Robotics & Computer-Integrated Manufacturing, Vol. 4, No. 1/2, pp. 77–82, 1988
Printed in Great Britain.

0736-5845/88 $3.00 + 0.00
Pergamon Press plc

• *Paper*

AN ATTEMPT TO IMPLEMENT EXPERT SYSTEM TECHNIQUES IN CAPP

ALOJZ SLUGA,* PETER BUTALA,* NADA LAVRAČ† and MATJAŽ GAMS†

*University of Ljubljana, Faculty of Mechanical Engineering, Ljubljana, Yugoslavia and †Jozef Stefan Institue, Ljubljana, Yugoslavia

This paper presents the development of a generative computer-aided process planning system that covers part of macro level planning activities. The system's task is to generate all appropriate machining operation sequences and select the most promising ones, according to a given set of criteria, while the final choice is to be made by the user. The system acts as a classical "production" system that interprets expert rules on input workpiece data according to the knowledge and data in the engineering knowledge base. By using the technological knowledge in rule form the system largely constrains the solution space and produces a reasonably small set of meaningful alternatives. The advantage of the expert system approach also lies in the program's flexibility which enables easy adjustment of the system according to specific shop-floor environments.

1. INTRODUCTION

The increasing need for the integration of design and manufacturing activities, as a consequence of the demand for higher production quality and efficiency, leads us to the development of a computer-aided process planning (CAPP) system as a still missing link between CAD and CAM.

The problem of developing a CAPP system has employed many researchers in the last decade. Several systems have been developed over the world.[1–3] These systems are mainly based on classical algorithmic techniques which are deficient mainly because of their inflexibility. Besides the reduction of planning time and paper work, the benefit of these systems is mainly due to rationalization of planning processes and standardization of process plans.

The process planning is based on judgmental expert knowledge and intuition. The encouraging results of prototype expert systems in particular domains, as for example the MYCIN project in the field of medical diagnosis, have influenced the idea of knowledge technology implementation in the process planning domain. This is reflected in some experimental systems such as GARI,[4] TOM[5] and EXCAP.[6]

In this paper the knowledge engineering approach to CAPP is presented and is implemented in the OPEX (operation planning expert) system.

2. PROCESS PLANNING

Process planning is the activity that determines the appropriate procedure to transform raw material (usually in some prespecified form) into a final product.[3] It represents a link between design and manufacturing activities. Its basic task is to determine by what means and how a product is to be manufactured economically and competitively.

Process planning involves, according to the part specification, determination of processes, machine tools, clamping devices, operation sequences, tools, machinability data and calculations of time and costs. The results obtained must be reflected in documentation.

Traditionally, process planning is performed by skilled planners. Planning is based on planners' decisions. The quality of the plan highly depends on individual skill, knowledge and experience. Planning is very time consuming, because much routine work and calculation must be done. Only little time is left for creative work and updating planners' knowledge.

Computer-aided process planning should diminish as much as possible deficiencies in traditional process planning. The advantages lie in planning-time reduction, in prompt preparation of information for decision making, as well as in planning objectives.

The real integration of CAD and CAM becomes possible with the introduction of a CAPP system in a chain. The integration can be obtained by a connective information flow and a central data base as an integration key element.

Within the concept of the CAPP system which we are developing, a hierarchical structure has been adopted as shown in Fig. 1. The system splits into three levels with regard to the nature of decisions to be made in the course of the planning process. On the global level, the determination of cutting processes and machine tools must be done. The decisions are based on global knowledge, group technology and general data. On the macro level, one defines bases for clamping a workpiece, clamping strategy and appropriate types of fixture set-ups. Operations and operation sequences are to be selected next. On the micro level, all the competing manufacturing alternatives are evaluated. First, one determines machine tools, tools, fixtures and cutting data. On the basis of these data, the best manufacturing alternative can be defined according to time and cost criteria. The best alternative is selected, detailed and documented.

