Object-oriented representation of manufacturing systems: State of the art and perspectives

Anna Bartolotta
Politecnico di Milano
p.zza Leonardo da Vinci 32
20133 Milano, ITALY
Tel. +39-2-2399-2726 Fax: +39-2-2399-2700
E-mail: bartolot@mail.ecopro.polimi.it

Marco Garetti
Politecnico di Milano
p.zza Leonardo da Vinci 32
20133 Milano, ITALY
Tel. +39-2-2399-4760 Fax: +39-2-2399-2700
E-mail: garetti@mail.ecopro.polimi.it

Abstract
The object-oriented representation of manufacturing systems appears to be a promising way to encompass the limits of more traditional tools that have been used in the past for the modelling of manufacturing systems. The paper compares traditional modelling techniques to the approach of object-oriented modelling and reviews the state of the art of the object-oriented applications developed within the manufacturing context. As a conclusion, guidelines for the development of a comprehensive object-oriented representation of manufacturing systems are outlined.

Keywords
Manufacturing systems engineering, modelling of manufacturing systems, object-oriented modelling

Advances in Production Management Systems N. Okino, H. Tamura & S. Fujii (Eds.)

1 INTRODUCTION

Traditional approaches to manufacturing systems design are not well suited to deal with today's dynamic environment where system elements are continuously being subjected to replacement and rearrangement due to rapid changes in product design and product mix and fast advances in manufacturing technology.

Today no integrated design framework is available to support the design of manufacturing systems, a situation that stands in evident contrast to that of the design of modern manufactured products, where the computer-aided technologies (CAD/CAM) provide a software integrated environment where to carry out the design process. This is true in spite of wealth of published researches on tools supporting specific problems in manufacturing systems design.

This paper is part of a research project aiming to study a Manufacturing Systems Engineering Workbench (Garetti and Bartolotta, 1995), i.e. a software environment in which tools supporting the design of manufacturing systems could be integrated in a common framework.

A corner stone in such kind of approach could be provided by a structured representation of the domain of manufacturing systems, i.e. a unifying abstraction enabling the management of all relevant information and knowledge associated with the process of manufacturing systems design. Upon such a framework, a comprehensive and consistent manufacturing systems database could be constructed to integrate all the design tools to be included in the manufacturing system engineering workbench. This way the designer should deal only with one central, generalised model of manufacturing systems, while on the other hand all design tools should operate on this central model.

Once made the one-time effort to insert all data related to a specific manufacturing system, further advantages could be derived if the database were updated in such a way as to maintain the exact correspondence and accuracy of the data as the manufacturing system evolves, supporting the manufacturing engineer in the redesigning activities.

The objective of this paper is the identification of the guidelines that could be usefully employed in the ideation of such a representation environment for manufacturing systems. The objective is addressed by the analysis of existing modelling methodologies, in particular object-oriented methodologies, and applications of the object-oriented approach to the manufacturing context.

2 REVIEW OF CURRENT MODELLING TOOLS

The traditional methods of systems analysis can be classified as either 'functional decomposition' oriented or 'data' oriented. In the functional decomposition method, a system is functionally decomposed using functions and processes as elementary building blocks: therefore system entities are passive data stores, manipulated by activities and procedures (see for example the IDEF$_0$ approach). The data oriented methods, on the other hand, concentrate on system data structure, making extensive use of data flow diagrams and entity/relationship concepts.

In the recent few years a third approach, the object-oriented (O-O) approach, both for systems analysis and design, has been emerging. There is no doubt that the concepts of O-O approach are fundamentally different from traditional structured methods and require a different way of thinking (Booch, 1994).

As known, the fundamental construct of this approach is the *object*, which combines both data structure and behaviour in a single entity. The state of an object is captured in its *attributes*, while the behaviour is encapsulated through the *methods* (or *operations*). An object can communicate with other objects through *messages,* which constitute its public interface to the other objects.

An object is either a *class* or an *instance* of a class. The O-O approach embodies two important concepts:

- *encapsulation* is the separation of the external aspects of an object, which are accessible to other objects, from the internal details, which are hidden from other objects;
- *inheritance* allows the reuse of the structure and behaviour of a class in the definition of new classes.

