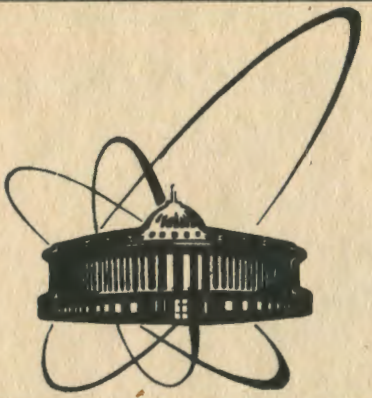

сообщения
объединенного
института
ядерных
исследований
дубна

E10-90-474

F.Fernandez Nodarse

NETSYS, AN AI SYSTEM BUILDING TOOL

1990

INTRODUCTION

Recently researchers have begun to experiment with software that is specifically designed for groups - a type of software often called groupware. Most of the designers of intelligent groupware systems have concentrated primarily on the underlying intelligent functionality of the system, not on their user interfaces.

The Artificial Intelligent (AI) methods and techniques have been applied to several High Energy Physics areas like machine vision, robotics, intelligent data retrieval, and combinatorial and scheduling problems. A key characteristic of AI programs is " heuristic search". The following table shows a classical comparison between AI and conventional computer programs :

Artificial Intelligence	Conventional Computer Programming
- Primary symbolic processes.	- Often primarily numeric.
- Heuristic search (solution steps implicit).	- Algorithmic (solution steps explicit).
- Control structure usually separate from domain knowledge.	- Information and control integrated together.
- Usually easy to modify, update and enlarge	- Difficult to modify.
- Some incorrect answer often tolerable.	- Correct answer required.
- Satisfactory answer usually acceptable	- Best possible solution usually sought.

Much useful research has taken place on deductive databases, which are a natural outgrowth of relational database theory. As a marriage between logic programming and database technologies, deductive databases can form the foundation for integrated, high-end reasoning system [1].

The AI system tools differ substantially in the degree of support they provide for control, knowledge representation, user interaction, and explanation capability, both due to improvements in the field over the past decade and because of the differing goals of their implementors.

The goals of this study are to develop an AI system building tool making computer work easy and familiar to physicists using relational database management tools, and to test some control strategies.

The purpose of knowledge representation is to organize required information into a form such that the AI program can readily access it for making decisions, planning, recognizing objects and situations, analysing scenes, drawing conclusion and other cognitive functions. The classical AI concepts used in this paper were taken from "Principles of Artificial Intelligent" book [2].

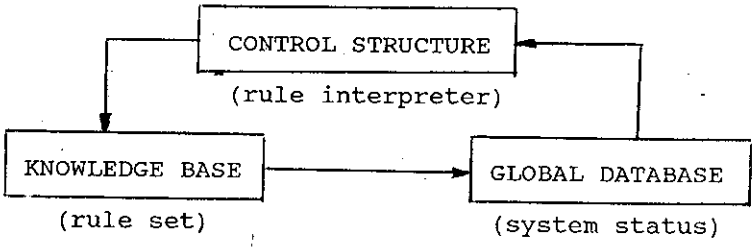

Fig. 1. A production system

OVERVIEW

The NETSYS is an AI system building tool, in which can be identified a central entity that might be called a global database that is manipulated by certain well defined operations, all under the control of some global control strategy.

The global database is the central data structure used in the AI production system (See Fig.1 A production system).

The production rules operate on the global database. Each rule has a precondition that is either satisfied or not by the global database. If the precondition is satisfied the rule can be applied.

Selecting rules and keeping track of these sequences of rules already tried and the databases they produced constitute what we call the control strategy for production system.

The operation of AI production system can thus be characterized as a search process in which rules are tried until some sequence of them is found that produces a database satisfying the termination condition.

Graphs (or more specially, trees) are extremely useful structures for keeping track of the effects of several sequences of rules. We can keep track of the various rules applied and the databases produced by a structure called a search tree. At the top or root of the tree is a description of the initial configuration. The variuos rules that can be applied correspond to links or directed arcs to descendant nodes, representing those states that can be reached by just one move from the initial state. A graph-search control strategy grows such a tree until a database is produced that satisfies the termination condition (Fig. 2).

