**FACTA UNIVERSITATIS**
Series: **Mechanical Engineering** Vol. 9, N° 1, 2011, pp. 79 - 100

# DISTRIBUTED AGENT-BASED INTEGRATIVE MODEL FOR MASS CUSTOMIZATION PRODUCT DEVELOPMENT *

*UDC 658.5*

**Dušan N. Šormaz**

Ohio University, Department of Industrial and Systems Engineering, Ohio University,
284 Stocker Center, Athens, OH 45701, USA,
E-mail: sormaz@ohio.edu

**Abstract**. *Recent developments in information technologies and internet have enabled shortening of product delivery cycles and on-line ordering of products with high level of customization with direct access of catalogues and product databases by end customers using web based interfaces. On the other side, manufacturing in most of industries has shifted focus from mass production, through just-in-time manufacturing, and cellular manufacturing into rapid response manufacturing with the goal of shortening product manufacturing cycles. Those two approaches have been combined in the development of mass customization approaches to meet the customer requirements. In such approach each individual customer selects the product options at an entry point into design and manufacturing. This enables the manufacturers to reduce all components of the order lead time: product design, process design, product manufacturing, and delivery time, but require that both design and manufacturing planning are supported by knowledge, information, and data models using modern technologies. This paper addresses the technologies necessary to achieve distributed integrative product and process model for product configuration and mass customization. The following technologies are identified as enablers for integrative model of design and manufacturing: a) knowledge management and knowledge based reasoning, b) generation of design options and manufacturing alternatives, c) visualization of the decision space for product configuration, d) product and process visualization, e) neutral XML based data communication and exchange, and f) distributed agent-based data processing and module development. The application of these technologies in integrative modeling and planning and their implementation in IMPlanner prototype are explained.*

**Key words**: *Agent-based Systems, CAPP, Manufacturing Planning, Product Development*

---

Received October 20, 2010
* **Acknowledgements**. Results reported in this paper are the work of the author and large group of graduate students during several years of research supported by funds from Russ College of Engineering and Technology and industrial partners.

## 1. INTRODUCTION

Computer Integrated Manufacturing (CIM) is still a moving target in industrial and manufacturing engineering research and applications. In spite of the decades of research and development in design and manufacturing automation, there is still a need to increase integration of these manufacturing engineering and planning tasks. However, availability of the internet and world-wide-web has enabled companies to deliver product catalogues on line to worldwide market, thus prompting the shift in approaches to CIM. Approaches that are based on customer product configuration using web browser interface and mass customization of product and underlying processes present new challenges to modern manufacturing integration. Such integration involves the transfer of data between applications, but it should also focus on data and model integrity, distributed processing of the data, incorporation of knowledge into the planning tasks, and so on. Computer aided process planning (CAPP) is rightly seen as an integration fabric for CIM with its relations to design (CAD), manufacturing (CAM), facility planning and scheduling tasks. Development of integrative process plan model which serves as an integrator during product development cycle enables decision making process on various levels of details with the final goal of delivering the final customer configured product on time with the desired quality and negotiated cost. The need for rapid mass customization requires that such model is transparent between product development tasks, that it can be easily saved, transferred, and re-created as needs arise.

The paper is organized as follows. Section 2 provides a brief overview of previous work in the areas of process plan representation and product configuration and mass customization. Section 3 presents integrative process model and its components. Section 4 explains the methods and procedures in the model that may be used for rapid product configuration and customization. Section 5 gives a brief overview of the integrative process plan prototype IMPlanner, while section 6 concludes the paper.

## 2. PREVIOUS WORK

Data and knowledge representation in process planning have received significant interest in research. An early work on ALPS [1] proposed a graphical representation for manufacturing processes and means for specifying serial, parallel and concurrent tasks. Since then, several papers addressed knowledge representation, for example, using frames and rules [2] or an object-oriented data model [3]. Recent results are in generation of the Process Specification Language (PSL) [4] as a neutral format for the specification of process representation and exchange of different ontologies or semantics between various domains. Development of process planning specific, NC data within STEP standard has been described in [5]. Work on XML [6], as a very flexible language that transfers both data and their description (metadata) prompted its widespread use in many research efforts.

The work on utilization of internet and world-wide-web as a support tool for product configuration and customization has been approached from several angles. The recent research [7] is focused on identifying a marketing strategy to capture customer requirements regarding mass customization. However, there is the recognized need for integrating marketing, design and manufacturing efforts into a coherent model and system in order to generate the maximal benefit from the distributed processing of customer requirements and their integration into design/ manufacturing cycle.

Agent-based approach to software development [8] has attracted significant attention in research community. The agent-based framework has been used in several research papers devoted to intelligent manufacturing and process planning. Paper [9] describes a dynamic process planning system in which agent approach provides for dynamical changes and updates in the process plans, while paper [10] describes an agent-based system for intelligent manufacturing, in which communication between various automated activities is carried through software agents.

## 3. Integrative Process Plan Model and Representation

Integrative process plan model developed in this work is based on process planning object model proposed in [3], which describes a data model for representation of process plans based on the different activities involved in manufacturing. A process planning representation model facilitates the development of algorithms for manufacturing problems like sequencing, scheduling etc. by reducing the over-all algorithm development time. The model accommodates a variety of data that may be needed in manufacturing planning algorithms. Components of the model are manufacturing process model, manufacturing planning object model, feature object model, and process object model and they are described in this section.

### 3.1 Manufacturing Process Model Dimensions

In this section we describe manufacturing process model, which is graphical model for representing planning functions and tasks for manufacturing processes. The model includes three dimensions: variety, time, and aggregation, which are concerned with the manufacturing planning tasks.

The basic entity of the manufacturing process model is a process, intuitively understood as an activity, usually planned in advance, with all necessary attributes. All manufacturing planning functions generate various planned tasks or activities (e.g., cutting with turning cutter, deforming with a press, machining on a single machine, processing job order, etc.). Each of these tasks (manufacturing execution tasks) has numerous attributes that have to be defined before the task can be undertaken (e.g., for cutting with turning cutter, one has to define part, tool, cutting parameters, space orientation of part and tool on a lathe, starting time, ending time). These attributes are usually defined by different manufacturing planning functions. In this section we identify dimensions of a model that are independent of planning function, require transfer or translation from one function to another and facilitate manufacturing integration.

Manufacturing process model, as mentioned earlier, consists of three dimensions: time, variety and aggregation. Each process is related to other processes with respect to these three dimensions, in a conceptual way shown in **Fig. 1**a.

<u>*Time dimension*</u> describes relative relation between several processes of the same type, along the timeline. This relation has several levels of certainty. The lowest level is when we specify that some (planned) process has to be performed before or after another process (this relation is known as precedence constraint). The relation may be refined

until final determination that these two processes have specified starting and ending times as defined by scheduling. The important property of this relation is its transitivity.

Variety dimension describes the sets of different processes that are generated within a certain level. Usually it is necessary to define a set of different processes in order to complete manufacturing task. For example, when defining process plan, there is a set of features that require a set of cutting processes. Another way for generation of different processes is alternative generation. For example, in selection of cutting processes, there are usually several alternative cutting methods to machine the same feature.

Aggregation dimension relates to various scopes of planned processes. Processes are defined with different levels of details and/or time frame (duration). This dimension explains that some process is a part of another process with the same part, machine, tool or some other attribute. This is necessary in order to distribute planning tasks among different functions. For example, in process selection task of process planning function, someone is concerned only with physical cutting processes available for a single feature in order to satisfy quality of the part, while during scheduling the main concern is to arrange various orders on a single machine in order to satisfy delivery dates for products.

