Integration of CAD and CAM System Using Macrofunction*

Yasumi NAGASAKA**, Happy WIBISONO***,
Hideyuki OHTAKI*** and Keiichi WATANUKI***

The macrofunction in the computer aided manufacturing (CAM) system is executed based on the macrocode which is transformed by a series of operations input by the user. By the use of this function, operations which should be performed by the user can be simplified. Therefore, if the transformation from mechanical drawings to NC code can be changed directly into macrocode, NC machining can be achieved easily. In our study, a function for transforming macrocode from data of CAD drawing to CAM data is developed. However, drawing data of a CAD system ("CAD-drawing") does not always coincide with drawing data of a CAM system ("CAM-drawing"), because in a CAM system, restricted conditions, such as machining conditions, must be considered. Consequently, drawing data must be represented by a structure and function which include information about restricted conditions. Thus, the concept of the relationship between "master and servant" for definitely expressing structure and function in an object model is specifically incorporated. As a result, even in the case where structure and function are changed, the knowledge base can be consistently maintained. This paper describes representation of knowledge in the knowledge base and method for generating a macrocode using the knowledge base.

Key Words: Expert System, Artificial Intelligence, CAD, CAM, NC, Knowledge Base

1. Introduction

The problem in integrating CAD and CAM systems is that drawing data of a CAD system ("CAD-drawing") cannot be used directly as drawing data of a CAM system ("CAM-drawing"). This is because in CAD-drawing (for example, IGES; Initial Graphics Exchange Specification) linear elements (straight line, circular arc, etc.) do not have direction, while in CAM-drawing (for example, EXAPT; Extended Subset of Automatically Programmed Tools) directions for linear elements and machining conditions are required. Therefore, the operator is forced to confirm the direction of linear elements in CAD-drawing and to add machining conditions one by one, manually.

Thus, many studies on adding machining conditions for CAD-drawing in a drawing process have been reported. Particularly, as is typical of a Function-model (representation of object named by features as model unit), CAM-drawing is generated based on CAD-drawing directly, by the model, structuring information of parts shape and machining conditions as knowledge. For example, TIPS by Okino et al.[1],[2] is famous.

As a result of these studies, it is now obvious that if linear elements of CAD-drawing are recognized and then model representation similar to a Feature-model is built on the CAM system, CAM-drawing can be generated based on CAD-drawing. If this idea further makes it possible to automatically perform the operation process from CAM-drawing generation to NC machining (corresponding to PART-program in EXAPT system) based on CAD-drawing, CAM operation will be greatly simplified, which will afford better manufacturing efficiency.

For that purpose, first, recognition of linear elements, which was reported previously[3], is necessary. Then, it is also necessary to build the model by which CAM-drawing is generated and maintained based on CAD-drawing. This can be achieved by establishing

* Received 6th December, 1993. Japanese original: Trans. Jpn. Soc. Mech. Eng., Vol. 59, No. 559, C (1993), pp. 947-954. (Received 25th August, 1992)
** ZEXEL Corporation, 3-13-26 Yakyu-Cho, Higashimatsuyama-Shi, Saitama 355, Japan
*** Department of Mechanical Engineering, Saitama University, 255 Shimo Okubo, Urawa-Shi, Saitama 338, Japan

the Feature-model (knowledge base) using knowledge representation based on the concept of the Object-model[4], which expresses object by function and structure. By this, it is possible to pursue the most suitable conditions (as reported previously[5]) in CAD-drawing. Regarding the automatic serial process, the NC-code can now be generated by codifying traditional serial operations (from CAM-drawing generation to NC-code generation) ("Macro-code") and by using the Macro-function of the CAM-system (the function of system operation based on Macro-code). Thus, CAD and CAM are now easily integrated, machining conditions are simply input, and NC machining is performed effectively in a simplified manner.

This report describes knowledge representation of the aforementioned system, the method of Macro-code generation, and examples of these methods.

2. Knowledge Representation of Knowledge Base

2.1 The overall system concept

Figure 1 shows the overall concept and management process of the system. Commercial CAD system (CADKEY) and AI tool ZES[5],[6] (which have been developed by the present author) are implemented in the Sun workstation. Commercial CAM system (SmartCAM) is implemented in a personal computer. Networking between workstation and PC in this system is conducted using ethernet. CAM system is connected to CNC Machine (Roland DG CAMM-3) using RS232C cable.

