Canadian Artificial Intelligence

Intelligence Artificielle au Canada

January/janvier
1987 No. 10

An official publication of CSCSI, the Canadian Society for Computational Studies of Intelligence

Une publication officielle de la SCEIO, la Société canadienne pour l'étude de l'intelligence par ordinateur

The Symbolics 3600— a new computing paradigm

Processing with symbols vs. computing with numbers. The first computers crunched numbers. And regardless of how powerful they've become, traditional systems still force you to deal with the world in quantitative terms.

Face sophisticated applications, and the limitations can become all too obvious.

An increasing number of computer scientists, researchers and program developers are discovering ways to break through this complexity barrier. Their vehicle—the Symbolics™ 3600.

The 3600 allows talented programmers and engineers to represent objects and knowledge far more flexibly than numeric formats allow.

Through the dynamic manipulation of arbitrary data structures consisting of symbols and their associated properties, a user can resolve problems previously considered impossible or infeasible.

SYMBOLIC PROCESSING WAS DEVELOPED TO SOLVE TWO KINDS OF PROBLEMS:

THE COMPLEX. AND THE IMPOSSIBLE.

A few typical applications. Custom VLSI engineering. The 3600 has a unique ability to deal with large, complex modeling structures. Semiconductor companies use it to assist in the development of the next generation of chips.

A development environment for complex software. Rapid prototyping and an incremental edit/compile/dynamic link/debug loop help make the 3600 one of the most effective programming environments ever developed.

Expert systems development. Using the 3600's powerful inferencing throughput and ability to handle very large knowledge bases, government agencies and Fortune 1000 companies are developing expert systems in such fields as process control, financial advisory services and image understanding.

Symbolics—the first name in symbolic processing. Symbolics was founded in 1980 to commercialize this new technology. Among the founders were the developers, at MIT, of the first hardware architecture designed specifically for symbolic processing.

Today, the 3600 represents the highest expression of symbolic processing technology. Its custom processor design incorporates a tagged memory architecture that manipulates symbols as efficiently as a traditional computer manipulates numbers.

Using Symbolics-enhanced Lisp, the 3600 provides a powerful integrated software environment which features object-oriented window system, incremental Lisp/Fortran-77/Pascal compilers and exceptional networking and graphics capabilities—for a new dimension in man-machine interaction.

To learn more about symbolic processing, write us at the address below.

symbolics (Canada) inc.
A Member of the NEXA Group

5915 Airport Road, Suite 200,
Mississauga, Ontario L4V 1T1
(416) 671-0510

Canadian Artificial Intelligence

Intelligence Artificielle au Canada

January 1987 No. 10 (New series)

ISSN 0823-9339

Senior Editor / Rédacteur:
Graeme Hirst, PhD
Department of Computer Science
University of Toronto
Toronto, CANADA M5S 1A4
Phone: 416-978-8747
CSNET, CDNNET: cscsi@toronto
UUCP: . . . !utcsri!cscsi

Managing Editor / Assistant á la rédaction:
Greg Ioannou, The Editorial Centre

Assistance / Aide:
Marina Haloulos, Jim des Rivières

Traduction / Translation:
Jean-Pierre Corriveau

Cover design, production assistance / Conception de la couverture, aide pour la production:
Cathy Ledden, University of Toronto Press

Please send subscriptions, memberships, and changes of address to:
CSCSI/SCEIO, c/o CIPS
243 College Street, 5th floor
Toronto, CANADA M5T 2Y1

Copyright © 1986, Canadian Society for Computational Studies of Intelligence. All rights reserved; *Canadian Artificial Intelligence* may not be reproduced in any form without the written permission of the editor. Printed in Canada by the University of Toronto Press. The opinions expressed herein are those of their respective authors, and are not necessarily those of their employers, CSCSI, *Canadian Artificial Intelligence*, the editor, CIPS, or the University of Toronto.

Copyright © 1986, Société canadienne pour l'étude de l'intelligence par ordinateur. Tout droit réservé; *Intelligence artificielle au Canada* ne doit être reproduite par quelque moyen que ce soit sans le consentement écrit de l'éditeur. Imprimée au Canada par la University of Toronto Press. Les opinions exprimées dans ce magazine sont celles de leurs auteurs respectifs et non pas nécessairement celles de leurs employeurs, de la SCEIO, de l'*Intelligence artificielle au Canada*, de l'éditeur, de l'Association canadienne informatique, ou de l'Université de Toronto.

Résumé en français, 8

Contents

Advertisers

Guidelines for electronic submissions

Canadian A. I. prefers submissions to be made by network mail when possible, to eliminate the need for retyping. However, please do not send text that has been through a document formatter; extra embedded spaces and hyphenated line breaks just have to edited out again before typesetting. The best thing to send is plain typing or *troff* source (preferably with *−me* macros).

Canadian Society for Computational Studies of Intelligence

Founded 1973

CSCSI is the Canadian society for the promotion of interest and activity in Artificial Intelligence. It conducts workshops and fully refereed national conferences, publishes this magazine, sponsors the journal *Computational Intelligence*, and coordinates activities with related societies, government, and industry.

To join CSCSI, use the membership form in this issue. Non-Canadian members are welcomed.

CSCSI is affiliated with the Canadian Information Processing Society and International Joint Conferences on Artificial Intelligence, Inc.

Société canadienne pour l'étude de l'intelligence par ordinateur

Fondée 1973

SCEIO est la Société canadienne encourageant l'intérêt et la recherche en Intelligence Artificielle. Elle organise des ateliers ainsi que des conférences nationales avec évaluation des articles soumis. Elle publie ce magazine, subventionne le journal *Intelligence Informatique*, et coordonne toute interaction avec, des sociétés parallèles, le gouvernement, et l'industrie.

Pour devenir membre de la SCEIO, veuillez utiliser le formulaire d'inscription de ce numéro. Les non-canadiens sont bienvenus.

La SCEIO est affiliée à l'Association canadienne informatique, et aux International Joint Conferences on Artificial Intelligence, Inc.

President / Président: Dick Peacocke, Bell-Northern Research, Box 3511, Station C, Ottawa, Ont K1Y 4H7. Phone: 613-727-2629. UUCP: . . . ! bnr-di! dick
Vice-President / Vice-Président: Renato De Mori, School of Computer Science, McGill University, Montréal, Qué H3A 2K6. Phone: 514-392-8274. UUCP: . . . ! musocs! renato
Secretary / Secrétaire: Bill Havens, Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1W5. Phone: 604-228-6450. CSNET, CDNNET: havens@ubc
Treasurer / Trésorier: Randy Goebel, Department of Computer Science, University of Waterloo, Waterloo, Ont N2L 3G1. Phone: 519-888-4439. CDNNET: rggoebel@water UUCP: . . . ! watmath! water! rggoebel

Canadian Artificial Intelligence

Intelligence Artificielle au Canada

Founded in 1974 as *CSCSI/SCEIO Newsletter*

[En français, page 9]

Canadian Artificial Intelligence is published quarterly by CSCSI/SCEIO, and is a benefit of membership in the society.

Canadian A. I. solicits contributions in English or French on any matter related to artificial intelligence, including:

- Articles of general interest.
- Descriptions of current research and courses.
- Reports of recent conferences and workshops.
- Announcements of forthcoming activities.
- Calls for papers.
- Book reviews (and books for review).
- Announcements of new AI companies and products.
- Opinions, counterpoints, polemic, controversy.
- Abstracts of recent publications, theses, and technical reports.
- Humour, cartoons, artwork.
- Advertisements (rates upon request).
- Anything else concerned with AI.

Please send submissions, either on paper or by electronic mail, to the editor at the address on the previous page. On-line submissions are preferred, but they should not contain justification spaces or hyphenated line breaks as these just have to edited out before typesetting; 'plain typing' is best.

Canadian A. I. is published in January, April, July, and October. Material for publication is due six weeks before the start of the month of publication.

Letters to the Editor

More Progress in Japan

I thought that it might be useful both to clarify some of the points discussed in "Is the Japanese Fifth Generation Project Faltering" (*Canadian A. I.*, September 1986) and to add some new fuel to the fire.

First, it is important to note that two kinds of hardware projects exist within ICOT's mandate for parallel machine architectures. The first is known as the Parallel Inference Machine (PIM) subsystem, whereas the second is called the Knowledge Base Machine (KBM) subsystem. During the initial phase of the project (FY 1982—1984), the DELTA database machine (referred to in Mr Bassett's article) was developed under contract by ICOT as a part of the KBM subsystem. Now in the second phase, both the PIM and KBM subsystems are being further developed and refined. A number of company laboratories are currently developing prototypes of alternative PIM architectures originally proposed during the first phase of the ICOT project. At the same time, ICOT researchers have stated that although the DELTA database machine was fast, it was not easy to use. A smaller version is now being developed with better hardware and software support for logic programming.

Second, although ICOT's research places a strong emphasis on hardware, software is by no means being overlooked! In fact, a number of different logic programming languages (KL0, KL1, ESP) have been developed. The following example illustrates the use of these languages within the ICOT mandate.

In late 1984, at the Second International Conference on Fifth Generation Computing Systems held in Tokyo, ICOT announced the prototype Personal Sequential Inference (PSI) machine, developed by Mitsubishi. At the time, a software environment for that machine was only available on the DEC 2060 development machine. Since then, the eight participating companies have contributed a forty-person development team whose mandate was to produce a programming environment for the PSI machine. Now, some two years later, that team has written an operating system/programming environment for the PSI that consists of (as of late April 1986) approximately 160,000 lines of code written in Extended Self-contained Prolog (ESP). This system, called SIMPOS, now provides the programming environment for the PSI, which was recently announced as a product by Mitsubishi (at a price of ¥30 million). I would be surprised if many companies in other countries would (or could) co-operate on the level demonstrated by the participants to the ICOT project.

Another point to make is that hardware engineers in Japan *are* interested in Prolog and in architectures for running Prolog quickly and efficiently. At the University of Tokyo, special interest groups exist in Electrical, Mechanical, and Precision Machinery Engineering for studying both Prolog and Lisp. Although "study groups" are a traditional way to master a new technology in Japan, I doubt that many similar such groups exist in engineering schools in North America.

As far as national projects in Japan are concerned, please add the following to your list of projects to watch. Japan's Ministry of International Trade and Industry (MITI, who else?) has announced plans to expand the work now going on in Japan in machine translation. The seven-year, ^{US}39 million project is aimed at developing translation systems for Japanese and the languages of Asian countries, including China and Thailand. The usual participants (Hitachi, Fujitsu, and NEC) will be involved, along with the National Electrotechnical Laboratory.

Finally, in April 1986, 19 Japanese companies joined together to form the Artificial Intelligence Joint Research Society and an Artificial Intelligence Centre was formed under the administration of ICOT and the Japan Information Processing Development Centre (JIPDEC), a service and research agency supported by government and industry. A third group known as the Artificial Intelligence Society was formed in July 1986.

The Artificial Intelligence Joint Research Society is being established initially to train researchers and will launch a joint AI

development project involving expert systems within two years. The Artificial Intelligence Centre expects a membership of about 300 companies and plans to provide a forum for exchanging information among member companies and to take surveys and studies on AI technology and markets. The Artificial Intelligence Society will be for individual members and is designed to coordinate research activities being undertaken by different academic societies and to help commercialize AI systems. The Society was established with 3,000 members.

The formation of a co-operative association involving many Japanese companies in a certain industrial sector is a typical reaction of Japanese industry to an emerging technology's potential market. This same reaction led to the formation of similar associations, such as the Japan Machine Tool Builders Association in 1951 and the Japan Industrial Robot Association in 1972. We all know what has happened in those industries!

We in Canada do not yet have a national initiative in this vital and strategic technological area. In light of recent National Research Council funding cutbacks, prospects look dim. At the very least, we can still talk about what *everyone else* is doing these days.

Mark Paul Turchan
Bell-Northern Research
Ottawa

Editor's notes

Editor becomes team

Graeme Hirst
Senior Editor

I am pleased to welcome to *Canadian A.I.* Greg Ioannou, who has become our first real paid staff member.

For the last 2½ years, I have held the voluntary position of editor, managing editor, advertising salesman, paste-up artist, and administrator of *Canadian A.I.*, all rolled into one person for CSCSI/SCEIO. Unfortunately, I couldn't continue to donate so much time, and asked the Society's executive to find a new editor. Instead, the executive found wealth — profits made by the Society from the Canadian AI Conference in Montréal last May — which is sufficient to buy genuine professional help.

Greg Ioannou

Greg Ioannou is a freelance professional editor, who, with his colleagues, operates The Editorial Centre in Toronto. Greg will be taking over most of the day-to-day editorial, administrative, and production work of the magazine. He will be Managing Editor. I will remain as Senior Editor, responsible for the quality and direction of the magazine.

The magazine will continue to be produced at the University of Toronto, and CIPS, the parent organization of CSCSI/SCEIO, will continue to handle memberships and subscriptions. Thus, there will be no change in the addresses for correspondence.

Look for continuing improvements to *Canadian A.I.* as more time is put into its production. As always we welcome contributions and suggestions from readers. □

Erratum

In the article "Is the Japanese Fifth Generation Project Faltering" (*Canadian A.I.*, September 1986), an incorrect value for the yen was given. The current value of ¥1,000 million is about ^{CDN}8 million, not $111 million.

New Bindings

David Etherington, from University of British Columbia to AT&T Bell Laboratories, AI Principles Research Department.

RAPID PROTOTYPING TOOLS FOR AI

LE_LISP

A new large Common Lisp subset developed by INRIA, Europe's top AI R&D institution. Both English and French versions available. Rich features include object extensions, multiple inheritency, color support, BIOS access, full screen editor, full debugger, and a large function set. The system enjoys easy transportability to a wide range of machines with efficient implementation and fast execution. $695.

GCLISP NEW PRODUCTS

A new line of products have been added to the popular Common Lisp AI development tools: an advanced processor version, a delivery package, a PC networking, and a graphics tool.

- 286 Developer ($1675)
- GCLISP RUNTIME ($140)
- GCLISP NETWORKS (from $555)
- Halo Graphics ($350)

LPA MICRO-PROLOG/APES KIT

Prolog and Expert System prototyping tool best suited for scientific and educational applications. Refined syntax gives flexibility appropriate for sophisticated programming. Several thousand copies sold throughout the world. $1073 (PC version, kit) or $710 (MAC version, micro-PROLOG only).

ARITY KNOWLEDGE SYSTEM DEVELOPMENT PACK

A new generation AI prototyping tool for PCs. A Prolog interpreter/compiler, a rule/frame-based Expert Systems shell with object hierarchy and inheritance, the external file and database access, a screen design kit, the linkage to conventional languages, and an IBM SQL access. $1715.

