CAMEX - AN EXPERT SYSTEM FOR PROCESS PLANNING ON CNC MACHINES

O. Eliyahu, L. Zaidenberg and M. Ben-Bassat

IET - Intelligent Electronics
14 Esser Tachanot St,
Ramat Hachayal, Tel Aviv,
Israel,69719

and

Faculty of Management,
Tel Aviv University
Tel Aviv, Israel, 69788
B25@TAUNIVM.BITNET

ABSTRACT

CAMEX is an expert system designed to plan machining processes for CNC (Computerized Numerical Control) cutting machines. At the present state of development it is constrained to parts for which 2 1/2 D description is sufficient. For this kinds of parts, CAMEX is able to read a drawing of a workpiece from an ordinary CAD file, to understand its 3-dimensional structure and generate a plan for producing the workpiece. CAMEX is implemented in FRANZ LISP on an APOLLO workstation

KEYWORDS

Expert systems, CAD/CAM, mechanical engineering, CNC machines, rule-based systems.

Topic

Expert Systems for Mechanical Engineering.

I. INTRODUCTION

A CNC (Computerized Numerical Control) cutting machine obtains as input a block of material; e.g. a rectangular block of aluminium, and produces a workpiece with a desired shape by a series of repeated cuts. A cut is characterized by several parameters including size and shape of the cutter; whether it is a rough or final cut, offset values, etc. Cut selection and ordering have not yet been automated. The person who decides on the machining process essentially bridges the gap between sophisticated CAD systems that are used to draw the desired workpiece, and sophisticated CAM systems that are capable of executing a given plan, once it is decided upon.

The CNC industry uses the term "technology" to describe the plan for producing a given workpiece; i.e. the sequence of cutters and their characterizing parameters. This term may be somewhat confusing out of the CNC context. Nevertheless, we elected to adopt it throughout the paper.

Generating a technology for complex and/or large parts may take a highly qualified expert weeks of intensive effort, and this establishes the need for some degree of automation, or, at least, decision support tools. The need for automation is enhanced by the realization that mistakes in the design are non-recoverable (one cannot "fill" material which has been removed by mistake) and very costly (CNC machine time is very expensive).

The planning process cannot be described by closed algorithms and/or formulas. It is based for the most part on human expertise, i.e. detailed knowledge about the characteristics of materials and machine capabilities, as well as experience and problem-solving skills.

CAMEX is an expert system designed to plan machining processes ("technologies") for CNC cutting machines. At the present stage of development, the use of CAMEX is restricted to parts for which a 2 & 1/2 D description is sufficient. These are parts which may be fully described by one projection, e.g. view from above, and associated one-valued function for defining the height (or depth) at each point. For parts of this kind, CAMEX obtains as input an ordinary CAD file for the desired workpiece and generates as output a technology for producing it.

CAMEX is implemented in FRANZ LISP on an APOLLO workstation (only the user-interface part of code is machine-dependent). It consists of about 12000 lines of LISP and about 3000 lines of "C" code.

II. GENERAL APPROACH

Three main components are required for a system such as CAMEX:

1. Problem representation, that is the system's ability to perceive (to "see") a workpiece in the various stages of its production and to recognize the legitimate tools and their capabilities.

2. Knowledge base, that is machine representation of the knowledge used by human experts in designing a CNC technology.

3. Inference and control mechanism, that is algorithms that - upon "understanding" a given workpiece - access relevant parts of the knowledge base and construct a technology for producing it.

III. PROBLEM REPRESENTATION

The starting point of a human expert is a technical drawing on paper or on computer screen displays. Looking at the drawings, he creates in his mind a 3-D model of the desired workpiece and then proceeds to generate the technology.

The first step in CAMEX development was to provide it with a 3-D model of the desired workpiece. We started by developing a language for describing the geometrical properties of the workpiece. The idea behind the language was the notion of a human CNC expert who has become blind and now requires an assistant to describe the desired workpiece to him.

In this language, the workpiece is described as a list of geometrical primitives with their attributes and the relations between them. Two types of primitives exist in the language : subparts and surfaces. Subparts are classified into 2 types: cavities and material. For example, profile is a primitive of type cavity, and by external profile we mean all the material that must be removed from the initial block in order to reach the external wall of the workpiece. Most of the primitives were chosen by virtue of their representing basic technological structures (pocket, profile, hole, bay etc.). The attributes of a primitive are described in terms of Dmax (diameter of largest possible cutter), Dmin (smallest corner diameter), etc. The relations between primitives are described in such terms as is-above, is-below, is-aside, is-limited-by, etc.

