VOL. 1/NO. 3

AUGUST 1976

cscsi/sceio newsletter

AN OCCASIONAL PUBLICATION OF THE CANADIAN SOCIETY FOR COMPUTATIONAL STUDIES OF INTELLIGENCE / SOCIETE CANADIENNE DES ETUDES D'INTELLIGENCE PAR ORDINATEUR

CONTENTS

FRAMED!

EDITORIAL COMMENT

A few months ago the call went out for submissions to the third CSCSI/SCEIO Newsletter to be produced by the University of Toronto AI Group. We were surprised and encouraged by the tremendous response on fairly short notice at a busy time of the year. Contributions covered a vast array of topics such as pattern recognition, computational vision, natural language processing, representation theory, game playing, among others. The submissions were also quite diverse in scope, ranging from abstracts of existing papers through statements of personal research philosophy and including such things as course descriptions, short papers, and summaries of "past glories". Unfortunately, the number of documents sent to us was so great that for reasons of economy we were forced to leave out a longer paper entitled "Notes on the PROLOG Language" from Luis Pereira (Divisao De Informatica, LNEC, Lisbon, Portugal) outlining his research in AI programing languages and theorem proving. We sincerely regret the necessity for this "pruning" and would urge interested readers to contact Dr. Pereira for copies of this document.

Among the contributions we were happily able to print, we especially call your attention to the proposed constitution for CSCSI/SCEIO (prepared by Prof. R.S. Rosenberg of U.B.C.) and accompanying questions and comments. It seems to us that the adoption of a constitution is a matter of some urgency if the CSCSI/SCEIO is continue to grow in an orderly way. We would also ask you to note (i) the conference to be held at U.B.C. in late August; (ii) the request for payment of this year's dues; and (iii) the plea for people to produce the next Newsletter.

We conclude with optimism for the prospects of CSCSI/SCEIO. The interest in last year's Ottawa workshop and this year's Vancouver conference, the number and diversity of contributions to the Newsletter, and the steady growth in membership, all auger well for the future.

Acknowledgements: We would like to thank all those who contributed to the Newsletter. We are especially grateful to the following members of the U. of T. AI group for their help in preparing, typing, formatting and editing the copy: John Mylopoulos, James Allen, Alex Borgida, Phil Cohen, Robin Cohen, Mary Horrigan-Tozer, Hector Levesque, Dick Peacocke, Corot Reason, Nick Roussopoulos, Laszlo Sugar and Harry Wong. Finally many thanks to Jan McCartney for the cover design and other illustrations.

Gord McCalla
Lucio Melli
Ray Perrault
(Assistant Editors)

PROPOSED CONSTITUTION FOR CSCSI/SCEIO

DRAFT VERSION 2 / MAY 15, 1976

ARTICLE I - NAME

This organization shall be called the Canadian Society for Computational Studies of Intelligence/Societe Canadienne des Etudes d'Intelligence par Ordinateur, hereafter referred to as CSCSI/SCEIO.

ARTICLE II - PURPOSE

A. CSCSI/SCEIO is organized and will be operated exclusively for educational and research purposes in the interest area of the Computational Studies of Intelligence and in the furtherance thereof.

B. The Society will promote the interests of professionals by:
   1. Setting up study groups which will investigate and report on relevant major issues.
   2. Organizing both seminar and tutorial meetings.
   3. Publishing the CSCSI/SCEIO newsletter containing information of interest to members.
   4. Forging and maintaining informed links with Government and Industry.
   5. Other appropriate means. No addition may contradict the main purpose stated in Article II.A above.

ARTICLE III - MEMBERSHIP

A. Membership is open to any person upon payment of dues, as determined from time to time by the Executive Committee of CSCSI/SCEIO.

B. Applicants for membership shall submit their applications to the Secretary-Treasurer of CSCSI/SCEIO.

C. The membership of any member of CSCSI/SCEIO will be terminated if he or she:
   1. Fails to pay dues within 30 days of the date on which they are payable;
   2. Resigns his membership in CSCSI/SCEIO, such resignation being effective after 30 days notice to the Secretary-Treasurer.

ARTICLE IV - OFFICERS

A. The governing body of CSCSI/SCEIO shall be the Executive Committee. Its members shall consist of the four officers of CSCSI/SCEIO: the Chairman, the Vice-

Chairman, the Secretary-Treasurer, and the Editor of the CSCSI/SCEIO newsletter.

B. The duties of the Chairman include:
   1. Calling and presiding at meetings of the Executive Committee and of CSCSI/SCEIO;
   2. Appointing all standing and ad hoc committees;
   3. Appointing ad hoc and standing committee chairmen and others as required;
   4. Appointing members to fill elective offices that may become vacant between elections through resignation or inelegibility of an incumbent officer.

C. The duties of the Vice-Chairman include:
   1. Presiding at meetings in the absence of the Chairman;
   2. Assuming the duties of the Chairman in the event of the Chairman's resignation or incapacity;
   3. Assuming any duties delegated by the Chairman.

D. The duties of the Secretary-Treasurer include:
   1. Keeping minutes of business meetings of CSCSI/SCEIO and of the Executive Committee;
   2. Maintaining records and correspondence of CSCSI/SCEIO;
   3. Notifying members of the Executive Committee of the time, place, and agenda of the Committee meetings;
   4. Supervising the financial affairs of CSCSI/SCEIO;
   5. Maintaining and reporting financial records of CSCSI/SCEIO;
   6. Reporting CSCSI/SCEIO finances annually.

E. The duties of the newsletter Editor include:
   1. Publishing a periodic Newsletter to be distributed to the membership;
   2. Maintaining a current list of correspondents (Newsletter reporters) based at local CSCSI/SCEIO centres.

F. The Chairman, Vice-Chairman, Secretary-Treasurer and Newsletter Editor shall be elected by the members of CSCSI/SCEIO to terms of office of two years, beginning June 1 of even-numbered years.

ARTICLE V - ELECTION OF OFFICERS

A. The Chairman shall appoint a nominating committee by November 30 in each odd-numbered year. This committee will nominate at least one candidate for each elective office and at least two candidates for one of the elective offices, and secure acceptance of nominees. The nominating committee shall inform the members of CSCSI/SCEIO of its slate of candidates, and solicit further nominations from the members at that time.

B. Ballots will be mailed first class from and returned to the Chairman not later than February 28 of each even-numbered year; they shall be mailed to all CSCSI/SCEIO voting members. Ballots shall state the last day for return of a voted ballot. This date shall be at least

30 days after the last ballots are mailed. Of the ballots returned, a plurality of votes cast for each office determines the winner of that office.

C. The ballots will be counted and all members of CSCSI/SCEIO shall be informed of election results no later than April 1 of each even-numbered year.

ARTICLE VI - MEETINGS

A. At least one business meeting of CSCSI/SCEIO will be held each year.

ARTICLE VII - AMENDMENTS

A.

1. A resolution by a simple majority of the Executive Committee shall be sufficient to cause a constitutional amendment to be voted on by CSCSI/SCEIO members. An amendment can be proposed to the Executive Committee by any of its own members, or by any member of CSCSI/SCEIO.
2. A petition by 10% of the members shall be sufficient to cause a constitutional amendment to be voted on by CSCSI/SCEIO members. The right to petition shall be independent of any decisions taken in accordance with the above Article VII.A.1.

B. The proposed amendment shall be voted on by the following mail balloting procedure:

1. The ballots shall be mailed out by first-class mail from (and returned to) the chairman. The ballot shall include (i) a copy of the proposed amendment including a specification of the date on which it will become effective; (ii) a copy of the article(s) in the existing bylaws that is (are) being proposed for amendment.
2. Only ballots received by the Chairman postmarked within 30 days after the last ballot was mailed out shall be valid.

C. The amendment shall become effective if a two-thirds majority of the valid ballots approve the proposed amendment.

ARTICLE VIII - DISSOLUTION

In the event of dissolution of CSCSI/SCEIO, all assets of CSCSI/SCEIO will be transferred tc the members.

Ccmments and Questions

1. Should we consider affiliation with CIPS or CSA? Is this even possible?

2. Is it necessary to register as a society or organization federally or provincially, for liability and tax purposes?

3. If dissolution occurs, what happens to the resources and/or debts of the society?

4. Should the annual meeting be held in conjunction with CIPS or at an annual workshop?

5. This constitution is a modification of the bylaws of SIGART. Are there any other factors which should be included?

6. The question of how to ratify or modify it is open. It could be circulated in the Newsletter or amendments solicited with final approval at some general meeting.

7. A French translation must be provided.

Four responses were received:

1. 2 No and 1 Yes. The yes response was received from Prof. G.H. MacEwen of Queen's University, President of the Computer Science Association. His letter has been included in this Newsletter.

2. We must consult a lawyer.

3. This seems to be a serious matter and 2 is necessary.

4. An annual workshop seems appropriate.

6. We should be able to ratify the constitution at the Vancouver meeting, August 25-27, 1976.

7. Do we have any volunteers?

A suggestion for getting the ball rolling:

The Steering Committee should act as the first nominating committee and the procedure of Article V be followed during early summer, so that the new executive would be determined for introduction at the August meeting.

DEPARTMENT OF COMPUTING
AND INFORMATION SCIENCE

Queen's University
Kingston, Canada
K7L 3N6

March 30, 1976.

Prof. R. S. Rosenberg,
Dept. of Computer Science,
University of B.C.,
Vancouver, B.C.
V6T 1W5

Dear Prof. Rosenberg:

I have just seen a memorandum describing the proposed organization of CSCSI/SCEIO. It contains a suggestion that affiliation with CIPS or CSA should be considered.

I thought I might make some comments in support of such a move; in particular, affiliation with CIPS as a special interest group which is the status that CSA enjoys. CIPS encourages special interest groups such as CSA or yourselves since it broadens their scope of interest. On our part, we have the office and organizational facilities of the CIPS secretariat available to us for fee collection, address records, etc. Our members may or may not choose to be members also of CIPS, but if they do membership rates are reduced.

In short, our experience with CIPS has been very good and I encourage you to explore special interest group status with CIPS.

Yours sincerely,

G. H. MacEwen,
President,
Computer Science Association.

GHM/le

PAST EVENTS

Report on the CSCSI/SCEIO Workshop
Ottawa, Ontario, May 28-29, 1975

John Mylopoulos
Dept. of Computer Science
University of Toronto

The second CSCSI/SCEIO Workshop was attended by 77 computer scientists, psychologists, engineers, mathematicians, medical doctors, etc. There was some participation from most large Canadian universities as well as a few non-Canadian ones (notably MIT), from government agencies (NRC, DRB, etc.) and industry (Bell Northern, IBM, etc.).

The Workshop program included three sessions during which 30 papers were presented on subjects ranging from pure A.I. to applications of A.I., Cognitive Psychology, and Pattern Recognition. The keynote address was delivered by Dr. P. Will, IBM Yorktown Heights, and it was followed by a panel discussion on "Canadian Needs and Artificial Intelligence".

A summary of the sessions is given below:

<u>Session I: AI and Computer Science</u>

1. J. ALLEN, Dept. of Computer Science, U. of Toronto, "Speech Understanding Research at the University of Toronto".

2. J. DE KLEER, MIT AI Laboratory "Electronic Circuits: A Domain for Exploring Debugging".

3. A. MACKWORTH, Dept. of Computer Science, U. of British Columbia, "An Exploration of Algorithms that Manipulate and Satisfy Networks of Binary Relations".

4. T.A. MARSLAND, Dept. of Computing Science, U. of Alberta, "Chess Playing Program Design".

5. L.A. MAZLACK, Dept. of Computing and Information Science, U. of Guelph, "A Case for Considering Non-Human Intelligence Modelling".

6. L. MELLI, Dept. of Computer Science, U. of Toronto, "The 1.PAK Experience".

7. C. RICH, MIT AI Laboratory, "Understanding Lisp Programs: Towards a Programming Apprentice".

8. L.K. SCHUBERT, Dept. of Computing Science, U. of Alberta, "Some Reflections on AI Languages by a Would-Be User".

9. D. SKUCE, Dept. of Electrical Engineering, McGill University, "Should Dynamic Verbs be Modelled by Programs?"

10. R. WOODHAM, MIT AI Laboratory, "Machine Vision and Productivity Technology; Making Machines more Useful".

Session II: AI and Cognitive Science

1. B. FUNT, Dept. of Computer Science, U. of British Columbia, "Preliminary Remarks on the Nature and Use of Direct Representations".

2. S. BECKWITH, Dept. of Music, York University, "Climbing the Music/Language Interface".

3. J. MYLOPOULOS, Dept. of Computer Science, U. of Toronto, "Nine Questions About Understanding Data Base Management Systems".

4. J. EDWARDS, Dept. of Psychology, York University, "Observations on the Use of Language Understanding for Psychological Analysis".

5. P. HERZBERG, Dept. of Psychology, York University, "Some Experiments with Production Systems".

6. J. GASCON, Dept. of Psychology, U. de Montreal, "A Program for Understanding Seriation Protocols".

7. I. McMASTER, Dept. of Computing Science, U. of Alberta, "A Proposal for Computer Acquisition of Natural Language".

8. G. McCALLA, Dept. of Computer Science, U. of British Columbia, "A Preliminary Report on a Frame System for Natural Language Dialogue".

Session III: Pattern Recognition

1. W.W. ARMSTRONG, U. de Montreal, "Use of Boolean Tree Functions to Perform High-Speed Pattern Classification and Related Tasks".

2. M. COHEN and G.T. TOUSSAINT, McGill University, "An Improved Algorithm for Detecting Noisy Lines in Noisy Pictures".

3. I. HENDERSON, Defence Research Establishment, "Military Applications of Image Processing"

4. S.H.Y. HUNG, National Research Council, "A Simple Method of Picture Data Compression".

5. Z.D. KALENSKY, Dept. of the Environment, "Forest Maps from LANDSAT Computer Compatible Tapes".

6. R.D. PEACOCKE, Dept. of Computer Science, U. of Toronto, "A Picture Processing Model".

7. M. SHAKER SABRI and W. STEENART, U. of Ottawa, "Digital Filter Realizations Based Upon the Discrete Hilbert Transform".

8. I. TOMEK, U. of Alberta and Acadia University, "A Generalization of Nearest Neighbour Classifaction".

It is hoped that the CSCSI/SCEIO Workshop will develop into a regular annual or bi-annual conference that brings together researchers from across Canada to exchange experiences and ideas.

The Thrill of Victory, The Agony of Defeat

Lawrence Mazlack
Dept. of Computer Science
University of Guelph

The first North American Go-Moku tournament was held on November 29. The competition used the University of Guelph (Guelph, Ontario, Canada) as a central clearing house for telephone relaying of moves.

Four teams competed. The initial response was twelve teams, however the non-availability of telephone support money, the Canadian postal strike (the tournament was held on the fortieth day of the strike), and programming problems steadily reduced the field (two were even lost on the tournament day).

The final standings were as follows (the number one program beat everyone, the number two program beat everyone but number one, etc.):

Program Name	Programmer(s)	Computing Power	Number Plies
ARTHUR	Mike Compton Apt. 2105 47 Thorncliff Pk. Dr. Toronto, Ont.	IBM 370/168 using VM 1500 PL/I statements	2-7
	Edward Johnston Arthur Coston Psychometrics Lab. U. of North Carolina Chapel Hill, N.C.	PDP-11/45 4K	0

FIVE-IN-A-ROW	Henry Baird 283A Franklin Ave. Princeton, N.J. 08540	PDP-8 4K	0
ANDY	Andy Sulkowski 12 Joycelyn Dr. Streetsville, Ont.	370/155 using TSO in 78K	3

The third ranked Canadian Gc player (Shein Wang) played ARTHUR and FIVE-IN-A-ROW and found that the programs played well. Nc attempt was made to place the programs on the human rank scale.

NEXT YEAR'S TOURNAMENT

Most of those who dropped out of this year's endeavor urged that a tournament be attempted next year and promised that they would enter. Considering this as well as the enjoyment this year's participants had, we decided to try it again. The dates are November 27-28, 1976. Please contact

Lawrence J. Mazlack
Computer Science
University of Guelph
Guelph, Ontario, Canada

for entry forms. If you want to, your program will be run for ycu at U. Guelph, providing it either fits on a version of a PDP-8, or will run in 125K under TSO. Additional local arrangements might be made.

Student participation is particularly encouraged. The Guelph program will be selected from among those written for an A.I. course.

There is a possibility of telephone support money.

A human Go-Moku expert will provide approximate human scale rankings cf the programs.

