NRC Publications Archive
Archives des publications du CNRC

Approach to distributed process planning and its architecture design

Wang, Lihui; Feng, Hsi-Yung; Cai, Ningxu

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:
https://doi.org/10.4224/21272476
IMTI ITFI Publication; no. IMTI-TR-009, 2002-08

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=ef1dd6a7-95f4-442f-9c58-3d9452f4cf70
https://publications-cnrc.canada.ca/fra/voir/objet/?id=ef1dd6a7-95f4-442f-9c58-3d9452f4cf70

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

INTEGRATED MANUFACTURING
TECHNOLOGIES INSTITUTE

INTSTITUT DES TECHNOLOGIES
DE FABRICATION INTÉGRÉE

REPORT
RAPPORT

Pages: 29

Fig.
Diag. 17

For
Pour IMTI

Date: August 1, 2002

SAP Project #:
de Projet de SAP: ____________

Unclassified

IMTI-TR-009 (2002/08)

Approach to Distributed Process Planning And Its Architecture Design

Submitted by
Présenté par ____________________
Group Leader

First Author
Auteur Premier ____________________
Lihui Wang

Approved
Approuvé ____________________
Director

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The era of today's design and manufacturing systems with downstream information flow is passing and will be gradually phased out, because the conventional design for manufacturing technologies are insufficiently flexible due, in part, to their rigid system architecture. Today's computer-aided technologies (CAX: CAD/CAPP/CAM, etc.) have existed for more than three decades. Despite all the accomplishments, they still need more capability of integrating the downstream manufacturing and control information into their early decision-makings. Most existing CAX systems are designed based on sequential information flow. They are not adaptive to the dynamic manufacturing shop floor environments. Further improvements can only be realized by a paradigm shift that allows downstream manufacturing restrictions to be considered at the early product design stage. Future DFM systems (design-for-manufacturing or design-for-machining) will be capable to satisfy both local and global objectives in a four-dimensional space, where related information can be exchanged from stream to space in many directions, seamlessly, concurrently, efficiently, and accurately [1].

In the last decade, the change in market requirements towards a larger variety of products in smaller batch size, has also led to the concept of next generation reconfigurable manufacturing system (RMS) being an integrated network of distributed resources simultaneously capable of combined knowledge processing and material processing. RMS is a manufacturing system which can be created by incorporating basic process modules, both hardware and software, that can be rearranged or replaced quickly and reliably [2]. It will allow flexibility not only in producing a variety of parts, but also in changing the system itself. It is required to be agile and fault-tolerant, and will contain collaborative and intelligent entities that can dynamically adjust themselves so as to achieve a global system objective. The manufacturing processes involved in an RMS are complicated, especially at machining shop floor where a large variety of products (usually in small batch size) are handled dynamically. The changing environment of an RMS characterized by large variety of products in small batch sizes requires creating an intelligent and dynamic process planning system that is responsive and adaptive to the rapid adjustment of production capacity and functionality. Traditional process planning methods are time-consuming and error-prone, if applied to such a changing environment. In response to the requirement and to facilitate the dynamic manufacturing activities of an RMS, a new process planning approach with appropriate open architecture is needed to handle the run-time process planning for distributed resources in such changing manufacturing environments.

The objective of this research is to develop the needed methodology and architecture for distributed process planning (DPP), a novel adaptive CAPP (computer-aided process planning) approach to improve the system performance of machining shop floors. The primary focus of this report is on the architecture of the new process planning approach, using function blocks as new controller language. The secondary focus is on the other supporting technologies such as machining feature-based reasoning and agent-based decision-making for process planning. Traditionally, the process planning is the task that transforms a product's design information into the manufacturing processes and determines the operation sequence in advance of actual machining [3]. Thus, maintaining the consistency of all process plans and keeping them optimized could be a difficult task. Different from traditional methods, the proposed approach uses a two-layer structure – supervisory planning and operation planning. The former focuses on product data analysis, machine selection, and machining sequence planning, while the latter considers the detailed working steps of the machining operations inside of each process plan. The supervisory planning is generally performed in advance at shop floor level, followed by the operation planning accomplished at run-time at machine level by open intelligent CNC controllers. This report explains in detail how a generic process plan and its corresponding operation plan can be achieved by using the proposed DPP approach.

2. LITERATURE REVIEW

From design to manufacturing of a mechanical product, a number of steps must be followed, such as geometry design, process sequencing, cutting tool selection, tool path planning, operation optimization, NC data generation, as well as fixture design and set-up planning. These preparatory tasks are generally referred to as process planning. Research on process planning has been a hot topic for decades. In spite of the benefits promised by the various developed CAPP systems, their adoption by industry is still painfully slow. Today, shop floor engineers and operators still have to meet and respond the challenges for producing highly complex products, especially in very small batches. An efficient adaptive process planning system is thus required to address the shop floor dynamism and to improve the shop productivity.

Since 1965 when Niebel reported the idea of using the power of computers to assist the process planning [4], subsequent research has been numerous. The first CAPP system, with the same acronym but for *CAM-I's Automated Process Planning* system, was developed in 1976 under the direction and sponsorship of CAM-I (Computer Aided Manufacturing – International) [5]. In the same year, another CAPP system called MIPLAN was developed by OIR (Organization of Industrial Research) [6]. Since then, research on CAPP has been very active. Until 1989, more than 156 CAPP systems have been reported and summarized by Alting and Zhang in their literature survey [7]. More than 300 papers have been published in this area during the last three decades [8]. Most of the papers appear to introduce only specific CAPP systems, although a few papers give a general survey [7-13]. Among many others, previous research studies on process planning include object-oriented approaches [14-16], GA-based approaches [17][18], neural network-based approaches [19][20], Petri net-based approach [21], feature recognition [22-27] or feature-driven approaches [28][29], and knowledge-based approaches [14][30-35]. These approaches and their combinations have been applied to various specific problem domains, such as tool selection [36-42], tool path planning [43-45], machining parameters selection [39][46], process sequencing [25][47][48], and set-up planning [49][50]. The accomplishments can be classified into four categories: variant, semi-generative, generative, and AI (artificial intelligence) based systems, depending on the planning approach itself, the data and knowledge used, and the modeling and analysis techniques applied. The variant approach marks the beginning of

CAPP systems, and is basically a computerized data-retrieval and editing method; whereas the semi-generative or generative approach is a knowledge-based method that partially or automatically generates a process plan according to a product's features and its manufacturing requirements. Most AI-based process planning systems can also be categorized into a (semi-) generative approach.

Other hybrid CAPP approaches include interactive process planning [16], rapid process planning [51], micro- and macro-planning [52][53], integrated process planning [54], dynamic process planning [55], incremental process planning [56], web-based process planning [57][58], and agent-based process planning [59-66]. No attempt is made in this report to compare and evaluate these different approaches, although some details to certain approaches are given below.

As decentralization of business grows, research focus of process planning is recently moving towards solving problems in distributed manufacturing environments. Tu *et al.* introduced a method called IPP (incremental process planning) for one-of-a-kind production (OKP) [56]. The IPP approach is used to extend or modify a primitive plan (a skeletal process plan) incrementally, according to the new features that are identified from a product design until no more new features can be found. A complete process plan generated by the IPP may include alternative processes. This means that a given part can also be processed by alternative machines in alternative sequences in a different plant.

Such kind of distributed manufacturing environment also changes the way of applying AI techniques to process planning. In addition to centralized AI approaches (*e.g.* genetic algorithms, neural networks, fuzzy logic, knowledge-based or expert systems, etc.), agent technology being one type of distributed AI approaches has attracted wide attention. Instead of being one large expert system, cooperative intelligent agents are being used in developing distributed CAPP systems. Among others [59-61], CoCAPP (Cooperative Computer-Aided Process Planning) attempts to distribute complex process planning activities to multiple specialized problem solvers and to coordinate them to solve complex problems [62]. The CoCAPP attempted to satisfy five major requirements: autonomy, flexibility, interoperability, modularity, and scalability. It utilizes cooperation and coordination mechanisms built into distributed agents with their own expert systems. Each agent in the system deals with a relatively independent domain of process planning. Collectively, the multiple agents can solve complex problems.

Shih and Srihari proposed a distributed AI-based framework for process planning [63]. Their approach decomposes the entire production control task into several subtasks, each of which is implemented by an intelligent agent. By working collectively, the agents can arrive at a solution for the problem. Similarly, Sluga *et al.* introduced a VWS (virtual work system) [64] as the essential building block for decision-making in a distributed manufacturing environment. The VWS represents a manufacturing work system in the information space, and is structured as an autonomous agent. It is a constituent entity of an agent network in which dynamic clusters of cooperating agents are solving manufacturing tasks. The decision-making in process planning is based on a market mechanism consisting of bidding-negotiation-contracting phases. The VWS approach aims at enabling dynamic decision-making based on the actual state of a given environment. The bidding-based approach is also useful to integrate product design, process planning, and scheduling [65].

CyberCut is a research project that aims to develop a networked manufacturing service for rapid part design and fabrication on the Internet [57]. A critical part of this service is an automated process planning module that is capable of generating process plans to satisfy the desired geometries and specified requirements. Three types of agents are designed to facilitate the CyberCut: primary process planning agent, environmental planning agent, and burr minimization tool path planning agent [66]. The multi-agent planning module incorporates conventional and specialized planning agents for environmental consideration and burr minimization. However, the interactions between those agents are based on human decisions.

Agent-based approach is also being recognized as one of the effective ways to realize adaptiveness and dynamism of process planning. Zhang *et al*. proposed an AAPP (agent-based adaptive process planning) system on top of an OOMRM (object-oriented manufacturing resources modeling) framework [16]. The OOMRM is used to model manufacturing resources capability and capacity in an object-oriented manner, which intends to encapsulate manufacturing system knowledge and the methods of using the knowledge, while the AAPP is implemented as a man-machine integrated process planning platform. Instead of automating process planning tasks completely, the AAPP system provides an interactive mode for experienced manufacturing engineers to map out more reasonable and flexible manufacturing processes for a realistic manufacturing environment. Five agents are used in the AAPP system to carry out part information classification, manufacturing resources mapping, process planning, human planning, and machining parameter retrieval. A contract net-based scheme is utilized as the coordination protocol between agents.

As partial trends in CAPP pointed out by H. A. ElMaraghy in 1993 [13], integrations of CAPP with either product design or manufacturing scheduling or both remain attractive to researchers and practitioners. Previous studies in this area include design-planning integration [51], design-to-control [28], reactive planning environment [55], machining feature-based product design and process planning [67][68], and planning-scheduling integration [18][69]. Appendix summarizes the latest developments in the area of CAPP. The listings consist of the name of the system or tool, research group and its reference, key features, and application domain.

Despite the achievements of those CAPP systems, their ability to recognize design specifications and distinguish whether existing process plans suit a new workpiece is still weak. This may be due to the fact that most of the developments attempt to recognize very detailed design information and lack adaptive-learning and decision-making capability. Although there exist a lot of efforts devoted to generative process planning, their adaptiveness to the changing environment remains insufficient. Most CAPP systems available today are centralized in architecture, vertical in sequence, and off-line in knowledge processing. It is difficult for a centralized off-line system to make adaptive decisions in advance, without the knowledge of actual status of machines at shop floors. Instead, a run-time distributed process planning system is required to deal with the dynamic situations (*e.g.* product changeover, job delay, unavailable fixture, tool or machine breakdown, etc.) regularly happening in today's machining shop floors.

