International Journal of the Korean Society of Precision Engineering, Vol. 1, No. 1, June 2000.

Development of Flexible Manufacturing System using Virtual Manufacturing Paradigm

Sung-Chung Kim* and Kyung-Hyun Choi**

* School of mechanical engineering, Chungbuk National University, Cheongju, South Korea,
**School of mechanical engineering, Cheju National University, Cheju, South Korea

ABSTRACT

The importance of Virtual Manufacturing System is increasing in the area of developing new manufacturing processes, implementing automated workcells, designing plant facility layouts and workplace ergonomics. Virtual manufacturing system is a computer system that can generate the same information about manufacturing system structure, states, and behaviors as is observed in a real manufacturing. In this research, a virtual manufacturing system for flexible manufacturing cells (VFMC), (which is a useful tool for building Computer Integrated Manufacturing (CIM),) has been developed using object-oriented paradigm, and implemented with software QUEST/IGRIP. Three object models used in the system are the product model, the facility model, and the process model. The concrete behaviors of a flexible manufacturing cell are represented by the task-oriented description diagram, TID. An example simulation is executed to evaluate applicability of the developed models, and to prove the potential value of virtual manufacturing paradigm.

Key Words : FMS, virtual manufacturing system, CIM, object-oriented paradigm, TID

1. Introduction

Recent trends in manufacturing systems, such as the need for customized products by small batches and for fast product renewal rates, have been demanding new paradigms in manufacturing. Therefore, the modern manufacturing systems are needed to be adaptable, and have the capability to reconfigure or self configure their own structure. Flexible Manufacturing Cells (FMCs) are generally recognized as the best productivity tool for small to medium batch manufacturing, and are also basic unit to construct a shop floor which is an important level for developing computer integrated manufacturing (CIM). However, due to its complexity, the modeling and operation methodology related to FMC should be verified before implementation.

As one of approaches to these requirements, Virtual Manufacturing (VM) approach has been introduced, and known as a effective paradigm for generating a model of manufacturing systems and simulating manufacturing processes instead of their operations in the real world. VM pursues the informational equivalence with real manufacturing systems. Therefore, the concept of Virtual Manufacturing System is expected to provide dramatic benefits in reducing cycle times, manufacturing and production costs, and improving communications across global facilities to launch new products faster, improve productivity and reduce operations costs for existing product shop [1,2].

With an object-oriented paradigm, computer-based technologies such as virtual prototyping and virtual factory are employed as a basic concept for developing the manufacturing processes, including the layout of the optimal facility, to produce products. Virtual prototyping is a process by which advanced computer simulation enables early evaluation of new products or machines concept without actually fabricating physical machines or products. Bodner, et al.,[3] concentrated on the decision problems associated with individual machines that assemble electronic components onto printed circuit boards (PCBs). Virtual factory is a realistic, highly visual, 3D graphical representation of an actual factory floor with the real world complexity linked to the production controlling system and the real factory. Virtual factories are increasingly used within manufacturing industries as representations of physical plants, for example, VirtualWork system for representation of shop floor factory[4].

Despite its benefits and applicability, VM systems should deal with a number of models of various types and require a large amount of computation for simulating behavior of equipment on a shop floor. To cope with this complexity in manufacturing, it is necessary to introduce open system architecture of modeling and simulation for VM systems.

In this paper, three models, which are product, device, and process models will be addressed. Especially process model for FMC will be emphasized using QUEST/IGRIP as an implementation issue. The open system architecture consists of well-formalized modules for modeling and simulation that have carefully decomposed functions and well-defined interface with other modules

2. Concept of virtual manufacturing

Virtual Manufacturing System is a computer model that represents the precise and whole structure of manufacturing systems and simulates their physical and logical behavior in operation, as well as interacting with the real manufacturing system. Its concept is specified as the model of present or future manufacturing systems with all products, processes, and control data. Before information and control data are used in the real system, their verification is performed within virtual manufacturing environment. In addition, its status and information is fed back to the virtual system from the real system.

Virtual environments will provide visualization technology for virtual manufacturing. The virtual prototype is an essential component in the virtual product life cycle, while the virtual factory caters for operations needed for fabricating products. Therefore, the developments in the area of virtual prototyping and virtual factory will enhance the capabilities of virtual manufacturing.

The major benefit of a virtual manufacturing is that physical system components (such as equipment and materials) as well as conceptual system comp vonents (e.g., process plans and equipment schedules) can be easily represented through the creation of virtual manufacturing entities that emulate their structure and function. These entities can be added to or removed from the virtual plant as necessary with minimal impact on other system data. The software entities of the virtual factory have a high correspondence with real system components, thereby lending validity to simulations carried out in the virtual system meant to aid decision-makers in the real system.