The rules are:

GO-MOKU RULES FOR COMPUTER PLAY

1.0 Each program will be limited to 15 minutes of clock time. Use of more than 15 minutes will result in game loss.

2.0 Each round will consist cf every program playing every other program twice. Each program will have the opportunity to move first.

3.0 The board will consist of a 19x19 matrix.

4.0 Move position markers consist of black and white "stones". Black moves first and may begin at any board position.

5.0 Once a stone is placed on the board, it remains in the same location until the game is terminated.

6.0 The board reference coordinates are alphabetic horizontally and numeric vertically. Coordinate A-1 occurs in the lower left hand corner of the board. The alphabetic coordinates excludes the letter I and includes the letter O ("oh").

7.0 Illegal moves.

7.1 If a player makes an illegal move and it is recognized as such by an opposing program, the opposing program wins.

7.2 If an illegal move is not recognized by an opposing computer program, play will proceed from that point. The move will be considered to be legal immediately after the second player fails to recognize its illegality.

7.3 If an unrecognized illegal move causes one stone to be placed upon another stone, the first stone will be removed from the board.

8.0 Winning.

8.1 A winning position requires five and no more than five stones of the same colour immediately adjacent in a row (horizontal, vertical, or diagonal).

8.2 A row constituting more than five adjacent stones of the same colour is not a winning position.

8.3 All winning positions must be demonstrated by playing to game end.

9.0 Claiming a win

9.1 A player does not win unless the player claims a win. The first player to validly claim a win for his own position wins the game.

9.2 If a player fails to claim a win, he may do so at a later time, providing he has a winning position at that time and his opponent has not validly developed and claimed a winning position.

9.3 If a player fails to claim a win, his opponent may

9.3.1 proceed with his own game, ignoring his opponent's winning position.

9.3.2 point out the winning situation and claim a draw.

9.4 Anyone falsely claiming a winning position for anyone loses the game. False win claims will be judged by a human referee. Assertations of a false win claim may be made either by a computer program or by the program's human representative within 15 minutes of a victory claim.

10.0 Program parameters may not be reset during the tournament.

The longest game played occurred between ARTHUR and FIVE-IN-A-ROW. The game follows.

	black	white		black	white
1.	K10	J9	21.	K7	E5
2.	K11	K9	22.	H8	G4
3.	L9	J11	23.	N8	G5
4.	L10	J10	24.	G6	H6
5.	J8	J12	25.	O7	M9
6.	J13	L8	26.	D12	E11
7.	M7	H11	27.	O10	H10
8.	G12	K13	28.	N7	L7
9.	G10	G11	29.	P7	Q7
10.	F11	H9	30.	O6	O9
11.	H10	L14	31.	M8	
12.	M15	F9		black claims win	
13.	G9	E10			
14.	M10	G8			
15.	D11	F7			
16.	E6	H7			
17.	J6	J7			
18.	E7	F8			
19.	F10	F6			
20.	F5	G7			

ANNOUNCEMENTS

CSCSI/SCEIO SUMMER CONFERENCE 1976
Vancouver, British Columbia,
August 25-27, 1976

The conference will be held at the University of British Columbia and will feature submitted papers, tutorial talks and informal sessions. The conference fee of $40.00 ($15.00 for students) will enable participants to attend the sessions and to receive the pre-printed conference proceedings.

Accommodation

The University is able to provide accommodation for participants and their families.

Agenda

- Natural Language Understanding (Text and Speech)
- Heuristic Problem Solving and Game Playing
- Automatic Programming and Debugging
- Computer Perception
- Psychological Aspects of A.I.
- Automatic Theorem-Proving
- Knowledge-Based Learning Systems
- Representation of Knowledge
- Applications of A.I.
- Robots
- Social Consequences of A.I.

For further details contact:

General Chairman
Dr. Richard S. Rosenberg
Dept. of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1W5

Programme Chairman
Dr. Alan K. Mackworth
Dept. of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1W5

MONEY . . .

We remind all CSCSI/SCEIO members and prospective members that annual dues of $3.00 shoud be paid at the Vancouver meeting or sent to

Prof. A.K. Dewdney,
Dept. of Computer Science
The University of Western Ontario
London, Ontario.

COMPUTER-CHESS WORKSHOP

The 1976 Canadian Computer-chess event takes the form of a workshop held at the University of Alberta June 26-27, under the direction of Dr. Tony Marsland of the Computing Science Department. A number of installation independent programs have been brought together to execute primarily on either an Amdahl 470V6 or a Nanodata QM-1. In addition to the customary 4 round tournament, the relative efficiency of the programs is measured, and discussed in the working sessions.

The programs Tree Frog, Coko, Chaos, and Wita are implemented on the Amdahl, while Chute is being converted from its OS implementation to MTS. The QM-1 is used in a Nova emulation to support Ostrich and Shrdlu. As of this writing (early June), the following people are scheduled to present their ideas at the workshop: L Kessler (U. of Manitoba), M. Valenti (U. of Toronto), R. Crook and R. Hansen (U. of Waterloo), K. Thompson (Berkeley), G. Courtois (U. of Colorado), T. Marsland (U. of Alberta), and S. Soule (U. of Calgary).

ANNOUNCEMENT

Any places open (e.g. research, teaching, etc.) are sought for a fairly bright young fellow whose NRC grant and D.Phil. will be finished this fall, in any aspect of computer visual perception (e.g. scene analysis, biomedical images, satellite images,...). B.Eng. and M.Eng. in electrical engineering from McGill. Previous work involved getting a computer to find outlines of lungs in front and side view X-ray images. Current work is summarized elsewhere in this issue ("Seeing Puppets Quickly"). Address: J.L.Paul, Arts Postgrad PH, University of Sussex, Falmer, Brighton, BN1 9QN, England.

THE NEXT NEWSLETTER

The acting editor-in-chief of the CSCSI/SCEIO Newsletter (Alan Mackworth) is asking for volunteers to produce the next Newsletter in about a year's time. The first three Newsletters were produced by UBC, U. cf Western Ontario, and U. of Toronto. Anybody else interested?

CURRENT RESEARCH

Z. Kalensky
Forest Management Institute
Canadian Forestry Service

My research activity for the past few years has been centered on the applicability of satellite multispectral images and their computerized processing to forest mapping. The study has been completed and the results summarized in three reports. The first one (Kalensky and Sayn-Wittgenstein, 1974) describes computerized forest mapping from Landsat (Earth Resources Satellite) multispectral imagery recorded at single date. The second report (Kalensky, 1974) describes computerized processing of Landsat imagery recorded at three dates. The third report (Kalensky and Scherk, 1975) concludes the study with the accuracy analyses of all the classification procedures used. A new accuracy measure which takes into account not only the difference in size of population but also its spatial displacement and thus important whenever the spatial delineation of classes is required, was introduced in this report.

Upon completion of the above study a new one was started applying the computerized image processing to digitized color aerial photographs. Considering the complexity of the human decision-making process involved in photo interpretation, a hybrid approach was selected combining the computerized scene segmentation with the identification of classes by visual interpretation techniques.

Ed. Note: Abstracts for the mentioned references are included in the Abstracts section of this newsletter.

University of Alberta

For the past several years, a small group of M.Sc. students under Sampson's supervision has been working on aspects of a computer system for natural language acquisition. An abstract of the latest report on this work, as presented at the June-July COLING meeting in Ottawa, is included elsewhere in this newsletter.

Together with graduate students under his supervision, Schubert is working on problems of memory organization in semantic networks, use of stereo vision in scene analysis, and decision making in problem solving. He is also completing personal projects in the areas of portrait generation and probability induction. In the former area, a quantitative statistical semantics of facial features is being programmed to allow generation of portraits from descriptive phrases such as "face wide, nose long, expression amiable." In his work on probability induction Schubert has developed the concept of a "sequence predictor" and shown how it provides a theoretical link between "inductive inference" theories for infinite sequences and theories of randomness.

Wilson is writing a book to be titled <u>From Associations to Structure</u>, which attempts to apply some of the concepts developed in AI to problems of cognition. The work is an attempt to show the compatibility of AI approaches with associationistic concepts.

Building a Dialogue System

G. McCalla
Dept. of Computer Science
Univ. of British Columbia

My Ph.D. research (currently being completed at UBC) has involved the construction of a computer model that takes part in conversations which might typically arise at a symphony concert. The area of conversation has been chosen because it necessitates general solutions to general problems, solutions which might not be considered when looking at a more restricted topic. The "concert" domain allows both task-oriented and non-task-oriented dialogues in the same basic context, thus revealing interesting features of each dialogue type.

To keep the model within both combinatorial and complexity barriers, some formalism is needed that combines flexibility and power with modularity. Minsky's frame proposal suggests a promising approach; consequently, I have adopted the "frame" (my version at least!) as the fundamental conceptual unit of the model. A computational description of a frame has been worked out and implemented, defining what it means to create a frame, to communicate with a frame, to modify a frame, to access data in a frame, and so on.

Three conversational situations in the "concert" domain have been isolated to form the basis for experiment. In dialogue 1, the model "talks" to a ticket seller in an attempt to "buy" tickets to the concert; in dialogue 2, the model uses much the same linguistic knowledge to "purchase" a drink from a bartender at intermission; and in dialogue 3 (also at intermission), the model "talks" with a friend about the first half of the concert. To successfully undertake these conversations, the model must exercise many talents that include achieving script-based tasks, engaging in non-scripted behaviour, scanning recently acquired knowledge, employing more or less the same frames in different contexts, and so on.

The current state of research sees dialogues 1 and 2 analyzed fairly completely in terms of their frame interactions with dialogue 3 somewhat less fully detailed. None of the conversations has yet been implemented (something I hope to remedy as soon as possible), but the frame analysis alone has led to many interesting observations including, among others, the delineation of several useful kinds of context surrounding a frame; the representation of linguistic and world knowledge in the same formalism; the fuzzing of the boundaries separating syntax, semantics, and pragmatics; the capability for semantic and episodic memory; and the usefulness of a model of the conversant.

Of course, there are still many difficulties, especially in the size of the frames and the occasional awkwardness of some of the constructs. More experience with the system should enable smoothing of a few of these rough edges. Naturally, too, the analysis is quite cursory in many areas: the model can often be fooled easily and would need much more information to begin to approach human levels of performance even in the rather restricted "concert" domain. Hopefully, the methods used here are of sufficient generality to allow such an expansion of knowledge to one day take place. So, although there is no immediate danger of the model *actually* going to a concert, buying tickets, getting a drink at intermission, or talking to a friend, it may at least have provided some insights into the computational structures necessary for dialogue.

News from the AI Group at Lisbon (May 1976)

Luis M. Pereira
Divisao De Informatica
LNEC, Lisbon, Portugal

Our concern has been mainly the construction of a compiler for the PROLOG (predicate logic) programming language into DEC-10 (PDP-10) assembly language (MACRO-10), written in PROLOG itself (for which there is a FORTRAN_IV interpreter). This project has been carried out in collaboration with the department of AI at Edinburgh University. We believe that the use of PROLOG (which has been increasing and widespreading) will be greatly enhanced by the availability of this compiler.

Another project, now starting, concerns the application of PROLOG to a question answering and question asking program which will advise DEC-10 users on how to use the DEC-10 system and also help them to debug their programs. Input will be in English and one goal is to have the program accept advice from experts, on how to advise.

One other area of research has been that of transformational linguistics using PROLOG. Our published work includes a PROLOG manual in Portuguese and "GEOM: a PROLOG geometry theorem prover" by H. Coelho and L. Pereira, March/76, available on request.

We would like to ask all AI groups to include us on their mailing lists. All correspondence should be sent to:

Dr. Luis M. Pereira
Divisao De Informatica
LNEC
AV Brasil
Lisbon-5, Portugal

Some Problems In and Approaches to Creating A Scientific Knowledge Base

Doug Skuce
Montreal Neurological Institute
McGill University

It is my feeling that the design of a scientific knowledge base is a worthwhile yet relatively unexplored area of AI research. In particular, medical and biological science could greatly benefit from some increased formalization.

Unfortunately, the bulk of this knowledge is expressed in natural language (NL), a medium poorly suited for the requirements of precision, brevity, and machine compatibility. My main objective then is to develop a medium of communication sufficiently like NL to be understandable to biomedical personnel, yet also possessing a clear semantic foundation suitable for machine "understanding" as well. Mathematically inclined persons lacking experience in an actual biomedical research environment often fail to appreciate the importance of the former requirement.

LESK (Language for Expressing Scientific Knowledge) is my developing approximation to such a medium. The emphasis is placed, at present, not on writing bits of programs, or on giving the illusion of thoroughness by exhibiting pages of BNF syntax, but on discovering and explicating the necessary underlying semantic constructs. This is done by considering two real micro-domains of neurology: the synapse (the communicating contact between nerve cells), and the spinal reflex arc. Interaction with live neuroscientists has been stressed.

In the first case, a composite description of the essential facts about synapses was solicited from seven scientists. I then have undertaken to translate these facts into LESK. For the reflex arc, one scientist was briefly familiarized with the basic concepts of LESK, and she and I have jointly developed a LESK description, I learning neurology while she learned LESK. It is important to note that in the first case, the scientists may verify that translation, whereas had I embalmed it in LISP, this would be impossible.

The semantic basis relies on several principles:

1) A hierarchy of semantic categories is presented, as precisely and comprehensively as possible. When expressing a fact, one is obligated to fit its components into the appropriate categories.

2) These categories may only relate to each other in certain prescribed ways, which are very general rules called semantic axioms. Any fact must then be a case of some axiom. Example: ONLY THINGS, GROUPS, AND PROCESSES HAVE STATES. A major aspect of my work then is to induct what are these axioms.

3) Most of biomedical knowledge has two aspects: static vs dynamic. Only the latter describes change. Usually one is interested in changes of real-valued quantities, such as voltages, frequencies and amounts of fluids. However, AI has almost neglected continuous change, and hence I have looked to the simulation literature for ideas on this. Since the currently fashionable tools for static knowledge and discrete change description found in AI include semantic nets, pattern-oriented languages, and something called f-----, I have attempted to synthesize these notions with the more analog/simulation oriented ones.

The resulting hybrid candidate for the basic "knowledge chunk" I call a relator. In addition to the above considerations, the relator has been considerably influenced by the notions of abstract data structures, parallel coupled processors, and program hierarchies a la Dijkstra.

There are five main types of relator:

1.	SIMPLE	These have one main argument, usually a THING. Often there are other "slots" which must be instantiated for each instance of the main argument (Skolem functions).
2.	GROUPS	These are n-tuples of concepts having a joint name, corresponding to a record in, say, PASCAL. They are noun phrases in NL.
3.	RELATION	These are the usual n-ary predicates.
4.	EVENTS	These are instantaneous changes of the before/after variety a la STRIPS.
5.	PROCESSES	These are collections of functions of at least one real variable, often a function of time, describing continuous change.

Note that the first three only describe static knowledge. Particular attention is paid in 4. and 5. to causal relations.

My main problems now are to:

1). Describe the semantics of the LESK statements using relators.

2). Try to clarify the mechanisms of relators, particularly their interaction. This is particularly difficult when asking them to do deduction or when they perform "simulations". In the latter, the two types of dynamic relators discover the result of some cause-effect sequence by allowing a simulated time to elapse.

The final product of this research will be a methodology, illustrated by actual examples, whereby largely qualitative scientific knowledge may be expressed in a form both comprehensible to the experts in the field, and yet compatible with the knowledge representation capabilities of contemporary AI-oriented computer systems. The chasm between "what is in the expert's head" and "what is in the machine" may thus be more comfortably bridged.

The Use of Context in Text Recognition

Godfried T. Toussaint
McGill University

Ever since the study of character recognition machines came into being there has been interest, although limited, in using context to improve the performance of these machines. One of the first papers to explore the idea was the classic one by Bledsoe and Browning in 1959. Their approach consisted of a basic dictionary look-up procedure. One makes a decision on a whole word rather than individual letters. Say five characters, representing a five letter word, are presented to the machine. With each character we have associated confidences, resulting from making measurements, that the character is any of the possible letters. The procedure consists of taking all the words of length five from the dictionary, computing a confidence for each word by combining the confidences of each character assuming the characters are the corresponding letters making up the word in question, and subsequently choosing the word with the highest confidence. This procedure assumes that the text presented consists only of words contained in the dictionary. Hence a large dictionary may be necessary.