3. DISTRIBUTED PROCESS PLANNING

3.1 Basic Requirements

The distributed process planning (DPP) should function as a transparent filter between product design data and NC control data. It should be able to transfer and process the design data so as to meet the requirements of subsequent NC machining. The basic requirements of the DPP system can be summarized as follows.

- Intelligent: to be able to handle uncertainty and to increase probability of success of controlled process.
- Fault Tolerant: to be able to react to the failure and to find substitution for the purpose of quick fault recovery.
- Portable: to apply DPP modules, especially the Operation Planning (OP) module, to different platforms including open CNC controllers.
- Extendable: to be able to include new databases and knowledge, or be adaptable to new applications.
- Distributable: to be able to distribute function block-embedded process plans over a set of CNC machine controllers.
- Reconfigurable: to be adaptive to the dynamic situations including job delay, product changeover, tool and machine breakdown, etc.
- Ease of Use and Reliable: to be easy to install and use. The system functionality should be consistent and reliable.
- Reusable: to be able to reuse most part of the DPP functions and modules.

3.2 Concept of Design-for-Machining

Machining features [68] are used as information carriers to facilitate product design, process planning, and NC machining. Machining features are those shapes such as step, pocket, slot, ring, and hole that can be easily achieved by the available resources and defined machining technologies. Each feature possesses a set of loosely coupled information about how to fabricate it, e.g. cutting tool ID, machining sequence, cutting strategy, and tool path generation logic. As shown in Fig.1, a product is first designed by subtracting machining features from a blank, same as materials being removed during machining. It is then passed to the DPP for supervisory planning, without feature recognition. Since the relevant information for DPP is kept by machining features that are part of the product's design data, the time and efforts for the subsequent tasks (e.g. machine/tool selection, tool path planning, cutting condition generation, and NC code creation, etc.) can be significantly reduced. Resource databases storing information of machining features, machines, cutting tools, and cutting technologies are shared among the design, planning, and control modules. From design to NC control, machining features are used for information retrieval, data exchange, and decision-making support. The idea of *design-for-machining* is to bring design and NC machining issues together, with as much integration as possible [70]. An obvious merit of the idea is being able to prevent a designed product that is later found to be difficult or even impossible to produce with existing resources.

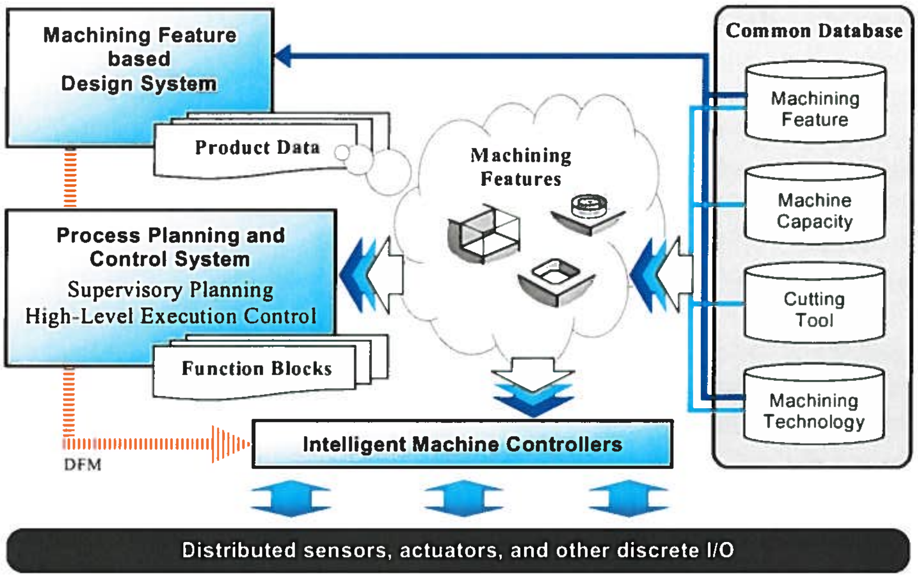


Figure 1. Concept of *Design-for-Machining*.

As machining features are used extensively in this report, twelve typical machining features are classified and illustrated in Fig.2. Machining features (as standard shapes that can be machined) are different from design features (geometric standard shapes such as vertex, curve, and face) in that each machining feature holds a set of loosely coupled information about how to fabricate it. Designing a product by combining these machining features properly, an engineer can easily conduct process planning using the information provided by the machining features and thus assure its manufacturability.

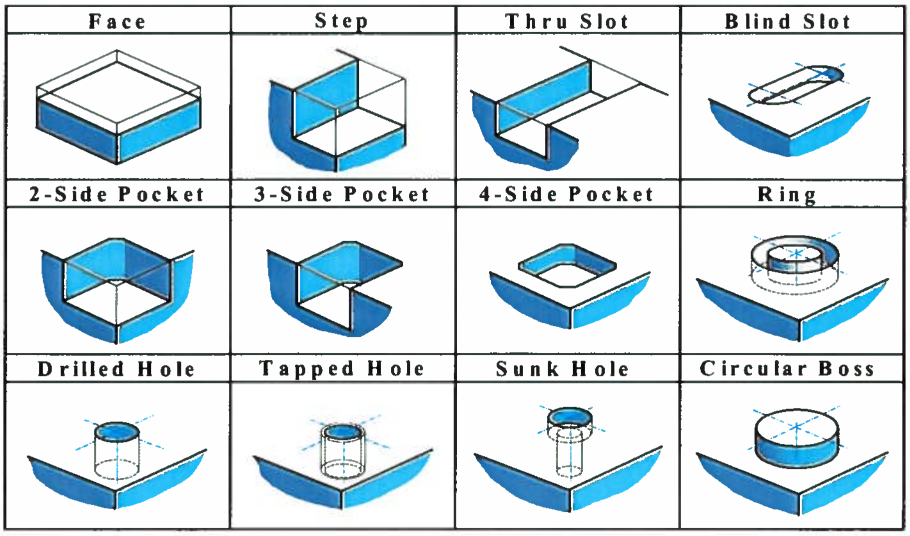


Figure 2. Typical machining features.

3.3 Approach to Distributed Process Planning

Reconfigurability is an important issue of today's dynamic shop floors. Traditional CAPP systems are weak in handling such dynamism. A process plan that is generated for a specific machine may have to be generated again when the machine is unavailable or broken. Such repetitive planning tasks happen frequently in dynamic environemnt. There is a need to develop a new approach to address the dynamism issue.

A process plan generally consists of two parts: *generic* data (machining method, machining sequence, and machine tool information) and *machine-specific* data (tool data, cutting conditions, and tool paths). A two-layer hierarchy is considered suitable to separate the generic data from those machine-specific data in distributed process planning [29]. As shown in Fig.3, the tasks of DPP can be divided into two groups and accomplished at two different levels: shop-level supervisory planning (SP) and machine-level operation planning (OP).

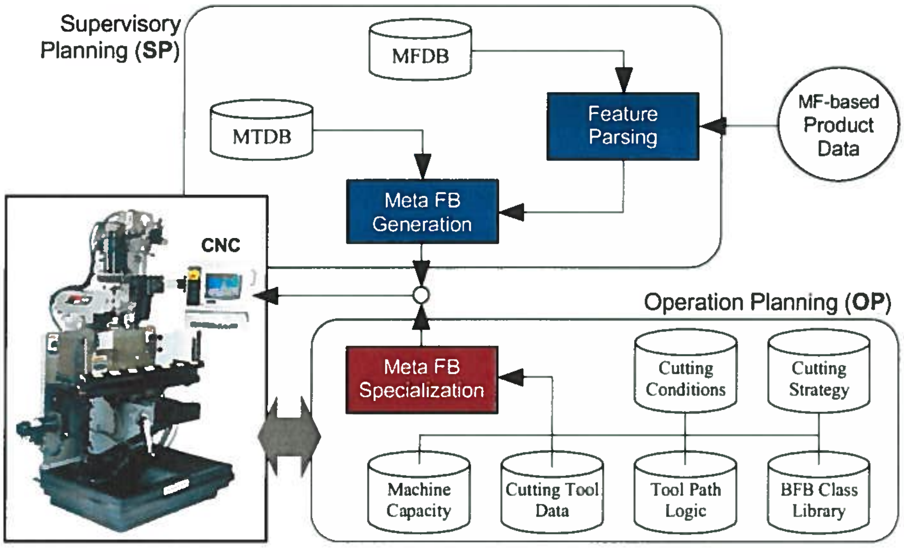


Figure 3. Distributed process planning.

Machining feature-based product data is initially passed to the SP module for feature parsing and feature interaction detection, followed by generic process plan generation. Decision-makings in SP module are facilitated by a machining feature database (MFDB) and a machining technology database (MTDB). The high-level process plans hold the necessary information for low-level operation planning, including machining method, selected machine tool, and required machining sequences. They are encapsulated into a set of meta function blocks and then dispatched to each dedicated machines. The detailed operation planning is accomplished by the CNC controller of each machine. Since the machine-specific data (tool paths and cutting conditions) are generated by appropriate machine controllers, the meta function blocks with high-level process plans become generic and open. This makes the meta function blocks suitable for serving as a new CNC controller language, and for cross-platform data distribution to different CNC controllers.

In the near future, multiprocessor-embedded CNC controllers will be powerful, open, and intelligent enough to handle numerous run-time exceptions while performing normal machining operations. The tasks of the detailed operation planning can, therefore, be assigned to such open CNC controllers. The cutting conditions and tool paths of a particular machining operation will be determined by the controller of the selected machine itself, based on the machine's dynamics, the information provided by high-level process plans, and by its own capability. It can be realized either through multithreading or by the PC front-end of a controller. Because a controller is knowledgeable about the dynamics of a machine and its run-time status, cutting conditions optimization, in-line surface error compensation, and adaptive axis control will become possible. Figure 4 illustrates how information flows and what tasks are undertaken within the two-layer planning structure.

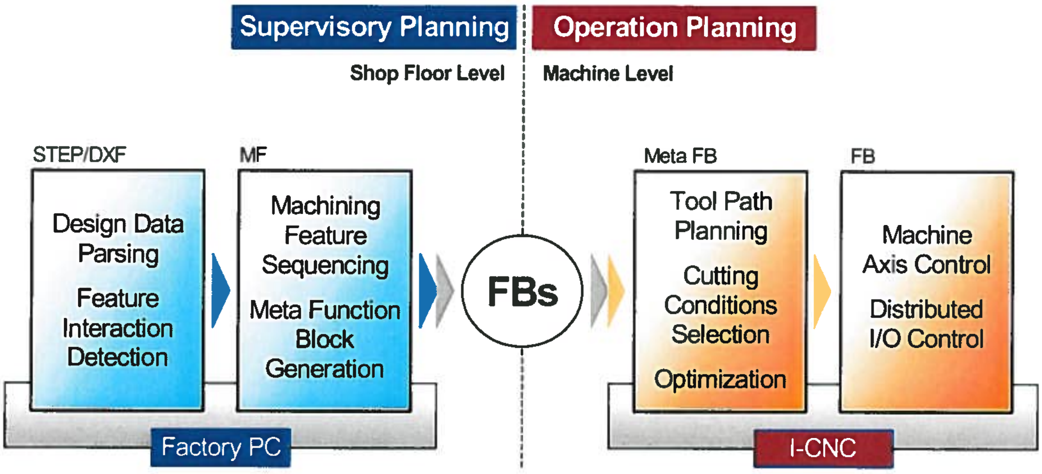


Figure 4. Information flow and task assignment.

