Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

OBJECT ORIENTED SIMULATION TOOLS NECESSARY FOR A FLEXIBLE BATCH PROCESS MANAGEMENT ARCHITECTURE

Chell A. Roberts
Terrence G. Beaumariage
Yasser Dessouky

Systems Simulation Laboratory
Industrial and Management Systems Engineering
Arizona State University
Tempe, Arizona 85287-5906

Michael K. Ogle

Department of Mechanical and Industrial Engineering
Louisiana Tech University
Ruston, Louisiana 71272

ABSTRACT

An increasing demand in the process industries for the production of relatively low quantity, specialized chemical compounds has escalated the use of batch processing techniques. The same principles that are being used to drive flexible manufacturing are needed to give flexibility to the process industry. A software based virtual plant that is operated on by a suite of expert system assistants is a potential solution to the flexibility needs. This paper presents an architecture for batch process management with particular emphasis on the simulation tools contained in the environment. Within the architecture are a suite of high level tools that interface with a user to obtain a high level of batch processing flexibility. These high level tools are being developed in a modular fashion at Arizona State University in conjunction with the CIM Systems Research Center and the Systems Simulation Laboratory.

Two simulation assistants have been proposed for a complete batch process management architecture. These include a batch sequencing simulation assistant and a control specification simulation assistant. These assistants also have the intended capability of using the plant specifications and modeling knowledge to construct simulation models for the purposes previously mentioned. It is felt that these tools will significantly increase the batch process management life-cycle time and increase process plant flexibility.

1 INTRODUCTION

An increasing demand in the process industries for the production of relatively low quantity, specialized chemical compounds has escalated the use of batch processing techniques. Similar to occurrences in the discrete part manufacturing industries, this market change requires increased flexibility by continuous process manufacturers.

Simultaneously, the software costs for design, development and maintenance of processes for these fine and specialty chemicals is increasing. Clearly a competitive advantage can be gained through advances in batch process automation, planning and control.

Traditional planning and integration tools for the continuous process industries have not been developed with batch processing in mind (Musier 1990). As a result, these tools have been unable to respond to the demand for increased production flexibility. Efficient batch processing requires intelligent tools to assist in the entire batch process life-cycle within an architecture that ties these tools together.

Recently, there have been many advances in software engineering and in system design tools and techniques that have not been exploited in the process industries (McCarthy 1990, Suydam 1987). These advances have the potential of solving many of the present problems. Among these tools are high level simulation tools (Baudel et. al 1988) that can be combined with expert systems, knowledge based tools, and object oriented techniques to form an integrated, high level software solution to the flexibility issue.

This paper presents an architecture for batch process management with particular emphasis on the simulation tools contained in the environment. Within the architecture are a suite of high level tools that interface with a user to obtain a high level of batch processing flexibility. These high level tools are being developed in a modular fashion at Arizona State University in conjunction with the CIM Systems Research Center and the Systems Simulation Laboratory. An approach that incorporates several intelligent assistants similar to the ideas used in the development of the other distributed

expert systems (Waters 1985), or focused expert systems, that operate on a variety of knowledge bases is being used. The system is codified in an object oriented environment to maintain reusability and flexibility.

The conceptual design as currently developed, illustrated in Figure 1, includes several intelligent assistant components: 1) a physical system specification assistant that aids in constructing a virtual plant, 2) a process plan assistant that aids in specifying the batch process, 3) a batch process manager assistant that aids in sequencing, equipment scheduling, and materials balancing, 4) a control specification assistant that creates a generic control specification, and 5) a control code generator that produces control code for the selected equipment. Two simulation assistants, the batch sequencing simulation assistant and the control specification simulation assistant, test, validate, and optimize the batch process and the control code. The first five of these assistants are described in the following section, with the simulation assistants described in a later section.

2 A GLOBAL ARCHITECTURE FOR BATCH PROCESS MANAGEMENT

As mentioned previously the batch process management environment contains several intelligent assistants. **The physical system specification assistant** interacts with a user to create the physical component of a virtual plant. The virtual plant is an object oriented software map of the actual plant. The physical characteristics, such as equipment type, equipment capacity, functional capability, size, connections, sensors, and other plant attributes are codified in a modular reconfigurable fashion. Reconfigurability is important because the physical plant is dynamic. Equipment may be periodically added and deleted. The physical system assistant has tools that aid the user in the construction of the virtual plant, such as a library of equipment types, tools for connecting the equipment, tools for adding sensors, a graphics editor, and tools for checking the validity of the user configuration. Some of these tools use a physical system knowledge base to accomplish their tasks. The physical specification knowledge base also aids in plant layout and design functions.