'Q&A' FROM SYMANTIC

Natural Language interface for popular database systems (including dBase III). Some 25,000 copies sold in the U.S. during the past 9 months. $495.

ORDER AND ENQUIRY: Call (613) 592-0084 or send Purchase Order to Applied AI Systems, Inc. Shipping and Handling charges and provincial sales tax (Ontario residents only) apply.

Applied AI Systems, Inc is an authorized dealer of the above AI software/hardware products.
Le_Lisp is a registered trademark of ACT Informatique.
GCLISP, Halo Graphics, 286 Developer, GCLISP RUNTIME, and CLISP NETWORKS are registered trademarks of Gold Hill Computers, Inc. micro-PROLOG, APES are registered trade marks of Logic Programming Associates Ltd. Q&A is a registered trademark of Symantics Inc.

Applied AI Systems, Inc.
P.O. Box 13550
KANATA, Ontario
Canada K2K 1X6

Telephone: (613) 592-0084

Intelligence Artificielle au Canada

Canadian Artificial Intelligence

Janvier 1987 No. 10 (Nouvelle série)

Résumé

Annonceurs

Symbolics, 2
La série des machines Lisp Symbolics 3600.

Applied AI Systems, 7
Outils de construction rapide de prototypes pour l'AI: Le Lisp, Gold Common Lisp, Micro-Prolog et APES, l'ensemble de dévelopement de systèmes de connaissance Arity, et l'interface en langage naturel Q&A de Symantec.

Lisp Canada, Inc., 11
Représentants canadiens pour la série LMI Lambda de machines Lisp.

Canadian AI Products Corp., 13
Colloques au sujet de l'IA.

AI West '87 and East '87, 17
La conférence sur l'AI et la technologie informatique avancée sera présentée à Long Beach, Californie, 22–24 avril 1987, et Atlantic City, New Jersey, 28–30 octobre 1987.

CCRIT, 19
En français.

Xerox Canada, Inc., 21
La série Xerox 1100 d'appareils pour la travail en intelligence artificielle.

Heurix Computer Research Inc., 31
Consultants pour les applications de l'IA.

University de Waterloo, 35
L'université de Waterloo a plusieurs positions disponibles à tous les échelons. Un doctoral en informatique ainsi que des résultats significatifs ou forte aptitude à la recherche sont requis.

Logicware, 39
TWAICE, un environnement pour l'emploi de systemmes-experts pour les ordinateurs IBM, les VAX, et les postes de travail utilisant le M68000.

Applied AI Systems, 40
Le répertoire pour le développement de systèmes en A, et Golden Common Lisp pour les IBM PCs.

L'échéance pour le numéro d'avril est le 15 février.

Intelligence Artificielle au Canada

Canadian Artificial Intelligence

Fondée en 1974 en tant que *CSCSI/SCEIO Newsletter*

L'*Intelligence artificielle au Canada* est publiée trimestriellement par la CSCSI/SCEIO, et est offerte gratuitement aux membres de la société.

L'*Intelligence artificielle au Canada* encourage les contributions, en français ou en anglais, portant sur l'intelligence artificielle. Ceci comprend:
- Des articles d'intérêt général.
- Des descriptions de recherche courante et de cours.
- Des rapports de conférences récentes et d'ateliers.
- L'annonce d'activités à venir.
- Des requêtes d'articles.
- Des critiques de livres (ainsi que des livres à critiquer).
- L'annonce de nouvelles compagnies en IA et de leurs produits.
- Des opinions, des répliques, tout ce qui est polémique.
- Des résumés de publications récentes, de thèses et de rapports techniques.
- Des trucs humoristiques ou artistiques, des bandes dessinées.
- Des annonces (s'enquérir des frais).
- Tout autre matériel touchant à l'IA.

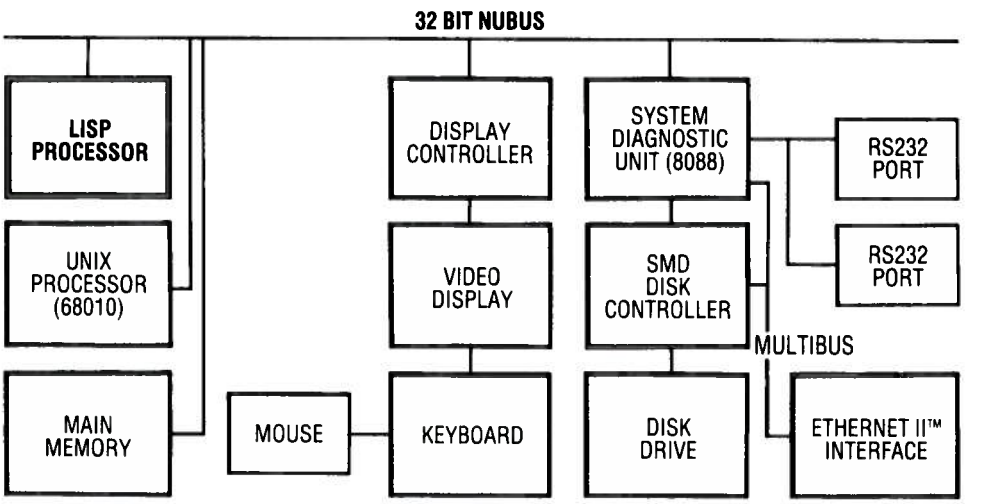
Veuillez expédier vos contributions, soit sur papier ou par courrier électronique, à l'éditeur dont l'adresse apparait à la page précédente. Nous préférons le courrier électronique mais ce qui est ainsi envoyé ne devrait pas contenir d'espaces de justification ni de mots à trait d'union puisque ceux-ci doivent être supprimés avant la mise en page; un texte 'tel quel' est ce qu'il y a de mieux.

L'*Intelligence artificielle au Canada* apparait en janvier, en avril, en juillet, et en octobre. Toute communication à publier doit nous parvenir au moins six semaines avant le début du mois de parution.

LISP Canada Inc.

Distributes and Services the LMI Lambda Machine

LMI Lambda Machine



USERS BENEFIT FROM:

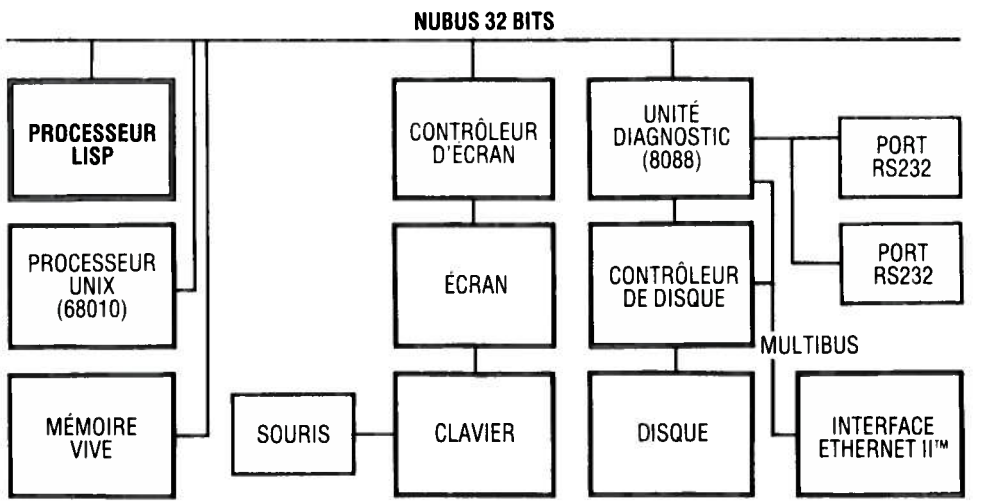
- MULTIBUS™ Compatibility
- Multi-processor Capability
  - LISP Processor
  - 68010 UNIX Processor
  - 8088 System Diagnostic Unit
- Virtual Control Store and Microcompiler
- Powerful Software Resources

LISP Canada Inc.
5252 de Maisonneuve W. Blvd.
Montreal, Quebec, Canada
H4A 3S5
Tel: 514 **487 7063**
Telex: 055-66382

LISP Canada inc.

Distribue et entretient la machine Lambda de LMI

Machine Lambda de LMI



LES AVANTAGES DU LAMBDA

- **Compatibilité MULTIBUS™**
- **Accès à de multiples processeurs**
  - **Processeur LISP**
  - **Processeur UNIX (68010)**
  - **Unité de diagnostic (8088)**
- **Microcompilateur**
- **Logiciels puissants**

LISP Canada Inc.
5252 de Maisonneuve W. Blvd.
Montreal, Quebec, Canada
H4A 3S5

Tel: 514 **487 7063**
Telex: 055-66382

CAIP CORPORATION

Canadian Artificial Intelligence Products

EXPERT SYSTEMS THE PRACTICAL ASPECTS

A SERIES OF ONE-DAY SEMINARS

Dates	Locations	Dates	Locations
October 28	Toronto	November 25	Vancouver
November 4	Ottawa	November 26	Edmonton
November 12	Montreal	December 2	London
		December 9	Toronto

CAIP Corporation is the leading Canadian-based organization concentrating exclusively on practical applications of Artificial Intelligence technology. Since its incorporation in 1984, CAIP has developed extensive experience in the analysis and development of knowledge-based expert systems, the fastest growing segment of the broad domain of artificial intelligence.

CAIP professionals comprise experienced knowledge engineers, business consultants, systems analysts and many of Canada's leading scientists from universities across the country. The ability to configure project teams of technical resources coupled with world renowned academics provides CAIP with a unique service offering.

CAIP provides government and commercial clients with a full range of services.

Application surveys
Feasibility analysis
Prototype development
Knowledge based system development
Knowledge and rule-based system acquisition and maintenance
Contract research and development
Product management
Educational services

CAIP wishes to thank the over 200 managers and executives from across Canada who made the recent series of seminars so successful.

The company is impressed by the growing number of initiatives being undertaken by industry and government in the knowledge based technologies.

As part of our range of client services, CAIP offers customized in-house seminars covering expert systems and their applications.

CAIP CORPORATION

Canadian Artificial Intelligence Products

106 Colonnade Road, Suite 220, Ottawa, Ontario K2E 7P4 (613) 727-0082

5915 Airport Road, Suite 200, Toronto, Ontario L4V 1T1 (416) 671-4650

Artificial Intelligence Research at Queen's University

Janice Glasgow
Dept Computing and Information Science
Queen's University
Kingston, Ontario K7L 3N6
(613) 545-6058

The Nial Project at Queen's University, led by Professors Janice Glasgow and Mike Jenkins, is involved with the design, implementation and application of the Nested Interactive Array Language, Nial. Nial is a programming language designed by Mike Jenkins of Queen's University and Trenchard More of the IBM Cambridge Scientific Centre. Q'Nial is a portable implementation of Nial developed at Queen's University. Q'Nial is available on many architectures, including large timesharing machines, Unix systems and 16-bit personal computers.

The project involves research in the following areas:

- The theory of array data structures and the Nial language.
- Software and hardware implementation techniques for Nial.
- Language extensions to integrate logic programming with databases.
- The design of a tool kit for knowledge based systems.
- Prototyping tools.
- Educational microworlds in Nial.

Nial and Artificial Intelligence

AI is one of the principal application areas of Nial, because the powerful operations for dynamic manipulation of both symbolic and numeric data permit straightforward implementation of AI techniques. Some of the features of Nial that support AI work are:

- The inclusion of an atomic class of symbols isomorphic to strings.
- A complete numeric type hierarchy, including complex numbers.
- Distributed arithmetic, logical, and comparative operations that eliminate many loops.
- Dynamic nested array data structures that are more general than the nested lists of Lisp.
- Convenient representation of association lists.
- Dynamic generation of variable names.
- Dynamic assignment and evaluation of variables.
- Three representations of program structures.
- The ability to "eval" an expression or "apply" an operation dynamically, either by name or by its internal representation.
- The inclusion of resolution logic primitives.

The Nial Artificial Intelligence Toolkit is one product of the on-going research of the Nial Project. The philosophy is to provide building blocks from which tailored knowledge-based systems are constructed. The goal is to provide direct implementations of basic techniques that can be understood well enough to be modified by the programmer constructing a particular application. The language features described above are used in the programs that make up the Toolkit.

Many of the tools being designed are in a preliminary form as the procedure being followed is one of an initial implementation, use of the tool in an application or two and then refinement of it into a more polished form.

Logic Programming and Provers

The goal of the research in logic programming is to integrate the functional and procedural styles of Nial with the declarative capabilities of logic. Initial experience suggests that the effective marriage of the two styles is possible and that the approach is useful in many knowledge representation problems.

In Nial, logic clauses are represented in arrays. Operations are provided that carry out the primitive steps of resolution logic from which flexible reasoning capabilities can be constructed. The

first version of the logic primitives allowed experimentation with a number of different proof strategies and user interfaces to the logic system. A reimplementation of the logic programming component of Nial is underway, based on this experience. It will use more effective algorithms for the resolution logic operations and will have a flexible built-in prover that, through the use of function parameters, can be tailored for a particular problem. The default behaviour of the prover is similar to the Prolog proof strategy.

The user interface to the logic programming component is through the use of operations that parse logic in a conventional Prolog-like notation into the internal representation as nested arrays. The logic programming component is triggered by calling the prover operation on a knowledge base of clauses with a goal. From within the logic programming component, any Nial operation can be triggered by the "do" or "eval" predicates. From this base of a resolution logic prover, other provers can be constructed that apply heuristics or modify the search strategy.

Roster Database Model

The relational model of data is an effective way to store data that has a static conceptual organization. The "roster" model is a generalization of the relational model that allows arbitrary objects to be stored and retrieved from a tabular format.

Three versions of the roster model are provided for different purposes. The "model" version is a functional implementation using nested arrays to store rosters. It is intended for conceptual design and implementation of a prototype application.

The "stored" version is a procedural version that stores rosters in nested arrays that can be effectively accessed and modified. It is suitable for applications in which the data can be conveniently stored in a large virtual workspace.

The "external" version is a procedural version that stores its data in a conventional database outside the Nial workspace. The interface is achieved by translating roster operations into database queries in the language of the database system. An interface to Ingres is available and work is underway on a general SQL interface. The external version is intended for use with an application that accesses data stored externally to Nial and needs Nial's expressive power of data manipulation.

The toolkit currently provides the "model" rosters, and a user interface package, RIPO (Rosters in, Prototype out). These are useful in prototyping applications that use a roster or relational model to store information. A roster editor is planned to assist in the design of rosters, and implementations of the stored and external rosters the toolkit.

Inference Engine

The inference engine is a knowledge representation tool that combines the roster model of data with the logic programming capability of the language. Its purpose is to explore the advantages and disadvantages of separating factual information, such as that stored in a conventional database in a corporate setting, from knowledge about what the facts mean and rules on how to use the facts for a particular task.