3.4 Agent-Based Decision-Making

The centralized process planning that has defined the traditional CAPP structure in today's manufacturing systems is no longer suited to the rapidly changing shop floor environments. For efficient use of manufacturing resources and increased flexibility, it is necessary to migrate a CAPP package into a distributed information processing system in which entities can work cooperatively towards overall system goals. This requires distributed parallel computation, asynchronous process coordination, and standard communication protocol. Such a planning system can be readily realized through the multi-agent paradigm. A multi-agent system comprises a set of autonomous interacting software entities called intelligent agents, which are knowledgeable in their local domain and able to share the responsibility of achieving multi-objective system goals through negotiation. A multi-agent system contains a population of heterogeneous agents inter-operating asynchronously with other agents through collaboration, based on a well-defined message-passing mechanism [71]. As a planning task develops, the relevant agents will be dynamically grouped into coordination clusters to facilitate their focus on the current task. These clusters will be active as long as required and be destroyed when no longer needed.

Figure 5 illustrates a multi-agent decision-making mechanism for our distributed process planning. As shown in the figure, relevant agents for a particular task form, or are formed, into virtual clusters [72]. These clusters are coordinated and arbitrated by mediator agents, such as *product manager*, *planning manager*, and *resource manager*. These mediators are coordinator agents that possess the organizational knowledge and meta-level rules to facilitate cooperation among intra- and inter-agent communities. Distributed process planning is thus conducted by the dynamic groupings of well-organized agent clusters and coordinated by the three mediators. Messages are exchanged among agents within/between the clusters for communication. Based on this architecture, a process plan can be created by the collaboration of multiple cross-functional agents through negotiation and information sharing.

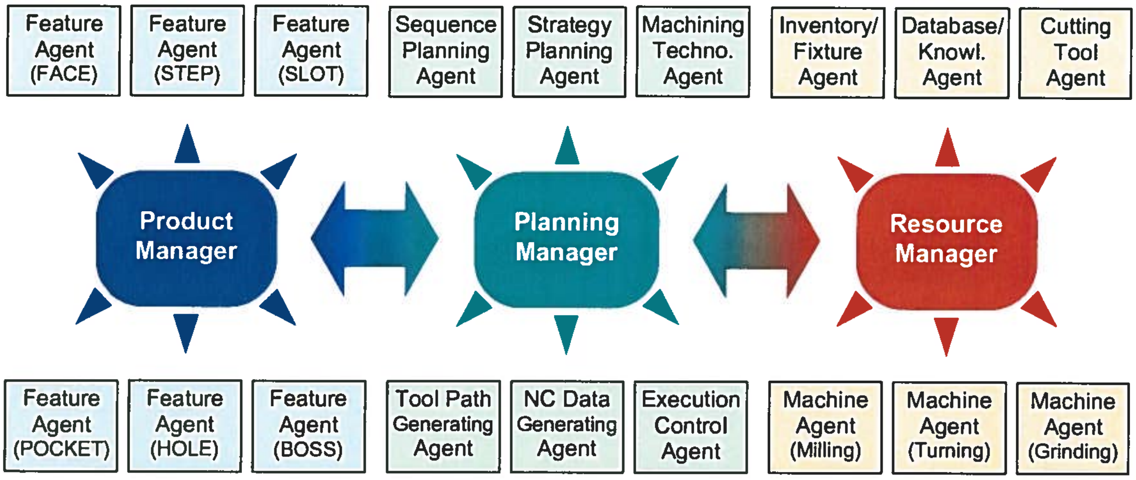


Figure 5. Agent-based decision-making.

Within the context, each machine on the shop floor is represented by an autonomous machine agent having knowledge about its own capability, assigned tooling, coolant, and spindle/axis information. Each cutting tool is represented by an autonomous tool agent that has knowledge about its tool type, dimension, remaining tool life, and tool conditions (tool wear, etc.). All the agents involved in a given task communicate with each other asynchronously and simultaneously for local and global decision-making. For example, the feature agents cooperate with the database agent and indirectly get access to the feature knowledge base for feature interaction and manufacturability assessment, while the tool-path generating agent negotiates with machine agents, tool agents, strategy-planning agent and the machining technology agent to find an optimal cutter trajectory. On the other hand, the sequence-planning agent is responsible for arranging an appropriate sequence of machining operations. Once the control data is finalized by the NC data generating agent, it will be passed on to the execution control agent, which interacts with the real control world, for distribution and execution. All the agents use KQML (knowledge query manipulation language) as standard protocol for communication. On behalf of their counterparts – physical devices, agents are used in negotiation and decision-making in the software domain. The same mechanism can be applied to both supervisory planning and operation planning, although different agents may be active for decision-making at each level.

By the nature of decentralization and for the ease of efficient data accessing, a distributed database is considered effective for the distributed process planning. As shown in Fig.6, the data contents of such a distributed database can be classified into generic and machine specific resources, which are referenced by the SP/OP module, or by the DPP system. The reasons for separating the data resources into categories are to keep the system functions generic and open, and to simplify the task of database management. In most cases, only the machine specific information is changing and needs to be updated dynamically. Increasing data transparency of the DPP-common resources and restricting data accessibility of the SP/OP-common resources can improve the entire system performance. One example is to allow the machine specific resources that are OP-common to be accessible only by the authorized machines.

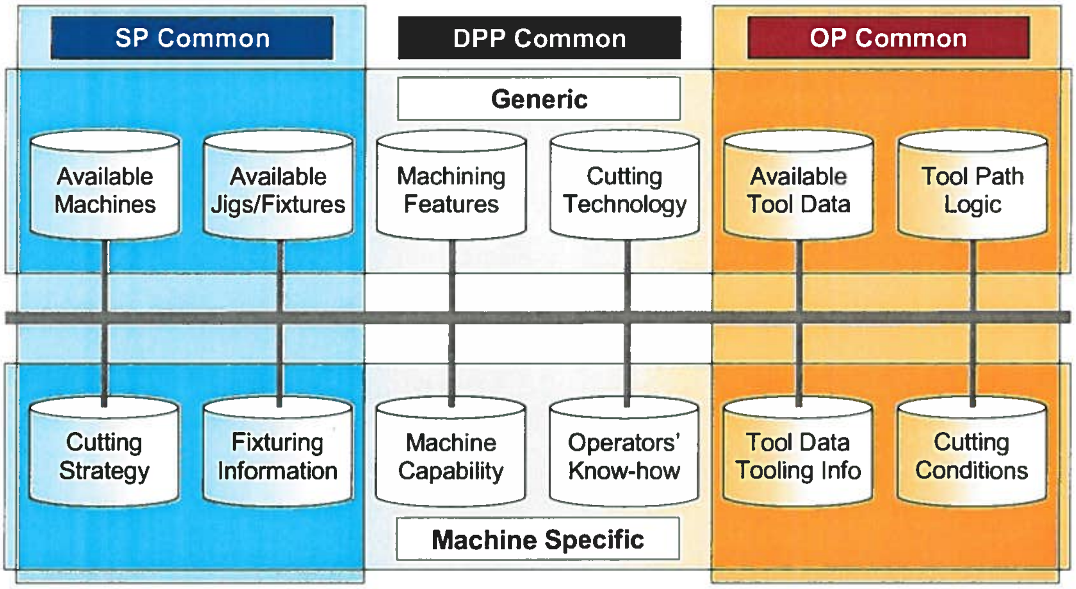


Figure 6. Distributed database.

3.5 Function Block-Based NC Control

The distributed process planning functions range from shop-level supervisory planning to machine-level operation planning. A generated generic process plan needs to be dispatched to a dedicated machine controller in a mutually understandable language. On the other hand, the reconfigurable shop floor machining requires a control model that is the unique combination of real-time control, distributed control, even-driven control, and intelligent control. Traditionally, process plans are hard-coded line by line in the so-called M and G codes (ISO 6983 and EIA RS274). Being more than 30 years old, these standards severely restrict the range of process planning information that can be communicated to a machine tool controller. An open and adaptive CNC controller language is required to replace the aging M and G codes.

In this research, function blocks are chosen for machining process encapsulation and CNC control execution. The concept of function blocks is described in the IEC-61499 specification [73], as an emerging IEC standard for distributed industrial processes and control systems. It is based on an explicit event driven model and provides for data flow and finite state automata based control. It is relevant to the distributed NC machining through machining process encapsulation and NC execution. The event driven model of function blocks is used for both DPP implementation and future extension towards integration with the dynamic shop floor scheduling. Being an atomic distributable and executable NC control function unit, a function block can encapsulate a set of machining process data (such as slot roughing, pocket finishing, and hole drilling, etc.) for a given machining feature. Figure 7 illustrates the internal structure of both the basic (left) and the composite (right) function blocks.

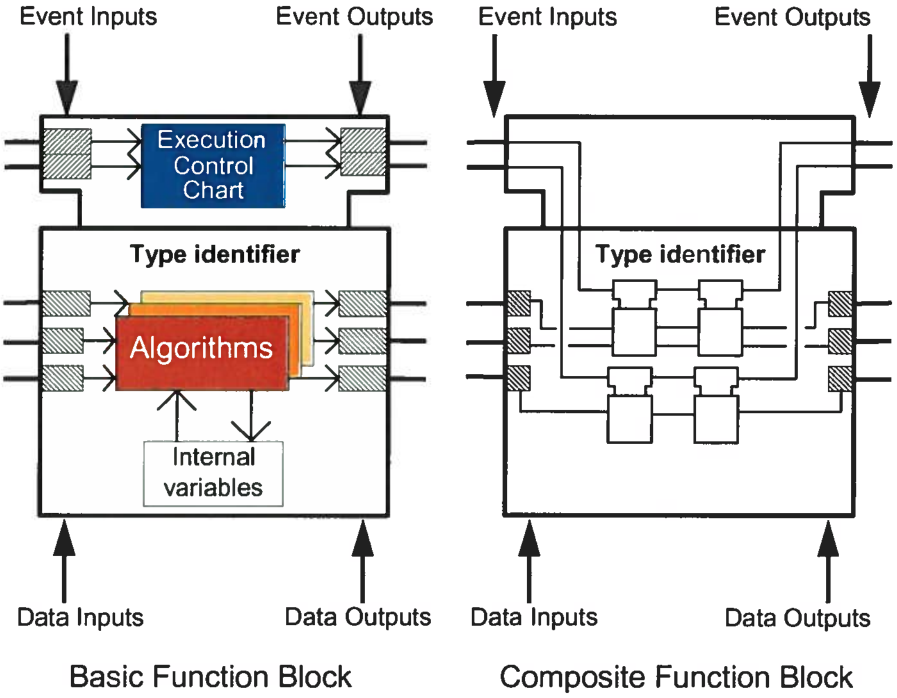


Figure 7. Function block structure.