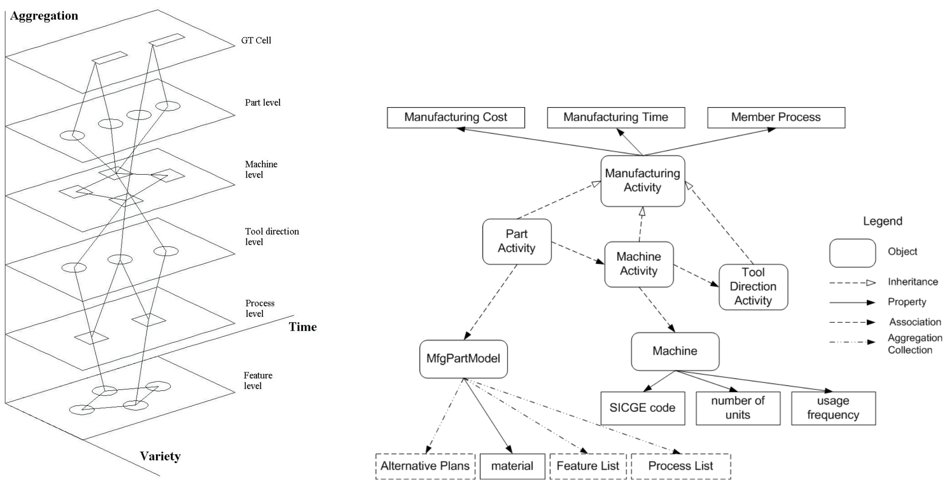



**Fig. 1.** Manufacturing process model: a) model dimensions, b) data representation

Therefore, aggregation is performed when we combine processes of the lower level that have some attribute in common in order to perform planning on a higher level.

Processes in the manufacturing process model are connected by relations between them. Different types of relations correspond to different dimensions. For example precedence is relation in time dimension, inclusion, exclusion, or alternative are relations in variety dimension, while aggregation dimension between processes is defined by aggregate (a-part-of) relations between them.

Three explained dimensions are not independent. Relations that hold between processes in one dimension impact other dimensions as well. For example, precedence order on feature level has to be replicated on all other levels when process planning is performed on machine level. Scheduling and machine load constraints may have an impact on process plans in term of selecting an alternative plan which may balance load. Generation of alternative cutting processes and selection of alternative machines or tools for individual features requires generation of alternative operations on machine level and provides choice of the most suitable alternative at scheduling level.

Manufacturing planning activity creates such a model in a distributed fashion (i.e., several specialists with different knowledge are involved, and the model is subject to change as manufacturing planning and /or execution progresses).

### 3.2 Manufacturing planning object model

Manufacturing planning object model is shown in **Fig. 1**b. The model is based on analysis of product and process design entities and includes hierarchical representation of manufacturing activities, that has manufacturing processes as its leaves, collection of manufacturing features and corresponding manufacturing processes, and a collection of machines used in the manufacturing system.

A manufacturing activity represents the core of the model. Any activity that contributes to the manufacturing of a part is called as a manufacturing activity. A manufacturing activity has attributes like manufacturing cost, manufacturing time, member process etc. A manufacturing activity can be classified as a part activity, a machine activity or a tool direction activity. A part activity describes the process plan for a part. There is an association relationship between part activity and part. Each part activity is associated with a part. A process plan for a part is a collection of the machines through which a part has to pass through to be completely manufactured. The part in turn can have multiple alternative process plans. The machining process of a part on each machine in its process plan is represented by a machine activity. Each machine activity is associated with a machine object. There is an aggregation relationship between part activity and machine activity. Each part activity is a collection of machine activities. The part object has attributes like a collection of its alternative process plans, part material, features list, process list etc. Each machine activity is a collection of tool direction activities. A tool direction activity holds directional information about a machining process. The member process attribute of manufacturing activity is used to store the aggregations of a manufacturing activity. Thus, the member process attribute of a part activity holds a collection of machine activity. The member process attribute of a machine activity holds a collection of tool direction activities.

### 3.3 Feature object model

Feature object model represents a hierarchical representation of various machining features with inheritance relations within it. The model is shown in **Fig. 2**. The major class is *MfgFeature* that abstracts all common properties for all features. Properties at this level include feature name, containing part, tolerance data, list of alternative processes, and precedence relations.

*MfgFeature* class is extended into several subclasses that correspond to machining feature types found in mechanical prismatic parts (such as *Hole*, *Slot*, and *Pocket*). These classes model properties of particular feature type and include different dimension parameters, and process capability data. However, model properties on general, feature, level are of generic nature and can be applied by extending this model to other domains (like rotational parts, sheet metal parts, and so on). Model also contains several auxiliary classes that serve the purpose of generating various feature relations and displaying them to user.

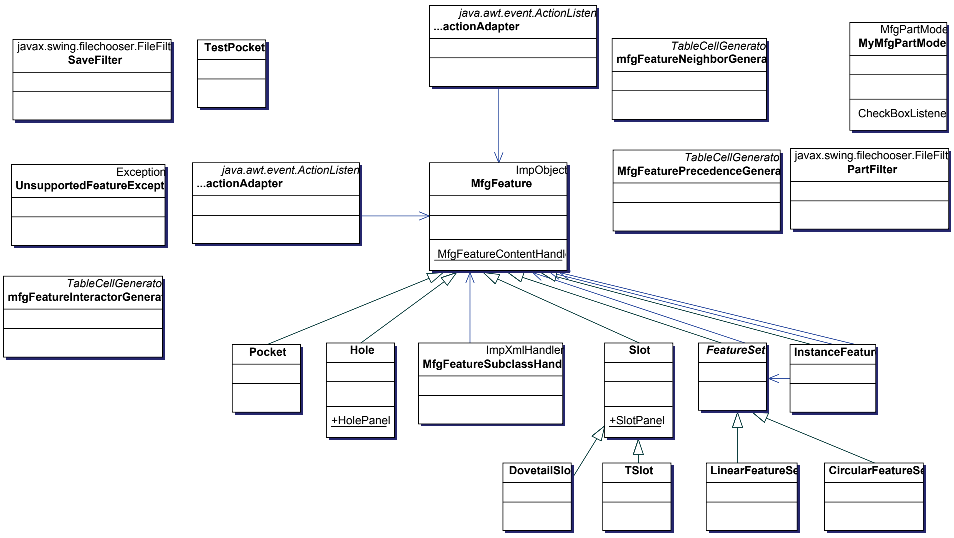



**Fig. 2.** Class diagram for features

### 3.4 Machining process object model

The knowledge about processes is also represented in an objected-oriented model. In order to map this knowledge representation, the machining processes are categorized based on their characteristics. **Fig. 3** presents this hierarchy in a UML based model.

The class *MfgProcess* represents the most generic process class, i.e., a process with the common data shared by the rest of the processes. Further distinctions are carried out based on the process characteristics such as hole making processes and profile generating milling processes. The hole making processes are further divided into *CoreMaking*, *HoleStarting* and *HoleImproving* processes, while the *Milling* process has as sub-type *EndMilling* process. These generalized classes are implemented as abstract classes and are shown with italicized titles in **Fig. 3**. The classes under these umbrella classes are for representing the actual machining processes (for example, *TwistDrilling*, *EndMIllingSlotting*, or *FaceMilling*). Therefore, based on the inheritance, *EndMillingSlotting* process acquires process information from *EndMilling*, which further leads to the parent class *MfgProcess*. The following paragraphs give a brief description of the *MfgProcess* and *EndMilling* classes, while description of other classes is part of working documentation.

*MfgProcess*: This class contains member variables such as feature, stock, workpiece, cutting parameters, constraints, tool, and tool path. These listed variables are used in every inherited process class as every process contains these components of machining process. Also, this class carries the GUI components for showing process information and a graphical interface to display a process.

*EndMilling*: The implementation for visualization is mainly concentrated on end milling operations, so this class contains some graphical components. This class provides methods to prepare the scene graph, which carries nodes to display machining process components and the animation for visualization. The GUI required for event-based interaction with the virtual world displayed is provided in this class. The subclasses --slotting and peripheral end milling-- use this implementation with distinctive modifications in tool approach for machining.