It is now widely recognised that the object-oriented (O-O) paradigm provides an excellent approach to manage and express complex systems. The O-O approach presents characteristics that make it particularly suitable to meet the requirements for the representation of manufacturing systems: in fact, unlike classical (functional) design, the object-oriented paradigm decomposes a system dealing with the classes of objects the system manipulates, not the functions the system performs. Therefore the definition of objects remain independent of the system functions being modelled. Only a relatively small number of changes will be needed as a consequence of the introduction of a new object to an existing system, in contrast with the functional-based approach, where a large part or even the whole model may need to be altered to cater for such a change.

Furthermore, it seems that all manufacturing functions can benefit from the object-oriented approach, thanks to the strong affinity between the manufacturing context and object-orientation (Nof, 1994). In fact, for each kind of key objectives of a typical manufacturing enterprise there is a strong analogy in object-orientation; for example, both focus on objects (parts in manufacturing) and both are based on methods to define operations and services (processes and manufacturing services).

Finally, the concepts of encapsulation and inheritance provide the object-oriented approach with characteristics of flexibility, realistic view, extensibility and reusability, all characteristics necessary for the representation of manufacturing systems.

3 REVIEW OF OBJECT-ORIENTED METHODOLOGIES

Unfortunately the interest in the new object-oriented approach, has led to many methodologies, resulting in a proliferation of definitions, interpretations and non-

standardised concepts; this is particularly evident in the software engineering area, where the object-oriented approach finds a lot of applications.
The following paragraphs provide a literature review of the O-O approach. In particular, in § 3.1 three of the most known methodologies used in the area of software engineering will be briefly described, while in § 3.2 some of the few applications of O-O approach already emerged into the manufacturing context will be critically reviewed.

3.1 Object-oriented methodologies

OOSA (Shlaer and Mellor, 1988) divides systems development into OO analysis and design.
OO analysis is described in three steps:

- *information modelling*: the focus is on abstracting the conceptual entities in the problem domain in terms of objects and attributes. The associations that exist between the entities are formalised as relationships that are based on the policies, rules and physical laws that prevail in the real world;
- *state modelling*: this step concerns the behaviour of objects and relationships over time. State models are used to formalise the life cycles of both objects and relationships. The state models, which consist of state transition diagrams and tables, communicate with each other by means of events. State models are defined by multi-layers of state transition diagrams to make the model of communication orderly and understandable.
- *process modelling*: the actions of the state models, which contain all required processing, are dissected into fundamental and reusable processes and are expressed by an enhanced form of the traditional data flow diagram. The processes so derived can be converted directly into operations of object-oriented design.

Four distinct diagrams are used in the design phase:

- *class diagram*, which shows the external view of a single class;
- *inheritance diagram*, which shows the inheritance relationships between classes;
- *dependency class*, which depicts the client-server (invocation) and friend relationships that hold between classes;
- *class structure chart*, showing the internal structure of the code of the operations of the class.

Another O-O methodology is due to Booch (Booch, 1994): while designing a complex system, he views the system itself considering multiple *perspectives*: namely, the logical and physical structure, and the static and dynamic semantics. Both dimensions are necessary to specify the structure and behaviour of an object-oriented system. For each dimension, Booch defines a number of diagrams, denoting a view of a system model:

- *class diagram*, used to show the existence of classes and their relationships in the logical design of a system; a single class diagram represents a view of the class structure of a system;
- *object diagram*, used to show the existence of objects and their relationship in the logical design of a system; a single object diagram is typically used to represent a scenario;
- *module diagram*, used to show the allocation of classes and objects to modules in the physical design of a system: a single module diagram represents a view of the module architecture of a system;
- *process diagram*, used to show the allocation of processes to processors in the physical design of a system: a single process diagram represent a view of the process architecture of a system;
- *state transition diagram*, used to show the state space of an instance of a given class, the events that cause a transition from one state to another, and the actions that result from a change of state;
- *interaction diagram*, used to trace the execution of a scenario in the same context as an object diagram.

The Object Modelling Technique (OMT) (Rumbaugh, 1991) distinguishes among three kind of models:

- *object model*, describing the static structure of objects in a system and their relationships. The object model consists of *object diagrams*, a certain kind of ER models, which describe the association and the relationships among objects;
- *dynamic model*, describing the interactions among objects in the system and hence the control aspects of a system. This dynamic part for every class is described in *state diagrams* and to identify events, states and transition first *event flow diagrams* are drawn;
- *functional model*, describing the data transformation of a system. This is done with *data flow diagrams*.