Each function block may comprise of an individual, named copy of data structure specified by its function block type, which persists from one invocation to the next. It, especially the basic function block, can have multiple outputs and can maintain internal hidden state information. Each basic function block can also have more than one algorithm. This means that a function block can generate different outputs even if the same inputs are applied. This fact is of vital importance for automatic cutting condition generation, after a function block has been distributed to a CNC controller, by changing the internal hidden state of the function block or using different algorithms. For example, the same function block designed for *pocket-roughing* may be shared by two different milling machines with different cutting conditions, simply by choosing the appropriate algorithm of the function block. Through inheritance, a *pocket-milling* function block can be reused to specify for *pocket-roughing* or *pocket-finishing*, depending on the message received and the algorithm used. For the ease of NC control code distribution, reuse, fault recovery, and dynamic shop floor reconfiguration, process planning data are best encapsulated into function blocks.

Similarly to object-oriented definitions, a function block type can be considered as a class, and a particular function block is the instance of that class. The functionality of multi-inheritance, overriding, and dynamic binding can be used to form a composite function block. However, different from the object-oriented approach, the behavior of a function block is controlled internally by a finite state machine whose operation can be represented by an execution control chart (ECC). As shown in Fig.7, The event flow of a function block determines the scheduling and execution of an embedded machining operation specified by the algorithms (methods) in basic function blocks. In terms of process encapsulation, basic function blocks encapsulate both the data and functions, while composite function blocks only encapsulate basic function blocks. A combination of both basic and composite function blocks is used to define the needed machining operations generated by our DPP system.

3.6 NC Control Application Generation

A function block-based NC control application consists of a network, in which the nodes are meta function blocks and the branches are data and event connections. Figure 8 shows how a function block-based application is generated for the case of pocket-milling. Typically, the necessary milling information can be extracted from meta working steps that define the generic machining technology and strategy. Each meta working step (WS) is predefined in a cutting technology database. It is an abstractive machining process containing machining data in a hierarchy, so that its subclass WS can easily inherit these attributes as needed. The WS's are finally selected and wrapped by meta function blocks for distribution and execution. A well-defined control application has correct event and data flows among the function blocks. The event flow determines the sequence of NC operations specified by individual WS's. Events and data always flow into the left side of each function block, while generated data and events flow out on the right side. Applications comprising interconnected function blocks can be distributed and utilized in machine controllers, as illustrated in Fig.8.

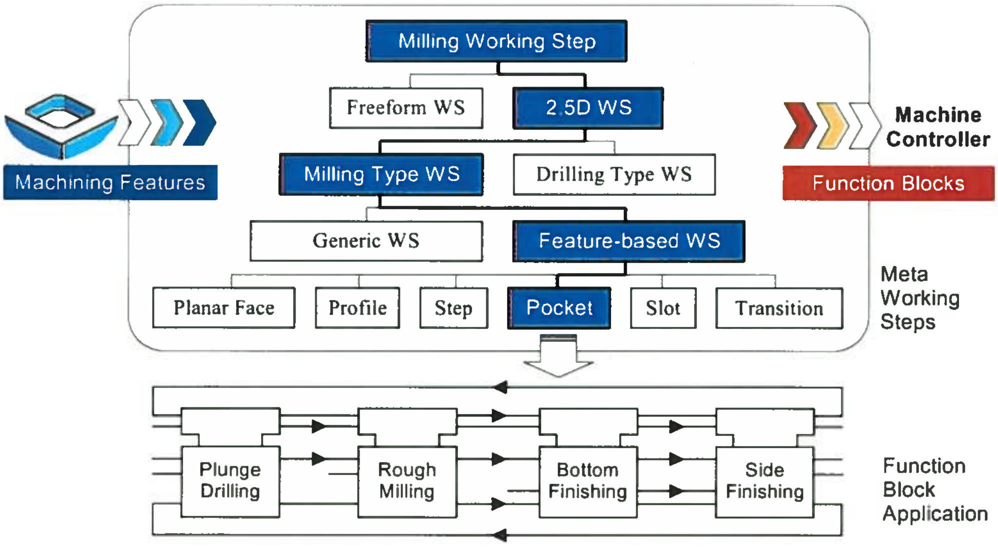


Figure 8. Function block-based NC control application.

The encapsulation of machining processes data enables and facilitates transparent distribution of NC codes over a set of distributed machine controllers. As shown in Fig.9, an NC application may reside completely within a single machine controller or may be distributed over several devices, to accomplish the entire machining operation. The communication between function blocks can be realized by a well-established message passing mechanism under real-time constraints to keep machining tasks on schedule. This capability and others are crucial for the next generation of NC machining in the ever-changing reconfigurable shop floor environment.

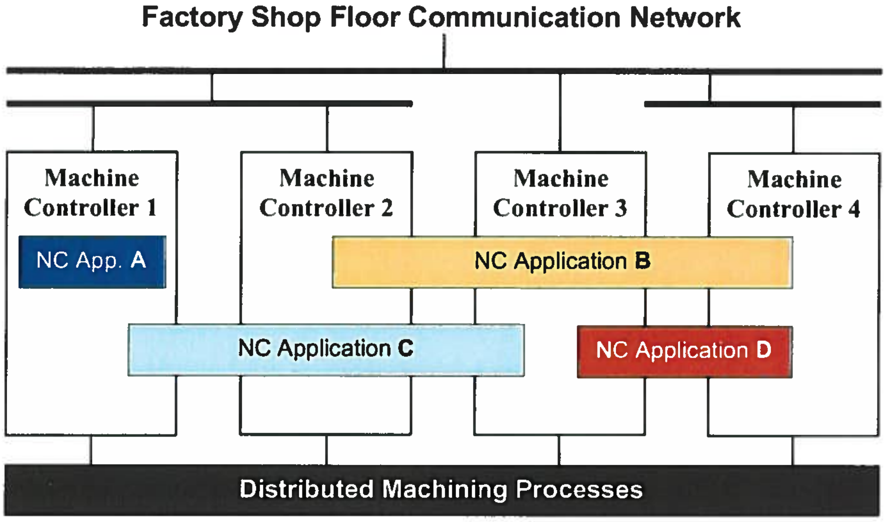


Figure 9. Distribution of NC control applications.

3.7 Process Plan Execution

Function blocks not only make possible the encapsulation of machining processes but also provide support for process plan execution. The execution control chart (ECC) for the state machine of a function block can be used to control their internal algorithms (embedded machining process plans). Figure 10 shows a typical example of an ECC for pocket roughing. Being a (representation of a) finite state machine, an ECC is made up of EC states, EC transitions, and EC actions. The initial EC state, *START* in this case, cannot have any EC actions associated with it. The occurrence of an event input, such as *PI_Init* and *PI_Cut*, causes the ECC to be invoked and the input variables (tool#, key_para, etc.) to be mapped. The EC transitions use a Boolean combination of conditions that may be comprised of event inputs, input variables, output variables, and internal variables. A triggered EC transition causes a change of EC state and this leads to the execution of an associated EC action, *Init* or *Cut* in this case. The EC action then sends out an event, *PO_Init* or *PO_Cut*, upon completion [70].

In the near future, programmable controllers will either provide function blocks as part of their device firmware or provide function block libraries from which function blocks can be selected and downloaded to an end user's application [74]. In the case of CNC control, vendor or user extensions to the standard M and G codes (ISO 6983 or EIA

RS274) could be encapsulated by wrapper function blocks and distributed to end-users to be directly utilized by their machining operations. In addition, function block-based process plan encapsulations enable and facilitate transparent distribution and dynamic scheduling of machining processes over a group of machine tools. As shown in Fig.11, a combined process plan represented by three function blocks could be distributed over three machine tools, depending on the process plan and the availability of each machine, with no interruption to the entire machining operation. The communication between the distributed function blocks can be realized by their flexible event and data flows. This process plan execution approach enables run-time process planning and dynamic rescheduling being done in parallel, whenever a run-time exception occurs, for rapid fault recovery. It is true and practical in a reconfigurable shop floor where a new machine is added or an existing machine is reconfigured.

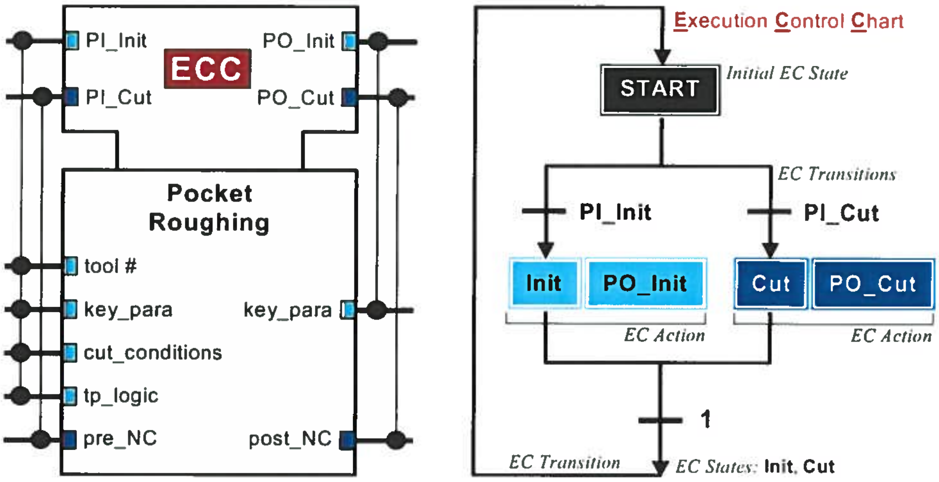


Figure 10. Execution control mechanism.

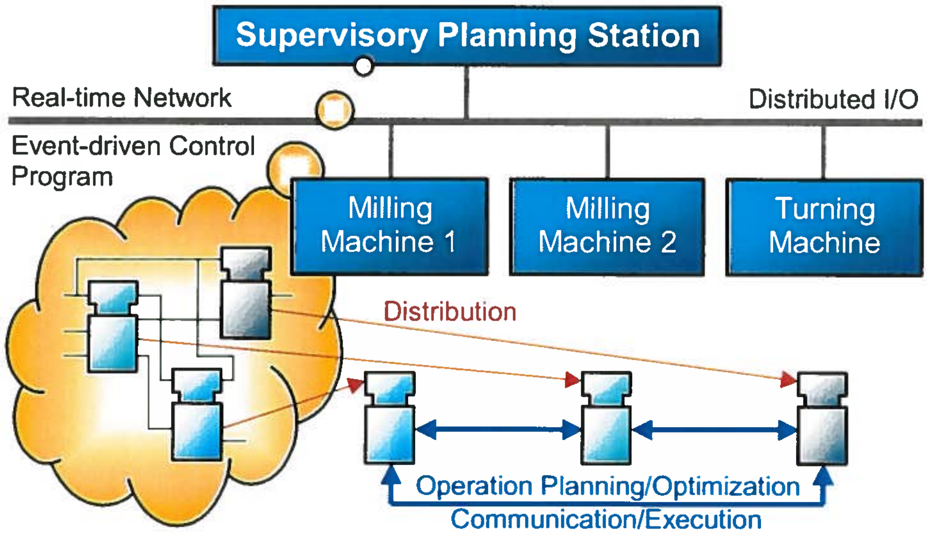


Figure 11. Process plan execution.