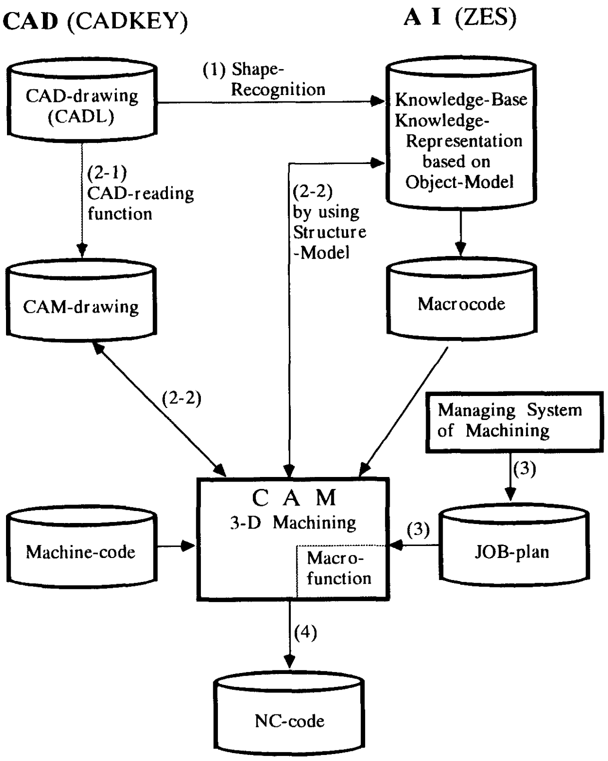


Fig. 1 The overall system concept

Now, let us describe the management process of the system as follows:

(1) CAD-drawing of CADKEY is transformed to CADL (CADKEY Advanced Design Language) format. The knowledge base is stored in the representation, as will be described in section 2.2. Then, CAD-drawing is recognized using Shape-Understanding of CAD drawing, as described in Ref.(3). Recognizing results are stored in the knowledge base (parenthesis 1 in Fig. 1).

(2) Generally in the traditional method, CAD-drawing data is converted to CAM shape files (CAM-drawing), using a converter provided by the CAM system. After that, the operator must confirm the direction of the shape, and then determine suitable machining conditions to cut the part, manually (parenthesis 2-1 in Fig. 1). However, in our system, direction of the shape and machining conditions are generated simultaneously, using the structure model (to be described in section 2.3) in the knowledge base (parenthesis 2-2 in Fig. 1).

(3) The information on suitable machining conditions for cutting the part is stored in the JOB-plan file, determined automatically by the machining management system[5]. The reasoning (inference) of this management system is performed by using a reasoning function of TMS (Truth Maintenance System)[7].

(4) Based on the information in the JOB-plan file (tool-file) and shape file, the NC code for operating the CNC machine in the most suitable cutting process is generated (parenthesis 4 in Fig. 1).

The macrocode in our system automatically performs the operation described above (parenthesis 2-2, 3 and 4).

2.2 Knowledge representation in knowledge base

Figure 2 shows the frame of knowledge representation which is incorporated by object model concept[4]. This object model in our system we call the unit of object.

Figure 2(a) shows the object model concept. In the example in the figure, object [A 1] is a part of object [A], and it has parts [A 11] and [A 12].

Figure 2(b) shows representation in the unit of object [A 1]. "HasPart, Part-of" in the figure expresses the relationship between "whole and part" of the objects. However, the relation of "HasStructure, Structure-of" and the relation of "HasFunction, Function-of" in the figure, show the relationship between "master and servant" for definitely expressing structure and function in an object model.

Because knowledge representation is expressed by the relationship as described above (HasStructure, HasFunction), even in the case where structure and function are changed, the knowledge base can be consistently maintained. More details on this matter will be discussed in section 2.4.

2.3 Structure model representation and function model representation

Here, as an example, rule representation and a template of the basic structure of a cam are presented in Fig. 3.

The left-hand figure shows that a template of the basic structure of a cam consists of linear elements "Base-Arc, Lift-Line, Top-Arc, Side-Arc". If needed, the arc element which is represented by the pointed line is prepared, in order to meet the function model in Fig. 4.

The right side shows rule representation to extract linear elements in the structure template from the CAD drawing. In this figure, for example to recognize "Base-Arc", the following rule is applied.

if: line element "Base-Circle" has been recognized, and the element is an arc, and the center of the arc is the center of the drawing,

then: this line element is recognized as "Base-Arc".

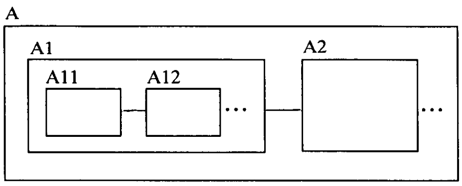
Figure 4 shows lift curve of cam and factors of function model corresponding to Fig. 3. Here, Lift start angle (θ_1), inflective angle (θ_2), lift end angle (θ_3), displacement (s), velocity (v), and acceleration (a), are described in the slot. For example, slot (Lift-Start) is described as follows:

Lift-Start:

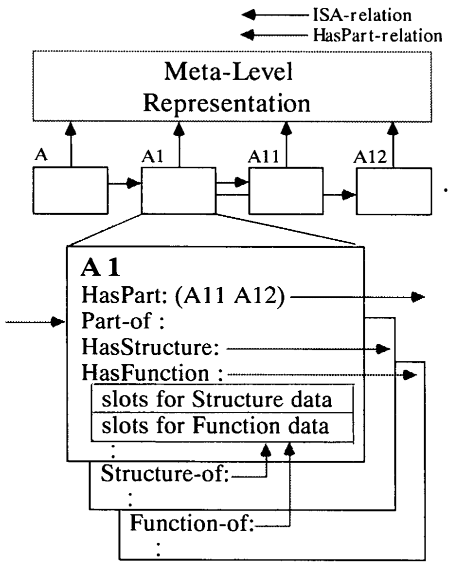
```
(− 180 (asin (/(− Base-Radius Top-Radius)
  (−(+ Max-Lift Base-Radius) Top-Radius))))
  (if-varied (Base-Radius) (Top-Radius)
  (Max-Lift))
```

