

A LOGIC REPRESENTATION OF MANUFACTURING CONTROL

William I. Bullers
Krannert School of Management
Purdue University
West Lafayette, IN 47907

Shimon Y. Nof
School of Industrial
Engineering
Purdue University
West Lafayette, IN 47907

Andrew B. Whinston
Krannert School of Management
and Computer Science Dept.
Purdue University
West Lafayette, IN 47907

This paper explores some of the typical problems in manufacturing systems planning and control, particularly those pertinent to automatic operation, and describes how artificial intelligence methods can be applied. We demonstrate how predicate logic and theorem-proving techniques using resolution can be used in a manufacturing environment. Assertions of fact and axioms representing the knowledge required are given in an underlying data base. Illustrative examples demonstrate how user problems, such as assignment of jobs to machines when conflicts occur, can be handled by a decision support system in the framework of resolution in a problem-reduction approach.

1. Introduction

Manufacturing planning and control involves managerial decision-making in three main levels of activities: the strategic level of production and inventory planning; the tactical level of manufacturing operations; and the operational control of processes and material flows. Computerized manufacturing systems (CMS), characterized by multi-purpose, computer-controlled machines connected by automated material handling networks, require a large number of timely decisions to be made at the various levels of operations. See [9] for the information and control aspects of such systems. The purpose of this paper is first to explore some typical manufacturing planning and control problems, particularly those pertinent to the automatic environment, and second, to describe how predicate logic problem-solving methods can be applied to them.

2. Predicate Logic Representation of Knowledge

First-order predicate logic has been used extensively as a knowledge representation language to describe states and problems. Moreover, as shown in [4], predicate logic comprises a programming language for expressing algorithms. Axioms in Horn clause format are interpreted as procedures to be invoked by theorem-proving mechanisms based on Robinson's resolution principle [8]. Deductive query systems such as Chang's [2], and Minker's [5] combine predicate logic with a data base system for knowledge storage and retrieval. These systems use re-

solution to partially prove theorems, then invoke the data base system to complete the proof. In this paper we demonstrate how a predicate logic representation of knowledge integrated with a data base can be used to model control in a manufacturing environment for the purpose of solving operations management problems.

3. Manufacturing Planning and Control Needs

In the three levels of manufacturing activities previously defined, planning and control are required in order to achieve efficient, productive utilization of resources to meet a schedule of production demands. Strategic and tactical planning and control have been performed quite successfully with the aid of information systems. It is in the operational level of control in which a real-time decision support interface between managers and machines is lacking, particularly in the CMS environment.

Three operational control issues described in [7] include Part Mix, Part Entry and Assignment, and Process Selection. The Part Mix problem is to select particular sets of parts for sub-periods from the larger set of parts scheduled for production in the whole period. The Part Entry and Assignment problem is to sequence individual part entry into the system and assign parts to machines when conflicts occur. The Process Selection problem is to decide which of several alternative processes, all producing the same part but requiring different machines and processing times, should be selected for a given

part.

While off-line analyses using simulation indicate the importance of such control issues, a different approach is necessary when decisions have to be made and implemented in a real-time environment. In the following sections a new approach is developed, and examples given to illustrate how some of the issues described here are handled.

4. Predicate Logic Representation of Manufacturing Control

CMS entities and simple relationships among them are modeled by predicates, which are represented in a network data base system as described in [1]. Complex relationships among the entities are defined by axioms. The axioms extend the data base by supplying procedural knowledge about how the assertional component is used for control tasks.

Many manufacturing control issues require the system status at a given time. Typical inquiries concern what processes are available for each part, what conflicts exist among parts competing for machine servicing, etc. Queries of this type can be handled by problem-solving techniques utilizing a static representation. Generation of the current state of a CMS is done concurrently with actual operation of the manufacturing system by an "operational" mode of the decision support system. Problem-solving on the "operational data base" is performed using a problem-reduction approach [6], in which a problem is given by:

- (1) a predicate logic goal statement describing the problem to be solved;
- (2) descriptions of problems whose solution is known, primitive problems;
- (3) a set of Horn clause axioms, problem-reduction operators, which transform problems into sub-problems.