Our aim is to obtain input data within the technological part model which is already generated by CAD[7] according to the approach to CAD/CAPP-/CAM integration introduced by Peklenik *et al.*[8]

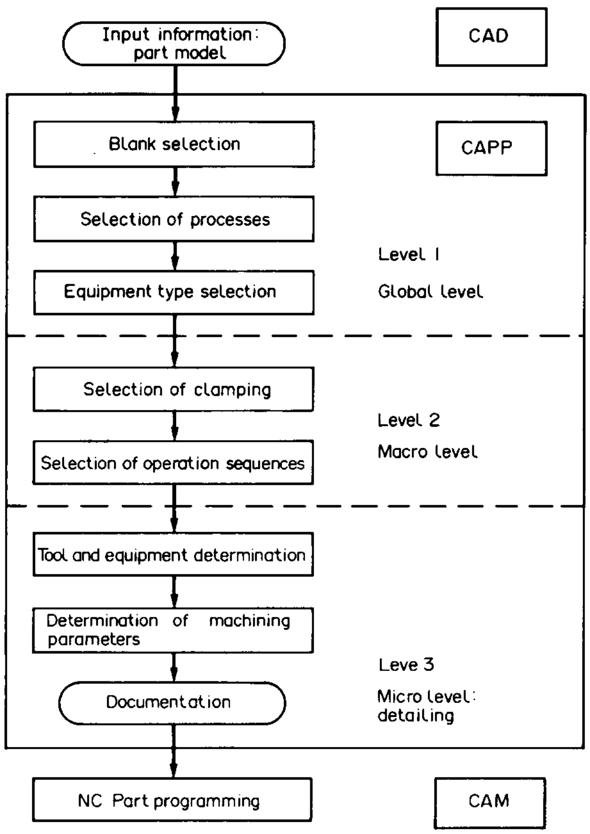


Fig. 1. The structure of the CAPP system.

Output should consist of the necessary documentation and the process plan, which should be stored in the same central data base.

3. EXPERT SYSTEM APPROACH TO CAPP

In recent years, there have been some quite promising attempts to implement AI methods in CAPP[4-6] which largely increased the interest in this field. Although almost all systems are still in the development stage, agreement is attained that the AI approach offers some advantages:

- greater flexibility due to the chosen expert system structure
- the possibility of capturing expert knowledge in the expert system formalism
- efficient methods dealing with searching trees with several optimal solution possibilities.

They use Pascal or Prolog or even a commercially available expert system shell or language.

Only in recent years, has AI gained wider attention, which is especially due to expert system techniques. Expert systems actually belong to pattern-directed systems and are their most successful branch.[9-12] Comparisons between pattern-directed systems and classical systems are shown in Fig. 2.

In classical systems we have data and program (= algorithm). The program calls and updates data. Flow control is more or less predetermined and has only a small number of possibilities in branching points.

In pattern-directed systems, rules compete for evaluation while the determination of which one of them can be triggered depends on data in the global data base. The interpreter's task is to check the rules

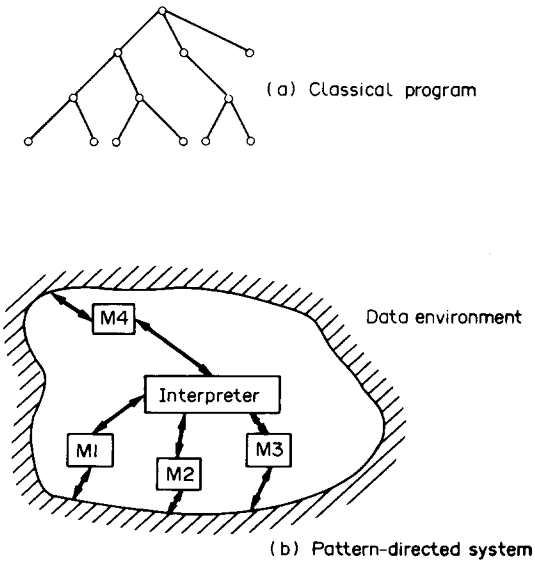


Fig. 2. Structure of classical and pattern-directed systems.

that match the data and to choose the most promising one.

In an expert system, we have a knowledge base (consisting of the domain expert knowledge, static and dynamic data) and an inference mechanism which interprets rules in the knowledge base according to the data. A communication interface should provide an explanation of how the system works and enable changing and updating of the system, expecially of the rules in the knowledge base.

More precisely expert systems have the following structure as shown in Fig. 3. Knowledge is usually represented in the form of IF–THEN rules or antecedent–consequent rule forms.[11] This form povides a natural and concise representation of behavioral knowledge of the sort: "If there is a state with characteristics A, then the appropriate thing to do is C." These rules are easily interpreted both by people who wish to understand what effects they will have and by machines that actually apply them to data.

