ARTIFICIAL INTELLIGENCE IN WESTERN EUROPE

J. Pitrat and E. Sandewall (session co-chairmen),
W. Bibel, G. Huet, H.-H. Nagel and M. Somalvico

INTRODUCTION

Artificial intelligence research in Britain is generally well known to the American audience. Research in continental Western Europe is less well known, for several reasons:

- it started later
- barriers of language and habit decrease the exchange of information through newsletters and at conferences
- for research on natural language understanding systems, which is a significant part of A.I., most researchers prefer to develop experimental systems in their own language. This makes it hard to publish the results in a foreign language
- there are some differences in research approaches. In particular, resolution theorem-proving has a relatively stronger position in Europe than in the U.S.

The intention of this panel is to present an overview of the research that is being done in the field in Western Europe.

The present paper for the proceedings is a concatenation of surveys for each participating country. Hopefully the panel discussion at the conference will be used partly for presenting the surveys, and partly for comparing the experience of these countries.

The panel contains participants from four countries: France, Western Germany, Italy, and Sweden. It has not been possible to include all countries and all projects in each country, but we do believe that the survey is representative for the significant research projects in continental Western Europe, and that it covers most of them.

The contributors from the different countries have selected different focus and style for their surveys. Uniformity in these respects was considered to be neither desirable nor possible to achieve. Credit is given to the authors at the end of the paper.

FRANCE

Our account of A.I. research in France covers work at the following major centers:

- University of Marseille-Luminy (director: A. Colmerauer)
- University of Paris 8-Vincennes (director: P. Greussay)
- Institut de Recherche d'Informatique et d'Automatique (IRIA) at Versailles outside Paris (directors: G. Huet and G. Kahn)
- University Pierre et Marie Curie in Paris (directors: J. Pitrat and J.-C. Simon)

- 1. University of Marseille-Luminy

In 1971, A. Colmerauer formed an Artificial Intelligence group at the campus of Luminy of the University of Aix-Marseille. The main area of research is programming in first-order logic. A programming language based on first order logic, PROLOG, has been designed and implemented. It has been distributed to several A.I. centers, and the group in Luminy is maintaining and improving it.

A PROLOG instruction is basically a first-order Horn clause. Certain predicates are evaluable, and the search is guided by special marks between literals. The PROLOG interpreter is based on an efficient linear resolution theorem prover.

Various applications of the PROLOG system are under development in Luminy, in particular:

a) A natural-language understanding system has been implemented for a substantial subset of the French language. The meaning of sentences is expressed using typed lambda calculus terms, which are analysed by so-called metamorphosis grammars, coded as PROLOG programs.

b) A symbolic manipulation system is under development. It is capable of doing integration, trigonometry and computation of limits in interactive mode, on expressions with several unknowns. The whole system is written in PROLOG.

- 2. University of Paris 8-Vincennes

The artificial intelligence group here has designed a version of LISP, called VLISP, which has several attractive features. It is based on very powerful control structures, most notably the concept of filters, which could be characterized as pattern-matching-controlled continuations. VLISP has been implemented on several computers. The implementation effort has been focused on efficient implementations on mini and micro-computers.

A new formalism for modeling various deductive processes, called CANON, is under development. It is based on the concept of meta-pattern-matching.

Among the applications developed in Vincennes, let us quote a programmer's assistant system for understanding LISP programs written by a novice programmer. The system is able to detect and correct various syntactic bugs, and a few semantic ones.

- 3. IRIA

In 1976, G. Huet completed his thèse d'Etat on unification in languages of order 1, 2, ..., ω, which provides a general framework for the study of pattern-matching problems. This work has applications in automating program transformations expressed as schema rewriting rules in a higher-order language. More recently, he worked on theoretical properties of simplification systems which are useful for the efficient mechanization of equational theories.

Concurrently, G. Huet and G. Kahn direct a project concerned with the conception and implementation of a programmer's assistant. This system, called MENTOR, is language independent, and based on a general notion of abstract syntax. The programmer does all phases of programming under the supervision of the system, and communicates with MENTOR using a specialized programming language operating on the abstract syntax. Powerful pattern matching operations are provided. The system is currently operating on PASCAL programs, and is itself entirely written in PASCAL. It is used to maintain and improve itself. A high-level optimization package for PASCAL is under development, and a symbolic interpreter is under completion. Further developments will focus on enriching MENTOR's programming language with coroutine-like control structures.

- 4. University Pierre et Marie Curie, Paris VI

The two main groups working on Artificial Intelligence are directed by professors Simon and Pitrat. Simon is also working in pattern recognition, which is not the subject of this paper. There are also some other researchers working independently. I shall describe the work done in four main areas: learning, theorem proving, applications to computer science, and natural language understanding, without specifying to which group each researcher belongs.

A. Learning and inference

J. Pitrat has realized a program [1] which learns to play chess. One of the main problems in chess is to choose what moves are to be considered. The program learns to generate a set of plans for any position from games which are given to it as data. It tries to understand why a move has been successful. If it understands, it generalizes the characteristics which are necessary for the success. The program has learned effectively to find plans, but as it had difficulties to use these plans, a program was written [2] which does not learn, but uses plans efficiently.

C. Lemaitre [3] wrote a program simulating a robot learning to structure its world. In a complex world, a robot with a small memory learned to recognize remarkable places and to connect them. So, when it decided to go somewhere, it could move there quickly. A man learning to move in an unknown town has a similar problem to solve.

C. Roche's program [4] learned to create operators useful for solving pattern recognition problems. For instance, it received patterns which belonged to one of two sets: in one set, two points are close and in the other the two point are distant. Initially, it received the arithmetic operators: plus, times etc. It was able to find the Euclidean distance for separating the patterns in the two given sets.

O. Carrière [5] wrote a program which solves psychological tests used for measuring intelligence. Tests such as sequences of numbers, letters, dominoes to complete, matrices with an empty square, and others were solved. The first step is to find the relation between the elements of the known sequence for inferring the missing element.

M. Jouannaud [6] studied the automatic inference of LISP functions from a finite number of examples. The program can often infer a recursive function from one example. If it can't, it asks for new examples and builds it by steps. With Treuil and Guiho [7] he also wrote a program learning some arithmetic laws.

M. Kodratoff extended Sommer's results and wants to implement them in a program. Specifically, he increases the number of recurrence relations and proves the existence of the function when the primitive recursive relations are more complex.