Figure 1 provides a description of a workpiece using the language. A similar approach to the workpiece-geometry description was used by Descotte and Latombe [Descotte and Latombe,1981].

It was soon realised, however, that describing real-world workpieces in such a language is a very time-consuming and error-prone activity. Worse still was the fact that CAMEX could in no way verify the user-supplied description. On the other hand, it was discovered that a substantial amount of the relevant geometrical information could be extracted from CAD files of the technical drawings. We proceeded therefore to developing a preprocessor that would generate a workpiece representation directly from the CAD files.

Today, CAD systems describe part geometry as a collection of low-level geometrical primitives: line segments, polylines (strings), circles and arcs, and splines (see Figure 2). This collection, when presented graphically to the human eye, allows the human brain to imagine the 3-dimemensional geometry of the part.

In addition to "real" geometrical primitives such as points and line segments which are actually seen when viewing the workpiece, a CAD file also contains a large amount of auxiliary information, such as dimension lines and textual material. Such information must be identified and removed from the CAD file if we are going to attempt automatic interpretation. Other problems in real-world CAD files are limited precision (can we assume 85.4 and 85.5 to be the same number ?), overlapping lines, etc.

The CAMEX preprocessor "cleans" the CAD file of non-relevant elements, scans it and produces as output a geometrical database that describes the workpiece in terms of higher-level primitives - such as pockets, holes, profiles, etc. These primitives are displayed to the user, who is requested to supply the height (depth) of each primitive. The resulting workpiece description is implicitly equivalent to the language description mentioned above. The preprocessor goes beyond the mere identification of the basic geometrical entities. It also searches the data base for thin walls boundaries, i.e. walls with a width less than some prespecified threshold. Thin walls play an important part in technological decisions, and it is more efficient to collect information about them in the preprocessing stage.

A one-to-one link is maintained between the original CAD primitives and the higher-level geometrical primitives. Thus the user may point to any region in the drawing and make queries in the form:

is a specified region a wall ?,
a hole ?,
a pocket ?

what are the neighbors of a specified region ?

(Such queries may also be used to check the claim that CAMEX really "understands" the workpiece.)

```
************************************
           profile-b
************************************
is-a: profile
is-above: ()
is-below: ()
is-aside-of: (wall-b7 wall-b8)
is-limited-from-above-by: ()
is-limited-from-below-by: ()
is-limited-from-side-by: ()
is-in: ()
z-high: 20
z-low: 0
d-max: 50
d-min: 20
r-fillet: nil
depth: 20
width: 60
clearance: nil
tolerance: nil
area: 10000
```

Figure 1: A description of a workpiece using CAMEX language.

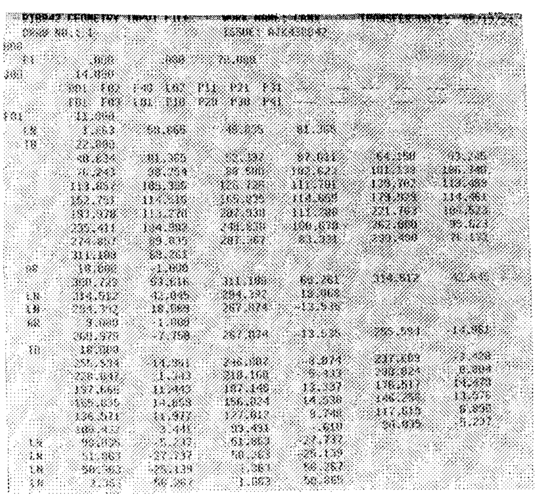


Figure 2: A typical CAD file.

IV. REPRESENTATION OF KNOWLEDGE.

Geometrical knowledge is embedded in CAMEX problem representation. Additionally, knowledge about the use of CNC machines for a variety of workpieces made of various types of material is provided in the form of rules.

Here are a few examples of the rules:

```
IF total-volume of nc-jobs for tool
     is less than 50000
   AND nc-jobs for tool are pockets
     only
   AND tool is larger than 16
   AND smaller tool is with rad 0
THEN change tool to smaller tool.

IF nc-entity is pocket or profile
   AND wall-thickness is less than 2.0
THEN set wall-offset to 0.05
   AND DEFINE wall-finish
```

The rule base is treated not as a static collection of knowledge chunks, but as a kind of very high-level programming language for describing the technology-generation process. This is achieved by rules that guide the control strategy and assist in breaking down the problem into subproblems and in determining the order in which the subproblems are to be solved. For instance, the rule

```
IF tool was changed
THEN PERFORM select nc-entity tools
```

guides the system to use a set of rules relevant for tool selection for a single nc-entity.