The upper parentheses express slot value, and the lower parentheses express extended daemon function "if-varied"[8]. The value of the daemon (for example "Base-Radius"), besides functioning as variable of slot value, is that it also functions as slot name. For example, in the case when slot "Lift-Start" is referred to, this daemon function is executed in advance. After that, evaluation in the daemon is conducted (for example, slot "Base-Radius" is referred to; then the result is substituted to variable "Base-Radius"). Then, evaluation of slot value (upper parentheses) is conducted. In the current method, a user defined routine for Lift-Start must be created in advance. Then, this routine can be called as an attached procedure of the frame representation of the knowledge base.

Consequently, with the use of model representation in the knowledge base as above, there is no necessity to build a user defined routine, unlike in the current method.

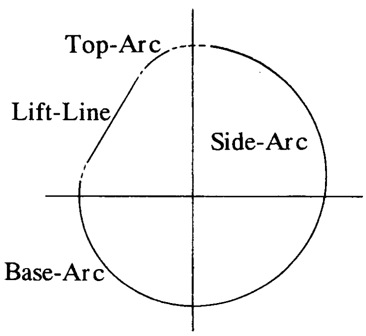


(a) The concept of object model



(b) Knowledge representation in a unit

Fig. 2 Knowledge representation's frame of knowledge base




```
:
(@p Base-Arc
 (goal ?status active ?segment nil
       ?exist (@get-val Base-Circle))
 (line ?type arc ?kind full_thin)
 (elements ?x0 $x_center ?y0 $y_center)
->
 (@make ?status matched ?segment Base-Arc))
:
```

