Manufacturing Planning and Predictive Process Models Integration Using Software Agents

Shaw C. Feng, Keith A. Stouffer, and Kevin K. Jurrens
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland 20899, U.S.A
{shaw.feng | keith.stouffer | kevin.jurrens}@nist.gov

Abstract

Intelligent agents provide a means to integrate various manufacturing software applications. The agents are typically executed in a computer-based collaborative environment, sometimes referred to as a multi-agent system. The National Institute of Standards and Technology (NIST) has developed a prototype multi-agent system supporting the integration of manufacturing planning, predictive machining models, and manufacturing control. The agents within this platform have access to a knowledge base, a manufacturing resource database, a numerical control programming system, a mathematical equation solving system, and a computer-aided design system. Intelligence has been implemented within the agents in rules that are used for process planning, service selection, and job execution. The primary purposes for developing such a platform are to support the integration of predictive models, process planning, and shop floor machining activities and to demonstrate an integration framework to enable the use of machining process knowledge with higher-level manufacturing applications.

Keywords: intelligent agents, manufacturing planning, machine control, multi-agent system, systems integration.

1. Introduction

Current computer-aided design (CAD) and computer-aided manufacturing (CAM) systems are typically operated in a stand-alone mode. According to a Daratech survey [1], the CAD/CAM industry is a $5.3 billion (USD) industry worldwide. Based on industrial needs, it is necessary to integrate other software systems, tools, existing databases, or knowledge bases with CAD/CAM systems. For example, most commercially available CAD/CAM systems do not have embedded databases for attributes like cutting parameters. Manufacturing engineers must look up the recommended cutting parameters in handbooks or from sources provided by vendors. This information may be outdated, not relevant for particular situations, or overly conservative. Scientifically developed mathematical models of cutting processes have been developed, such as models for predicting machining stability and estimating cutting tool life [2]. These process models and the machining process knowledge they contain must be made available to design and manufacturing engineers within the appropriate CAD/CAM systems early in the product lifecycle to have the largest impact. The use of sub-optimal machining process parameters has a significant impact on manufacturing costs, causing longer machining times, shortened tool life, and reduced part accuracy.

Some fundamental problems that cause the manufacturing inefficiency include:

(1) Interoperability among heterogeneous software systems and tools is generally not available. The design, manufacturing, and supply chain software systems are not interoperable across software ownership boundaries.

(2) Computing environments are usually distributed, such as those for design knowledge bases, material databases, resource databases, manufacturing process modeling systems, and process knowledge bases. Additionally, these systems are not linked with CAD/CAM systems.

(3) The use of sub-optimal process parameters increases the manufacturing cost because of increased tool wear or under-utilized tools and machines. Optimal cutting parameters are not always available to manufacturing engineers because the Numerical Control (NC) programming systems and physics-based cutting process models are not linked with each other.

(4) Web-based collaboration between the process planning activity and the process parameter optimization activity is currently not available. This impedes collaborative activities in a distributed computing environment within a company's engineering department.

To alleviate these problems, industry requires technologies that can enable interoperability among software systems in a distributed computing environment and integrate various capabilities into the user's software systems. Some specific solutions are as follows:

(1) Mechanisms for software components to communicate with each other across computer networks with as few changes as possible to the original software,

(2) Processes (software behavior) that can be changed and services that can be added without changing existing codes,

(3) Software capabilities that are defined and registered in a database where others can access,

(4) Representation of the domain knowledge and objects shared by different software systems is open and well defined, and

(5) Web-accessibility built into agents so that they can be remotely controlled and monitored. The agents should enable collaborative work in process planning and design.

The objective of the work described in this paper is to use autonomous agents [3] to integrate manufacturing applications, including manufacturing process planning and machine control, and supporting software, such as knowledge base systems, database management systems, and mathematical modeling software tools. The capability to support the manufacturing planning and control functions is embedded in software agents. The functions in the manufacturing software become the agents' capabilities. Agents communicate with each other using an augmented Agent Communication Language (ACL) [4] by sending requests and responses in a standard format with necessary information. The message content format for our applications is not supported by the current ACL specification. Message content format and interaction protocols are defined as an extension of ACL. The purpose is to integrate commercial software systems across system boundaries. Specifically, we use the scenario of process planning and NC machine control as an example to illustrate a prototype integration framework based on machining process models, an agent knowledge base, a manufacturing resource database, and interfaces to both the process planning application and the machine controller.