B. Theorem proving

10 theses [8,9,10,11,12,13,14,15,16,17] were made in this area and others are on the way. I will not present all these works, but only give the general idea which guide them. I do not believe that theorem proving is an efficient method for synthesizing programs, understanding natural language or solving robot problems. So, the goal of these theorem proving programs was only to prove mathematical theorems. Mathematics will probably be one of the first areas where programs will be as good as humans. But it is necessary to examine how the mathematician is working. In some areas, there are generally few choices for a human being in a given situation. Only one or two possibilities are a priori worthwhile. But there is a great variety of situations. M Vivet is studying how a program can use a set of methods for solving mathematical problems.

Another idea is to use a graph for helping the program. Some part of the fomula may be coded in the graph. For instance D. Pastre [16] replaces relations by arcs of a graph. The rules modify the formulas and the graph. Many steps become obvious when we consider the graph. The formulas are shorter.

Another use of a graph is to have a convenient storage. M. Buthion [17] wrote a program solving compass and straight edge construction problems in plane geometry. Unlike Gelernter's system, the diagram is not used for verifying whether a property is true. But the diagram stores certain facts,e.g. that some points are on the same line, or on the same circle.

We think that it is important to study the mathematician. Once, as we were interviewing, a mathematician said that he never used the usual notation for his theory, except when he was reading or publishing a paper. He himself worked with a representation which was not very clear, but entirely different from those used in the books. In such cases, a program now solves problems completely differently from how the mathematician does it. It is then normal that the program's performance is not so good, and an interesting problem is to study the representations really used by the mathematician, not when he is publishing, but when he is proving theorems.

C. Applications to computer science.

A first approach is the realization of programs which process programs which for various reasons have already been written. Pitrat wrote for instance a general game playing program [18]. The rules of a specific game were given as two separate programs written in a special language: one of them generated legal moves, and the other checked whether a given position was a winning one for either player. If the general program is to play well, it must be able to analyze the rules of a game for extracting useful information, particularly what conditions are necessary if a move is to be legal.

B. Falvigny [19] wrote a program for finding anomalies in a program; an example of an anomaly is the existence of an instruction such that, if it is executed, the program loops indefinitely. An anomaly is not always an error, but it indicates that, somewhere in the program, something is probably not correct.

A. Adam and J.P. Laurent [20] are writing a program to correct a student's program. This program finds which instructions are false and which instructions could be placed instead. For this, it compares the statement of the problem with the program given by the student.

Another application is program synthesis. Usually, we try to define the problem in a descriptive language. For instance the program of A. Adam [21] generates programs which check whether a graph has some property; the program of P. Gloess [22] generates programs translating a formula written in some notation to the equivalent formula written in another notation. In that case, the program receives the formal definition of both notations. F. Garijo's program [23] generates a program computing a function defined by a set of formulas. M. Masson's program [24] generates programs checking whether a tree has a given property.

J.L. Laurière's program [25] receives as data the descriptive statement of a combinatorial problem (with or without optimization). The data of a particular problem is given with the statement. The program first finds a solution, then, if optimization is needed, it finds the optimal solution and proves that this solution is optimal.

The method used by the program changes with the data. If the data includes certain values, the method used by the program works only because of these special values. For practical problems, its results were sometimes as good as those of programs which were written for this particular problem.

D. Understanding natural language

We want to understand how a program can understand. One method for finding this is to write a program understanding a story, then answering questions on this story. G. Sabah [26] and C. Loyo Varela [27] give the program a story in French. The program's goal is to find the motivation of the individuals who acts in this story. It uses pragmatic rules to find their motives.

Other programs answer questions in French on a blocks world (F. Rousselot [28]), in Spanish on the same domain (M.F. Verdejo [29]), in Arabic on the constitution of Lebanon (B. Wakim).

Another method for the same goal is to write a program solving problems given in a natural language: electricity exercises in French (M. Rousseau [30]) or mechanics exercises in Arabic(M. Gheith). The last method is to write a program which paraphrases. Pitrat is writing such a program for the French language: annotations of chess games.

We are mainly interested in the representation of the meaning of the sentences and to find what information is necessary for understanding. Pragmatics have a great importance in our approach, but it is necessary to give the program a huge amount of information.

Working in another direction, but independently, D. Coulon and D. Kayser [31] have written a program learning to identify answers to a given question expressed in a natural language. From a sample of such questions, the program builds rules, and with these rules, it can classify the new answers. The rules are built using some elementary operators. This program is useful for teaching machines.

GERMANY, FEDERAL REPUBLIC OF

For quite some years there has been research in Germany in areas which are usually considered part of artificial intelligence; but it was not until 1975 that the scientists working in these areas came together to identify themselves as a group with common scientific interests. Since then quite a bit has achieved.

- The group officially became a special interest group ("Fachgruppe") of the German association for informatics (Gesellschaft für Informatik - GI) and, like other areas in informatics, it is represented within the GI by a committee ("Fachausschuss") whose members are scientists working in the field of A.I. and of pattern recognition. Those from A.I. are W. Bibel, J. Laubsch, H.-H. Nagel (chairman), and G. Veenker.

- A German A.I.-newsletter with similar intentions like the SIGART- or AISB-newsletter now appears regularly on a quarterly basis and is sent to about 200 addresses (currently edited by W. Bibel).

- In this relatively short time of somewhat more than two years five meetings or workshops exclusively devoted to A.I.-research have been organized, not counting contributions to national conferences with A.I.-sections.

- An international conference on A.I. will take place in the third week of July 1978 in Hamburg organized by the GI in co-operation with the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB) from Britain. Local arrangements will be the responsibility of H.-H. Nagel from Hamburg. Dr. Derek Sleeman from the computer based learning project, University of Leeds will act as program chairman.

According to that relatively short history it is not surprising that there is no real center of A.I.-research nor any really big A.I.-project in Germany. The activities are scattered over many places, with some clustering taking place at the universities of Hamburg and Karlsruhe.

The scientific activities mainly concentrate on two areas of research: processing of naturally represented information (written natural language and vision) and automatic deduction. The common future perspectives for both are probably knowledgeable, human oriented programming systems. In the following we subdivide these areas and list names of people working in the respective subareas and their affiliation (which is normally "Institut für Informatik, Universität <name of city>", if not stated otherwise). This list is based on the necessarily incomplete information which is at the authors' disposal and therefore cannot be exhaustive.