4. ARCHITECTURE DESIGN

Today's manufacturing shop floors, characterized by large variety of products in small batch sizes, require dynamic process planning capabilities that are responsive and adaptive to the dynamic changes of production capacity and functionality. Traditional CAPP systems are weak in handling such dynamism. A process plan that is generated for a specific machine may have to be regenerated when the machine is unavailable or broken. Such repetitive planning tasks happen frequently in dynamic shop environment. Targeting the dynamism of process planning, this report reveals a detailed architecture design based on our pioneer research work [75] and the latest development.

4.1 System Architecture

Figure 12 illustrates a generic extended system architecture required for the proposed DPP approach. It consists of three major subsystems for design, planning, and control, which share one dynamic database. Our focus is at the planning and control part, assuming the product data is machining feature-based and ready to be used as system inputs. A real-time network that links with a secured factory network enables the functionality of each module. There is more than one OP (operation planning) modules, each of which runs in an intelligent open CNC controller, compared with the only one SP (supervisory planning) module residing at the shop floor level. The concept of *design-for-machining* is thus actualized by the new architecture through continuous machinability checking, from design through process planning to machine control.

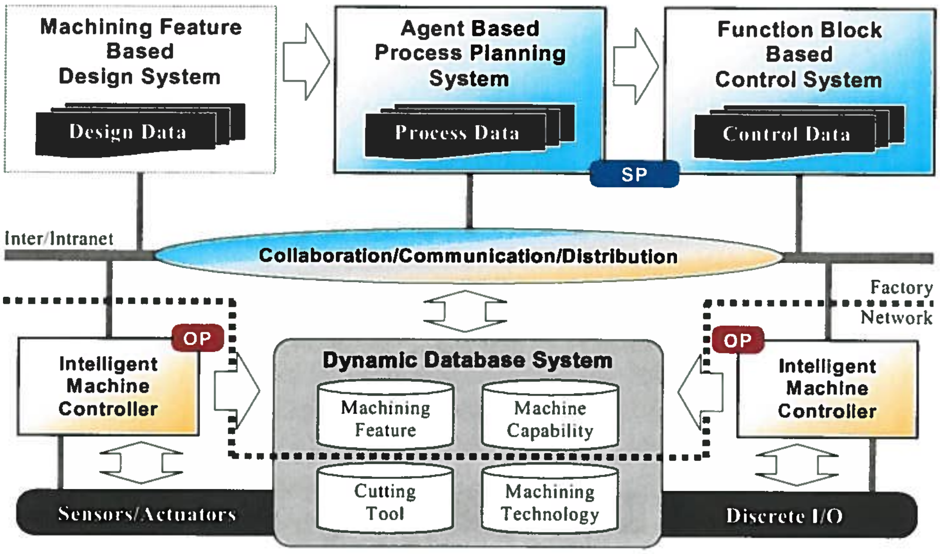


Figure 12. Overall system architecture.

4.2 Two-Layer Structure

As outlined earlier, the DPP approach is based on a two-layer structure: the high-level supervisory planning and low-level operation planning. The former focuses on product data analysis, machining feature parsing, machine/jig/fixture selection, and machining sequence planning for a given product. The latter considers the detailed operations for each machining features of the product, including cutter selection, cutting condition assignment, and tool path planning, before appropriate NC control codes are generated. Figure 13 gives the entire hierarchy of the two-layer process planning.

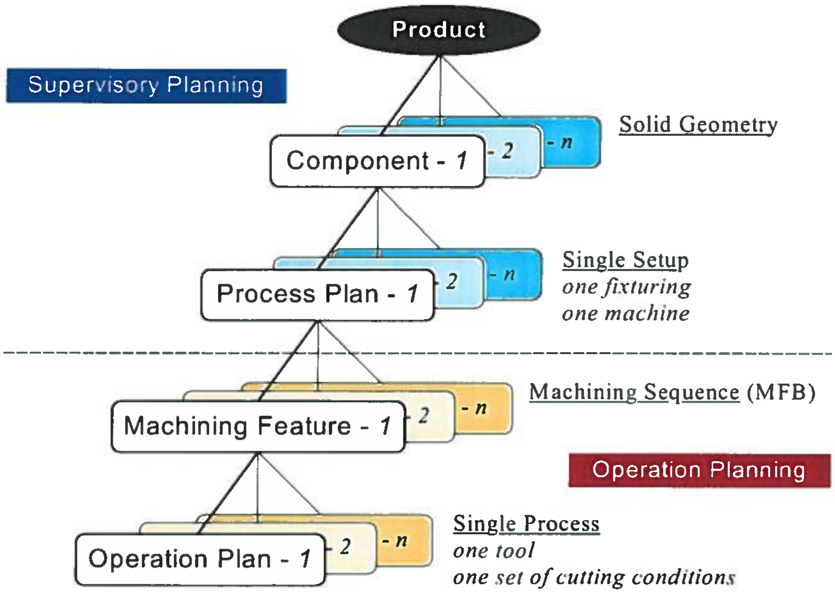


Figure 13. Two-layer structure.

At the supervisory planning level, the output of the module is a set of well-connected meta function blocks with the sequenced machining features and process plans embedded. Each process plan is generated for one single setup – *one fixturing on one machine*. At the operation planning level, the meta function blocks are specialized by adding machine specific data (tool type, tool #, cutting conditions, and tool path, etc.). Each of the generated operation plans describes the detailed machining operation for one single process – *one cutting tool under one set of cutting conditions*.

4.3 Functional Architecture

The proposed DPP is designed to be a dynamic system with intelligent functionality across CNC controllers. The functional requirements of a DPP and its architecture are summarized and shown in Fig.14, supported by distributed databases.

An intelligent CNC controller should be able to interact with actual machining processes through sensors, actuators, and other discrete I/Os at hardware level. It takes meta function blocks as inputs, performs operation planning, and controls motions of axes. Moving the functions of operation planning to individual CNC controllers helps in separating the hardware dependency of control codes and provides clear location transparency. With this architecture, it is possible to have another similar machine pick up the responsibility and carry out the work of the machine that fails, for quick fault recovery using the same generic process plan and runtime OP processing. At the shop floor level, only the generic functions of a DPP (e.g. data parsing, machine selection, and sequence planning) are assigned to the SP module. The data provided by the SP module is also needed for shop floor scheduling and execution control (process plan distribution). The decision-makings of a DPP, especially inside of the SP module, are facilitated by the multi-agent negotiation described below.

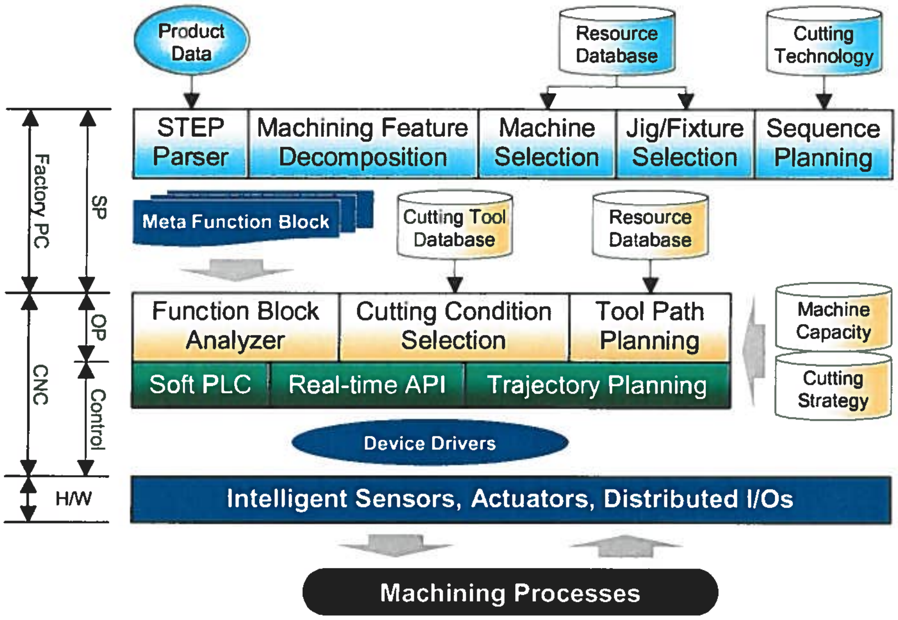


Figure 14. Functional architecture.

4.4 Software Architecture

The software architecture of the DPP is shown in Fig.15, where the SP Kernel is a heterogeneous multi-agent system. An agent is an autonomous software entity that inter-operates asynchronously with other agents through a well-defined messaging mechanism [72]. A heterogeneous multi-agent system involves a group of dissimilar agents who are knowledgeable in their local domains and share the responsibility of achieving the overall system objectives through negotiation. Relevant agents for a particular SP task form or are formed into clusters that are coordinated and arbitrated by mediator agents (e.g. SP planner, machine facilitator, feature facilitator, etc.). These mediators possess the cross-domain knowledge and meta-level rules to facilitate cooperation among multiple agents. For example, a feature agent cooperates with the *DB* (database) *Agent* through the *Feature Facilitator*, and indirectly gets access to machining feature knowledge base for the ease of feature parsing and machinability assessment. Other major agents include *SP* (sequence planning) *agent*, *MT* (machining technology) *agent*, *machine agents*, and *tool agents*. All agents use KQML (Knowledge Query Manipulation Language) as negotiation protocol.

In addition to the SP kernel, a lightweight OP kernel is designed for low-level operation planning running in open CNC controllers. It utilizes machining feature-based reasoning for decision-makings. The software architecture also includes utilities for function block generation, resource management, execution control, and interfaces to end-users and machine system itself. The architecture design is partially based on the IEC-61499 function block standards [73]. It requires the CNC controllers being open architecture-based so that the OP module of the DPP can be dispatched.

