A MODULAR PROCESS PLANNING SYSTEM ARCHITECTURE

Steven R. Ray
Factory Automation Systems Division
National Institute of Standards and Technology

ABSTRACT

A general purpose architecture for a modular process planning system is presented. Based upon emerging national standards in manufacturing, it offers easy integration among planning subsystems.

INTRODUCTION

There are a number of core issues common to all process planning applications. These include the representation of processes, the representation of process plans, and the definition of an architecture which can communicate these representations internally and with other systems. If a consensus can be reached within the process planning community on these central issues, compatibility between future planning systems should be far easier to achieve. Work has been underway within the Automated Manufacturing Research Facility (AMRF) at the National Institute of Standards and Technology (NIST) to address these problems.

THE AMRF

The AMRF [1,2,3] was established in 1981 to serve as a test bed facility to support research in measurement techniques and computer interface standards that are required for automated machining of parts in small lot sizes. The primary thrust of the project was to establish clear interface specifications and support modular structures to allow plug-compatibility between systems. This plug-compatibility allows both a flexible manufacturing environment and offers the capability of incremental automation in existing facilities. Results of this work are already contributing to the formulation of standards for low level robot interfaces, N/C machine tool interfaces [4], communication standards and PDES (Product Data Exchange Specification) [5,6].

REPRESENTATION

Work Element

In many computer applications the difficulty of solving a problem is dependent upon the representation scheme used. In automated manufacturing the early determination of an appropriate representation is also important. For example, how should some metal-cutting process be communicated throughout a factory? In the AMRF this type of question led to the development of the concept of a *work element* [7]. Work elements represent individual processing steps, which can be combined to create process plans. They can be thought of as operators in a state space. Whenever a work element is invoked, a state transition takes place. A process plan corresponds to a sequence of operators applied to an initial state (perhaps a part blank), resulting in a goal state (a finished part).

Directed graph structure.

A process plan can be thought of as collection of individual processes (or work elements) into a coherent structure directed toward accomplishing some goal. The structure for relating these work elements must allow for:

1. Processing precedence - determine the sequence of tasks. This is the basic capability of any process plan structure.
2. Alternative sequences - express different task sequences which provide the same result. Should also provide a means for a scheduler to determine which sequence is currently optimal. The decision should be deferred until scheduling or manufacturing time.
3. Parallel actions - explicitly show how multiple task sequences within a plan can be performed at the same time. It is assumed that separate plans can be executed in parallel as separate jobs.
4. Decomposition - should support the concept of hierarchical process plans, where a process at one level within a hierarchy can be expanded into a collection of processes at a lower level.
5. Synchronization - provide for synchronization between multiple parallel task sequences within a plan (as in item 3) and between plans.

Presented at the 1989 IIE Integrated Systems Conference, Atlanta, Georgia, November 1989

6. Resource monitoring - provide the means for collecting and updating statistics for resource availability and utilization to support scheduling and resource allocation.
7. Extensibility - support extensibility by not constraining the user to a fixed functionality. Users must be able to customize process plans to support their facility.

To support these requirements, a process plan structure (*ALPS*) was defined for use within the AMRF. ALPS, an acronym for **A L**anguage for **P**rocess **S**pecification, [8], is a language based upon a directed graph structure, with work elements represented by nodes in the graph. The nodes are connected to one another by directed arcs that indicate temporal precedence. There are seven major classes of graph nodes: termination, task, split, join, synchronization, resource, and information. The roles and use of each class of node is explained in reference [8].

The use of graph structures for representing processes is not new. Related work has been done, particularly in assembly planning, using AND/OR graphs to represent the states of an assembly, from individual components to the completed assembly [9,10]. The work on ALPS extended these concepts to provide for richer branching behavior and a larger number of node classes.

During the ongoing development of the ALPS language, numerous issues were brought to light concerning the desired behavior of a process plan representation. These issues are being brought to the attention of the international standards community through the voluntary Product Data Exchange Specification (PDES) organization. The PDES organization is currently formulating a draft standard for a process plan representation as part of its larger mission of defining a standard for representing all information relating to a product life cycle. As a PDES standard process plan becomes available, it will be adopted within the AMRF.

PROCESS PLANNING ARCHITECTURE

The definition of a work element and a process plan structure satisfy one of the requirements for a process planning system, namely the communication of a manufacturing specification to downstream systems such as scheduling and execution. There still remains the problem of a general, expandable architecture for the process planning system itself. The need for an architecture arises because process planning systems in the future will likely be built from multiple subsystems, or modules, each of which handles an aspect of process planning. These modules could be expert systems, numerical systems or interactive systems.

