Published in: Working Notes, Symposium on "AI & Manufacturing",
American Association for Artificial Intelligence,
28-30 March 1989, Stanford University

OBJECT-ORIENTED MANAGEMENT PLANNING SYSTEMS FOR ADVANCED MANUFACTURING

D. H. Norrie*+, O.R. Fauvel*, B. R. Gaines+

Department of Mechanical Engineering (*)
Department of Computer Science (+)
The University of Calgary,
2500 University Drive,
Calgary, Alberta, Canada.

ABSTRACT

Two Integrated Management Planning Systems to integrate and implement process planning, cell grouping, scheduling, simulation and related functions are described. One (FLEXES) is based on a consortium of knowledge-based systems utilizing expert system, database and simulation techniques. The other (OBJIMPS) comprises an interacting system of part objects, machine objects and other 'physical' objects together with the necessary 'management' objects for supervision planning and associated functions.

INTRODUCTION

The research to be described began in late 1986 with Gaines' observation [1] that the complex information flows within an integrated manufacturing system could be processed using 'knowledge islands' each of which carries out a particular function (such as machine tool grouping, simulation or operation procedures) or stores certain data (such as parts specification, material properties or machining parameters). It was subsequently recognized that the function of the knowledge islands could be carried out by knowledge-based systems. An overall system for design and management of computer-integrated manufacturing could thus be a 'consortium' of knowledge-based systems each carrying out particular functions and interacting with the others according to need. By 1987, a particular design and management system was under investigation, namely a decision support system for flexible manufacturing planning (FLEXES), using such a structure.

An alternative strategy is more radical and more fundamentally object-based. Instead of a module simply processing data *about* objects such as parts and machines, the module would comprise a group of specialised objects which would interact with part objects, machine objects and objects of other types. That is, the whole 'Integrated Manufacturing Planning System' would be an interacting set of planning objects, part objects, and machine objects together with any necessary supervisory and other objects. This strategy is being worked out for the Object-Oriented Integrated Management Planning System (OBJIMPS) described subsequently.

THE FLEXES PROJECT

A decision support system (FLEXES) is being developed for the planning of flexible manufacturing (FMS) systems using a combination of expert system, database and simulation techniques. The need for such FMS management support systems is well documented [2-4]. Expert systems are suited to the complex procedures and heuristics of these systems [5-7] but require integration not only with databases to handle the volume of information used but also with special purpose packages such as simulators for specialized tasks [8]. FLEXES would interact with the management of an integrated manufacturing system at the equipment procurement stage to specify needed equipment, at the system design stage for machine selection, grouping and layout, and at the system operation stage for process scheduling and operation. There are six basic modules: - a machine selection optimizer; a database of machine parameters and capabilities; a production flow analyzer; a rapid modelling techniques system; a simulator for flexible manufacturing system (FMS) configurations; and an FMS expert system supervisory and coordinator shell.

A study of the foundational information systems has indicated [1] a granularity in manufacturing knowledge from macro-level ('rough') to micro-level ('detailed') This leads to a guiding design principle: "First structure the system according to the principal factors, then configure the detailed structure by progressively incorporating the factors of decreasing significance". FLEXES is intended to not only configure the manufacturing system (macro-level) but also to provide normative plans for process scheduling and operation (intermediary level). To assist day to day scheduling and operation (detailed level), a computer aided process planning (CAPP) module would subsequently be added to FLEXES. Briefly stated, the input to FLEXES

would be product data such as number of parts, their shapes and tolerances and the outputs would include listing of machine tools/robots/AGV's required, flexible cell configurations, plant layout, and basic machine schedules. Capital operating cost data would be also generated, as well as production rates. A more detailed description of FLEXES and its operation is given in [9].

Machine Selection Optimizer

The machine selection optimizer is designed to operate in the following way. For each product part, each of its 'features' (prismatic face, cylindrical surface, hole, etc.) would be specified to the system and stored in the database. The selection expert system then uses previously defined rule sets to determine possible production processes (or sequences thereof) which could be used to produce each feature. It next determines which machine tools could be used for each process, taking into account also the size, shape, materials and tolerances of the part. At this point, heuristics are used to group together the most appropriate set of machines for producing all features of all parts. Among the heuristics being considered are: fewest number of different processes (saves rejigging, setup, transfer); fewest number of different machines (reduces plant training and maintenance costs); capital cost of multiple-use machines versus single-use machines. The optimizer is being developed using the object-oriented expert system NEXPERT together with the DBASE 3+ relational data base.