Such an organization has the advantage of efficiency, because at every step of technology generation only a small group of rules is eligible for checking. Thus the cycle time of each rule application is independent of the total number of rules in the rule base. The primary disadvantage is inconvenience in the debugging of the knowledge base, because the meaning of some rules may depend on the organization of the rule base.

The rules are formulated by the experts in structured English. CAMEX has a rule translator module that automatically translates rules into LISP and adds them to the knowledge base (see Figure 3). This provides the experts with a great deal of independence in maintaining the knowledge base and in checking the impact of rule modifications.

V. INFERENCE MECHANISMS

CAMEX works in four main steps:

Step 1. Removal identification: CAMEX

starts by identifying a set of fillers (cylinders with arbitrary cross-sections), which, upon removal from the initial block of material, produce the workpiece. There are generally many such sets (one is obtained by defining a filler for each region with z-coordinate less than the height of the initial block), but technological considerations make some sets illegal, and some preferable. The rules for choosing fillers are constraints on legal removals. For example, in Figure 4, alternative (a) consists of the two cylinders above the regions A and C with z-extent from 3mm to 16mm, and the cylinder above the region B with z-extent from 12mm to 16mm. Alternative (b) consists of the cylinder above regions A, B and C with z-extent from 12mm to 16mm, and the two cylinders above regions A and C with z-extent from 3mm to 12mm.

The process is basically a depth-first search, and in most practical cases the search space is quite small (perhaps tens of possibilities). To handle the rare cases where the search space is large, rules of thumb are used to limit the space. At the end of this search we have a set of fillers each of which corresponds to a removal (pocket, profile, top-of-wall etc.).

The resulting list of removals approximately corresponds to the list of nc-primitives which, in the early version of CAMEX, were explicitly entered by the user (see the section on Problem Representation and Figure 1). Size parameters (such as Dmin for pockets and profiles) and spatial relations between removals (aside-of, above, etc.), which previously were explicitly specified by the user, are now determined easily from the geometrical database on an as-needed basis.

Step 2. Technology generation for individual removal. Rules such as:

```
IF nc-entity is pocket
  AND fillet rad is smaller
    than 5.0
  AND floor thickness is greater
    than 3.0
THEN set floor offset to 0.0
AND set wall offset to fillet rad
```

are applied in a forward-chaining manner to produce a list of operations for each removal. Each removal (nc-entity) defines one or more operations (cuts, nc-jobs). Each operation is defined by the relevant removal, the diameter and corner radius of the cutter, and other parameters.

Step 3. Cutter optimization: The purpose of this step is to achieve better utilization of the cutters by taking a global view of the workpiece.

The relevant rules have the form:

```
IF cutter is used only once
THEN remove it from
    part-tools-list
  AND retry select-part-tools.
```

Different criteria may be used for cutter optimization. For example, time of processing of each workpiece may be crucial for large production lines; however, for prototyping, time of generating a feasible technology (not necessarily an efficient one) may be more important.

Step 4. Sorting of operations: Basically, the relevant rules for this step are constraints on the legal order of operations. Two kinds of constraints exist: "must be" rules and "should be" rules. For instance:

```
wall-top MUST BE before profile
  or pocket with the same wall
```

```
nc-jobs for the same tools SHOULD
BE sequenced by decreasing volume
```

Both kinds of rules imply a partial order of the operations. It is usually not really important which operation comes first, as long as all constraints are satisfied.

VI. IMPLEMENTATION

The main system modules are:

Geometry Understanding Module. Reads the CAD system output file and builds a data structure describing the workpiece geometry. Because of the "noise" included in real world drawings (even computerized ones), a certain amount of interaction with the user is needed at this stage.

Rule Translator Module. This module accepts rules from the user in structured

English and translates them into internal LISP form. This module may be considered as a compiler from a high-level language describing technology generation, into LISP.

Control and Inference Module. This module controls the overall operation of CAMEX; calls procedurally implemented steps; and triggers and fires rules from the rule-base.

Explanation Module. This module traces the process of rule application; translates rules from internal (LISP) form to English and generates explanations regarding the system reasoning process. It allows the user to ask questions such as why a particular operation was added, why, or in what context, a particular rule was used, etc. This module may be considered as a kind of a symbolic domain-oriented debugger.