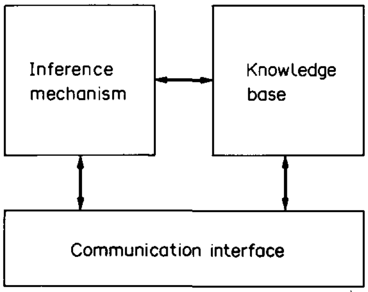


Fig. 3. Structure of expert systems.

Several advantages come from the expert system structure:[12]

- modularity: each rule defines a small, relatively independent piece of knowledge
- incrementability: new rules can be added to the knowledge base more or less independently of other rules
- modifiability: the present rules can be changed quite independently of other rules
- transparency: a rule itself is transparent, while the chaining of rules is explained by special mechanisms in the communication module and
- it enables user-friendliness (why/how facilities, etc.).

Among languages suitable for programming expert systems Prolog is perhaps the most important and successful. Prolog is based on first order predicate logic and has many powerful features such as unification and backtracking. Prolog can be seen as a theorem prover which tries to prove all statements.[13] The speed of programming an expert system in Prolog is such a great advantage that it fairly overcomes its slow execution.

4. SYSTEM OVERVIEW

Because of the complexity of a CAPP system it is reasonable to develop such a system in several stages. At first a module for operation sequence planning, called OPEX, was developed. The operation sequence planning module can be used as a stand-alone module, as a technological aid for NC part programming and other activities.

4.1. Program description

The program is split into two parts. First, the program generates all sensible alternatives of operation sequences for all individual features of the workpiece according to their attributes. Next, on the basis of the results obtained, the program generates all sensible operation sequences for the workpiece as a whole, adds the number of cutting tools and estimates machining time for each alternative. Results (operation sequences) are sorted according to the number of necessary tools. Solutions with the same number of tools are sorted according to the value of the performance function.

The generation of alternative operation sequences for individual features is based on backward planning and recursion similar to that of Matsushima *et al.*[5] and Derbyshire and Davis.[6] Backward planning implies that one generates operation sequences from the final state of the feature, i.e. from the goal that has to be achieved through intermediate states to the initial state, the blank. This process runs recursively from state to state.

4.2. Program structure

The system structure can be seen in Fig. 4. It fits well into the expert system frame with the exception that its communication module is at a base level.

The system consists of an engineering knowledge base and an inference mechanism. The engineering knowledge base can be divided in facts, such as static data, and a rule base in which planning logic and technological expert knowledge are included.

Inference mechanism. It consists of a Prolog program within which an interpreter performs and controls the appropriate application of rules. For example, an element of the interpreter for evaluating logical expressions is:

```
log_exp(LE and RE):-
    log_exp(LE), log_exp(RE)
log_exp(LE or RE):-
    log_exp(LE); log_exp(RE).
```

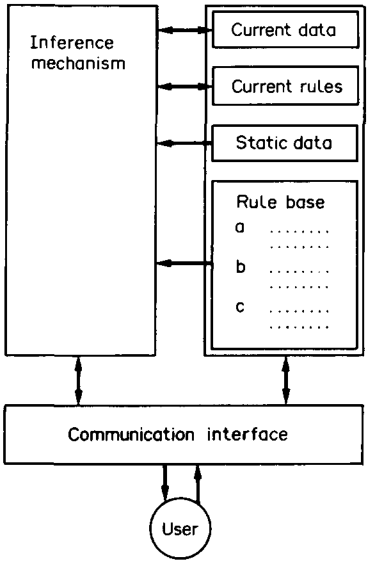


Fig. 4. Structure of OPEX.

The mathematical scheme is based mostly on intervals. An interval defined as a pair of numbers where the first number represents the lower limit and the second number the upper limit. The interval operations are:

- set operations: union, intersection, max and min_limit, subset, disjoint, included;
- arithmetic operations: +, −, *, /, div, mod;
- comparisons: =, <, >, etc.

Rules. At present there are three kinds of rules:

(a) Rules for applying basic machining operations.
The rule form is:
operation Operation_name
if Condition
then New_changes
end.

In this case Condition and New_changes (in Prolog words beginning with capital letters are variables) are expressions combined from subexpressions connected with reserved words such as and, or, not, in.

Example
```
operation drilling
if
    gdb:fc is_a cylinder_in                and
    gdb:dc included interval (3,40)        and
    gdb:lc/max(gdb:dc) = 10
    gdb:nc subset interval (11,12)
then
    fc: = is_a blank                       and
    dc: = 0                                and
    nc: = undefined
end.
```

(b) Rules that define various possibilities of linking basic machining operations within an individual feature.
The rule form is:
from Operation 1 to Operation 2
if Condition end.
Example:
from boring to drilling
if true end.

