Applying Object-Oriented Systems & AI to Computer Integrated Manufacturing Systems

—목적지향시스템 및 인공지능을 응용한 컴퓨터 지원제조시스템에 관한 연구—

南 昊 基*

요 약

본연구는 컴퓨터 지원 제조시스템(CIM)의 특징 및 의사결정시 고려되어야 할 사항들이 서술되었다.
CIM의 복잡성을 표현하기 위해서 목적지향시스템 및 인공지능의 개념·구조가 상세히 설명되었다.
CIM하에서 독립적 시스템을 통합하는데 발생되는 통신정보를 표현하기 위해서 목적지향시스템 및 인공지능 기법이 이용되었다.

1. INTRODUCTION

Computer Integrated manufacturing (CIM) cannot be constructed just by integrating the automated devices available at present, such as CAD, CAM, CAPP, etc. systems. It is the systems approach of linking together the various automation tools available today, so as to enable the control of an entire manufacturing operation, as well as related business functions.

Until the current state of a munufacturing operation is known and can be reacted to at any given point in time, the virtues of expert planning and control systems can never be fully exploited. The result of every business decision must be measured on whether or not the decision has maximized the throughput of company to meet its demand and minimized the manufacturing cost. We find production personnel having to rely on "intuition" to make daily decisions. What is more extraordinary is that their "gut feel" or "intuition" often work.[3]

As the manufacturing environment becomes more automated, people will not be able to handle all the asynchronous events in an acceptable period of time. Therefore, there exists a need in the manufacturing environment to replicate this intuition. Intuition can be translated into computer terms as an expert system that can learn as everchanging environmen.

To realize successful CIM, the automation of friendly communication between experts is required, which plays an important role in integration of the various function of CIM systems. The Emerging technologies in both AI and Objected oriented system are responsible for achieving high performance transmission of information as in the friendly communication between the experts.

In this paper, we have introduced the formalized structure of concepts to represent the complexities of CIM systems. To express the information in CIM systems we make use of an powerful today's emerging techniques. (Object-oriented systems and Artificial Intelligence)

*인천대학교 산업공학과 조교수
접수 1990년 4월 25일

2. DECISION MAKING IN CIM SYSTEMS

Computer integrated manufacturing refers to the integrated use of computers in all sectors of a manufacturing environment : from production planning and design, through scheduling and quality control, to fabrication, manufacture and handling of finished products.

The characteristic of CIM systems [1] are

−hierarchical systems in structure and processes.

−systems with several conflicting objectives or strains expressing a network of relationships among system variables.

−system with adaptive and maintenance mechanism to answer to external variations and keep the obtained equilibrium.

Therefore, a CIM system can be viewed as a team of intelligent cooperating objects or entities arranged in a hierarchical structure composed of many different levels. The object are connected together through a tree like network which messages are sent and responses received.

The highest level is a facility level. Decision making in this level involves Master Production Schedule(MPS). The next one is the shop level. A shop is made up of a number of manufacturing cells connected through communication links. At this level, the system has to control the dispatch of material handling equipment and monitor cell conditions etc. Below the shop level is the manufacturing cell is a group of machine tools and associated material handling facilities that are managed by a supervisory computer called a cell-host. The cell-host takes care of several functions. It schedules the flow of jobs within a cell and instructs individual cells about tasks that each must perform. At the lowest level are the individual machine tools, for which individual operations must be scheduled.

The goal is to fully integrate all aspects of management either all essential elements of the process. What are important is minimizing the distance/barriers from the management function to the process being managed, reducing the layers of management by expanding the span of control, and minimizing data filtering/interpretation by indirect labor.

2.1 MAJOR ISSUES IN CIM SYSTEMS

A CIM system consists of a team of intelligent entities that are tied together and the following characteristics :

1) Multiple entities distributed over several distinct levels, and arranged in a hierarchical structure.
2) A communication system among the entities for sending and receiving messages for instructions, control and updating.
3) Autonomy and intelligence among the entities so that decision about a wide range of issues can be taken without intervention, either from humans or higher levels in the system.

We identify certain isssues that must be studied.

1) Number and type of objects, and the nature of inter-object communications.
2) The size of an information database allocated to each object and its relation to global database.
3) Degree of intelligence and autonomy permitted to each object.
4) Allowance for the representation of time-dependent concept and deductive decision making.