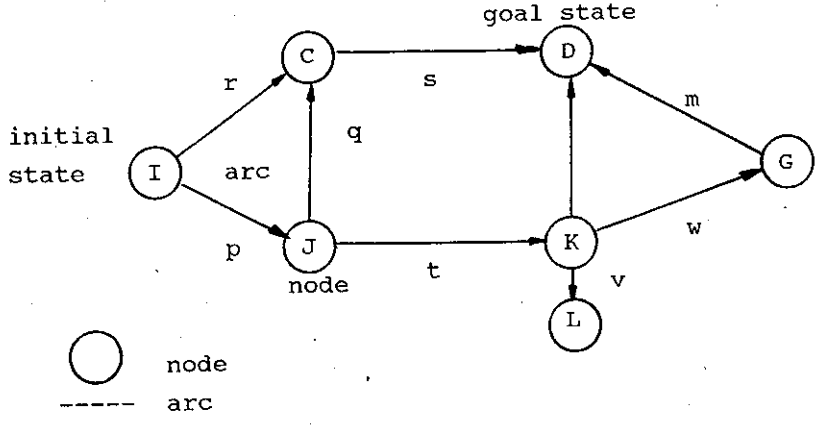

Fig. 2. State graph for a simple problem

A graph consists of a not necessarily finite set of nodes. Certain pairs of nodes are connected by arcs and these

arcs are directed from one member of the pair to the other. Such a graph is called a directed graph and NETSYS works with directed graph. Often it is convenient to assign numeric values or coefficients (costs) to arcs and/or nodes to represent the cost of applying corresponding rule.

In the NETSYS, the nodes and arcs are explicitly given. This explicit specification may be impractical for large graph. In the NETSYS, we use the notion of a successors operator, that is applied to a node to give all of the successors of that node (and the cost of the associated arcs).

Using the FOXPLUS database management system (DBMS) as an infrastructure, we developed the NETSYS (version 1).

The requirements of the NETSYS system are : IBM XT/AT, PS/2 or compatible personal computer, DOS 3.0 or higher, hard disk and 640 Kb RAM.

NETSYS CONTROL STRATEGY

We distinguish two major kinds of control strategies in the NETSYS: irrevocable and tentative. In the irrevocable control regime an applicable rule is selected and applied irrevocably without provision for reconsideration later. In the tentative control regime, an applicable rule is selected (either arbitrarily or perhaps with some good reason), the rule is applied, but provision is made to return later to this point in the computation to apply some other rule. The user can operate the NETSYS using irrevocable or tentative control regimes. As a tentative control regime, the NETSYS uses graph-search control. In graph-search control, provision is made for keeping track of the effects of several sequences of rules simultaneously. When infallible local knowledge is available an irrevocable production system can use it to construct the explicit global knowledge of a solution (without having the explicit global knowledge originally).

We can think of the NETSYS graph-search control strategy as a means of finding a path in a graph from a node representing the initial database to one representing a database that satisfies the termination condition of the production system.

In the NETSYS all the coefficients/costs that the user can assign to arcs and nodes can be used in the graph search control strategy. For our purposes the nodes are labeled by databases or programs, and the arcs are labeled by rules.

In the simplest type of problem, we desire to find a path (perhaps having minimal cost) between a given node S representing the initial state or database and another given node t, representing some other database. The more usual situation, though, involves finding a path between a node S and any member of a set of nodes {ti} that represent databases satisfying the termination condition. We call the set {ti} the goal set, and each node t. in {ti} is a goal node.

All problems have certain aspects: an initial situation, a goal (desired situation) and certain operators (procedures or generalized action) that can be used for changing situations. In solving the problem, a control strategy is used to apply the operators to the situation to try to achieve the goal (See Fig. 3). The control strategy operates on the procedures to generate a sequence of actions (called a plan) to transform the initial conditions in the situation into the goal conditions. Normally, there are also constraints (specifying the conditions necessary for a specific procedure to be applied) which must be satisfied in generating a solution. In the process of trying to generate a plan, it is necessary for the problem solver to keep track of the actions tried and the effects of these actions on the system state.