For virtual manufacturing, three major paradigms have been proposed, such as Design- centered VM, Production-centered VM, and Control- centered VM. The design-centered VM provides an environment for designers to design products and to evaluate the manufacturability and affordability of products. The results of design-centered VM include the product model, cost estimate, and so forth. Thus, potential problems with the design can be identified and its merit can be estimated. In order to maintain the manufacturing proficiency without actual building products, production-centered VM provides an environment for generating process plans and production plans, for planning resource requirements (new equipment purchase, etc.), and for evaluating these plans. This can provide more accurate cost information and schedules for product delivery. By providing the capability to simulate actual production, control-centered VM offers the environment for engineers to evaluate new or revised product designs with respect to shop floor related activities. Control-centered VM provides information for optimizing manufacturing processes and improving manufacturing systems.

The virtual manufacturing approach in this paper is close to Control-centered VM. Fig.1 illustrates the viewpoint of the functional model of the virtual flexible manufacturing cell. Since the activity Execute real manufacturing systems depicts a model of real factory, it possibly replaces real factory. All manufacturing processes except physical elements of virtual manufacturing, such as design, process planning, scheduling, are included in the activity Operation of Virtual factory. The activity Execute simulation for virtual factory is a separate simulation model of VM system. With this virtual factory, parameters (e.g, utilization, operation time, etc.,) associated with operating a flexible manufacturing cell are simulated. And these results can provide the possibility of controlling manufacturing processes and predicting potential problems in the real manufacturing.

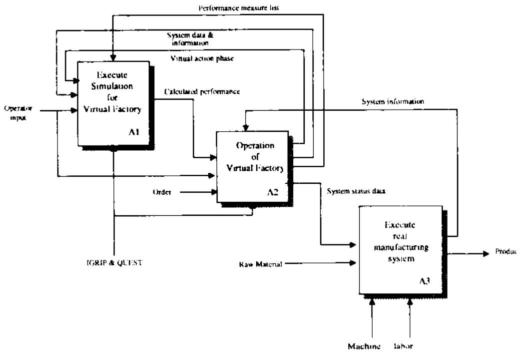

Fig. 1 IDEF0 model for virtual manufacturing

3. Object modeling for virtual flexible manufacturing cells

Object-oriented technology may provide a powerful representation and classification tools for a virtual flexible manufacturing cell. It may also provide a common platform for the information sharing between sub-modules, and provide a richer way to store/retrieve/modify information, knowledge and models and reuse them. In the context of an object oriented approach, a model is simply an abstraction, or a representation of an objects or process.

VFMC requires a robust information infrastructure that comprises rich information models for products, processes and production systems. As shown in Fig. 2, three models, that is product model, facility model, and process model, are developed for virtual flexible manufacturing cells.

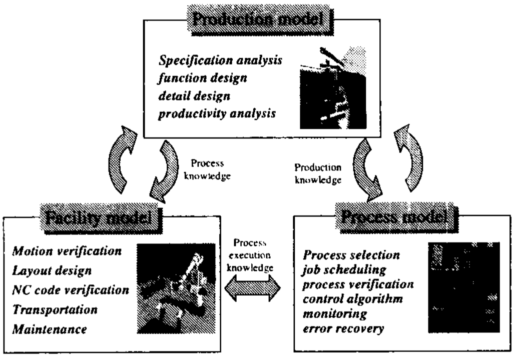

Fig. 2 Object models utilized in virtual flexible manufacturing cell

A product model is a generic model used for representing all types of artifacts, which appear in the process of manufacturing. It represents target products, which include conceptual shape information as well as analysis module for a specification, productivity, and strength. A facility model contains information about machines consisted of a virtual flexible manufacturing cell. By using the model, innovative tooling and methods can be evaluated without the cost of physical machine prototypes and fixture mock-ups. A process model is used for representing all the physical processes that are required for representing product behavior and manufacturing processes.

3.1 Product model

A product model holds the process and product knowledge to ensure the correct fabrication of the product with sufficient quality. It acts as an information server to the other models in the VFMC. It also provides consistent and up-to-date information on the product lifecycle, user requirements, design, and process plan and bill of material. An instance of Class **Part** provides detailed information about a part to be fabricated in VFMC. Sub-classes like ProcessPlan, BOM, and NcCode, are aggregated into the class **Part**. Classes ProcessPlan and BOM manipulate information and data associated with process plans and bill of materials, respectively. Class NcCode deals with NC programs, which interacts with CAD/CAM systems. With incorporation with the facility model, this developed NC programs can be verified and checked for collisions and interference with any workpiece or tooling in the fixture. This can avoid costly machine crashes and reduce risk during initial equipment installation and produce launch. Furthermore, productivity can be improved by avoiding nonproductive time for program prove out on the machine tool and by using the simulation environment to train operators of new machines.

