Proceedings of the 1999 IEEE
International Symposium on Assembly and Task Planning
Porto, Portugal - July 1999

PROCESS AND EXECUTION PLANNING FOR MANUFACTURING SYSTEMS

João Rocha*, Ana Moura*, Carlos Ramos*, Zita Vale+
**Departamento de Engenharia Informática* *+Departamento de Engenharia Electrotécnica*
Instituto Superior de Engenharia do Porto
Rua S. Tomé, 4200 PORTO, PORTUGAL
Email: jrocha@ipp.pt

Abstract

This paper deals with a system called TPMS (Task Planning for Manufacturing Systems). The main aim of this system is to automatically generate programs or sets of commands for robots, AGV's, numerical control machines, etc, without spending much time programming the flexible manufacturing system components.

A description of how to automatically generate a high level plan for the task will be presented as well as a methodology to automatically calculate the duration of each operation from the machine program.

The main contributions are the following ones: to link symbolic and execution planning; the referred aims at working for complete manufacturing systems (and not for controlling only one robot); different industrial operations (assembly, welding, drilling, ...) can be considered at the same time.

1. Introduction

Process planning involves the preparation of a plan to describe operations and their sequence, machines, tools ant the parameters needed to transform one or more components into a final product [1]. The automatization of process planning, named as CAPP (Computer Aided Process Planning), represents the link between design and manufacturing in a CAD/CAM system.

More global, complex and competitive manufacturing systems markets imply a need for frequently redesigning products and reprogramming. However, manufacturing systems reprogramming is a complex task involving a considerable set-up time. Most of this time is spent in defining how the new product will be manufactured and in programming the different machines of the manufacturing system (e.g. CNC, robots, AGV's, conveyors, ...).

Increasing of system's flexibility is very important to achieve efficiency and productivity improvements. Now the emphasis is on versatility and intelligence leading to ideas such as intelligent design and automatic planning.

This paper deals with task and execution planning for manufacturing systems. Task and execution planning involves two phases. In the first one a symbolic plan for the manufacturing task is generated. In the second one the symbolic operations of the high level plan are converted to sets of programs, instructions and commands to control the manufacturing system's equipment. For this purpose we are developing a system called TPMS (Task Planning for Manufacturing Systems) [2-4] that will be presented in this paper.

2. State of the Art and Background

Although several systems have been developed to deal with automatic task planning, they vary greatly in their domain of application and in the methods they employ.

SIPS [5] has as main objective the generation of the least cost plan for each operation (and not for the global system). Bourjault [6] has described an interactive system for assembly sequence generation of a product. A liaison graph representing parts connections is created. De Fazio [7] has improved this methodology allowing, the user to write logic formulas representing the sequence of operations. A liaison graph is also with each node representing the set of all components, assembled or not and each path, an assembly operation. Homem de Mello [8] uses AND/OR graphs to represent assemblies. With this kind of graphs, the number of nodes is drastically reduced, particularly for tightly coupled assemblies. In this case, each node represents an assembly, sub-assembly or individual component and each path an assembly operation. Wilson [9] has improved the efficiency of plan generation from AND/OR graphs using some kind of geometric reasoning. Now, only feasible sequences of operations are generated.

GARI [10], MACHINIST [11] and DAS [12] are systems that allow the generation of plans considering several setup procedures. DAS, for instance, looks for the least cost plan (where the least cost plan is the one with the fewest setups).

Some of the new trends in Manufacturing Systems are intelligence, agility and flexibility [1][1].

---

[1] Readers interested in a classification of Assembly Task Planning may consult reference [13].

0-7803-5704-3/99/$10.00©1999 IEEE

TPMS goal is to point into de direction of this trends, thus allowing the automatic generation of programs or sets of commands for robots, numerical control machines and other components of the system without the need of spending much time programming these components. It will be then possible to automatically obtain a plan to the industrial task and to manage the control code (program, instructions) of each machine and robot.

3. High Level Task Planning

3.1. Introduction

Task and Execution Planning involve two phases: first a high-level plan composed of symbolic operations is automatically generated and represented; then the programs for controlling the several machines of the Manufacturing System are generated.

The high-level planning makes it possible to generate a plan considering all processing, symbolic and geometric constraints. It is also possible to generate not only one, but several plans that could be dynamically activated according to the current situation.

"Planning refers to a deliberate behavior, a carefully weighted scheme for action towards some goal" [15].