A top-down proof procedure is applied to carry out resolution, deriving new goal statements from old ones by applying the problem-reduction operators. The primitive problems obtained are then solved by retrieval procedures that evaluate predicates representing assertions in the manufacturing data base.

Control issues in a dynamic domain require a "simulation" mode of the decision support system to forecast future states of the CMS. The predicate $TIME(t)$ is used to denote time, and a state-space approach [6] is used to update the system. The representation of a problem consists of:

- (1) the operational data base of assertions denoting the initial state;
- (2) a predicate logic statement of a goal state;

- (3) a set of Horn clause axioms, state-space operators, to transform states into other states.

A bottom-up proof procedure is applied to derive new assertions from old ones. The operators utilize an ADD and DELETE list similar to that used in STRIPS [3] to handle secondary implications of an operation in addition to the consequent in the Horn Clause axiom. The assertions generated for the "simulation data base" are then used as newly solved primitive problems in a top-down theorem-proof. Comparison of criteria across different simulated states determines operational parameters that should be used in the CMS operation over the forecast horizon.

5. Illustrative Manufacturing Control Problems

Table 1 shows a partial list of predicates needed to represent the machines, parts, operations, etc., and their relationships in a CMS.

Table 1

Computerized Manufacturing System Predicates

Predicates	Interpretation
MCH(mch)	machine "mch"
OPERATION(op)	operation "op"
PART(part,part-type)	part
COMPLETE-PART-OP (part,op,t)	part/op complete at t
MCH-OP(mch,op,op-time)	mch/op process time
PART-ROUTING (part-type,op,next-op)	part routing

The complex relationships are specified by axioms. Problem-reduction operators for the Part Assignment problem determine (1) if the first operation for a part is next to be performed, (2) if a subsequent operation is next, and (3) a possible schedule of a part to a machine for an operation. These three axioms are defined as:

- $$\begin{aligned}
 (A1) \quad & \forall x \forall y \forall z \forall w \\
 & \text{NEXT-PART-OP}(\text{part}=x, \text{op}=y) \leftarrow \\
 & \quad \text{PART}(\text{part}=x, \text{part-type}=z), \\
 & \quad \text{OPERATION}(\text{op}=y), \\
 & \quad \text{PART-ROUTING}(\text{part-type}=z, \text{op}=y), \\
 & \quad \sim \text{PART-ROUTING}(\text{part-type}=z, \text{op}=w, \text{next-op}=y) \\
 & \quad \sim \text{COMPLETE-PART-OP}(\text{part}=x, \text{op}=y) \\
 (A2) \quad & \forall x \forall y \forall z \exists w \\
 & \text{NEXT-PART-OP}(\text{part}=x, \text{op}=y) \leftarrow \\
 & \quad \text{PART}(\text{part}=x, \text{part-type}=z), \\
 & \quad \text{OPERATION}(\text{op}=w), \\
 & \quad \text{OPERATION}(\text{op}=y), \\
 & \quad \text{PART-ROUTING}(\text{part-type}=z, \text{op}=w, \text{next-op}=y) \\
 & \quad \text{PART-ROUTING}(\text{part-type}=x, \text{op}=y), \\
 & \quad \text{COMPLETE-PART-OP}(\text{part}=x, \text{op}=w), \\
 & \quad \sim \text{COMPLETE-PART-OP}(\text{part}=x, \text{op}=y) \\
 (A3) \quad & \forall x \forall y \forall z \\
 & \text{POSSIBLE-SCHEDULE}(\text{part}=x, \text{mch}=y, \text{op}=z) \leftarrow \\
 & \quad \text{NEXT-PART-OP}(\text{part}=x, \text{op}=z), \\
 & \quad \text{MCH-OP}(\text{mch}=y, \text{op}=z)
 \end{aligned}$$

With these axioms, we can solve a Part Assignment problem such as: "On what machine can part #p2 be scheduled for its next operation?" This problem, expressed as the goal statement:

$\exists y \exists z$ POSSIBLE-SCHEDULE(part=#p2,mch=y,op=z)

is reduced to a set of primitive problems by invoking axiom A3, then A1 or A2. A unique "y,z" pair satisfying the primitive problems solves the Part Assignment problem. For multiple "y,z" occurrences, additional axioms specify the particular machine-operation pair to be chosen.