The three kinds of models separate a system into orthogonal views that can be represented and manipulated within a uniform notation. The three different models are cross-linked (a complete description of a system requires all three models) but each model can be examined and understood by itself to a large extent.

None of the existing O-O methodologies has really achieved the status of being a widely recognised standard comparable to some of the conventional methodologies. Furthermore, the proliferation of methodologies used to refer to essentially similar and basic functional capabilities leads to be suspicious about the real necessity of adopting a methodology in order to take advantage of the object-oriented approach. In addition, when these methodologies (specific to the area of software engineering) are considered for use within the context of manufacturing systems analysis, it becomes clear that there are certain features that need to be catered for (Wu, 1995).

3.2 Object-oriented methodologies

In literature, we can found some attempts to apply the O-O approach within the manufacturing context. A literature review shows that these attempts don't follow necessarily a precise existing methodology (with its notation and graphical diagrams), but they use only the basic principles of the object-oriented orientation. This leads to inherent difficulties in formally evaluating these candidate methodologies on a common basis, because of major differences in their underlying philosophies and their consideration of only some aspects. In the following paragraph, a few examples of these attempts will be described and analysed.

Typically the object-oriented approach to describe manufacturing systems has been mainly used to build simulation models. The objective is the development of software modules that can be combined by the designer to build a simulation model.

Shewchuck and Chang (1991) present a hierarchical structure of object classes, consisting of three class libraries. These classes can be classified into two broad categories. The first category, to which the *base classes* and the *simulation support classes* belong, contains objects providing the software functions which allow the background simulation processing tasks, such as time advance, event triggering, entity creation, list processing, etc. to be performed. The second category, the *manufacturing systems simulation (MSS) classes,* includes objects providing the reusable building blocks for modelling manufacturing systems.

Among the MSS classes, Shewchuck and Chang identify three main classes:

- *part object classes,* used for modelling parts in the system, can be distinguished in three classes: *partType*, *opData* and *part*;
- *resource object classes,* used for modelling resources used in manufacturing system, can be distinguished in three classes: *resourceClass*, *resourceType* and *resource*;
- a *production system class,* used for modelling information which are not present in other classes, such as the shop calendar, production shift length and data related to the method by which parts are introduced into the system.

Shewchuck and Chang's work is clearly a research paper aiming at investigating the potentiality of the object-oriented approach when applied to the simulation of manufacturing systems: they concentrate only on the most significant classes (such as machines and parts), ignoring many others (e.g. transport systems). Notwithstanding this, we can find some suggestions in their effort of modelling this kind of class. In fact their approach seems to gain maximum modularity and consequently flexibility.

For example, because part process routing data requirements can vary greatly from one application to another, routing data is not included in part object; instead, separate routing data objects are used, and part objects contain pointers to these objects. Through this linkage, part objects know their routing data, while the physical class separation of routing data from part type data facilitates the development of different forms of routing data classes for various applications. The use of a separate operation data object for each part type operation allows great flexibility in describing part process routings.

Although simulation is the most frequent application field, it's possible to find, especially in recent papers, attempts to generalise this kind of application.

An on-going research project at Oklahoma State University's Centre for Computer Integrated Manufacturing (Mize, *et al.*, 1992; Pratt, *et al.*, 1991), which aims 'to develop an object-oriented modelling environment for manufacturing systems', has been exploring alternative approaches to the modelling and simulation of complex system. The prototype modelling environment under study should emphasise reusability and could be used for many specific approaches to systems analysis (e.g. simulation and queuing). Reusability is achieved through the provision of separate modelling constructs for physical, information and control decision elements of a system. In the authors' opinion, designing for reusability involves the identification of behaviours that are useful in more than one context. In general this implies a system design which adheres rather strictly to the "one component-one function" doctrine. According to the authors' opinion, the demonstration of the feasibility of creating logical decision modularity among the decision elements of a modelled system is an interesting conclusion derived during the prototype environment development. This is equivalent to "plug compatible decision elements", which can be replaced as desired in any location in the model.