Feature-Based Approach

The knowledge representation system chosen to symbolically represent the shape of a part is critically important for the design of the machine selection optimizer as well as for the other modules of FLEXES. The features approach is used in FLEXES, where examples of simple features are surface_of_revolution, prismatic_surface, complex_surface and cylindrical_hole. Such a feature-based representation is suited naturally to being embodied in an object-oriented representation [10], where each feature is an object in a larger universe of objects.

Production Flow Analyzer

The output from the selection optimizer includes for each part the set of machines and the process sequence required for that part. This output is stored in the database and becomes a major portion of the input to the production flow analyzer. This module uses group technology (techniques such as the PFA matrix method) to group together those parts that can be produced by a common set of machines. Such a set of machines is then used as the basis for a flexible machining cell (FMC). The output from the production flow analyzer comprises the requisite FMC's and the associated parts routing.

Rapid Modelling Unit

The FMC's derived by the production flow analyzer need to be balanced for machine loading (this may require substitution of one or more machines by a unit with higher or lower production rate or the specification of multiple identical units). These tasks are carried out by the rapid modelling unit, using the recently developed Rapid Modelling Technique (RMT) [11]. RMT relies on mathematical modelling using recent advances in the theories of queuing networks and reliability modelling. For at least a portion of the needed capability in the rapid modelling module, available RMT implementations such as Manuplan II will be investigated.

Simulation Module

The output from the RMT module comprises essentially balanced FMC's and remaining stand-alone units. The simulation module is now used for the problem of how to integrate these cells and machines into a complete plant, with the inclusion of transfer vehicles or systems. For the simulator, SIMSCRIPT could be used, taking input from the modeller and the database, with graphic output of the FMS operation and performance being obtained by SIMFACTORY. As alternatives, SIMAN and the use of SMALLTALK will be explored.

Database

A relational data base such as DBASE can be used as the current work demonstrates but an object-based system would be advantageous. Object-oriented databases (GEMSTONE, G-BASE) will therefore be evaluated.

Supervisor

The supervisory shell could be based on a relatively simple expert system such as EXSYS or a more sophisticated one such as NEXPERT or GURU. With EXSYS, the frames capability would be an advantage, whereas with NEXPERT its object orientation would be particularly valuable. Programming the shell in SMALLTALK with the Xerox HUMBLE expert system on top of SMALLTALK would give highly desirable flexibility.

Implementation

The design of flexible manufacturing systems using artificial intelligence techniques is in its infancy, although several expert systems for process planning have been developed [12]. A prototypical system for analyzing the output from a FMS simulation and optimizing the configuration has recently been described [13] and demonstrates the advantages of these approaches. FLEXES is much broader in scope, is designed using a foundational theoretic paradigm, and incorporates a number of new approaches particularly in the machine selection and associated modules.

By early 1988, the conceptual development had progressed to the stage where detailed development was begun. The machine selection optimizer was envisaged as being the most difficult and so was chosen for the initial development. This module considers each part in turn and from its features determines successively the set of possible manufacturing process sequences and the minimal set of specific machines needed. The feature 'cylindrical_hole' was selected as being representative and a sub-system developed which implemented the machine selection optimizer function for this feature. The sub-system was programmed using the NEXPERT object-oriented expert system interfaced with the DBASE 3+ relational database. There were 37 rules required, totalling 500 lines of code. The DBASE programs, which were used for both processing and data storage/retrieval, numbered 110 comprising 5,000 lines of code in total. The sub- system has been developed, tested and evaluated. For details of the structure of the machine selection optimizer, the algorithms for process selection and machine-set minimization, and programming approaches see [9].