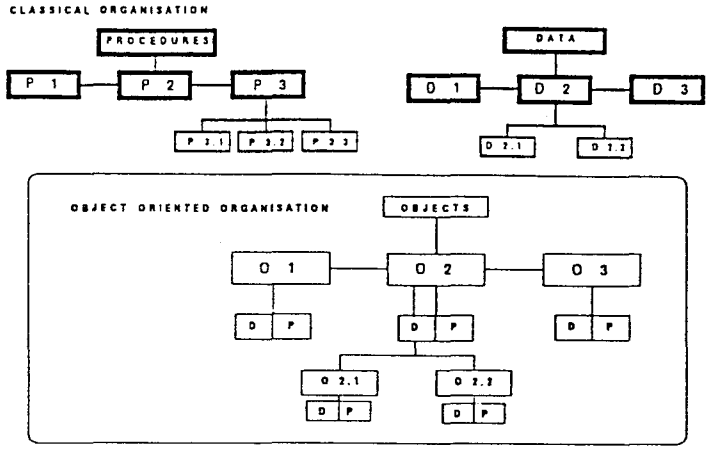
3. OBJECT—ORIENTED SYSTEMS

It may be helpful to briefly recall what are the characteristics of object-oriented systems and why these

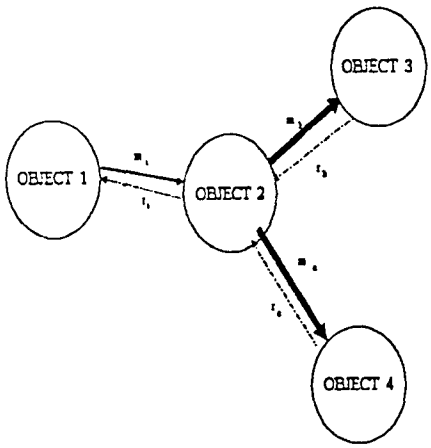
characteristic are important to us.

3.1 HOW THEY WORK

1) Grouping data and procedures : Standard programming techniques organize procedure and data separately. In contrast, object-oriented systems deals with objects. An object has local data stored in its private memory and local procedures that can manipulate the data.



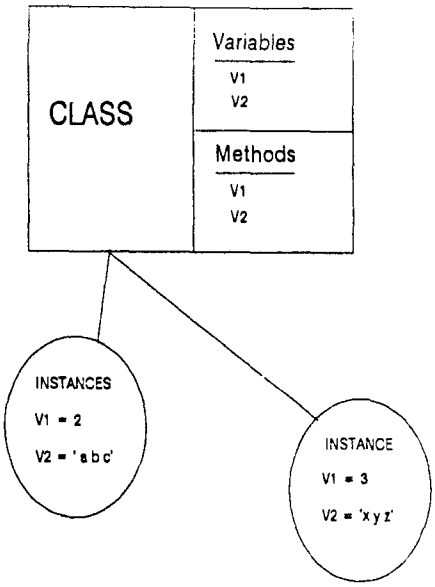
2) Messages : A local procedure of an object called "a method," is triggered in another object. The execution of the method may alter the local data and possibly send messages to other objects. When the method is completed, the control and value (which may be an object) is returned to the calling method which is then continued.



MESSAGES

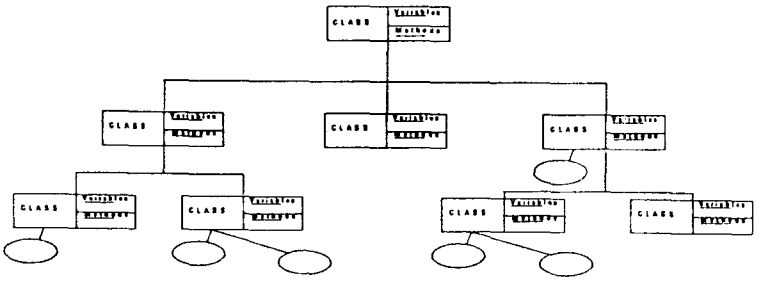
3) Classes and instances : Every object is an instance of a class. There may be many instances of a class. During the execution of program, some instances are created, other may be cancelled. Every instance

has the set of variables defined at class level but it has its own values. Every instance knows the same methods. The methods are defined at class level but are executed at instance level i. e. they use the variable of the instance.