1. Processing of naturally represented information
   a) natural language processing, syntax

semantics, translation
b) scene analysis
c) cognitive systems; formal representation of knowledge, semantic nets
d) (linguistic) question answering systems
e) (procedural) problem solving

There is a large image analysis group at the Forschungsinstitut für Informationsverarbeitung und Mustererkennung (FIM) at Karlsruhe working with Kazmierczak. Their work has a large overlap with scene analysis (b, c): Bohner, Röcker, Schärf, Stiess.

A number of people working in this area are in Hamburg: Bertelsmeier, Kemen, Nagel, Neumann, Radig (b, c); Schefe, Wahlster, Wittig (a, c, d, e); Boley (c), and several others. At Stuttgart there are Laubsch (a, b, c), Hanataka (b), Krause (a, c). At TU München work Janas, Ripken, Schwind (a, c); at Saarbrücken Weber (a). Fischer (c) at the Bildungstechnologisches Zentrum, Darmstadt, Görz (a, c) at Erlangen. Brecht and others (c, e) work at the Institut für Deutsche Sprache, Mannheim and Bonn. Klaczko (c) at Frankfurt. Scene analysis (b) is pursued by Foith at the Institut für Informationsverarbeitung in Technik und Biologie at Karlsruhe.

2. Automatic deduction

a) theorem proving in logical systems
b) program synthesis/analysis/verification
c) deductive question answering systems

The strongest group working in this area has its center in Karlsruhe: Deussen, Siekmann, Wrightson (a); Raulefs (b) and others; at Bonn work Darlington (a, b, c); Förning, Veenker (a); at Aachen Richter (a); at TU München work Bibel, Schreiber (a, b); v. Henke (b) works at the GMD, St. Augustin; Winterstein (a) at Kaiserslautern; at TU Berlin work Bergmann, Noll (a); Schneider, Konrad (e). Kudlek (a) works at Hamburg.

3. Chess, etc.

There are several game playing systems. In a tournament held in 1975 the most successful German chess playing system was one developed by Zagler and others at the TU München.

4. Miscellaneous

A surprisingly efficient system for solving differential equations has been developed by Schmidt, Bonn. - A formula manipulation system was designed by Steinbrüggen, TU München. - AI methods are involved in a system for solving chemical problems which is being developed by Ugi and Brandt, Organ. chem. Labor, TU München.

In the area of automatic deduction a lot of work has been invested in the theoretical foundations (such as development and comparison of various complete theorem proving methods especially those based on natural deduction system, as well as semantics and representation of programs). But now, as software for string processing at the computing centers in Germany becomes more comfortable, there is a tendency towards building efficient, specialized, practicable systems rather than devloping further (complete) theoretical methods. But obviously most groups are still too small to provide the manpower necessary for the design and implementation of really powerful system. However, the perspectives for the near future seem to be not too bad in that respect.

Work in speech understanding is still pursued as part of pattern recognition. In contrast to this, people working in image analysis seem to be more aware of AI implications for their work. Although the larger image analysis projects at FIM/Karlsruhe and the Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR) Oberpfaffenhofen cannot yet be considered as AI projects, it would not be surprising to see major ventures into this area in the future.

ITALY

The survey of A.I. research in Italy has unfortunately been delayed in the mail.

SWEDEN

Artificial Intelligence research in Sweden first developed in the Computer Science departments of some universities. In the last few years, some A.I. has been done in Linguistics and Psychology departments as well.

Uppsala and Linkoeping Universities. The Informatics Laboratory ("Datalogilaboratoriet") at Uppsala University was formed in 1970. In 1975 the major part of the group moved to Linkoeping University, while a minor part remained in Uppsala. The two groups have therefore similar profiles and will here be described together.

The group in Uppsala was formed by a few people who had worked on typical A.I. problems such as data-base structures for question-answering systems, heuristic search, and high-level languages. However, in specifying the area of research for the new group, we chose to invent the term (technology for) small data bases, rather than artificial intelligence.

In practice, "small data bases" has a considerable overlap with artificial intelligence. The data base of a Q.A. system is a typical small data base. In general, small data base technology is concerned with data bases where the conceptual complexity of the data and/or the operations that are to be performed on them are the major problems, in contradistinction to conventional data bases where the complexity of information is small, and the volume of data and economy of operation are the dominating problems. LISP-like languages, and very-high-level languages based on LISP, are software tools for small data bases. Natural language would probably for many purposes be an ideal means of communication with a small data base. The term small data bases is perhaps equivalent to "knowledge-based systems" that is now often used internationally.

The significant difference between "small data bases" and "artificial intelligence" is that the former term implies a stronger emphasis on software tools (such as programming languages) and on applications, and that it avoids the science-fiction-like overtones which have been both an attraction and a detriment for A.I. In general, we believe that our term has been useful. However, it could clearly not be used in a country where "big" is a positively loaded word, and "small" is negatively loaded.

The following are some of the major projects that have been concluded or nearly concluded in these groups:

- Implementation of an INTERLISP system for IBM 360 and 370 series computers (Jaak Urmi et al, Uppsala)
- Design of special hardware for a LISP system (Jaak Urmi, Linkoeping)
- Work on data bases for natural language understanding system, based on the idea that predicate calculus or an extension thereof should be used as a specification language for both the data base and the storage and retrieval operations. Predicate calculus was not used for the representation of the NL information, but rather as the notation wherein the representation of various semantic constructs was expressed. In a sense it was used as a programming language, although one which was not directly implemented, but instead one which served as the abstract program from which a concrete program in a concrete programming language (such as LISP) was written.
(Erik Sandewall, Uppsala)

- Work on comprehension of natural language texts that describe processes, through implementation of a system which accepts descriptions of patience (solitaire) games in natural Swedish, analyzes the text, and performs the game. The system has been designed for real natural language texts, taken from books from a bookstore, rather than for constructed examples. The domain is interesting because the games are processes that involve conditional branches as well as loops. Since each game description is really an algorithm, the sublanguage used in describing games can be viewed as a natural programming language. The work therefore has a potential interst for other branches of computer science as well.
(Mats Cedvall, Uppsala)

- Implementation of a compiler for ATN grammars. This work was done in the early stages of the patience game project, as a development of tools, achieved speed-up with a factor of ten in the IBM 370 INTERLISP.
(Mats Cedvall, Uppsala)