The knowledge base is stored in rosters of two types: rosters that have factual information, and rosters that contain logic clauses that represent knowledge about the facts. The former rosters are unnecessary in that they can be replaced by clauses in the latter roster, one to represent each fact. However, for a large database it is more efficient to allow the retrieval of such information to be handled by database primitive operations than by the logic prover.

The related prover can obtain knowledge from either the clauses, the factual rosters, or user input to complete the proof of the goal clause. Reasoning is done with a mixture of resolution steps and database lookups. One advantage of this approach is that once the prover has been integrated with an external representation of a roster system, it should be possible to add a reasoning capability to applications that access an existing database used for other corporate purposes.

Fuzzy Rosters with Fuzzy Reasoning

In building knowledge-based systems it is often necessary to reason with knowledge that is uncertain. We are investigating the use of Zadeh's possibility theory to allow fuzzy reasoning on information stored in fuzzy rosters.

An extended version of the inference engine is being designed to allow fuzzy data to be stored in rosters with a grade of membership function. Fuzzy versions of the roster operations are defined, as well as fuzzy comparators on the data. A consistent mathematical treatment of this seems feasible.

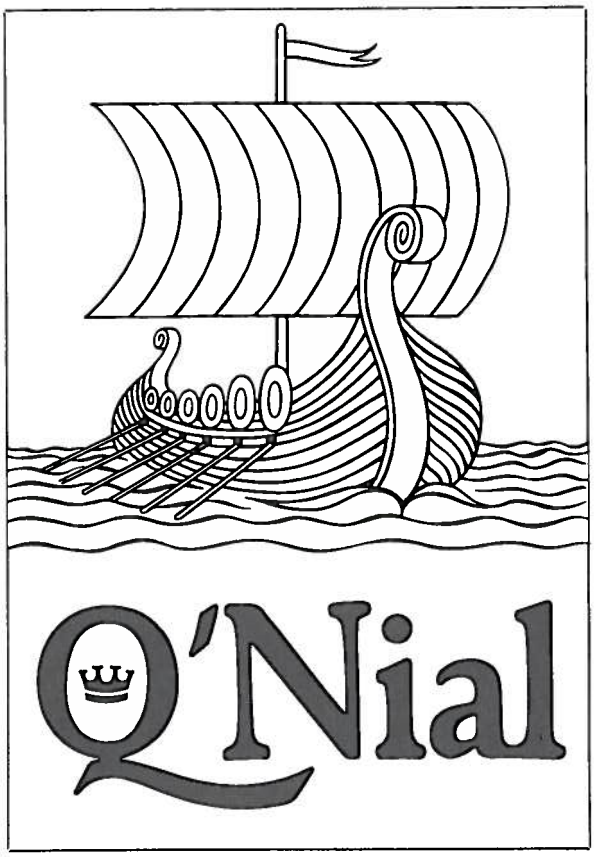
A version of the prover is being designed that uses fuzzy reasoning on logic rules stated about uncertain data stored in fuzzy rosters. It appears that the combination of this prover with the fuzzy rosters provides a systematic treatment of uncertainty in knowledge based systems.

Nial Frame Language

Many of the practical knowledge based systems built in Lisp are based on a "frame" knowledge representation. Although different frame systems exist, they are based on the use of nested association lists to build a semantic network of nodes, and on built-in inferencing that uses the hierarchical structure of the network.

The Nial Frame Language (NFL) uses nested association lists organized in a semantic network to achieve a frame knowledge representation strategy. It supports automatic inferencing along certain hierarchical paths in the network. The usual pattern of frames, slots, and facet association lists is used, with both generic and individual classes of frames.

The frame language provides a convenient starting point for a knowledge-based system in which there is a natural mapping of the structure of knowledge in the domain to the hierarchical organization provided by frames. For example, a frame system can be used to organize the rules in a rule-based expert system into a hierarchy of groups of related rules. This subdivides the knowledge base into manageable parts that can be maintained more easily.



Rule Interpreters

A simple approach to rule based systems has been built using frames to represent *if-then* rules. Rules are frames with slots for name, English version, *if*-part, *then*-part, forward rules and backward rules. The clauses of the rule subparts are Nial expressions that are executed to achieve the required test or action. Using the same rule base, models of both forward and backward chaining interpreters are presented.

While these rule interpreters were constructed for pedagogical use, they have been used as the basis for two small expert systems.

Demon-driven Natural Language Parser

One approach to natural language parsing is to use a domain-dependent conceptual lexicon to drive the parsing from semantic considerations. This approach appears to be well suited to query analyzers over specific domains because the vocabulary and the nature of the discourse are quite specific and structured.

The technique is to scan the text and to have each word that is recognized trigger processes (demons) that either resolve ambiguity or fill in gaps by finding words to complete the conceptualization frame associated with the parent word. Multiple demons can be triggered to fill a particular gap; when one is satisfied it kills off all associated demons.

Demons are represented by dynamically generated variables that contain the name of the concept being sought and the name of the procedure that is to be executed to find it. A list of active demons is maintained and each is tested after each word is processed until no further triggering occurs. Such a system is being implemented in Nial.

Extended Nial

The Q'Nial system is being extended to include a window package, a component file system, and a built-in editing capability.

The window package allows input/output to be directed to a particular screen area and allows viewing of objects that are larger than the physical window. It provides a high-level programmable interface that simplifies the construction of user interfaces. It is a limited capability intended to be used under program control, rather than as a multi-window access to the operating system. The facility allows similar user interfaces to be provided across Unix, PC-DOS, and IBM mainframe environments.

The file system access capability of Q'Nial is extended to allow random access to strings of text stored in a component file system. With functions that map arrays to and from character strings, the file system can be used to import and export programs and data to and from the workspace.

The built-in editor is part of the window package. The window setting determines a screen area in which text can be examined or edited. The editor can handle text that is too large for the window, using scrolling to access the parts out of sight. The editor provides primitive screen editing capabilities which can be embedded in a more elaborate program tailored to a particular need.

Educational Environment

Nial is being used as an environment to develop educational microworlds. In general, these microworlds are Nial workspaces in which operations have been provided that allow the student to exercise concepts in the domain of study. It is possible to access all features of the language while in this learning environment and thus augment or modify the microworld as desired.

Existing microworlds in Nial include ones for turtle graphics, algebraic manipulations, symbolic logic reasoning and a Nial tutor. The mathematical nature of the language makes it particularly suitable for learning algebra and logic. Future work in this area includes using AI tools to represent and manipulate knowledge about the domain and user of a microworld.

References to Related Publications

E. Blevis. "A Basis for Effective Logic Programming in Nial." Master's Thesis, Queen's University, 1986.

R. Chau, J.I. Glasgow and M.A. Jenkins. "A Framework for Knowledge Based Systems in Nial." Submitted to the 1987 Phoenix Conference on Computers and Communication.

J.I. Glasgow. "Artificial Intelligence Programming Languages." *Unix World Magazine*, August 1986.

J.I. Glasgow, M.A. Jenkins, E. Blevis, R. Chau, E.A. Hache and J. Whybra. "Experience with AI Programming in Nial." Proceedings of STeP-86, the Finnish Symposium on Artificial Intelligence.

J.I. Glasgow, R. Chau and P.L. Douglas. "Building Expert Systems for Chemical Process Engineering Using Nial." Proceedings of the 6th International Workshop on Expert Systems and their Applications, Avignon, France, April 1986.

J.I. Glasgow, M.A. Jenkins and C.D. McCrosky. "User Defined Parallel Control Strategies." Proceedings of the 1985 Symposium on Logic Programming, 22-28.

J.I. Glasgow and R. Browse. "Programming Languages for Artificial Intelligence." *Journal of Computers and Mathematics with Applications*, **11** (5), 1985, 431-448.

E.A. Hache. "Nial Frame Representation Language." Master's Thesis, Queen's University, 1986.

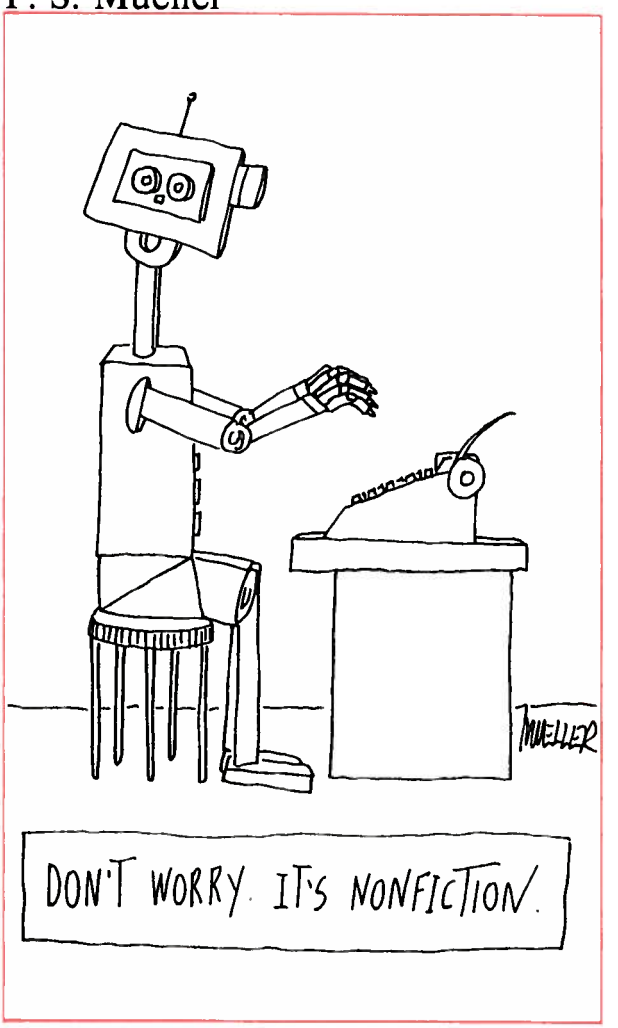
M.A. Jenkins, J.I. Glasgow and C.D. McCrosky. "Programming Styles in Nial." *IEEE Software Engineering*, January 1986. Also in the Proceedings of the 19th Hawaii International Conference on Systems Sciences, 1986.

M.A. Jenkins and W.H. Jenkins. "The Nial Reference Manual." Kingston: Nial Systems Ltd., 1985.

C.D. McCrosky, J.I. Glasgow and M.A. Jenkins. "A Candidate Language for Fifth Generation Systems." Proceedings of the ACM Annual Conference, 1984.

T. More. "Notes on the Diagrams, Logic and Operations of Array Theory." in O. Bjorke and O. Franksen (eds.), *Structures and Operations in Engineering and Management Systems*. Trondheim, Norway: Tapir Publishers, 1981, 497-666.

P. S. Mueller



Don't Miss the Leading Applications–Oriented Forum for Users of AI Systems and Equipment

AI '87 emphasizes a widening array of applications for increasing productivity and reducing costs in business offices, industrial plants, laboratories and many professions.

Exhibition Highlights Newest Technology

Nearly 150 display booths of AI software and hardware provide a unique opportunity to evaluate

AI's potentials and to select from among the most current and sophisticated systems, equipment and technology available today.

AI 1987

3rd Annual Presentation

Artificial Intelligence and Advanced Computer Technology Conference/Exhibition

April 22-24, 1987
Long Beach Convention Center
Long Beach, California

Plan Now to Attend!

Mail Coupon Today for Details

TO: Tower Conference Management Co.
331 W. Wesley St., Wheaton, IL 60187

☐ I am interested in ATTENDING AI '87. Please send advance registration form and conference program details.

☐ My company is interested in EXHIBITING. Please send further information.

Name _____
Title _____
Company _____
Address _____
City _____ State/Country _____
Postal Code _____
Phone _____
Telex _____

AI '87 Conference Program to Feature Commercial, Industrial & Government Applications

More than 100 papers will be presented in 27 technical sessions, highlighting such topics as:

- Expert Systems
- AI Workstations
- Manufacturing
- AI Languages
- Aerospace/Defense
- Machine Translation
- Medical/Health Care

Four Tutorials and a Workshop round out this informative, applications-oriented program.

Organized By:

Tower Conference Management Company
331 W. Wesley St., Wheaton, IL 60187
(312) 668-8100 Telex: 350427

Co-Sponsored By:

DIGITAL DESIGN
THE ELECTRONIC SYSTEM DESIGN MAGAZINE

DM Data Inc.

SCS Society for Computer Simulation

USAI
Applied Artificial Intelligence Reporter

The Catalogue of Artificial Intelligence Tools

Alan Bundy

The *Catalogue of Artificial Intelligence Tools* is a kind of mail-order catalogue of AI techniques and portable software. Its purpose is to promote interaction between members of the AI community. It does this by announcing the existence of AI tools, and acting as a pointer into the literature. Thus the AI community will have access to a common, extensional definition of the field, which will: promote a common terminology, discourage the reinvention of wheels, and act as a clearinghouse for ideas and software.

The catalogue is a reference work providing a quick guide to the AI tools available for different jobs. It is not intended to be a textbook like the *Artificial Intelligence Handbook*. It intentionally only provides a brief description of each tool, with no extended discussion of the historical origin of the tool or how it has been used in particular AI programs. The focus is on techniques abstracted from their historical origins.

The original version of the catalogue was hastily built in 1983 as part of the U.K. SERC-DoI IKBS Architecture Study. It has now been adopted by the U.K. Alvey Programme and is both kept as an on-line document undergoing constant revision and refinement and published as a paperback by Springer-Verlag. Springer-Verlag have agreed to reprint the *Catalogue* at frequent intervals in order to keep it up to date. The on-line and paperback versions of the catalogue meet different needs and differ in the entries they contain. In particular, the on-line version was designed to promote U.K. interaction and contains all the entries which we received that meet the criteria defined below. Details of how to access the on-line version are available from John Smith of the Rutherford-Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, U.K.. The paperback version was designed to serve as a reference book for the international community, and does not contain entries that are only of interest in a U.K. context.

By 'AI techniques' we mean algorithms, data (knowledge) formalisms, architectures, and methodological techniques, which can be described in a precise, clean way. The catalogue entries are intended to be non-technical and brief, but with a literature reference. The reference might not be the 'classic' one. It will often be to a textbook or survey article. The border between AI and non-AI techniques is fuzzy. Since the catalogue is to promote interaction, some techniques are included because they are vital parts of many AI programs, even though they did not originate in AI.

By 'portable AI software' we mean programming languages, shells, packages, toolkits, etc, which are available for use by AI researchers outside the group of the implementor, including both commercial and non-commercial products.

We have not included in the catalogue separate entries for each slight variation of a technique, programming language, etc. Neither have we always included details of how to obtain the software, nor descriptions of AI programs tied to a particular application, nor of descriptions of work in progress. The catalogue is not intended to be a dictionary of AI terminology nor to include definitions of AI problems.