Another advantage of the separation of physical, information and control objects is that it allows the system modeller to think to these elements independently during the model development. This provides a more natural modelling environment. In other words he/she needs not to be concerned with information and control aspects when developing the physical model. The process involving the selection of the appropriate physical components without being constrained by concerns regarding how to model information flows.

The papers describing this research project specify neither the object hierarchy nor the process followed to identify them, but they offer several useful remarks from which to start for the development of the representation environment.

Wu (1995) suggests an object-oriented systems analysis method, called hierarchical and object-oriented manufacturing systems analysis (HOOMA). This method exploits the features of two general methods of object-oriented analysis, OOA (Object-oriented analysis) by Coad and Yourdon and HOOD (Hierarchical O-O design) by Graham, and is developed to support specifically the requirements of systems analysis and definition within the manufacturing context.

The analysis procedure of HOOMA can be summarised as follows:

- *system decomposition*: HOOMA approaches this task by using 'functional subjects' (FS), whose hierarchy is identified through function block diagrams (FBDs);

- *identification of classes and objects*: the use of sub-system relationship diagram (SRD), which shows how data flow around a system FS, can provide a useful aid in tackling this phase;
- *specification of static structures*: this step requires describing the manufacturing system in terms of the structures of its class and objects (generalization-specialization and whole-part relationships), using graphical presentation of these structures similar to that used by OOA;
- *specification of dynamic interaction*: it consists in the specification of the object life cycle using state transition charts (STCs) and modelling of objects' interaction through activity cycle diagrams (ACDs);
- *sub-system integration*: the sub-systems are integrated into one whole graphic model to convey the necessary information about the problem domain.

In author's opinion, the method succeeds in bridging the division between the function-based approach and the pure O-O approach. In addition, HOOMA allows temporal logic to be encapsulated in the system objects, and hence dynamic processes can be represented through object interaction.

Wu (1995), unlike the other authors, recognises the need to follow a procedure, a sequence of steps in order to build a complete model of manufacturing systems, and proposes a method similar to those used in the area of software engineering. The procedure he suggests is particularly suited for manufacturing systems modelling. In our opinion, the use of functional subjects and function block diagrams in the first phases of the system analysis can address to an inconvenient decomposition of the domain (more functional than object-oriented) with the consequence of having a representation not very flexible to changes. The use of state transition charts and activity cycle diagrams for outlining the interactions among objects appears to be a good solution, just because both tools have already been applied successfully to the representation of dynamic interactions in manufacturing systems.

4 GUIDELINES FOR A COMPREHENSIVE O-O REPRESENTATION OF MANUFACTURING SYSTEMS

The suggestions, highlighted in the precedent sections, can be summarised in a framework so to provide useful guidelines for the development of what is needed for an effective use of the O-O approach to manufacturing, i.e. to create a comprehensive O-O representation environment for manufacturing systems.

- **Maximum modularity.** In order to obtain maximum flexibility, it's convenient to modularise, as much as possible, manufacturing information. According to Shewchuck and Chang's approach, technological data (such as process plans), for example, shouldn't be included in part object, but they should be considered as objects apart; in the same way, operations included in a process plan should be considered as object apart, in order to use the same operation object for different process plans.

- **One component-one function doctrine.** In Mize's approach three classes of objects have been identified, physical, information and control. The one component-one function doctrine suggests to separate these kinds of objects. In such a way, the components become building blocks from which a total system model could be constructed. In fact, if an object performs more than one of the three basic functions, its use becomes limited to situations in which all of its functions are required.
- **Hierarchical structure**. The O-O approach well accomplishes the representation of manufacturing systems related information organised in a hierarchical manner. For example, the decision making process can be modelled through a hierarchy of control levels: shop control level, cell control level, workstation control level and server control level. A hierarchy of objects corresponding to the controllers of each level can be easily constructed, exploiting the inheritance property, which allows forming low-level classes (subclasses) starting from a high-level general class.
- **Modelling tools employment.** As described, HOOMA procedure consists in a sequence of steps, using several tools and mechanisms. Following this concept, a clear procedure composed by steps, more or less supportable by modelling tools, like activity cycle diagrams or state transition charts, could be an essential guide for the complete explicitation of each aspect of a manufacturing system.

Useful suggestions can also be derived from the area of software engineering. In particular, among the different methods described in section 3.1, it's worthwhile noting that the Object Modelling Technique by Rumbaugh (1991) shows important analogies with Mize's objects classes, as highlighted in Table 1.