The following projects are less directly related to artificial intelligence, but are natural consequences of the small data base profile:

- Work on partial evaluation as a program development technique. This work has been done iteratively by designing a partial evaluator, using it for a number of applications, and revising the partial evaluator. The recent version of the system has been able to successfully partially-evaluate LISP code that had been written by an expert LISP programmer without previous knowledge that his program was going to be used in the experiment.
(Anders Haraldsson and others, Uppsala and Linkoeping)

- Compiler for partial evaluator. In conjunction with the work on partial evaluation, there has been work on a partial-evaluator-compiler, which transforms a general-purpose program to a corresponding program generator, and on methods for proving that a proposed use of these programs is correct.
(Erik Sandewall and others, Uppsala)

- A method for formal definition of the semantics of programming languages, demonstrated through a definition of Simula. The definition is an idealized compiler, and is significant because it is both effective (it exists as a LISP program, and has been used to compile and run a systematically designed test batch of Simula

programs, which is significant for debugging the definition), and legible (the report describes how a number of real questions about Simula semantics are answered by referring to the formal definition).
(Mats Nordström, Uppsala)

- An auxiliary system for design of man-machine dialogues. The system represents the dialogue as a network, where each node represents an interaction, and is associated with a typeout from the system, a syntax for possible user responses, indications for which node to go to next, etc. The system, called IDECS, contains an interpreter for the network with advanced facilities such as backtracking and undoing; an interactive editor on the network, and of course facilities for entry and presentation of the conversation network. The system was designed in response to a practical need, and enables one to let an end user try his hand at the proposed dialogues, find out what details have to be changed, make the change at the spot, and let the final user try again. This system was developed in a cooperation project with the Swedish Bureau of Statistics and has been used in several application projects. (Sture Hägglund and others, Uppsala and Linkoeping)

All of the last three projects relied very heavily on the use of LISP as an implementation language, and probably could not have been completed if a conventional language had been used. They indicate that interactive programming technology in LISP is a significant spin-off result of A.I. research.

The projects that have been described represent a sample which characterizes the profile of the activities. A few additional projects should also have been described but have had to be omitted because of space restrictions.

Stockholm University. Sten-Åke Tärnlund and a few co-workers have worked since around 1970 with resolution theorem-proving. After initial work on implementation and experimental use of a theorem-prover on a mini-computer, the group has recently worked in the area of "using predicate calculus as a programming language". This work has included the implementation of several well-known algorithms in this language, an analysis of the properties of the language, and correctness proofs of algorithms expressed in the language.

Research Institute of National Defense, Stockholm. A group headed by Jacob Palme undertook a project in the period 1968-73 to implement a fairly efficient question-answering system for a simplified and somewhat un-natural English. The project resulted in a general-purpose parser system and a "semantic net" type data base that was quite sophisticated considering its efficient implementation (in PL/360). There was also an experimental implementation with Esperanto as the input language.

The projects that have been described so far have been performed in a computer science environment. There are also a number of projects which were performed in other milieus. In 1975 a common interest group comprising people of these different backgrounds was formed.

Stockholm University, department of Linguistics. Carl-Wilhelm Welin has studied the problem of "understanding" natural Swedish through implementation of a system which accepts weather bulletins. These bulletins, published from the central weather bureau, are in Sweden customarily written in a sublanguage, which contains a large number of stereotypes and which at least initially appears to be sufficiently limited to be ideal as a test environment for a system endowed with non-trivial understanding.

Stockholm University, department of Psychology. Anders Eriksson has done work on problem solving methods in the context of the 8-game (reduced 15-game) Protocols of human subjects solving 8-puzzles were collected, and used as the basis for a model of problem-solving. The derived model uses production schemas and has a GPS-like flavor. The work was presented as a thesis. Additional A.I.-oriented projects have been started at the department.

Uppsala University, Computing Center. Anna-Lena Sågvall Hein directs the development of tools for natural language processing, oriented mostly towards the needs of the language departments of the university and the computing center's speciality, medical data processing. A system for morphological analysis of Russian was the first project, and has been operational since 1973. A current project with a distinct A.I. flavor is an experimental system for understanding authentic medical reports on medial column fractures in the hip, written in Swedish. The system translates each sentence into a domain-specific internal representation. Work on detection of conflicts in the knowledge base comes next. - Other projects in the group are a chart-analysis parser according to Kay, and a system for inflectional analysis of Finnish.

REFERENCES - FRENCH CONTRIBUTION

[1] J. PITRAT: A program for learning to play chess. Pattern recognition and artificial intelligence. Edited by Chen, Academic Press, 1976, p. 399-419.

[2] J. PITRAT: A chess program which uses plans. Artificial intelligence.

[3] C. LEMAITRE: Problèmes de planification et apprentissage dans le cas d'un programme de simulation de robot. Thèse de 3° cycle, Paris VI, 29-5-74.

[4] C. ROCHE: Generation of recognition operators by stochastic and heuristic learning. Computer oriented learning processes. Edited by J.C. Simon. Nato Advanced Study Institute series. Noordhoff 1976, p. 237-255.

[5] O. CARRIERE: Réalisation d'un programme heuristique qui résout des tests psychologiques de mesure du facteur G. Thèse de 3° Cycle, Paris VI, 19-6-73.

[6] J.P. JOUANNAUD, D. GUIHO, J.C. SIMON: Inference of functions with an interactive system. Machine Intelligence 9.

[7] J.P. TREUIL, G. GUIHO, J.P. JOUANNAUD: Une méthode d'apprentissage de concepts. Panorama de la nouveauté informatique en France. AFCET, 1974, Tome 1, p. 77-89.

[8] J. PITRAT: Une programme de démonstration de théorèmes. Dunod, 1970.

[9] J.L. DELHAYE: Datal: Un program de démonstration automatique de théorèmes. Thèse de 3° Cycle, Paris VI, 9-4-70.

[10] J.P. LAURENT: A program that computes limits using heuristics to evaluate the indeterminate forms. Artificial intelligence, Vol. 4, N° 2, Summer 1973, p. 69-94.

[11] M. VIVET: Un programme qui vérifie des identités en utilisant le raisonnement par récurrence. Thèse de 3° Cycle, Paris VI, 29-22-73.