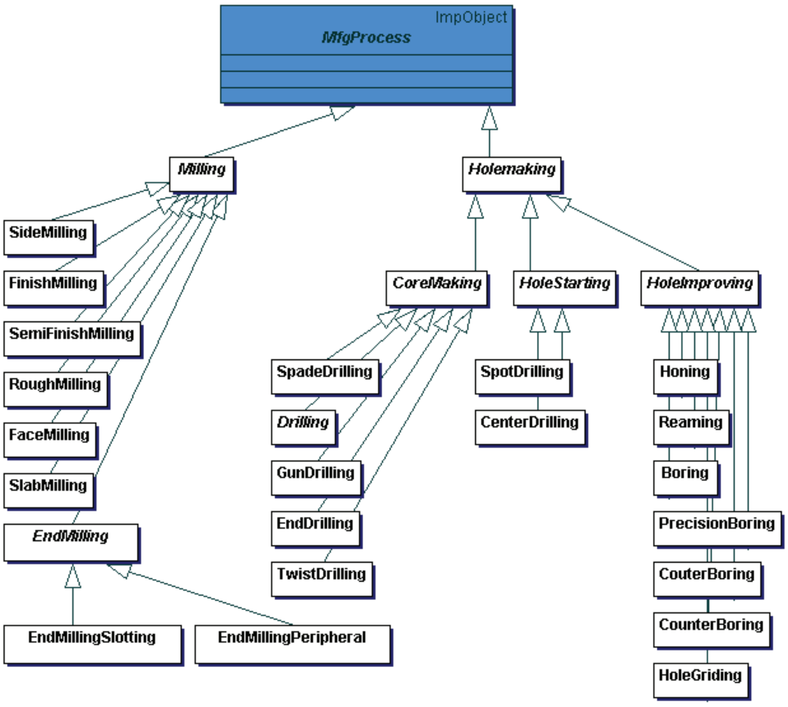


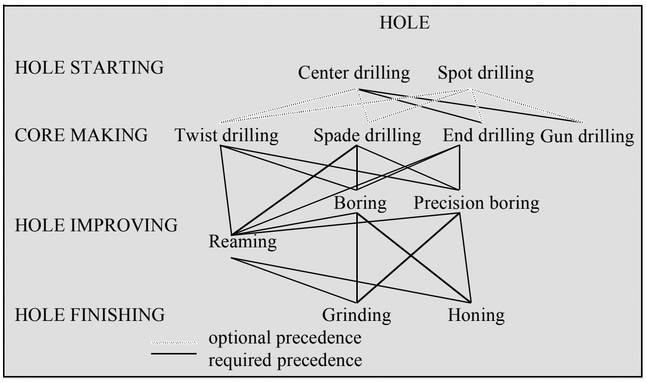

**Fig. 3.** Machining process hierarchy

## 4. Product Configuration and Customization

The integrative process plan model described in the previous section enables rapid product configuration and customization. Product configuration is based on knowledge management that is embedded into the model. This knowledge enables generation of design options and planning alternatives in rapid, on-line mode by utilizing rule-based reasoning which converts the customer requirements into the one-of-a-kind customer driven product specification. Such specification, further, can be transferred between various manufacturing planning tasks in a neutral XML format to various product/manufacturing decision points throughout the enterprise. This section describes the example of those technologies within integrative process plan model.

### 4.1 Knowledge Management and Knowledge Based Reasoning

Process plan knowledge represents the wealth of the manufacturing enterprise experience in particular domain. It may contain the knowledge from the literature, from machine and tool vendors, or from the enterprise own historical data. The importance in capturing that knowledge cannot be overemphasized, because that experience is the manufacturer only competitive advantage in the tight product market. It is imperative that this knowledge be captured into permanent repository and utilized throughout product development cycle. Illustration of this approach is given in Fig. 4. Hole making knowledge (see [11] for details) includes tolerance capabilities of various operations and precedence requirements between them that can be represented in the form of a precedence graph as shown in Fig. **4**a. Capturing that knowledge (see [12] for details) into XML format, as shown in Fig. 4b enables consistent application in all hole manufacturing tasks, and transfer between various applications, and computing platforms.