The process specification assistant aids a user in the development of a process plan, or a set of process plans. As orders and inquiries are received from customers, a process engineer checks the process plan data base to see if a process plan already exists for this product. If not, a new process plan is generated. The process plans contain an ordered set of sub-processes that could be used to produce a particular chemical batch. There are frequently many different process plans for producing the same product. This intelligent assistant uses a process specification knowledge base to generate these process plans. This knowledge base contains rules about chemical reactions as well as process plan syntax knowledge. Together, the intelligent assistant and the process knowledge base are used by the chemical engineer to develop and maintain the processes for new and specialty chemicals.

The virtual plant and the process plans must be sufficiently robust such that other intelligent assistants can use these specifications for aiding the user in making decisions. One assistant that utilizes these specifications is the batch process manager assistant.

One role of **the batch process manager assistant** is to match processes with the physical virtual plant. This matching process will produce groupings of actual equipment that can be used to produce the required chemical batch. This is called equipment selection. There may be several different groupings based on each possible process plan for the same product. There also might be no groupings in which case the process knowledge base might be invoked to suggest possible equipment additions or modifications that could be used to produce the product.

The use of each grouping of equipment carries with it certain penalties or opportunity costs. Some subset of the group of equipment might be already scheduled for other orders. By changing the schedule there is a cost associated with moving the other order to a different set of equipment or a different time. Also, the selection of one process plan over another might necessitate a different mix and quantity of inputs and by-products which in turn changes the cost associated with the equipment and process plan selection.

The batch process manager assistant uses the data base of current customer orders and a knowledge base that contains rules about customer priority, equipment cleaning, process preference, equipment history, batch sequencing rules, cost history, cost estimates, and other knowledge to aid the user in the selection of the appropriate grouping. Sometimes equipment schedules for existing orders might be changed to accommodate the new request. Once selected, this grouping can then be scheduled for that particular batch. The selection of the actual equipment also results in information that will be used for material balancing, or chemical inventory control. Another benefit of the batch process manager assistant is in its use as a cost and time estimating tool for customer inquiries.

Control code must be developed for each batch based on the sequence, the process plan, and the selected equipment. One major impediment to batch processing is the amount of time that it takes to develop and validate the control code. Most processing plants are not