[12] R. DALLARD: Présentation d'un programme de démonstration de théorèmes d'arithmétique. Thèse de 3° Cycle, Paris VI, 24-4-74.

[13] M. GRANDBASTIEN: Un programme qui résout formellement des équations trigonométriques par des procédés heuristiques. Thèse de 3° Cycle, Paris VI, 14-6-74.

[14] A DURAND: Un programme de démonstration d'exercices d'algèbre. Thèse de 3° Cycle, Paris VI, 13-6-75.

[15] A. HERTZ: Programmes de démonstration de théorèmes formulables en logique des prédicats du 1er ordre avec égalité. Thèse de 3° Cycle, Paris VI, 2-12-75.

[16] D. PASTRE: Démonstration automatique de théorèmes en théorie des ensembles. Thèse de 3° Cycle, Paris VI, 22-10-76.

[17] M. BUTHION: Un programme qui résout formellement des problèmes de construction géométrique. Thèse de 3° Cycle, Paris VI, 13-6-75.

[18] J. PITRAT: A general game playing program. Artificial intelligence and heuristic programming. Edited by Findler and Meltzer, Edinburgh University Press 1971, p. 125-155.

[19] B. FLAVIGNY: Un programme détecteur d'erreurs de logique dans les programmes. Revue Française d'Automatique d'Informatique et de Recherche Opérationelle. Informatique, Juillet 1975.

[20] A. ADAM, J.P. LAURENT: Mise en évidence de relations d'ordre dynamique entre les instructions d'un programme. Panorama de la Nouveauté Informatique en France, AFCET. Tome 1, p. 169-179.

[21] A. ADAM: Gadget: un programme de génération automatique de programmes sur les graphes et les ensembles. Thèse de 3° Cycle, Paris VI, 2-4-73.

[22] P. GLOESS: Gener: un générateur de programmes. Thèse de 3° Cycle, Paris VI, 19-10-76.

[23] F. GARIJO: Sudi: Un programme qui vérifie des problèmes et écrit des programmes optimisés. Panorama de la Nouveauté Informatique en France. Tome 1, p. 67-75.

[24] M. MASSON: Un programme qui génère des programmes. Panorama de la Nouveauté Informatique en France, AFCET 1976, Tome 1, p. 381-390.

[25] J.L. LAURIERE: Un langage et un programme pour énoncer et résoudre des problèmes combinatoires. Thèse d'Etat, Paris VI, 21-5-76.

[26] G. SABAH: Un système de compréhension du langage naturel. Panorama de la Nouvearuté Informatique en France, AFCET 1976, Tome 1, p 269-281.

[27] M.C. LOYO VARELA: Une expérience informatique de compréhension du langage naturel. Thèse de 3° Cycle, Paris VI, 8-4-76.

[28] F. ROUSSELOT: Simulation d'un robot qui comprend et exécute des ordres donnés en français. Thèse de 3° Cycle, Paris VI, 13-6-75.

[29] M.F. VERDEJO: Etude du langage naturel. Simulation d'un robot capable de mener un dialogue en espagnol. Thèse de 3° Cycle, Paris VI, 30-10-75.

[30] M. ROUSSEAU: Résolution automatique d'exercices d'électricité posés en français. Revue française d'automatique d'informatique et de Recherche Opérationelle. Informatique, Juillet 1975.

[31] D. COULON, D. KAYSER: Grammatical inference on natural language samples in the framework of C.A.I. Computer oriented learning processes. Edited by Simon, Nato advanced study institute series - Noordhoff 1976, p. 193-205.

For the other contributions, research reports can be obtained by writing to the respective projects, or to the panelists for that country.

PANELISTS

Affiliations of panelists:

J. Pitrat	University Pierre et Marie Curie (Paris VI) (France)
E. Sandewall	Linkoeping University, (Sweden)
W. Bibel	Technische Universitaet Muenchen (Munich, Federal Republic of Germany)
G. Huet	Institut de Recherche d'Informatique et d'Automatique (France)
H.-H. Nagel	Institut fuer Informatik, Universitaet Hamburg (Federal Republic of Germany)
M. Somalvico	Istituto de Elettrotecnica ed Elletronica, Politecnico de Milano (Italy)

Authors of the sections of this paper:

Introduction and post-editing	Sandewall
France	
Luminy, Vincennes, IRIA	Huet
Paris VI	Pitrat
Germany	Bibel and Nagel jointly
Italy	-
Sweden	Sandewall

(Continued →)

ARTIFICIAL INTELLIGENCE IN WESTERN EUROPE

J.Pitrat and E. Sandewall (session co-chairmen),
W.Bibel, G.Huet, H.H. Nagel and M. Somalvico

Addendum

ITALY

Our account of A.I. research in Italy covers work at the following major centers :

- Milan Polytechnic Artificial Intelligence Project (MP-AI Project) of the Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Milan (Director: M. Somalvico).
- Non-Numerical Information Processing Section of the Istituto di Elaborazione dell'Informazione del CNR , Pisa (Director : U. Montanari).
- Speech Understanding Group of the Centro per l'Elaborazione Numerale dei Segnali del CNR, Politecnico di Torino, Turin (director: R. De Mori).
- Artificial Intelligence Group of the Laboratorio per la Dinamica dei Sistemi e per l'Elettronica Biologica del CNR, Padua (Director: E. Pagello).

- 1. Milan.

In November 1971 Marco Somalvico has formed the Milan Polytechnic Artificial Intelligence Project (MP-AI Project) as a research project at the Laboratorio di Calcolatori of the Istituto di Elettrotecnica ed Elettronica of the Politecnico di Milano.

Its research activity has been based on the assumption that Artificial Intelligence is that discipline, within Computer Science, which studies the theoretical foundations, the main methodologies,the design and the construction of programs which allow the computer to operate mechanisms and to perform activities that are usually considered as exclusive of human intelligence.

We can examine, in the sequel, the main research directions which have been followed at the MP-AI Project [1].

A. Theory of Problem Solving.
The process of representing a problem involves necessarily a passage between two different worlds, namely the world of the intuitive notion that the man has of the problem, and the world of the formal and precise description of the problem, that will be given to the computer.

This passage is achieved by means of an appropriate selection of only one part of the information which is pertinent to the intuitive knowledge of the problem; the selected information is then arranged in a structured form, thus yielding the representation of the problem.