Entries are short (abstract length) descriptions of a technique or piece of software. They include a title, list of aliases, contributor's name, paragraph of description, information on availability and references. If you would like to submit an entry for the catalogue, please send it in the format shown below to:

Alan Bundy
Department of Artificial Intelligence
University of Edinburgh,
80 South Bridge,
Edinburgh EH1 1HN SCOTLAND
Phone: +44-31-225-7774 ext 242
JANET: Bundy@UK.Ac.Edinburgh
ARPANET: Bundy@Rutgers.Arpa

Format for entries

Title:
Alias:
Abstract: [Paragraph length description of tool or technique]
Contributor: [Your name]
References: [Aim for the most helpful rather than the 'classic' one]
Availability: [*e.g.* commercially available with documentation and support, available as a research vehicle only with limited documentation]
Environment: [Necessary supporting software/hardware]
From: [Contact address for distribution, including telephone number and electronic mail address if appropriate]

Canadian A. I. invites letters to the editor on any AI topic.

CANADIAN WORKPLACE AUTOMATION RESEARCH CENTER

Research Scientist in Natural Language Processing

The Canadian Workplace Automation Research Center (CWARC), which is part of Communications Canada located in the Montreal area, invites applications for a position of research scientist in natural language processing.

The incumbent will be involved in research and development in the area of machine-aided translation. He or she will participate in the design of new translation models and in the implementation of these models on computer systems. The incumbent will also be expected to prepare scientific articles on this work for publication.

Candidates should have a doctorate degree or all but dissertation, with specialization in natural language processing or related domain. They should also have experience in conducting research and publishing research reports in the areas of computational linguistics and/or artificial intelligence.

Knowledge of English and French is essential for this position. Candidates who do not satisfy the linguistic requirements will have to undertake language training.

Send complete curriculum vitae to the following address:

Quebec Regional Office
Public Service Commission
Guy Favreau Complex
200 Dorchester Blvd. West
West Tower, 8th Floor
Montreal, Quebec H2Z 1X4

CENTRE CANADIEN DE RECHERCHE SUR L'INFORMATISATION DU TRAVAIL

Chercheur en traitement automatisé des langues naturelles

Le Centre canadien de recherche sur l'informatisation du travail (CCRIT), un organisme de Communications Canada situé dans la région de Montréal, sollicite des candidatures pour un poste de chercheur en traitement automatisé des langues naturelles.

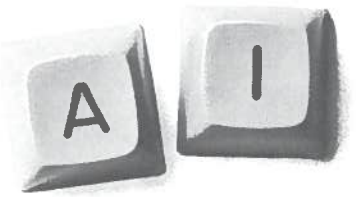
Le titulaire effectuera des activités de recherche et de développement dans le domaine de la traduction assistée par ordinateur. Il participera à la conception de nouveaux modèles de traduction ainsi qu'à l'implantation de ces modèles sur ordinateur. Il rédigera des articles scientifiques sur ces travaux pour fins de publication.

Les candidats devraient avoir complété leur scolarité au niveau du doctorat, avec spécialisation en traitement automatisé des langues naturelles ou un domaine connexe et posséder de l'expérience dans l'accomplissement et la publication de travaux de recherche reliés au domaine de la linguistique informatique et/ou de l'intelligence artificielle.

La connaissance du francais et de l'anglais est essentielle et les candidats ne possédant pas les qualifications linguistiques requises devront entreprendre une formation linguistique.

Faire parvenir un curriculum vitae détaillé à l'adresse suivante:

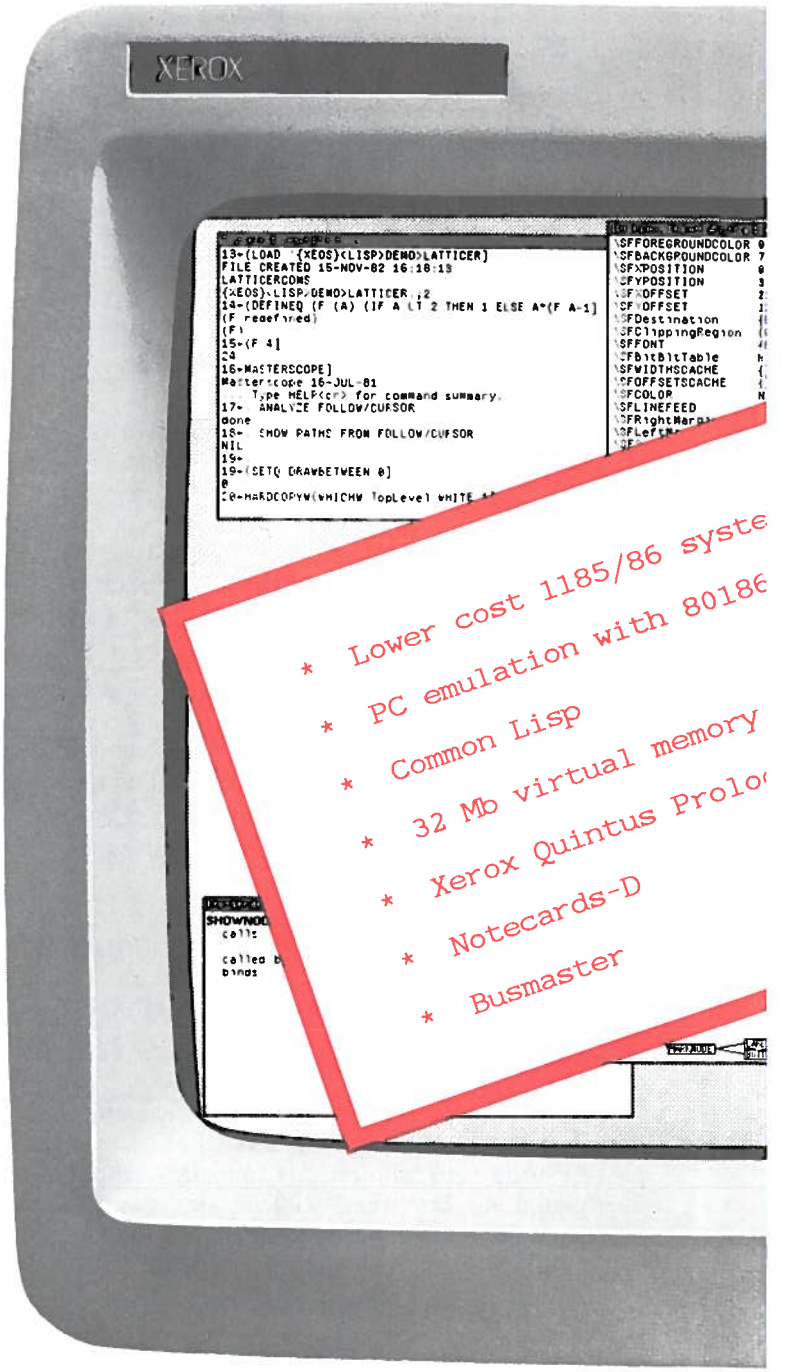
Bureau régional du Québec
Commission de la Fonction Publique
Complexe Guy Favreau
200 ouest, boul. Dorchester
Tour Ouest, 8e étage
Montréal (Québec) H2Z 1X4



The keys to Artificial Intelligence are at Xerox.

The Xerox 1108 Series of Artificial Intelligence Workstations provide an affordable, high-performance line of personal computers in an integrated, interactive environment that greatly enhances programmer productivity.

- High resolution graphics display
- Interactive user interface
- Ability to mix interpreted and compiled code
- Multiprocessing capacity
- Support of an extensible, interpreted language
- Display-oriented programming tools
- Local area networks and data communications through XEROX ETHERNET
- 8 Mbytes virtual memory



Power Tools for Programmers

1. Display Editor and Inspector

The display-based structure editor allows the interactive editing of programs and other list data. Structure-based editing exploits the form of an object, emphasizes the meaning of its parts, and thus reduces errors. The data inspector extends the philosophy to both system and user data types, allowing easy inspection and modification of any object in the system.

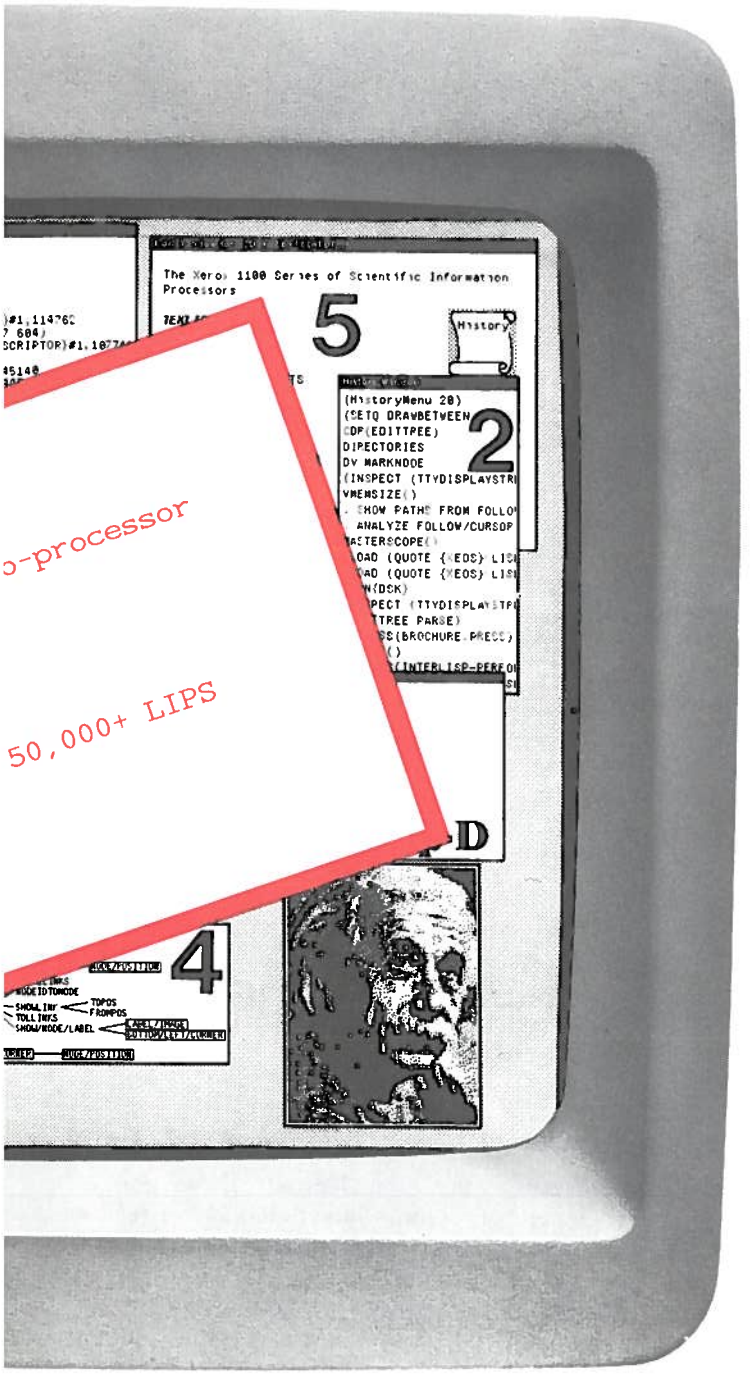
2. Programmer's Assistant

The Programmer's Assistant provides an intelligent assistant and bookkeeper that frees the programmer from much mundane detail. The Programmer's Assistant includes an error analysis capability and also monitors and records all user inputs. For example, a history is kept of the commands typed, their side-effects, and the results. Thus, one can request that a previous command or sequence of commands be repeated, modified and then repeated, or even undone (which undoes all the changes it may have caused). Also provided is a spelling corrector that automatically corrects spelling mistakes using information from the local context. To simplify file management for the programmer, Interlisp-D automatically keeps track of where in the file system each object is stored and which ones have been modified. In response to a simple request, the system can therefore save the user's state, updating all changed files automatically. The Programmer's Assistant provides a programming environment which cooperates in the development of programs allowing the user to concentrate on higher level design issues.

3. Debugging Tools

Debugging tools allow the user to break and trace

XEROX



arbitrary functions, and examine the state of the machine at any desired level of detail. Not only can the state of a suspended computation be displayed and perused graphically, but it can be manually unwound to a specified point, the offending program edited, and execution resumed, all without loss of state. Also included is the capability of specifying complex, user-defined intervention conditions, such as allowing breaks only when a given function is called from another given function. These debugging tools allow bugs to be tracked down quickly and easily.

4. Program Analysis

The Masterscope facility can analyze a user's program and use that information to answer questions, display the program's structure and assist in the process of making modifications automatically. Because Masterscope is interfaced with the file package and editor, it re-analyzes a program whenever it is modified. Information about program calling structure, variable and data structure usage, and side effects can be graphically displayed and used to provide a map or browser for the system. The same information can be used to make systematic changes automatically. Further, Interlisp-D's measurement tools can be used to analyze the behavior of a system after it has been developed to pinpoint those areas that may need improvement.

5. A Professional Workstation

A high bandwidth user interface is provided by combining the mouse and the high resolution display. The mouse permits the user to specify and manipulate positions or regions on the screen. The interactive display facilities include complete raster graphic functions as well as a display management system supporting multiple overlapping windows, menu driven selection of operations, and a wide range of built-in graphical abstractions. Functions are also provided to display text in multiple fonts, manipulate raster images, and draw spline curves. The large format, high resolution display and the sophisticated multiple window system allow concurrent sessions, close-up views, and simultaneous displays of multiple representations of complex data. It is easy to create windows with text, graphics, or both and to make them scroll, update and interact in useful ways with the end user.

6. Knowledge Programming System (Optional)

LOOPS extends the programming environment to provide a powerful tool for research and expert system development. LOOPS combines four programming styles:

- Procedure-Oriented
- Data-Oriented
- Object-Oriented
- Rule-Oriented

Xerox Canada Inc.

Advanced Systems Group
703 Don Mills Road, North York, Ontario M3C 1S2 416-429-6750

XEROX is a registered trademark of XEROX CORPORATION used by XEROX CANADA INC. as a registered user.

Opinion

Artificial Intelligence, Partial Information Domains, and Frame Technology

Paul Bassett
Netron Inc
99 St Regis Crescent North
Downsview, Ont M3J 1Y9

To AI or not to AI; that is the question. Artificial Intelligence is the new vogue. From knowledge-based expert systems and English language interfaces, to robots, and even intelligent terminals, the 'in' thing today is to claim your product is smart.

When is it legitimate to claim to have AI? The confusion is likely to grow much worse before the novelty and market gloss wears off. The Japanese claim to be doing it with their so-called fifth generation projects. Other countries and companies are budgeting large sums in a desperate effort to avoid being left in the dust. Unfortunately, staying out of the dust may mean staying in the fog!