a)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ProcessRelation>
- <edu.ohiou.implanner.processes.TwistDrilling>
<precedes>edu.ohiou.implanner.processes.Boring</precedes>
<precedes>edu.ohiou.implanner.processes.PrecisionBoring</precedes>
<precedes>edu.ohiou.implanner.processes.Reaming</precedes>
<Parameter smallestToolDiameter="0.0625" />
<Parameter largestToolDiameter="2.0" />
<Parameter negativeTolerance="(* 0.007 (sqrt ?dia ))" />
<Parameter positiveTolerance="(+ (*0.007 (sqrt ?dia)) 0.003)" />
<Parameter straightness="(+ (*0.005 (** (/ ?depth ?dia) 3)) 0.002)" />
<Parameter roundness="0.004" />
<Parameter parallelism="(+ (* (** (/ ?depth ?dia) 3) 0.001) 0.003)" />
<Parameter perpendicularity="(+ (* (** (/ ?depth ?dia) 3) 0.001) 0.003)" />
<Parameter Depth="(* 12 ?dia)" />
<Parameter truePosition="0.008" />
<Parameter surfaceFinish="100.0" />
</edu.ohiou.implanner.processes.TwistDrilling>
+ <edu.ohiou.implanner.processes.EndDrilling>
+ <edu.ohiou.implanner.processes.Boring>
```

b)

**Fig. 4.** Process knowledge: a) precedence relation graph, b) knowledge capture in XML format

### 4.2 Generation of Design Options and Planning Alternatives

On-line product configuration and customization require rapid evaluation of the impact of the customer requirements on product design and manufacturing with evaluation of design feasibility and quality, manufacturing cost and due dates and overall enterprise performance in the case of mass customization of various products. Such evaluation can be performed only if process plan model exists for each product family with AND-OR graph representation of product options and their compatibility. Alternative manufacturing plans also need to be captured for rapid process re-planning in the cases of fluctuation of market demand for different products. This is achieved by generating a hierarchical integrative process plan model for each product with the dimension as shown in Fig.1a. A simple example that illustrates the generation of that model is shown here. The part design, its features and feature precedence network (FPN) are shown in **Fig. 5**. The FPN provides constraints for sequencing of feature operations and thus serves as a generator for a network of alternative processes shown, as process plan network (PPN) in **Fig. 6** for the same example.

The network shown in **Fig. 6** is a portion of the complete PPN in which any path from the S mode to the E mode represents an alternative part activity to make the part. The algorithm for generation of this network is beyond the scope of this paper and it is described in [13].

### 4.3 Distributed Application of the Decision Space, Product, and Process Visualization

The integrative process plan model is enabler for distributed application of several interrelated design and manufacturing planning tasks. Many of these tasks require different views from the model necessary to focus decision making in each particular task as shown in **Fig. 7**. The realistic scenario that starts with on-line product configuration is shown. The customer accesses only product definition data in order to perform product configuration and customization. The selection of product options may trigger the need for decision making and/or optimization in other tasks, such as process selection, setup or capacity planning within enterprise. The customer action in product configuration triggers intelligent agents that convey modified product specifications to product designers and manufacturing planner which make their decisions to accommodate the customer requirement, but at the same time to optimize the overall system performance based on requirements from different customers.

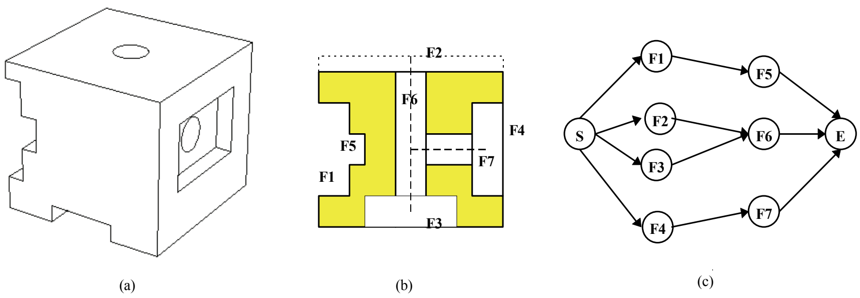



**Fig. 5.** Sample part: a) design, b) features, c) feature precedence network

As result they send agent back to customer with a quote of updated information about product due dates and cost. In such way the customer is able to accept the quote or modify its requests in real time. Other functions and task depicted in **Fig. 7** may also be impacted by the customer decisions, so appropriate intelligent agents are sent to perform negotiations between tasks and to provide for overall optimal performance.

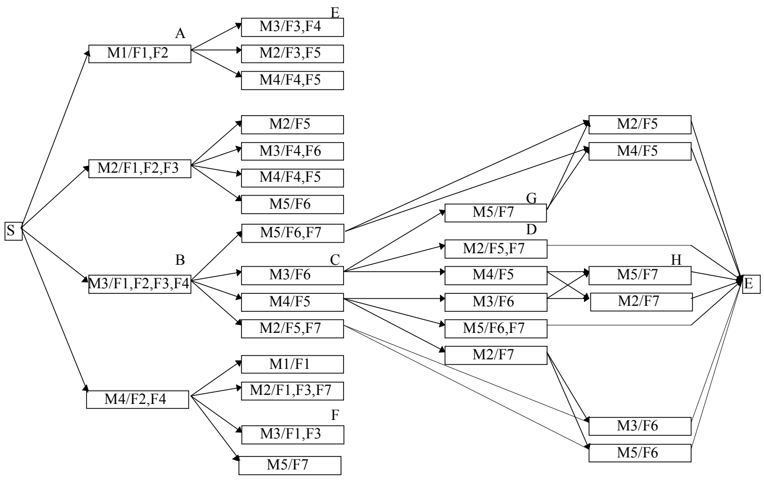



**Fig. 6.** Segment of process plan network alternatives for the example

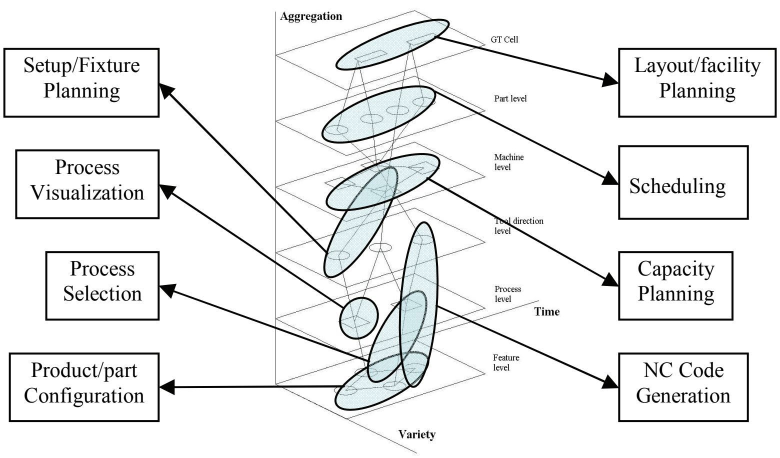



**Fig. 7.** Different views from the integrative process plan model

### 4.4 Agent Based Modeling

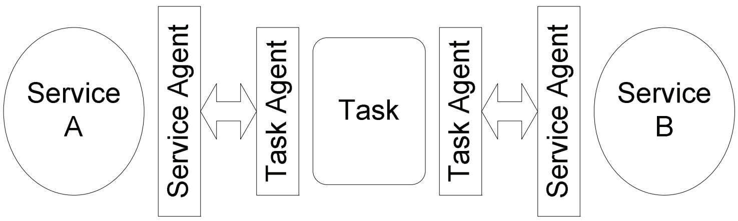
An agent is a software or a subsystem possessing some degree of autonomy and intelligence. Some features of agent software include: a) Makes autonomous decisions, 100% situation handling, b) Degree of intelligence, c) Knowledge base, and d) Interaction with environment, other agents, humans.

A multi-agent system, is a perfect “society” of such agents, where they co-operate with each other to accomplish complex tasks. This multi-agent system can solve industrial problems which are impossible to be solved by a monolithic system. The decentralized architecture does not require a meta-entity or platform to supervise activities and keep a “population count”.

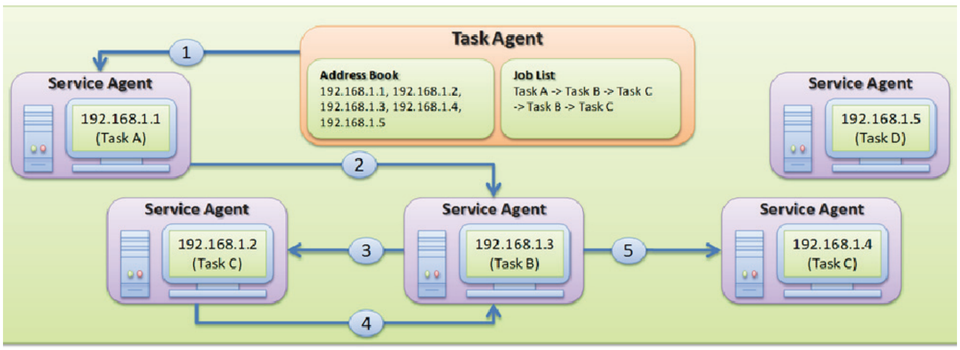
Agents transfer data through a series of services distributed among different computers by searching for Internet Protocol (IP) address.

The basic approach to agent-based programming is shown in **Fig. 8**. Protocol in **Fig. 8**a illustrates a task which requires using two services to be completed. Since services may be offered to many different tasks, a unified interface (agents) between tasks and services is introduced.

Task agent and service agent exchange messages, in which task agent sends message related to data needed for task, service agent sends the data to service. Upon completion of service, the service agent sends results back to task agent, which in turn interprets the results to the task. An example of distributed processing by agents is shown in **Fig. 8**b. Properties of two types of software agents are shown in **Table 1**.




a)




b)

**Fig. 8.** Agents: a) Agent-based communication protocols, b) Distributed processing

**Table 1.** Properties of agent types

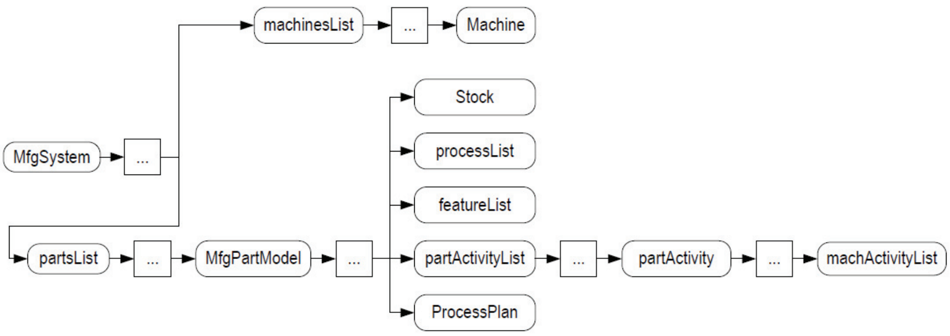
| | Task Agent | Service Agent |
|---|---|---|
| **Actions** | Searching for the server which accepts current job in the job sequence | Processing the jobs waiting in the queue |
| **Core Engine** | Rule base system | Queue policy |
| **Contained Info** | Content<br>Job sequence<br>Address book | Service<br>Task queue<br>Service history |
| **Life Cycle** | Birth - Initiates job sequence<br>Die - Complete all jobs | Birth - Register service<br>Die - Remove service |
| **States** | Uninitiated/Searching/ In-transit/ Waiting /In-Service/Complete | Idle/Busy |

### 4.5 Neutral XML Knowledge and Data Representation

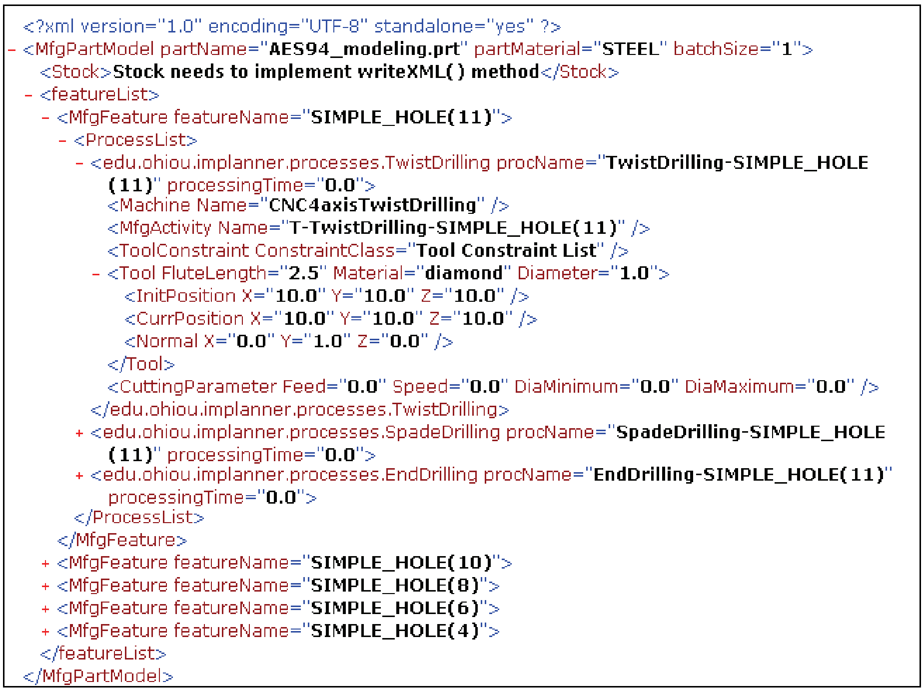
The persistent and neutral data storage mechanism for the integrative process plan model is based on XML as portable, platform independent language that carries both data and metadata (definitions of the data) in the same file or database. XML data representation is based on the developed XML schema. A XML schema is a set of rules that have to be satisfied in valid XML document [6]. **Fig. 9**a shows the XML Schema definition for the integrative process plan model.

The root element of the schema is <*MfgSystem*> that holds all information about a factory model (which is the extension of process plan model). This element holds two collection elements <*partsList*> and <*machinesList*>. The <*partsList*> element is a collection of part models represented by the <*MfgPartModel*> element. The <*machinesList*> is a collection of the available machines represented by <*Machine*> element. Each <*MfgPartModel*> element has a list element called <*partActivityList*>, which is a collection of alternative process plans, for the enclosing part element. These alternative process plans are represented as <*PartActivity*> elements on the part model. Each <*PartActivity*> element has a <*machineActivityList*> element. The <*machineActivityList*> element is a collection of all the <*MachineActivity*> elements in the enclosing process plan. Each <*MachineActivity*> element encloses a <*Machine*> element that holds information about the machine on which the machining operation is done. Each <*MachineActivity*> element holds a <*toolDirectionActivityList*> element, This element holds the spatial information about the tools used in this machining activity. <*toolDirectionAcitivity*> in turn, holds <*mfgProcessName*> element as reference to a machining process. The hierarchical relationship between parts, machines, part activities, machine activities, tool activities, features and processes is in such way incorporated into the XML document.

An example document that represents the result of process selection for a sample part is shown in **Fig. 9**b. Adherence of the XML document to the XML schema is guaranteed by XML parsing tools. Details on methods for writing XML documents and passing XML input are given in [14].




a)



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <MfgPartModel partName="AES94_modeling.prt" partMaterial="STEEL" batchSize="1">
    <Stock>Stock needs to implement writeXML( ) method</Stock>
  - <featureList>
    - <MfgFeature featureName="SIMPLE_HOLE(11)">
      - <ProcessList>
        - <edu.ohiou.implanner.processes.TwistDrilling procName="TwistDrilling-SIMPLE_HOLE
            (11)" processingTime="0.0">
            <Machine Name="CNC4axisTwistDrilling" />
            <MfgActivity Name="T-TwistDrilling-SIMPLE_HOLE(11)" />
            <ToolConstraint ConstraintClass="Tool Constraint List" />
          - <Tool FluteLength="2.5" Material="diamond" Diameter="1.0">
              <InitPosition X="10.0" Y="10.0" Z="10.0" />
              <CurrPosition X="10.0" Y="10.0" Z="10.0" />
              <Normal X="0.0" Y="1.0" Z="0.0" />
            </Tool>
            <CuttingParameter Feed="0.0" Speed="0.0" DiaMinimum="0.0" DiaMaximum="0.0" />
          </edu.ohiou.implanner.processes.TwistDrilling>
        + <edu.ohiou.implanner.processes.SpadeDrilling procName="SpadeDrilling-SIMPLE_HOLE
            (11)" processingTime="0.0">
        + <edu.ohiou.implanner.processes.EndDrilling procName="EndDrilling-SIMPLE_HOLE(11)"
            processingTime="0.0">
        </ProcessList>
      </MfgFeature>
    + <MfgFeature featureName="SIMPLE_HOLE(10)">
    + <MfgFeature featureName="SIMPLE_HOLE(8)">
    + <MfgFeature featureName="SIMPLE_HOLE(6)">
    + <MfgFeature featureName="SIMPLE_HOLE(4)">
    </featureList>
  </MfgPartModel>
```