Fig. 3 Rule representation and a template of basic structure of a cam

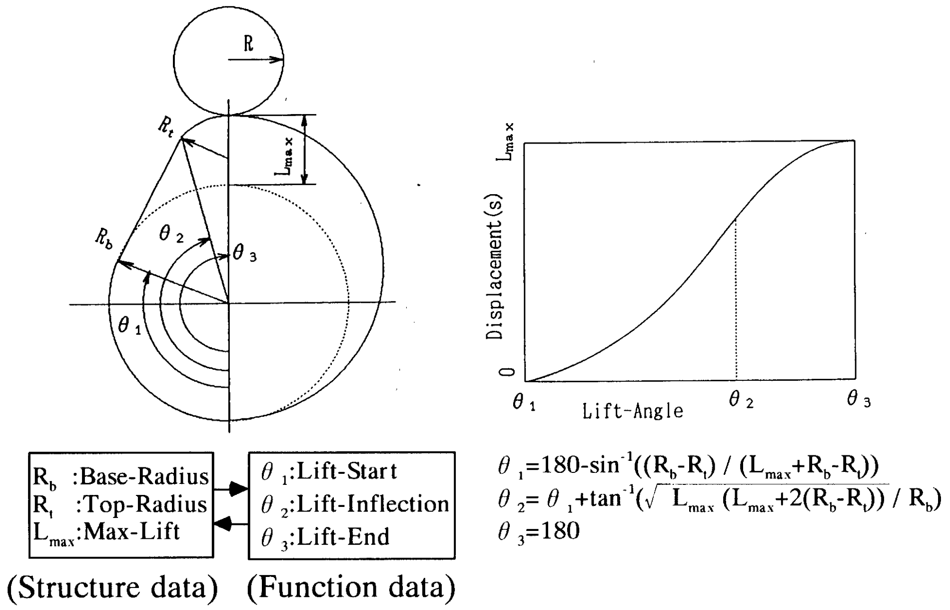


Fig. 4 Function Modeling

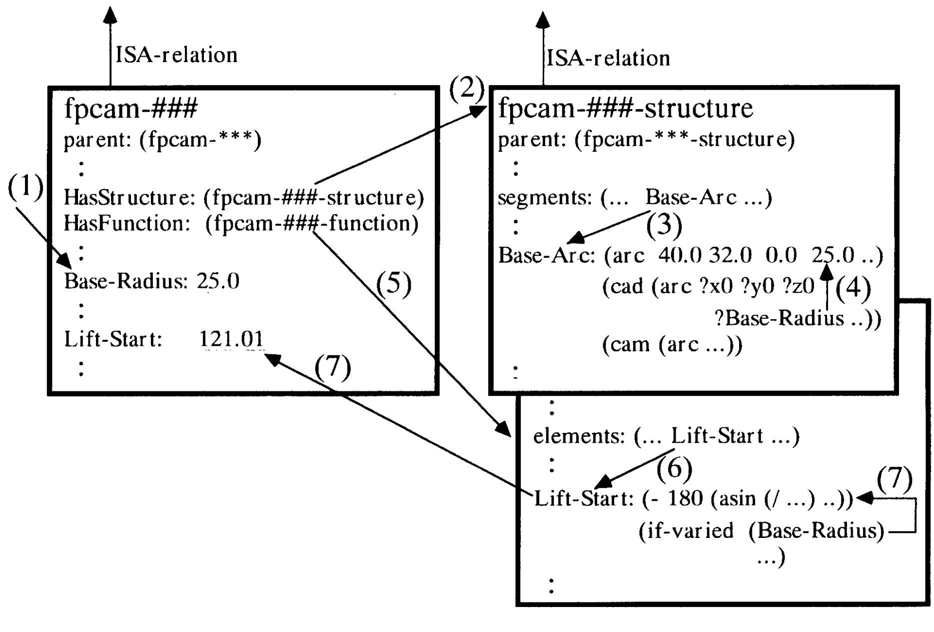


Fig. 5 Knowledge base management process

2.4 Model representation's management

Figure 5 shows the process of how to manage model representation in Fig. 3 and Fig. 4 without contradictions. The example in Fig. 5 shows how slot "Base-Radius" in the knowledge base is maintained consistently when the user renews a CAD-drawing data base. Now, let us explain this process in order of the numbers in the figure.

(1) Consider there to be a change in the value of slot "Base-Radius".

(2) By the master-servant relationship "Has-Structure", unit fpcam-###-structure which is functioning as structure model is referred to ; then steps (3) and (4) are performed.

(3) Slot "Segments" which expresses linear elements in structure model is referred to ; then for all of the values step (4) is performed.

(4) Examine whether the slot which has been changed is used or not, by using the value of attribution[8] "cad". If that slot is used, the slot value will be renewed. Here, attribution is the area for storing information of the slot, and it has specific attributes and functions as a kind of daemon function.

(5) By the master-servant relationship "HasFunction", unit "fpcam-###-function" which is functioning as a function model is referred to ; then steps (6) and

(7) are performed.

(6) Refer to slot "elements" which expresses the elements of the function model; then for all of the values, step (7) is done.

(7) Examine whether the slot which has been changed is used or not, by using attribution "if-varied". If that slot is used, the slot will be evaluated; then the results of the evaluation are maintained in a suitable slot.

Thus, the knowledge base in this system flexibly copes with study to pursue the most suitable machining conditions, when incorporating CAD-drawing information into CAM-drawing.

3. Macro Function and Code Generation

3.1 An outline of macrofunction in CAM systems

Figure 6 shows the concept of macrofunction in a commercial CAM system (SmartCAM). Macrofunction is executed based on the macrocode, as shown in Fig. 7. By using this function, input of a series of operations, which is usually done by the operator, can be performed automatically by the system.

Figure 7 shows descriptions of macrocode in CAM system (SmartCAM). Now, let us describe those codes as follows:

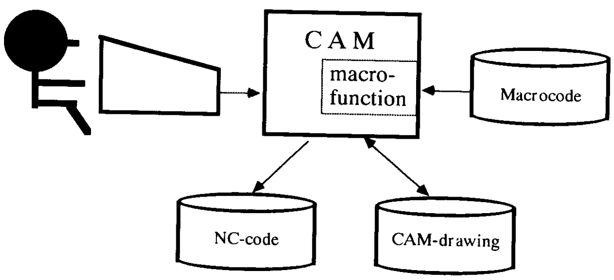


Fig. 6 The concept of macro function

(1) *** : Input Command
file ... file command
save ... save in a file
code ... create NC-code
:
(2) @*** : Input Key
@macstartv1... start macro-function
@cr ... input return key
@f5 ... input function-key-5
@up ... input control-key-up
:
(3) $*** : Input Data
$67 ... input data "67"
$test1 ... input data "test1"
:
(4) #*** : Input comment

Fig. 7 Kinds of macrocodes

"* * *" : this kind of code is used to execute commands provided by the CAM system. For example, "Files, Save,... " performs the operation to save a file.

"@ * * *" : this kind of code is used to perform key stroke operations (input operations from keyboard). For example,"@cr" means hit return key, and "@f 5" means hit function key "f 5".

"$ * * *" : this kind of code is used to input data. For example, "$67" means input data 67.

"# * * *" : this kind of code expresses a comment statement.

3.2 Generation of macrocode

Figure 8 shows the function by which to generate macrocode in our system. The execution (inference) of this function is performed by referring to the frame of the macrofunction (arrows in Fig. 8); then the generation of macrocode is done by determining the variables "? * * *" (see dotted frame in Fig. 8). Here, the frame of macrocode which is provided by the system includes a series of operations from setting up machining condition to the generation of NC code. Now, let us explain the procedure for generation of macrocode in order of numbers in the dotted frame in Fig. 8.

(1) (@macstartv 1) : start macrofunction.