Table 1: Overview of the OMT method and analogies with Mize's object classes

OMT	OBJECT MODEL	static structure of the objects and their relationships	*PHYSICAL OBJECTS*
	FUNCTIONAL MODEL	data value transformations within a system	*INFORMATION OBJECTS*
	DYNAMIC MODEL	aspects of a system changing over time (control aspects)	*CONTROL OBJECTS*

For these reasons, the OMT method seems to be better suited than other methodologies (described in section 3) to be applied to the manufacturing domain.

5 CONCLUSIONS

Tools traditionally used for the modelling of manufacturing systems have shown important drawbacks, such as the limited usefulness of the method, the incomplete description of the manufacturing system properties, the limited flexibility of the model. Despite the interest and the results already obtained in modelling the manufacturing systems thorough the use of the object-oriented approach, a comprehensive and fully satisfactory methodology has not been reached in this concern yet.

Further research is therefore needed to fully exploit the potential of O-O approach in the manufacturing field.

6 REFERENCES

Booch, G. (1994) Object-oriented analysis and design - with applications. Benjamin Cummings, California.

Garetti, M. and Bartolotta, A. (1995) General concepts of a Manufacturing Systems Engineering workbench as a tool for the re-engineering of manufacturing systems, in *Re-engineering the Enterprise*, (eds. J. Browne and D. O'Sullivan), Chapman & Hall, London.

Mize, J.H., Bhuskute, H.C., Pratt, D.B. and Kamath, M. (1992) Modeling of integrated manufacturing systems using an object-oriented approach. *IIE Transactions*, 24(3), 14-26.

Nof, S. Y. (1994) Critiquing the potential of object orientation in manufacturing. *International Journal of Computer Integrated Manufacturing.* 7(1), 3-16.

Pratt, D.B., Farrington P.A., Basnet, C.B., Bhuskute, H.C., Kamath, M., and Mize, J.H. (1991) A framework for highly reusable simulation modelling: separating physical, information and control elements. *Proceedings of the 24th Annual Simulation Symposium*, IEEE Computer Society Press, 245-261.

Rumbaugh, J., Blaha, M., Premerlani, W., Frederick, E., and Lorensen, W. (1991) Object-oriented modelling and design. Prentice-Hall, New-Jersey.

Shewchuk, J.P., and Chang, T.-C. (1991) An approach to object-oriented discrete-event simulation of manufacturing systems. *Proceedings of the 1991 Winter Simulation Conference*, Phoenix, AZ, 302-311.

Shlaer, S. and Mellor S.J. (1988) Object-oriented analysis: modelling the world in data. Prentice Hall, New Jersey.

Wu, B. (1995) Object-oriented systems analysis and definition of manufacturing operations, *International Journal of Production Research*, 33(4), 955-974.

7 BIOGRAPHY

Anna Bartolotta is a Ph.D. student of the Dipartimento di Economia e Produzione of the Politecnico di Milano. She received a M.Sc. degree in Management and Industrial Engineering in 1994 at the Politecnico di Milano, presenting a degree thesis entitled “Framework for a Manufacturing System Engineering Workbench”. Her research interests include advanced tools and techniques for the design and management of manufacturing systems.

Marco Garetti is Professor of Industrial Technology at the Dipartimento di Economia e Produzione of the Politecnico di Milano, Italy. From 1971 to 1974, after graduating in Mechanical Engineering at Politecnico, he worked at Alfa Romeo first on the mechanical modelling of a car passenger and on vehicle-body computer aided design after. In 1974 he joined Politecnico di Milano as a researcher, working in the field of industrial production, automated production systems and CIM. He is currently heading an Italian government-supported research program on integrated production systems. Prof. Garetti is a senior member of the Computer and Automated Systems Association of the Society of Mechanical Engineering (CASA/SME), a member of IFIP W.G. 5.7, a member of the editorial board of PIXEL (an Italian review on CAD-CAM) and of the International Journal of Production Planning and Control.

This work has been developed within the PLANTFABER project, No. 22031, titled "An integrated software workbench as a tool for the re-engineering of the Manufacturing Systems Engineering process", in the frame of the ESPRIT IV, Technologies for Business Processes programme.