Operators are the elements that can be used to change the state of the "world" and operations are instantiations of operators. Once an operation is applied the state of the parts' "world" changes.

Some operators are able to transform parts or change their shapes (e.g. drilling or milling) and others are able to join parts (e.g. welding). This means that after an operation we may have a transformation of the parts handled by the operation.

The general goal we want to achieve consists of a set of subgoals. We will use operators to achieve these subgoals. The sequence of operations for the task is the symbolic plan for the task.

However, the subgoals cannot always be achieved in any order. Often the accomplishment of some goals is restricted by constraints that represent interferences among subgoals (known as negative interactions).

For manufacturing tasks, the main constraints are the process constraints, feasibility constraints and geometric constraints. The constraints imply precedence relationships to guarantee that operations will be executed in the correct order. The syntax we will use is the following one: operation1 ***before*** operation2.

It is important to acquire knowledge about these constraints. Knowledge about feasibility and geometrical constraints can be automatically acquired with some geometric reasoning capabilities [140].

Feasibility constraints are detected by the verification of a set of feasibility constraint rules. An example of a rule of this type may be the following one: for insert(Ob, Ob1): Ob must be available; Ob1 must be available and the hole of Ob1 must be performed.

Geometric constraints are obtained by the analysis of the operation geometry. Collisions are a complex problem to deal with in Manufacturing Tasks. The geometric constraints related to the collision analysis can be automatically obtained. In the CIARC system, two kinds of constraints related to collisions are automatically acquired: constraints to avoid collisions between a robot and the objects and constraints to avoid collisions among objects being handled and other objects [14].

Once all constraints are determined is then possible to design the precedence graph where each node represents an operation and each path a constraint among operations. Since all the constraints are considered in the graph design, only feasible sequences of operations could be generated from the graph.

3.2. An example

Let us consider an example for making a pawn, as shown in Figure 1, starting with an aluminum cylinder. The surface of the cylinder must be uniformly adjusted and some facing and turning are necessary. By default, we do not know the order by which the operations must be done. The sequence of operations will be obtained by the feasibility and geometric constraints between parts and tools. There are also some processing constraints.

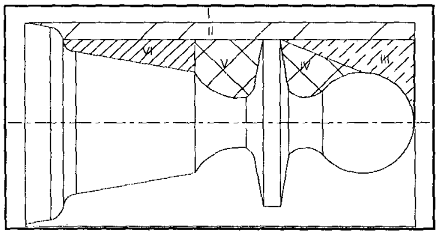


Figure 1 - An example

The operations are defined in Table 1. However, we do not know in which order they will be carried out. To know this, we will generate all feasibility, geometric and processing constraints. They are the following:

<u>processing constraints:</u>

For this type of material we must adjust the surface before starting turning. This means:

s_remove *before* turning(II, T03)

<u>geometric constraints:</u>

turning(II, T03) *before* facing(III, T03)
turning(II, T03) *before* facing(V, T01)
turning(II, T03) *before* turning(VI, T03)
facing(III, T03) *before* facing(IV, T01)

The precedence graph that is automatically generated by our planning methodology is shown in Figure 2.

The plan generation is achieved considering all the precedence relations (obtained from constraints), who can be represented by a graph.

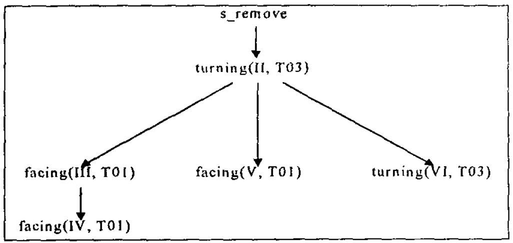


Figure 2 - Precedence graph

The generation of all plans can be very time-consuming for complex precedence graphs. To minimize this problem an algorithm was implemented, considering, for each operation, the time needed for concluding that operation as well as the time for changing from one operation to another. Then it will be possible to select the best plan considering the operation's times.

The cost for each node (corresponding to an operation) is obtained by adding the cost of the two components described above with the cost of the parent node. The cost of first node (node 0) is zero.

It is also necessary to estimate the merits of each node we generate in order to search more promising paths first. We do that by considering the sum of times for changing tools' and setup for all operations needed to obtain a plan. Since this is an underestimation of the total time needed to execute the plan, the algorithm is guaranteed to find an optimal (as defined before) plan with respect to time.

In the algorithm, the configuration of the manufacturing system is also considered, since the sequence of operations is important due to the time needed for changing from one operation to another.