(2) (new...) : initialize machining conditions to generate new drawing data. In this step tool-file; (? tool-file) which stores machining condition information is loaded. In this example, the value of slot "* * *" is substituted for variable "? * * *"; then (new wax 1 @cr) is evaluated, so as a result, tool file "wax 1. jsf" is loaded.

(3) (cam-drawing) : perform the generation of CAM-drawing. In this step, slot "cam-drawing" is referred to; then equations as shown below are evaluated.

(drawing start 0 line 1 line 2 line 3
line 4 start 1 base-arc
lift-line ⋯)

Here, function "drawing" evaluates attribution "cam" (shape macrocode) by the order of slots start 0, line 1, line 2, ⋯. For example in the case of "start 1", attribution is evaluated as follows:

(start_proof $ 40.0 @cr $ 7.0 @cr).

Then the result is stored in macrocode. In the example in this figure,"second, third" means the reference of the second and third slot values. Now, the procedures in the case where the shape macrocode is undefined will be described in subsection 3.3. After that the following macro information is stored.

((start_prof $ 0.0 @cr $ 0.0 @cr)
(line $ 0.0 @cr $ 80.0 @cr) ⋯
(start_prof $ 40.0 @cr $ 7.0 @cr)

(arc 2. cw $25.0 @cr $40.0 @cr accept @cr @cr @cr) …)

(4) (@f5 full) : redraw CAM-drawing.

(5) (files save $mactest …) : save drawing data in file "mactest. sh3". By this step, the generation of macrocode to generate CAM-drawing is completed, so it is possible to stop the process at this step, if desired.

The following are steps in the generation of macrocode to generate NC code.

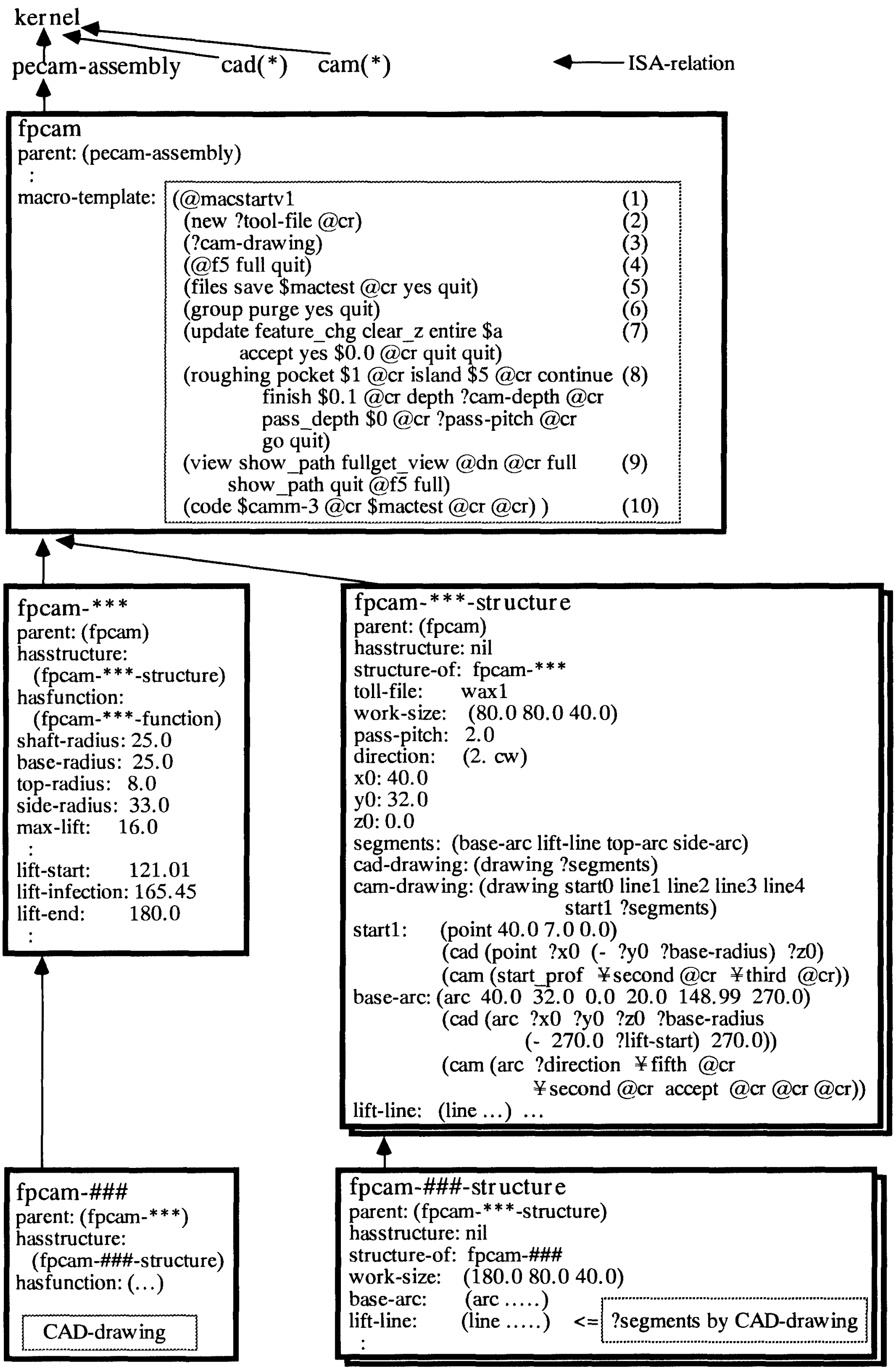


Fig. 8 Macrocode generation function

(6) (group purge ⋯) : data compression.

