligent systems incorporating models of the user and various subsystems.

In the first phase we decided to concentrate on two classes of terminals. We will have a small number of very powerful terminals incorporating a mini-computer and a fast-refresh full video display. There will be a larger number of alphanumeric displays which will be connected to a Data General Eclipse Computer. The Eclipse and the mini-computer terminals will be
connected in a megahertz network. This internal network will be linked (through the Eclipse) to other campus facilities and to the ARPANET. The system is configured so that we can easily add a large processor and file store.

The initial software for the facility will be quite straightforward. The idea is to supply (in ARPA jargon) TELNET, FTP (file transfer) and process-to-process communication for all users and subsystems. The Eclipse will serve as the controller for the system and will also provide an editor, file system, and hard copy facilities. The key to the software design is a very simple kernel consisting mainly of a message switching program. Each user and subsystem (whether local or remote) will be represented within the Eclipse by one or more processes. All communication between processes will be by system-mediated messages. The design is nearly complete and seems to provide great flexibility and safety without excessive overhead costs.

One easy consequence of the kernel design is that a user can communicate simultaneously with several system subprocesses (multi-processing the user). Another feature is the ability to easily experiment with a variety of user interface packages—no other processes should be effected by a change in the user interface. New devices and subsystems should also be easy to add. We expect to have a version of this system running by late summer (1975).

The next step will be to free the user from many concerns currently involved in multi-site computing. To cite a simple example, one should be able to list a file from some IBM system in their editor format and EBCDIC characters as easily as a file on one's own disc pack. This type of facility requires only some system design and the specification of directories and conversion subsystems. The high grade terminals which include mini-computers
will enable us to work on incorporating real-time response constraints as in the ARPANET graphics protocol. The next step will be to incorporate some automatic fielding of control messages from subsystems. This requires that the ITS incorporate models of subsystems and be able to generate responses to some of their messages, while referring others to the user.

As we incorporate more sophisticated models of various resources and subsystems, an ITS can begin to get intelligent. In addition to automatically controlling various processes, the ITS can begin to choose the appropriate method of carrying out a command. Notice here the strong connection to the Automatic Strategy Generation and Automatic Coding problems. The other direction for research is in modelling of individual users. We hope to work along these lines too, but have not formulated any concrete plans (nor staffed) for this effort.

II. Automatic Programming

We believe that great progress can be made by combining well understood techniques from A.I. and compiler theory\(^7\). There are three closely related areas of investigation which we intend to pursue.

a. Very High Level (VHL) Languages

The idea here is to explore the extent to which domain-dependent knowledge and reasoning ability can be embedded in a compiler. Once again, we are studying both individual tasks and general properties of such "smart compilers". The most advanced of our focused efforts is AL\(^5\), a language for assembly-robot programming being developed in cooperation with Stanford. The AL compiler includes a dynamic model of the task environment and sophisticated routines for computing arm motions.
and other real-time code. An even more ambitious attempt to include automatic task sequencing for assembly tasks is underway. This work is closely related to the automatic strategy generation effort discussed in Section A.I. Our second domain-dependent VHL language project is just getting started; it will be an intelligent statistical assistant and is being developed in cooperation with the Statistics and Psychology Departments at Rochester.

We believe that it is possible to construct such very high level languages for many domains of discourse and that rapid progress in automatic programming can be made in this way. The difficulty is, of course, that someone has to write all those smart compilers. A large part of our research effort is aimed at reducing the work involved in constructing VHL languages. One aspect of this effort is automatic coding research, discussed next. Another aspect is a renewed examination of extensible compilers and translator writing systems(8) in light of current problems and techniques.

b. Automatic Coding

In addition to being easy to use, a programming system must yield good solutions. As we move to higher level languages, more of the burden for good programming falls on the compiler. Our work in automatic coding(14) uses program flow analysis, program performance monitoring, characterizations of various operations, heuristic search, and non-linear optimization techniques to attempt
to compile good code. The first problem attacked was the choice of machine level data structures for the SET and LIST constructs of SAIL\(^{(14)}\). We are extending this, first to other high level information structures and then to additional language features.

c. Relational Programming

We believe that many programming tasks, particularly for complex modelling problems, can be made much easier by the use of relational programming. In these schemes, a user programs in terms of operations among his data elements, rather than in terms of operations on specific data structures. We mean "user" here in a broad sense: programs which write other programs are included (example: VHL compilers). Such a "user" formulates plans (i.e. programs) to solve its problems from specific knowledge about a particular task domain. To the extent that such knowledge is represented in a relational model, it seems convenient for the "user" to express plans in terms of relations among elements of the model.

Relational programming is an attractive notion, but the current difficulty is that there is a huge difference in efficiency between data structures which are hand-tailored to fit the problem, and any known general-purpose implementation of relational data structures.

A major aspect of our work of the last ten years on LEAP\(^{(18)}\) and SAIL\(^{(9)}\) has been concerned with binary relations. We have two current efforts along these lines. The first is concerned with (relational) programming in terms of n-ary relations. The first subtask is to extend our automatic coding techniques (II b above) to automatically tailor an efficient implementation of n-ary
relations for a given program. The second is an attempt to develop a unified way of handling relations, pattern matching, structures (as in PL/I), procedures and generators. This effort could eventually lead to the development of a new programming system.