Let us use the example above again. We must know the duration of all operations (Table 1), the times involved in tools changing and the setup time needed to move tools from one operation to another (Table 2). These values are automatically obtained as described in chapter 4.

Table 1 - Duration of the operations

N.	Operation	Duration (sec)
1	s_remove	60
2	turning(II, T03)	74
3	facing(III, T01)	161
4	facing(IV, T01)	122
5	facing(V, T01)	172
6	turning(VI, T03)	77

In Table 1 *na* means not applicable - i.e. it is not possible to make a transition from one operation to the other one, because some constraints apply.

To estimate the merits of the first node (considering times involved in the manufacturing process) we must sum the times for all operations. Since we have, at least, one tool change (we use tool *T01* <u>and</u> tool *T03*), it is possible to add also the time for one tool change operation (in order to obtain a more accurate estimate).

Table 2 - Setup times (from one operation to another)

	1	2	3	4	5	6
1 - s_remove	-	0	na	na	na	na
2 - turning(II, T03)	0	-	0	na	1	2
3 - facing(III, T01)	na	0	-	0	1	2
4 - facing(IV, T01)	na	na	0	-	0	1
5 - facing(V, T01)	na	1	1	0	-	0
6 - turning(VI, T03)	na	2	2	1	0	-

The time for one tool change operation should be added to all estimations until there is no absolute need for, at least, one tool change operation (although they can occur). Note that this is always an underestimate of the total time needed to completely manufacture the pawn, since there are, at least, some setup times (as in Table 2) to be considered.

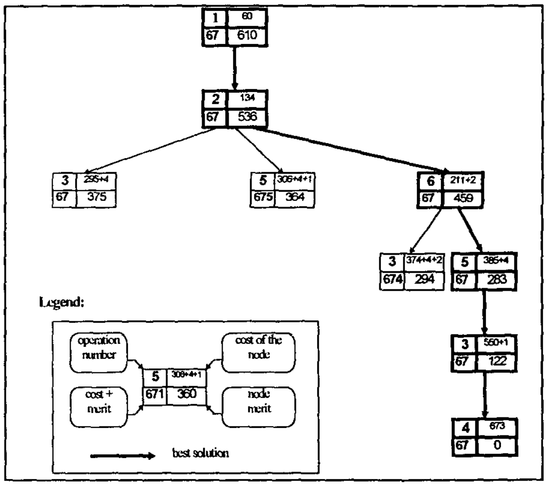
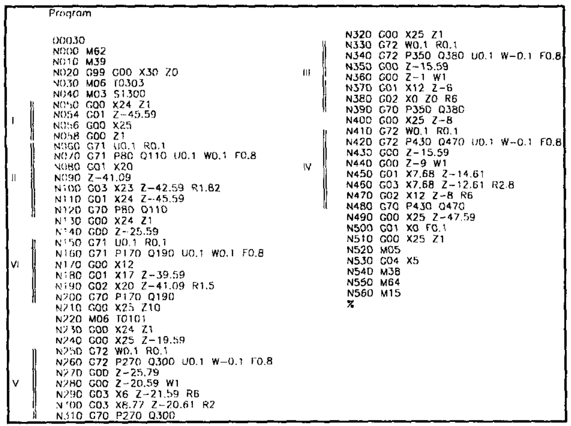


Figure 3 - Algorithm application

It is also possible to quickly generate one possible plan. In the example represented Figure 3, in the first expansion sequence, one solution is immediately achieved. However, it is possible (as seen before) to find a better solution.

Using the methodology described in the algorithm above (Figure 3) it is possible to generate an optimal plan, since all times involved in the operations, tools' changing and setup are considered.