3.2 Facility model

Real manufacturing cell may consist of NC machines, robots, conveyors, and sensory devices. The architecture of class corresponding to the real manufacturing cell is shown in Fig. 3, and represents the factory model. In VFMC, characteristics of the factory model include a detailed representation of machine behavior over time, a structure to the model that can configure and reconfigure easily, and a realistic and three-dimensional animation of machine behavior over time. Virtual machines defined within this model may be used to estimate accurately the merit of a process plan, and, based on this evaluation, determine appropriate process conditions to improve (and even optimize) the plan. Virtual robot contributes to unload and load parts into/from machines, and is used to find optimal paths without any collisions. With virtual operation, the fidelity of the machining and robot utilizing time and cost estimates is expected to improve. In addition, accurate modeling will predict the quality of the machined part, which cannot be determined easily and reliably without producing several physical prototypes. This information is invaluable to both the designer and the process planner. Physical entities such as machines and workpieces have the explicit representation as 3-D models for their shapes, positions, and orientations. 3-D models are conveniently used for calculating, geometrical attributes, checking spatial relations, and displaying computer graphics.

3.3 Process model

By assigning a finite set of states to each device in a cell (idle, busy, failed, etc.), the process of cell control can be modeled as a process of matching specific state change events to specific cell control actions, decision algorithms, or scripts. With this model, cell processes are represented a Task Initiation Diagram (TID) using an object-oriented approach. The methodology behind developing TID regards the tasks to be performed by the cell or any of its constituent machines for being primal, and employs the multi-layered approach. Sensory signals indicating the change of state of machines are used to trigger or initiate tasks. A task may be simple and require a relatively short time to execute, or may be complex and lengthy.

Formally, a Task Initiation Diagram (TID) is defined as the four-tuple TID=(T, SR, C, O). Task Initiation Diagrams are composed of two basic components: a set of Rest states SR and a set of tasks *T*. Tasks, in turn, are classified into three groups: the cell configuration dependant task (*Td*), the cell configuration independent task (*Ti*), and the cycle transit task (*Tt*). Cell configuration dependent tasks are those which require some coordination among cell components to carry out the task. For example, the task **load** as in aRobot load a part to:aMill requires that the actions of aRobot and aMill be coordinated. Cell configuration independent tasks require only one cell component to perform the task. The task **moveTo** as in Robot move to:*MachineName* is an example of a cell configuration independent one, because it is carried out by the Robot without interacting with other components. *Tt* tasks are used for the transition from one cycle to another, and thus derived automatically by the system in order to complete a production job. State SR indicates rest states where cell constituents must be wait for next task. This state is given at any instant by the collection of states of its constituents. These composite states are depicted in the Task Initiation Diagram by ellipses, e.g., *R11/3* or *M13/4*. The last number of the symbols indicates how many individual states are required to determine this composite state.

To complete the diagram, it is necessary to define the relationship between the states and the tasks. This can be done by specifying two functions connecting states to tasks: the condition function **C**, and the output function **O**. The condition function **C** defines, for each task *Ti*, the set of states for task C(*Ti*). Some condition functions may use guiding parameters in addition to a set of states. As an example, C(*Ti*) uses a Remaining Processing Time (RPT) to cause transition to the desired state. The output function **O** defines for each Task *Ti* the set of output States for the transition O(*Ti*).

The Operation Initiation Diagram (OID) is the second layer diagram of the Task_Initiation Diagram (TID). In the same way of TID to represent the model, the Operation Initiation Diagram OID is defined as the four-tuple, OID(task)=(OP,Sv,C,O). The symbol OP defines set of operation required for a given task. The operation, OP, is categorized into two groups: guided operations OPg and unconditional operations OPu. A guided operation is one that requires an external trigger to start it. Unconditional operations are ones that start automatically on the onset of all the necessary states.

The symbol Sv indicates the set of visit-state. The **visit**-state, Sv, indicates an interaction between two machines and hence requires coordination among them. The symbol of this state has the pattern **R-M--** for the robot, as an example, the state *RvMnm*. The small letter *v* represents the visit-state of the robot associated with location, *Mn* represents a machine served by the robot, and *m* represents the index of one of the visit locations. During the completion of the task, the **busy** states are employed, and indicate transitional states between operations or two executions without interaction. They can be recognized from the robot state symbol, *Rtn*. The small letter *t* indicates the state of the robot associated with transition. These states are useful in avoiding collisions with obstacles. The condition operator **C**, defines the set of state and guiding conditions necessary for each operation OPi i.e. C(Opi). The output operator **O**, defines the set of states resulting from each operation OPi, i.e., O(OPi).