The fog only gets thicker when you consult most workers in the field. AI is still in gestation: the bastard child of computer science and cognitive psychology and linguistics and neurophysiology and . . . . Will the real father please stand up?

Depending on whom you ask, the question "What is AI?" elicits many answers, from "AI is anything you want to call AI" through "Only systems that learn possess AI", to "Who is the customer?" None of these answers are very helpful. If anything, AI is a Garden of Eden full of naively optimistic people beating around the bush, building all kinds of exotic devices in an as-yet-unsuccessful attempt to climb the tree of automatic learning and eat its forbidden fruit.

Now let me briefly survey various efforts generally considered to be AI, with apologies to any offended AI worker whose area I might have missed. Historically, one of the oldest yet still active areas is the analysis of games such as checkers and chess. Arthur Samuel in the late 1950s produced a checker-playing program that could improve through actual play. It quickly became much better than Samuel, and a rewritten version became a world-class player (only a handful of people could beat it).

Chess turned out to be a much tougher nut. A few years ago some naive researchers lost a $5000 bet that a program would soon beat a grandmaster in match play. Today, though a few programs do play master level chess, they still do not learn. They rely primarily on a brute force search of possible moves. Unlike a human player, a program's knowledge of chess is too shallow to save it from wasting over 99.99 percent of its time analyzing useless moves.

Chess is an example of how tricky it is to know what is AI and what is not. Should a program that can trounce you at a thinking game be considered clever just because its speed allows it to exhaustively crunch through larger haystacks to find sharper needles? On the other hand much fruitful AI research can come from game analysis because complex problems can be sharply and easily defined, allowing most of the research effort to be focused on strategies for skill development on the part of the program.

Another early effort was Newell, Shaw, and Simon's General Problem Solver (GPS). They came up with a general way to represent problems, and a set of operators that could either solve the problem directly, or break the problem into sub-problems and repeat the process. However, the GPS operators often failed to decompose a problem properly because it either did not have, or could not handle, vital contextual knowledge, procedural constraints, and partial solution feedback about the problem.

GPS illustrates the classical AI problem, which is still unsolved: how to endow a machine with general intelligence. GPS was strong at representing a wide variety of problems but was very weak when it came to solving them. However there are two sides to this coin. Expert systems have been developed that demonstrate specific intelligence and are quite powerful. Yet no one knows how to combine the power of the expert with the universality of the generalist.

Examples of successful expert systems include XCON for configuring computer systems, MYCIN, which recommends a course of treatment for bacterial infections, and DENDRAL, a system that works out the structure of chemical compounds. At the heart of all these systems is a very simple concept. Each expert uses a set of rules of the form:

> If *A* and *B* and *C* and . . . are all true,
> then do *X* and *Y* and *Z* and . . . .

where *A, B,* etc, are conditions about the problem that the system can test, and *X, Y,* etc, are actions the system can perform that help to solve the problem, either by direct output or by indirect changes to the data base about the problem. By repeatedly executing rules whose 'if' parts are satisfied, the system gradually transforms the original data about the problem into a solution. The trick is to collect and codify sufficient knowledge to solve most of the problems that a human expert can solve. The new buzz-word 'knowledge engineering' denotes this tricky exercise.

It is important to understand that, as with chess and GPS, these systems do not learn: they can't create, modify, or break their own rules either to solve a specific problem or to broaden and deepen their existing expertise.

Another major AI initiative attempts to get machines to understand natural languages such as English. Here the problem is to correctly interpret strings of symbols (or sounds) in terms of some model of the world that the machine can use internally to construct an appropriate response. Natural language is deliberately full of ambiguities. This makes it easy to communicate efficiently, using pronouns, for example, to mean complex things that may not even have names. "Tie your shoe laces" is a simple way to state a common problem whose solution defies a precise rendering in any natural language. Constructing an appropriate response in this case clearly calls for an internal model of knots that is far removed from English text.

Of course ambiguity can also be very troublesome for humans. Because I do not bake, recipes containing directions such as "Add a pinch of salt to taste, then whip until moderately stiff" frustrate me, even though cooks find them just fine. Unless both parties to a communication share very similar models about it, a misunderstanding is virtually certain. The current state of the art permits natural language interfaces to some computer databases, since they constitute reasonably unambiguous models.

My survey ends with the very important but somewhat arcane research into vision and robotics. If artificial intelligence were to be simply defined as endowing machines with natural intelligence, this area would hardly qualify. Even flies can see and move better than current machines. The problem for vision systems is analogous to natural language: how to correctly interpret two-dimensional arrays of light intensities in terms of some three-dimensional model of the world that the system can use in real time to construct appropriate responses.

As with natural language, a visual scene is full of ambiguities. The problem is made even more difficult because the constraints imposed by natural language grammars are much better understood than the implicit regularities in the world that natural vision systems unconsciously and seemingly effortlessly exploit to resolve ambiguity. Almost all robots today are blind. Needless to say real-time hand-eye coordination is still only a research goal. Robots are successful at simple, repetitive tasks. Robots are trainable. But do they learn?

So what direction should AI be taking? In my previous article ("Japan finds itself on the wrong path", *Canadian A. I.*, September 1986), I argued that formal logical approaches, such as those upon which Japan has based its fifth generation project, are unsuitable. Such approaches are limited to Complete Information Domains (CIDs), while AI must deal with Partial Information Domains (PIDs) — domains that are inconsistent, incomplete, or both.

While the ability to reason logically is of obvious importance, it ought to be combined with a strong capability to handle conflicting data and constrain the proliferation of useless inferences. Just as one rationalizes when personally held beliefs or principles conflict with one another, a computer can be made to cope when its individually consistent (piecewise consistent) models of reality are collectively inconsistent.

Coping with PIDs

Assumption-based Truth Maintenance Systems are a step in the right direction. So is Frame Technology.

Frame Technology enables the systematic reconciliation of real-world inconsistencies using a unique compositional approach. Context-sensitive (*i.e.*, operationally efficient) software is manufactured from an inventory of context-free (*i.e.*, reusable) components — frames — which have been software-engineered to be machine-adaptable to an infinite variety of application requirements.

sharpen in particular problem domains. And its behaviour will appear novel, even surprising to its creators, because it will have evolved myriads of its own internal PID representations and meta-representations that defy easy analysis.

When is all this going to happen? Perhaps sooner than you may think. Perhaps not. The hard part of any research is knowing the right questions to ask. If my paradigm for cognition is feasible, it suggests that the right questions concern the discovery of those few double- or triple-meta-representations that span general intelligence. If I am barking up the wrong tree, cognition may well be much further into the future. But sooner or later, like the achievement of flight in artificial birds, the achievement of cognition in artificial intelligence is inevitable. And the world will quickly become a much different place. □

Paul Bassett is Vice-President of Research at Netron Inc, which develops and markets automated software engineering systems based on Bassett Frame Technology, an automated process for designing, manufacturing, and maintaining software. This is the second of two articles based on Mr Bassett's address to the DECUS U.S. Symposium earlier this year.

International Workshop on User Modelling

Robin Cohen
University of Waterloo

An invitational international workshop on user modelling was held August 30—31 at Maria Laach, West Germany. This event was sponsored by the German Science Foundation and was organized by Wolfgang Wahlster and Alfred Kobsa of the University of Saarbrucken. The workshop followed the close of COLING-86, held nearby at the University of Bonn.

A total of 24 researchers from Germany, the U.S., the U.K., the Netherlands and Italy joined yours truly as the representative from Canada to make this a truly international event with a good blend of backgrounds. There were a total of 13 long presentations and 6 short presentations, organized with ample time for questions, to promote discussion. As is usual in a successful workshop, the schedule had to be pushed ahead successively to accommodate the long discussions that followed most presentations.

The aim of the workshop was to chronicle some current research in the area of user modelling for natural language understanding. The subtopics included: the role of user modelling in NL dialogue systems (W. Wahlster, J. Carbonell, A. Kobsa, M. Colombetti), modelling the user's goals and plans (K. Morik, S. Carberry), modelling the user's misconceptions (K. McCoy, B. Goodman), modelling the user's knowledge and beliefs (D. Chin, C. Paris, A. Jameson), and general prospects for the future (T. Finin, K. Sparck-Jones). A collage of topics was offered in the short talks (E. Rich, R. Cohen, A. Joshi, E.Schuster, A. Quilici, W. Mann). Finally, there were participants who contributed to the discussion but did not present talks (D. Appelt, H. Marburger, E. Nessen, B. Neumann, G. Retz-Schmidt, M. Rosner).

The papers presented at the workshop will be published in some form within the next year; the actual venue for publication is still under discussion.

In all, the event was a success, due to the fortuitous selection of interested, inquisitive researchers, tremendous organization by the Saarbrucken duo, and truly pleasant location for the event in a comfortable lodge near an ancient abbey.

Use this form to join CSCSI/SCEIO, to subscribe to *Canadian A. I.* or *Computational Intelligence*, and to order publications

Canadian Society for Computational Studies of Intelligence

Société canadienne pour l'étude de l'intelligence par ordinateur

Application for Membership and / or Journal and Conference Proceedings Order

To join CSCSI/SCEIO and receive *Canadian Artificial Intelligence*, fill out this form (or a photocopy of it) and send it to CIPS (which administers membership for the society) at the address below, with the appropriate fee. You need not be Canadian to be a member. This form can also be used to subscribe to the journal *Computational Intelligence*, and to purchase back issues of *Canadian Artificial Intelligence* and CSCSI/SCEIO conference proceedings. Mail it to:

CIPS, 243 College Street (5th floor), Toronto, CANADA M5T 2Y1

( ) I wish to join CSCSI/SCEIO and receive *Canadian A. I.* ($25).

( ) I am a student (subtract $10). ( ) I am a member of CIPS (subtract $10).

( ) Please send me the following back issues of *Canadian Artificial Intelligence* ($10 each including postage, except $15 for #3):* .....

( ) Please enter my subscription to *Computational Intelligence* at the CSCSI/SCEIO non-institutional member discount rate ($16).†

( ) Please send me the following CSCSI/SCEIO conference proceedings ($30 for 1986; $25 for 1982 and 1980; 1984 is out of print. Add $5 for postage within Canada, $7 outside):

( ) Montréal, 1986. ( ) Saskatoon, 1982. ( ) Victoria, 1980.

Total enclosed: $..... Cdn. / U.S. (Payment may be made in U.S. dollars at the current rate of exchange.)

Name .....

Mailing Address .....

.....

.....

*If an issue you request is out of print, a photocopy will be provided. Issue #3 (March 1985) includes the supplement *Towards a Canadian Fifth Generation Research Plan.*

†*Computational Intelligence* subscriptions are filled by its publisher, the National Research Council of Canada. CIPS only certifies your eligibility for the discount and forwards your order.

Report from Europe

The Seventh European Conference on Artificial Intelligence (ECAI-86)

Takashi Gomi
Applied AI Systems Inc
PO Box 13550
Ottawa, Ont K2K 1X6

The 1986 European AI Conference was held 21–25 July in the Civic Centre in Brighton, the famous British summer resort. The largest-ever ECAI was attended by some 800 participants, including many from the continent and the U.S. Some 90 invited talks, submitted papers, and panels were presented. Two streams were given to long papers and panels (30–60 minutes each), while one was dedicated to short or 'position' papers (15 minutes each). A few open workshops were also held in nearby hotel rooms, including one on connectionism.

The quality of the papers was most satisfying. The presentations of the papers in all streams were in most part rigidly synchronized. This, combined with the fact the long paper sessions were presented in neighbouring halls, made movement between the streams very easy, unlike the chaos at IJCAI–85. In all, one could see a lot of thinking had gone into the preparation of the conference (programme, organization of the proceedings, grouping of the submitted papers, the nature and frequency of of invited papers, the length and timing of breaks and lunches, etc.). The organizers, in particular general chairman Benedict du Boulay of Sussex, the program chairman Luc Steels of Vrije Universiteit Brussels, and local arrangements chairman David Hogg of Sussex, clearly succeeded in making the conference stimulating as well as enjoyable for the majority of those attending.

In addition to more conventional expert systems, natural language processing, and robotics tutorials, an impressive series of tutorials addressed advanced issues, such as planning, learning, and qualitative reasoning. Exchanges were lively during question period, and the lectures stimulating. In particular, the QR seminar, given by Pat Hayes (Schlumberger) and A.G. Cohen (Essex), was very comprehensive, covering the history between the 'Naive Physics Manifesto' (Hayes 1979) and the latest developments in the field. The relationship between QR and common-sense reasoning and causal models was also discussed.

During the conference, W.F. Clocksin and A.J. Morgan of Cambridge University gave a paper in which they had applied qualitative modelling for the synthesis of control methods. The new sub-discipline is appropriatively called 'Qualitative Control'. A simple simulation experiment of the coupled-tanks plant amply demonstrated the effectiveness of the approach, to the obvious satisfaction of many in the audience. A team from Italy led by F. Garini of Ispra gave a paper on the qualitative modelling of liquid molecules as actors.

The three advanced tutorial topics are the key components of an intelligent autonomous entity. That the organizers of the conference chose them as subjects at this time was suggestive of the general trend in European AI research.

Expert Systems

Several impressive presentations of the intermediate results from the EC's ESPRIT programme, now completing its second year, were seen during the conference. One of them, a deep expert system approach applied to an industrial system (Gallanti et al.), drew considerable attention for its effective use of a deep causal modelling to a practical application. Again, the influence of qualitative reasoning was visible, as well as Randy Davis's diagnosis based on structure and function, in the modelling and the method of predicting system behaviour. The government engineering research organisation (CISE) has cooperated with Politecnico di Milano (host of IJCAI–87) to construct a realtime expert system to monitor the operation of the steam condensor in a thermal power plant. A detailed case study was presented.

François Jakob of Laboratoires de Marcoussis, France, gave a paper on the EXTASE system, a

similar expert system to CISE's; this one processes alarms from the vacuum distillation tower of a refinery. His group used a truth-maintenance mechanism to support a framework for hypothesis-driven reasoning. This architecture is important as the tolerance to sensor failure is mandatory in their system.

Another ESPRIT consortium, composed of three researchers from Denmark (R.P. Worden et al., Nordjysk Udviklings Centre) and Logica of Britain tackled the serious problem of "What does it take to be a truly effective decision aid, now that expert systems are best conceived of as decision aids for human operators, rather than being decision makers themselves?" They contend that the key question of how to make expert systems useful had long been ignored, resulting in a large number of expert systems that were useless in practical applications. In particular, many expert systems tended to lose track during a consultation session, and became totally oblivious to the intentions of the user for the rest of consultation session.