b)

**Fig. 9.** XML data model: a) XML schema, b) a sample data file for process plan model.

## 5. DISTRIBUTED PLANNING APPLICATIONS

The integrative process plan model has been implemented within the IMPlanner system (for details of the system see [15] and [16]) shown in **Fig. 10**. The properties of this system are: 1) it relies on existing software tools for CAD/CAM, CAPP, optimization, and simulation, 2) it provides for distributed processing of process plans across enterprise and virtual enterprise in intranet/internet environment, 3) it utilizes available technology and emerging standards for internet computing (namely Java and its tools JNI, Sockets, and RMI), and 4) it utilizes standards for data storage and exchange (relational databases and XML). Those properties enable evolutionary development of the system, and incremental transfer of the technology into enterprises.

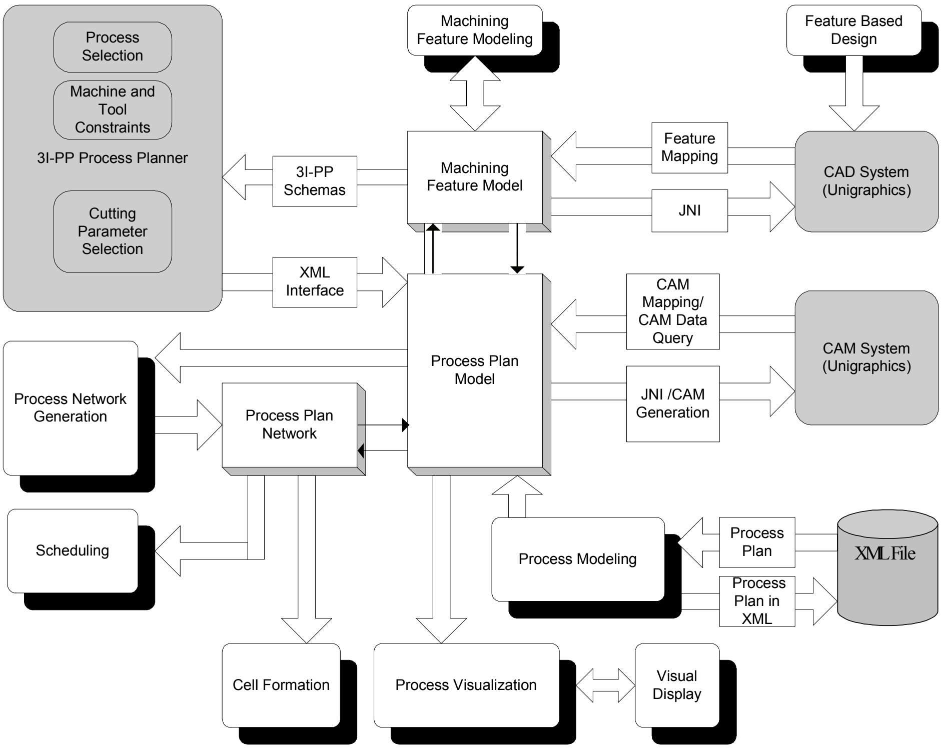



**Fig. 10.** The architecture of IMPlanner prototype

Dark shaded modules in **Fig. 10** represent existing software tools. Three modules shown by sharp rectangle (with light grey shadow) in the middle represent process plan representation model. Rounded rectangles with black shadow represent applications that are being developed in this project. Sharp rectangles on arrows represent interfaces utilized between different modules. It is important to note here that, due to software development tools selected for this system (namely, Java, HTML, JNI and XML), all of these modules (existing and those under development) may exist on different computers with various operating systems, and on geographically distant locations.

### 5.1 Manufacturing Feature Modeling

Manufacturing feature modeling is the task between feature recognition and process planning. For automated process planning it is necessary to represent design in the form of manufacturing feature model. While adopting current research results in feature recognition, we feel that the line between FR and CAPP is blurred and here we address those issues: feature interactions, and hybrid dynamic (or simultaneous) model that includes geometry and features.

The input to this module is a feature-based representation of part (feature-based CAD model provided by user). Example of the model retrieved from CAD file is shown in **Fig. 11**. This input is then processed to develop the machining feature model of the part that holds the set of features to be analyzed. This involves the querying of the part model for geometric information about features using interface to CAD system. A complete machining feature model is developed for the given part and is utilized in deducing precedence relationships between features. The commercial CAD system handles all system needs pertaining to geometry generation, geometric computations and queries on solid model. The prototype system is interfaced to the CAD system (Unigraphics). The feature interaction analysis system processes the machining feature model to produce feature relationship information. This information is presented in terms of precedence constraints as the feature precedence network. In this procedure, features that are related to each other are identified and ordered in terms of which feature needs to be machined prior to the other feature.