(c) Rules for combining operation sequences that define which operation sequences should be adopted for a combination of features.
The rule form is:
combination Operation 1 and Operation 2
if Condition end.
Example:
combination drilling and drilling if true end.

Technological knowledge is expressed in terms of IF–THEN rules (rules of form a). Planning logic as common knowledge of the domain is represented in rules of forms b and c.

Facts. Facts are tool data, machinability data, quantitative characteristics of the domain such as time standards, etc.

Global data base. All dynamic data are stored in a global data base, i.e. data on features, such as cylinder, cone, face, etc. Each feature is described by a list of geometrical (dimensional) and technological (surface quality) attributes in a similar form to the program input.

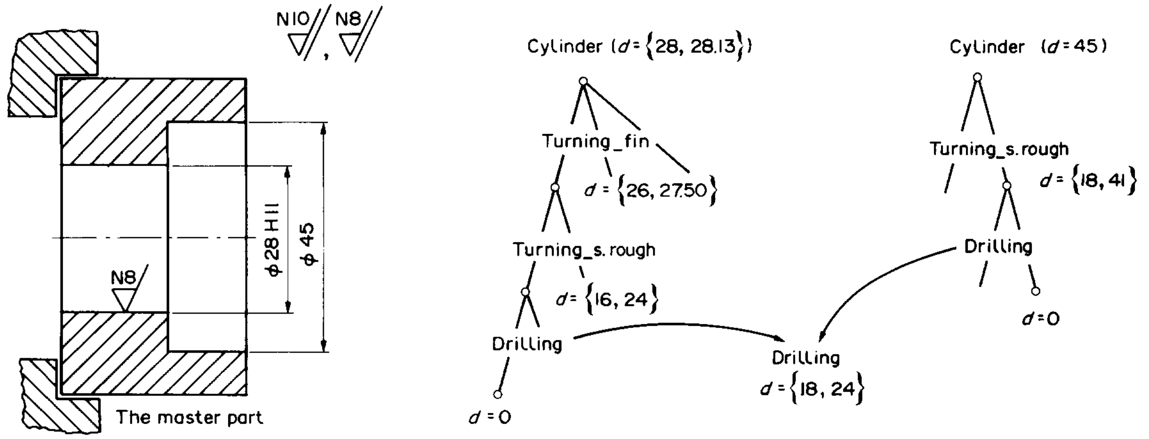
4.3. Input output

The program input consists of all data of a workpiece and of a blank necessary for planning activities. The program output is a list of operation sequences. An example can be seen in Fig. 5 where the task is to machine a hole. The hole has three features, two internal cylinders and one face. A cylinder is defined by three parameters: diameter, length and surface quality. Any data can be a number or an interval.

Input data:

cylinder_in(dc = 45, lc = 15, nc = 10),
face_in(nc = 10),
cylinder_in(dc = (28,28.13), lc = 20, nc = 8).

The program's output consists of a list of sorted alternatives of machining sequences. P.I. in the output table means the value of the performance function and N.T. means the number of needed tools.



The trees of operation sequences for the two cylinders

Fig. 5. A case study.

Output data:

Machining sequence	P.I.	N.T.
1 drilling (), rough_turning (), . . .	211	2
. . . .		
3 drilling (), rough_turning (), . . .	370	3
. . . .		

4.4. Planning process description

The rule interpreter matches the condition part of a rule with the data in the global data base. If the match suceeds, the rule is applied and the global data base is updated according to the action part of the rule. This process is repeated recursively until the stop conditions are achieved. This kind of algorithm is called forward-chaining or data driven. Planning logic (rules of form b) limits possible branching of the solution tree in order to obtain only sensible solutions. In this way the program develops a tree of allowed operation sequences for any individual feature. The starting (initial) state in the global data base is the finished feature while the stop (goal) state represets the blank. Nodes in a tree represent intermediate states while branches represent operations. The intermediate state describes the state of a feature in terms of dimensional and quality tolerances. Planning operation sequences for an individual feature are thus obtained. For machining, the order of operations in a sequence is reversed.

When the solutions for the whole list of workpiece features are found, the system tries to find all operations that are common to two or more features. Planning logic (rules of form c) defines what operation sequences of individual features can be combined in a compound operation sequence for the workpiece as a whole. One can combine equal operations of two or more features. When an intersection of feature states for the same operation exists, the operation is suitable for these features. In this way new alternatives with corresponding intersection states are generated. This principle of combining is particularly important in bulk removal operations. On the basis of the number of operations for each alternative, the number of tools needed for machining is determined.