Another approach to using context was investigated recently starting in the late sixties. This approach consists of using "bigram" or "trigram" transition probabilities and assumes the language to be a Markov source. A "bigram" is the probability that a certain letter follows a given letter. Clearly in this approach only the n-grams need to stored. If n is small this storage is much smaller than a dictionary and using dynamic programming we can make decisions on words with far less computation than in the Bledsoe-Browning scheme described above.

In the past there has been controversy with regards which of the two approaches is the best: dictionary look-up or Markov methods. Presently I am in charge of a project funded by NRC to investigate this area. Rajjan Shinghal is presently doing a Ph.D. thesis on this topic and our goal has been to resolve this controversy. It turns out that the basic simple Markov method is inexpensive with respect to computation and storage but does not exhibit a marked error-correction capability. On the other hand the Bledsoe-Browning approach has an impressive error-correction capability but is costly. We have been able to define classes of algorithms that, as special cases, reduce to either Markov methods, dictionary look-up methods, or combined methods. Furthermore we are in the process of testing algorithms that use "bigrams" and dynamic programming to search only limited parts of the dictionary. Some of these algorithms obtain 95% of the error-correcting capability of the dictionary look-up methods while requiring only a slight additional cost over the Markov methods.

Image Processing in the Department of Electrical Engineering at McGill University

Samir I. Shaheen and Martin D. Levine

The McGill University Computer Graphics and Image Processing Group is using the following equipment:

Computer Hardware:

The primary processing system is a DEC PDP-15 computer with 32k core, half a million words of disc space, two DEC tape drives, high speed reader/punch, and line printer. Communication with the system is via a Tektronix graphics computer terminal model T4002. This computer is connected to three other processors. One of these is a 4k DEC PDP-8 with two DECtape drives which is interfaced to the PDP-15 via a direct memory access channel. It is to the PDP-8 that most pictorial input and output devices have been connected to-date. The second processor is the IBM system situated at the McGill Computing Centre which is connected to the PDP-15 via a 2400 baud communication link. The third is an 8k PDP-11/10 which is interfaced to the PDP-15 and shares a common memory with it. The laboratory is in the process of alterations designed to connect all image-related input/output devices to the PDP-11.

Special Hardware:

Three kinds of picture inputs exist. The oldest input is an ITT image dissector camera which, together with an associated integrator, is capable of scanning and digitizing an optical image. The latter may be a 35mm. transparency illuminated by a light box or a histological slide viewed directly through a microscope. A second input is a standard ITC VF-301 television camera which, when used in conjunction with a Hughes Model 639 scan converter can provide a dynamic picture source. The new system uses a highly sensitive cohu model 4353 silicon diode camera as the electro-optical device and a sequential column digitizer for scanning and digitization.

Hardcopy, black and white, and color displays are available as output from the system. The Tektronix (T4002) terminal can be utilized to output vector and alphanumeric data and is used for man-machine interaction. A light pen is incorporated in the system for sub-picture selection on the B/W monitor. The Hughes scan converter can accept data randomly under program control and output an image on a black and white conrac 525 line television monitor. Alternatively, the scan converter output may be displayed on a color TV by means of pseudo-coloring hardware. A Tektronix display Unit Type 602 with an attached Poloroid Camera and a computer controlled electronic shutter are used to obtain hardcopy.

Software:

The McGill Image Processing System (MIPS) has been developed as an interactive system for digitizing, displaying, and processing pictures. It allows the user to easily utilize all the complex devices in the Computer Graphics and Image Processing Laboratory in the Electrical Engineering Department, McGill University.

Our group has directed its research towards the solution of specific applications (e.g. biomedical imaging), developing the interactive MIPS system, and designing a knowledge-based scene analysis system. The following represents some of the research activities and interests of the group.

1. Measurement of Human Granulocyte Movement

M.D. Levine   Youssry Youssef

Chemotacis is an important phenomenon relevant to many areas of physiology and developmental biology. The visual methods for measuring this phenomenon have been for the most part subjective. The research project under study concerned with the development of an automatic image processing technique applicable to the cinephotomicrographic analysis of this directional cell movement.

Such a technique could use a statistical analysis to determine whether the cells are moving in a random fashion or whether they are being influenced by external forces.

2. Computer Studies of Fibroblast Morphology

Grant Brighten   M.D. Levine

A package of interactive computer programs has been developed to analyse the texture of digitized images. Texture feature vectors are produced through an interactive process whereby the user supervises a series of computer operations to produce the most useful set of texture measures for a particular application. The features may be used for image classification or as an aid in picture segmentation. Currently, the package is being used to detect the presence of diseased cells in tissue cultures from patients with osteogenesis imperfecta, a genetic disease of the connective tissue. Tissue cultures were photographed at the Montreal Children's Hospital and processed by computer in the Department of Electrical Engineering at McGill. These cultures are now being examined using the texture analysis package with the objective of discriminating normal and diseased cells.

3. Computer Synthesis of Line Drawing Using Semantic Nets

Ray Giustini

A program which employs a semantic memory for the generation of simple line drawings has been developed. Input to the program is by means of a Picture Language (PL) whose syntax is also defined. The semantic memory used is modeled after Quillian's. Compilation of a PL input by the program results in the creation of a Short-Term Memory (STM) which partially defines the objects to be drawn. Any missing information which is necessary to draw the objects is supplied by the semantic memory.

A number of examples which demonstrate some features and shortcomings of the system are described. The results of these examples, together with proposed changes which would improve the versatility and efficiency of the program, demonstrate the feasibility of incorporating a semantic memory into the software of an interactive graphics display system.

4. Interactive Feature Measurement and Selection

Jack Russ   Martin D. Levine

An interactive package is being developed so that various useful features of selected objects in complex pictures may be measured. A light pen is used for tracing the region containing the objects which are then segmented automatically using pattern recognition techniques. This package is one of the MIPS modules undergoing development and will be subsequently used for the interactive analysis of the geometry of the dynamic physiologic reactions of blood platelats.

5. Segmentation of Natural Images

Juhan Leemet   Martin D. Levine

The main objective of this project is to write a program that will segment an image into fragments having well defined global properties based on feature derived from the three primary colors of visual perception. These features (intensity, hue, saturation and texture) are similar to the psychological variables that people use in performing perceptual discrimination. By using global properties of the image for planning and a clustering algorithm to decide the region membership of individual picture elements, the program finds a good set of fragments for partitioning the image. It then passes descriptions of these fragments on to other high level programs for further processing and interpretation. Additional areas of investigation are in color description and transformation, clustering algorithm and parameters, and spatial connectivity constraints on clustering in multidimensional feature space.

6. Scene Segmentation and Interpretation

David Ting   Martin D. Levine

One of the problems in analyzing digitized pictures is assigning possible interpretations to the individual scene fragments which would yield an optimal meaning for the entire scene. In this project an approach using heuristic search and dynamic programming attempt to assign interpretations to the fragments which would be locally optimal and globally consistant with the given scene description. The scene description provides the knowledge concerning the picture and includes the geometric properties of the regions and the spatial relations between them. The scene description will ultimately be obtained by intergating a relational database model of the picture to be analyzed.

7. Computer Generation of Realistic Pictures

Masoyoshi Aoki
Visiting Assistant Professor

It is almost impossible to generate realistic pictures by computer and one approach to this problem is to use real subpictures as building blocks. In this approach, a three dimensional object could be represented by a set of real pictures depicting each of its elementary surfaces. Projective transformations could then be utilized to construct a picture and this is being implemented on the MIPS system. Such an approach can also be used for stereo picture generation and realistic animation.

8. An Associative Database for Creating World Models for Image Analysis

Soly Haboucha Martin D. Levine

A high level image processing system must incorporate a considerable amount of information regarding the problem domain under study. In this project, a relational data base is utilized as the building block for constructing an associative memory for picture information. Both three-dimensional and the appropriate two-dimensional data will be maintained.

9. Binary Array Processing

Dr. A. Reeves
Visiting Assistant Professor

The basic features of a Binary Array Processing (i.e. one which processes bitplanes rather than numbers) have been studied and a parallel hardware organization for a binary array processor has been developed. An APL computer simulation of this processor has been used to develop both a program notation for the processor and some image processing algorithms. A simulator has also been written in FORTRAN which interfaces with the MIPS system. This can be used for testing image processing algorithms on full sized images.

10. A Color Graphics Video System

A. Malowany, M.D. Levine, D.Dood, R. Hum, D. Monteith.

An interactive Color T.V. video display system has been developed for the computer graphics and image processing laboratory in the Electrical Engineering Department at McGill. A Data disc is used for storage and refreshing of a color image which is displayed on a Conrac color T.V. monitor. A high resolution mode plots 776 pixels per raster line while a low resolution mode offers 4096 shades. Software currently supports the interactive creation and filing of pictures as well as the display of wire frame and solid three dimensional objects with shading. In addition our color video system includes a computer-interfaced Hughes scan converter unit with hardware for pseudocoloring of gray scale images. Pseudocolors may be assigned at a manual station for high resolution mode in the disc refreshed system. Three color graphics facilities are being used in the study of computer models, of algorithms, and of hardware configurations aimed at real time animation in simulators. The system is also being used in the development of a knowledge-based scene analysis system for real and synthetic images.

11. Scene Description and World Model

Samir I. Shaheen Martin D. Levine

A relational database package has been developed on the Time Sharing System (TSO) on the McGill University IBM system. A relational algebra sublanguage has been implemented for either interactive interrogation of the database by a casual user or to be incorporated by other high level programs for the general purpose image understanding system.

The relational database world model is designed to be used as an associative memory for scene analysis or synthesis. Research is presently underway to develop a high level image analyzer which will take as input pictorial data and output a symbolic description based on the information in the database.

12. Depth Perception

David Rosenberg Martin D. Levine

One problem indigenous to visual interpretation is that of depth perception. In the past, little work has been done on this subject, most of it using two cameras or a camera/laser set-up. This project deals with monocular depth perception utilizing contextual depth clues. It attempts to discover depth relationships and object occlusion by consideration of the information in either one picture or two, the latter not necessarily a stereo pair.

Courses Directly Related to the Research

- Digital Signal Processing	(E. E.)	304-512
- Image Processing	(E. E.)	304-529
- Real Time Systems	(E. E.)	304-531
- Computer Graphics	(E. E.)	304-532
- Artificial Intelligence	(E. E.)	304-625
- Real Time Operating System Design	(E. E.)	304-636
- Computer System Analysis	(E. E.)	304-637
- Advanced Computer Design	(E. E.)	304-638
- Heuristic Programming	(C. S.)	308-425
- Database Systems	(C. S.)	308-520
- Information Structure	(C. S.)	308-610
- Data Structure and List Processing Techniques	(C. S.)	308-611
- Pattern Recognition I	(C. S.)	308-644
- Pattern Recognition II	(C. S.)	308-645
- Advanced Topics: Mini-Computers Software	(C. S.)	308-762
- Advanced Topics in Pattern Recognition	(C. S.)	308-766

People

Dr. Steven Zucker who is presently a Research Associate with Dr. Azriel Rosenfeld at the University of Maryland will be joining the group in September as an Assistant Professor.

Dr. Masayoshi Aoki, Visiting Professor, Faculty of Engineering, Seikei University, Tokyo, Japan is spending a sabbatical year with the group doing research in computer vision.

Dr. Alfred S. Malowany is presently on Sabbatical at the IBM Research Laboratories in San Jose, California working on computer graphics and will be returning in September.

Seeing Puppets Quickly

J. L. Paul
University of Sussex

Until a few years ago, the traditional goal of scene analysis programs was to recognize a scene consisting of blocks sitting on a table, the blocks not representing anything else, e.g. an arch, person, etc., just blocks sitting on a table. Several years ago, Max Clowes, here at the University of Sussex, decided to try his hand at recognizing scenes, still in the blocks world, in which the blocks are meaningfully related. To this end, he constructed a puppet consisting of a trunk, head, two arms, and two legs, the arms and legs free to rotate forward and backward with respect to the trunk. This puppet was then photographed in a number of postures, e.g. sitting, standing, walking, etc., with a strong shadow and no background.

When I came to start work on a D.Phil., I noticed that to interpret the photographs of the puppet, almost no knowledge of polyhedral objects was required. In fact , in looking at the pictures, one first noticed the puppet's posture and orientation, and if one looked closer, one could see that the parts were composed of polyhedral objects. Now, a "traditional" scene analysis program would have worked in the opposite direction, first finding lines, perhaps looking for junctions, closed regions, or line configurations that would suggest a block, then finding the relations between the blocks (or perhaps the images of the blocks) to suggest a puppet (e.g. Winston).

My research here consists of trying to get the computer to see these pictures in pretty much the same way we seem to. That is, I want it to be able to give a rough 3-d description of the scene in terms of the approximate direction the puppet is facing, the approximate position of it's limbs with respect to the trunk, perhaps a posture label if possible, e.g. "sitting", "standing", etc. This description should not be found in the "traditional" manner, bottom-up, as outlined earlier, but rather in a middle-out manner.

This is not the place to describe, or justify, the "guts" of the program. However, let me say a few things about it.

-The "regions" sought from the output of the line finder consist of clusters of parallel lines, rather than, say, a loop of lines, or a region of uniform intensity. The hope is that, in most cases, these regions will correspond to a part of the puppet.

-The model the computer uses for finding the puppet is 2 1/2 dimensional (i.e. each part is 2-d, but they can occlude each other), rather than full 3-d.

-Job control is accomplished by putting the jobs run onto a job queue. The scheduler then examines the descriptions of the jobs, rather than just the job priority, say, to decide what to do next.

-The program tries to find as good a puppet as it can from the regions found. If not all the parts of the puppet are found, it goes back to the derived line diagram, trying to explain other lines in terms of the missing parts, or finding excuses for what part is not visible, eg. an arm might be hidden by the trunk.

A Helpful Computer Conversant

Philip R. Cohen
Department of Computer Science
University of Toronto

This thesis will deal with the problem of man-machine conversation by examining language from a "speech act" point of view. The concept of a speech act, developed by philosophers of language such as Austin and Searle, attempts to account for what a speaker is doing in saying something. REQUEST, ASSERT, WARN, SUGGEST, BET, etc. are examples of speech acts. The import of this analysis is that linguistic acts resemble non-linguistic acts and thus both can be treated in a more uniform fashion.

I concentrate on dialogues in highly structured situations where the machine has the motivation of helping its user. In this environment, the machine must be able to analyze and generate speech acts. I take the position that, at the speech act level, analysis and generation are part of a more general planning process. Thus, the machine decides what to say based upon a global plan for being helpful. In order to do this, the machine must have strategies for helping and must maintain models of its conversants, including their beliefs and goals.

A prototype system is being developed that incorporates a representation of knowledge, based upon a partitioned semantic network, which allows beliefs to be shared. The system is to be given the internal representation of an utterance, should identify it as some speech act, and should generate a speech act (perhaps a few) under the assumption that the conversants are engaged in a purposeful helping dialogue. The system's analyses of utterances are centered around inferring the plans of its conversants, and discovering how the uttered speech acts could serve to further those plans.

A Programming Advisor

Lucio F. Melli
Department of Computer Science
University of Toronto

Large programs present large problems: they are difficult to write, debug, test and modify. This "complexity barrier", as Winograd[1] so aptly describes it, may be conquered by programming systems which are capable of "understanding" programs and of acting as programmers' assistants. This is not as implausible as it may first seem, since what is needed is a knowledge-based system (a currently hot area in AI) whose domain of knowledge is programming. Although there are many opinions of what the assistant should do, Winograd sees it as a highly interactive system behaving as a "moderately stupid" assistant that relieves the programmer of the burden of memory work, checking and drawing more-or-less straightforward conclusions. It is not intended to be an automatic programmer, nor a program checker which guarantees correct and efficient programs. Rather, it is supposed to be a system which helps to magnify the programmer's effort and aid in producing programs.

A problem with the programmer's assistant is the fact that "understanding" and "assistance" take on different meanings depending on the competence and goals of the typical programmer using the system. In other words, there is no clear understanding of what such a system is supposed to know or what it is supposed to do. It is therefore interesting to question whether there could exist an ideal programmer's assistant capable of alleviating the programming problems of one and all. Conversely, if there exists a continuum of assistants, what are common characteristics (if any)?

The goal of the research is to gain a better understanding of what is meant by a "programmer's assistant" and the feasibility of constructing such a system. In particular we are considering a system which behaves as a programming advisor, i.e. a system to help novice programmers with their programs. Although one could argue that such an advisor is skirting the main problem since it is not dealing with large programs, we hope to show that there exist certain similarities between a program advisor system and a programmer's assistant system. Therefore, our research can be viewed as a step toward a programmer's assistant.