The team recommends separating knowledge required to be a good assistant from that required to deal with domain issues, and have devised a plan for the interaction between the two through an example expert system in the domain of electromyography. The system demonstrates the basic elements of responsible assistant behaviour. The paper then proposes extensions to the model, which would link it to the forefront research issues in AI in general. These include reason maintenance, rule learning, deep expert systems, and intelligent tutoring systems.

A work by Wielinga and Breuker of the University of Amsterdam titled "Models of Expertise" contains an attractive theoretical proposition. In order to preserve the extreme flexibility that experts have when performing a task, the group offers a hierarchical model to represent the experts' knowledge. The management of complexity is another goal of their research. They contend that, by applying the "divide and conquer" technique to a mixed level of abstractions, thus clarifying the hierarchical structure of the knowledge, the complexity could be kept to a minimum.

The domain level, the lowest level in the knowledge hierarchy, represents concepts, relations, and structures through axiomatic objects. This well corresponds to today's many other KR methods. The level above, the inference level, describes the domain level by looking after meta-classes and knowledge sources embedded in the inference structure. This again is very common in many existing schemes. The task level then manages the inference level by handling goals and tasks through the task structure objects. The strategic level governs the task level by incorporating plans, meta-rules, repairs, and impasses as objects. The paper emphasizes that such a hierarchy is effective in acquiring the expert's knowledge,which is often expressed randomly at various levels of abstraction.

The research is a formalization of various thoughts and ideas generated by others in recent years: Alan Bundy (metalevel inference); Randy Davis (reasoning about control); William Clancey (control knowledge representation); and De Greef (hierarchical knowledge acquisition).

Natural Language

In the natural language session, Margaret King of the University of Geneva gave a survey paper on current work in machine translation research. Pessimism dominated the talk. King has recently published *Machine Translation Today: the State of the Art*, which appears to be a 'must read' book for understanding the field.

Professor King first introduced a few syntactically oriented pre- or post-editing translator's aids as practical commercial products, using one such system, ALPS, as an example. She contends that such systems "are quite 'successful', much to the surprise of the academic community." The success hinges on the dramatic increase in translators' productivity realized with the aid. The more advanced systems, such as METAL (University of Texas), Mu (Kyoto University), and the European Community's Eurotra, were introduced next. Eurotra intends to cover all nine official languages of the Community.

Apparent bias in favour of linguistic approaches, and the tendency to stay away from the use of world knowledge, thus keeping the model highly general, are evident in Mu, Eurotra, and the work by Vauquois and Boitet, which influenced the first two. Remedies to the overgeneralization in Europa were well outlined. The 'compositional mapping' theory currently entertained by the Eurotra group was explained in detail.

Eventually, the "researchers' delight" nature of the project, which has already consumed 27 million ECU (approximately $38 million Canadian), became obvious. I can only recall a strong criticism of the project by Steven Young of Cambridge (Speech Recognition), who has built a remarkable prototype of a speech understanding system under the Alvey Programme: "You can't build an understanding system without world knowledge. I can't see how it [Eurotra] would ever work."

Professor King concedes: "present day systems [to replace human translation] are not that good, and there is no immediate prospect of them getting dramatically better."

The Dyer clan (Quilichi, Dyer, Flowers) of UCLA presented their AQUA intelligent Unix advisor, one of the several hopefuls at the UCLA AI Lab as the successor to the BORIS deep understanding model. It is a substantial departure from Wilensky's UC, which now appears greatly simpler in comparison. Consistent with the Schankian school of cognitive modelling, AQUA adopts an elaborate structure that tries to represent the user's intentions. A problem-classification technique and the advice generation based on a similar experience that the system recalls take over. Wilensky's famous "remove file" situation is represented in AQUA as follows, complete with rich natural language parsing and generation:

User:

> I tried to remove a file with the "rm" command. The file was not removed, and the error message was "permission denied". I checked and I own the file.

AQUA:

> To remove a file, you need to be able to write into the directory containing it. To remove a file, you do not need to own it.

The system does not simply show the way to carry out the desired task, but attempts to understand the user's misconception and correct it. Like many other understanding systems of the type today, it takes several minutes to compute the above example on an Apollo, and hence is still a few years from practical application. Nevertheless, AQUA currently has a memory of about 50 planning experiences and has been used to process at least one story in each user problem class.

Paul Jacobs of GE (Schenectady) presented a new natural language generator called KING (Knowledge INtensive Generator). He uses the 'Ace' knowledge representation framework (Jacobs, ECAI−84) to organize linguistic and conceptual knowledge into hierarchies, and "structural associations" to join metaphorically and referentially related knowledge structures. His argument of improved extensibility and adaptability of the generation system was convincing. KING can be considered one of the key new generation natural language generators that may replace David McDonald's long-lived Mumble/Decision Tree model.

Invited Talks

Invited talks were given by Ehud Shapiro (Weizmann Institute), Jorgen Siekmann (Universität Kaiserslautern), Gianfranco Prini (Universita' di Pisa, Delphi), and Danny Hillis (Thinking Machines Inc.).

Both Shapiro and Siekmann gave an overview lecture on their field of research: Shapiro on concurrent Prolog and Siekmann on unification. Shapiro's lecture was very similar to one he had given in other places (FGCS'84 Tokyo and SLP−85 Boston, for example), clear and well summarized. Professor Siekmann also gave a well-structured introduction on Unification covering history, applications, the special theory, and the general theory. He also provided a well-written paper for the proceedings.

Professor Prini is an Associate Professor of Computer Science at Universita' di Pisa, and also the president of a fledging AI company Delphi. He gave a lecture on the prospect of the AI business in Europe. It was well researched and summarized. Only his manner of presentation was severely disorganized. The points are well made, though. He predicts that Europe is facing a widespread economic opportunity with a stable growth rate. He contends high-technology companies should exploit the situation. In particular, AI would be in high demand as a mandatory step to the growth of high technology, as the shortage of software production capability will be felt in Europe soon. He thinks there is no other way to circumvent it. AI would provide a means to mitigate the crisis by providing methods of representing the functionality at a higher level of abstraction (not limited to the 'declarative' nature of logic programming).

In spite of the long history of AI R&D in Europe, he continues, the business aspects of the technology leaves lots to be desired there. He attributes most of the difficulties to a medieval banking system ("Essentially the way de' Medici set up 400 years ago"), the small and fragmented market, lack of large customers, and obsolete universities. The rest of his talk attempted to give suggestions for the remedy.

Danny Hillis, from Thinking Machines Inc., gave a convincing presentation and an appealing demonstration of their first version of the Connection Machine. The structure, operating principle, methods of programming, and the strength and the limitations of the machine were all very well explained. A sample Common Lisp program written both for conventional machines and for the Connection Machine clearly highlighted what would be required to do in parallel AI programming.

Finally, Hillis showed a false-colour video tape of the output from a simulation of fluid flow against barriers. An average of 16 million particles involved in the flow were simulated at near real-time. Both the initial shock wave and the stable state of the flow were shown in a two-dimensional plane in colour. The display was attractive, as well as highly provocative. Hillis implied the role of the machine in naive physics allows scientists to investigate the qualitative behavior of fluid. He thinks this type of application will be the immediate contribution of the machine to real-life problem solving. He certainly anticipates, however, the AI applications that will follow in a similar manner. (I want to see the learning process in the Boltzmann machine displayed this way!)

Panels

Two panels were held: one on social responsibility of AI research and other on the European AI business.

The social-responsibility panel, which was chaired by Professor Margaret Boden of Sussex, included Alan Bundy of the University of Edinburgh and two lawyers knowledgeable in AI. It did not go the way organizers had planned. At times the session became quite boring. It picked up Alan Bundy's concern that there were some questionable AI businesses operating and their poor-quality products or misleading claims were damaging the credibility of AI. This would eventually lead to reduced funding for legitimate AI research. If someone not knowledgeable purchases a 'natural language system', he/she would anticipate all utterances would be readily understood by the system, he warns. He proposes that a "code of practice" for AI products be administered by vendors' associations.

The lawyers, of course, were supportive of Bundy's proposal, though differing in the details in approach. They spoke legalese fluently, explaining case laws and self-regulations, but it became obvious that only a few of the audience supported their concern for charlatans possibly drawing the community into AI winter. One insisted that damage done by AI systems would not be much different from those by conventional software. Others suggested that there were enough market forces to eliminate offending ones very quickly. Some pointed out the difficulty of drawing up or enforcing such standards. Someone pointed out that Bundy's claim was yet another reflection of the standard British "nanny mentality", overprotecting consumers. So much for his method of "cowboy bashing".

The true excitement began when someone in the audience stood up and suggested, "Why aren't the panel addressing the issues that I would see as the really important social implications about artificial intelligence?" Chairman Margaret Boden quickly intervened and attempted to shut the questioner flat. Squabbles continued for a while. Eventually, Professor Boden ruled questions concerning military AI as "out of order". It was as if Professor Boden, a sociology professor at Sussex, who is known as a well-established authority on the subject of "Social Issues of AI", had decided not to answer the acute questions. The atmosphere after the declaration resembled a meeting of the Commonwealth delegates in which Prime Minister Thatcher held tightly to a "principle". To be fair, she was trying to conduct some practical conversations. Note, we Canadians were more open and flexible on this at CSCSI−86 in Montreal.

The other panel was titled something like "Is there a future for the European AI business?" Those participating included various EC officials, representatives from active AI businesses, and funding agencies. A sombre tone dominated the panel. The American AI businesses were depicted as a dominating force, and various approaches to assure European success were discussed. Japan was also considered an eventual grave threat against which the panel did not seem to have suggestions to protect the European AI industry. Only Professor Prini maintained a positive perspective for the future of Europe. Given by a practitioner both in academia and industry, it was a relief.

At the end of each day, a half-hour talk was planned by a 'Visionary Speaker'. Two impressive talks were given: one by Harold Kahn of the U.S. and the other by Sir Clive Sinclair of Sinclair Research Limited.

Kahn, the well-known computer artist, used visual facilities effectively to review about 10 years of development of his computer-generated paintings. The development was almost a reversal of actual historical development of painting, however. His earlier work (late 1970's) looked very much like Kandinski's proto-abstract paintings of blotches and streaks of bright colours (one particularly resembles the "The First Abstract Watercolour" of 1910 by Kandinski). This eventually passed the stage where loosely drawn figures resemble Art Nouveau figures of the turn of the century, which soon progressed into Cézannean group figures of the early 1890s (à la Bathers). His latest works now combine the casualness of Impressionists' relaxed domestic figures with dominating trees and other plant

forms of the Barbizon School.

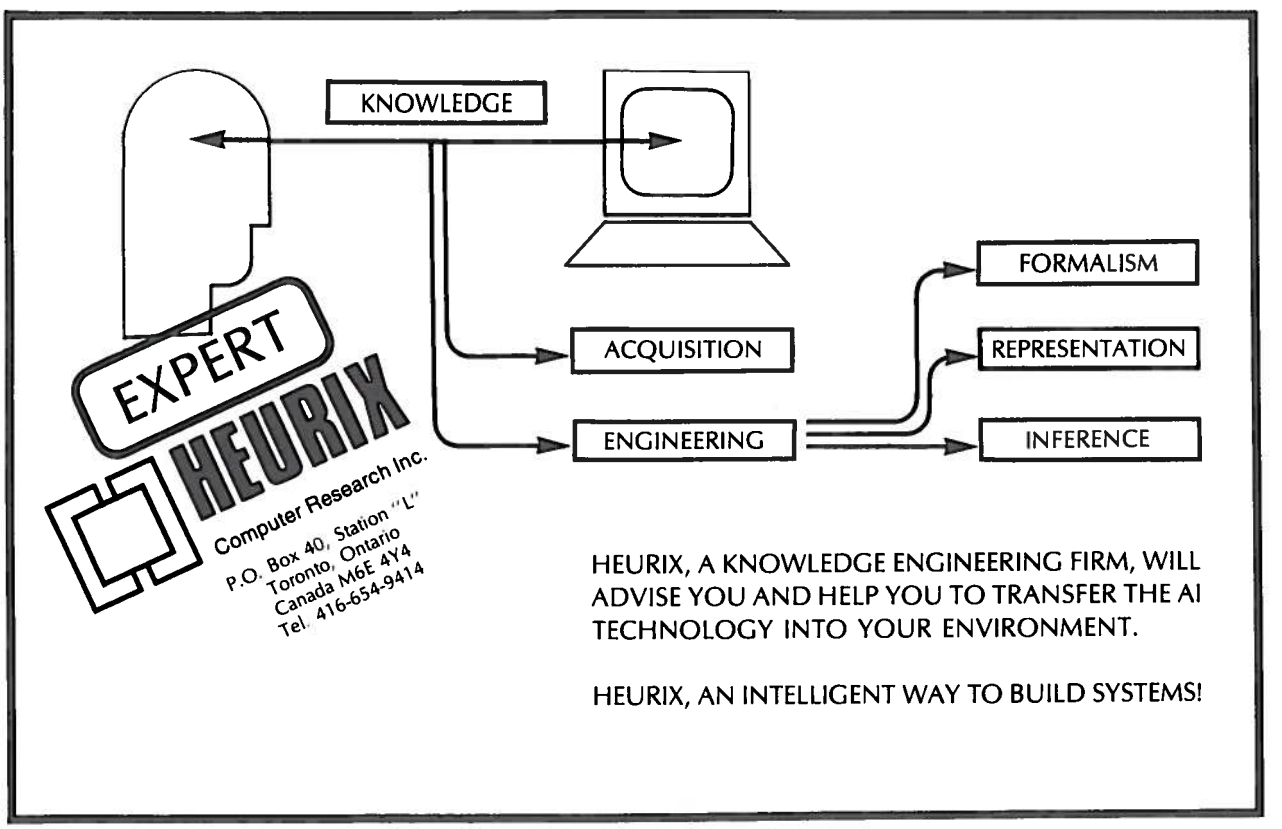
The reversal of progress in his art is solely due to the use of increasingly refined knowledge representation methods he incorporated incrementally into his generators. The final works depicted the "Statue of Liberty" in striking reality complete with the decadance and grossness of Rococo festivity of the 18th century — his presentation was symbolizing what has happened in the past decade or so of the history of AI itself. Who said local knowledge was that important?

It was clear Sir Clive Sinclair was profoundly respected at the conference. The British national hero who put hand-held calculators in everybody's hand and home computers in millions of homes spent the entire 45 minutes talking about Sci-fi AI. According to his predictions, by the year 2000, we will begin to use androids of reasonable intelligence. By 2010, such intelligent creatures will be widely used. By 2020, all modes of transportation will become autonomous. We will be facing a social problem involving androids by 2040 or so. Given with the tone of authority of the old gentleman who devoted his life to the promotion of science in a layman's world, the sermon sounded authentic. Unyielding tone and posture gave credence to his statements. Have you seen an industrialist this determined lately in this beautiful country of ours? Not just hard-nosed, but one with vision and a good deal of self-sacrifice, I mean. Come to think of it, the British AI scene is full of this lot: Michael Brady of Oxford, David Thomas of Imperial College, Brain Oakley of Alvey, Karen Sparck Jones of Cambridge, Donald Michie of Turing Institute. In places, his lack of detailed knowledge in AI showed. Who cares? They are achieving what they said they would five years ago.