There is a related, but importantly different problem, in the use of large data bases. Instead of choosing data structures for each program, one must choose structures which will be shared by many users and programs. A first project along this line is nearly complete(4). It involves an interactive processor for file definition in the SPIRES information retrieval system. The user is asked to provide answers to carefully selected questions about expected file use and the system selects a structure and associated routines. We plan considerably more work in this area.

C. Computer Engineering

I. Computer Engineering Research Facility

To provide a vehicle for research in computer architecture, operating systems, and related areas, the Department of Electrical Engineering is constructing a medium-scale research computer system. This system consists of four high-speed microprogrammable processors, each with 64-bit data paths, a large main memory, and various peripheral equipment, including a minicomputer which serves as a multiplexor channel.

This system is expected to service a number of research needs. It will allow emulation via microcode of different architectures, making possible the study of unusual machine structures in actual operation. Because it is dedicated to computer systems research, this computer will provide a mechanism
for experiments in operating systems and performance measurement which are normally difficult to perform. It is planned to connect the computer to the very fast network described in Section B.I.

II. Computer System Modelling

Computer system modelling is a relatively new field which uses methods of queuing theory and stochastic processes to derive useful models for computer system and computer-system component performance. Specific results have been obtained on modelling rotating storage devices, such as disks and drums (27), and on modelling memory structures and storage hierarchies (29). Techniques for deriving approximate queuing models are under investigation, and empirical performance measurements at the University Computing Center are planned.

Research on operating systems is directed at the allocation of memories and processors. Memory allocation in parallel processing systems with multiple memories is under investigation by graph theory models. Queuing theory is being used to evaluate different central processor scheduling strategies for multiprogramming and time sharing computer systems (28).

D. Production Automation Project

The Production Automation Project is an applied research project at the University of Rochester conducted in collaboration with industrial partners.

The Project is concerned with theories of mechanical manufacturing, design, and assembly, and with technologies and systems relevant to these topics. Its goals in the 1972-77 period are:

1) the development of mathematical modelling schemes to describe, in a geometrically complete manner, parts, stock (raw materials), assemblies, and the capabilities of particular tools; and
2) the development of algorithms to produce automatically, from mathematical models (of parts, stock, and tools), manufacturing plans and command data for N/C machine tools. Design, implementation, and testing of integrated computer systems which embody such description and planning schemes are an important part of the Project's work.

Thus far the Project's efforts have been directed mainly at modelling problems. A high level Part and Assembly Description Language (PADL) has been developed (16), and an experimental PADL system is expected to enter industrial feasibility testing in 1975. The PADL system constructs mathematical models of parts from part definitions written in PADL. At present these models are used to produce (automatically) mechanical drawings in a variety of styles—dimensioned orthographics, sections, perspectives, and so forth (17). In the Project's second major phase, mathematical part models will become inputs to experimental manufacturing planning systems.

E. Applications

I. Adaptive Modelling and Data Processing

Adaptive modelling and data processing techniques are being developed and applied to a number of problems ranging from undersea acoustic-communication channels to the processing of ultrasonic images being used for the diagnosis of cardiac ailments. These techniques are based on an interesting combination of theories from the fields of optimization, successive approximations, and difference equations with stochastic coefficients. Current work centers on the use of signal design techniques for either enhancing or eliminating Doppler induced effects caused by ship motion, blood flow, and the like.
II. Biomedical Applications (in collaboration with the Medical Center)

a. Automated Cytopathology: An automated prescreening system for cytopathology is under development with current efforts directed to the design and fabrication of an automated slit-scan flow system \(^{(25)}\). In this system, cells flow in solution to a measuring station where a decision is made on each cell as to its pattern class using an on-line PDP-12 computer.

b. Computer Controlled Diagnostic Ultrasound: Research using computers as system components is being directed toward development of ultrasound cine-cardiograms for patient service and clinical investigation \(^{(13)}\). The direct recording of ultrasound video data in digital form, computer reformatting of time-motion records into cine frames, and rapid display on a cathode ray tube are the major steps in this process. The research includes the development of a very rapid display system and the incorporation of compound scanning techniques for better illumination of cardiac structures. Linear and non-linear digital signal processing is being investigated for improving image quality \(^{(21)}\).

III. Digital Image Processing

Digital Image Processing at the University of Rochester provides a unique opportunity to get involved in research combining many diverse specialties \(^{(2)}\). Research is being conducted in which the processing of two-dimensional images is of primary importance. Some active projects include tomographic imaging in nuclear medicine, diagnostic ultrasound in cardiology, computer tracing of dendrites \(^{(6)}\), and automated cytology \(^{(1)}\). Besides projects in biomedical areas, related research in pattern recognition, coding, data compression, and image restoration is also taking place.
REFERENCES


25. Wheeless, L. L., Patten, S. F., and Onderdonk, M. A. "Slit-Scan Cytofluorometry Data Base for Automated Cytopathology," accepted for publication in Acta Cytologica.