(7) (update feature_chg clear_z ⋯) : setting up the value of z axis of tool path.

(8) (roughing ⋯) : performing roughing operation. In the example in Fig. 8, determination of machining conditions such as depth of cut, pass depth, and rate of feed is obtained from the tool-file (2). The values are determined based on materials or tools in the most suitable cutting process[5].

(9) (view ⋯) : simulation of tool path. If some undesirable condition is detected, it is also possible to execute a compulsory stop in this step.

(10) (code ⋯) : generation of NC code. According to the tool path which is viewed in step (9), NC code is generated; then the information is saved in file "mactest. t".

So, by the use of the function and structure model representation as described in subsection 2. 3, the most suitable drawing information can be kept as macro information. Furthermore, by the use of the information in the tool-file (job-plan file), information on the most suitable machining conditions can be expressed as macro information, also. As a result, NC machining can be performed more accurately and easily.

3. 3 Generation of shape macrocode

3. 3. 1 CAD-Drawing and CAM-Drawing Figure 9 shows the relationship between CAD-drawing and CAM-drawing. In this example drawing data of a circular arc are expressed as follows:

in CAD-drawing: (arc x0 y0 rad ang1 ang2)
in CAM-drawing: (arc pre-line direction
rad x0 y0 x_end y_end ang_end)

As shown in this example, in CAM-drawing, in addition to CAD-drawing data, we must have information on the pre-line (relationship with the previous line, that is tangential or intersection), and information on the direction of the segments. This is because CAM-drawing data must have information to cut the part in the machining process. So, in contrast with CAD-drawing, in CAM-drawing, information on the pre-line and the direction of the segments are important matters. In order to manage this information, slots "segments" and "direction" in structure model units (see Fig. 8) are presented. Now, information in the dotted frame in Fig. 8, as shown in Fig. 10, can be created by using drawing data of the CAD system (CAD-drawing).

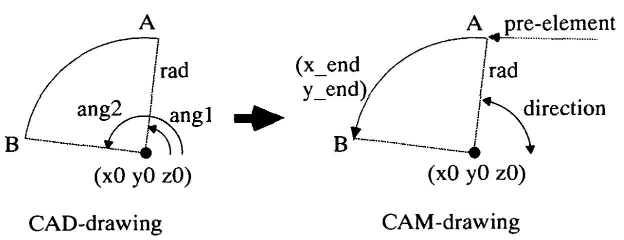


Fig. 9 The relation between CAD-drawing and CAM-drawing

Figure 10 is the flowchart of the process for generating arc elements of CAM-drawing from CAD-drawing.

As described in section 3. 2, the value of slot "direction" in the unit is substituted for variable "?direction" (see Fig. 10). On the other hand, in "%∗∗∗" in the figure, which shows the contents of CAD-drawing, the order corresponding to "∗∗∗" is incorporated, as the shape of slot value is stored in unit "cad". For example, "%x 0" in the figure is read and evaluated as "second" by the shape of the aforementioned CAD-drawing.

As a result, drawing data of the CAM system (CAM-drawing) can be generated based on CAD-drawing by using the structure model. Because of this, CAD-drawing can be used directly as CAM-drawing.

3. 3. 2 Process of the generation of shape macrocode Figure 11 shows the process of the generation of macrocode in the case of the circular arc. The figure in this example shows the case of referring to shape macrocode of slot "Base-Arc".

Now, let us explain the process in order of the numbers in the figure.

(1) Referring to attribution "cam" of slot "Base-Arc". In the case where "cam" is undefined, the process will be continued to step (2). However, in the case where it has already been defined, step (5) will be performed.

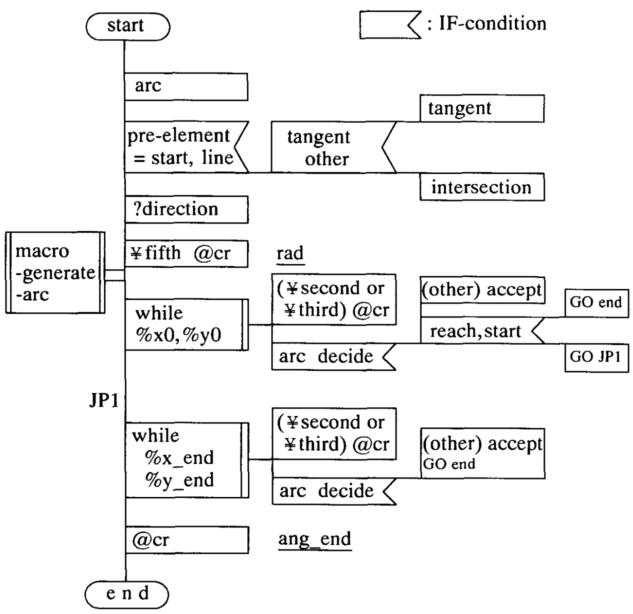


Fig. 10 The flowchart of the process to generate arc elements

(2) Referring to the value of slot "Base-Arc", then recognizing the line segments as an "arc".