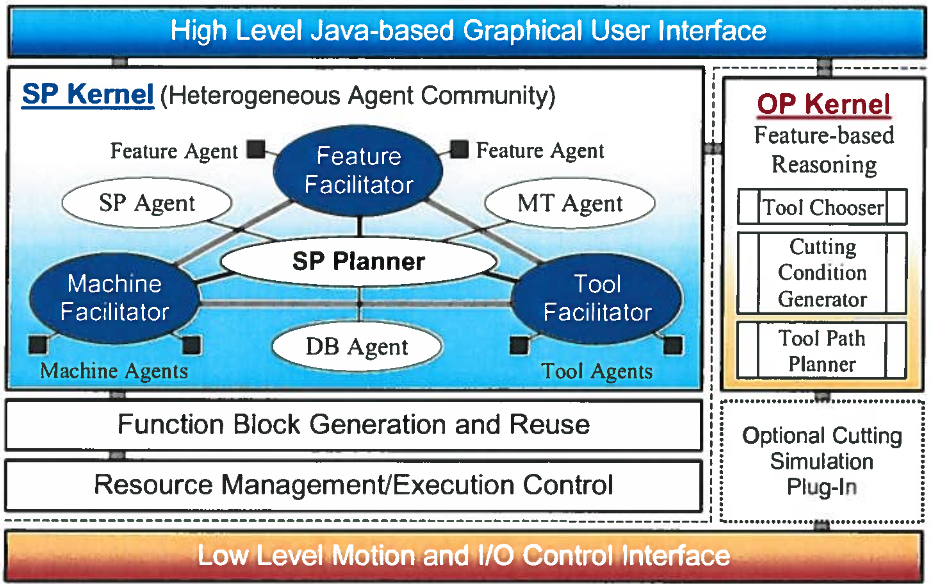


Figure 15. Software architecture.

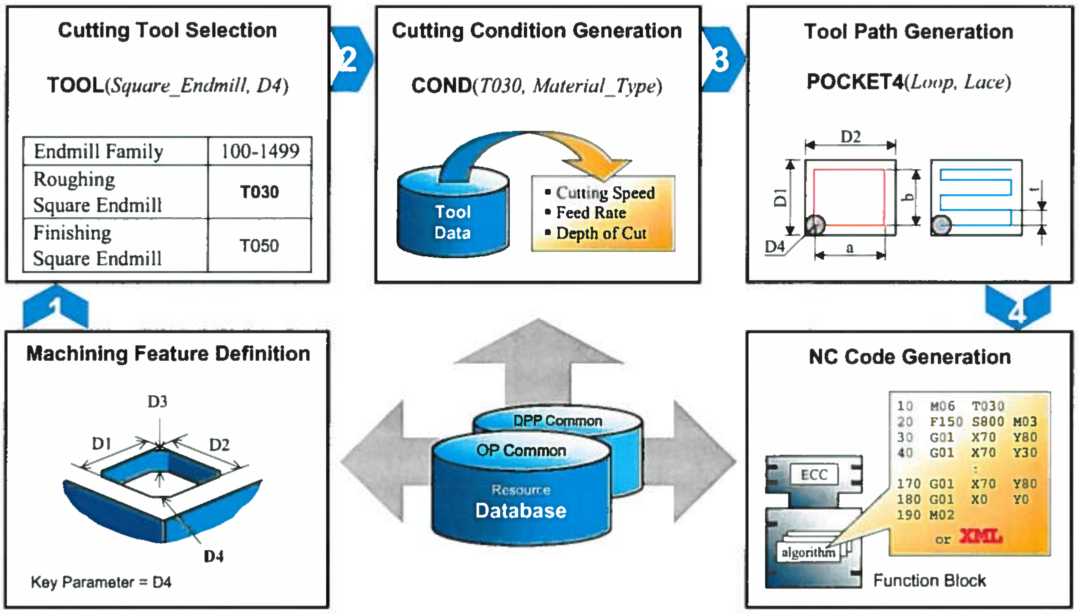
5. SYSTEM IMPLEMENTATION ISSUES

5.1 Integrated Approach

The concept of DPP and its architecture design are based on three elements – agents, machining features, and function blocks – as the enabling technologies for our DPP system implementation. This integrated approach includes self-contained agent-based decision-making, feature-based reasoning, and function block-based control. The high-level supervisory planning, especially the machining sequence planning and machine selection, will largely rely on the function of agent-based decision-making, whereas the feature-based reasoning will be used for low-level operation planning including cutting tool selection, cutting condition generation, and tool path planning. Function blocks, on the other hand, will be used as a new controller language for machining process data encapsulation, distribution, dynamic rescheduling, and execution control.

As agent-based decision-making and function block-based NC control are described separately in section 3, this section will focus on feature-based reasoning through an example of 4-side pocket milling. The concept of feature-based reasoning used mainly in the OP module is illustrated in Fig.16. Its procedure is straightforward, if the needed data stored in the OP- and DPP-common database are retrievable.

Each machining feature possesses a key parameter for tool selection (*D4* in the case of a 4-side pocket shown in Fig.16) together with other non-dominant parameters for its geometry definitions. It also has a set of defined technical information loosely coupled with it. This includes recommended tool type and tool path generation logic. The key parameter of a feature is used to determine the tool diameter of the recommended tool type. In this example, a square end mill *T030* is selected for rough milling based on *D4*. If the workpiece material of the product is known, the cutting conditions (cutting speed, feed rate, depth of cut, and step over, etc.) can easily be retrieved from a coupled cutting tool database. Accordingly, the tool path for the rough milling can be generated based on the tool data, cutting conditions, and the suitable cutting strategy, using predefined tool path generation logic. Once the cutting conditions and tool path are found, the final machine specific control codes (for roughing in this case) are generated and encapsulated as one of the algorithms in the function block *Pocket Roughing* for NC execution. Since the operation planning with feature-based reasoning is done in an open architecture CNC controller, it is possible to achieve a local optimal plan for the selected machine tool based on its capacity and dynamics. In the case of a machine failure, the same *Pocket Roughing* function block may be redirected to another machine of similar functionality, with its encapsulated machining data overridden partially by the new CNC controller for local optimization. If equipped with in-line inspection capability, a CNC controller can easily fulfill error compensation by adjusting the control codes in a function block to improve the finished surface quality.



Endmill Family	100-1499
Roughing Square Endmill	**T030**
Finishing Square Endmill	T050

```
10   M06   T030
20   F150  S800  M03
30   G01   X70   Y80
40   G01   X70   Y30
           :
170  G01   X70   Y80
180  G01   X0    Y0
190  M02
```

Figure 16. Feature-based reasoning for operation planning.

5.2 System Implementation

This is an on-going research project. A complete DPP system includes the following five software modules: 1) Feature Parsing Module, 2) Function Block Designer, 3) Agent-Based Decision-Making Engine, 4) Execution Control Module, and 5) FB Processor. Our goal of system implementation is to ensure that a DPP-generated process plan will possess portability, scalability, convertability, and reusability, and to be able to apply the DPP into the future RMS (reconfigurable manufacture system) framework.

Figure 17 shows the five software modules and a set of distributed databases for the DPP system implementation. The functionality of the DPP covers data and information processing from product design data (inputs) to NC control codes (outputs). STEP data (ISO-10303, Standard for Product Data Exchange) with meta process plans (AP213) and machining feature (AP214, AP224) information embedded will be the input data to the DPP system. It is followed by the feature parsing (macro feature sequencing and feature interaction detection), machine selection, process sequencing, and function block design, supported by an agent-based decision-making engine in the SP planning phase. The execution control module is responsible for shop floor resource coordination and meta function block distribution, which communicates with other shop floor planning systems (e.g. resource planning/scheduling/maintenance, and job re-routing, etc.). As mentioned in section 3, it is the responsibility of each open CNC controllers to fulfill the OP planning with seamless control of dedicated machine tools.

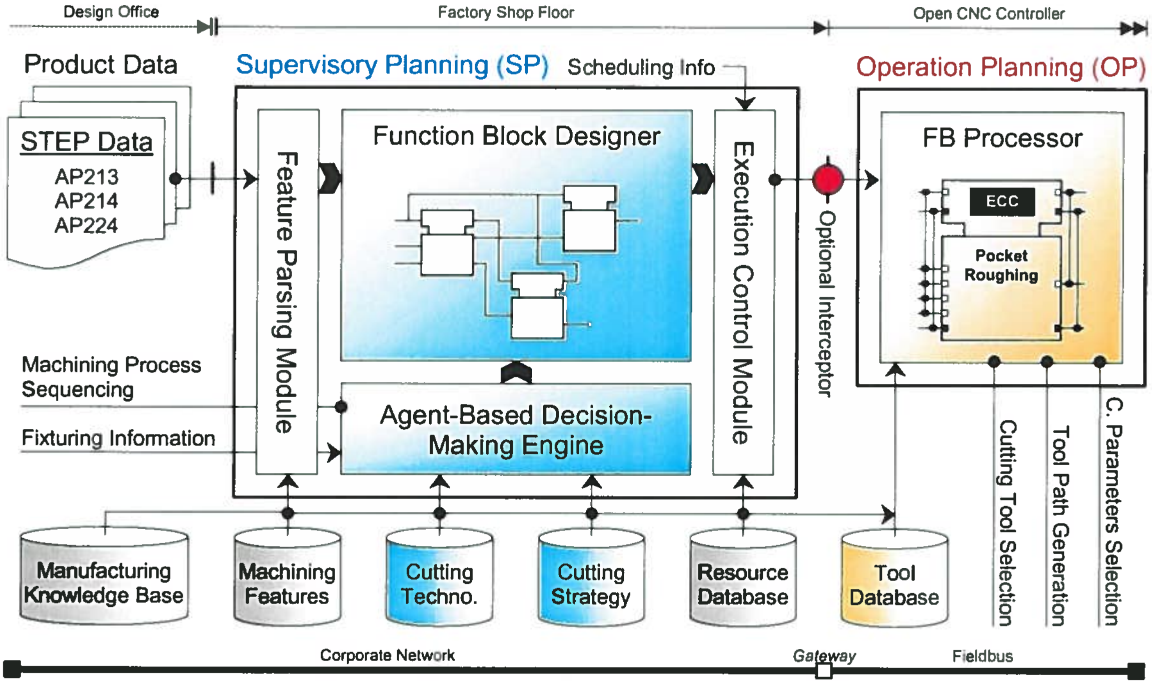


Figure 17. Software modules for DPP system implementation.

Currently, a prototype of the DPP system is under development at National Research Council of Canada. The core of the system is being implemented using Visual C++ and Java. The phase 1 of the system implementation includes: 1) R&D of an agent-based decision-making engine, 2) development of an iconic function block designer, and 3) research on process planning and scheduling integration at SP level. The phase 2 will conduct: 1) R&D on operation planning and 2) DPP system integration.

For the sake of future Internet connectivity, an optional high-level web user interface is planned for implementation using Java for remote process planning and NC execution. It is useful to a distributed manufacturing environment where production engineers can fulfill their duties with high location transparency.

6. CONCLUSIONS

In response to the ever-changing machining shop floor environments, this research proposes a new approach for distributed process planning (DPP) based on a design-for-machining concept. Within the context, the relevant aspects of machining feature based reasoning, multi-agent based decision-making, and function block-based CNC control are described in this report. Machining features are used as information carriers from product design through process planning to machine control, while agents are adopted for intelligent decision-makings within the distributed process planning. Function blocks, on the other hand, are chosen as new controller language to encapsulate high-level machining process plans for cross-platform execution and for quick fault recovery. The detailed DPP architecture design is presented based on a two-layer structure for both supervisory planning and operation planning. The proposed DPP approach is expected to improve the responsiveness, flexibility, and productivity of machining shop floors. The dynamism of process planning enabled by the DPP methodology is crucial to the next generation reconfigurable manufacturing system (RMS), and it is beneficial to apply the DPP to the RMS framework.