In the NETSYS the user can select uninformed or heuristic graph search procedure. Some arbitrary scheme must be used in the graph-search algorithm, if nó heuristic information from

the problem domain is used. The resulting search procedure is called uninformed. As uninformed graph search procedure the NETSYS uses depth-first search. For many tasks it is

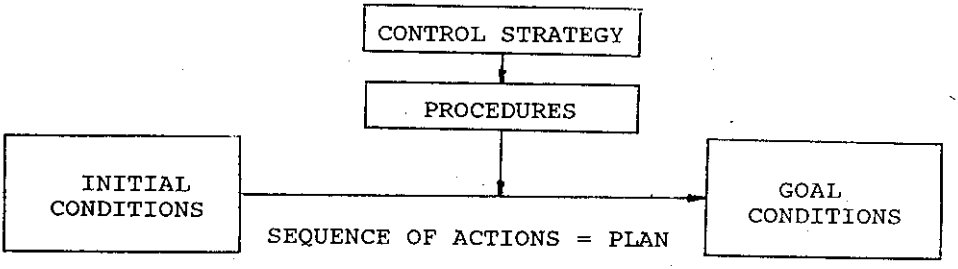

Fig. 3. Problem solving strategy

possible to use task-dependent information to help reduce the search. Information of this sort is usually called heuristic information and search procedures using it are called heuristic search methods. One important method uses a real-valued function over the nodes called an evaluation function. Evaluation functions have been based on a variety of ideas: attemps have been made to define the probability that a node is on the best path; the distance or difference metrics between an arbitrary node and the goal set have been suggested; or in board games or puzzles, a configuration is often scored points on the basis of those features that it possesses that are thought to be related to its promise as a step toward the goal.

As a standard operation functions, the following search procedures were implemented in the NETSYS (See Fig. 4):

1- find any/all paths between two nodes
2- find the optimal path between two nodes depending on the optimal path definition like that :
   - minimal sum of node and/or arc coefficients in the path
   - maximal sum of node and/or arc coefficients in the path
   - minimal number of nodes in the path
3- step by step depending on the user's selection : the user selects the node successor in the graph based on a list of all node successors at the actual or

another level, helped by NETSYS operation options.

4- find all the paths between the initial and goal nodes at the level : DEMO-PATH

5- find the neuro-path between two nodes based on the selection of the next arc in the path by :

$$\min\{ \mathrm{mod}(Parc_j - (\sum_{i=1}^{n} P_i * X_i - O))\}$$

where :

$Parc_j$ - weight/ coefficient of the output arc (j) of the node

P_i - weight/ coefficient of the input arc (i) of the node

O - weight/ coefficient of the node

$$X_i = \begin{cases} 1 - \text{this input arc of the node is in the path (ancestor)} \\ 0 - \text{" " " " isn't in the path} \end{cases}$$

Remember that a neural network is a directed graph consisting of a number of nodes, or processing elements. Each node processes the incoming signal based on the values of constants stored in it. When a processing element (node) is activated, it evaluates all its inputs and computes their respective weight values. If the weight value is above a certain threshold, the computing unit (node) generates an output value that is used as input by the other processing element (only the weight values of input change during learning). Training a neural network is a matter of adjusting weights, either manually or automatically [3].

Always the search procedure checks if the circular path exists, and depending on the user criteria take it into account or avoid it.

The use of the basic subprograms set, like a skeleton of the graph search procedure, may help the application designer to construct and include his operation function in the Operation Function Menu of his application. It may be a good way to obtain an optimal solution for the user problem.

```
------------------------------------------------------------
                        NETSYS
                  AI SYSTEM BUILDING TOOL
                  OPERATION FUNCTIONS MENU
    LEVEL : _____
    APPLICATION : _____
------------------------------------------------------------
      CODE                  DESCRIPTION
       -        STANDARD NETSYS OPERATION :
       S           - STEP BY STEP DEPENDING ON USER SELECTION
       W           - FIND ANY/ALL PATHS BETWEEN TWO NODES
       P           - FIND OPTIMAL PATH BETWEEN TWO NODES
       D           - DEMO : FIND ALL PATHS BETWEEN INITIAL
                            NODES AND GOAL NODES AT THE LEVEL
       N           - FIND NEURO-PATH FROM INITIAL NODE
       U        FUNCTION DEFINED BY APPLICATION DESIGNER
       H        HELP

       Q        RETURN TO PREVIOUS MENU
------------------------------------------------------------
-------> TYPE ONE OPTION : ____
```

Fig. 4. Operation functions menu

All the operation functions operate in the multilevel structure defined by the application designer.

```
------------------------------------------------------------
                        NETSYS
                  AI SYSTEM BUILDING TOOL
                       MAIN MENU
------------------------------------------------------------
    CODE                  DESCRIPTION
     D          CREATE/UPDATE THE MULTILEVEL STRUCTURE
     I          INSERT/UPDATE DATA IN THE MULTILEVEL STRUCTURE
     R          OPERATE THE USER APPLICATION
     C          RESTORE THE USER APPLICATION
     S          SAVE THE USER APPLICATION
     E          ERASE THE USER APPLICATION
     H          HELP AND GENERAL DESCRIPTION

     Q          QUIT TO DOS
------------------------------------------------------------
-----> TYPE ONE OPTION : _____
       TYPE THE APPLICATION NAME : _____________
       TYPE YOUR PASSWORD : _____________
```