(3) Executing generation function of shape macrocode corresponding to line segments (in this case the example is "macro-generate-arc") by using attached procedures.

(4) Substituting for attribution "cam" the results of the evaluation performed by the attached procedures. In this example, the following are substituted :

(cam (arc (2. cw) ¥fifth @cr ¥second
@cr accept @cr @cr @cr))

(5) Evaluating shape macrocode, then maintaining macro information to generate CAM-drawing.

Therefore, even in the case where the shape macrocode is undefined, by the use of the generation function described above, this problem can be treated easily.

3.4 Example

Figure 12 shows the execution of commercial CAD system (CADKEY) from AI tool (ZES)[6] and displaying cam drawing and hierarchy of the knowledge base. By using the "cad_recognition" command (see the figure), segments and dimensions of cam drawing which are recognized are maintained in inheritance units (∗ in the figure) of the hierarchy.

Figure 13 shows CAM-drawing which is generated based on CAD-drawing. The left part of the figure shows the lists of structure model representation

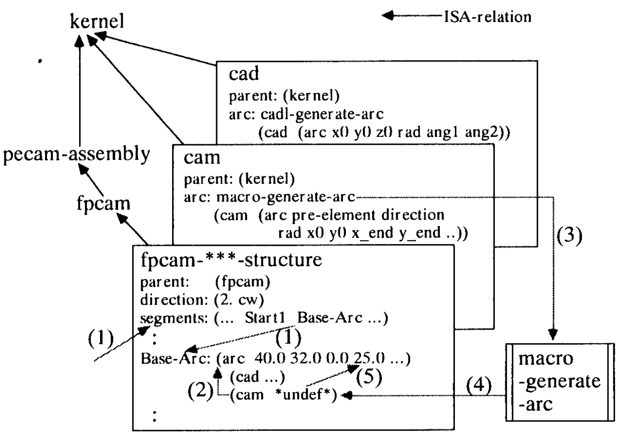


Fig. 11 The process of the generation of shape macrocode

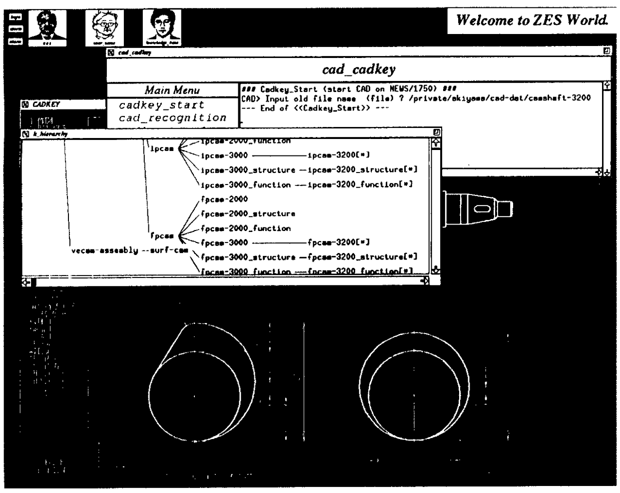


Fig. 12 The execution of recognition function of CAD-drawing

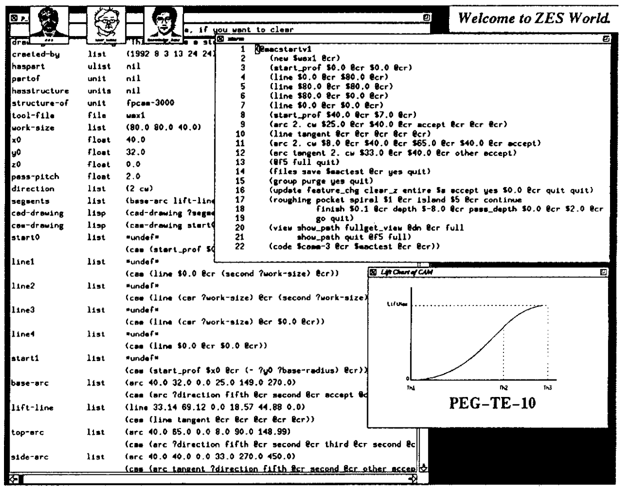


Fig. 13 Result of CAM-drawing generated based on CAD-drawing

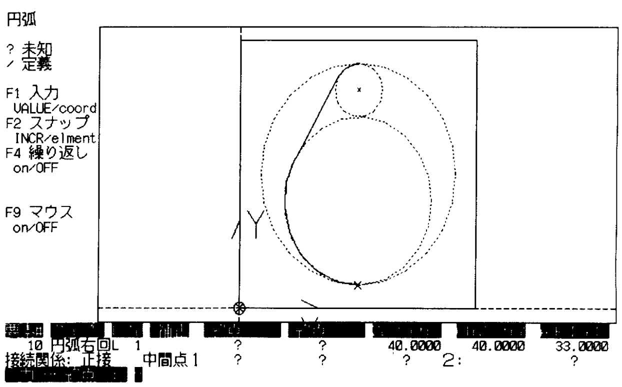


Fig. 14 An example of the execution of macrofunction (generating CAM drawing)