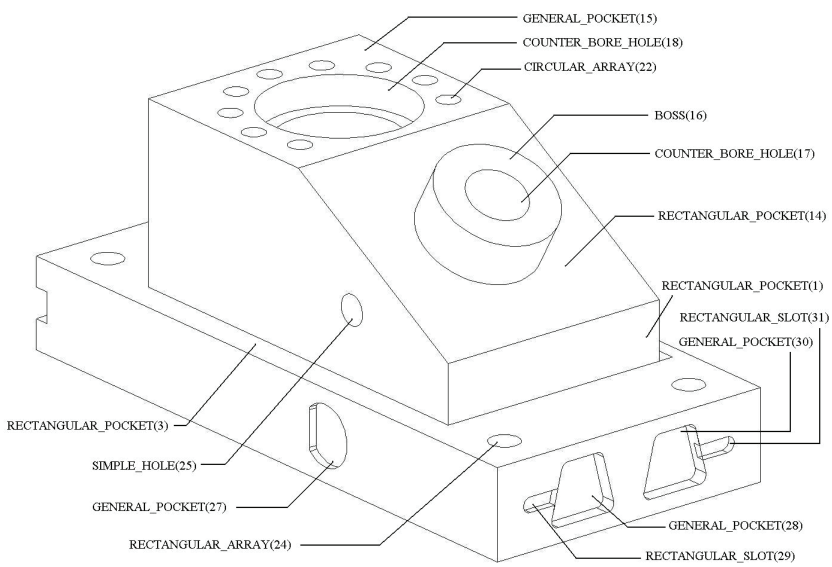



**Fig. 11.** Geometry and features of Anc101

Interacting features are identified based on feature adjacency, which in turn is inferred from face adjacency information based on queries performed on CAD model of the part. Once the set of interacting features are determined and composed into feature interaction graph, features are analyzed pair-wise in a feature interaction analysis engine in order to determine feature precedence. This procedure is applied continuously until the complete set of interacting features is exhausted. Finally, the generated precedence information is organized into a feature precedence network for the part model. This procedure has been applied on a series of mechanical parts from NSF design repository. An example of this parts is Anc101 part which geometry and form features implemented in Unigraphics as shown in **Fig. 11**. **Fig. 12** shows feature precedence network which was generated automatically by applying feature interaction analysis algorithm.

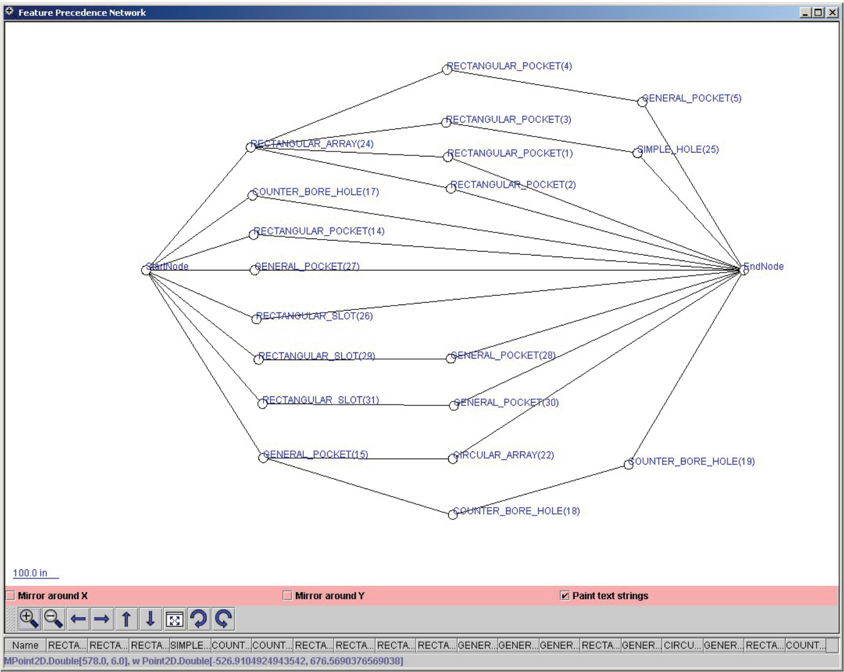



**Fig. 12.** Feature precedence network for Anc101 example

### 5.2 Rule Based Process Selection

Rule based system in process planning context is the system which generates alternative processes/ machines for each part feature by utilizing available facts (knowledge) about the processes, machines and decision logic (rules to make decision based on part design requirements). Feature in this context is defined as geometrical shape which is manufactured using single tool.

Knowledge of the system is detailed knowledge of i) manufacturing processes – manufacturing capability of processes like: drilling, reaming, milling, grinding etc., ii) machine and tool facts – available machines and tools with their specifications, iii) feature knowledge, which includes feature facts like: types of feature, feature dimensions, quality, tolerance and surface finish requirements. Steps in execution of rule based process selection system include process selection, machine selection, toll selection, and processing time calculation.

When rule based system is run, working memory of the system is populated with process facts, machine facts, tool facts, inserted feature facts and available rules. Based on these facts, several rules are activated and fired; they in turn create or modify the existing facts in the working memory. These creation and modification of facts trigger other rules to execute and chain reaction of executing rules continuous until there are no rules to fire, which signifies the completion of process planning procedure.

By running the rule based system, depending upon available knowledge, the topology and the tolerance requirement of each feature, alternative processes (machines) capable of machining that feature are generated with machining time and tool requirement information. Rule based process selection logic has been described in [17]. Algorithm for the process selection enables the selection of multiple processes for a single feature if its tolerances so require (for example for a high quality hole, processes like drilling, reaming, and boring may be selected). After the selection of the machine and tool constraints it estimates the machining time and cost of manufacturing the feature. Result of this application on running Anc101 example is shown in **Fig. 13**.

After performing those two described tasks manufacturing process model is created in the memory with all necessary relationships. This model can be displayed in the form of a 3d graph as shown in **Fig. 14**a, in which cylinder represents part, spheres represent features and cones represent manufacturing processes. Different graph showing activity aggregation is shown in **Fig. 14**b in which larger sphere represents PartActivity, boxes represent MachineActivities, and small spheres represent ToolDirActivities.