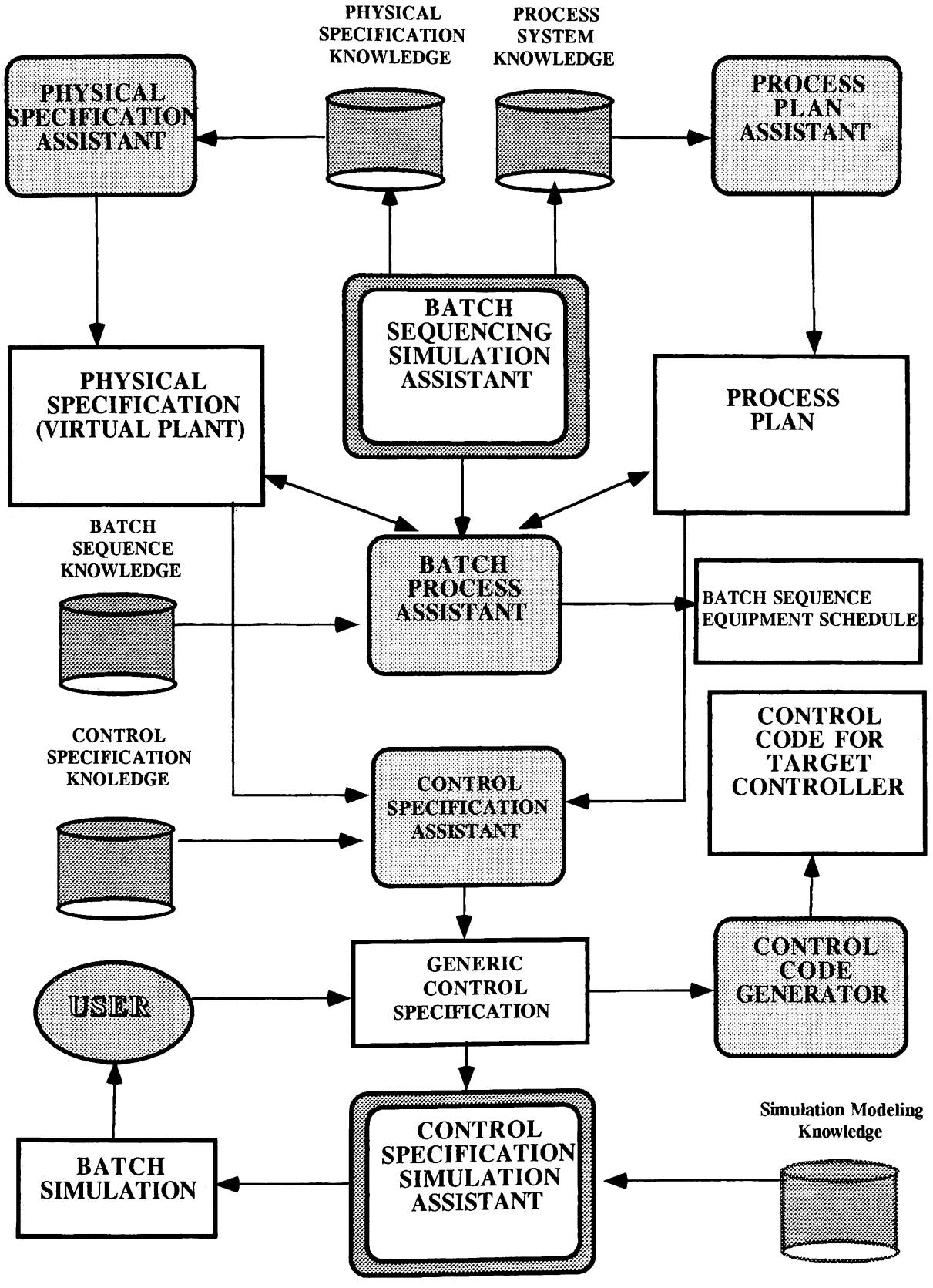

FIGURE 1. HIGH LEVEL SIMULATION TOOLS ARE USED WITHIN THE BATCH MANAGMENT ARCHITECTURE TO MAKE DECISIONS, ADD KNOWLEDGE, AND VALIDATE SYSTEM SPECIFICATIONS

configured for rapid product change. In some instances changeovers may take several months. **The control code generation assistant** is used to reduce change-over time. The physical and process specification can be used to create a control specification for testing and in turn the actual control code. This is because the physical virtual factory contains detailed information about the equipment, including all of the connected sensors and actuators. The control code generation assistant uses a knowledge base to aid in design and development of the control system. Although the equipment might exist for a particular process plan, there is no guarantee that the sensors and control computers are in place to supervise the batch process.

The user interfaces with the process plan, the virtual plant, and the control specification assistant to create a control specification for the system actuators and sensors independent of the actual computer controller. The specification is a high level control abstraction sufficiency descriptive to undergo validation. Generic real-time control specification has been discussed by Hatley (1987) and is currently being augmented and tested by the authors. The generic control specification is then mapped into a target language for target control hardware. A knowledge base consisting of rules that map the generic specification into a machine specific target is used by the control code generation assistant. The independent nature of the specification and the actual control code make it feasible to replace an entire computer control system without changing the control specification. It is still possible, however, that the control specification contains logic errors.

As depicted in Figure 1, there are several important simulation tools that are part of the architecture. Two simulation assistants will be introduced following a discussion on the objected oriented architecture underlying this project.

3 THE OBJECT ORIENTED ARCHITECTURE

One of the most significant aspects of this project is the manner in which the concepts and implementation are being developed. Having experience in the area, the individuals involved in the project decided that an object oriented approach would exhibit the most flexible and natural mechanism for the creation of the management environment. In order to explain the reasoning behind this, first consider the task at hand. The desired computer based system consists of several relatively autonomous subsystems which share a number of knowledge sources. These knowledge sources must be able to maintain information in several forms, including rules, attributes, etc., while being able to efficiently represent the complex interrelationships between the physical and conceptual entities.

The object oriented structure provides direct benefits in several ways. First, the basic tenant of OOP is the existence of an encapsulated data and procedures software element. As Shannon (1987) states: ". . . the underlying notion of the object is to organize and store pieces of information relating to a single concept into a single location". Of course, in addition to its data, an object also has its functional capabilities. An object provides specific functions and a structured interface to the external software environment while controlling access to and use of internal information. By structuring the research environment in terms of objects, a software system which is understandable, modifiable, reusable, and flexible results. The four key features of object oriented programming, encapsulation, message passing, late binding, and inheritance (Wilson 1987), provide the basic characteristics needed to achieve these software goals. Each of the previously mentioned semi-autonomous subsystems will consist primarily of hierarchically related objects with links between subsystems where feedback on current analysis activities are necessary. As it becomes necessary to revise separate subsystems and their components, the remainder or the batch management environment maintains its integrity.