An agent-based approach is used to take advantage of the flexibility offered when needing to integrate additional software tools or to add additional software agent functionality. When a software agent needs to behave differently, only the agent's knowledge representation must be changed. The code needs little or no changes. Standards published by the Foundation of Intelligent Physical Agents (FIPA) [5] are available for multi-agent system implementations, and have been adopted in this work.

The rest of this paper is organized as follows. Section 2 of this paper provides a survey of the current state of intelligent agents in design and manufacturing. Section 3 describes the proposed integration framework for process planning and machine control for automated manufacturing. Section

4 presents the implementation of the integration framework using available tools. Section 5 presents some conclusions from this work, and is followed by the disclaimer and references.

2. The State of Multi-agent Systems in Manufacturing Applications

In prior work [6], an architecture for integrated knowledge-based preliminary process planning functional components has been developed. The architecture consists of an inference engine, a product database, a manufacturing knowledge base, and a resource database. The inference engine is used to select preliminary manufacturing processes and resources, and to estimate manufacturing cost/time based on manufacturing knowledge according to product information. The product database stores product information pertinent to preliminary process planning. The manufacturing knowledge base has process planning rules. The manufacturing resource database stores descriptions of tools, machines, labor skills, and computer software capabilities. The semantics for information exchange used by agents between design and process planning is based on an integrated design and manufacturing object model, described in [7, 8].

Potential applications of multi-agent systems in concurrent design and manufacturing have been described in [9, 10], with examples given in the areas of product design, manufacturing planning, shop-floor control, and manufacturing equipment control. A design and manufacturing optimization environment has been developed in [11] using a web- and agent-based framework. Multi-agent systems have also been applied to the area of shop-floor scheduling, to address scheduling of machine tools based on job requirements [12]. The agents negotiate with each other to find the best utilization of resources, including cutting tools, machine tools, and fixtures.

The lack of an integration scheme for the integration of design and manufacturing applications costs industry a great deal in using manufacturing software tools. The tools are typically isolated and cannot interoperate across system boundaries. Multi-agent technology provides software developers with methods and tools to link various engineering software tools. This technology can enable the flexible integration that is needed by industry.

3. Integration Framework

A framework for integrating predictive process models, process planning systems, design systems, and machine tool control that enables the flexible integration of heterogeneous manufacturing applications has been developed by staff of the NIST Predictive Process Engineering (PPE) program. The framework describes the agent behavior, communication methods, and agent system structure. The integration framework includes the component architecture, interaction sequence model, message and communication format, agent platform, process knowledge, and supporting tools. This framework has an emphasis on information requirements and modeling for enabling the integration of software and hardware systems for integrated predictive process engineering.

Figure 1 goes here
Figure 2 goes here
Figure 3 goes here

Figure 4 goes here

3.1 Component Architecture
The component architecture describes the software components used in the agent-based integrated design, planning, and control system. Figure 1 shows agents and an agent group that are executed in