Question period was active too. Contrary to my expectation, several hundred filled the hall, most of them academics, and in general they supported his notion. Only a few complained: "I think you should be talking about centuries, not decades, to see that kind of progress."

The exhibition was very much like the ones at IJCAI and AAAI, except much smaller. Exhibition by many of today's key players in the AI business (e.g., Symbolics, Carnegie Group, Sun, Digital) were augmented by European consultancies and manufacturers (e.g., Delphi, Cognitive Applications of Sussex, CGE Research Centre and Ferranti of the U.K., and Framentec of France). In all, some 30 exhibitors were present (compared to 100 at AAAI−86 in Philadelphia). □

Takashi Gomi is the founder of Applied AI Systems, a software and consulting firm.



Book Reviews and Publishing News

Implementing Mathematics with the Nuprl Proof Development System

R. L. Constable et al.

[Computer Science Dept., Cornell University]

Prentice-Hall, Englewood Cliffs, 1986, 299 pp.
ISBN 0-13-451832-2

Reviewed by
Innes A. Ferguson
Bell-Northern Research

The main purpose of this book is to describe the Nuprl (Proof Refinement Logics, version "nu") computer system, developed by Constable et al. at Cornell University in order to assist users with the task of mathematical problem solving. This text is primarily aimed at mathematics and computer science undergraduates, although it should be pointed out that certain reference sections will be of interest only to actual users of the system.

Nuprl uses a formal theory of mathematics known as constructive logic to support the interactive creation of proofs and formulas. Together with associated definitions and theorems, these can be used to generate and evaluate new proofs. The system also includes a programming language (the metalanguage ML), primarily as a means for developing proof-generating programs. One of the distinguishing features about Nuprl is that unlike most other theorem-generating systems, it claims to provide a uniform language for expressing mathematics, namely constructive type theory.

The content of the book can be divided into three sections: a system tutorial, a reference manual, and a summary of research conducted around Nuprl. Worth reading before the first of these are the last two sub-sections of the overview chapter (1), which discuss the place of Nuprl in computer science, and in particular its relationship to logic, semantics, automated reasoning, and artificial intelligence.

The tutorial section comprises chapters 2 to 6. Chapter 2 introduces types in Nuprl, both by extending the typed lambda calculus, and by relating the idea of a type in Nuprl to the concept of a set, and to the concept of a data type in a programming language. Chapter 3 introduces the major system components from a user's viewpoint, with the purpose of explaining how to write statements and definitions to express mathematical propositions. Conceptual issues involved in using type theory are also discussed. Chapter 4 gives an informal introduction to proofs in Nuprl, an overview of the commands required to create theorems, and some example proofs drawn from logic and elementary number theory. Chapter 5 describes the system as a programming tool: how terms are extracted from proofs and computed to return values. Finally chapter 6 introduces the concept of refinement and replacement tactics (theorem-proving heuristics) and the metalanguage ML in which they are written.

The reference section spans chapters 7 to 9. The first of these chapters describes the various parts of the Nuprl system: the command language, the object library, window management, the text and proof editors, together with an introduction to text macros or definitions. Besides giving a brief description of the theoretical foundations of Nuprl (its semantics and type system), chapter 8 also provides a summary of the system's refinement rules for constructing proofs, followed by their associated ML constructors. Chapter 9 introduces Milner's ML in some more detail, and extends the explanation of tactics introduced earlier in the text, showing which tactics are made available by the system, and how to construct new ones from provided tools.

The remaining three chapters constitute the section on Nuprl research, and will mainly be of interest to advanced users of the system. Chapter 10 contains several recommendations for constructing non-trivial theorems, working with proofs and structuring knowledge in a theory by using definitions and various abstraction techniques. Chapter 11 gives a description of some pre-provided mathematical libraries for such things as manipulating lists and sets, handling rational and real numbers, and even implementing a denotational semantics for a simple programming language. The last chapter of the book looks at some proposed extensions to the system;

in particular, recursive definitions to handle inductive and partial function types.

The text is both interesting and well written, and should prove to be a useful contribution to the fields of automated reasoning (in particular, theorem generation and manipulation) and semantics. The subject matter is fairly mathematical and requires some familiarity with formal logic and type theory. □

Innes Ferguson is with the Knowledge Technology Group (AI Development) at Bell-Northern Research, Ottawa

Readings in Artificial Intelligence and Software Engineering

Charles Rich and Richard C. Waters (editors)

Los Altos, CA: Morgan Kaufmann Publishers, 1986, xxi+602 pp
ISBN 0-934613-12-5, softbound, $US26.95

Reviewed by
Evangelos E. Milios
University of Toronto

Archival collections of papers are an efficient way of disseminating knowledge about a field that is not yet mature enough to be presented in textbook form. AI in software engineering is definitely such a field because, despite the fact that efforts to bring artificial intelligence ideas and techniques to bear on software engineering are as old as AI itself, programming is still more art than science, with commercialization of research results not yet in sight.

Relevant questions to be answered for the prospective reader about the book in review could be: Is the collection well-balanced? Is it current? Is the classification of the papers widely acceptable and useful? How much guidance does it offer the reader in putting the material in perspective?

The book includes 34 articles, divided into 11 chapters, starting out with an eleven-page introduction to the field of AI and software engineering, written by the editors. Twenty-three of the papers were published after 1980, and twelve of them were published within the last two years. Therefore the collection is very much current. The introduction is a useful map of the material in the book and the field as a whole. Each chapter is preceded by a short commentary, which attempts to put the papers of the chapter in perspective. I find this style (the same as that of *Readings in Knowledge Representation*, published in 1985 also by Morgan Kaufmann) very useful, because it enables the reader to quickly get an overview of the material without having to read through each individual article.

One thing I found misleading about the book is its title, which seems to promise balanced coverage of two subareas, AI in software engineering and the software engineering of AI, *i.e.*, AI programming. However, only two out of the 34 papers specifically address AI programming (chapter XI) by presenting a very-high-level overview of the area. The coverage of AI in software engineering, though, is sufficiently complete and well-balanced.

The book starts with two chapters on deductive synthesis and verification of programs. Then it goes on to transformational approaches (chapter III) (between programs at roughly the same level of abstraction) and very-high-level languages (chapter V). Chapters on intelligent assistants (chapter VII) and programming tutors (chapter VIII) represent efforts to provide partially automated solutions to the programming task by constructing assistant systems that interact with the programmer. The book ends with chapters on programming knowledge (chapter IX) (covering several proposals for encoding programming knowledge as frames/schemata/plans and an interesting experimental study supporting the fact that expert programmers have and use programming plans as well as programming conventions), the role of domain knowledge in programming and requirements specification (chapter X), and finally artificial intelligence programming (chapter XI). Two chapters (IV and VI) cover natural language program specifications and programming by example respectively, two areas in which not much work is currently going on.

Although the lines between the areas addressed in different chapters are often fuzzy, such as between transformational approaches and very-high-level languages, the overall classification of papers makes good sense and proceeds from the more theoretical in the beginning of the book to the more practically relevant towards the end. The book does have a comprehensive bibliography, which however is not annotated.

In summary, I feel that the book has been successful in providing a well-rounded overview of the current state of affairs of AI in software engineering. As such, it is suitable as a textbook for an advanced graduate course and as a reference book for researchers in the field. Software practitioners might find useful some of the later chapters of the book, especially on programming knowledge and AI programming. □

Evangelos Milios is a Research Scientist in the Artificial Intelligence Group, Department of Computer Science, University of Toronto.

Books Received

Motion: Representation and perception (Proceedings of the ACM Siggraph/Sigart Interdisciplinary Workshop)

Norman I. Badler and John K. Tsotsos (editors)
[University of Pennsylvania and University of Toronto]

NY: North-Holland, 1986, xi+345 pp,
Hardbound, ISBN 0-444-01079-3, $^{US}59.00

On knowledge base management systems: Integrating artificial intelligence and database technologies

Michael L. Brodie and John Mylopoulos (editors)
[Computer Corporation of America and University of Toronto]

(Topics in information systems, no. 4)
NY: Springer-Verlag, 1986, xxi+660 pp.
Hardbound, ISBN 0-387-96382-0, $^{US}38.00

Natural language processing: A knowledge-engineering approach

Richard E. Cullingford
[Georgia Institute of Technology]

Totowa, NJ: Rowman & Littlefield, 1986
Cloth, ISBN 0-8476-7358-8, $^{US}36.95

Prolog programming for artificial intelligence

Ivan Bratko
[E. Kardelj University, Yugoslavia]

(International computer science series)
Wokingham, England: Addison-Wesley, 1986
Paper, ISBN 0-201-14224-4, $^{CDN}39.90

Essential Lisp

John R. Anderson, Albert T. Corbett, and Brian J. Reiser
[Carnegie-Mellon University and Princeton University]

Reading, MA: Addison-Wesley, 1987
Paper, ISBN, 0-201-11148-9, $^{CDN}34.25

Readings in natural language processing

Barbara J Grosz, Karen Sparck Jones, and Bonnie Lynn Webber (editors)
[Harvard University, University of Cambridge, and University of Pennsylvania, resp.]

Los Altos, CA: Morgan Kaufmann Publishers, 1986, xv+664 pp. Paperbound,
ISBN 0-934613-11-7, $^{US}26.95

Common LISPcraft

Robert Wilensky
[University of California, Berkeley]

NY: Norton, 1986, xvii+500pp
Paperbound, ISBN 0-393-95544-3, $^{US}26.95

Decision and intelligence (Robot technology, volume 6)

Igor Aleksander, Henri Farreny, and Malik Ghallab
[Imperial College of Science and Technology, London]

Englewood Cliffs, NJ: Prentice-Hall Inc and
Toronto: Prentice-Hall Canada Inc, 1987
Hardbound, ISBN 0-13-782079-8; 203 pp.
$^{CDN}60.15
[Translated from French. Original edition published 1986 by Hermes Publishing.]

Indexes and Bibliography (Robot technology, volume 8)

Englewood Cliffs, NJ: Prentice-Hall Inc and
Toronto: Prentice-Hall Canada Inc, 1987.
Hardbound, ISBN 0-13-782046-1; 105 pp.;
$^{CDN}40.55 [Original edition published 1986 by Kogan Page Ltd]

Intelligent information systems: Progress and prospects

Roy Davies (editor)
[Univerity of Exeter]

(Ellis Horwood series in artificial intelligence)
Chichester: Ellis Horwood; Distibuted in Canada by John Wiley Hardbound, ISBN 0-85213-896-0;
300 pp.; $^{CDN}82.95

Abstracts of papers in *Computational Intelligence*, 2(4), November 1986

Learning Rules for Graph Transformations by Induction from Examples

Malcolm Bersohn
Department of Chemistry
University of Toronto

The input to the described program, in learning mode, consists of examples of starting graph and result graph pairs. The starting graph is transformable into the result graph by adding or deleting certain edges and vertices. The essential common features of the starting graphs are stored together with specifications of the edges and vertices to be deleted or added. This latter information is obtained by mapping each starting graph onto the corresponding result graph. On subsequent

input of similar starting graphs without a result graph, the program, in performance mode, recognizes the characterizing set of features in the starting graph and can perform the proper transformation on the starting graph to obtain the corresponding result graph. The program also adds the production to its source code so that after recompilation it is permanently endowed with the new production. If any feature that lacks the property "ordinary" is discovered in the starting graph and only one example has been given, then there is feedback to the user including a request for more examples to ascertain whether the extraordinary property is a necessary part of the situation.

On the Consistency of Commonsense Reasoning

Donald Perlis
Department of Computer Science
University of Maryland

Default reasoning is analyzed as consisting (implicitly) of at least three further aspects, which we call oracles, jumps, and fixes, which in turn are related to the notion of a belief. Beliefs are then discussed in terms of their use in a reasoning agent. Next, an idea of David Israel is embellished to show that certain desiderata regarding these aspects of default reasoning lead to inconsistent belief sets, and that as a consequence the handling of inconsistencies must be taken as central to commonsense reasoning. Finally, these results are applied to standard cases of default reasoning formalisms in the literature (circumscription, default logic, and non-monotonic logic), where it turns out that even weaker hypotheses lead to failure to achieve commonsense default conclusions.

A framework for computing extra-sentential references

Tomek Strzalkowski and Nick Cercone

We are concerned with developing a tractable method for computing references over sentence boundaries. Our lambda-calculus–based meaning representation incorporates valuable features of Fregean-type semantics (à la Lewis, Montague, Partee, and others) along with features of situation semantics devoloped by Barwise, Perry, Partee, etc. In the theory we develop, our concern is *how to get the reference.* We consider a series of selected *two-sentence stories,* which we use to illustrate referential interdependencies between sentences. We explain the conditions under which such dependencies arise and formalise a set of rules that specify how to *compute* the reference. We restrict the discussion at this time to two-sentence stories to avoid most of the problems inherent in *where to look for the reference.* We are aware of this problem but restrict our considerations in this paper to situations where a reference, if it can be computed at all, has a unique antecedent. Thus we consider examples such as *John wants to catch a fish. He (John) wants to eat it* and *John interviewed a man. The man killed him (John).*

Subscription information

Computational Intelligence is published by the National Research Council of Canada and sponsored by CSCSI/SCEIO. Non-institutional CSCSI/SCEIO members may subscribe for $^{CDN}16, a considerable discount on the regular price. To subscribe, use a copy of the all-purpose order form in this issue of *Canadian A. I.* The form must be sent to CIPS, who will certify your membership and forward your request to the NRCC. If you wish to subscribe without joining CSCSI/SCEIO, write to: Distribution R-88 (*Computational Intelligence*), National Research Council of Canada, Ottawa, Ontario, CANADA K1A 0R6. Regular rates are $^{CDN}37 for individuals, $^{CDN}75 for libraries; add $10 extra for postage outside Canada. Make cheques payable to "Receiver General of Canada, credit NRCC".

University of Waterloo
Department of Computer Science

The University of Waterloo invites applications for faculty positions at all ranks. A Ph.D. in Computer Science is required, with evidence of outstanding research accomplishment or potential. Salary according to experience. Applications should include a curriculum vitae and the names of three references and should be directed to F.W. Tompa, Chairman, Recruiting Committee, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

The Computer Science Department at the University of Waterloo, part of the Faculty of Mathematics, is a diversified group of about 40 faculty members, conducting research in artificial intelligence, logic programming, graphics, distributed systems, VLSI, analysis of algorithms, databases (including dictionary databases), data structures, symbolic computation, parallel processing, computer networks, multiprocessor systems and other areas of Computer Science.