Fig. 5. Main menu of netsys

NETSYS REGIMENS

In the NETSYS we distinguish three main regimens (See Fig. 5) :

1- create/update the multilevel structure ("D")

2- insert/update data in the defined structure ("I")

3- operation of the user application ("O")

Using the NETSYS tools, the application designer define the structure of his application. This structure is based on levels (up to 99 levels). Each level can include more than one graph. Relations between graphs at the same level or at other levels can be established. Each node has its code, short description, coefficient cost and optionally, an associated procedure (that can be automatically executed from this node), or an associated database (to which the user can access from this node). Input/output files to/from the node can be defined.

```
------------------------------------------------------------
                        NETSYS
                AI SYSTEM BUILDING TOOL
          CREATE/UPDATE MULTILEVEL STRUCTURE MENU
  APPLICATION : _____________________________
------------------------------------------------------------
  CODE                    DESCRIPTION
   D      INITIAL MULTILEVEL STRUCTURE DEFINITION
   L      INCLUDE/EXCLUDE A LEVEL IN THE MULTILEVEL STRUCTURE
   C      CREATE/UPDATE USER APPLICATION HELPS
   N      DEFINE/UPDATE STRUCTURE OF AUXILIARY DESCRIPTION
   F      MODIFY THE USER OPERATION FUNCTION
   H      HELP

   Q      RETURN TO PREVIUOS MENU
------------------------------------------------------------
--------> TYPE ONE OPTION : ___
```

Fig. 6. "D" regimen menu

The arc and node costs/coefficients can be used by the heuristic function in the graph search control strategy. The associated procedure may be, for example, an expert system or a scientific programs chain.

The NETSYS user can define for each level an auxiliary description of the nodes, adding new description fields.

In the "D" regimen, the user defines/updates the number of levels; the application identification and level description (that he can display using help screens); the structure of the auxiliary node description at each level (optional); and the user control strategy to be used in the "O" regimen (optional) (See Fig. 6).

In the "I" regimen, the NETSYS user inputs/updates data in the defined structure (See Fig. 7 and 8). It means, he inserts or updates the information about the basic or auxiliary description of the node including the associated database and associated procedure, about the arcs description, and about the input/output files description. Delete and list options were included in the update submenus.

```
------------------------------------------------------------
                        NETSYS
                AI SYSTEM BUILDING TOOL
     DATA INSERT/UPDATE MENU IN THE MULTILEVEL STRUCTURE
    LEVEL : _____________________
    APPLICATION : _____________________
------------------------------------------------------------
          CODE              DESCRIPTION
           N                NODE
           L                RELATIONS WITH OTHER LEVELS
           F                FILE
           R                RELATIONS BETWEEN NODES
           C                CHANGE ACTUAL LEVEL
           H                HELP

           Q                RETURN PREVIOUS MENU
------------------------------------------------------------
------> TYPE ONE OPTION : _____
```

Fig. 7. "I" regimen menu

In the "O" regimen the user operates a user application that was developed using NETSYS tools, and defines the operation options and operation function to be used in the

irrevocable or tentative control strategy (See Fig. 9). The utilities may be called to help the operation process.

```
------------------------------------------------------------

                        NETSYS
                AI SYSTEM BUILDING TOOL
                NODE UPDATE/INSERT MENU

     LEVEL: _____________________
     APPLICATION: _____________________
------------------------------------------------------------
      CODE                DESCRIPTION
       N          UPDATE BASIC/AUXILIARY NODE DESCRIPTION
       A          CREATE/UPDATE ASSOCIATED DATABASE
       R          UPDATE/INSERT RELATIONS BETWEEN NODES
       L          UPDATE/INSERT RELATIONS WITH ANOTHER LEVEL
       I          UPDATE/INSERT INPUT FILES TO NODE
       O          UPDATE/INSERT OUTPUT FILES FROM NODE

       Q          RETURN TO PREVIOUS MENU
------------------------------------------------------------
-------> TYPE ONE OPTION :_____
```