Classes and Instances

4) Class hierarchy : The set of class is organized according to a relation "subclass" which induces a tree structure. A class inherits the properties of its parents, i. e. it has their instance variables and methods. However a class may redefined a method of an upper level. In this way, methods can defined for the most generic classes, the exception being treated for the specific classes that require a different handling.



Class Hierarchy

5) Abstraction : Abstraction is a method of tracking complex problems. The purpese is to develop a conceptual model of reality, emphasizing important properties of the entities at each level in the model, while suppressing the less important ones. The following are a few aspects of abstraction that can be useful in creating complex conceptual models :

—Classification : grouping entities that similar properties into a class. The inverse of classification is instantiation.

—Aggregation : treats a group of objects as a single object. An aggregate object can be decomposed into instances of the component objects.

—Generalization : is a form of abstraction where similar objects are related to a higher level generic object. The constituent objects are considered specialization of the generic object.

—Association : is a form of abstraction where a group of similar objects is considered as higher level set object. The details of the member objects are suppressed and the properties of the set object are emphasized.

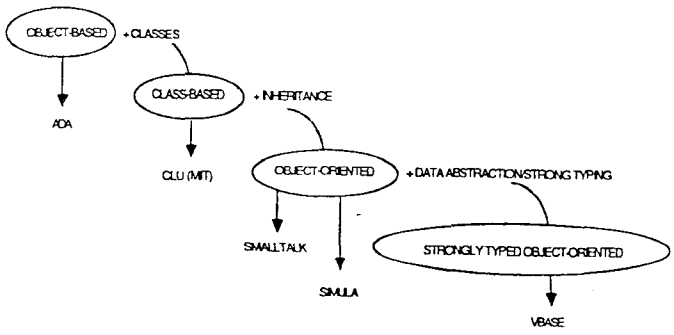
6) INHERITANCE : A class may inherit operations from superclasses and may have its operations inherited by subclasses. Inheritance from a singles superclass is called single inheritance. Inheritance from multiple superclasses is called multiple inheritance.

3.2 CLASS—BASED & CLASSLESS SYSTEMS

In Wegner's paper[9], he clearly structures elements (objects, classes, inheritance, data abstraction, strong typing, concurrence, persistence). Below figure represents Wegner's paradigm for a strongly type langauge.

Wegner's paradigm works nicely to control the access and assignment of values to the instance variables. In the simpler applications where strong modularity is not reguired, both the objects and the classes can be managed within the language, thus providing a uniform mechanism for both their design and their implementation. However in the more complex applications, stricter control must be adherent thus strengthening the object's modularity. This level of control is found in both the data abstraction and strong tying of the objects.

This class-based paradigm is very useful if the desired behavior is known and a class of objects can be defined. However, in a highly responsive interactive environement, it is often necessary to wait until a message is received to determine how behavior from individual objects needs to be used.



In Lieberman paper[4], Prototype systems allows creating individual concepts first, than generalizing them by saying what aspects of the concept are allowed to vary. In Leiberman's classless system, structure is not known but is determined during the process through the "re-collection" of similar prototype objects found in the system. During this "re-collection," each prototype object reacts according to the current state of environment. With this concept in mind there is an alternative mechanism for sharing knowledge. This alternative is called delegation.

3.3 BENEFITS

1) Natural approach : like the objects described above, humans have some private memory and behave, according to the specie they belong to, in response to messages received from others. Therefore our mental structures are accustomed to analyze systems composed of sets of such instances. As most modern sophisticated machine also have a private memory and a type dependent behavior, the object oriented modelling approach not only looks very natural it is also very close to the real world.[2]

2) Reuse, customization, extendibility : Although many problems contain a lot of similar elements, it is in general very difficult to adapt them to a slightly different environment or to extend them. The reason is that, as a result of separate organization of data and procedures used in classical languages, many details have to be modified at a lot of different places[5]. Object oriented languages are much better in that respect, the interface between objects is much stable and more general; the polymorphic features promote not only the definition but also the implementation and use of generic functions. Incremental programming and the possibility to write and test early parts of the programs at high abstraction level, even before defining the detailed formats at the lower levels, allow for fast prototyping.