the agent platform. The agents include a design agent, a group of process planning agents, the capability repository agent, and the manufacturing control agent. Humans interact with the agents through the graphical user interfaces (GUI). The design agent communicates with a CAD system to send and retrieve information about part design. A CAD system defines the shape and attribute information of a design of part components or an assembly. Figure 2 shows agents in the process planning agent group. The process planning agent communicates with a Computer-Aided Process Planning (CAPP) system and a CAM system to generate numerical control (NC) programs. CAPP software provides functions for selecting processes and resources and for generating process sequences. The agent intelligence, in the form of rules, is stored in and can be retrieved form the knowledge base by an agent. The NC software is used to create tool paths and NC programs for running computer numerically controlled machine tools. The agent sends and retrieves information about machining process planning to and from both CAPP and CAM systems. It also interacts with the tool material selection agent to select an appropriate cutting tool material based on the product attributes. The tool material selection agent selects the cutting tool material based on workpiece material, tool life requirements, and desired cutting speed. Tool material selection rules are stored in a knowledge base. The maximum material removal rate evaluation agent uses a mathematical model to calculate the maximum material removal rate based on the dynamics of the specific machine tool, the configuration of the cutting tool/tool holder/machine spindle, and factors in the machining process. The agent calculates an optimal set of cutting parameters, including cutting speed, feed rate, and depth of cut, based on the specific conditions. Agents execute on an agent platform, which manages the messages between agents. Figure 3 shows that the capability repository agent performs on the agent platform and uses a database for storing and retrieving the capability information about each agent that provides services. The agent platform manages the agent activities. Figure 4 shows the platform functions and components, which are largely based on the FIPA specifications. The major functions include agent management, message transport service, and agent directory facility. The messages sent by agents are dispatched and directed by the platform.

Figure 5 goes here

3.2 Interaction Model

The interaction model specifies the timing and sequence of function calls – the interactions among agents. Figure 5 shows interactions among agents for the scenario used in the integration framework. All the agents, except the process planning and capability repository agents, must first register their capability in a database. The database stores the capability information, i.e., the service that a service provider agent can provide. Note that the Directory Facility in Figure 4 only manages agents checking in and out and the addresses of those agents, not the capability information. The capability repository agent manages the registration and deregistration process. From within the CAM system, the manufacturing engineer launches the process planning agent to determine and provide the optimal cutting parameters. The process planning agent then requests the capability repository agent to provide information on agent(s) who can select cutting tools. Based on the knowledge in the capability repository, the capability repository agent replies that the tool selection agent provides the service of cutting tool selection. The process planning agent obtains the tool material information by sending a request to the tool material selection agent. With this information, the process planning agent then interacts with the Maximum Material Removal Rate (MMRR) model agent to determine an

optimal set of process parameters based on analysis from both the machining stability perspective (i.e., highest material removal rate while maintaining a stable cut) and the cutting tool life perspective (i.e., recommended material removal rate to achieve specified cutting tool wear rate and tool life). The MMRR model agent interacts with both the machining stability model agent and the tool life model agent. With the result, the NC program can therefore be completed within the CAM system. The CAM system operator then send the completed NC program to the manufacturing execution agent to machine the part.

3.3 Message Format

Agents communicate with each other by exchanging messages. A message has two sections: message header and message content (also known as message body).

The header contains the information regarding the sender, receiver(s), subject, date, and time that the message is sent by the sender, date and time that the message is received by the receiver, and the priority. In the sender and receiver, there is a slot that contains the agent identification (AID). Using the AID, the information about an agent, such as its name, can be retrieved from a repository or agent directory.

The message content contains information regarding the intent. There are two types of content: illocutionary and perlocutionary. The illocutionary message is used to inform other agents, such as registering an agent's capability. The perlocutionary message is used to request actions of other agents, such as a request for machining process optimization or a call for proposal. Message content has the following attributes: an action verb, an object, preconditions, and constraints. The action verb is used to indicate the type of action to be taken by the receiver, such as request, propose, and query. An object is the result or expectation. Examples of classes of objects can be found in [8]. Classes related to the milling process have been applied. Preconditions are the properties that the sender may supply. Constraints are limitations with which expected results should be constrained. They are specified to provide information to the receivers to produce valid results. Agents use the predefined knowledge base to obtain intelligence to process messages. A body of knowledge supports the intelligence of an agent.

3.4 Knowledge

The source of agent intelligence is the knowledge base. This knowledge base contains rules that govern the agent behaviors. An agent's knowledge includes how to inquire about the capability of other agents and how to perform special tasks. For example, special tasks may include functions such as tool material selection, machining stability analysis, or tool life evaluation. Basic rules for tool material evaluation can be commonly found in literature on process planning for metal cutting [13]. More specific and customized rules can be entered into the knowledge base. The rules are structured using propositional logic. The rules for both machining stability analysis and tool life evaluation are in mathematical form. The machining stability analysis is based on machining measurements and a machining chatter analysis and the tool life evaluation is based on the Taylor tool wear principle, and both methods are documented in the literature [2]. These two mathematical models are implemented using available mathematical software tools.