However, even the most structured and rich representation,which might be obtained within this passage, will always present a difference from the unformal knowledge about the problem.

This distance, which reminds the similar gap between physical phenomenon and physical law, can only be narrowed,but can never be wholly deleted.

The previous considerations have been the basic motivations for the development of a "theory of problem-solving" devoted to the understanding of the different aspects of the problem-solving process, and centered around the focal point of problem representation.

This theory of problem-solving, which is being developed at the MP-AI Project, is intended to achieve the following goals:

- formalization of problem representation methods [2,5,8];
- formalization of solution search techniques [3];
- formalization and selection of "good" problem representations [6];
- automatic evaluation and use of heuristic information [4 , 10];
- generalization and operation of learning processes [7];
- structured organization of a problem in subproblems as a basis of automatic programming [9];
- design of a general structure for a problem solver [11].

The future research directions will be the theoretically guided design and experimentation of a problem solver based on the obtained results.

B. Languages for Artificial Intelligence. On the UNIVAC 1108 computer of the MP-AI Project, have been implemented, and experimented, the following interpreters: LISP, MICROPLANNER, PROLOG, MAGMALISP, REDUCE.

These interpreters have been utilized as tools for designing AI systems in problem solving, robot plan formation, natural language understanding, automatic deduction.

In particular, the characteristics and limitations of MICROPLANNER and PROLOG have been examined in detail.

Studies and a comparison between the procedural embedding and the axiomatic approaches to knowledge representation have been carried on [12, 13].

C. Robot Plan Formation. The research activity which has been performed, has been devoted to utilize MICROPLANNER and FIRST ORDER PREDICATE CALCULUS as representation languages devoted to in plan formation [14,15,16,17].

Also, an algebraic theory of plan formation has been developed and two systems, written in LISP, have been developed [18].

In particular, the following results have been obtained:

- Formalization of environment respresentation;
- Formalization of robot manipulating activities (plans, actions);
- Development of a system for checking the correctness of plans;
- Development of a system for the optimization of plans;
- Development of a system for plan formation, of means-end type;
- Development of a system for plan formation, based on partial ordering of actions.

D. Role of Classical and Non-Classical Logic. The PROLOG language has been experimented in application to robotics.

The limitations on the present day relationals programming languages, have been considered, mainly in the adoption of search strategies.

When FIRST ORDER PREDICATE LOGIC is utilized as a programming language, it is essential not only the possibility of proving a theorem but also the technique followed within the proof.

The logic programming language has been arranged in two levels, both using predicates.

Thus, the program becomes a couple, i.e. representation program and control program.

Interesting research results, have been obtained, mainly in the semantic domain of robotics, and in using temporal features in logic [19, 20, 21, 22].

E. Natural Language Understanding. A quite large system, the DONAU (Domain Oriented NAtural language Understanding) system has been developed for understanding Italian.

A first version of DONAU has been developed and tested, with success, in the semantic domain of robotics.

DONAU is written in LISP and MICROPLANNER and is based on the previous work done by the Artificial Intelligence Group of the University of Grenoble, France (director G. Veillon, with J.C. Latombe, A. Lux) in developing the PIAF system for the syntactic, morphologic, and phonetic analysis of French and Italian.

A joint research program on PIAF and DONAU has been developed and financially supported, between Milan and Grenoble.

The semantic analysis of the Italian input sentences is based on a discrimination net.

The extraction of operative information and the control of legality are part of the modular architecture of DONAU [23,24,25,26].

Also first studies in the semantic domain of data-bases have been developed [27,28].

F. Robotics. In the Laboratory of Robotics of the MP-AI Project, a two arms, two light sensors, six force sensors, six degrees of freedom robot, the SUPERSIGMA robot, is now operating.

Its main tasks are devoted to the assembly of complex mechanical systems.

The SUPERSIGMA architecture is composed of eight NATIONAL SC/MP microcomputers, one LABEN 70 minicomputer; it will be eventually connected to the UNIVAC 1108 computer.

The research activity in this field has been devoted to the following goals :

- emergency recovery problem: the problem solving techniques have been applied to automatically solve the problem of recovering from the interruption of the execution of the normal assembly algorithm, when defective component parts are encountered [29, 30];
- assembly language: the MAL (Multipurpose Assembly Language) language has been developed for enabling the mechanical designer to describe the assembly algorithm; the MAL translator can execute either the source program (the deterministic assembly algorithm) or the output of a simple plan formation program which receives, as input, the SUPERSIGMA task representation;
- multimicrocomputer architecture: the use of eight microcomputer enables the conception of the whole SUPERSIGMA robot as a special computer (and <u>not</u> as a computer connected to the mechanical structure).
  This special computer has a set of machine instructions which is composed by the classical machine instructions of the LABEN 70 minicomputer augmented of new machine instructions (e.g., MOVE Δx_1, GRASP, etc.).
  Such new machine instructions correspond to the problems whose solution algorithms are programmed in each microcomputer [31, 32, 33].
- manipulating of concurrent tasks: both the SUPERSIGMA architecture (with the parallel activity of the eight microcomputers) and the MAL design (with the execution of concurrent processes controlled by a circular precedence scheduler) enable the execution of concurrent manipulating activities; the efficiency increase in the execution of task in the goal of the present research activity [34];
- social and economic implication of robotics: the organization of work in the factory assembly lines is going to be greatly influenced by the advent of robots; these research problems are presently investigated [35, 36].

- 2. <u>PISA</u> .

This research group has been formed since 1971 by Ugo Montanari and has developed a great deal of theorical and experimental reserach in A.I.

A. <u>Problem Solving.</u>
The chromosome matching has been considered as a target for investigating the efficiency of the Hart-Nilsson-Raphael algorithm; new properties of this algorithm have been introduced.

The foundations of dynamic programming and the relationships with heuristic searc methods have been investigated by showing that the AND/OR graph is a general model of dynamic programming and that each dynamic programming problem can be reduced to the problem of finding an optimal solution tree in an AND/OR graph [38,39,40].

In particular, a heuristic search algorithm has been presented, which combines the advantages of dynamic programming, i.e.,the identification of common subproblems, with the advantages of branch-and-bound, i.e.,the use of heuristic estimate to speed up the search [41].

The study of deriving, from a decision table, a decision tree which is optimal in same sense, has been recently investigated.

In particular, the minimization of the average execution time has been obtained by utilizing heuristic information [42].