Fig. 8. Node update/insert menu

```
------------------------------------------------------------

                        NETSYS
                AI SYSTEM BUILDING TOOL
                    OPERATION  MENU

   LEVEL : _____________
   APPLICATION : _____________
------------------------------------------------------------
          CODE              DESCRIPTION
           F          OPERATION FUNCTIONS MENU
           O          OPERATION OPTIONS MENU
           U          UTILITIES MENU
           C          CHANGE ACTUAL LEVEL
           H          HELP

           Q          RETURN PREVIOUS MENU
------------------------------------------------------------
-------> TYPE ONE OPTION : _______
```

Fig. 9. Main menu of "O" regimen

The operation options of "O" regimen (See Fig. 10) are : to activate the operation options for all the nodes in the path or only the goal node; to display the auxiliary

description and/or associated databases; to display the information about input/output files; to display the graphic diagram of a path (See Fig. 11); to execute the associated procedure; and to use the utilities set to help NETSYS operation. The application user can select any combination of operation options.

```
------------------------------------------------------------------
                          NETSYS
                  OPERATION OPTIONS MENU
   LEVEL : _____________
   APPLICATION : _____________
------------------------------------------------------------------
                      CODE     OPTION DESCRIPTION
* OPTIONS REGIMEN      S   - STEP BY STEP (PATH)
                       E   - FINAL or GOAL NODE
* NODE DESCRIPTION     B   - BASIC
                       X   - BASIC + AUXILIARY
                       A   - BASIC+AUXILIARY+ASSOCIATED DATABASE
* FILE VIEW            N   - NO FILE
                       I   - INPUT FILES
                       O   - OUTPUT FILES
                       F   - INPUT AND OUTPUT FILES
* GRAPHIC VIEW         G
* UTILITIES            U
* MODULE ACTIVATION    M   - ASSOCIATED MODULE ACTIVATION
* HELP                 H

* QUIT                 Q   - RETURN TO PREVIOUS MENU
------------------------------------------------------------------
-----> TYPE ONE OPTION : ____
```

Fig. 10. Operation options menu

Using these utilities (See Fig. 12) the user can display the node description (auxiliary or basic) at the selected level, display the graphic diagram of a path, access to associated databases (See Fig. 13) and modules, change the selected level, compute the path between two nodes and display initial (root) and goal nodes set at the selected level.

The user employs interactive screen menus to select the next action to be taken. All the actions are executed on the basis of multilevel structure defined by the application designer.

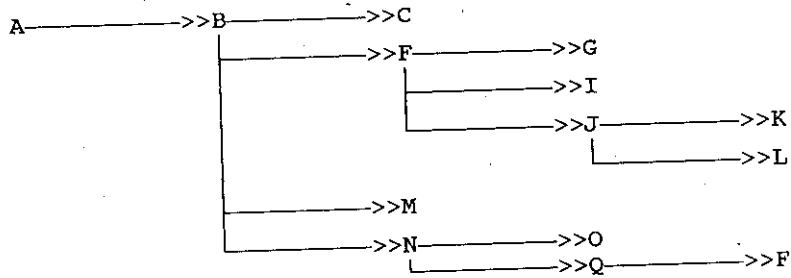

Fig. 11. Example of grapfic diagram of graph

```
------------------------------------------------------------
                        NETSYS
                   UTILITIES MENU

  LEVEL : _____________
  APPLICATION : _____________
------------------------------------------------------------
     CODE                DESCRIPTION
     ----                -----------
       D          LEVEL DESCRIPTION
       L          RELATIONS BETWEEN LEVELS
       R          RELATIONS BETWEEN NODES AT THIS LEVEL
       F          FILE SUBMENU AT THIS LEVEL
       N          NODE : BASIC/AUXILIARY DESCRIPTION,
                         GRAPHIC GRAPH AND ASSOCIATED DATABASE
       S          ASSOCIATED DATABASES MENU
       X          AUXILIARY DESCRIPTION AT THIS LEVEL
       M          ASSOCIATED MODULE
       C          CHANGE ACTUAL LEVEL
       H          HELP

       Q          RETURN TO PREVIOUS MENU
------------------------------------------------------------
-------> TYPE ONE OPTION : _____
```

Fig. 12. Utilities menu

Three kinds of user may be defined depending on his access attributes : application designer, application updater and end user.