3.5 Manufacturing Resource Database

The manufacturing resource database contains information about the equipment and tools used in the machining process, such as data and attributes for the machine tools, cutting tools, and fixtures.

The structure of the database is relational, and the database can be accessed by external programs via the Java database connector. In the database, a machine tool is defined by a set of attributes, such as the maximum power, the maximum cutting force in each axis, the maximum workspace dimensions, tool magazine information, the number of cutting axes, the maximum spindle speed, and available cutting tools. A cutting tool is defined by another set of attributes, such as tool identification, tool length, tool size, number of cutting edges, cutting angles on each edge, tool material, and tool overhang as mounted in its holder. Other resource information, such as for fixtures, workpiece materials, and operator skills can also be stored in the database. An object model on machining resource information can be found in [7].

4. Implementation

Several agents, software integration methods, a knowledge base, and a resource database have been implemented based on the Java Agent Development Framework (JADE) [14] a multi-agent platform. Each agent has a specific goal and is programmed with specific functions. Most agents are supported by software tools and all run on the JADE platform.

4.1 Process Planning Agent

Four agents have been implemented in the process planning agent group to perform tasks related to process selection, tool selection, process parameter selection, and tool life estimation. Respectively, these are the process planning agent, tool material selection agent, maximum material removal rate model agent, and tool life evaluation agent. The process planning agent generates the machining process plan and NC programs for controlling computer numerically controlled (CNC) machine tools. The NIST implementation uses Pro/Advanced_NC as the CAM system to first generate the machining plan. All the machining operations, workstations, tools, and cutting sequences are defined in this system. Using an interface developed using Pro/Toolkit, Pro/Advanced_NC interacts with the process planning agent, and data is transferred through a pair of Transmission Control Protocol/Internet Protocol (TCP/IP) [15] sockets. This agent communicates with the tool material selection agent to recommend cutting tool material by providing information about workpiece material, production rate, production quantity, and cost constraints. The agent also communicates with the maximum material removal rate model agent to obtain optimal process parameters, including cutting speed, feed rate, and depth of cut. After a process plan is created, NC programs are then generated. The agent also requests estimates of the manufacturing time and cost. The generated process plan, NC program, and information about machining time and cost are stored by the process planning agent and ready to send to other agents upon request.

4.2 Maximum Material Removal Rate (MMRR) Model Agent

The MMRR model agent has the capability and knowledge to determine the set of process parameters that result in the highest material removal rate for milling operations while maintaining a stable cut (i.e., without machine tool chatter). The process knowledge is captured in a set of mathematical representations in the machining stability model agent. Based on the dynamic characteristics of the machine tool, cutting tool, tool holder, and the interaction with the workpiece, a suggested maximum material removal rate is calculated in terms of the depth of cut, cutting speed, and feed rate. The machining knowledge has been implemented through a set of equations in Matlab. The agent interacts with MatLab using the application programming interface (API) in the C programming language, and the data is transferred through TCP/IP sockets. The depth of cut,

cutting speed, and feed rate are needed to complete the NC program generation. The suggested maximum material removal rate will be adjusted based on the tool life evaluation to be provided by the tool life model agent.

4.3 Tool Material Selection Agent

The tool material selection agent is structured so that it has the knowledge to suggest tool material. Tool material selection is based on the conditions of the workpiece material, production rate, and product quantity. The selection rules are in propositional logic and stored in a knowledge base developed using the Java Expert System Shell (JESS) [16]. JESS is also used as an inference engine. Based on the conditions, JESS returns the suggested tool material. The agent interacts with JESS via the Java API built in JESS to retrieve the knowledge by firing rules. Rules for this knowledgebase can be typically found in sources such as manufacturer's catalogs, reference books, and other literature.