In Figure 7 we show the end result of CAMEX, i.e. a technology for the entire workpiece.

VII. ACNOWLEDGEMENT

This project was supported by the Engineering Division of Israel Aircraft Industries, Inc. and performed in collaboration with their staff. The authors are specifically grateful to Mr. R. Razi for many useful discussions.

REFERENCES:

[Descotte and Latombe,1981] Y. Descotte & J. Latombe. "GARI : A problem solver that plans how to machine mechanical parts". In Proc. IJCAI-81, Vancouver, 1981.uh. 766-772.

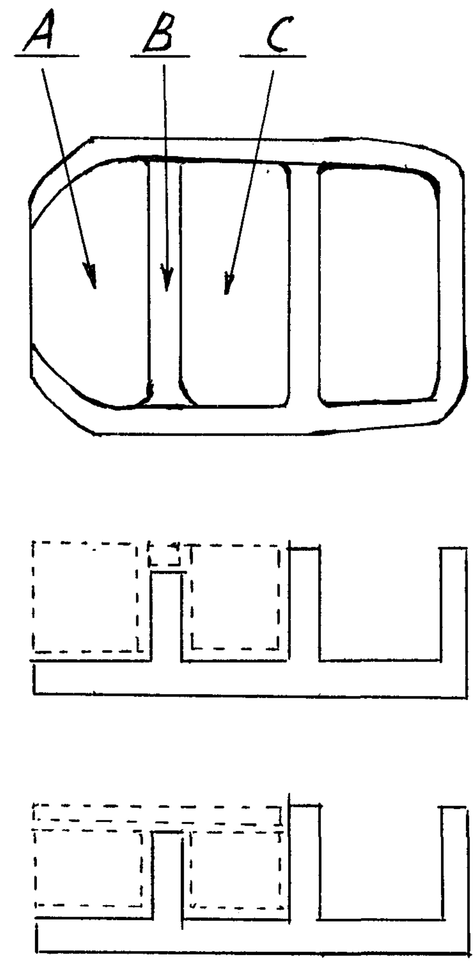
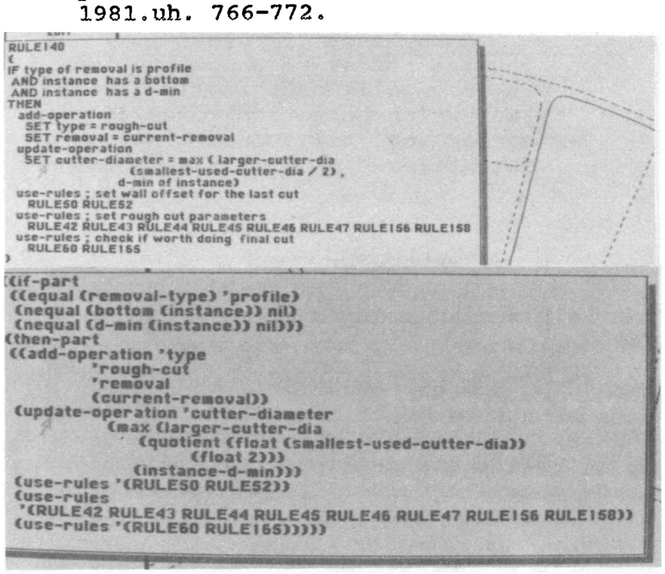