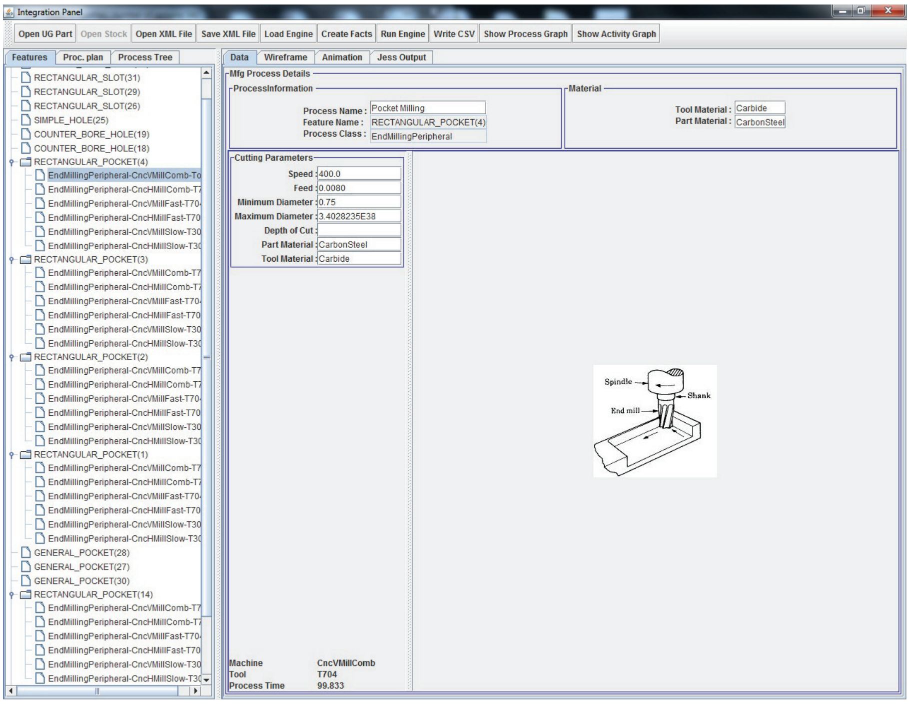



**Fig. 13.** Result of rule-based process selection for Anc101

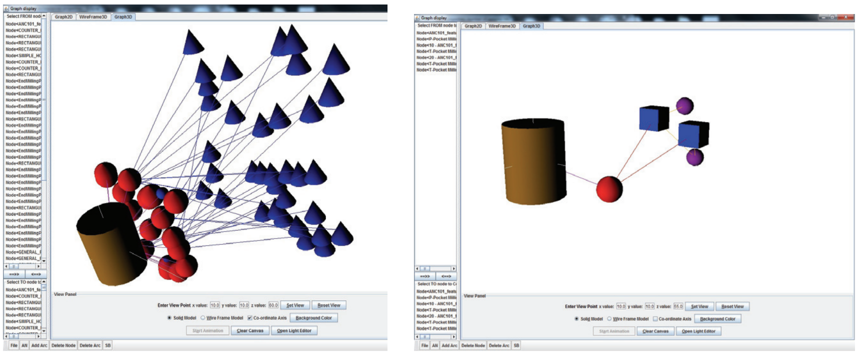


**Fig. 14.** Manufacturing Process Model for Anc101 Example:
a) Feature-process tree, b) Plan activity tree

Potential for higher information content exists in such that each node may be shown as a 3d model of corresponding node, for example for part its solid model can be shown, and for machine its 3d model also can be shown. Another possibility is to provide textual display of data for currently selected node. The model can be also executed in agent framework by selecting a node and sending the node and its related nodes to another agent for further processing. Information overload can be prevented by selecting what set of relations should be shown in the graph and by utilizing agent based approach described in section 4.4 for distributed processing with only required data. However, this developed process model also provides means for changing the focus of planning task from global to local and vice-versa and inspecting impact of decision on the overall model.

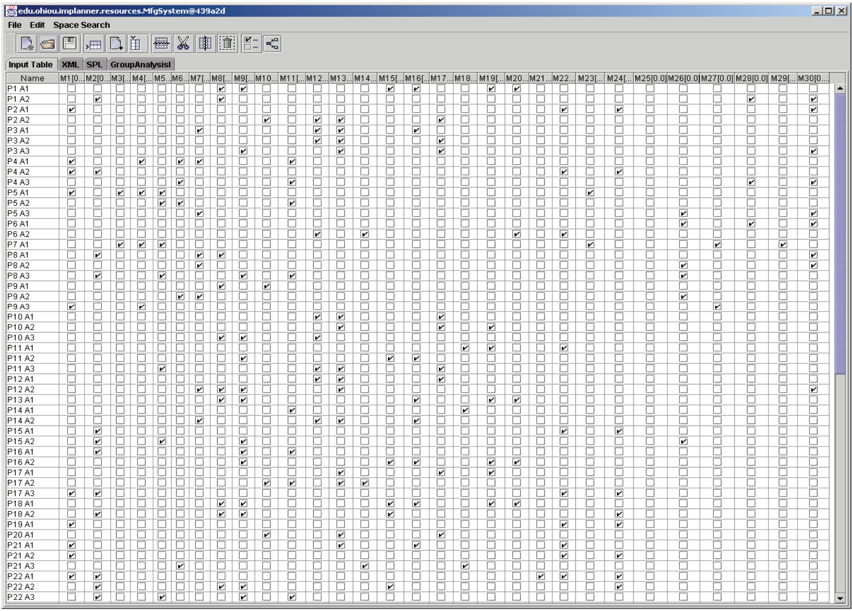
### 5.3 Manufacturing Cell Formation

This is the application for generation of manufacturing cells using space search algorithm. The group of parts with alternative process plans is an input into the procedure. Alternative process plans may be generated by other components or application in the system. In this application they are shown as part-machine matrix. Cell formation algorithm applies space search based on key machine and forms part families in a manner similar to PFA procedure [18]. As a result, machine cells and part families are created according to user selectable optimization criteria (such as intercellular transport, number of exceptions, load balance, etc.). The application of the procedure is shown in **Fig. 15**. Part-machine matrix for example with 40 parts (total of 89 alternative process plans) and 30 machines in shown in **Fig. 15**a, and resulting manufacturing cells are shown in **Fig. 15**b. Machine in the same cell are shown with the same color shade.  The system also shows which alternative routing for each part was selected and any exceptional elements in the final part-machine matrix.

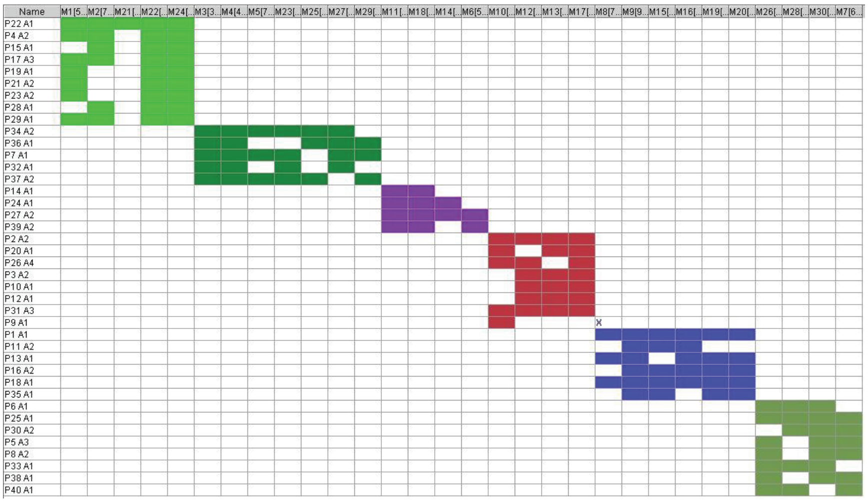
### 5.4 FMS Scheduling and Simulation

Scenario for this integrated approach is the following. For a group of six designed parts it is necessary to perform process planning and select the process plan from several alternatives in order to design FMS control module, which will be studied by simulation.

The modules integrated into this scenario are shown in **Fig. 16**. Feature mapping module and rule based process planner, are executed to create necessary date. FMS Control simulation model is briefly described in this section.




a)



b)

**Fig. 15** Cell formation: a) Part-machine matrix (partial), b) Generated cells

FMS simulation model represents FMS control procedures. It is developed as a discrete event model in Arena [18]. It is capable of applying several control policies for selection of alternative routings, and several dispatching rules to order processing of parts on machines. Control policies include best plan, random plan, and dynamic model with feature focus which selects alternative for each manufacturing feature. The last model, which we call feature focused dynamic model requires integrated approach in which all three modules (feature mapping, process planning and FMS controller) participate.

In a feature focused dynamic simulation model, appropriate machining processes are generated on demand by running rule based system. This rule based system not only selects capable processes for the feature but also selects alternative machines available to make that feature with tool and processing time information. Alternatives for every part are sent to the FMS controller which is responsible for collecting the data about the FMS status and then selects the best process plan for this status to be used in the next manufacturing period. This module will provide required information to simulation model when prompted.

In this novel approach, at every predefined time interval, the simulation model will collect and send current machine utilization or queue size for every machine to the FMS controller. Based on the information fed into the FMS controller module, this module will select the best machine in terms of lowest machine utilization or queue size out of alternatives to make each feature. After selecting the best machines, process plan in simulation model is updated with the best machines for part routing data. New entities arriving at this point of time uses new updated process plan. This selection of process plan continues at same predefined time interval.