The generalization of AND/OR graphs has been investigated, and the relationship with grammars has been introduced [43].

B. <u>Theorem Proving.</u>
In the line of Kowalsky, the logic formalisms have been considered as programming languages,the proof checkers as interpreters.

Thus a powerful logic formalism and an efficient proof checker can allow to prove manually theorems whose automatic proof would be otherwise very far from the state of the art.

A new type of efficient unification algorithm has been designed [44].

More recently this unification algorithm has been merged with Boyer and Moore's technique for storing shared structures in resolution based proof checkers [45].

C. <u>Languages for Artificial Intelligence</u>.
A new version of LISP, MAGMALISP has been developed by Montangero, Pacini, and Turrini in the last years.

The characteristic of its interpreter enables the availability of powerful primitives useful for developing, within MAGMA-LISP, non-deterministic languages.

One of such languages, ND-LISP, has been proposed and implemented.

Levi and Sirovich have developed a pattern matching language and a descrimination network for efficient pattern matching.

This tools are available on an IBM 370 computer, together with INTERLISP.

The application to automatic and interactive program verifiers is currently being investigated.

- 3. TURIN.

Research activity in Speech Understanding has been carried on since 1971 [46].

The use of syntactic methods for automatically interpreting speech pattern, has been investigated.

The utility of such method is sustained because most of the phonetic knowledge of a language is available in terms of descriptions of spectral patterns or linguistic events [47].

The problem of representing and using such knowledge in a digital computer has been studied.

The details of the implementation of a syntax controlled acoustic encoder of a speech understanding system (SUS) have been discussed.

Moreover, the learning capability in the SUS system has been developed

Present research activity is devoted to choose a semantic domain, e.g., robotics, in which the application of the SUS is practicable.

- 4. PADUA

Since 1974, Enrico Pagello is developing a new group devoted to A.I. researches.

LISP, PROLOG, and REDUCE constitute the available languages for experimental activity.

The main research work is devoted to theorical investigations on the role of logic in A.I.

In particular, the main research direction is constituted by the use of logic as a programming language for describing both the problem and the contral representation to a robot plan formation system [20, 21, 22].

REFERENCES - ITALIAN CONTRIBUTION

[1] M. SOMALVICO : The Milan Polytechnic Artificial Intelligence Project. Project Technical Report·1977. MEMO MP-AIM-68·(In Italian).

[2] A. SANGIOVANNI VINCENTELLI and M. SOMALVICO : Theoretical Foundations of State-Space Approach to Automatic Problem-Solving. (In Italian). Alta Frequenza. Vol. XLIV, N.3. March 1975.

[3] A. SANGIOVANNI VINCENTELLI and M. SOMALVICO : Problem-Solving Methods in Computer Aided Medical Diagnosis. Proceedings Twentyth International Scientific Conference on Electronics. Rome, Italy. March 1973.

[4] A. SANGIOVANNI VINCENTELLI and M. SOMALVICO: State-Space Approach in Problem-Solving Optimization. Proceedings Fifth IFIP Conference on Optimization Techniques. Springer Verlag, New York, Inc. Rome, Italy. May 1973.

[5] A. SANGIOVANNI VINCENTELLI and M. SOMALVICO: Theoretical Aspects of State-Space Approach to Problem-Solving. Proceedings Seventh International Congress on Cybernetics. Namur, Belgium. September 1973.

[6] D. MANDRIOLI, A. SANGIOVANNI VINCENTELLI, and M. SOMALVICO : Toward a Theory of Problem Solving. Marzollo, A. (ed.). Topics on Artificial Intelligence. Springer Verlag, New York, Inc. October 1973.

[7] D. MANDRIOLI, A. SANGIOVANNI VINCENTELLI, and M. SOMALVICO : A General Approach to Learning in Problem-Solving. Proceedings NATO Advanced Study Institute on Computer Oriented Learning Processes. Nordhoff International Puglishing, Leyden. Bonas, France. August 1974.

[8] G. GUIDA, D. MANDRIOLI, A. PACI, and M. SOMALVICO : A Formal Framework for an Unitary Approach to the Theory of Problem Solving. Proceedings IIASA Workshop on Artificial Intelligence: Question Answering Systems. Laxenburg Schloss, Austria. June 1975.

[9] G. GUIDA, D. MANDRIOLI, A. PACI, and M. SOMALVICO : An Integrated Theory of Problems as an Algebraic Base for Complexity Understanding and Automatic Problem Solving. Proceedings Seventh IFIP Conference on Optimization Techniques. Springer Verlag, New York, Inc. Nice, France. September 1975.

[10] G. GUIDA and M. SOMALVICO: Semantics in Problem Representation and Search. Information Processing Letters. Volume 5, Number 5. November 1976.

[11] G. GUIDA, D. MANDRIOLI, and M. SOMALVICO: An Integrated Model of Problem Solver. Information Sciences. Volume 12, Number 3. 1977.

[12] G. GINI and M. GINI: Cognitive Information Retrieval by Goal Oriented Languages. Proceedings IIASA Workshop on Artificial Intelligence: Question Answering Systems. Laxenburg Schloss, Austria. June 1975.

[13] G. GINI and M. GINI : Conniver Programs by Logical Point of View. Proceedings Fourth International Symposium on Mathematical Foundations of Computer Science. Springer Verlag, New York, Inc. Marianske Lazne, Czechoslovakia. August 1975.

[14] P. CORTI, G. GINI, M. GINI, and E. PAGELLO : Mathematical Logic and Goal-Oriented Languages in Robot Problem Solving. (In Italian). Alta Frequenza. Vol. XLV, N. 1. January 1976.

[15] P. CORTI, G. GINI, and M. GINI : Object Manipulation and Goal-Oriented Languages. (In Italian). Proceedings Thirteenth International Automation and Instrumentation Conference on Human Engineering. Milan, Italy. November 1974.

[16] G. GINI and M. GINI : Control of Intelligent Robots and Goal-Oriented Languages. Proceedings Conference on Applied Robotics. Cesky Krumlov, Czechoslovakia. May 1975. Also, The Industrial Robot. Volume 2. Number 2. June 1975.

[17] G. GINI and M. GINI: Programming a Robot by Goal Oriented Languages. Proceedings Third Workshop on Graphs, Systems, and Cybernetics. Bucharest, Roumania. August 1975.