We are currently in the process of compiling dialogues between a novice programmer and a human advisor. The domain of discourse for the dialogues has been limited to:

1) the problem to be solved.

2) a specific program attempting to solve the problem.

3) the programming language used.

A further constraint is that the advisor prefers specific questions rather than more general ones (e.g. "Why doesn't my program work?"). These dialogues will be used in determining what facilities and information are required from such an advising system. With such information, it will be possible to study the feasibility of constructing a computer system to handle some of the aspects of advising beginning programmers.

1. Winograd, T.; "Breaking the Complexity Barrier Again"; SIGPLAN Notices, Vol. 10, No. 1 Jan. 1975, pp. 13-22.

A Picture Processing Model

Richard Peacocke
Dept. of Computer Science
University of Toronto

This abstract outlines work on a formalism and programming language extensions for picture processing. The kind of picture processing concerned is pictorial pattern recognition where structure is to be elicited from raw picture data. Low-level processing has generally used a discrete rectangular grid of intensity values as a model of the picture and causes transformations to be made on it. Following Narasimhan[1] these transformations take the complete picture array for both input and output arguments. The model used in this work is based on regions[2,3] and other geometric structures out of which pictures and parts of pictures can be described. Regions have two components, firstly, a geometric specification of the domain and secondly, properties and functions defined over that domain. Various operations on the structures are described and relationships between them are shown. The operations depend as little as possible on the particular representations chosen for the structures.

Some basic techniques used in pictorial pattern recognition have been studied in order to show the relationships which exist between them. Correspondence between these techniques and the structures and operations of the formalism is demonstrated. A selection of programs and algorithms has been made which covers many of the techniques and a wide range of pictorial pattern recognition applications from chromosome analysis to the blocks' world. Parts of these algorithms have been coded using a programming language which consists of an ALGOL-like base with the addition of picture structures and operations drawn from the formalism.

The description of one of the examples, cloud cover mapping[4], has resulted in a reformulation of the published algorithm in such a way as to reduce the rate of increase in the number of picture points which need to be considered as picture resolution becomes finer. Another application being developed is the measuring of DNA molecules in electronmicrographs. POLLY[5], a programmable film scanner, is being used. Its measuring operation is designed for following the line-like tracks encountered in bubble chamber experiments. Since the DNA molecules are not line-like and the pictures contain certain irregularities, regional analysis is used to supplement the local line-finding procedures.

References

[1] Narasimhan R.; "Labelling schemata and syntactic descriptions of pictures"; Information and Control, Vol.7, July 1964, pp.151-179

[2] Muerle J.L. and Allen D.C.; "Experimental evaluation of techniques for automatic segmentation of objects in a complex scene"; in Pictorial Pattern Recognition, Cheng G.C. et al (eds.), Thompson Book Co., Washington, 1968, pp.3-13

[3] Brice C.R. and Fennema C.L.; "Scene analysis using regions"; Artificial Intelligence, Vol.1, 1970, pp.205-216

[4] Strong J.P. and Rosenfeld A.; "A region colouring technique for scene analysis"; CACM, Vol.16, No.4, 1973, pp.237-246

[5] West E.C.; "POLLY: A system for the analysis of scientific photographs"; Canadian Research and Development, Nov-Dec 1971

A Semantic Network Model of Data Bases

Nick Roussopoulos
Department of Computer Science
University of Toronto

The Semantic Network Model of data bases (SNM) assumes the availability of a semantic network storing knowledge about a data base. The organization of knowledge offered by the network is utilized in dealing with semantic problems in data management. Some preliminary results have been reported in [1].

Semantic networks offered features which we have found very useful for dealing with problems of data management, such as the ISA hierarchy, some distinction between different kinds of semantic relationships, an aesthetic appeal, etc. On the other hand, they suffered from the lack of a much needed representation of quantification, misconceptions about their meaning and the creation of their constructs, etc. Nevertheless we decided to base our data model on semantic nets and face some of their problems.

We have augmented semantic networks with two new features. First, a representation of quantification is provided which has the following characteristics:

(a) It has the representational power of a many-sorted first order logic.
(b) It uses a frame structure for expressing logical units of quantification.
(c) It handles numerical as well as the standard universal and existential quantifiers.
(d) It handles both open and closed forms of quantification (with or without free variables). Those two forms are identified with "can-hold" and "does-hold" intensional knowledge.
(e) It distinguishes two kinds "does-hold" knowledge "complete" and "incomplete", useful for the self-awareness of the network's extensional (factual) knowledge.
(f) It comes along with a set of inferences for organizing quantified statements on the ISA hierarchy algorithmically; this organization allows efficient "matching" of quantified statements and certain type of consistency and/or redundancy checking.

Second, a semantic network algebra is defined and allows to expand the network consistently (guarantees the syntactic correctness of semantic nets). The operators on the network also allow, in certain cases, to determine whether or not a constructed structure is meaningful.

Although the SNM is based on a semantic network representation of knowledge it should be emphasized that the model depends only on features of semantic networks rather than a particular implementation of such a representation. These features are, the ISA hierarchy, the distinction of "concepts", "events" and "characteristics", the proposed representation of quantification and the semantic operators. The model uses the relational data structure [2] for storing its extensional knowledge but offers its own algebra for manipulating it.

The SNM offers solutions to three fundamental problems in data management.

(a) The generation of a meaningful relational schema from the semantic net.
(b) Meaningful data base operations that are the effects of corresponding semantic network ones. The results of the data base operators are associated with the results of the semantic network ones and this allows derived data base relations to be treated like primary ones (evolution of the data base).
(c) Maintenance of the data base integrity during updates. Algorithms are provided for deciding whether or not the

logical constraints expressed through quantification are violated on updates.

In dealing with each of these problems, other solutions proposed in the literature are compared.

REFERENCES

1. Roussopoulos, N., & Mylopoulos, J., "Using Semantic Networks for Data Base Management", First International Conf. on Very Large Data Bases, Boston, Sept. 1975.

2. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks", Comm. of ACM, vol. 13, no. 6, June 1970.

Application of AI to Data Management

John Mylopoulos
U. of Toronto

Data Management is faced today with a number of problems related to the semantics of data bases that have direct counterparts in AI. The aim of this project is to explore the possible application of methods developed within AI to the solution of such problems.

A major concern for AI as well as DM is the representation of information about a "world" e.g. a world of employers, employees, managers, departments, salaries etc., or a world of students, professors, grades, courses, prerequisite courses etc. In DM, "models of data bases" have been developed (relational, network, hierarchical) in terms of which a Data Base Management System (DBMS) designer can represent certain kinds of information about his data base. Recently there has been a trend toward more sophisticated models [1, 2, 3] which attempt to capture a greater portion of the semantic information about a data base. Such models are beginning to distinguish between different types of semantic entities (e.g. "objects" and "relationships") but are still lacking in several respects when compared to representations developed within AI.

Our work on this issue has concentrated on the development of a "Semantic Network Model" (SNM) which uses an ISA hierarchy, some form of frames, quantification and pattern matching to model the world of a data base. In order to justify such a model, we consider semantic problems of data bases and show how a SNM can contribute to their solution. Some of the problems we are considering are:

(a) The algorithmic generation of a relational schema from a SNM for a data base, to be used by a relational DBMS.

(b) The definition of meaningful semantic operations on the SNM, along with a specification of corresponding data base operations.

(c) The maintenance of semantic consistency for the data base.

(d) The development of "natural" query languages in terms of which a casual user can communicate with a DBMS.

Our work so far includes the design and implementation of a prototype natural language front end to a relationl DBMS (TORUS [4, 5, 6, 7]), and an extensicn of the representation used in TORUS along with a study of how the extended representation might be used to solve problems (a) - (c) mentioned earlier ([8, 9, 10]). In the future, we propose to continue this work in several directions:

(a) Define more formally the SNM. For this work we have found a paper by Abrial [11] particularly useful.

(b) Provide a programming system for constructing, maintaining and accessing a semantic network.

(c) Survey the work that has been done on the semantics of data bases within DM, evaluate it and relate it to corresponging work within AI.

(d) Design and implement a system in terms of which a designer can generate "easily" a DBMS for a "simple" data base.

(e) Design and implement a second version of TORUS.

The persons working on various aspects of this project in June 1976 are Nicholas Roussopoulos, Harry Wong, John Tsotsos, Hector Levesque and John Mylopoulos.

1. Bracchi, G., Paolini, P., Pelagatti, G. "Binary logical associations in data modelling" Proc. IFIP-TC-2-WC, Black Forest, 1976.
2. Chen, P. "The entity-relationship model: Toward a unified view of data" TODS Vol.1, #1, 1976.
3. Deheneffe, C., Hennebert, H., Paulus, W. "A relational model for a relational model for a data base" Proc. IFIP-74, North-Holland, 1974.
4. Mylopoulos, J., Borgida, A., Cohen, P., Roussopoulos, N., Tsotsos, J., Wong, H. "TORUS - A Natural language understanding system for data management" Proc IJCAI, Sept 1975.
7. Mylopoulos, J., Borgida, A., Cohen, P., Roussopoulos, N., Tsotsos, J., Wong, H., "TORUS - A step toward bridging the gap between a data base and a casual user" _Information Systems_, 1976 (to appear).
8. Roussopoulos, N. "A Semantic Network Model for dta bases" Ph.D thesis, 1976 (forthcoming).
9. Roussopoulos, N., Mylopoulos, J. "Using semantic networks for data base management" Proc. Conf. on VLDB, Sept 1975.
10. Roussopoulos, N., Mylopoulos, J. "The semantics of semantic integrity" (submitted for publication).
11. Abrial, J. "Data Semantics" in Klimbie and Koffeman (Eds.), _Data Base Management_. North Holland, 1974.

A Computer Model of Conversation

Participants: James Allen, Phil Cohen, Robin Cohen,
Prof. C. R. Perrault, Corot Reason, Mary Horrigan-Tozer
Department of Computer Science
University of Toronto

This project is concerned with the modelling of language use in a conversation. The aim is to develop techniques for making a machine a more helpful conversant. We are examining problems that have essentially been ignored in current AI natural language research. For instance, our model will have strategies for deciding what to say, for inferring a speaker's intentions, and for being helpful. In order to do this, the system maintains an explicit model of the user's beliefs which includes the user's model of the system's beliefs, etc. This technique is implemented by using Hendrix' partitioned semantic nets.

Our methodology centers around the viewpoint that language is primarily a purposeful means of communication. Following philosophers of language (e.g. Austin, Searle, Grice) we employ the concept of a speech act to analyze what a speaker is doing in the process of uttering something. This concept enables us to include speech acts in plans and employ current planning techniques in the generation and recognition of speech acts.

Regarding individual research: Mary Horrigan-Tozer is examining the speech act structure of real dialogues in information acquisition settings. Robin Cohen is studying the problem of time and tense employing the belief model. Phil Cohen is developing a system which can identify and generate speech acts based upon planning. James Allen is interested in the uses and influences this approach has at the surface language level.

Our prototypes are being implemented in the SPITPLUS version of SPITBOL, which was developed by Walter Berndl and allows for interactive execution and monitoring of SPITBOL programs on OS/MVT with TSO. Currently implemented are: a partitioned semantic net package, a knowledge-source-blackboard message passing scheme (a la CMU) and a MEMOD style network executor and definition package for creating and executing semantic nets. The partitioned net package was develcped by Prof. John Mylopoulos.

Two Low Level Vision Projects

A. K. Dewdney
Department of Computer Science
University of Western Ontario

The projects described below under the titles IMAGE MATCHING TECHNIQUES and TELECONFERENCING SOFTWARE were carried out during the last ten years under N.R.C. and D.O.C. grants, respectively.

The first project, started in May '75, ended in September '75 with the completion by John Lemieux, research assistant, of his M.Sc. project entitled "Image Matching - A Set of Preliminary Experiments". The second project, assisted by Mr. Reginald Quinton and begun in May '75, is due to end by July '76. The following descriptions are little more than expanded abstracts (with pictures): in a few months fuller descriptions of this work should be available.

Image Matching Techniques

In a wide range of automated picture-processing applications, it is desired to find in picture B the best match for an object (subpicture) in picture A. This is especially true of space-serial (1) and time-serial (3) picture pairs: in the first example A and B are digitized stereo photographs. In the second case, A and B are digitized satellite photographs of cloud cover over the Pacific Ocean, taken at different times. In the first case one wishes to "register" the photographs by finding objects common to both and thus compute range information about these objects in the scene. In the second case, it is desired to construct a wind-velocity map of vast areas of the Pacific by following individual cloud formations though successive scenes.

For the greatest reliability possible, under the circumstances, it is best to examine the entire picture B and, in such a case, a variety of techniques are available such as correlation (1, p. 5) or the fast Fourier transform (3, p. 282). In spite of certain advantages of the latter technique over the former, both require $O(n^2)$ time where n is the linear dimension of the digitized picture.

The Image Matching project sought to explore the advantages of employing, wherever possible, a hill-climbing (more properly, "valley-descending") technique for image matching. It works as follows: suppose a measure f of difference between matrices is given. This measure, for example, might be

$$f(M,M') = \quad (m(i,j) - m'(i,j))^2$$

If M(1) contains an image of an object in A and M(2) is a given submatrix of B having the same dimensions as M(1), then we first compute f(M(1),M(2)). Next, the four matrices M(2,1), M(2,2), M(2,3), M(2,4), obtained by translating M(2) one pixel north, south, east, and west, are matched against M(1) and the one yielding a minimum value of f(M(1),M(2,j)), j=1,2,3,4, say M(2,k), is chosen. If $f(M(1),M(2,k)) < f(M(1),M(2))$, then M(2) is replaced by M(2,k) and the algorithm continues, otherwise it halts with M(2) as output. Whenever this algorithm works, it requires only C(n) computation time.

For reliability, this technique requires that one begin with a matrix M(2) from which one can "descend" directly to the target matrix. The likelihood of making such a choice, depends on the size of "convergence area" of M(1) in B, that is, the set of locations in B from which the algorithm will successfully find the best match for M(2). It was surprising to discover how frequently a large convergence area existed.

A study (2) involving a wide variety of natural and artificial scenes convinced us that the convergence area was nearly always large enough to be worth "aiming for".

In isolation, the convergence technique for image-matching has very poor reliability. Its strength, however, apppears in a context where good predictions can be made about where to begin the convergence process. such contexts are certainly available when the space or time between pictures A and B is small enough. An interesting feedback effect can be exploited in the time-series picture domain: speed-up due to a more efficient image-matching algorithm makes faster image input possible. This, in turn, makes the algorithm faster and more reliable, etc. The latter effect has yet to be studied in detail however.

Many interesting theorectical questions surround the convergence technique of image-matching. In a given domain what is the expected converence time for successful matches? Is there a good mathematical characterization of submatrices having "large" areas of convergence?, and so on. These questions have been studied in an abstract geometric picture domain, but only partial results have so far been obtained.

The references given below represent only a tiny fraction of the available literature directly connected with the subject of image-matching.

References

(1) Hannah, M.J., Computer matching of areas in stereo images, Stanford Artificial Intelligence Project, Memo #239 (1974).

(2) Lemieux, J., Image matching - a set of preliminary experiments. M. Sc. Project, U.W.O., Dept. of Computer Science, September, 1975.

(3) Leese, J.A. and Novak, C.S. and Taylor, V.R., The determination of cloud pattern motions from geosynchronous stellite image data, Pattern Recognition 2 (1970), 279-292.

Teleconferencing Software

An interesting and challenging communications problem involves the real-time transmission of T.V. imagery over low-bandwidth channels such as telephone lines. Research activity in this area has focused on the construction of mathematicals models to capture the intuitive notion of redundancy obvious from a glance at ordinary television imagery. One non-trivial subproblem of the above sort, suggested by the D.O.C. some years ago, was to transmit televised "conference scenes" over telephone lines.

These scenes typically involve a group of people seated behind a ccnference table.

One method of capturing redundancy is to use a computer to discover only those areas of the scene where people are found and tc transmit the smallest possible picture submatrix containg "significant" changes in their positions relative to the ccnference scene background. Here, "significance" results when a simple measure of different between correspondinding portions of successive scenes exceeds a given threshold.

Conference scene participants are not located afresh with each new digitized frame transmitted from camera to computer, that is, the problem of finding them is not treated as the problem of recognizing people in static pictures - a hard and somewhat unrewarding project. Instead, it is possible to take advantage of the time series of frames to isolate quite successfully "snapshots" of participants by comparing the current submatrix containing them with the corresponding portion of an earlier scene which did not contain them. For example, Figures 1 and 2 show a digitized picture of the author and his "ghost", respectively, seated behind a desk. The complement of Figure 2 contains the author's snapshot isolated from the scene.