The challenges of this research can be summarized as follows:

(1) To realize the dynamism and responsiveness of process plans through a two-layer structure that can separate generic process data from machine-specific data. The machine-specific data can be modified quickly according to shop floor changes.

(2) To increase the openness and reconfigurability of machining shop floors by using function blocks as inter-controller language. With generic process plan embedded, a function block can be directed freely according to shop floor configuration.

(3) To achieve local optimization of a process plan through machine level operation planning. As the controller of a machine is knowledgeable of its dynamics, run-time status, and machine's capability, an optimal plan can be reached easily.

(4) To develop distributed intelligence for the DPP system that can facilitate numerous decision-makings during process planning. An agent-based approach is adopted to implement a core decision-making engine for machining process sequencing.

(5) To seek seamless integration between DPP and scheduling. This integration can be done early at supervisory planning level where machine selection is conducted, so that the operation planning can be carried out by a scheduled machine.

(6) To provide users with high reliability and ease of use. These issues are addressed by developing an intelligent iconic user interface, through which users can design their process plans as function block networks. A web-based GUI is an option.

REFERENCES

[1] L. Wang, An Approach to Collaborative Design and Intelligent Manufacturing, *Proceedings of SCI'99/ISAS'99*, Vol.7, pp.431-437, 1999.

[2] M. G. Mehrabi, A. G. Ulsoy and Y. Koren, Reconfigurable Manufacturing Systems: Key to Future Manufacturing, *Journal of Intelligent Manufacturing*, Vol.11, pp.403-419, 2000.

[3] P. T. Chang and C. H. Chang, An Integrated Artificial Intelligent Computer-Aided Process Planning System, *International Journal of Computer Integrated Manufacturing*, Vol.13, No.6, pp.483-497, 2000.

[4] B. W. Niebel, Mechanized Process Selection for Planning New Designs, *ASTME* paper 737, 1965.

[5] C. H. Link, CAPP, CAM-I Automated Process Planning System, *Proceedings of the 1976 NC Conference*, Arlington, Texas, USA, 1976.

[6] A. Houtzeel, The MYCLASS System, *Proceedings of CAM-I's Executive Seminar – Coding, Classification, and Group Technology for Automated Planning*, CAM-I, Inc., Arlington, Texas, USA, 1976.

[7] L. Alting and H. Zhang, Computer Aided Process Planning: the State-of-the-Art Survey, *International Journal of Production Research*, Vol.27, No.4, pp.553-585, 1989.

[8] H. B. Marri, A. Gunasekaran and R. J. Grieve, Computer-Aided Process Planning: A State of the Art, *International Journal of Advanced Manufacturing Technology*, Vol.14, pp.261-268, 1998.

[9] R. Weill, G. Spur and W. Eversheim, Survey of Computer Aided Process Planning Systems, *Annals of CIRP*, Vol.31/2, pp.539-552, 1982.

[10] H. J. Steudel, Computer-Aided Process Planning: Past, Present and Future, *International Journal of Production Research*, Vol.22, No.2, pp.253-266, 1984.

[11] W. Eversheim and J. Schulz, CIRP Technical Reports: Survey of Computer Aided Process Planning Systems, *Annals of CIRP*, Vol.34/2, pp.607-614, 1985.

[12] R. A. Wysk, T. C. Chang and I. Ham, Automated Process Planning System: an Overview of Ten Years of Activities, *1st CIRP Working Seminar*, Paris, January, 1985.

[13] H. A. Elmaraghy, Evolution and Future Perspectives of CAPP, *Annals of CIRP*, Vol.42/2, pp.739-751, 1993.

[14] I. B. H. Lee, B. S. Lim and A. Y. C. Nee, Knowledge-based Process Planning System for the Manufacture of Progressive Dies, *International Journal of Production Research*, Vol.31, No.2, pp.251-278, 1993.

[15] D. N. Sormaz and B. Khoshnevis, Process Planning Knowledge Representation Using an Object-Oriented Data Model, *International Journal of Computer Integrated Manufacturing*, Vol.10, No.1-4, pp.92-104, 1997.

[16] Y. Zhang, S. C. Feng, X. Wang, W. Tian and R. Wu, Object Oriented Manufacturing Resource Modeling for Adaptive Process Planning, *International Journal of Production Research*, Vol.37, No.18, pp.4179-4195, 1999.

[17] F. Zhang, Y. F. Zhang and A. Y. C. Nee, Using Genetic Algorithms in Process Planning for Job Shop Machining, *IEEE Transactions on Evolutionary Computation*, Vol.1, No.4, pp.278-289, 1997.

[18] N. Morad, and A. Zalzala, Genetic Algorithms in Integrated Process Planning and Scheduling, *Journal of Intelligent Manufacturing*, No.6, pp.169-179, 1999.

[19] C. R. Devireddy and K. Ghosh, Feature-Based Modeling and Neural Network-Based CAPP for Integrated Manufacturing, *International Journal of Computer Integrated Manufacturing*, Vol.12, No.1, pp.61-74, 1999.

[20] L. Monostori, Z. J. Viharos and S. Markos, Satisfying Various Requirements in Different Levels and Stages of Machining Using One General ANN-Based Process Model, *Journal of Materials Processing Technology*, Vol.107, pp.228-235, 2000.

[21] K. Y. Lee and M. Y. Jung, Flexible Process Sequencing Using Petri Net Theory, *Computers & Industrial Engineering*, Vol.28, No.2, pp.279-290, 1995.

[22] Y.-J. Tseng and S. B. Joshi, Recognizing Multiple Interpretations of Interacting Machining Features, *Computer-Aided Design*, Vol.26, No.9, 1994.

[23] C. Hayes and P. Wright, Automatic Process Planning: Using Feature Interactions to Guide Search, *Journal of Manufacturing Systems*, January, 1989.

[24] Y. S. Lee, D. M. Dhaval and P. Krishnaswami, Application of Virtual Convex Polytope in Transforming Protrusions for Automated Process Planning, *Computer & Industrial Engineering.* Vol.36, No.1, pp.67-96, 1999.

[25] Z. Gu, Y. F. Zhang and A. Y. C. Nee, Identification of Important Features for Machining Operations Sequence Generation, *International Journal of Production Research*. Vol.35, No.8, pp.2285-2307, 1997.

[26] A. Chep and L. Tricarico, Object-oriented Analysis and Design of a Manufacturing Feature Representation, *International Journal of Production Research*. Vol.37, No.10, pp.2349-2376, 1999.

[27] X. G. Ming, K. L. Mak and J. Q. Yan, A PEDS/STEP-Based Information Model for Computer-Aided Process Planning, *Robotics and Computer-Integrated Manufacturing*. Vol.14, No.5-6, pp.347-361, 1998.

[28] L. Wang, Integrated Design-to-Control Approach for Holonic Manufacturing Systems, *Robotics and Computer Integrated Manufacturing*, Vol.17, pp.159-167, 2001.

[29] L. Wang and D. H. Norrie, Process Planning and Control in a Holonic Manufacturing Environment, *Journal of Applied Systems Studies*, Vol.2, No.1, pp.106-126, 2001.

[30] K. Iwata and Y. Fukuda, KAPPS: Know-how and Knowledge Assisted Production Planning System in the Machining Shop, *Proceedings of the 19th CIRP International Seminar on Manufacturing Systems*, Penn. State University, 1987.

[31] F. Giusti, M. Santochi and G. Dini, KAPLAN: a Knowledge-based Approach to Process Planning of Rotational Parts, *Annals of CIRP*, Vol.38/1, pp.481-484, 1989.

[32] G. C. Vosniakos and B. J. Davies, Knowledge-Based Selection and Sequencing of Hole-Making Operations for Prismatic Parts, *International Journal of Advanced Manufacturing Technology*, Vol.8, pp.9-16, 1993.

[33] J. A. Stori and P. K. Wright, A Knowledge-Based System for Machining Operation Planning in Feature Based, Open-Architecture Manufacturing, *Proceedings of ASME Design Engineering Technical Conference*, Irvine, CA, 1996.

[34] X. G. Ming, K. L. Mak and J. Q. Yan, A Hybrid Intelligent Inference Model for Computer Aided Process Planning, *Integrated Manufacturing Systems*, Vol.10, No.6, pp.343-353, 1999.

[35] V. V. Kryssanov, A. S. Kleshchev, Y. Fukuda and K. Konishi, Building a Logical Model in a Machining Domain for CAPP Expert Systems, *International Journal of Production Research*, Vol.36, No.4, pp.1075-1089, 1998.

[36] A. C. Lin and C.-L. Wei, Automated Selection of Cutting Tools Based on Solid Models, *Journal of Materials Processing Technology*, Vol.72, pp.317-329, 1997.

[37] K. O. Edalew, H. S. Abdalla and R. J. Nash, A Computer-Based Intelligent System for Automatic Tool Selection, *Materials and Design*, Vol.22, pp.337-351, 2001.

[38] T. Lim, J. Corney, J. M. Ritchie and D. E. R. Clark, Optimizing Tool Selection, *International Journal of Production Research*, Vol.39, No.6, pp.1239-1256, 2001.

[39] B. Arezoo, K. Ridgway and A. M. A. Al-Ahmari, Selection of Cutting Tools and Conditions of Machining Operations Using an Expert System, *Computers in Industry*, Vol.42, pp.43-58, 2000.

[40] P. G. Maropoulos and R. P. Baker, Integration of Tool Selection with Design, Part1: Feature Creation and Selection of Operations and Tools, *Journal of Materials Processing Technology*. Vol.107, No.1-3, pp.127-134, 2000.

[41] P. G. Maropoulos, R. P. Baker and K. Y. G. Paramor, Integration of Tool Selection with Design, Part2: Aggregate Machining Time Estimation, *Journal of Materials Processing Technology*. Vol.107, No.1-3, pp.135-142, 2000.

[42] I. Yellowley, R. J. Seethaler and F. W. Yeung, The Integration of Process Planning, Monitoring and Control in a Machine Tool Environment, *Proceedings of SPIE, the International Society for Optical Engineering*, Vol.3832, pp.38-47, 1999.

[43] J. Y. Jung and R. S. Ahluwalia, Prismatic Part Feature Extraction and Feature-Based Tool Path Selection, *Journal of Design and Manufacturing*, Vol.3, pp.1-19, 1993.

[44] R. M. Boogert, H. J. Kals and F. J. van Houten, Tool Paths and Cutting Technology in Computer-Aided Process Planning, *International Journal of Advanced Manufacturing Technology*, Vol.11, pp.186-197, 1996.

[45] N. Balasubramanian and S. Raman, Modeling Gradual Process Variables in Path Planning, *Technical paper of Society of Manufacturing Engineers*, 1998.

[46] K. Hashmi, M. A. El Baradie and M. Ryan, Fuzzy Logic Based Intelligent Selection of Machining Parameters, *Computers & Industrial Engineering*, Vol.35, No.3-4, pp.571-574, 1998.

[47] S. H. Yeo, B. K. A. Ngoi and H. Chen, Process Sequence Optimization Based on a New Cost-Tolerance Model, *Journal of Intelligent Manufacturing*, Vol.9, pp.29-37, 1998.