4.4 Tool Life Model Agent

The tool life Model agent has the capability and knowledge to estimate the usable life of a cutting tool before the tool wear impacts the accuracy of the resulting parts. A common economic model for tooling states that machining cost is a function of the tool cost and the tool wear, which is related to cutting speed based on the Taylor tool life principle. The tool wear aspect of this model is formulated in a set of mathematical equations and implemented in MatLab. This agent interacts with the MatLab software similar to the maximum material removal rate agent. The input parameters include the tool material, production rate, cost of tools change, and range of cutting speeds. The output is the recommended cutting speed that minimizes the machining cost.

4.5 Capability Repository Agent

The capability repository agent has two responsibilities: (1) to enter an agent's capability into the agent capability database upon request and (2) to retrieve the information to those agents that need it. This agent keeps active by listening to requests from new agents entering into the agent platform. The database contains information about the agent identification, name, access methods, and capability. The description of capability is in a specific format and uses pre-defined terms. The description and terms are available to agent programmers so that agents can use them to locate and select appropriate (agent) service providers.

4.6 NC Machining Agent

The NC Machining agent interfaces with a CNC machine tool. The machine tool for this implementation is a Bridgeport 3-axis milling machine that was retrofitted with an Enhanced Machine Controller (EMC). The EMC is a personal computer (PC)-based, open architecture controller for machine tools that consists of a hierarchy of modules based on the NIST Real-Time Control System (RCS) Reference Model Architecture [16]. The controller has standard interfaces between functional modules for servo control, trajectory generation, constrained motion modes, discrete event logic, task sequencing, and a graphical user interface. This EMC port is a full-software version running on Linux, using real-time Linux for motion control. The primary mission is to receive NC programs from the Process Planning agent and start the CNC machine tool with the NC programs to machine a part. This agent interacts with the open machine controller, described in Section 3, installed on a CNC machine tool. The agent and the process planning agent established a communication link between the machine tool and a CAM system. In order for the user to see the machining process, a controllable

pan/tilt/zoom camera has been implemented to send video to the user. This allows the user to monitor the machining with a PC using a web-browser and an internet connection. The video within the remote access page is a server-pushed Motion-JPEG image with an imagemap overlaid on it. The resolution of each image is 352 x 240 pixels with a video frame rate of 30 frames/second. Each image of the video has an overlaid imagemap, allowing the collaborator to click on the image where the new center of interest should be. The required pan/tilt angles are calculated and the camera moves to recenter the image. The zoom of the camera is controlled by a graphical scrollbar.

Figure 6 goes here
Figure 7 goes here
Figure 8 goes here
Figure 9 goes here

4.7 Agent graphical User Interface

The graphical user interface to the agents has been developed using Java Swing. The agent graphical user interface has two primary functions: to start an agent and to inform the user about how the agent is performing. Figure 6 shows panes in the typical user window. As shown in this figure, the user can start an agent. Figure 7 informs the user about connection information for application software used by the agent. Figure 8 illustrates sample messages that have been sent by this agent and received from other agents. The messages contain the respective agent name and message content, and can be set to also display the time and date of the message if desired. Figure 9 shows a sample status display for the agent while performing tasks, such as information about what function is started or finished and what message is sent or received. The graphical user interface provides a window for users to "see" what and how an agent is working within the multiagent platform.

5. Concluding Remarks

The NIST PPE program has developed and demonstrated a prototype agent-based integration framework to enable the use of machining process knowledge with higher-level manufacturing applications. Within this framework, data can be transmitted from a client agent to the appropriate service-providing agent(s). Agents handle communications and negotiations between software systems, which do not communication and negotiate with each other. Wrapped by agents, the systems, the database, the knowledge base, and the mathematical solving software can interoperate with each, i.e., agents enable interoperability. Additionally, all the agents are Web-accessible so that designers, process planners, and NC programmers can collaborate to search for the optimal process plan for a designed part.