But how does one find the "current submatrix containing them" (the participants)? First, a participant's last snapshot is used as matrix M(1) in the convergence technique described in the previous section; the matrix B represents a portion of the current scene in which one may reasonably expect to find him, based on his most recent position and, if necessary, velocity. The best match in B for M, which the convergence algorithm is able to return, is accepted as the basis for a new snapshot provided it falls below an acceptable match threshold. If the convergence algorithm fails, however, a more general search called "change-detection" is carried out. This involves scanning a submatrix somewhat larger than B, computing all pixels in which changes in grey-level exceeding a certain threshold are found, finding the largest connected subset of pixels thus "changed", and finally determining the smallest submatrix containg this subset. It has been possible to "track" two people in this manner with a preliminary version of this software, but only when they remain reasonably separate. There are, in this context, a number of approaches to the occulaticn problem (eg. treating each of the two participants as "background" in turn and seeing which one remains in the present research period.

The low-level processes just described are coordinated by higher level tracking software whose job it is to maintain a snapshot file and to transmit a corresponding submatrix when a new snapshot it fcund necessary.

Thus, participants at a teleconference of the sort described above would watch their distant colleagues continually painted anew on a large panoramic screen in some temporarily frozen gesture which, while not as rich in visual interest as a continuously moving representation, nevertheless expresses what is going on.

Figure 1

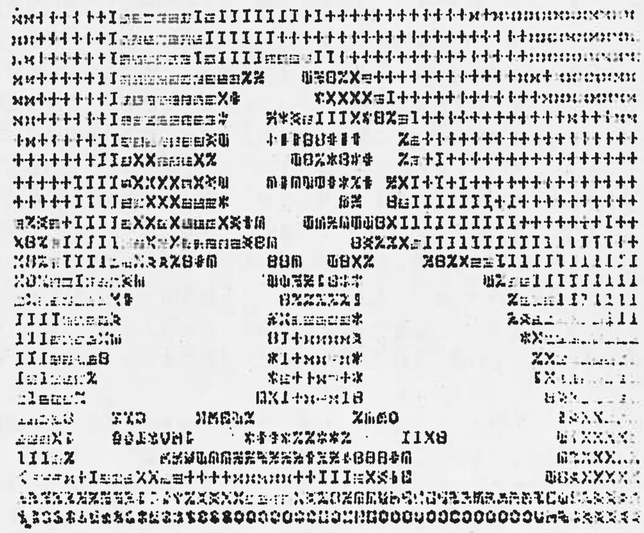


Figure 2

Implementation of a "Vision System"

A. H. Dixon
Department of Computer Science
University of Western Ontario

In the CSCSI newsletter, Volume Two, a description of hardware resources acquired by the Artificial Intelligence Group at UWO for research into Scene Analysis was reported. One of the primary software goals has been the establishment of a "vision laboratory" to allow for the study of various design strategies in a total system to generate descriptive text of input received through a television camera.

Software on the INTERDATA 7/32 has now been implemented and performs two tasks; the construction of a <u>partial</u> line drawing of the scene and the execution of specific sampling routines on the image as requested by programs running in POP-2 on the PDP-10. To allow for the testing of alternate schemes, the system has been implemented in modules to perform the tasks of detecting local discontinuities, accumulating local evidence, hypothesizing edges through curve fitting, and refining and verifying the hypothesized edges through a "second look at the scene". The hypothesized and verified edges are then passed to the PDP-10 where they are received and interpreted as a partial line drawing by a POP-2 program.

The current implementation uses convolution operators [Marr,2] to detect discontinuities. A modification to the rho-theta parameterization as described in Duda and Hart [1] is used to accumulate local evidence. Instead of applying the transformation over the entire picture plane, the plane is segmented into smaller squares and the transformation applied so as to accumulate discontinuities in each square. The dominant rho-theta value is selected for each square of the picture plane and the set of values is then partitioned into subsets, a process which corresponds to the hypothesis of edges. To each subset is fitted a line. This is used to define a trajectory across the camera image over which a convolution operator is applied ("second look") to verify and refine the location of the edge.

The goal of the system described above is not to construct a complete line drawing of the scene but rather a reasonably reliable partial line drawing to allow a higher level system to make inferences about missing lines in its effort to interpret the existing lines in a meaningful way. The expectation of finding a line currently missing can then invoke a request to the INTERDATA program to apply its verification routines to confirm or reject the hypothesis. By "higher level" system is meant a goal directed program having available knowledge about the scene domain in order to make useful inferences. Such a system is currently under construction using the planner type language POPLER 1.5.

Two design principles will influence the implementation of this system. First, the construction of an interpretation is to mediate the analysis of the image, and secondly, the role of information about the scene domain in constructing that interpretation is to be demonstrable. Consequently an attempt will be made to separate as much as possible data structures and procedures reflecting knowledge about the scene domain from the control structure and procedures for generating an interpretation. Such an implementation will be in keeping with the spirit of a vision laboratory research tool. It will allow one to experiment with different scene domains as well as observe any degradation of performance which results from removing portions of the knowledge base.

Three similar but distinct scene domains are currently being used for testing of both low and high level systems; Euclidean geometry diagrams, a polygonal "blobs" world, and a polyhedral blocks world. Partial line drawings have been constructed by the INTERDATA system and used to generate an appropriate data structure representation in POP-2, to allow for the use of line and vertex labelling schemes such as described by Huffman, Clowes, or Waltz. A complete scene analysis system for at least the simplest of the specified domains is scheduled for completion by the end of the year.

REFERENCES

1. R.O.Duda, P.E.Hart, Pattern Classification and Scene Analysis, John Wiley, 1973, p.335.

2. David Marr, "The Low Level Symbolic Representation of Intensity Changes in an Image", MIT AI Memo 325, December 1974.

The Analysis of Motion in Quiet Scenes

D. C. Hogg
Department of Computer Science
University of Western Ontario

The analysis of motion within a scene is a relatively untouched problem, perhaps because at first it appears to be dependent upon successful analysis of the individual snapshots that might be taken at intervals throughout the period of motion. However, recent contributions suggest that the very existence of motion within a scene can be an important tool in automatic scene analysis and should therefore be considered seriously in such systems.

I am in the process of implementing a real time motion analysis system on our Interdata Computer with T.V. camera attached. A simple region growing approach is used to consolidate local evidence of motion. The motion centres, so identified, are the subject of a low level study, although this initial analysis is limited by the desire to operate in real time. Our intention is to transfer the intermediate information so generated, to a POP-2 program running on the PDP-10 where the kernel of a system already exists for drawing 'intelligent' conclusions. As a preliminary to a more advanced motion analysis system, these conclusions will involve the recognition of simple motion across the scene. This high level system is based on a hierarchical decision making process, allowing natural expansion for the recognition of more complex scenarios.

A snapshot of the scene is taken between 3 to 8 times every second at each point in a 64 by 64 mesh, using a sampling algorithm due to Dr. A. Dixon. The overall system is being built in a modular fashion to allow for future expansion at the low level and flexibility at the high level where most of our experimentation will be done. The interprocessor link allows information flow in either direction, between the low level routines on the Interdata and the POP-2 program on the PDP-10 and it is therefore hoped to provide high level guidance to increase the reliability of our low level system. The link might also be used to cause further low level study of selected motion centres or to eliminate uninteresting studies.

In summary, the complete system will perform a simple analysis of a scenario, without the aid of stationary scene decompositions and could be used in future as an aid for such automatic scene analysis.

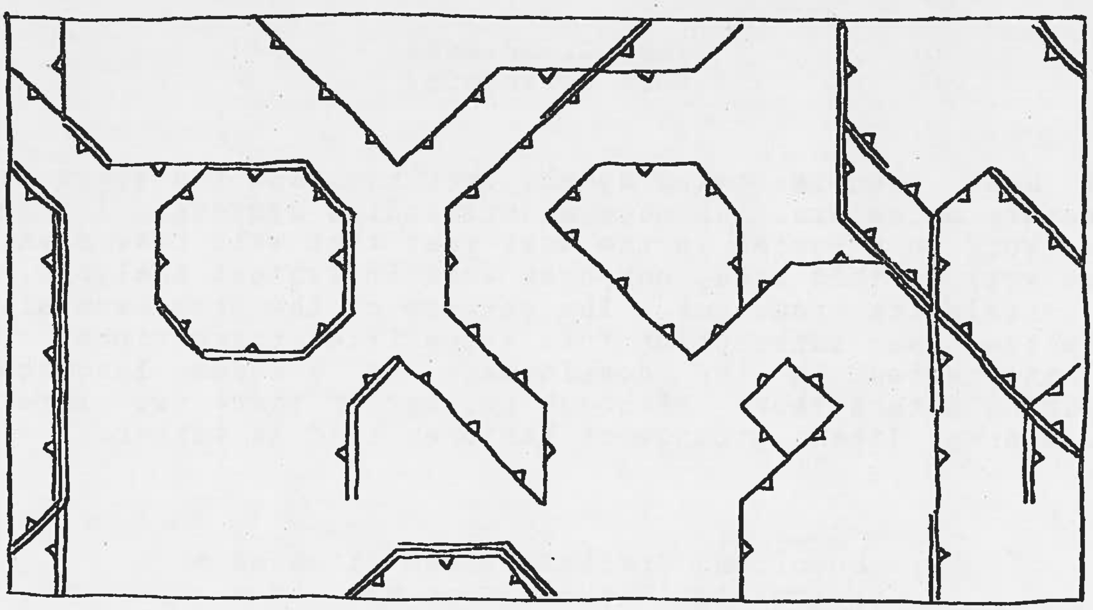
Significant Contours

Donald Kuehner
Department of Computer Science
University of Western Ontario

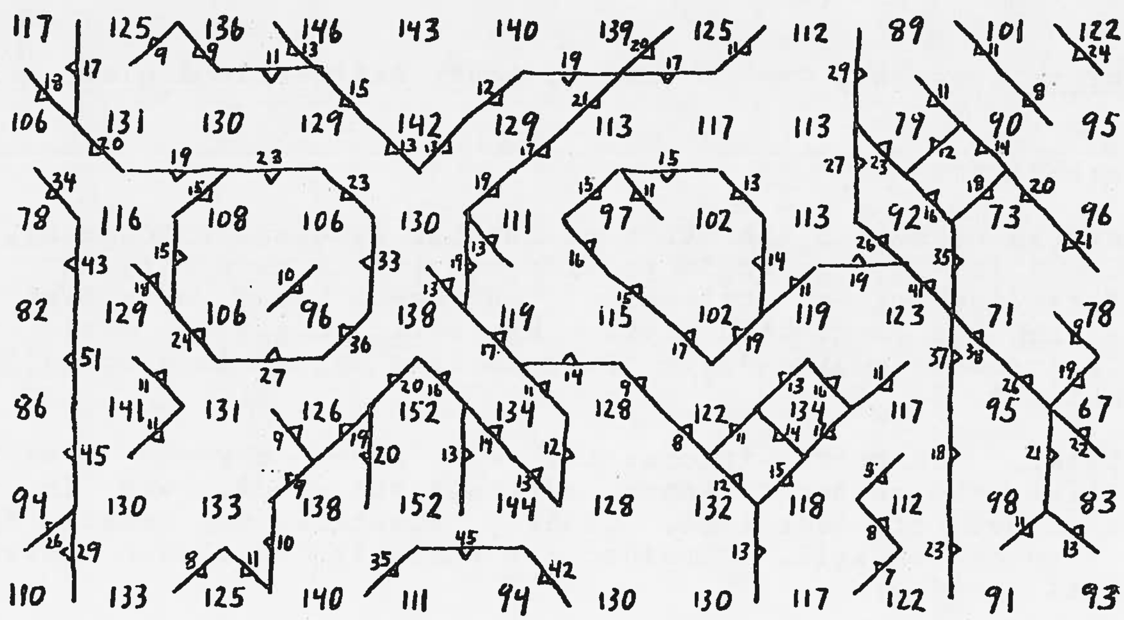
During the last year, the technique of significant contours has been developed. Differences between adjacent picture points are deemed significant if they are above a fixed threshold. Adjacent significant differences are joined if this is a best approximation to a contour line. These significant contours are labelled with an arrow on the darker side, and their thickness or a number is used to indicate the strength of the contour. The resulting line drawings seem intuitively satisfactory. The accompanying pictures are of the middle of a face with eyes, nose and some hair showing.

117	125	136	146	143	140	139	125	112	89	101	122
106	131	130	129	142	129	113	117	113	79	90	95
78	116	108	106	130	111	97	102	113	92	73	96
82	129	106	96	138	119	115	102	119	123	71	78
86	141	131	126	152	134	128	122	134	117	95	67
94	130	133	138	152	144	128	132	118	112	98	83
110	133	125	140	111	94	130	130	117	122	91	93

Raw Digitised Data



A Significant Contours Drawing
of Two Eyes and a Nose



Raw Data with Significant Contours,
Direction and Strength

Personal Research Aims

Jack L. Edwards
York University

I have been immersing myself over the past two years in the literature on natural language understanding systems. I hope to begin work on a system in the next year that will take advantage of the work in this area, computer work in content analysis, and some simulaticn research. The purpose of the programme will be to analyze human interaction from typewritten transcripts. I am also interested in the development of a common language for describing interaction. Although neither of these twc interests are new ones, little groundwork has been laid in either.

A Model and Stack Irplementation of a Conversation between Some Man and a Smart-Aleck Computer

Sydney J. Hurtubise
Dept. of Conceptual Dependency
Faculty of Representation
U. A. I.
North Eay, Ontario

Keywords: Speechax, Cash Register, Lout, Arithmetic Engine, A.I., E.O.

1. Introduction

At the outset of the study of natural language processing, it was hoped that syntax would provide everything necessary for the understanding of an utterance. However, only a subset of everything was accounted for. Syntactic analysis could not handle sentences like "Time flies in like manner to a colourless green idea".

Later, semantic information was added through semantic primitives the number of which, although not exact, was in the right order of magnitude. Clearly, something was missing from this approach as well. Consider the following dialogue between speakers S and A:

S: (lets loose a rat in A's house)

A: "Eh?"

S: "What?"

A: "I believe you telieve I believe."

A simple word for word literal understanding is obviously inadequate to explain the effect that has taken place. Essentially, what is missing is an understanding of the rules and regualations necessary to be a winner at the language game. I call this information speechax.

I will propose a model of dialogue that will take into account language, not only as a form of behaviour, but as an expression of outrage at man's inhumanity to man in general.

2. Divisions of Knowledge

The knowledge used by the system (KUBTS) is divided into four parts (DIFP):

A. Knowledge about the speaker (KATS)

B. Division of general semantics (DOGS)

C. Knowledge about the speechax (KATS)

D. Formulations of direct output replies (FODOR)

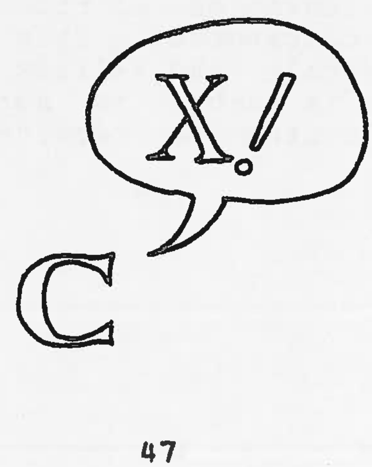
A. Knowledge About the Speaker

This section is concerned with the modelling of the conversant in terms of what he believes, what he believes he believes, what he believes he knows, etc. A major issue to be handled is that of intersection since very often, what a person believes interacts with what he believes he believes. A case in point is the ever-present Henry cartoon character.

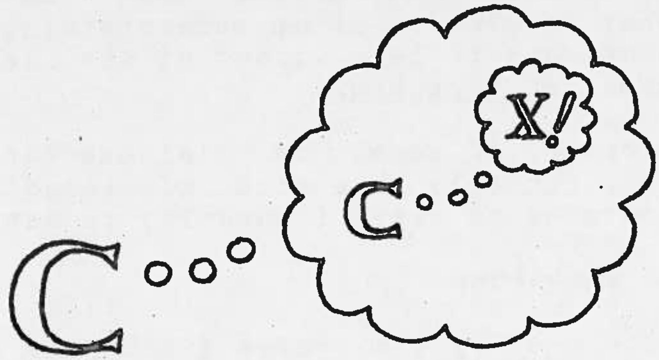
I propose constructing belief clouds which are spaces linked to the conversant via a sequence of linking spaces. Associated with and different from belief clouds are spaces called speakin' clouds which share some, though not all, of the above properties. The span of all relevant speakin' clouds in a particular dialogue is called the speakin' span. Perhaps, a few examples would help.