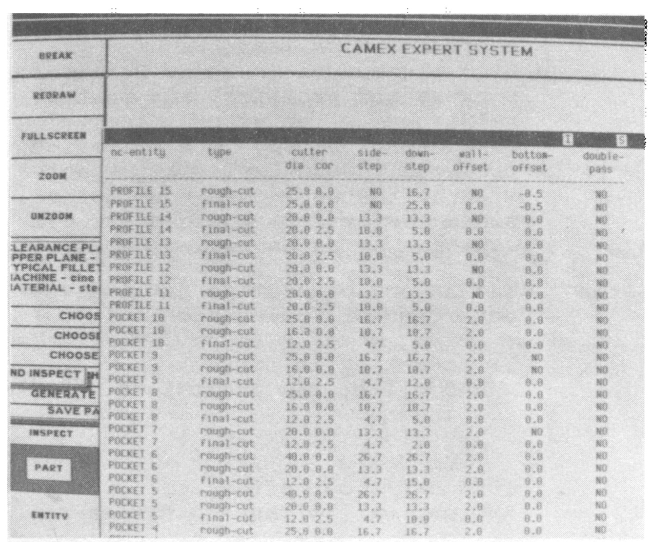
Figure 4: The possible sets of removals that yield the same end product.



```
RULE140
(
IF type of removal is profile
 AND instance has a bottom
 AND instance has a d-min
THEN
 add-operation
  SET type = rough-cut
  SET removal = current-removal
 update-operation
  SET cutter-diameter = max ( larger-cutter-dia
                (smallest-used-cutter-dia / 2),
              d-min of instance)
 use-rules ; set wall offset for the last cut
  RULE50 RULE52
 use-rules ; set rough cut parameters
  RULE42 RULE43 RULE44 RULE45 RULE46 RULE47 RULE156 RULE158
 use-rules ; check if worth doing final cut
  RULE60 RULE165
)
```

```
((if-part
 ((equal (removal-type) 'profile)
  (nequal (bottom (instance)) nil)
  (nequal (d-min (instance)) nil)))
(then-part
 ((add-operation 'type
     'rough-cut
     'removal
     (current-removal))
 (update-operation 'cutter-diameter
     (max (larger-cutter-dia
       (quotient (float (smallest-used-cutter-dia))
         (float 2)))
       (instance-d-min)))
 (use-rules '(RULE50 RULE52))
 (use-rules
  '(RULE42 RULE43 RULE44 RULE45 RULE46 RULE47 RULE156 RULE158))
 (use-rules '(RULE60 RULE165)))))
```

Figure 3: Automatic rule translation



nc-entity	type	cutter dia	cutter cor	side-step	down-step	wall-offset	bottom-offset	double-pass
PROFILE 15	rough-cut	25.0	0.0	NO	16.7	NO	-0.5	NO
PROFILE 15	final-cut	25.0	0.0	NO	25.0	0.0	-0.5	NO
PROFILE 14	rough-cut	20.0	0.0	13.3	13.3	NO	0.0	NO
PROFILE 14	final-cut	20.0	2.5	10.0	5.0	0.0	0.0	NO
PROFILE 13	rough-cut	20.0	0.0	13.3	13.3	NO	0.0	NO
PROFILE 13	final-cut	20.0	2.5	10.0	5.0	0.0	0.0	NO
PROFILE 12	rough-cut	20.0	0.0	13.3	13.3	NO	0.0	NO
PROFILE 12	final-cut	20.0	2.5	10.0	5.0	0.0	0.0	NO
PROFILE 11	rough-cut	20.0	0.0	13.3	13.3	NO	0.0	NO
PROFILE 11	final-cut	20.0	2.5	10.0	5.0	0.0	0.0	NO
POCKET 10	rough-cut	25.0	0.0	16.7	16.7	2.0	0.0	NO
POCKET 10	rough-cut	16.0	0.0	10.7	10.7	2.0	0.0	NO
POCKET 10	final-cut	12.0	2.5	4.7	5.0	0.0	0.0	NO
POCKET 9	rough-cut	25.0	0.0	16.7	16.7	2.0	NO	NO
POCKET 9	rough-cut	16.0	0.0	10.7	10.7	2.0	NO	NO
POCKET 9	final-cut	12.0	2.5	4.7	12.0	0.0	0.0	NO
POCKET 8	rough-cut	25.0	0.0	16.7	16.7	2.0	0.0	NO
POCKET 8	rough-cut	16.0	0.0	10.7	10.7	2.0	0.0	NO
POCKET 8	final-cut	12.0	2.5	4.7	5.0	0.0	0.0	NO
POCKET 7	rough-cut	20.0	0.0	13.3	13.3	2.0	NO	NO
POCKET 7	final-cut	12.0	2.5	4.7	2.0	0.0	0.0	NO
POCKET 6	rough-cut	40.0	0.0	26.7	26.7	2.0	0.0	NO
POCKET 6	rough-cut	20.0	0.0	13.3	13.3	2.0	0.0	NO
POCKET 6	final-cut	12.0	2.5	4.7	15.0	0.0	0.0	NO
POCKET 5	rough-cut	40.0	0.0	26.7	26.7	2.0	0.0	NO
POCKET 5	rough-cut	20.0	0.0	13.3	13.3	2.0	0.0	NO
POCKET 5	final-cut	12.0	2.5	4.7	10.0	0.0	0.0	NO
POCKET 4	rough-cut	25.0	0.0	16.7	16.7	2.0	0.0	NO

Figure 5: A technology for the entire workpiece.