Another benefit of an object orientation is that physical system components, ie. equipment, materials, etc., conceptual system components, ie. process plans, equipment schedules, knowledge bases, etc., and associated characteristics can be easily and naturally represented through the creation of objects emulating their structure and function. One positive attribute of this approach is the fact that physical and conceptual system objects can be added to and removed from the virtual plant as necessary with minimal impact on the other knowledge in the system. Another positive attribute is the level of correspondence which the software components of the management environment have with the components of the real world system. This benefits the environment developers in recognizing what the system must contain and the environment users in understanding how to use the system within their production environment.

As a development environment, the researchers have chosen Smalltalk-80. Reasons for this choice include, complete development environment, availability on several computing platforms (RS 6000, 386 PC, Macintosh, and DEC Vax Stations), familiarity with the paradigm and language, availability of expert system shell tools, and the availability of a kernel object oriented simulation capability. Baudel et. al. (1988) have discussed the use of Smalltalk for Simulation of Batch Processes.

4 THE BATCH SEQUENCING SIMULATION ASSISTANT

The batch sequencing simulation assistant has three primary responsibilities. First, it aids in the decision making procedure providing approximations for incomplete knowledge. Second, it accelerates the knowledge acquisition process. Third, it provides a mechanism for validation of the physical specification and batch process tools.

Often knowledge based tools are under-utilized because they are incomplete and the deductive reasoning capabilities are deficient. A knowledge based chaining process may fail to reach resolution. Also there are many operating characteristics in a process plant that are stochastic. Determination of some of these characteristics is vital to achieving optimal or near optimal decisions and plant efficiencies. There are potential cascading effects of each decision that might change other system components and operational policies. Some of these important characteristics include:

* The average time a batch takes to produce.
* The average inventory levels.
* The process bottlenecks.
* The impact of breakdowns on the system.
* The impact of a scheduling strategy on the system.
* The impact of adding or deleting equipment to the system.
* The impact of order cancellations on the system.
* The impact of irregularities in vendor supplies on the system.
* The cost of processing based on changing system attributes.

The acquisition of sequencing and scheduling knowledge, like most knowledge acquisition, is an ongoing process. Frequently, knowledge is gathered from the people who are the system experts. With a dynamic system, new events can occur with which even the experts don't have experience. Likewise, there may exist even better rules than those generated by the experts. Simulation is a mechanism for generating new rules for conditions that have not occurred, but could, and for improving on the existing rule base.

The expansion of the knowledge base is connected to the expert system validation process. Meta knowledge can be used to modify automatically a rule base but at some point it is generally imperative that the human participate in the validation. This validation is both a numerical as well as a graphical simulation of the virtual plant. A time based graphical simulation is conducted using the physical virtual plant, the process plans, and the batch process manager output. One advantage of the graphical simulation is its ability to communicate or be presented to a large audience and thus eliminate the need for numerical decoding. These capabilities are increased by the modular object oriented approach that has been taken.

5 CONTROL SPECIFICATION SIMULATION ASSISTANT

Often control code errors are not detected until the code is tested on the actual process. On-line validation can be extremely costly since process equipment must be out of production during the testing process. An alternative solution is to use computer simulation to reduce the validation time.

Computer simulation for control code validation can also be very costly to perform. A simulation model must be created which can interact with the actual control specification. Inputs to the system must be analyzed and distributions formed. It is also desirable to have some inputs that the operator controls. Controls such as push buttons, levers, dials in the actual system need to be represented and available for input to the simulation. The operator of the simulation model must also be able to provide these inputs in a pseudo real-time mode. Fortunately, part of the simulation specification is already available.

The physical specification of the virtual plant and the process specification already contain information that is needed for the simulation model. As has been mentioned, each piece of equipment has an object oriented software map called an instance. The object oriented approach allows us to add functionality to the object for simulation purposes. For example, we may have an instance of a mixing tank. This mixing tank instance has a data structure for current level and for its own iconic representation. The encapsulated procedure for the mixing tank object will include the ability to display itself and the ability to display its current level.

The control simulation is conducted using a next event calendar and the keyboard for system inputs. Many inputs are modeled with distributional functions, however, many of the inputs to the model come from a manual input source. To achieve this the inputs are tied to the keyboard with the aid of the control specification simulation assistant. A pseudo real-time simulation is maintained using fixed time increment updates generated by the computer clock. This update scheme gives the user the ability to interact dynamically by evoking system events. With each time increment the calendar is also checked for scheduled random events.