Example:

"C says X" is represented by



"C believes he believes X" is represented by



In some sense, this representation is more analogue than propositional since in the system it will be stored as pictures on a 256 x 255 grid of 64 intensity values.

B. Division of General Semantics

This section is concerned with the general understanding of utterances at a semantic level. Typically, this will involve a hard-wired semantic network of parallel procedures. Each procedure is linked to every other procedure by an edge labelled "ISA" for purposes of identification.

The understanding of an utterance is accomplished by buck-passing. When a procedure is activated by the appearance of a message, it, of course, doesn't know what to do; so it sends the message to a special procedure called the buck-passer (BP). The buck-passer, being equally uninformed, passes the message on to all other procedures (in accordance with the ignorance principle postualated by McCalla and Kuttner 1975). This is also very similar to the short attention span effect noted by many psychologists.

No explicit context mechanism has been provided for. The reason for this is that within the model of the conversant is the fact that the conversant never utters anything out of context. Hence, the "current context" is the <u>only</u> context.

One of the major innovations of this approach to semantics is the lack of a syntactic component. This is neatly circumvented by the fact that within the beliefs of the conversant is the belief that the system is unable to handle syntax, hence the expectation of the system to receive utterances without any syntactic component.

C. Knowledge about the Speechax

This section is concerned with the conventions of speechax. Basically, this is a production system where each production corresponds to a rule of the language game. The system attempts the first rule, and if it is unable to carry it out, tries the second rule, etc. Typical "rules" might be "Try your best to be polite", "Be alert", "Cleanliness is next to godliness", etc.

In the analysis of the conversant's speechax most of the typed input is ignored since it is well known to psychologists that people don't always say what they mean. As an alternative, visual cues via a television camera, are looked for and the corresponding speechax is inferred. Fairly standard edge detectors and smile finders are used to get an idea of what the conversant is trying to mean. Once a speechax, such as PROMISE, CRY, etc., is determined, the picture of the conversant is modified to include above him the appropriate belief and/or speakin' cloud. Sometimes, expectations are set up as to what speechax the conversant will use next, in which case the system infers that the conversant will fulfill his expectations. This is done in real time for no real reason.

D. Formulations of Direct Output Replies

Finally, this last section is concerned with the actual production of output by the system. Since the system has no way of knowing what hardware will produce the output (e.g. teletype, television, electrical stimulus), it makes every attempt to be brief. The following conversation actually took place between myself as "C" and a prototype system. The time between questions and answers was left out for obvious reasons.

C: "John Smith lives at 65 St. George Street"

S: "Crap"

C: "What is John Smith's address?"

S: "Crap"

C: "Help help help help help"

S: "*** A15 - CAR TAKEN OF FULLCELL"

3. Conclusion

Unfortunately, time and space considerations preclude going into more detail here (a more exhaustive analysis can be found in Hurtubise 1976). Suffice it to say that some of the ideas expressed are tentative and may have to be revised if necessary funds are denied. It is hoped that this work will encourage investigations into abandoning completely the pragmatics of dialogue and conceivably semantics as well.

Acknowledgement:

I am thankful for the preparation of this manuscript.

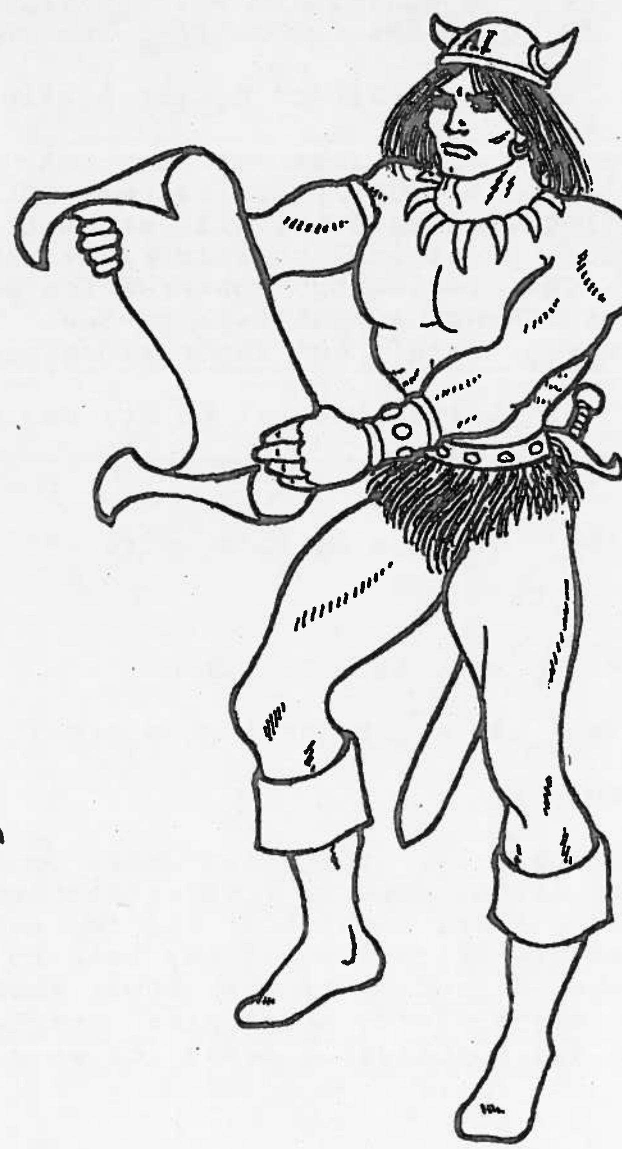
References:

"Henry: An Ontological Viewpoint", Cosmopolitan, 1973.

"Repairing Frames for Fun and Profit", Ladies Home Journal, 1976.

"An Extensible Feature-Based Procedural Question Answering System to Handle the Written Section of the British Columbia Driver's Examination", Second CSCSI/SCEIO Newsletter, G. McCalla and M. Kuttner, 1975.

"A Model and Stack Implementation of a Conversation between Some Man and a Smart-Aleck Computer", Third CSCSI/SCEIO Newsletter, S. J. Hurtubise, 1976.



"I'LL HAVE A HAMBURGER!"

ABSTRACTS

Thematic Map of Larose Forest from ERTS-1/MSS Digital Data

Z. Kalensky and L. Sayn-Wittgenstein
Canadian Forestry Service
Department of the Environment
Ottawa, Canada

One promising way to interpret ERTS data is to use computers for the analysis of digital data recorded on magnetic tapes. Both spectral and spatial characteristics can be used for interpretation. The potential of the method was tested on a forested area near Ottawa.

ERTS Thematic Map from Multidate Digital Images

Z. Kalensky
Forest Management Institute

(Printed from the Symposium Proceedings of the Commission VII (Photo interpretation and remote sensing) of the International Society for Photogrammetry held in Banff, Alberta, Canada, October 7-11, 1974)

Multidate and multispectral ERTS-1/MSS digital images were combined to produce thematic maps of the Larose Forest test area. Two pattern recognition systems based on the maximum likelihood decision rule were used to classify calibrated earth radiance data into seven subscenes:

(a) four-channel system for multispectral classification of images recorded at one date;
(b) twelve-channel system for multidate and multispectral classification of images recorded at three dates.

Classified digital images were recorded on computer compatible tapes and printed by an electron beam image recorder to produce separation film transparencies for color photomaps. Digital images were also printed by a line printer and displayed as computer printouts with alphabetically coded subscenes (agricultural, coniferous and deciduous forest) are presented in summary tables. Overall classification accuracies for multispectral, single-date thematic maps range from 67 per cent to 81 per cent and for the multidate map is 83 per cent.

Accuracy of Forest Mapping from Landsat Computer Compatible Tapes

Z. Kalensky and L.R. Scherk
Forest Management Institute,
Canadian Forestry Service and
Computing Devices Company, Limited, respectively
Ottawa, Ont., Canada

Presented at the Tenth International Symposium on Remote Sensing of Environment, Ann Arbor, Mich., Oct. 6 - 10, 1975

This paper summarizes the results of a two-year study on computerized processing of Landsat-1/MSS digital imagery undertaken jointly by the Forest Management Institute and Computing Devices Company, Ltd. The study examined the applicability of Landsat multispectral images recorded on computer compatible tapes (CCT) to forest mapping. A supervised classification was based on the Gaussian Maximum-Likelihood Decision Rule. The input imagery consisted of CCTs of Landsat scenes and their multidate combinations. Reported are accuracies and consistencies of computerized delineation and identification of the coniferous forest, deciduous forest and nonforest land as a function of the date of Landsat and their multidate combinations. Included also are results obtained from single- and multidate processing of Landsat imagery consisting of only two spectral bands: 5 (red) and 7 (second near-infrared) as well as those obtained when classifying one scene by IMAGE 100.

The overall classification accuracies ranged from 67% to 81% for single-date imagery and were consistently above 80% for multidate combinations.

New accuracy measures called class mapping accuracy and overall mapping accuracy were introduced in addition to the usual presentation of accuracy results in confusion tables. The mapping accuracies facilitate a direct comparison of classification results with their pictorial displays on TV monitors, photographic images and maps.

Computer Acquisition of Natural Language: Experimental Tests of a Proposed System

Janet King, Ian McMaster, and Jeffrey R. Sampson
Department of Computing Science
University of Alberta

(abstract of paper presented at COLING76)

A computer program which could learn to converse in natural language, in a fashion analagous to a child's acquisition of his

native tongue, might prove useful in a variety of research contexts. Linguists could study language acquisition in less capricious experimental settings than those provided by young children. Artificial Intelligence workers could improve and extend their mechanical language understanding systems by means of an acquisition component. As a step toward such goals, a small group of workers at the University of Alberta have been developing and testing a comprehensive programmable model of natural languauge acquisition.

This report begins with a sketch of some highlights of the acquisition process as presently understood by linguists. There follows a brief discussion of key methodological issues, such as the choice of acoustic or orthographic input, the selection of a suitable grammatical formalism, the nature and amount of non-linguistic input, the realism of an external environment, and the role of cognitive development.

Several computer oriented natural language systems with important acquisition components are reviewed, including Schwarcz' (1967) linguistic performance model, Kelley's (1967) hypothesis testing system, and the robot oriented scheme of Harris (1972). Study of these systems can suggest pitfalls to be avoided and successful features to be incorporated in the design of a new model.

A new Complete Language Acquisition Program (CLAP) is proposed. CLAP's major components include a perceiver, Semantic Base, Action Taker, Short-Term Memory, Lexicon, Parser, Responder, and components for modifying the parsing and responding strategies on the basis of experience. The program's grammar is of the augmented transition network (Woods, 1970) variety. CLAP acquires language through the sequential activation of five strategies: segmentation and meaning association, linear ordering, structural generalization, conflict resolution, and using discourse. Each strategy is described in some detail to show, among other things, how CLAP emphasizes the primacy of comprehension over production and the role of a realistic external environment (two features lacking in most of CLAP's predecessors).

At least the first three of CLAP's strategies are sufficiently well defined for immediate implementation, using established methodologies from artificial intelligence and computational linguistics.

Since most of the above research is thoroughly discussed elsewhere (MccMaster, Sampson and King, 1976), the balance of this report is devoted to recent attempts to test aspects of CLAP. Results are now available from two experimental implementations of part of CLAP's first strategy (segmentation and meaning association). Both systems were programmed in MACLISP.

The first vocabulary acquisition system correctly learned the meanings of many object names contained in natural discourse about Winograd's blocks world environment. Two sets of utterances were used as input data. The first corpus was produced by an adult speaker who knew nothing about the system. The second corpus was contrived by the experimenter to provide maximal learning opportunities (consistent with natural discourse). Results were somewhat better for the second corpus.

The program detected correlated occurrences of lexical items in input sentences and objects in an experimenter designated focal region of the environment. Algorithms were employed for expanding the focal region and generating and adjusting weighted links from words to concepts. Incorrectly learned words served to illustrate three basic kinds of errors the system could make, some of which seem to parallel errors made by children when learning to talk.

The second experimental study introduced actions, and the verbs describing them, as potential material for the acquisition routines. Two corpora were again used, the first consisting of a mother's actual remarks to her child as they observed the environment. The second corpus was again experimenter contrived.

Where the first study used Winograd's predicate calculus type of representations for environmental foci, this experiment adopted a case-grammar type of viewpoint, strongly influenced by Schank's (1973) theory of the components of a conceptualization. The system used just four primitive ACTs: PTRANS, PROPEL, GRASP, and MOVE. Each element of a conceptualization had its own internal representation, as a data item, procedure, or attribute determined by a procedure. Correlations were formed as in the first experiment.

Results of the second study raised a number of important questions about the overall methodology, including the assumption that structural morphemes would develop no meaningful concept connections. Also, new methods were required for dealing with multiple utterances pertaining to the same focal region. Despite these problems, the second experimental system confirmed the success of its predecessor by correctly learning many lexical items, including those for most actions which could be associated with a single word. Also, as in the earlier results interesting comparisons could be made between incorrectly learned correlations and similar mistakes made by children.

To date, then, CLAP has been neither vitiated nor totally supported by experimental studies using fragmentary implementations. Further research will focus initially on implementation of the segmentation aspect of CLAP's first strategy.

REFERENCES

Harris, L.R. A model for adaptive problem solving applied to natural language acquisition, Ph.D. Thesis, Cornell University Ithaca, New York, 1972.

Kelley, K.I. Early syntactic acquisition, Report no. P-3719 Rand Corporation, Santa Monica, California, 1967.

McMaster, I., Sampson, J.R., and King, J.E. Computer acquisition of natural language: A review and prospectus. International Journal of Man-Machine Studies, in press, 1976.

Schank, R.C. Identification of conceptualizations underlying natural language. In R.C. Schank and K.M. Colby (eds.), Computer Models of Thought and Language, San Francisco: W.H. Freeman and Co., 1973.

Schwarcz, R.M. Steps towards a model of linguistic performance: A preliminary sketch. Mechanical Translation, 10, 39-52, 1967.

Winograd, T. Understanding Natural Language. New York: Academic Press, 1972.

Woods, W.A. Transition network grammmars for natural language analysis. Communications of the ACM, 13, 591-606, 1970.

ANTICS - A System for Animating Lisp Programs

Mark Stephen Dionne
Dept. of Computer Science
Univ. of British Columbia

An animated film is an excellent medium for conveying complex ideas in computer science. A system has been developed which produces animated films, film strips, or slides depicting the execution of LISP programs. The design and implementation of this system is discussed and it is compared to existing systems.

The system, named ANTICS, may be used by entering very simple commands, and produces real-time animation. The system may be "backed up" interactively and atomic values may be changed. Advanced commands and a set of graphics primitives are available which permit an instructor or film-maker to control minute details of the animation and to add features not explicitly provided by the system. ANTICS may therefore be used as an interactive educational tool or as an animation system.

ANTICS is very economical to use. A three minute film showing the operation of the recursive function MEMBER was produced for a total cost of $12.00. The film is included as part of the thesis. The system is dependent on IBM 370/168 and Adage Graphics Terminal hardware, but the design, which is based on the organization of the LISP EVAL function, could be used on other systems.

Consistency in Networks of Relations

Alan K. Mackworth
Dept. of Computer Science
Univ. of British Columbia

(Tecnical Report 75-3)

Artificial Intelligence tasks which can be formulated as constraint satisfaction problems, with which this paper is for the most part concerned, are usually solved by backtracking. By examining the thrashing behaviour that nearly always accompanies backtracking, identifying three of its causes and proposing remedies for them we are led to a class of algorithms which can profitably be used to eliminate local (node, arc and path) inconsistencies before any attempt is made to construct a complete solution. A more general paradigm for attacking these tasks is the alternation of constraint manipulation and case analysis producing an OR problem graph which may be searched in any of the usual ways.

Many authors, particularly Montanari and Waltz, have contributed to the development of these ideas; a secondary aim of this paper is to trace that history. The primary aim is to provide an accessible, unified framework, within which to present the algorithms including a new path consistency algorithm, to discuss their relationships and the many applications, both realised and potential, of network consistency algorithms.

How to See a Simple World

Alan K. Mackworth
Dept. of Computer Science
Univ. of British Columbia

(UBC Technical Report 75-4, also presented at the NATO Advanced Study Institute on Machine Representations of Knowledge, Santa Cruz, California, July 1975.)