[18] G. GUIDA and P. RIVERA: A Model for a Robot Planning System. MEMO MP-AIM-55.

[19] E. PAGELLO : Logic for Robots. (in Italian). Proceedings Thirteenth International Automation and Instrumentation Conference on Human Engineering. Milan, Italy. November 1974.

[20] E. PAGELLO : Logic and Interpreters. Proceedings IIASA Workshop on Artificial Intelligence: Question Answering Systems. Laxenburg Schloss, Austria. June 1975.

[21] M. COLOMBETTI and E. PAGELLO: Programs, Computation and Temporal Features. Proceedings Fifth International Symposium on Mathematical Foundations of Computer Science. Springer Verlag, New York, Inc. Gdansk, Poland. September 1976.

[22] M. COLOMBETTI and E. PAGELLO : A Logical Framework for Robot Programming. Proceedings Second CISM-IFTOMM Symposium on the Theory and Practice of Robots and Manipulators. Warsaw, Poland. September 1976.

[23] G. GINI and M. GINI: A Natural Oriented Language Based Question Answering System. Proceedings SSCS Workshop on Computer Models of Natural Language, Casuality, and Intention in Language Understanding. Castagnola, Switzerland. August 1975.

[24] M. BERNORIO, M. BERTONI, A. DABBENE, and M. SOMALVICO : Man-Machine Interaction in Quasi Natural Language. (In Italian). Proceedings Annual AICA Conference. Milan, Italy. October 1976.

[25] M. BERNORIO, M. BERTONI, A. DABBENE, and M. SOMALVICO: A Domain-Oriented Natural Language Understanding System for Man-Machine Interaction with Dynamic Data-Bases. Proceedings IIASA Workshop on Natural Language for Interaction with Data Bases. Laxenburg Schloss, Austria. January 1977.

[26] M. BERNORIO, M. BERTONI, A. DABBENE, and M. SOMALVICO : Quasi Natural Language Understanding in the Semantic Domain of Robotics. Proceedings Conference on Applied Robotics. Plsen, Czechoslovakia. October 1977.

[27] G. GUIDA: Artificial Intelligence Techniques for the Design of Data Bases. (In Italian). Proceedings An-

nual AICA Conference. Milan, Italy. October 1976.

[28] G. GUIDA: Ideas About Design of Natural Language Interfaces to Query Systems. Proceedings IIASA Workshop on Natural Language for Interaction with Data Bases. Laxenburg Schloss, Austria. January 1977.

[29] G. GINI, M. GINI, and M. SOMALVICO: Emergency Recovery in Intelligent Robots. Proceedings Fifth International Symposium on Industrial Robotics. Chicago, Illinois, USA. September 1975.

[30] P. CORTI, G. GINI, M. GINI, and M. SOMALVICO : Problem Solving and Automatic Emergency Recovery: Towards the Design of Intelligent Industrial Robots. Proceedings Second CISM-IFTOMM Symposium on the Theory and Practice of Robots and Manipulators. Warsaw, Poland. September 1976.

[31] L. MEZZALIRA and M. SOMALVICO : Towards Firmware Problem Solvers. Proceedings International Symposium on Mini- and Microcomputers and Their Applications. Zurich, Switzerland. June 1975.

[32] F. AIRAGHI, G. BERTIN, L. MEZZALIRA, and M. SOMALVICO: Design and Experimentation of a Microprogrammed Problem Solver. Proceedings International Symposium on Mini- and Microcomputers and Their Applications. Zürich, Switzerland. March 1976.

[33] R. BISIANI and R. CASSINIS: The Development of a Multi-Microprocessor System to be Used in the Control of an Industrial Robot. Proceedings International Symposium on Mini- and Microcomputers and Their Applications. Zürich, Switzerland. March 1976.

[34] M. COLOMBETTI, P. CORTI, and E.PAGELLO : Manipulating by High Level Languages. Proceedings Conference on Applied Robotics. Plsen, Czechoslovakia. October 1977.

[35] E. PAGELLO and G. PASQUALOTTO : Artificial Intelligence: A New Scientific Ideology or a New Technology? Scientia. Autumn. 1973.

[36] C. CIBORRA: Industrial Robots and Organization of Work. Proceedings Twentythird International Scientific Conference on Electronics. Rome, Italy. March 1976.

[37] U. MONTANARI : Heurstically Guided Search and Chromosome Matching. Artificial Intelligence, Vol. 1, n.4. 1970.

[38] A. MARTELLI and U. MONTANARI : On the Foundations of Dynamic Programming. Topics in Combinatorial Optimization (S. Rinaldi ed.) CISM, Springer Verlag.

[39] A. MARTELLI and U. MONTANARI : Programmazione Dinamica e Punto Fisso. Atti Convegno di Informatica Teorica, Mantova. November 1974. 1-19. (In Italian).

[40] A. MARTELLI and U. MONTANARI : From Dynamic Programming to Search Algorithms with Functional Costs. Proc. Fourth Int. Joint Conf. on Artificial Intelligence, Tbilisi. September 1975. 345-350.

[41] A. MARTELLI and U. MONTANARI : Additive AND/OR Graphs. Proc. Third Int. Joint Conf. on Artificial Intelligence, Stanford.August 1973. 1-11.

[42] A. MARTELLI and U. MONTANARI: Optimizing Decision Trees Through Heuristically Guided Search. Internal Report B76-22. IEI-CNR. Pisa. November 1976.

[43] G. LEVI and SIROVICH F.: Generalized AND/OR Graphs and Their Relation to Formal Languages. Internal Report B73-15. IEI-CNR Pisa. November 1975.

[44] A. MARTELLI and U. MONTANARI: Unification in Linear Time and Space: A Structured Presentation. Internal Report B76-16. IEI - CNR. Pisa. July 1976.

[45] A. MARTELLI and U. MONTANARI : Theorem Proving with Structure Sharing and Efficient Unification. Internal Report S-77-7. ISI. Università di Pisa. February 1977.

[46] A.R. MEO and G.U. RIGHINI: A new technique for Analyzing Speech by Computer. Acustica, vol. 25, pp. 261-268. Nov. 1971.

[47] R. DE MORI : Syntactic Recognition of Speech Patterns. Communication and Cybernetics. Volume 14: Syntactic Pattern Recognition, Applications. Ed. K.S. Fu. Springer-Verlag,Berlin, Heidelberg.1977.