For such an assembly of subsystems to work effectively together to solve process planning problems, there needs to be a mechanism for sharing information between systems. Further, this mechanism should not impose any particular problem-solving strategy. The architecture described below attempts to address these requirements and thus to provide a neutral framework within which multiple process planning subsystems can cooperate.

The architecture under development is based upon a number of generally accessible databases. This approach is similar to others looking at integrating manufacturing systems and distributed problem solving [11,12]. In the approach described here, the databases contain all of the input and output information for all of the planning subsystems. Thus, any information which is to be shared between systems is communicated by means of one or more of the databases. Information which is manipulated internally by a subsystem, such as rules, tables or other data structures, is still stored locally within that system. In addition to the ALPS schema, two other database schemas have been defined. They are called the Manufacturing Resource Model and the Plan Formulation Model. Within the AMRF, these models are being implemented using a commercial object-oriented database system.

Manufacturing Resource Model

The Manufacturing Resource Model maintains the specifications and status of all resources found in a manufacturing facility. Categories of resources include material, equipment, human, and information. The model further classifies these and other categories into several hundred specific classes. The model itself is generic to discrete manufacturing facilities, with a current emphasis on machine shops. An actual implementation of a database built from the model would then have data corresponding to a particular shop floor, including such items as each cutter, fixture, machine tool, etc. The Manufacturing Resource Model is useful to several systems in a manufacturing facility. Process planning systems use it when considering how to produce a product, based upon what manufacturing tools are available. In addition, libraries of work elements understood by computerized controllers are defined in the Manufacturing Resource Model. These work element definitions are referred to when composing process plans. Scheduling systems use the model to determine the availability and loading of resources.

Plan Formulation Model

The Plan Formulation model is an abstract model that allows great flexibility in process planning problem solving. The model is designed to be used as a scratch-pad shared among the planning subsystems. The model is centered around three concepts: goals, actions and resources. Goals are defined as any desired intermediate or final state of a product or of the facility. A goal may be decomposed into a set of subordinate sub-goals, and a goal may have precedence relationships with other goals. The goal decomposition can occur to any number of levels. A goal is either decomposed into sub-goals, or is accomplished via an action. An action may support a goal and may further be decomposed into subordinate actions. As with goals, an action may have precedence and other types of constraints imposed on it. Finally, an action may require the use of one or more resources, by reference to an entries in the Manufacturing Resource Model. A process plan can be considered as formulated when every identified goal has either an associated action, or is decomposed into a number of sub-goals.

This discussion of the Plan Formulation model is necessarily abstract because the model must be applicable to a wide variety of problems within process planning. For example, a problem usually encountered early in a process planning session is the identification of manufacturing features for a machined part. These features would appear in the Plan Formulation model as goals, with each of them being a sub-goal of the higher goal of manufacturing the designed part. Actions associated with these goals would be the manufacturing processes identified by a process selection system. The intent is that by keeping the information abstract, different systems will be able to share partial solutions to a planning problem in a mutually understandable way.

ALPS Model

The Plan Formulation model is specifically designed to address information needs during process planning. The model is not particularly suitable for use by downstream systems, such as schedulers, resource managers, and execution systems. A scheduling system needs information such as the sequencing requirements of processes, trade-offs between alternative approaches, resource commitments, costs, and delivery times. For example, a scheduler is not interested specifically in why certain processes are required, but in just how long a process takes, how much it costs, and what resources are needed. Furthermore, the tasks identified in the Plan Formulation model are not stratified according to a control hierarchy. Thus, the data in the Plan Formulation model described above is not in a form convenient to other systems. There is a need to re-express this data in terms which are compatible with the rest of the manufacturing environment. This re-expression is specified by the ALPS model. Within the planning architecture, a system (or person) must convert the implicit information present in the Plan Formulation model into the explicit form defined by the ALPS model. Thus, the precedence constraints and optimization criteria which are present in the formulation model must reappear as collections and sequences of activities directly amenable to automatic execution. What is put into the ALPS model is the "how" of a process plan, but not the "why."

Uniform Access to Databases

For the above mechanism of information sharing to be effective, there must be some uniform method of accessing the information in any of the database models, regardless of their implementation. In the AMRF, the uniform access method is defined by the Integrated Manufacturing Data Administration System (IMDAS) [13]. The objective of the IMDAS project was to integrate any number of distributed and heterogeneous databases such that they could be addressed in a uniform manner. The IMDAS system consists of a number of distributed data servers (DDAS), each of which supervises one or more basic data servers (BDAS). A database client - typically a computer program - sends a database request to a DDAS using a query language similar to SQL [14]. This query is converted to an internal standard format called a query tree format and is compared against a global data dictionary. The data dictionary identifies the information entities in all of the subordinate databases. The DDAS then fragments the query tree, if necessary, into several sub-trees to be dispatched to separate databases. If all the information to support a query is contained in one database, this fragmentation does not occur. The query tree(s) is then passed to the appropriate BDAS, which translates the query into the native query language of that particular database. Finally, the results of a query are merged with the results from any other participating databases, and passed back to the client.