THE OBJIMPS PROJECT

OBJIMPS is envisaged as a complete interactive system comprised of modules which carry out process planning, cell grouping, scheduling, simulation and other functions, with each module having its own 'rules' or expertise embedded within it. The user interacts through a supervisory module with the process planning, cellgrouping, scheduling, simulation and other modules which also interact with each other through their submodules. Each module, submodule or other component of the system is an object, with interaction by messages sent from object to object. The whole would appear to the user as an 'intelligent' system, which for convenience is called a "manufacturing planner". At a later stage, a design system would be interfaced with this planner to facilitate integration of design and manufacturing planning and to allow part designs to be downloaded. As well as the part geometry, material tolerance, surface finish and so on, the part description in terms of features would be included in the design since there is a strong correlation between a part feature (prismatic surface, hole, etc) and the manufacturing operation required for it.

Structural Approaches

For the process planning and cell grouping modules currently being worked on, two approaches to structuring these modules into objects are being considered. The first has objects relating to the functional subdivisions used in FLEXES. An alternative functional approach under consideration would have the process planning and cell grouping objects carry out more conventional functions:-

Process Planning: This module would determine for a part the operations to be carried out, which machines are suitable for each operation, fixtures and tools needed, operation times and so on.

Cell Grouping: The procedure for cellgrouping would include setting up a parts-machines matrix, processing this matrix to show potential machine clusters for part groups, and defining the cells from the machine clusters taking other relevant factors into consideration.

The Database

For the database in OBJIMPS, there are at least three possibilities: (a) the object-oriented language's own capability for storing data (e.g. in Smalltalk, using arrays for data storage), (b) a relational data base manipulating tables or arrays of data stored and (c) a fully object-oriented data base.

Generality, Particularity and Inheritance

For a class of objects, such as CNC_Machine, the general properties and capabilities will be defined at the Class level and instances of the class may inherit these. Every effort will be made to retain generality as far down through the system levels as possible and to introduce particularities at the lowest levels practical. One cogent reason for doing this is to allow new objects such as planners, schedulers, machines and AGVs to be later introduced into the system with the minimum disturbance (i.e. changes) to the higher levels.

The Object Knowledge Base and Inferencing

The same principle of specifying the particularities at the lowest level possible will be used in structuring individual objects. In general, within each object, the specific knowledge (data) for the object will be separated into a facts sub-object (e.g. an Array) and a rules sub-object (containing the specific rules (procedures) for the object's actions). Together, the facts and rules (data and procedures) would comprise the knowledge base in the expert system sense. What corresponds to the inference engine code would be elsewhere within the object (for a very specific kind of interpretation) or inherited from a higher level (for more general kinds of interpretation or inferencing). To illustrate, an instance of a CNC_Machine could have facts such as manufacturer, model number, maximum horsepower and maximum height of a part, stored in anArray and rules such as [(IF) 'spindle is set to on' (AND) 'part program begins' (THEN) 'close safety gate'] stored in some other convenient data structure. Since the rule set may simply require consecutive execution, the 'inference engine' may only be a sequential line-by-line interpreter. In contrast, a process planning object may have a backward-chaining inference engine which makes use of the whole knowledge base of the

object i.e. both facts and rules. It will be seen that each object can have, in effect, its own expert system incorporated for 'deciding' how to act or react. The advantage is that not only each object can have its 'expert system' embedded, as simple or sophisticated as needed, but that this capability can be inherited from a higher class. To change the characteristics of a particular object, may require only an appropriate change to a single knowledge base.

Simulation

In an object-oriented simulation, each factory component such as a machine tool or automated guided vehicle (AGV) would be an object which stores data about itself and responds to messages from other objects. The whole simulation therefore would comprise an interactive process by which parts, machines, AGVs, planners, schedulers, and so on interact with each other through messages in a sequence or order depending on individual conditions. For example, if a machine tool found that a part was prismatic it would send a message to a particular scheduler but if the part was rotational a different scheduler would be addressed. A supervisory object could be addressed to request higher level instructions e.g. "Have broken cutter, what to do now?"