4. Control architecture for VFMC

Cell operation involves tasks to be performed on single machines independent of others, and tasks that to require the cooperation of two or more machines. In cases where a task calls for the coordination of two or more machines, the cell controller has to be involved to ensure proper execution of that task. For tasks involving a single machine, the primary function of a controller is to schedule the start of the task, and waits for its completion to command the nest task. In order to accomplish these functions, the cell controller is designed as a hybrid structure of both hierarchical controller and decentralized controllers as shown in Fig. 3. The controller consists of three different layers. The Scheduler, the Decentralized Control layer, and the Virtual Device layer. In the figure, the passing of information and message are indicated by arrows.

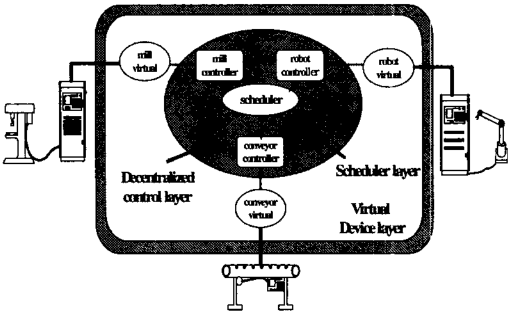

Fig. 3 Control structure of VFMC

The Scheduler is a core component that receives the states of all the machines in the VFMC from the Decentralized Control layer, and decides the appropriate next task. It then dispatches the next task to be executed to the Decentralized Control layer. It uses the process knowledge bases that contain the routine cell task rules that are generated from the TID. The Decentralized Control layer consists of virtual drivers for the virtual machine that mimic to physical machines. Their main role is to perform the harmonization and the cooperation between the cell components in order to carry out the task called for by the Scheduler layer. They provide a device independent interface to the actual cell components by translating the generic commands and error messages of the corresponding machine. The virtual driver in the layer communicator and pass messages with each other. A virtual driver send commands to the corresponding physical machine, and receives the state of that machine, through that Virtual Device in the Virtual Device layer.

The lowermost layer of the controller consists of the Virtual Devices which monitor and continuously mirror, in real time, the state of the physical machine they represent. Each machine state is analyzed by its Virtual Device and reported to the corresponding Virtual holons as required. The Virtual Devices also serve as conduits for commands from the Virtual holons to the physical machines.

5. Case experiment

For the simulation, as shown in Fig. 4 a shop floor consisting of four manufacturing cells is modeled with QUEST®, and each cell consists of several NC machines, a robot, a conveyor. During simulation, three different parts are to be manufactured and their lot size is decided as 500. Virtual flexible manufacturing cell is expected to greatly support assessing the manufacturability of a candidate design and to provide accurate estimates for processing times, cycle times and costs, as well as product quality. In order to support process design, the tool path specified in a NC program is verified through simulation with V-CNC®. In this case a solid model of the work piece and a model of the machine tool are used to simulate the machining process. The cutting tool follows the prescribed path removing material from the work piece. Thus, infeasible tool trajectories (such as those interfering with portions of the machine tool, fixtures or the work piece) are easily detected and corrected. In addition, the cutting parameters specified in the plan are evaluated whether these are appropriate or even feasible. For example, high depth of cuts may lead to machine tool chatter and thus damage the work piece, the cutting tool or even the machine tool itself. By using IGRIP®, a virtual robot is simulated to evaluate the performance of planned path and process performance with respect to the various production schedules.

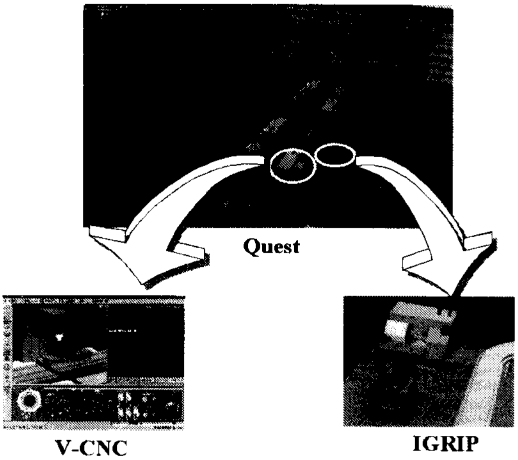

Fig. 4 Virtual FMC simulation environment

Fig. 5 shows a Task Initiation Diagram and corresponding cell behaviour description. For conciseness, states and tasks are cryptic instead of being verbose. Tasks in the task initiation diagram would consist of a concerted group of subtasks or operations involving more than one constituent of the cell, and in such a case is termed composite tasks. These are shown the framed boxes in the diagram. As an example **TE4** (unload part1 to Mill1) involves concerted operations involving the Machine and the Robot. Tasks involving a single machine are called simple tasks and are shown by a box, e.g., **TE2** (machine part). Further, Fig. 6 illustrate an Operation Initiation Diagram for the task unload . All states and operations are cryptic. The verbose descriptions for the symbols of the some states and operations are shown in Table 1 and 2.