For the process planning system, the three databases described earlier are being implemented as part of the IMDAS system. For a process planning subsystem to participate in a problem it only needs to know what information is present within the IMDAS, and it must be able to generate queries in the standard format.

Common Memory

To allow the planning subsystems, running on a variety of computers, to communicate with the databases, an additional implementation layer is required. In the AMRF this layer is called common memory [15], and was implemented to allow any computer processes to appear to share regions of memory. Any computer process can declare itself as a reader and/or a writer of a common memory area. In this manner, two processes can communicate with one another, even if they run under different operating systems and use different programming languages. The common memory layer is used for all communications with the IMDAS as well, since the IMDAS is simply another process.

PROTOTYPE IMPLEMENTATION

To test and demonstrate the concepts described above, a prototype modular process planning system is being implemented, shown in Figure 1. In addition to the three models described in the paper, the figure shows the PDES part model, and a status model. The part model is the repository of the product design and serves as the principal interface from design to process planning. The status model will form part of the AMRF implementation and will track the state of the

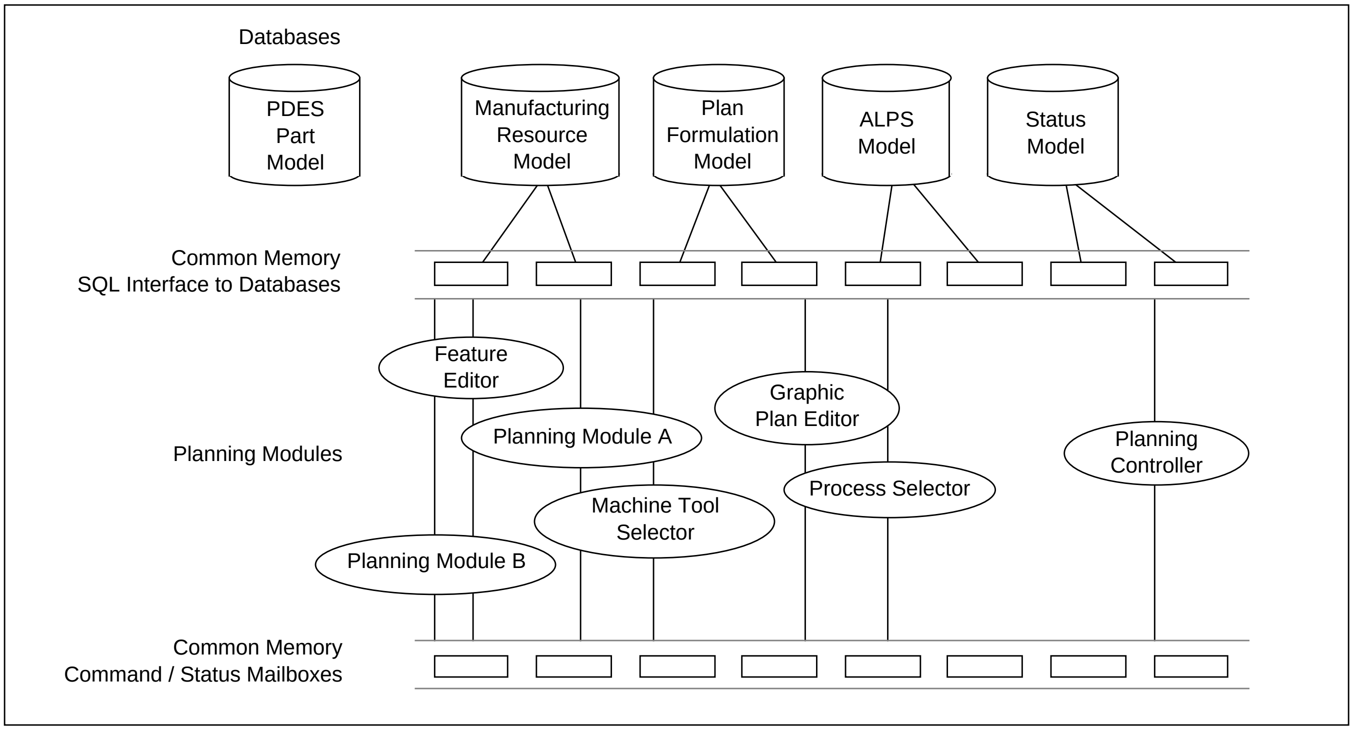

Figure 1. Modular Planning System Architecture

planning system itself, to provide for recovery in case of an unexpected system failure. The other major feature not yet discussed is the planning controller shown at the right of the figure. The purpose of the controller is to coordinate the actions of all the other planning modules. Hence, the controller is the agent which dictates the problem solving-strategy within a given implementation. In any particular implementation, the rules governing the behavior of the controller should be defined to suit that site. However, the framework within which the controller operates can be the same for all implementations. The construction of a planning controller is the subject of ongoing research.