The simulation process envisaged would be dynamic with more similarities to an ecological system than to a hierarchical model. A key advantage to the object-oriented approach, especially where a language such as SMALLTALK is used, is that new objects and even whole classes of new objects can be grafted into the simulation with minimum disruption to the previously existing system. This offers the possibility of beginning with a small system covering say, process planning, and progressively extending it over all aspects of a factory. The ultimate aim is to simulate a complete factory system e.g. an existing plant (large or small) or a factory-of-the-future. The initial simulations would not be described as intelligent but since adaptability (using, for example, rules in a knowledge base) could be incorporated into objects, such simulations might well be precursors of later 'intelligent' systems. While current simulations would run on a single CPU computer, future ones will run on parallel processing systems. New developments in the 'virtual time' concept (References 14, 15, 16) offer attractive possibilities in parallel processing for object-oriented simulations.

To illustrate the object-oriented approach, Fig. 1 is presented. This shows the 'report' from a CNC machine object (a milling centre) after being sent a scheduler 'message' with instructions to use numbered part programs (e.g. 'rough cut' 4, 'fine cut' 8) for machining a particular part. This figure is taken from a Smalltalk-based CNC simulation designed as an exploration of how to handle part programs, schedules and graphic data.

MACHINE: Vertical CNC Milling Machine
MADE IN: MillMan Corporation
MODEL: Mc-23A
SCHEDULER PROGRAM: ('setup' 'roughcut' 4 'toolchange' 2 'finecut' 8 'toolchange' 3 'finecut' 5 'unload' 'setup' 'roughcut' 6 'toolchange' 1 'finecut' 11 'unload')
MACHINING COMPLETED.

TOTAL ROUGHCUT TIME:	143 minutes
TOTAL FINECUT TIME:	99 minutes
TOTAL SETUP TIME:	64 minutes
TOTAL TOOLCHANGE TIME:	5 minutes
TOTAL UNLOAD TIME:	6 minutes
TOTAL MACHINGING TIME:	317 minutes

Figure 1(a) - Simulation output (text)

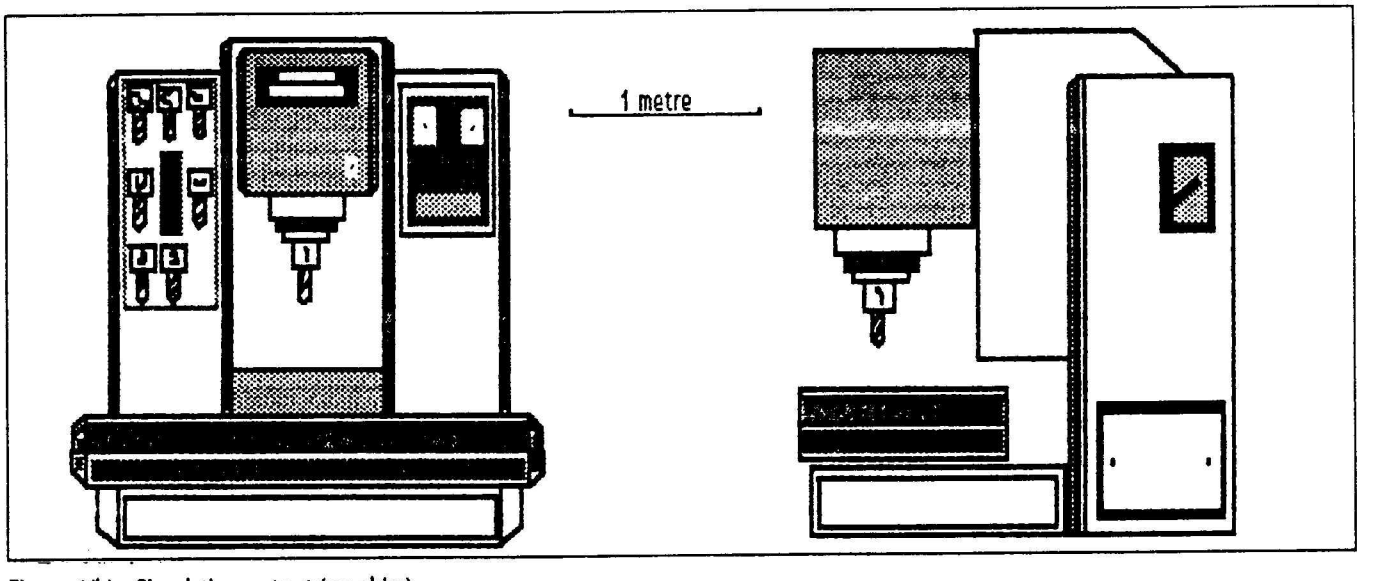

Figure 1(b) - Simulation output (graphics)

Project Status

As of the date of this report, structure diagrams for major modules and their component objects are being developed and programming is under way for some portions of modules. The project is moving from broad conceptual development into the more detailed and programming implementation stage. SMALLTALK is currently being used for the programming, based on SUN workstations and 386 machines. For further details on the OBJIMPS project see [17].