In Figure 4, the program to build the pawn is shown (generated for a *GE FANUC SERIES OT* 1), corresponding to the best solution found.



```
O0030
N000 M62
N010 M39
N020 G99 G00 X30 Z0
N030 M06 T0303
N040 M03 S1300
N050 G00 X24 Z1
N054 G01 Z-45.59
N056 G00 X25
N058 G00 Z1
N060 G71 U0.1 R0.1
N070 G71 P80 Q110 U0.1 W0.1 F0.8
N080 G01 X20
N090 Z-41.09
N100 G03 X23 Z-42.59 R1.82
N110 G01 X24 Z-45.59
N120 G70 P80 Q110
N130 G00 X24 Z1
N140 G00 Z-25.59
N150 G71 U0.1 R0.1
N160 G71 P170 Q190 U0.1 W0.1 F0.8
N170 G00 X12
N180 G01 X17 Z-39.59
N190 G02 X20 Z-41.09 R1.5
N200 G70 P170 Q190
N210 G00 X25 Z10
N220 M06 T0101
N230 G00 X24 Z1
N240 G00 X25 Z-19.59
N250 G72 W0.1 R0.1
N260 G72 P270 Q300 U0.1 W-0.1 F0.8
N270 G00 Z-25.79
N280 G00 Z-20.59 W1
N290 G03 X6 Z-21.59 R6
N300 G03 X8.77 Z-20.61 R2
N310 G70 P270 Q300
N320 G00 X25 Z1
N330 G72 W0.1 R0.1
N340 G72 P350 Q380 U0.1 W-0.1 F0.8
N350 G00 Z-15.59
N360 G00 Z-1 W1
N370 G01 X12 Z-6
N380 G02 X0 Z0 R6
N390 G70 P350 Q380
N400 G00 X25 Z-8
N410 G72 W0.1 R0.1
N420 G72 P430 Q470 U0.1 W-0.1 F0.8
N430 G00 Z-15.59
N440 G00 Z-9 W1
N450 G01 X7.68 Z-14.61
N460 G03 X7.68 Z-12.61 R2.8
N470 G02 X12 Z-8 R6
N480 G70 P430 Q470
N490 G00 X25 Z-47.59
N500 G01 X0 F0.1
N510 G00 X25 Z1
N520 M05
N530 G04 X5
N540 M38
N550 M64
N560 M15
%
```

Figure 4 - Program for a *GE FANUC S. OT* lathe

4. Code generation and operations times

4.1 Introduction

As described in previous chapters, it is necessary to calculate the times of each operation. For this calculation two modules are involved: CEPP (Conversion and Execution of Process Plans) and ARTC (Areas Remove Time Calculation).

The first module generates the program for the execution of each operation on a specific machine. The second module calculates the time needed to execute the program on that machine.

4.2 CEPP

The module named CEPP allows the CAD/CAM integration using IGES and DXF file formats. These files allow swapping the information between CAD systems and applications and CAD/CAM systems, supplying information to the existing machines preparatory functions, such as linear, circular interpolations, facing and turning cycles. It can be said that the developed method involves three phases:

Phase 1 - Interpretation and handling of files generated from CAD software.

These files have information on the geometry of the part that it is intended to execute. The CAD generated information is represented in different levels, indicating the precedence between the areas to be removed. Once determined the areas to be removed, the process planner described before generates the plan, e.g. the sequence of the removal operations.

The CEPP module defines four different type areas (Figure 5). Each of these areas concerns to a different contour and machining orientation, due to predefined machining cycles. These cycles imply specific cut and orientation type, which must be correctly chosen by the programmers. After that, the CAD files are analyzed gathering the information in order to reorganize the vertices to simplify the CAD/CAM process.

Phase 2

According to the collected information, the machining cycles are automatically chosen and the cutting tools are automatically selected from a tool database.

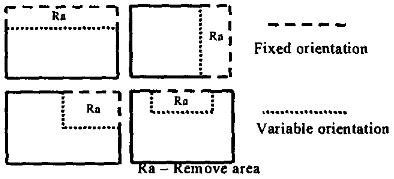


Figure 5 - Areas type treated by CEPP

Phase 3

The automatic generation of the program file is achieved during this phase.

Once CEPP process is applied, it is necessary to calculate the partial and total machining time, which concerns the machining process and the tool changes. This can be considered a fourth phase of the process.

4.3 ARTC

This calculation is a complex process hence some information needed (areas size and shape) is not available in the programs generated by CEPP. ARTC has been developed in order to gather all the related information from the CEPP generated files which applies a new methodology concerning these problems.

Depending on the type of areas, two situations come across:

1. If the area is a rectangle, square or triangle:

The time needed it can be easily determined, and is defined applying the formula (1) from mechanical engineering:

$$T = \frac{A}{Vc \times Pc \times a} \times 60[\text{sec}] \qquad (1)$$

- A= Area =H*L [mm^2] (2)
- Vc = Cutting speed [rpm]
- a = Tool feed rate [mm/rot]
- Pc = Depth of cut [mm]

2. If the area is not a rectangle, square or triangle (areas III, IV and V presented in Figure 8):

Then it is limited by a contour composed by curves and lines segments. In this case, to calculate the time needed for the operation we need to calculate the area with the equations of the part shape.