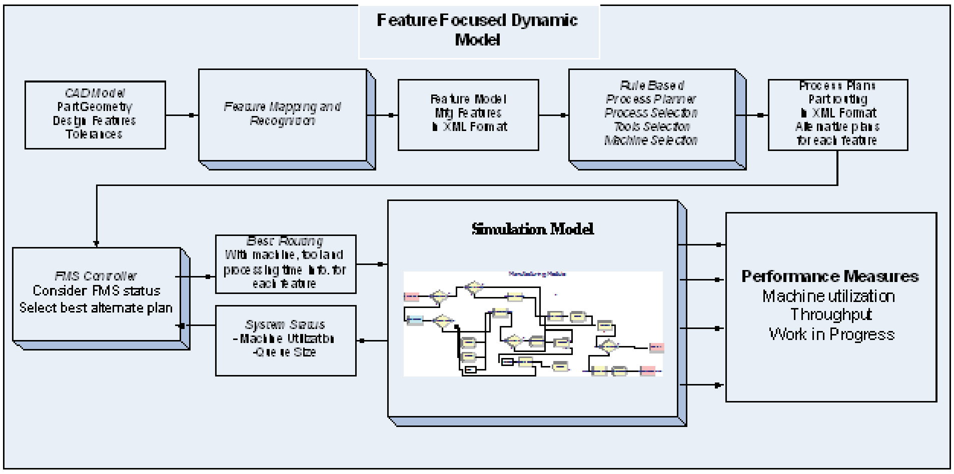



**Fig. 16.** Integration of Feature Modeling, Process Planning and FMS Control Modules

Each entity runs through simulated FMS following given process plan. Five dispatching rules are applied for prioritizing between parts for processing at each machine. These dispatching rules are i) FIFO, ii) SPT, iii) SIPT, iv) LIPT, and v) Dissimilarity Maximization/ Minimization. After the completion of all the operation for each part, the part leaves the system and their performance measures are recorded. The performance measures considered are machine utilization, average flow time, and work in progress.

Simulation model has been run for three scenarios: best process plan, random process plan and dynamic process plan. Preliminary results of running model for best and random process plan show a potential improvement of 20% reduction in production cycle.

## 6. CONCLUSIONS

This paper describes the generic framework for utilization of integrative modeling and planning in web-based mass customization solutions. Integration framework for mass customization consists of the procedures for capturing the customer requirements and mappings to convert these requirements into appropriate components of the integrative design and manufacturing model to generate the necessary product specification, cost/price estimation, manufacturing plans, and delivery dates. The expected benefits from integrative modeling and planning are reductions in order delivery cycles, product development cost, and manufacturing cost. Improvements in product quality and customer satisfaction that result from integrative modeling and planning are also expected.

The developed framework is under implementation and testing within IMPlanner prototype. The process plan model has been completed. Algorithms for rule-based process selection, process visualization and process network generation have been implemented. The current work is in developing procedures for browser based interface to capture product requirements and distributed agent-based processing of those requirements within the model.

## REFERENCES


1. B. A. Catron, and S. R. Ray, ALPS: A Language for Process Specification. *International Journal of Computer Integrated Manufacturing*, 4, 105-113. 1991.
2. J. Y. Park, B. Khoshnevis. A real time computer aided process planning system as a support tool for economic product design. *Journal of Manufacturing Systems*, 12(6):181-193, 1993.
3. D. N. Sormaz and B. Khoshnevis, Process Planning Knowledge Representation using an Object-oriented Data Model, *Int. Journal of Computer Integrated Manufacturing*, Vol. 10, No. 1-4, p. 92-104, 1997.
4. C. Schlenoff, A. Knutilla, S. Ray, *Proceedings of the Process Specification Language (PSL) Roundtable*, National Institute of Standards and Technology, Gaithersburg, MD, 1997
5. S T Newman, Integrated CAD/CAPP/CAM/CNC Manufacture for the 21st Century, Invited paper *Flexible Automation and Intelligent Manufacturing, FAIM2004*, Toronto, CA, 2004.
6. S. Holzner, *Inside XML*, New Riders, Publishing, Indianapolis, Indiana,2001.
7. F. Piller, M. Müller, A new marketing approach to mass customisation, *Int. Journal of Computer Integrated Manufacturing*, Vol. 17, No. 7, pp. 583-593, 2004.
8. *The Foundation of Intelligent Physical Agents (FIPA),* http://www.fipa.org/, accessed May 2006.
9. L. H. Wang, W. M. Shen, 2003, DPP: An agent-based approach for distributed process planning, *Journal Of Intelligent Manufacturing*, 14 (5): 429-439
10. W. Shen, D.H. Norrie, 1999, Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey, *Knowledge and Information Systems, an International Journal*, 1(2), 129-156
11. B. Khoshnevis, W. Tan, and D. Sormaz, *A process selection rule base for hole-making*, Research Report, Factory Automation Systems Division, National Institute of Standards and Technology, Gaithersburg, MD, 1993.
12. A. Wadatkar, *Process Selection for Hole Operations Using a Rule Based Approach*, Msc thesis, Ohio University, 2004.

13. D. Sormaz, B. Khoshnevis Generation of Alternative Process Plans in the Integrated Manufacturing System, *Journal of Intelligent Manufacturing*, Vol 14, No. 6., 509-526, December 2003.
14. D. N. Šormaz, J. Arumugam, R. S Harihara, C. Patel and N. Neerukonda, Integration of product design, process planning, scheduling, and FMS control using XML data representation, *Robotics and Computer-Integrated Manufacturing*, Volume 26, Issue 6, December 2010, Pages 583-595
15. D. N. Sormaz, J. Arumugam, S. Rajaraman, Integrative Process Plan Model and Representation for Intelligent Distributed Manufacturing Planning, *International Journal of Production Research*, Vol. 42, No. 17, p. 3397 - 3417, 2004.
16. Dusan Sormaz, Distributed Modeling of Manufacturing Activities using Integrative Manufacturing Process Model, *International Journal of Industrial Engineering and Management* (IJIEM), Vol. 1, N.1, p. 9-18, 2010.
17. D. N. Sormaz, J. Arumugam and C. Ganduri, Integration of Rule-based Process Selection with Virtual Machining for Distributed Manufacturing Planning, in L. Wang, W. Shen, *Process Planning and Scheduling for Distributed Manufacturing*, Springer Series in Advanced Manufacturing, 2007, p. 61-90
18. J. L. Burbidge, *Production flow analysis for planning group technology*, Clarendon Press, 1989
19. W. D. Kelton, R. P. Sadowski, N. B. Swets *Simulation with Arena* McGraw-Hill Higher Education, 2009.

# DISTRIBUIRANI INTEGRACIONI MODEL ZA RAZVOJ PROIZVODA KORIŠĆENJEM AGENATA I MASOVNE KOSTUMIZACIJE

**Dušan N. Šormaz**

*Nedavni razvoj informacinoh tehnologija i interneta je omogućio skraćenje ciklusa isporuke proizvoda i poručivanje proizvoda preko interneta sa visokim nivoom zadovoljavanja zahteva potrošača (mass customization) i direktnim pristupom katalozima i bazama podataka od strane krajnjih potrošača preko interneta. Sa druge strane, proizvodnja u većini industrijskih grana je promenila akcenat polazeći od masovne proizvodnje, preko 'just-in'time proizvodnje, grupne proizvodnje na današnju proizvodnju sa brzom reakcijom u cilju skraćenje proizvodnih ciklusa. Ova dva prilaza su spojena u razvoju metoda za zadovoljenje pojedinačnih zahteva potrošača u cilju zadovoljenja potreba tržišta. U ovom prilazu svaki pojedinačni potrošač (kupac) odabira opcije proizvoda na ulazu u projektovanje (dizajn) proizvoda i planiranje proizvodnje. Ovo je omogućilo da proizvođači skrate sve komponente u ciklusu isporuke proizvoda: dizajn proizvoda, planiranje postupaka i proizvodnje, sam proizvodni ciklus i vreme isporuke gotovih proizvoda. Međutim, taj prilaz zahteva da su i razvoj proizvoda i planiranje proizvodnje potpomognuti sistemima znanja, informacija i modelima podataka uz upotrebu modernih tehnologija. Ovaj rad prikazuje neophodne tehnologije za ostvarenje distribuiranog integracionog modela za definisanje proizvoda i zadovoljenje zahteva pojedinacnih potrošača (mass customization). Sledeće tehnologije omogućavaju razvoj integracinog modela za proizvode i proizvodnju: a) upravljanje znanjem i primena znanja u odlučivanju, b) generisanje opcija za proizvode i alternative za proizvodnju, c) vizualizacija prostora odlučivanja u konfigurisanju proizvoda, d) vizualizacija proizvoda i proizvodnog procesa, e) neutralni XML format za razmenu podataka, i f) distribuirana obradu podataka i razvoj aplikacija uz primenu agenata. Primena ovih tehnologija u integracionom modeliranju i planiranju i njihovo korišćenje u razvoju prototipa IMPlanner (Intelligent Manufacturing Planner) su objašnjeni.*

Ključne reči: *aplikacije sa agentima, CAPP, planiranje proizvodnje, razvoj proizvoda*