In accordance with Canadian immigration requirements, although all applicants will be considered, preference must be given to Canadian citizens and permanent residents of Canada.

Forthcoming Conferences, and Calls for Papers

Canadian Conferences

Second International Workshop on Natural Language Understanding and Logic Programming

17– 20 August 1987

Vancouver, B.C.

The workshop will consider fundamental principles and important innovations in the design, definition, uses and extensions of logic programming to natural language understanding and, conversely, the adequacy of logic programming to express natural language formalisms.

The main topics of interest are logic grammar formalisms; formal representations of natural language sentences and texts; natural language generation; uses of techniques for logic grammars in other grammar formalisms; compilers and interpreters for grammar formalisms; and applications (natural language front ends).

For more information and a copy of the call for papers, write to:

Patrick Saint-Dizier
Department of Computer Science
Simon Fraser University
Burnaby, B.C. V5A 1S6

□ □ □ □ □

Second Knowledge Acquisition for Knowledge-Based Systems Workshop

19– 23 October 1987

Banff, Alberta

A problem in the process of building knowledge-based systems is acquiring appropriate problem-solving knowledge. The objective of this workshop is to assemble theoreticians and practitioners of A.I. who recognize the need for developing systems that assist the knowledge-acquisition process.

To encourage the exchange of ideas, the workshop will be limited to about 40 participants. Attendance will be limited to those presenting their work, one author per paper. Five copies of an abstract or a full-length paper should be forwarded, by 15 April 1987, to:

John Boose
Advanced Technology Center
Boeing Computer Services
PO Box 24346
Seattle, Washington 98008

□ □ □ □ □

International Conference on Computer-Assisted Learning in Post-Secondary Education

5– 7 May 1987

University of Calgary
Calgary, Alberta

For more information, write to:

Heidi Guenter, Conference Office
Education Tower, Room 102
University of Calgary
Calgary, Alberta T2N 1N4
Phone: 403-220-4987

□ □ □ □ □

U. S. Conferences

National Conference on Artificial Intelligence and Law

27– 29 May 1987

Northeastern University
Boston, Massachusetts

In recent years there has been an increased interest in the applications of artificial intelligence to law. Some of this interest is due to the potential practical applications. A number of researchers are developing legal expert systems, intended as an aid to lawyers and judges; other researchers are developing conceptual legal retrieval systems, intended as a complement to the existing full-text legal retrieval systems. But the problems in this field are very difficult. The natural language of the law is exceedingly complex, and it is grounded in the fundamental patterns of human common-sense reasoning. Thus, many researchers have also adopted the law as an ideal problem domain in which to tackle some of the basic theoretical issues in AI: the representation of common-sense concepts; the process of reasoning with concrete examples; the construction and use of analogies. There is reason to believe that a thorough interdisciplinary approach to these

problems will have significance for both fields, with both practical and theoretical benefits.

The purpose of this First International Conference on Artificial Intelligence and Law is to stimulate further collaboration between AI researchers and lawyers, and to provide a forum for the latest research results in the field. The conference is sponsored by the Center for Law and Computer Science at Northeastern University. The General Chair is:

Carole D. Hafner
College of Computer Science
Northeastern University
360 Huntington Avenue
Boston, Massachusetts 02115
Phone: 617-437-5116 or -2462
INTERNET: hafner.northeastern@csnet-relay

Authors are invited to contribute papers on the following topics:

- Legal Expert Systems
- Conceptual Legal Retrieval Systems
- Automatic Processing of Natural Legal Texts
- Computational Models of Legal Reasoning

In addition, papers on the relevant theoretical issues in AI are also invited, if the relationship to the law can be clearly demonstrated. It is important that authors identify the original contributions presented in their papers, and that they include a comparison with previous work.

Authors should submit six (6) copies of an extended abstract (6 to 8 pages) by 15 January 1987 to the Program Chair:

L. Thorne McCarty
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
Phone: 201-932-2657
INTERNET: mccarty@rutgers.arpa

Notification of acceptance or rejection will be sent out by 1 March 1987. Final camera-ready copy of the complete paper (up to 15 pages) will be due by 15 April 1987.

□ □ □ □ □

Third Workshop on Uncertainty in Artificial Intelligence

10– 12 July 1987

Seattle, Washington

This is the third annual AAAI workshop on uncertainty in AI. The first two workshops have been successful and productive, involving many of the top researchers in the field. The 1985 workshop proceedings have just appeared as a book, *Uncertainty in Artificial Intelligence*, in the North-Holland Machine Intelligence and Pattern Recognition series. The general subject is automated or interactive reasoning under uncertainty.

This year's emphasis is on the representation and control of uncertain knowledge. One effective way to make points, display tradeoffs and clarify issues in representation and control is through demonstration in applications, so these are especially encouraged, although papers on theory are also welcome. The workshop provides an opportunity for those interested in uncertainty in AI to present their ideas and participate in discussions with leading researchers in the field. Panel discussions will provide a lively cross-section of views.

Papers will be carefully reviewed. Space is limited, so prospective attendees are urged to submit a paper with the intention of active participation in the workshop. Preference will be given to papers that have demonstrated their approach in real applications; however, underlying certainty calculi and reasoning methodologies should be supported by strong theoretical underpinnings in order to best encourage discussion on a scientific basis. To allow more time for discussion, most accepted papers will be included for publication and poster sessions, but not for presentation.

Four copies of a paper or extended abstract should be sent to the program chairman by 10 February 1987. Acceptances will be sent by 20 April 1987. Final (camera-ready) papers must be received by 22 May 1987. Proceedings will be available at the workshop.

For more information, contact:

Tod Levitt
Advanced Decision Systems
201 San Antonio Circle
Suite 286
Mountain View, California 94040
Phone: 415-941-3912
INTERNET: levitt@ads.arpa

□ □ □ □ □

Fourth Symposium on Logic Programming

31 August to 4 September 1987

San Fransisco, California

The Conference solicits papers in all areas of logic programming. Please submit full papers, indicating accomplishments of substance and novelty, and including appropriate citations of related work. The suggested page limit is 25 double-spaced pages. Send eight copies of your manuscript no later than 21 February 1987 to:

Seif Haridi

SLP'87 Program Chairperson
Swedish Institute of Computer Science
Box 1263
S-163 13 Spanga
Sweden
INTERNET: nea!sics!seif@mcvax.uucp

Acceptance will be mailed by 15 April 1987. Camera-ready copy will be due by 7 June 1987.

□ □ □ □ □

Expert Systems in Telecommunications

9 March 1987

New York, New York

A one-day symposium focussing on network design, management, and maintenance, as well as the user interfaces to these systems.

For more information, contact:
Doug Antonelli
IBM — E04/664
P.O. Box 12195
Research Triangle Park, North Carolina 27709
Phone: 919-254-0147

□ □ □ □ □

Outside North America

AI87 JAPAN

27— 30 October 1987

Osaka, Japan

For more information, contact:
Secretariat of AI 87 JAPAN
Shohaku Bldg.6-23, Chayamachi Kitaku,
Osaka, 530 Japan

□ □ □ □ □

Seventh International Workshop on Expert Systems and Their Applications

13— 15 May 1987

Avignon, France

"The largest European event dedicated to artificial intelligence." For more information, contact:
Avignon'87
BP 45
92193 Meudon Cedex
France

□ □ □ □ □

Second International Symposium on Knowledge's Engineering

6— 10 April 1987

Universidad Politécnica
Madrid, Spain

For more information, contact:
D. Josè R. Chelala
Alvarez de Baena, 3—2°
28006 Madrid, Spain

AAAI Sixth National Conference on AI

The AAAI is inviting academic institutions and non-profit research laboratories to participate in the Exhibit Program at the Sixth National Conference on Artificial Intelligence, 14—16 July 1987, in Seattle, Washington. This program, which was initiated last year, was considered one of the highlights of the 1986 conference.

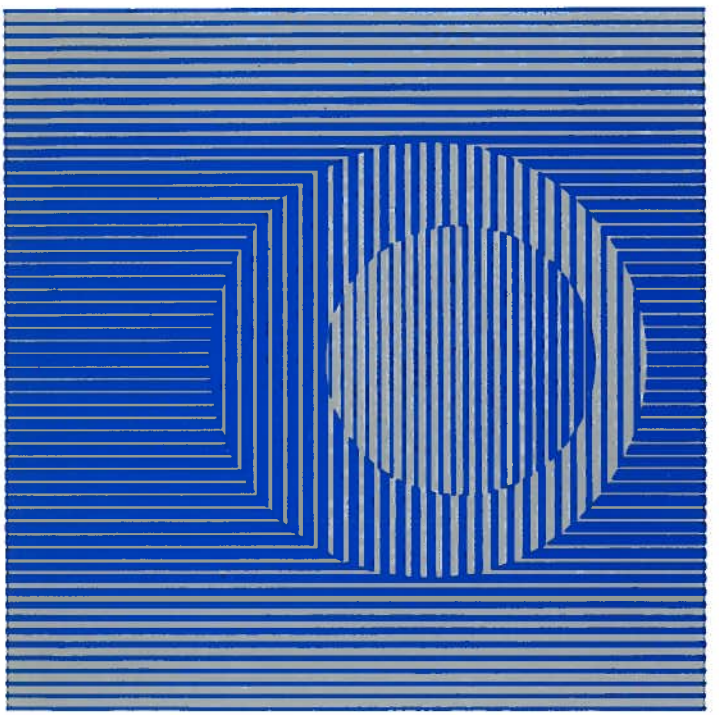
AAAI will provide each institution with one free 10-foot-square booth, room to describe the demonstration in the Exhibit Guide, and assistance with logistical arrangements. AAAI cannot provide assistance with some direct costs, including shipping equipment to the site, telephone lines, and housing. However, vendors may be able to assist with equipment needs. Last year, many hardware vendors donated equipment for the university demonstrations.

If you or your department are interested in participating, please contact:

Steven Taglio
AAAI
445 Burgess Drive
Menlo Park, CA 94025, U.S.A.
(415) 328-3123
AAAI-Office@sumex-aim.arpa

Deadline for the April issue is 15 February.

FINALLY...EXPERT SYSTEMS IN MAINSTREAM ENVIRONMENTS



Product features:

- English-Like Rules
- Two Inference Engines
- Incremental Compilation of Knowledge Base
- Extensive Explanation Facilities
- Knowledge Analyser
- Access to MProlog and External Procedures
- Custom Screens and User Interface
- Case Data Manager
- Command Language or Menu-Driven Interface
- Frame-Based Taxonomy
- Host Editor Interface
- Problem Decomposition
- Debugging Support
- Table Access
- Spelling Corrector
- KE Toolkit
- On-Line Help

TWAICE is available under IBM VM/CMS, IBM MVS/TSO, DEC Vax/VMS, DEC Vax/UNIX, Nixdorf Targon 35, Pyramid 90x, Sun, ISI, Apollo and Tektronix 4404-6.

The TWAICE package includes the full MProlog system and a complete support, education and maintenance program.

Logicware's expert system shell, Twaice™, lets you develop and deliver expert systems on IBM mainframes, DEC VAX and M68000 Workstations.

For more information on the TWAICE shell and/or our MProlog language, write or call;

Logicware, Inc.
Suite 200,
5915 Airport Rd.
Mississauga, Ontario
L4V 1T1
(416) 672-0300

Logicware, Inc.
70 Walnut St.
Wellesley,
MA 021181
(617) 237-2254

Logicware Inc.

TM–IBM, MVS, VM/CMS: International Business Machines, Inc./UNIX: Bell Labs/SUN: Sun Microsystems Inc./Apollo: Apollo Computer Corp./ISI: Integrated Solutions Inc./Tektronix: Tektronix Corporation/Targon: Nixdorf Computer AG/Pyramid: Pyramid Technologies/Twaice: Nixdorf Computer AG M68000: Motorolla Inc.

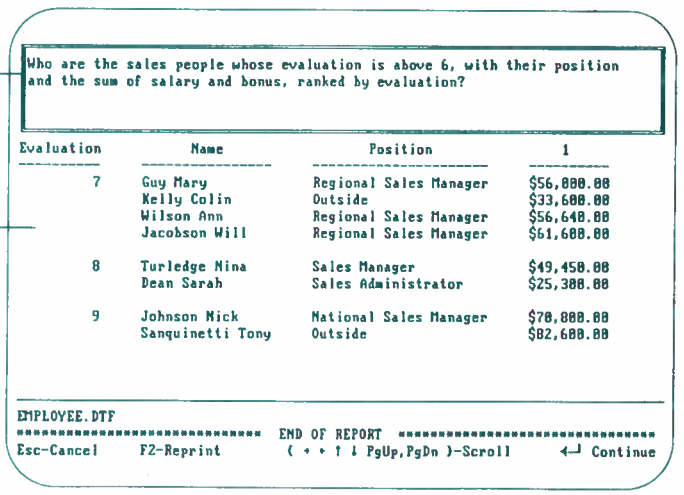
R–DEC VAX: Digital Equipment Corporation

How does Q&A's artificial intelligence work?

- Simple: user manages data with normal English sentences: "Show me the houses that cost between $100,000 and $200,000, listed by price and bedrooms."
- Powerful: use English sentences to search, sort, calculate and present information, prepare and print reports, fill in or change data.
- Software that learns from you; the more you use it the smarter it gets.
- Entirely menu driven with on-screen prompts.
- pfs-like interface, which 1.5 million users already know how to use.
- Context-sensitive, Concurrent and Custom help.
- Automatically prepares regular reports.
- Not copy protected.
- Speed data entry with programmability, macros, lookup tables, mass updates, data validation, initial values, custom Help screens and calculated fields.
- Flexible searches with 115 indexed fields, multiple keywords and 21 search operators.
- 40 levels of sorting
- 16 million records per database file.
- As fast as dBASE III.
- What you see is what you get...immediately.
- Integrated spelling checker of 60,000 words, user expandable.
- Line drawings; great for organization charts.
- Documents up to 14" wide.
- Link documents for unlimited length.
- Automated mail merge, print envelopes.
- Laser and 40 other popular printers.

Type a request for information in regular English.

Q&A reads and understands the request, then retrieves the information and automatically displays a report!



Q&A makes data-base access in artificial intelligence more efficient than ever before.
Q&A is available from Applied AI Systems, Inc for $495 + tax + S&H.
For a list of other software handled by Applied AI Systems, Inc see advertisement inside magazine.

Q&A is a registered trademark of Symantec Corporation.

Applied AI Systems, Inc.
P.O. Box 13550
Kanata, Ontario
Canada K2K 1X6
Telephone: (613) 592-0084