Figure 6 represents the general case, with line segments defined from points P1-Xn to P6-X1 and also the equation, $z = f(x)$:

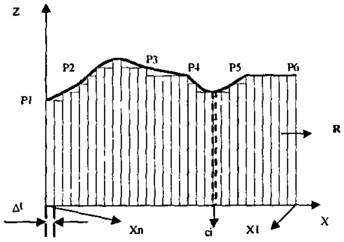


Figure 6 - Measure of the region area

From the definition: "Assuming that the function $f(x)$ of the curve is continuous in the closed interval [X1, Xn], with $f(x)=0$ for all the X in [X1, Xn] and that R is the region limited for the curve $z = f(x)$, the X axes and the lines X=X1 and X=Xn. We divide the interval [X1, Xn] in n ranges, each one of size Δt. Then, if $f(ci)$ is the minimum absolute value of the function in the range, the measure of the region area R will be given by:

$$A = \lim_{n \to \infty} \sum_{i=1}^{n} f(ci)\Delta t \text{ "} \qquad (3)$$

To calculate this area it is necessary to know the equation of its contour $z = f(x)$. However, in generic cases it is not simple to obtain the curve equation. Thus, the proposed method assumes the calculation dividing the machining contour part in several known and predefined part types.

The file generated by CEPP contains all the information needed, relating the profile type, line segment and curve type. The rectangular areas (Δt) presented at Figure 6, correspond to the tool steps defined in machining cycles by the depth of cut parameter (U).

Based on the initial point (P6) and the U parameter, all the tool steps are known, including the X co-ordinate in $f(x)$. The distance between X1 and the curve is given by the X axis perpendicular line, corresponding to the distance value of the cutting movement.

The time (T) required to remove this minimal area is obtained according to the distance between the curve and the axis, and the tool feed rate:

$$T = \frac{d}{F} \qquad (4)$$

- T = Machining time [*min*]
- d = incremental distance to remove [*mm*]
- F = Tool feed rate [*mm/min*]

The tool return time is given by:

$$Tr = \frac{T}{2} \qquad (5)$$

Tr is a fast movement and does not depend on the cutting speed. The sums of all partial times give us the machining time.

$$Ttotal = \sum_{j=1}^{n} (T_j + Tr_j) \qquad (6)$$

where n represents the tool steps, corresponding to the minimal removal area as presented on (7).

$$n = \frac{Xn - X0}{U} \qquad (7)$$

Figure 7 shows a little gray area that is not removed during the removal cycle. The tool moves only until it reaches the contour co-ordinate.

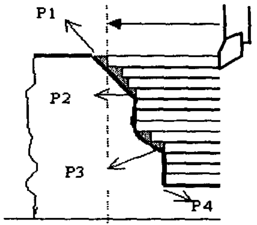


Figure 7 - Finishing

All gray areas are removed in the finishing cycle, done at high cutting and tool feed rates. Hence, the finishing cycle time has no significance in the overall machining process and can be disregarded.

At some points of the program we can find instructions indicating that a tool change operation has to be performed. The tool changing operation comprises three parameters (Figure 10):

- the movement to reference point return (Tz);
- the indexing time, corresponding to the turret rotation, from current tool to the requested one (Ti);
- the cutting positioning movement to the most unfavorable position with rapid traverse (Tc);

4.4 An example

Lets consider the example of Figure 8, where five different operations are defined.

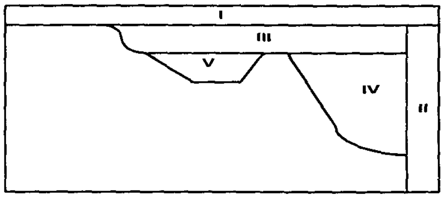


Figure 8 - An example

The NC code is parsed in order to obtain the needed information to perform the calculations.

The calculation of stock removal time (areas II and I) is made applying the formula (1) and (2). This areas are quadrilateral so is not needed the calculation of the part shape but only the area to be removed.

To calculate the part shapes areas removal times (areas III, IV and V) it is necessary to determine the equations of the lines and curves of the part contour.

Area III: The calculation referring this kind of area can be simplified.

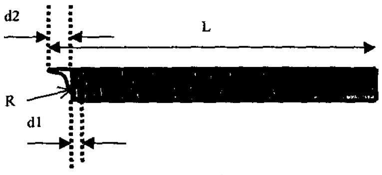


Figure 9 - Simplifying areas

The distance L (Figure 9) is much bigger than the distances d1 and d2. Although the tool tip does the specified contour, the time consumed in these distances is much smaller compared with the time needed in L and it is considered irrelevant to the total time. Thus, to simplify the calculations, this area type is considered as a rectangle and calculated as areas II and I.