[48] I. T. Kim and H. W. Suh, Optimal Operation Grouping and Sequencing Technique for Multistage Machining Systems, *International Journal of Production Research*. Vol.36, No.8, pp.2061-2081, 1998.

[49] S. K. Ong and A. Y. C. Nee, Fuzzy-Set-Based Approach for Concurrent Constraint Set-Up Planning, *Journal of Intelligent Manufacturing*, Vol.7, No.2, pp.107-120, 1996.

[50] R.-R. Wu and H.-M. Zhang, Object-Oriented and Fuzzy-Set-Based Approach for Set-Up Planning, *International Journal of Advanced Manufacturing Technology*, Vol.14, No.6, pp.406-411, 1998.

[51] F. Kimura, High-Quality Product Realization through the Tight Integration of Product Design and Process Planning, *Proceedings of CIRP International Workshop on CAPP*, pp.199-208, Hanover, September, 1989.

[52] M. Srinivasan and P. Sheng, Feature Based Process Planning in Environmentally Conscious Machining – Part 1: Microplanning, *Robotics and Computer Integrated Manufacturing*, Vol.15, No.1, pp.257-270, 1999.

[53] M. Srinivasan and P. Sheng, Feature Based Process Planning in Environmentally Conscious Machining – Part 2: Macroplanning, *Robotics and Computer Integrated Manufacturing*, Vol.15, No.1, pp.271-281, 1999.

[54] B. Khoshnevis, D. N. Sormaz and J. Y. Park, An Integrated Process Planning System Using Feature Reasoning and Space Search-based Optimization, *IIE Transactions*, Vol.31, pp.597-616, 1999.

[55] W. H. ElMaraghy, Integrating Assembly Planning and Scheduling – CAPP Related Issues, *Annals of CIRP*, Vol.41/1, 1992.

[56] Y. Tu, X. Chu and W. Yang, Computer-Aided Process Planning in Virtual One-of-a-Kind Production, *Computers in Industry*, Vol.41, pp.99-110, 2000.

[57] C. S. Smith and P. K. Wright, CyberCut: A World Wide Web Based Design-to-Fabrication Tool, *Journal of Manufacturing Systems*, Vol.15, No.6, pp.432-442, 1996.

[58] D. N. Sormaz, J. Arumugam, P. Borse, S. Jain and S. Thirupalli, Distributed System for Modeling and Visualization of Process Plans, *Proceedings of the 5th International Conference on Engineering Design & Automation*, pp.163-168, Las Vegas, 2001.

[59] H. G. Park and J. M. Baik, Enhancing Manufacturing Product Development through Learning Agent System over Internet, *Computers & Industrial Engineering*, Vol.37, pp.117-120, 1999.

[60] J. Sun, Y. F. Zhang and A. Y. C. Nee, A Distributed Multi-Agent Environment for Product Design and Manufacturing Planning, *International Journal of Production Research*, Vol.39, No.4, pp.625-645, 2001.

[61] F. L. Zhao and P. S. Y. Wu, A Cooperative Framework for Process Planning, *International Journal of Computer Integrated Manufacturing*, Vol.12, No.2, pp.168-178, 1999.

[62] F. L. Zhao, S. K. Tso and P. S. Y. Wu, A Cooperative Agent Modeling Approach for Process Planning, *Computers in Industry*, Vol.41, pp.83-97, 2000.

[63] W. Shih and K. Srihari, Distributed Artificial Intelligence in Manufacturing Systems Control, *Computers & Industrial Engineering*, Vol.29, No,1-4, 199-203, 1995.

[64] A. Sluga, P. Butala and G. Bervar, A Multi-Agent Approach to Process Planning and Fabrication in Distributed Manufacturing, *Computers & Industrial Engineering*, Vol.35, No.3-4, pp.455-458, 1998.

[65] P. Gu, S. Balasubramanian and D. H. Norrie, Bidding-Based Process Planning and Scheduling in a Multi-Agent System, *Computers & Industrial Engineering*, Vol.32, No.2, pp.477-496, 1997.

[66] D. Dornfeld, P. K. Wright, F.-C. Wang, P. Sheng, J. Stori, V. Sundararajan, N. Krishnan and C.-H. Chu, Multi-Agent Process Planning for a Networked Machining Service, *Technical Paper of North American Manufacturing Research Conference (NAMRC XXVII)*, MS99-175, pp.1-6, 1999.

[67] X. Liu, CFACA: Component Framework for Feature-based Design and Process Planning, *Computer-Aided Design*, Vol. 32, No.7, pp.397-408, 2000.

[68] L. Wang, W. Zhao, A. Ma'ruf and T. Hoshi, Setup-less Fabrication Technology Incorporated with Machining Feature-based CAD/CAM System for Machining Workshop, *Proceedings of the IMEC'96*, pp.95-97, 1996.

[69] D. Little, R. Rollins, M. Peck and J. K. Porter, Integrated Planning and Scheduling in the Engineer-to-Order Sector, *International Journal of Computer Integrated Manufacturing*, Vol.13, No.6, pp.545-554, 2000.

[70] L. Wang, Using Function Blocks for Distributed Process Planning and Control, *CD-ROM Proceedings of the 2001 ASME International Mechanical Engineering Congress and Exposition*, New York City, November, 2001.

[71] A. S. Rao and M. P. Georgeff, Modeling Rational Agents within a BDI-Architecture, *Proceedings of Knowledge Representation and Reasoning*, pp.473-484, 1991.

[72] L. Wang and W. Shen, Agent-Based Decision-Making for Distributed Process Planning, *CD-ROM Proceedings of the 3rd International World Manufacturing Congress*, Rochester, NY, September, 2001.

[73] IEC TC65/WG6, Function Blocks for Industrial-Process Measurement and Control Systems (Part-1: Architecture), *IEC-TC65/WG6 Committee Draft*, 1999.

[74] R. Lewis, Design of Distributed Control Systems in the Next Millennium, *Computing & Control Engineering Journal*, pp.148-152, August, 1997.

[75] L. Wang and H.-Y. Feng, An Architecture for Distributed Process Planning Using Function Blocks, *Transactions of North American Manufacturing Research Institution of SME*, Vol.30, pp.613-620, 2002.

APPENDIX: Summary of Latest CAPP Systems

Name of system	Research group	Key features	Application domain
FBD-MANS	Lee et al., 1999 USA [24]	Encapsulating the protrusions with virtual convex polytope to generate the manufacturing feature model.	Feature recognition
GCAPP	Gu et al., 1997 Singapore [25]	Using a fuzzy model and feature manufacturability to evaluate feature priorities and identify important features.	
OOSA for CAPP	Chep et al., 1999 France & Italy [26]	Representing manufacturing feature by using the object-oriented system analysis method.	
N/A	Ming et al., 1998 China [27]	Developing an information model for CAPP by using the object-oriented modeling and the Product Data Exchange Step/Standard of Exchange Product data (PDES/STEP) technologies.	
ATS	Edalew et al., 2001 UK [37]	Developing a procedure for tool selection and determining cutting parameters and cost utilizing mathematical modules and heuristic data.	Tool selection
EXCATS	Arezoo et al., 2000 UK [39]	Selecting cutting tools and conditions for machining operations by using an expert system.	
VITOOL	Maropoulos et al., 2000 UK [40][41]	Presenting the methods for feature creation, operations and tools selection, and machining time estimation.	
IPAC	Yellowley et al., 1999 Canada [42]	Integrating process planning and remote monitoring within an existing open architecture CNC.	
M-TSP	Kim & Suh, 1998 Korea [48]	Determining the optimal group and sequence of operations for a multistage by incorporating the expert system and mathematical programming.	Sequence planning
N/A	Balasubramanian & Raman, 1998 USA [45]	Modeling gradual process variables in path planning including forces, temperatures, and tool wear.	Path planning

N/A	Wu & Zhang, 1998 China [50]	Employing object-oriented technology to represent setup planning knowledge and generate alternative set-ups and fuzzy-set theory to produce optimal set-ups.	Set-up planning
N/A	Ong & Nee, 1996 Singapore [49]	Presenting a Fuzzy-set based approach that can be applied for concurrent constraint set-up planning.	
IPP in OKP	Tu et al., 2000 New Zealand & China [56]	Proposed CAPP framework includes the reference architecture for structuring a CAPP system in virtual OKP, a new CAPP method named incremental process planning (IPP), and an optimal cost analysis model.	System integration
CoCAPP	Zhao et al., 2000 China [62]	Presenting a new cooperative agent model for process planning that satisfies five major requirements: autonomy, flexibility, interoperability, modularity and scalability.	
N/A	Gu et al., 1997 Canada [65]	Describing a bidding-based approach to the integration of computer-aided design, process planning, and real-time scheduling.	
N/A	Ming et al., 1999 China [34]	Developing the architecture of a hybrid intelligent inference model using both expert system and neural networks for implementing a CAPP system.	
AAPP	Zhang et al., 1999 USA & China [16]	Proposing object-oriented manufacturing resource modeling (OOMRM) and adaptive agent-based process planning (AAPP).	
CFACA	Liu, 2000 China [67]	Developing a framework by using component technology in feature-based design and process planning.	
N/A	Srinivasan & Sheng, 1999 USA [52][53]	Dividing process planning into two phases - microplanning and macroplanning. In microplanning, process, parameters, tooling and cutting fluids are selected for the individual features. While in macroplanning, interactions between features are examined.	
CAPPES	Kryssanov et al., 1998, Japan [35]	Introducing a new formal method to design CAPP expert systems.	

CyberCut	Smith & Wright, 1996 USA [57]	Providing following services via Internet: (1) a design for manufacturing CAD interface written in Java^{TM}, (2) a choice between two CAPP systems, and (3) access to an open architecture machine tool for fabrication of mechanical parts.	
MAPP for CyberCut	Dornfeld et al., 1999 USA [66]	Describing a multi-agent process planning module in a networked machining service environment.	
IAI-CAPP	Chang & Chang, 2000 Taiwan [3]	Integrating fuzzy logic (FL) and artificial neural networks (ANN) to perform the dynamic recognition and adaptive-learning tasks of plans, adopting the ideas of important (critical) feature concept for evaluating the suitability of existing process plans for incoming product designs, and utilizing the technique of expert system.	
ANN-based CAPP	Devireddy & Ghosh, 1999 Canada [19]	Integrating the featured-based design and artificial neural net works-based planning aspects of manufacturing.	
N/A	Zhang et al., 1997 Singapore [17]	Adopting genetic algorithm to deal with process planning problems in a concurrent manner in generating the entire solution space by considering the multiple decision-making activities.	
3I-PP	Khoshnevis et al., 1999 USA [54]	Describing the architecture of a new integrated process planning system with three modules: feature completion, process selection, and process sequencing. A knowledge-based approach is applied to feature completion and process selection, and a space search algorithm to sequencing.	
N/A	Morad & Zalzala, 1999 Malaysia & UK [18]	Providing an approach to integrate process planning and scheduling using genetic algorithms. The criteria include machine capabilities, cost to operate, and processing times.	Planning scheduling integration
N/A	Little et al., 2000 UK [69]	Proposing and developing planning and scheduling reference models for engineer-to-order sector. Extended event process chains and planning-scheduling process models are created.	