This is not a comprehensive survey of machine vision which, in its broadest sense, includes all computer programs that process pictures. Restricting attention to scene analysis programs that interpret line data as polyhedral scenes makes it possible to examine those programs in depth, comment on revealing

mistakes and anomalies, explore the interrelationships and exhibit the thematic development of the field. Starting with Roberts' seminal work which established the paradigm, there has been an evolutionary succession of programs and proposals each approaching the problem with a different emphasis. In addition to Roberts' program this paper expounds in detail work done by Guzman, Falk, Huffman, Clowes and Waltz. These programs are presented, compared, contrasted and criticized in order to exhibit the development of a variety of themes including the representation of the picture-formation process, segmentation, support, occlusion, lighting, the scene description, picture cues and models of the world.

Artificial Intelligence and Linguistics
A Brief History of a One-way Relationship

Richard S. Rosenberg
Department of Computer Science
University of British Columbia

(from Proceedings of the First Annual Meeting of the Berkeley Linguistics Society. February 15-17, 1975 Berkeley, California.)

For the past fifteen years there has been a serious interest in the processing of natural language (English) by researchers in Artificial Intelligenece (A.I.). This processing has included machine translation, question-answering systems, man-machine dialogue, and speech understanding. This keen interest has engendered an awareness of and concern with the ongoing activity in contemporary linguistics. Therefore, it may be of interest to linguists to discover what has been adapted and used. As is often the case when two fields interact, the borrowings of one have usually been late and have ignored the unsettled state of theories in the other. Thus a brief history of the relation (almost always one-way) between A.I. and linguistics is presented.

Some of the works in A.I. surveyed range from those of the early sixties such as Lindsay's SAD SAM, Green et al BASEBALL and Bobrow's STUDENT to more recent efforts including Wood's transition network grammars, Winograd's SHRDLU, and Schank's conceptual dependency models.

In one way or another, these computer programs and others depend upon the work of Chomsky, both <u>Syntactic Structures</u> and <u>Aspects</u>, Halliday's systemic grammar, and some of the ideas of generative semantics as developed by G. Lakoff, McCawley, and Fillmore.

Interpretation Directed Segmentation of ERTS Images

D.W. Starr and A.K. Mackworth
Department of Computer Science
University of British Columbia

(To be presented at ACM/CIPS Pacific Regional Symposium, June 1976)

Automatic interpretation of images from Earth Resources Technology Satellite 1 (ERTS-1) can be used in a variety of applications with considerable accuracy. Most systems however, classify strictly on a point by point basis, with no use of any spatial knowledge. Standard photo-interpretation techniques are combined with some techniques from Artificial Intelligence to produce an increase in accuracy over a point-by-point classification method. Traditional classification methods are used to obtain an initial segmentation of the image. Then, a controlled region merging process allows the regions with unambiguous interpretations to influence the interpretation of neighbouring regions, thereby introducing considerable context sensitivity into the interpretation process. Results are given of an experiment to interpret areas of different forest cover.

Key Words and Phrases

Multi-spectral scanner, ERTS, maximum likelihood function, Artificial Intelligence, scene analysis, region merging, resource inventory.

A Word Based Bi-directional Speech Parser

Corot C. Reason
Dept. of Computer Science
University of Toronto

This paper describes the design of a speech parser that is primarily syntax directed but whose performance is augmented by semantic information made available via techniques suggested by Riesbeck. Additionally the parser operates bi-directionally which allows for a certain amount of flexibility during the course of a parse. An implementation of the design has been completed, however the program deals with just a small vocabulary and grammar and is only a first approximation to a full speech parsing system.

The parser is driven by data attached to the words in the lexicon, the most important of which are verbs. Attached to verbs are structures called "assemblies" that specify case frames that the verb may appear in. If a verb is located in a speech utterance, the parser strives towards filling in the arguments of the case frames. Assemblies may be only partially completed at a given point in the parse and yet, if certain boundary conditions are met, the part of speech the assembly is building (say a relative clause) may be used to fill arguments in other assemblies.

Naturally, location of a verb in an utterance may be a rare occurrence. All words in the lexicon have information that can guide the parser to find neighbouring words. For instance the location of an adjective may lead to the discovery of an entire noun phrase including a relative clause.

The parse begins by locating stressed syllables in the utterance. The stress information is used to build a list of words likely to appear in the stressed area. The parser then tries to satisfy the expectations about the area in the utterance local to the words, as suggested by the data attached to the words in the list. A less subtle starting strategy is used if no real progress is made using these words.

Since the parsing of left or right recursive language constructs in a bi-directional manner leads to difficulties, small pitch-pause anomalies in an utterance may be of help in marking the end of major syntactic units. Empirically, speakers do insert such patterns, although in an inconsistent manner. The expectation of these anomalies can be included in assemblies to help disambiguate the parse of complicated language structures.

At any stage a complete or incomplete assembly can be packaged and handed to a semantics unit for evaluation as a theory about the utterance content. Part of the work the parser must do involves choosing the most reasonable assemblies to work on. In fact the parser may arrive at two complete parses for an utterance and let a higher level process (such as a semantics unit) decide as to which seems to be correct.

An Experimental Evaluation of Chess Playing Heuristics (M.Sc. Thesis)

Laszlo Sugar
Dept. of Computer Science
University of Toronto

This thesis outlines the results of research into chess playing program evaluation. A methodology for experimentally determining chess heuristic performance is described. It is based on statistical analysis of heuristic "reactions" to pre-scored moves. Application of this methodology resulted in the creation of a library of master games and the evaluation of the components of a specific chess program, CHUTE. The results

identified problem areas in the program; for example, initial move selection missed the "ideal" move 69% of the time.

An analysis of error bounds on lookahead strategy and move selection is also presented. This analysis provided the basis for the interpretation of the experimental results.

A Prototype Motion Understanding System

J.K. Tsotsos
Dept. of Computer Science
University of Toronto

This thesis presents a design and implementation strategy for a visual motion understanding system. A representation for motion concepts is described. This representation includes many of the semantic components found necessary in order to describe motion satisfactorily. Secondly, a design proposal is presented for the computer system. This system uses the representation determined in order to compute semantic structures containing English motion terms such as verbs and directionals, which describe the movement of the objects observed from image sequences. Finally, the current implementation progress of the system is briefly described and examples of its operation are given.

A Survey of Data Semantics

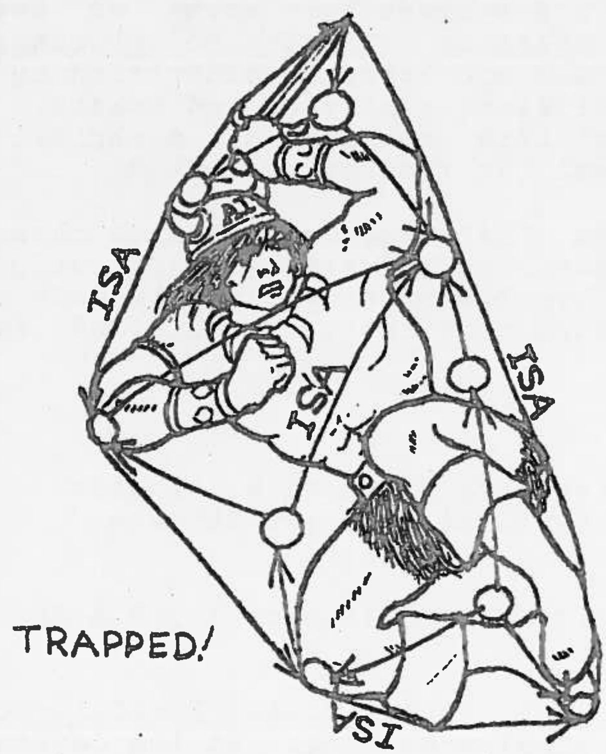
Harry K.T. Wong
John Mylopoulos
U. of Toronto

Recent work has demonstrated that the three major data base models (hierarchical, network, relational) are lacking in one way or another, in dealing with certain difficult, semantic problems. Some of these problems are : the relational schema generation; the semantic integrity of the data base and the decision of when a query is meaningful etc. Within a span of 4 years, a number of data base models have been developed (1,2,3,4) to deal more adequately with the representation of the semantics of data. Examined more closely, these models exhibit certain similarities with corresponding models in AI. Most of these models fall into one of the three main representations used in AI (Predicate Calculus, Procedural representation, Semantic network). A lot of these efforts would have been spared had there been better communications between the two research areas. Hence, it is the major goal of this paper - to bridge the gap between the AI and data base literature so that data base researchers will know what AI has to offer and vice versa.

The paper is organized as follows: The three basic AI approaches to representation are examined feature by feature. The uses of each feature are discussed, examples from both AI and data base models adopting that particular feature are given and models where such a feature is absent are discussed regarding their inadequacies. To give an example, for semantic networks, the following features are discussed: the distinction between objects and relationships; the ISA hierarchy; the representation of quantifications; the distinction between generic and specific concepts etc. The first feature is used to provide some capability of maintaining the consistency of the data base during deletion and addition. Because of the omission of this feature, Codd's relational theory is then shown to be lacking from a representational point of view. Data base models having this feature include (1,3,4). The bulk of the paper consists of this kind of feature analysis, examples and comments. The last section of the paper summarizes and discusses open problems.

References

1. Senko, M.E., Altman, E.B., Astrahan, M.M. and Fehder, P.L. (1973) "Data structures and accessing in data-base systems", IBM Systems J. 1973, pp. 30-93.
2. Chen, P. (1976). "The entity-relationship model: toward a unified view of data", TODS vol. 1 no. 1.
3. Bracchi, G., Paolini, P and Pelagatte, G. (1976). "Binary logical associations in data modelling", Proc.IFIP-TC-2-WC, Black Forest, 1976.



COURSES

University of Alberta

The Department regularly offers three courses with substantial AI content. In the past year, Dr. Jeff Sampson taught CS464: Introduction to Adaptive Systems, Dr. Len Schubert gave CS551: Artificial Intelligence, and Dr. Kelly Wilson offered CS665: Seminar in Artificial Intelligence.

CS464 is a course designed to introduce 3rd year honors students to a variety of interrelated topics in the theory and applications of computing, topics that are often not encountered until graduate studies, if at all. Approximately the last 35% of the course consists of an introduction to the major areas of AI research, including pattern recognition, game playing, theorem proving, problem solving, and natural language processing. Other portions of the course deal with selected topics in information theory, automata and biological information processing systems. Sampson's new text for the course, Adaptive Information Processing, will be published this summer by Springer-Verlag.

CS551 is a first course in AI for graduate students (and 4th year undergraduates who have CS464 as a prerequisite). Three broad topics are covered: (1) problem solving, including detailed study of A*, MULTIPLE, resolution systems, GPS, and STRIPS; (2) planning, both as manifested in the above problem solving systems and in AI languages and systems like those of Hewitt, Sussman, and Fahlman; (3) semantic information processing, including detailed study of scene and language understanding systems like those of Winston, Quillian, Winograd, and Schank. In the course, Schubert emphasizes both mastery of technical detail and appreciation of underlying theoretical ideas.

For this year's CS665 seminar, Wilson chose the topics of natural language processing and memory organization. There was some time spent on hypotheses regarding the mode of operation of human memory, including the reorganization of material during consolidation.

University of British Columbia
Dept. of Computer Science

Artificial Intelligence I (CS 502)

As an introduction to Artificial Intelligence, this course will concentrate on problem-solving. In the course of examining a variety of problem-solving systems we will study the evolution of the main issues: task specifications, search, problem reduction, advice, learning, planning, analogy, generality/competence/efficiency tradeoffs and the central topic of AI the representation and the use of knowledge. A variety of schemes for knowledge representation will be discussed including the following: state spaces, evaluation functions, programs, networks, production systems, assertional data bases and the first order predicate calculus.

As CS512 LISP-based Symbolic Computation is not offered this year, material on AI programming languages will be interspersed at appropriate points through this course and its successor in the spring, CS 522 Artificial Intelligence II. In this course students will be expected to complete exercises in LISP and MICRO-PLANNER in problem-solving, graph-searching, pattern-matching and theorem proving.

Students will also be required to write and present a critical exposition of a major AI program or theory.

Texts:
Nilsson, N.J. (1971) Problem-solving Methods in Artificial Intelligence, McGraw-Hill.

Documentation of LISP/MTS and MICRO-PLANNER will be made available.

References:
Feigenbaum, E.A. and Feldman, J. (eds.) (1963) Computers and Thought, McGraw-Hill.

Minsky, M. (ed.) (1968) Semantic Information Processing, MIT Press.

Artificial Intelligence II (CS 522)
A. K. Mackworth

This course will continue the twin threads established in CS 502, namely, schemes for representing knowledge and their implemenation in AI programming languages. The main goal will be to understand current attempts to develop a so-called representation theory: a meta-theory of theories for the representation of knowledge for perception, inference and action.

The original AI quandary was: are there any ways in which this task can be automated? Now that we have partial answers to that question for many limited task domains, the current AI quandary is : what framework do we use to judge this embarrassingly large number of representational schemes? Furthermore, with such a framework available, how can we then develop the schemes we have in more systematic ways? And, finally, what then are the implications for AI programming languages?

In addition to the schemes discussed in CS 502, some of which will be further developed, we will consider feature spaces, Bayesian decision theory, grammars, analogical schemes, schemes for plans, frames (a.k.a schemata) and the actor formalism. For the sake of concreteness, a convenient area in which to develop most of these notions and those of representation theory itself (which consists, at the moment, of identifying adequacy) is computational vision. Many of the examples will be taken from that area but others from robot planning, problem solving and natural language will also be used.

A programming language development will be tied to shifts in representational schemes. LISP, Conniver, the Planner project and frame languages will be paid most attention.

Students will be required to design, implement and present a project touching on some of the issues of the course.

TEXT: None

Prerequisite: CS 502

Computational Linguistics I (CS 503)
R.S. Rosenberg

Course description

This course will be directed towards the study of natural language understanding from a computational point of view. However, because of limited time (and interest) , we will not be concerned with such areas as content analysis, statistical approaches to language or the like. Rather the major effort will be devoted towards the exploration of natural language understanding within the question-answer framework. As such the course will be a mixture of current linguistic theory (Chomsky, Fillmore, Lakoff) and various computational models (Woods, Winograd, Schank, etc.).

The basic aim is to provide a reasonable overview of current research in computational linguistics. The basic method is to use a mixture of lectures on a variety of theories and approaches coupled with a major ongoing project. It is hoped that exposure to a number of approaches as well as the invaluable experience to be gained by an actual in-depth implementation will result in a healthy respect for the extraordinary complexity of natural language.

Project Description

The course has been offered on three previous occasions with the following projects:

1) A system to query a data base of LISP programs.

2) A system to carry on a dialogue (questions, commands) with a robot in a Winograd type "blocks world".

3) A system to act as a command and query language for MTS.

For this term, I would like to use chess as the domain of interest. The natural language system will use English as a command and query language for chess boards. A user should be able to cause the system to carry out various tasks, such as moving pieces about (legally, of course), and to determine any aspect of the current board, using ordinary English. The system is to respond to the user's inputs by changing the board configurations and by returning answers in response to questions.

Typical inputs might be:

1. How many white pawns are there?

2. Does black have two rooks?

3. Move the white queen to B6.
   (You will have to adopt some chess notation to indicate piece positions. The system should check on the legality of the move as well.)

4. Is there a black bishop which is under attack by a white piece?
   (This is a difficult sentence and obviously requires appropriate chess knowledge about "attack". Grammatically this is also a passivized relative clause.)

5. Does black have more pieces than white?
   (Note use of comparative here. Also, in these sentences the words black and white are used both as nouns and adjectives.)

6. Is there a black knight next to a white piece?
   (There is a need for spatial knowledge.)

7. Remove the white's bishop (with black's queen).
   (Note the use of possessives, the problem of choosing which white bishop, and the occurrence (or lack of occurrence) of the phrase "with black's queen". The resulting board should be displayed.)

The foregoing description should be taken as a preliminary one. The problem will be progressively refined. The class will be divided into groups of students, each group responsible for an implementation. All systems will be implemented in LISP using the Woods' ATN grammar and probably MICRO PLANNER.

Computational Linguistics II (CS 523)
R. S. Rosenberg

We will be concerned in this course with the problem of analysing human conversation. Much of the effort in Artificial Intelligence (AI) and Computational Linguistics (CL) over the past few years has been devoted to the development of question-answering (Q/A) systems. Such systems are characterized by the fact that they deal almost entirely with isolated question-answer pairs. This somewhat artificial situation ignores most of the features of human communication. If the primary goal is to study the nature of natural language processing, it is important to be as aware as possible of the dimensions involved.