Areas IV and *V*: These areas comprise lines and curves, so it is necessary to calculate these geometric form equations to obtain all Z values along the shape. Reading the program file, the following equations are obtained:

With the contour equations and the formulas (4), (5) and (6), the two areas machining times are calculated. In addition, it is necessary to calculate the tool changing operations time (**T1** - tool 1 and **T2** - tool 2).

T1: Since the time needed to perform this operation depends on the previous program executed, it is impossible to calculate it exactly. Nevertheless, it possible to consider the worst case:

$Tct = 7 * Ti + Tc$ (initially, the tool is in the zero position)

T2: $Tct = Tz + 2 * Ti + Tc$

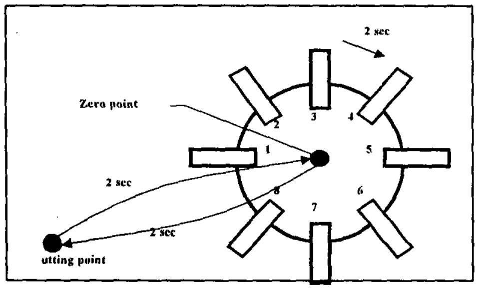


Figure 10 – Tool change operation times

5. Conclusions

This paper has described and illustrated TPMS (Task Planning for Manufacturing Systems). The main contributions of TPMS are the following ones:

- to put together symbolic planning and execution planning;
- address problems of realistic Manufacturing Systems (and not for controlling only one robot);
- to be able to automatically calculate the times of the operations involved;
- to be able to automatically generate the program to a set of manufacturing machines.

References

[1] A. Kusiak. *Intelligent Manufacturing Systems*, Prentice-Hall International Series in Industrial and Systems Engineering, 1990

[2] J. Rocha and C. Ramos. Task and Execution Planning for Flexible Manufacturing Systems. In *Proc. of. the Second International Conference on Integrated Logistics and Concurrent Engineering - ILCE'94*, pages 131-139, 1994

[3] J. Rocha and C. Ramos. Task Planning for Flexible and Agile Manufacturing Systems. In *Proc. of IEEE International Conference on Intelligent Robots and Systems - IROS'94*. pages 105-112, 1994

[4] J. Rocha, and C. Ramos. Plan Representation and Generation for Manufacturing Tasks. In *Proc.* of *International Symposium on Assembly and Task Planning - ISATP'95*, pages 22-27, August 1995

[5] D. Nau. Automated Process Planning Using Hierarchical Abstraction.. In *Texas Instruments call for papers on AI for Industrial Automation*, 1987

[6] A. Bourjault. *Contribution a une approche méthodologique de l'assemblage automatisé: Elaboration automatique des séquences opératoires*. Thèse d'État, Univ. de Franche-Comté, 1984

[7] T. De Fazio and D. Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation*, 3(6): 640-658, 1987

[8] L. Homem de Mello. *Task sequence planning for robotic assembly*. PhD Dissertation, Carnegie Mellon University, 1989

[9] R. Wilson. *On geometric assembly planning*. PhD Dissertation, Stanford University, Rep. n. STAN-CS-92-1416, 1992

[10] Y. Descotte, J Latombe. Gari: A Problem Solver that Plans How to Machine Mechanical Parts. In *Proc. of IJCAI*, pages 766-772, 1981

[11] C. C. Hayes. MACHINIST: A Process Planner for Manufacturability Analysis, 1995

[12] D. Das, S. Gupta and D. Nau. Reducing cost set-up by automated generation of redesign suggestions. In *Proc. ASME Comp. in Engineering Conference*, 1994

[13] S. Gottschlich, C. Ramos and D. Lyons. Assembly and Task Planning: A Taxonomy. IEEE Robotics and Automation Magazine, 1(3): 4-12, 1994

[14] C. Ramos and E. Oliveira. CIARC: A Multi-Agent Community for Intelligent Assembly Robotics Systems with Real-Time Capabilities. In *Proc. of International Conference on Systems, Man and Cybernetics*, 1993

[15] M. Ghallab and A. Milani. *New Directions in AI Planning*, IOS Press, pp. V., Frontiers in Artificial Intelligence and Applications, vol. 31, 1996