ACKNOWLEDGEMENTS

The enthusiam, dedication and major contributions of the initial research group has been much appreciated: Murray Mowchenko, Sofie Bakalchuk. The assistance of the following who joined the research group more recently is also gratefully acknowledged: Datong Guo, Peter Goldsmith, Molu Olomolade.

Special thanks are due to the Natural Science & Engineering Research Council of Canada (Research Grants #A4192 and #A4948) and also to Neuron Data Ltd for their support of the project through the NEXPERT software.

REFERENCES

1. Gaines, B.R., *"The Design of Expert Systems for Planning Flexible Manufacturing"*, Proceedings of Japan - USA Symposium on Flexible Automation Osaka, Japan, pp 761-768, July 1986.

2. Elavia, J.D., *"Where Do I Start with CIM"*, Proceedings of Autofact '85, Detroit, pp 1.1-1.21, November 1985.

3. Miller, R.D., *"Simplification, Management Control, The Intelligent Automation"*, Proceedings of Autofact '85, Detroit, pp. 1-14, November 1985.

4. Nassr, J.J., *"Considerations in Establishing a CIM Environment"*, Proceedings of COMPIN-'85, IEEE 85CH2136-0, Montreal, pp. 68-72, September 1985.

5. Kempf, K.G., *"Manufacturing and Artificial Intelligence"*, Robotics, Vol. 1, No. 1, pp. 13-25, May 1985.

6. Liu, D., *"Intelligent Manufacturing Planning Systems"*, Autofact '85, Detroit, SME Tech. Paper MS85-1070, pp. 1-17, 1985.

7. McDonald, M.E., *"The Smart Foreman - Expert Systems in Manufacturing"*, Detroit, SME Technical Paper MS85-1112, pp. 1-9, 1985.

8. Spooner, P.D., *"A Simulation Based Production Control System"*, Proceedings of the Advanced Manufacturing Systems Conference, Chicago, pp. 93-100, 23-25 June 1987.

9. Norrie, D.H., Fauvel, O.R., Gaines, B.R., *"A Flexible Manufacturing Expert System"*, Report 420, Department of Mechanical Engineering, University of Calgary, Alberta, Canada, February 1989.

10. Newquist, H.P., *"'ntelligent Data: Getting to First Base"*, AI Expert, Vol. 3, No. 1, pp. 21-24, January 1988.

11. R. Suri, *"RMT Puts Manufacturing at the Helm"*, Manufacturing Engineering, Vol. 101, No. 12, pp. 41-44, February 1988.

12. Gupta, T., Ghosh, B.K., *"A Survey of Expert Systems in Manufacturing and Process Planning"*, Private Communication, 1988.

13. Mellichamp, J.M., Kwon, O.J. and Wahab, A.F.A., *"Designing Flexible Manufacturing Systems with Expert System Technology"*, Private Communication, 1987.

14. Jefferson, D. etal, *"Distributed Simulation & the Time Warp Operating System"*, Operating Systems Review, vol. 21 (5); pp 77-93, November 1987.

15. Lomow, G., Cleary, J., Unger, B., *"A Performance Study of Time Warp"*, Distributed Simulation 1988, pp 50-55, Society for Computer Simulation, San Diego, February 1988

16. Unger, B., Lomow, G.A., Andrews, K., *"A Process Oriented Simulation Package"*, Distributed Simulation 1985, pp 76-87, Society for Computer Simulation, San Diego, January 1985.

17. Norrie, D.H., Fauvel, O.R., Gaines, B.R., *"Object-Oriented Planning, Scheduling and Simulation in Manufacturing Systems"*, Report No. 421, Department of Mechanical Engineering, University of Calgary, Alberta, Canada. February 1989.