Three criteria are used to estimate the suitability of every alternative. These are the number of tools used for an alternative, the time performance function and the index of the operation utility.

The number of tools is limited by the machine tool selected. On the other hand batch size might influence the choice of the number of tools. At present, only alternatives with a predefined number of tools are suggested to the user.

The value of the time performance function is calculated for every alternative. It is the sum of estimated machining time for each operation in a sequence. The alternative with the minimum value of the performance function at defined number of tools is supposed to be the best solution. Anyway, the final decision is left to the user.

The basic strategy in operation planning is to remove most of the material in rough operations. When considering combining operations this is not always the case. The utility index is a ratio between a volume that could be removed by a combined operation on the intersection upper limit and a volume that could be removed by the same operation on individual feature upper limit. The utility index reflects the efficiency of the combined operation. It only serves as a test analysis.

Once the operation sequence is obtained, the selection of cutting tools and cutting data can be done. This part of the program is still under development.

5. CONCLUSIONS

The OPEX system forms a module for operation sequence planning and is treated as a part of the complex CAPP system. It serves as an aid to the user in decision making. It prepares and selects the most promising solutions while the final choice is made by the user.

The system is based on expert system techniques within which the engineering knowledge base, inference mechanism and communication interface are integrated. Domain knowledge is expressed in terms of production rules. An interpreter with built-in recursion uses the forward-chaining strategy.

The program is written in Prolog and runs on a VAX 11/750 in the POPLOG programming environment.

The planning logic of the system is based on characteristics of an individual feature. Further efforts should be concentrated on the investigation of expert knowledge for judging the relevance of the generated alternatives from a more complex point of view, i.e. the characteristics of a workpiece as a whole, tools, clamping and machine tools are to be considered. We expect that introduction of this kind of knowledge will bring the system to a higher level of performance.

In system development much more work should be done on the communication interface. A higher degree of user-friendliness will be achieved through graphics simulation and with a direct link to the CAD object model at the input and the NC-part programming system at the output.

The flexibility and modularity of the OPEX system enable the implementation of the system in operation sequence planning for a wide spectrum of rotational parts to be machined on NC lathes. Implementation is possible simply by adapting a basic set of rules and adding the rules relevant for the given shop-floor environment.

REFERENCES

1. Weill, R., Spur, G., Eversheim, W.: Survey of computer-aided process planning systems. *Ann CIRP* **31** (2) 1982.
2. Eversheim, W., Schulz, J.: Survey of computer-aided process planning systems. CIRP Working Seminar on CAPP, Paris, 1985.
3. Wysk, R.A., Chang, T.C., Ham, I.: Automated process planning systems, an overview of ten years of activities. 1st CIRP Working Seminar on CAPP, Paris, 1985.
4. Descotte, Y, Latombe, J.C.: GARI: a problem solver that plans how to machine mechanical parts. *Proc. IJCAI*, 1981.
5. Matsushima, K., Okada, N., Sata, T.: The integration of CAD and CAM by application of artificial-intelligence techniques. *Ann. CIRP* **31** (1) 1982.
6. Darbyshire, I., Davies, E.J.: EXCAP—an expert system approach to recursive process planning. 16th CIRP International Seminar on Manufacturing Systems, Tokyo, 1984.
7. Grum, J., Logar, B., Hlebanja, G., Peklenik, J.: Design on the database for CAD based group technology. *Robotics Computer-Integr. Mfg* **4**: 49–62, 1988.
8. Peklenik, J., Grum, J., Logar, B.: An integrated approach to CAD/CAPP/CAM and group technology by pattern recognitions. *Mfg Systems* **14** (1) 1985.
9. Nilsson, N.J.: *Principles of Artificial Intelligence*. Tioga, 1980.
10. Winston, P.H.: *Artificial Intelligence*. Addison-Wesley, 1984.
11. Waterman, D.A., Hayes-Roth, F. (eds.): *Pattern-Directed Inference Systems*. Academic Press, 1978.
12. Bratko, I.: Expert systems and Prolog. Pergamon-Infotech State of the Art Report on Supercomputer System Technology, London, 1982.
13. Clockin, W.F., Mellish, C.S.: *Programming in Prolog*. Springer, 1981.