The intelligent agents are used to optimize process performance through standard agent communication protocols and implemented mathematical process models with specified goals and constraints. The process planning agent optimized the NC programs with regard to tool life and material removal rate using services provided by two additional agents. This multiagent system has demonstrated an approach for system interoperability. The agents communicate with each other using the FIPA standard message format and a common vocabulary. When the system is further expanded, the agents will also compete with each other for providing the best possible services.

Disclaimer

No approval or endorsement of any commercial products by the National Institute of Standards and Technology is intended or implied. Certain commercial software and hardware systems are

identified in this paper in order to facilitate understanding. Such identification does not imply that these systems are necessarily the best available for the purpose.

Acknowledgement

References

1 MCAD News, "Mechanical CAD/CAM Market Forecast to Top $5.3 Billion in 2001," Daratech Incorporated, 9 March 2001.
2 G. Boothroyd and W. Knight, <u>Fundamentals of Machining and Machine Tools</u>, 2nd edition, Marcel Dekker, Inc. published, 1988.
3 M. Wooldridge, <u>Introduction to Multi-Agent Systems</u>, John Wiley & Sons, 2002.
4 Agent Communication Language (ACL), Foundation of Intelligent Physical Agents ACL Message Structure Specification, August 2001.
5 The Foundation for Intelligent Physical Agents, http://www.fipa.org.
6 S. Feng and E. Song, "Preliminary Design and Manufacturing Planning Integration Using Intelligent Agents," the proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design (CSCWiD), Rio de Janeiro, Brazil, September 2002, pp. 270 – 275.