Each user has his NETSYS password to access a user application, and his access attributes ("D", "I" and "O"). Using the access attributes, the application designer defines in which regimen other users can work, and protects the data and the multilevel structure of his application.

```
------------------------------------------------------------
                        NETSYS
                AI SYSTEM BUILDING TOOL
                ASSOCIATED DATABASE MENU
LEVEL : _____________________
APPLICATION : _____________________
------------------------------------------------------------
CODE                        DESCRIPTION
----                        -----------
  D         ASSOCIATED DATABASES DESCRIPTION AT THIS LEVEL
  S         SEARCH IN AN ASSOCIATED DATABASE
  L         LIST AN ASSOCIATED DATABASE
  U         CREATE/UPDATE/ERASE AN ASSOCIATED DATABASE
  R         RESTORE/SAVE DATABASES
  C         CHANGE ACTUAL LEVEL
  H         HELP

  Q         RETURN TO PREVIOUS MENU
------------------------------------------------------------
-------> TYPE ONE OPTION : _____
```

Fig. 13. Associated database menu

The user with "D" attribute (application designer) can operate with all the main NETSYS menu options : save/restore the user application, define/update the user application structure, erase the user application, insert/update data in the structure defined, or operate the user application. In "D" regimen the application designer defines the application name, passwords and user atributte access.

The user with "I" attribute (application updater) can save, insert/update data in the defined structure and operate the user application, that was implemented using NETSYS tools.

The user with "O" attribute (end user) only can operate the user application that was developed using NETSYS tools.

FINAL REMARKS

The final remarks of this study are following :

1- computer system can be more useful when two basic rules are followed : expose incomplete knowledge and make it easy to include informal knowledge and processing.

2- The choice of the operation function critically determines search results. Some heuristic search methods greatly reduce search efforts, but do not guarantee finding

minimal cost paths. The precision of our heuristic function depends on the amount of heuristic knowledge it possesses about the problem domain and on the particular factors specific of a given problem. The selection of the heuristic function is crucial in determining the heuristic power of search algorithm. The NETSYS feature to select or construct the operation function and to test some control strategies permits the application designer to select the best global control strategy for his application.

3- The NETSYS utilities, interactive menus and operation options constitute a debugging and diagnostic tools, that are user-friendly. The feature of NETSYS node to optionally include auxiliary description, associated database or associated procedure, was very useful.

4- The remaining problems of many expert system building tools are : essentially no support for genuinely distributed problem solving, that is, true parallel execution of problem solving subtasks across multiple processors; no provision of relations that represent time intervals or spatial relations; weak or no support for functions in the algebraic sense; the absence of a sufficient variety of mechanisms for representig "confidence" or "belief"; inadequate support for interaction with the external environment; and poor support for managing internal conflict (for example, the absence of good support tools for detecting and dealing with conflicting data, conflicting rules, and conflicts between the user's expectations and the rule-based system's conclusions in an interactive problem-solving environment). Work on some of these deficiencies is already under way.

5- In certain applications, the neural network was configurated in only a fraction of the time it would have taken a knowledge engineer to configure and build the expert system, and it seems to avoid the impasse of having to laboriously construct and maintain expert system. Particularly you shall check the training time, the perfomances estimates and debugging diagnostic requirements

when you choose neural network configuration. The most important features of many neural network models are their distributed nature, the ability to take incomplete data and produce approximate results, and their parallelism, speed and trainability that make them fault-tolerant. The neural network are fast and efficient for handling large amounts of data. One criticism of classic neural network models is that they are not as flexible at representing knowledge as standard methods are.

6- Up to now, classical neural networks have not replaced conventional methods of computing - especially those that deal with high-speed numeric processing - but well complement them and add to their utility. Accurracy, computational power and logic are not among their strong points. The combination of traditional computers and neutral network could unravel problems that otherwise would remain unresolved.

7- When you can achieve the same or better results with conventional technologies, AI system isn't recommendable to make use of.

The NETSYS system can be used in intelligent retrieval from complex databases system, expert consulting system, automatic process and combinatorial and scheduling problems. NETSYS may be very useful when the heuristic knowledges about the problem domain and on the particular factors specific of a given problem are incomplete.

Based on NETSYS, an AI system building tool, intelligent assistants can be developed . These can take the form of expert system and intelligent computer-aided instruction system.

REFERENCES

1. Ketonen J.A., "Toward Reasoning about Data", AI EXPERT, Febrary 1989, p. 44-49.
2. Nilsson N.J., "Principles of Artificial Intelligence", Palo Alto, CA, Tioga Pr., 1980.
3. Obermeier K.K. and Barron J.J., "Time to get Fired Up", BYTE, August 1989, BYTE, vol 14 number 8, p. 217-224.

Received by Publishing Department
on October 19, 1990.