In addition, the shop floor behavior is simulated with Quest, and thus the efficiency of all the machines can be reported. Therefore, various process methodology and logic can be applied and experimented without a real shop. This simulation becomes a part of strategic decision-making, allowing to receive information on production in the event of reconfiguration. It also gives the opportunity to view the whole shop floor and manufacturing cells while production is going on.

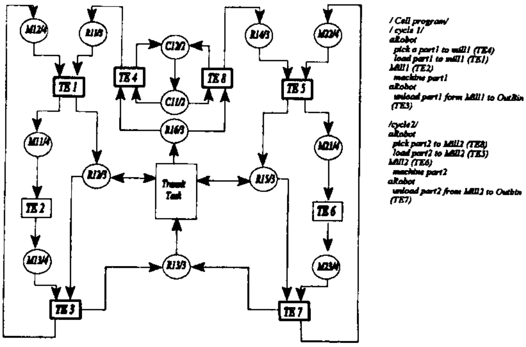

Fig. 5 Task Initiation Diagram for VFMC

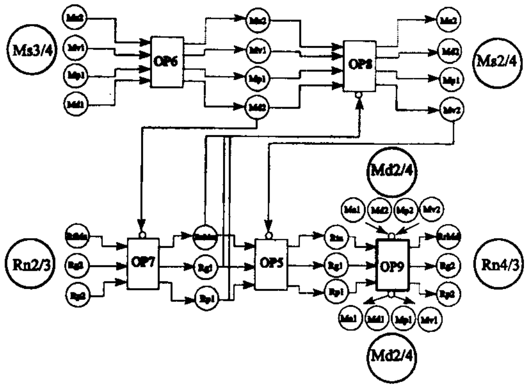

Fig. 6 Operation Initiation Diagram for Task unload

Table 1 Operation List

NAME	DESCRIPTION
OP5	Robot retract from machine
OP6	Machine open door
OP7	Robot approach to unload
OP8	Machine open vise
OP9	Robot deliver or load to machine

Table 2 Robot and Machine Composite States

NAME	DESCRIPTION
Mn2/4	Ready to load
Mn3/4	Ready to unload
Rn2/3	Ready to unload part
Rn4/3	Wait for delivered part

6. Conclusion

In this study, the concept of virtual manufacturing is investigated, and three models, such as the product, the facility, and the process model, are developed for virtual flexible manufacturing cells. A product model is a generic model used for representing all types of parts, which appear in the process of manufacturing. A facility model contains information about machines consisted of a virtual flexible manufacturing cell. A process model is used for representing all the physical processes that are required for representing product behavior and manufacturing processes.

The methodology behind developing VFMC is an object-oriented paradigm that provides a powerful representation and classification tools. For the implementation IGRIP/QUEST is used to model all 3D virtual machines involved models, and to simulate the whole factories where manufacturing events are concerned. The concrete behaviors of simulation are described by the task-oriented description (TID). Also the result of simulation is demonstrated to prove the applicability of the virtual manufacturing paradigm. The potential of virtual manufacturing is to support manufacturability assessments and provide accurate cost, lead-time, and quality estimate is a major motivation for further research and development in this area.

References

1. Iwata, Kazuaki Virtual Manufacturing System as Advanced Information Infrastructure for Integrating Manufacturing Resources and Activities, Annals of CIRP, Vol. 46, No. 1, pp. 399, 1997.
2. Kimura Fumihito "Product and Process Modeling as a Kernel for Virtual Manufacturing Environment," Annals of CIRP, Vol. 42, No. 1, pp. 147-151, 1993.
3. Bodner, D., Park, J., Reveliotis, A., and McGinnis, F., Integration of structural and perfromance-oriented control in flexible automated manufacturing , Proceedings of 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, USA, pp. 345-250, 1999.
4. Onosato, M., and Iwata, K., Development of a Virtual manufacturing System by Integrating Product Models and Factory Models, Annals of the CIRP, Vol. 42, No.1, pp. 475-478, 1993.