7 K. Jurrens, M. Algeo, and J. Fowler, "Beyond Product Design Data: Data Standards for Manufacturing Resource," <u>Rapid Response Manufacturing: Contemporary methodologies, tools and technologies</u>, edited by J. Dong, Chapman and Hall, London, Chapter 6, pp. 84-120, 1998.
8 S. Feng and E. Song, "A Manufacturing Process Information Model for Design and Process Planning Integration," Journal of Manufacturing Systems, Vol. 22, No. 1, 2003, pp. 1- 16.
9 W. Shen, D. Nore, and J. Barthes, <u>Multi-agent Systems for Concurrent Design and Manufacturing</u>, Taylor and Francis Publishing Co., 2001.
10 G. Tian, G. Yin, and D. Taylor, "Internet-based manufacturing: A review and a new infrastructure for distributed intelligent manufacturing," Journal of Intelligent Manufacturing, Vol. 13, No. 5, 2002, pp. 323 – 338.
11 W. Shen, Y. Wang, Y. Li, Y. Ma, and H. Ghenniwa, "WebBlow: A Web/Agent based MDO Environment," the proceedings of the 7th International Conference on CSCWiD, Rio de Janeiro, Brazil, September 2002, pp. 251 – 257.
12 W. Shen, D. Norrie, "Dynamic Manufacturing Scheduling Using Both Functional and Resource Related Agents," Integrated Computer-Aided Engineering, 2002, 8(1), pp. 17 - 30.
13 D. Nelson and G. Schneider, <u>Applied Manufacturing Process Planning</u>, Prentice-Hall, Inc., 2001.
14 Java Agent Development Framework, http://www.cselt.it.
15 Comer, D., <u>Internetworking with TCP/IP</u>, Vol. 1, Principles, Protocols, and Architecture, 3rd edition. Prentice Hall, 1995.
16 Java Expert System Shell, http://herzberg.ca.sandia.gov/jess.
17 J. Albus and A. Meystel, "A Reference Model Architecture for Design and Implementation of Intelligent Control in Large and Complex Systems," International Journal of Intelligent Control and Systems, Vol. 1, No. 1, 1996, pp. 15 - 30.
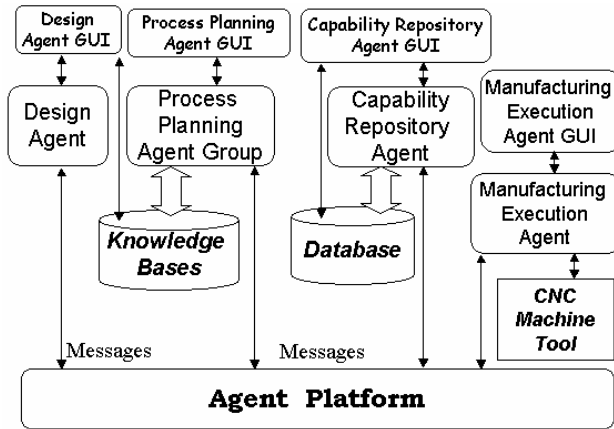
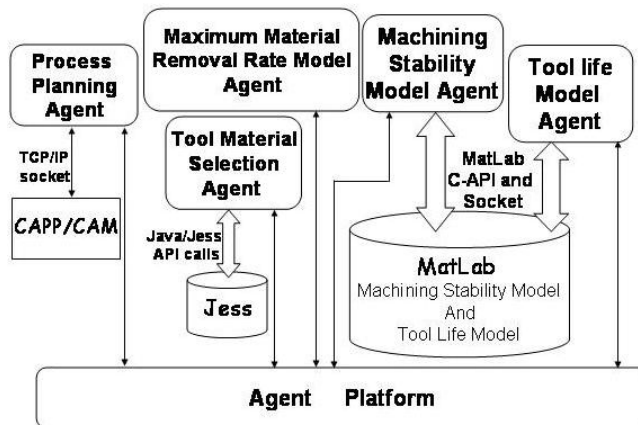**Figure 1** Multi-agent system architecture
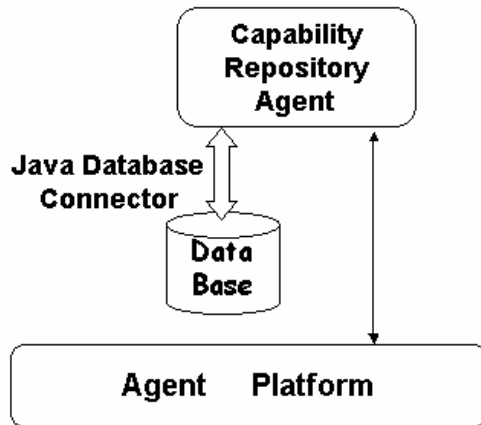