CONCLUSIONS

A number of unifying issues have been identified to support automated process planning systems in the future. Certain representation and architectural issues can and should be agreed upon to enhance future progress in process planning research. These include a standard representation for processes and process plans, and standard information models to support the creation of these plans. Three such models have been discussed. The Manufacturing Resource Model describes the specifications and state of every resource in a manufacturing facility. The Plan Formulation model serves as a "scratch-pad" for planning subsystems to share partial planning solutions. The ALPS model serves as a repository for process plans in a form usable by other systems in a manufacturing environment. These models are accessed in a generic manner by building the database implementations as part of a distributed database architecture, such as the IMDAS. This approach allows any number of planning subsystems to participate in the formulation of process plans. The only requirement is that a system be able to generate standard database queries and agree upon the structure of the supporting database models. It is not important which standards body sponsors the effort to formulate these models. What matters is the need to proceed with their definitions as quickly as possible. With a consensus in the areas described in this paper, plug-compatible process planning components become a realistic goal.

REFERENCES

[1] Simpson, J.A., R.J. Hocken and J.S. Albus, "The Automated Manufacturing Research Facility of the National Bureau of Standards," *Journal of Manufacturing Engineering,* 1, Vol. #1, 1982.

[2] Hocken, R. and P. Nanzetta, "Research in Automated Manufacturing at NBS," *Manufacturing Engineering*, 91, Vol. #4, 1983.

[3] McLean, C.R., "An Architecture for Intelligent Manufacturing Control," *Proc. Summer 1985 ASME Conference,* Boston, MA, August 1985.

[4] EIA PN1924 "Numerical Control Message Specification - A Companion Standard of EIA-511," Electronics Industries Association, Washington, D.C., May 1989.

[5] Henghold, W.M., G.C. Shumaker and L. Baker, "Considerations for the Development and Implementation of PDES within a Government Environment," *Report AFWAL-TR-89-8009*, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, OH, February, 1989.

[6] ISO Draft Proposal 10303, ISO Technical Committee 184, subcommittee 4, submitted November 1988.

[7] Ray, S.R., "A Knowledge Representation Scheme for Processes in an Automated Manufacturing Environment," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Atlanta, GA, October 1986.

[8] Catron, B.A. and S.R. Ray, "ALPS - A Language for Process Specification," *International Journal of Computer Integrated Manufacturing, special issue on Process Planning and Design for Manufacture* (submitted), expected January 1990.

[9] Homem de Mello, L.S. and A.C. Sanderson. "AND/OR Graph Representation of Assembly Plans," *Proc. of AAAI-86*, Vol. 2, 1113-1119, Philadelphia, PA, Aug 11-15, 1986.

[10] Passler, E. et al., "Production System Design: A Directed Graph Approach," *Journal of Manufacturing Systems*, Vol. 2, No. 2, 107-116.

[11] Naylor, A.W. and R.A. Volz, "Design of Integrated Manufacturing System Control Software," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 6, November/December 1987.

[12] Cutkosky, M.R. and J.M. Tenenbaum, "A Methodology and Computational Framework for Concurrent Product and Process Design," *Mechanism and Machine Theory*, in press, 1989.

[13] Libes, D. and E. Barkmeyer, "The Integrated Manufacturing Data Administration System (IMDAS) - An Overview," *International Journal of Computer Integrated Manufacturing*, Vol. 1, No. 1, January 1988.

[14] Date, C.J., "A Guide to the SQL Standard," Addison-Wesley, Reading, MA, 1987.

[15] Libes, D., "User-Level Shared Variables," *Proceedings of the Summer 85 USENIX Conference*, May 1985.

The research reported in this paper was partially supported by the Navy Manufacturing Technology Program and by the Department of Defense Computer-Aided Acquisition and Logistic Support (CALS) Office.

BIOGRAPHICAL SKETCH

Dr. Steven R. Ray is project manager for the process planning team in the Factory Automation Systems Division, Center for Manufacturing Engineering, National Institute of Standards and Technology. His work focuses on the architectural, representation and interface issues in automated process planning. This effort is part of the AMRF (Automated Manufacturing Research Facility) project at NIST. Dr. Ray received his Bachelors degree in Physics from Bristol University (England) and his Ph.D. in Mechanical and Aerospace Engineering from Princeton University.