For simulation, a set of state-space operators describe the operation of the manufacturing system. For example, the end of a machine's operation performance is defined by:

```
(A4)  $\forall x \forall y \forall z \exists t_1 \exists t_2 \exists t_3$ 
      END-MCH-OP(mch=x,part=y,op=z,t=t3) ←
      INIT-MCH-OP(mch=x,part=y,op=z,t=t1),
      MCH-OP(mch=x,op=z,op-time=t2),
      ADD(addend1=t1,addend2=t2,sum=t3),
      delete list: INIT-MCH-OP(mch=x,part=y,op=z)
      CURRENT-PART-OP(part=y,op=z)
      add list: MCH-IDLE(mch=x),
      COMPLETE-PART-OP(part=y,op=z,t=t3)
```

Using such axioms, we solve a problem useful in Process Selection: "Given a shortest processing time scheduling rule, when will part #p2 complete its next operation?" The goal statement:

$\exists z \exists w$ NEXT-PART-OP(part=#p2,op=z),
COMPLETE-PART-OP(part=#p2,op=z,t=w)

is solved by first invoking the simulation mode of the decision support system. A bottom-up theorem-proof derives an assertion of the form:

COMPLETE-PART-OP(part=#p2,op=#op_z,t=#t_w)

in the simulation data base. Axiom A1 or A2 is then invoked in a top-down manner to reduce the NEXT-PART-OP sub-problem. Finally, the primitive problems are solved by data base retrieval. Simulating the completion time for a number of alternative scheduling rules enables us to select the rule that provides the best value of a criterion measure of performance for completion of a number of part operations over the simulated time horizon.

6. Conclusions

In this paper we have explored the application of predicate logic and mechanical theorem-proving to the management of a computerized manufacturing system. Predicate logic permits a descriptive non-procedural representation of knowledge related to the operation of the facility. The predicate calculus language allows both static and dynamically changing knowledge to be modeled, while theorem-proving permits

"new" knowledge to be deduced from the current knowledge base. From a decision support viewpoint, the resolution procedure allows the knowledge base to be applied to a user's specific manufacturing control problems.

Further work is needed to implement such a system. Questions on how axioms should be stored and how resolution can be effectively carried out must be investigated. From a user's point of view, a more natural, English-like problem statement language is needed. In addition, relationships between the problem-solving techniques described and classical integer programming models of scheduling would be of interest.

ACKNOWLEDGMENTS

This work supported in part by the National Science Foundation (grant MCS 76-24675 for Computerized Decision Support Systems). Mr. W. Bullers gratefully acknowledges the fellowship support made possible by a grant from IBM.

REFERENCES

- [1] Bonczek, R., Holsapple, C. and Whinston, A., "The Integration of Network Data Base Management and Problem Resolution", (to appear in Information Systems).
- [2] Chang, C.L., "DEDUCE 2: Further Investigations of Deduction in Relational Data Bases", IBM Research Report RJ 2147 (29410), May, 1978.
- [3] Fikes, R.E. and Nilsson, N.J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", Artif. Intell. 2, 1971, p. 189-208.
- [4] Kowalski, R., "Predicate Logic as Programming Language", Proc. IFIP 74, 1974, p. 569-574.
- [5] Minker, J., "An Experimental Relational Data Base System Based on Logic", Logic and Data Bases, (Gallaire, H. and Minker, J., eds.), Plenum Press, New York, 1978, p. 107-147.
- [6] Nilsson, N.J., Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, N.Y., 1971.
- [7] Nof, S.Y., Barash, M.M. and Solberg, J.J., "Operational Control of Item Flow in Versatile Manufacturing Systems", Proc. 5th Int'l. Conf. on Production Research, 13-16 August, 1979, Amsterdam, The Netherlands.
- [8] Robinson, J.A., "A Machine-Oriented Logic Based on the Resolution Principle", Journal of The ACM, 12:1, January 1965, p. 23-41.
- [9] Talavage, J.J. and Barash, M.M., "Information and Control in Computerized Manufacturing Systems", Proc. Int'l. Fed. of Automatic Control Conf., Tokyo, Japan, October, 1977.