Figure 2 Process planning agent group

Figure 3 Capability repository agent
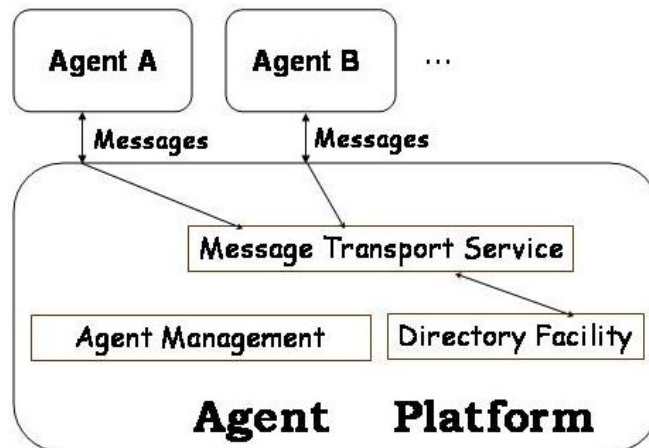

Figure 4 Agent platform
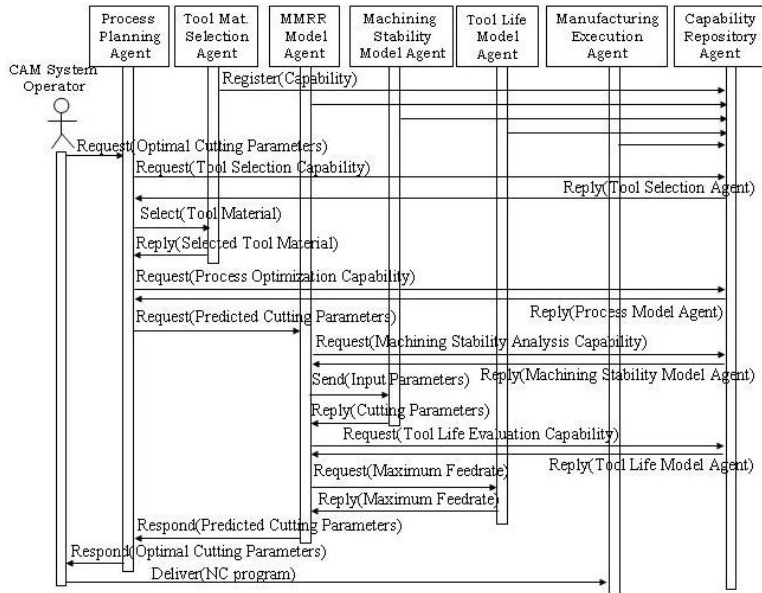
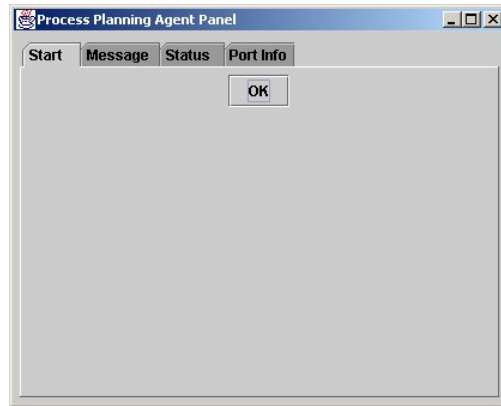**Figure 5** Agent interaction diagram

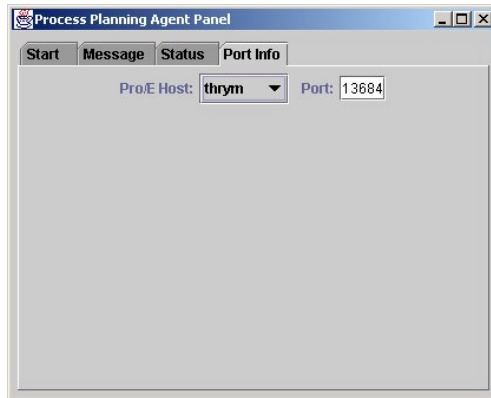**Figure 6** GUI for starting an agent
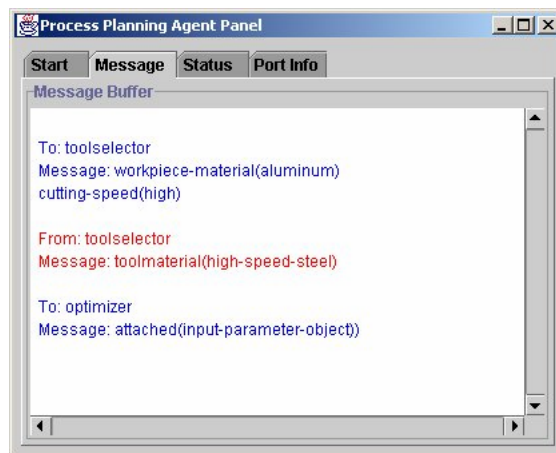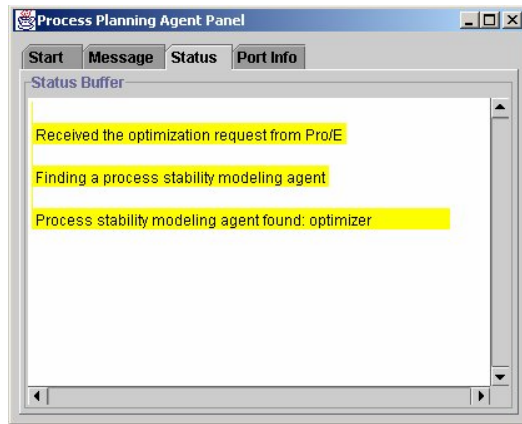
Figure 7 Port to an application
software tool

Figure 8 Messages in an agent

Figure 9 Status of an agent