3) Network hiding : By the mechanism of interaction through message, object-oriented systems are prepared for distributed implementation. During development and test, all the objects may reside in the same computer. But the objects never know whether message go far away to another computer via network or to a local subroutine. The whole object set may then be lately partitioned and distributed on a set of interconnected processors without upsetting the program structure.

3.4 OBJECT ORIENTED SYSTEMS AND CIM

Most of the benefit mentioned above are in particular valuable for CIM[8]. It is clear that the natural decision model is a set of human and automated actors interacting through codified messages. Parallel and distribution are almost always an ubiquitous prerequisite. Reuse, adaptability and extendibility are also of paramount importance. To preserve the continuity of operations, the system must evolve locally, i. e. parts of it must be upgraded to emerging technologies without perturbing the global coherence.

4. EMERGING TECHNOLOGY WITH AI

The fundamental tools required to support decision making may be represented in an object-oriented paradigm. Here we apply a few select elemens of artificial intelligence against this paradigm.

4.1 SYMMETRIC RULE PROCESSOR

In Marvin Minsky's Book[6] common sense reasoning differs from logical reasoning in that the process stops after every step to check whether the data found is as expected. If expectation are not met, common sense reasoning will switch directions and hypothesis. Common sense reasoning can be applied using both forward and backward chaining to resolve open issues. Neron Data[7] define this as symmetric rules processing where common sense behavior is an integration of both data-driven and hypothesis-driven motivations. The application of common sense reasoning to help identify the purpose and intent of a message is the key to the perceptual interpretation of a message received.

4.2 OPEN AT ARCHITECTURE

In building intelligent artificial systems for the CIM world, it is mandatory to think of those systems performing therir tasks in information-rich and rapidly evolving environments. We are concerned with the necessary interaction with the environment in terms of gathering information, detecting contradictions or

changes, quickly shifting the attention and resources, and providing the external world with answers or actions, in relation to a reasoning process.

System has been designed to allow easy integration into existing computing environments. The integration capabilites are :

1) Make system communicate with other process. (Event-driven architecture)
2) Customize systems's interface.
3) Extend the processing capabilities with optimized external routines or existing math libraries.
4) Link system to custom database.
5) Embed systems's reasoning capabilities into large scale application.

4.3 INTELLIGENT I/O INTERFACE MACHINE

1) Massively parallel, large-to-fine-grained multiple processor system typically capable of performing both numerical and symbolic processing.
2) Multi-tasking, single-user, microprocessor-based, two to thirty MIPS workstations capable of performing both numerical and symbolic processing.

4.4 Geographically Independent, Distributed Database Machine

Geographical independent network of linearly expandable, fault tolerant, parallel processing computers with a fully distributed operating system capable of performing distributed, fault tolerant SQL read/write routines.

5. CONCLUSION

In the informational processing world, the system engineers are the master mechanics. In the past few years they have been inundated with new tools in both artificial intelligence and object-oriented systems. After understanding these tools, the system engineer's real task is to state the requirements for a complex manufacturing environment and determine what tool or complementary set of tools can be applied to satisfy those requirements.

6. REFERENCE

1. Bartalanffy, L., "General Systems Theory-A Critical review" Systmes Behavior, Harper & Row 1972, pp. 29−49.
2. Booch, G., "Object Oriented Development," IEEE Transactions on the software Engineering. Vol. SE2, No. 2, Feb 1986, pp. 211−221.
3. Joyce, R., "Accepting Direct Control in Manufacturing," February, 1987 APICS Factory of the Future Conference.
4. Lieberman, H., "Using Prototypical Objects to Implement Shared Behavior in Object Oriented Systmes" OOPSLA 1986.
5. Meyer, B., "Reusability : The case for object oriented Design," IEEE Software, March 1987, pp. 50−64.
6. Minsky, M., "Society of the Mind," 1986, Simon & Schuster.
7. Neron Data Inc., Nexpert Object Fundamentals, 1988.
8. Tayor, W. A., "Object-Oriented Programming : a new approach for automatic factories," CIM REVIEW 1986, pp. 55−58.
9. Wegner, P., "Dimensions of Object-based Language Design," OOPSLA 1987.