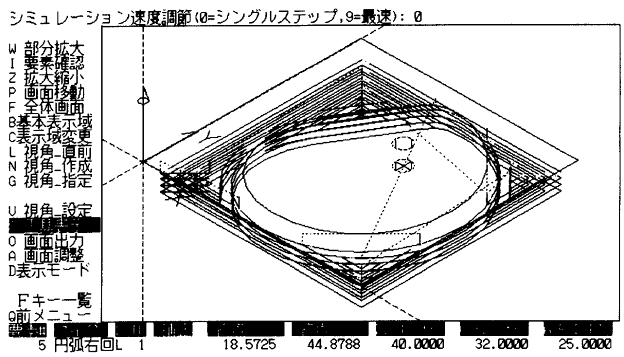


Fig. 15 An example of the execution of macrofunction (generating tool path)

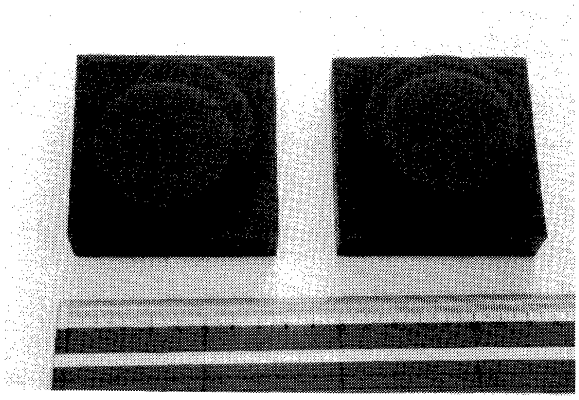


Fig. 16 Cams machined using macrocode

which maintained unit "fpcam-3000-structure". Here, the top right part of the figure is the list of macrocodes which is generated, while the bottom right of the figure shows the lift curve of function model representation.

Figures 14 and 15 show the execution of macro function, after macrocode has been transferred to CAM system (SmartCAM) in the personal computer. Here, Fig. 14 shows the step of the generation of an arc of CAM drawing, when the "arc" shape macrocode is executed. Fig. 15 shows the simulation of tool path (execution of step (9) in Fig. 8). Dotted line in the figure expresses movement of the tool.

Figure 16 shows objects (cam) machined using the macrocode which is generated based on CAD-drawing in Fig. 12, and using the macro function in Figs. 14 and 15.

4. Conclusions

The following conclusions are drawn from the present study:

(1) An object is represented by a structure and function in the knowledge base. In order to manage the relation definitely, the concept of the relationship between "master and servant" is specifically incorporated.

(2) By the use of the macrocode generation function, operations from transforming CAD-drawing to CAM-drawing, to the generation of NC code, which should be performed by the user can be simplified.

(3) When using the shape macrocode generation function, we can easily cope with mechanical drawing in a CAD system.

References

(1) Okino, N., Kakazu, Y. and Kubo, H., TIPS-1, Proc. of PROLAMAT '73 (IFIP/IFAC), Budapest, Computer Languages for Numerical Control, (1973), p. 141, North-Holland Pub. Co.

(2) Okino, N., Kakazu, Y. and Kubo, H., TIPS-2, Proc. of PROLAMAT '73 (IFIP) Striling, (1976), p. 385, North Holland Pub. Co.

(3) Nagasaka, Y., Mochiduki, Y., Ohtaki, H., Ishikawa, Y., Generation of Recognition-Rule for Shape-Understanding of CAD Drawings, Trans. Jpn. Soc. Design Eng. (in Japanese), 27-12, (1992), p. 537.

(4) Ueno, H., Densi Tsusin Gakkai Giho, AI 86-4 (in Japanese), (1986), p. 21.

(5) Nagasaka, Y., Akiyama, M., Ohtaki, H. and Ishikawa, Y., Managing System of Knowledge Base in Machining, Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 58, No. 552, C(1992), p. 2560.

(6) Nagasaka, Y., Ohtaki, H. and Ishikawa, Y., Management System of Knowledge Base for Mechanical Engineering based on Frame Representation, Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 58, No. 547, C(1992), p. 975.

(7) Doyle, J., A Truth Maintenance System, Artificial Intelligence, Vol. 12(1979), p. 231.

(8) Nagasaka, Y., Ohtaki, H. and Ishikawa, Y., Suitable Daemon Function for Building Knowledge Base in Mechanical Engineering Field, Trans. Jpn. Soc. Mech. Eng., (in Japanese), Vol. 58, No. 549, C(1992), p. 1680.