We wish to survey a number of approaches from different areas including philosophy, linguistics, psychology, sociology, and of course AI. The limited time available means that in some instances only a brief examination will be possible. The following list of areas, topics, names, etc. is illustrative rather than definitive.

Philosophy of Language

Austin, John — How to Do Things with Words
Performative-constative

Searle, John — Speech Acts
What is a Speech Act
A Classification of Illocutionary Acts
Indirect Speech Acts

Grice, Paul — Logic and Conversation
Various papers on Meaning

Putnam, H. — The Meaning of "Meaning"

Linguistics

Selected papers from such linguists as: C. Fillmore, G. Lakoff, J. McCawley, W. Chafe, J. Ross, S. Anderson, B. Fraser, L. Karttunen, especially those concerned with performatives, conversational postulates, implicatures, and presuppositions. Other papers will be found in various volumes of the Chicago Linguistic Society Meetings.

Psychology and Sociology

Goffman, Erving — Relations in Public Frame Analysis

Cicourel, Aldo   <u>Cognitive Sociology</u>
Berne, Eric   <u>Games People Play</u>

Bateson, Gregory   <u>Steps Toward an Ecology of Mind</u>

Selected papers of D. Norman and D. Rumelhart, W. Labov, E. Rosch, S. Fillenbaum and D. Slobin

<u>Artificial Intelligence and Computational Linguistics</u>

Winograd, T. Notes from a course on computational models of semantics.

Schank, R. and Colby, K. <u>Computer Models of Thought and Language</u>.

Rustin, R. <u>Natural Language Processing</u>

The work of R. Schank and his students, C. Rieger, C. Riesbeck, and N. Goldman, of W. Woods and his associates at BBN, and of Winograd, Bobrow, and Kay at XEROX PARC.

This list represents a wealth of material which, as mentioned previously, we will only begin to explore. The course will require extensive reading and presentation of materials. Each student will also write a paper or a program, if that seems appropriate. Participation by students from the departments of linguistics, philosophy, and psychology, is encouraged. Finally, a number of questions and problems which may set the tone for this course are given:

How is it possible to limit a domain appropriately and still adequately study the interesting aspects of human conversation?

What is the relation between logical and non-logical reasoning in linguistics? Can these two be distinguished?

What is the role of imagery in linguistic processing?

Can current notions in the philosophy of language be "turned into programs"?

What are semantic nets and how are they used to represent knowledge?

What are some of the factors involved in human conversation?

University of Toronto
Dept. of Computer Science

Artificial Intelligence (CSC 2505F)
G. McCalla

Prerequisite: an undergraduate A.I. course or consent of instructor

Two of the most basic concerns of AI have been representation of knowledge and problem solving. Recently there has been a trend towards viewing these seemingly separate areas as two aspects of a broader paradigm. This course looks at various representation schemes, considers the important problem solving methodologies and the programming languages useful for implementing them, and finally analyzes the current attempts to construct a more unified theory.

Course outline:

Introduction:
what is A.I. and why study it?

Representation I:
ad hoc systems, predicate calculus, assertional representations, semantic networks

Problem Solving:
heuristic search, resolution, GPS, PLANNER, STRIPS

AI Programming Languages:
MICROPLANNER, CONNIVER, POPLER 1.5, 2.PAK

Representation II:
procedures, procedural/declarative controversy, procedural networks, "high level" constructs (frames, scripts, etc.), analogical representations

Conclusion:
where do we go from here?

Topics in Computational Linguistics (CSC 2528S)
R. Perrault and G. McCalla

Prerequisite: CSC 2505 or consent of instructors

This course considers some current topics of interest to the computational study of language. It is expected that the participants will have a reasonably sound background in A.I.

Course Outline:

The MARGIE System:
the problem of primitives, conceptual dependency, Rieger's inferences

High Level Representations:
episodes (Schank), scripts, frames

Philosophy of Language:
historical (Frege, Russell, Wittgenstein), intentions (Grice), speech acts (Austin, Searle)

Planning:
GPS, STRIPS, NOAH (Sacerdoti)

Sociolinguistics:
issues and directions

Applications of Artificial Intelligence
J. Mylopoulos

Prerequisite: CSC 2505 or consent of instructor

General goal:

This course is intended to describe and analyze some of the more successful and/or promising applications of A.I. to other research areas, mostly within Computer Science.

Specific goals:

Many different types of systems use large amounts of information about a physical or abstract "world". For example, a CAI system for teaching grade 11 mathematics uses information about grade 11 mathematics, a medical diagnosis system uses information about diagnosis, a DBMS uses information about a data base, a compiler uses information about a programming language, etc. Let us call the information about a world used by a system a "knowledge base" (KB).

The goals of the course are to study the design, implementation, and use of knowledge bases, abstractly as well as through existing examples.

Course Outline:

Introduction

Knowledge Bases:
modelling a world, implementing a knowledge base

In our first attempt a group of 6 students decided to tackle the blocks micro world of Winograd's SHRDLU. The project divided roughly into the parsing phase, the semantic phase (in which command strings in an interlingua were generated) and a world model phase (in which the interlingua is interpreted, anaphoric referents are assigned, extensions of definite descriptions are identified in the world, and the required action is carried out subject to the simulated restrictions of the micro-world)

The parsing phase primarily involved becoming familiar with Woods' ATN system, making existing algorithms run efficiently on our POP-2 based system, and making small modifications to a large grammar borrowed from a U.B.C. thesis by Jean Jarvis. Innovations here included a translation system (a sort of compiler-compiler) which reads an arbitrary grammar network in conventional ATN notation (which included LISP code), automatically translates the whole network into a POP-2 program and then compiles the program on top of a run-time package to produce a fast and compact parser (currently using 39K of PDP-10 core). Another modification was the addition of an "interrupt" facility in the parser which allows it to recognize idiomatic expressions, equivalent forms, and even system control words without a preprocessing stage.

The design of the semantic component was of course critically dependent on the interlingua into which we were parsing and that in turn depended on the design of the world model. Here close cooperation among students was mandatory and much of the higher level decision making was done in regular classroom workshop sessions (which alternated with lectures on linguistic and A.I. topics). The final design reflected our reliance on Julian Davies' incomparable POPLER 1.5 system (an implementation of all of Hewitt's PLANNER plus CONNIVER and various other facilities thrown in!). POPLER's actor facility led us to a command language (or interlingua) which was so English-like it was eerie. In retrospect I'm not sure that the choice of obvious English words to name actor routines didn't create a false impression that the parser was just simplifying the input sentences and putting in parentheses! For example the interlingua form corresponding to sentence (a) is given in (b):

(a) pick up the large red block in the box.
(b) (ACHIEVE *GRASP*(THE)(LARGE)(RED)(BLOCK)(IN*(THE)(BOX)*)**)

A nice consequence of this formalism (in which objects are characterized by stringing together "constraint lists" of actor forms) is that it allowed rather simple rules of composition. When in the course of a parse a noun is encountered, a semantic structure consisting of an actor (such as "(BLOCK)") is inserted. If a noun phrase consisting of adjectives and a noun is parsed a string of actors--one for each lexical item other than quantifiers and other "closed class" words--is simply concatenated together. The special actor "(THE)" signals the requirement that the constraint list have a unique extension (if it in fact does not have a unique extension the anaphora mechanism is evoked to search for recent mentions). Similarly if a noun phrase has an embedded prepositional phrase, actors forms are again simply strung together (as with "IN" in example (b) above). The only exception to this simple composition-by-concatenation rule (at least in our microworld) occurs with embedded sentences (e.g., relative clauses). In this case the whole embedded sentence is not an independent constraint but rather something being asserted of one of the noun phrases in the embedded sentence constrains a matching noun phrase in the higher level sentence. In other words we have the problem of binding an inner and an outer noun phrase so that constraints on one are added to constraints on the other. This is accomplished by using a pair of reserve words: an actor "SUCHTHAT" which dominates a variable "THATTHING" in the embedded sentence. Embedding may be nested so that we get constructions like (d) from sentences like (c) (taken from an actual run):

(c) Where is a box which is supported by a table which is avocado?
(d) (INFER*LOCATE*(BOX)(SUCHTHAT*(THATTHING)(ABOVE*(TABLE) (SUCHTHAT*(THATTHING)(AVOCADO)*)*)*)**)

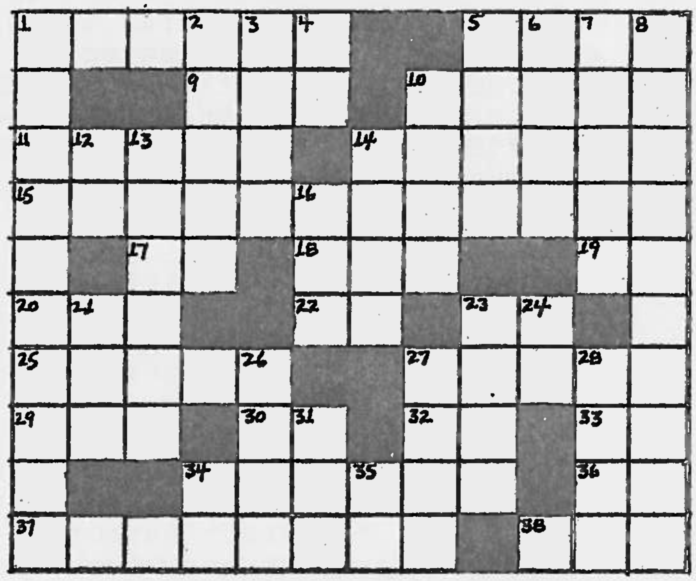
In our current system case grammar is not used and all anomalies (even ones that a case grammar would treat as a syntactic violation) are detected by examining the interlingua string either for the wrong number of arguments to a key action (e.g.,"move", "locate") or for the absence of a permissible referent for a constraint list in the world model. These result in a failure back to the parser and a search for a new parse. A variety of anaphoric references are handled by keeping a list of recently mentioned objects and their descriptions (If this does not resolve the uncertainty of the system will say, "I don't know which...you mean").

The use of POPLER for implementing the world model makes a number of functions easy and elegant to implement. For example the context mechanism would make it easy to arrange for it to keep track of its history (to answer "when" and "why" questions). This has not yet been done.

The above was a necessarily brief outline of a project accomplished in just a few months by students new to A.I. A few years ago it would have taken perhaps years to do the same thing. Not only were students able to produce a rewarding piece of work but, perhaps even more importantly, they ran into very many of the difficult problems of A.I. and language analysis in the process. Discovering the limitations of their system gave important insight into state-of-the-art problems in the field. Simply lecturing about such problems would have been no substitute for the first hand experience.

The First Annual Super-Duper CSCSI/SCEIO Crossword Puzzle

Prepared by A.I. Man
(with the help of G. McCalla and L. Melli)

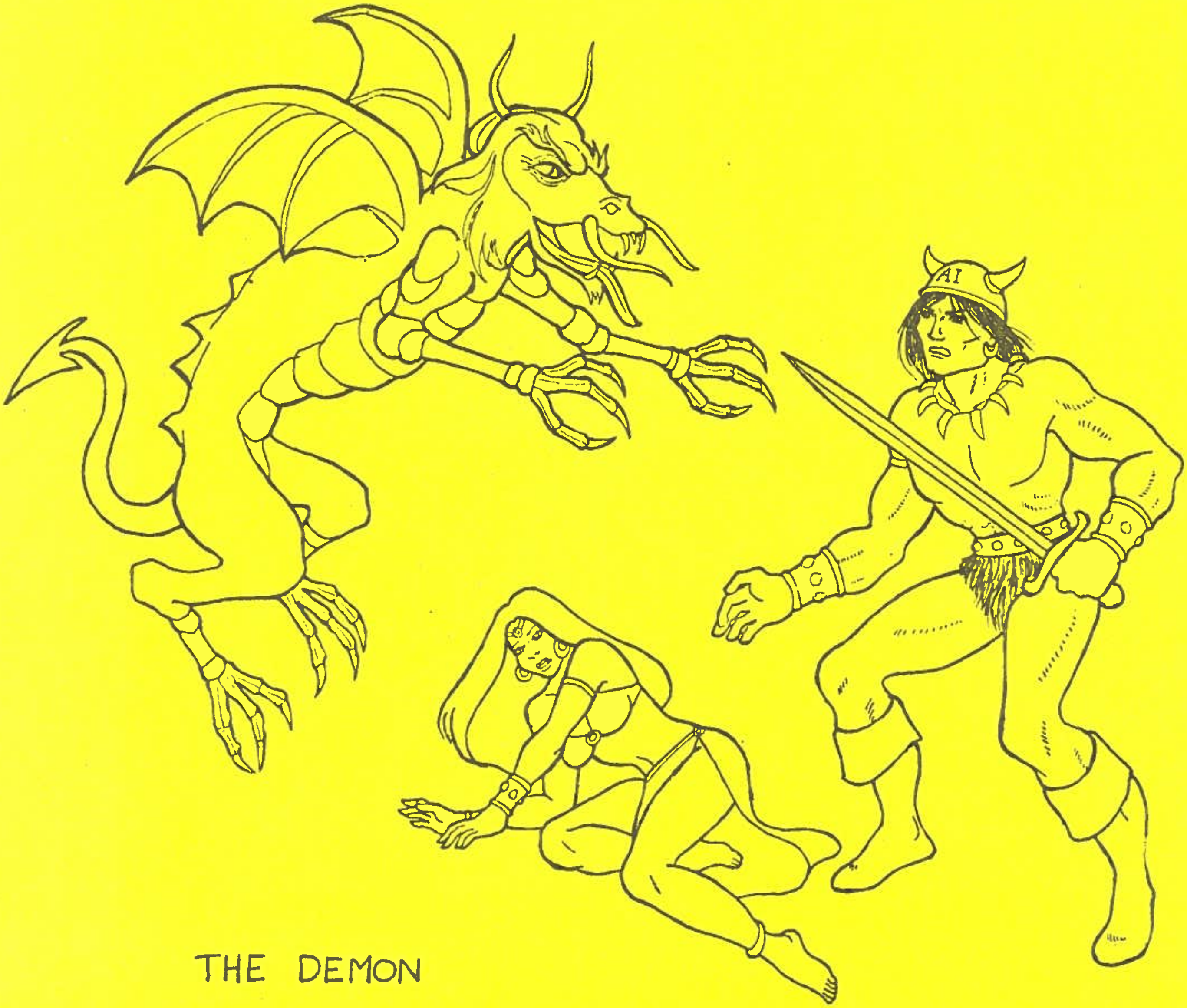


Across:

1. Best friends of 26 down (6)
5. Sacerdoti didn't need an Ark (4)
9. The brand on most AI solutions (3)
10. He Drew it (5)
11. A head piece which isn't at the tip of South America (5)
14. One way of labelling information (4,2)
15. Half of the AI pursuit (12)
17. Research and development (abbr.) (2)
18. Turing, Mackworth, Newell, etc. (3)
19. Royal initials (2)
20. The relationship between Clyde and elephant (3)
22. In AI hot air often expands to this (2)
23. A prominent West Coast AI facility (abbr.) (2)
25. Dreyfus is one such (5)
27. To cut down the branches (5)
29. Information management system (abbr.) (3)
30. Not Washington power (2)
32. The opposite of AI (2)
33. A well known logical formalism (abbr.) (2)
34. Set up by Minsky (6)
36. Expression of satisfaction (2)
37. What to do with side-effects (English spelling) (8)
38. Name of a friendly French girl (3)

Down:

1. The other half of 15 across (10)
2. Where is the green block? (2,3)
3. I declare! A number! (4)
4. Region of U.S.A. containing LNR group (abbr.) (2)
5. Not much AI done here in Alaska (4)
6. List of potential successor nodes (in book of 14 down) (4)
7. It took a good planner to quote her (5)
8. Not hierarchy (10)
10. Does Fahlman leave marks to identify dogs? (4)
12. (-- BLOCK BOX) (2)
13. Abstractly speaking, how does Schank go there? (6)
14. More than one falsity (4)
16. MIT place to do AI (3)
21. Recipe for a new atom: GEN--- (3)
23. How you communicated things to a speech understanding system (4)
24. ISA direction (2)
26. To him Shakespeare is right: all the world really is a stage (4)
27. A logical growth (4)
28. Newell's was a discriminating design (4)
31. SCHOLARly education (3)
34. The music sounds like it is far (2)
35. Ph.D. dissertation in the raw (2)



THE DEMON