As new input events occur, the simulation assistant processes the changes in system states and maps these

states to the generic control code specification produced by the control code generator assistant. System states are updated based on the specification by sending messages to the equipment instances. The equipment instances receive these messages and update their own internal states which are then displayed on the monitor.

6 IMPLEMENTATION/DEVELOPMENT ISSUES

The backbone of the proposed batch process management system is a virtual representation of the process plant. A robust virtual plant would include a software representation of the physical and behavioral plant. There are several primary obstacles to the creation of such a robust virtual plant. They include:

1. Adoption of a total CIM approach to plant improvement.
2. Identification of an incremental approach to software codification.
3. Physical characterization of the plant.
4. Behavioral characterization of the behavioral plant.
5. Integration of the existing plant data base to the virtual plant.
6. Maintenance of the virtual plant.

Adoption of a total CIM approach to plant improvement is imperative to the implementation of a virtual plant. Software development is not a solution for management practices, employee moral, corporate culture, inadequate quality control, inadequate plant architectures, and the majority of other common problems faced in industry. A virtual factory provides a means for examining and controlling current operations. The virtual plant provides a comprehensive platform with an underlying data environment linked to the plant. The inherent structure of this type of environment lends itself to the manipulation of the data through simulation for exploring alternative strategies. The existence of the structured data would reduce significantly the level of effort necessary to construct simulation models from enterprize level models all the way down to hardware control models.

Identification of an incremental approach to software codification is also an important issue in the implementation of these tools. A large existing process plant may consist of thousands of physical devices and many layers of formal and informal control. Likewise, the plant is dynamic. An implementation approach must be selected that provides a gradual codification of the plant. The software structures should be reusable and modifiable which would suggest an object oriented environment. One perceived limitation is the execution speed of object oriented languages. However, "the message passing overhead with OOP is no heavier than an equivalent procedure call in a non-OOP language (Ford 1990)." Over the next few years as processors and computer architectures continue to advance this will become less of an issue.

Another relevant issue is the characterization of the physical plant. The task of characterizing physical devices by assigning and quantifying attributes can be enormous. Ideally standards would be developed for such characterizations and device manufacturers would provide this information. The majority of the device attribute information would not be contained in most corporate manufacturing databases. Concurrent to the characterization of the physical plant is the behavioral characterization of the plant. Using an OO implementation, the behaviors would be codified as methods and messages of each object or relationships between the objects. This task would also represent a significant undertaking. This task would be an ongoing task due to the dynamic nature of the plant.

Once a virtual plant has been codified the existing plant data base should be tied to the virtual plant. Process plans, requirements specification, process statistics, and other plant control information can be used by the virtual plant. Interface programs will need to be written that query the database from within the virtual plant. Methods will need to be added to the virtual plant to act on the data, particularly for simulation purposes.

The virtual plant will transition in stages. As portions of the virtual plant are implemented there will be a corresponding functionality also available. The entire virtual plant will not need to be implemented before it is usable. New characteristics will continually be added as will new methods. One of the biggest dangers would be insufficient maintenance leaving a virtual plant that did not duplicate the actual facility.

7 USING THE OBJECT ORIENTED VIRTUAL PLANT TOOLS

There are many conceptual uses for the virtual plant. Suppose that a customer calls desiring to know the availability and cost of a certain chemical. An engineer would sit at a terminal and invoke the virtual plant. The first step might be to interface the plant database to find candidate process plans to manufacture the chemical. From a set of process plans methods would be invoked that matched the candidate process plans with actual equipment on the plant floor. This matching is made possible by the virtual plant. Given that there are some set of possible equipment matches the engineer could ask for scheduling information. An expert system assistant would use the virtual plant with the scheduling

simulation module in conjunction with the plant database to determine several possible schedules dependent on the equipment utilized. This information could also contain cost of production data. Conceivably each alternative equipment set could correspond to a different production cost. The engineer would then be able to tell the prospective customer delivery and cost information for the chemical batch.

Suppose that the customer was dissatisfied with the production date and was prepared to pay more for an earlier delivery. The engineer could then instruct the simulator to preempt other orders to attempt to advance the schedule. There could be a cost associated with order preemption that would be transferred to the customer. In a large plant the process of check all of the prospective combinations and manipulating the scheduling information manually would be inconceivable.

One of the biggest problems facing the assignment of a chemical to a particular set of process equipment is the time necessary to create the control code to produce the desired compounds. This is another place where the virtual plant simulation tools would be utilized. Based on the process plans, code could be generated, or at least tools would be developed to aid in the creation of control code. Validation of the code would require simulation.

An important phase in the traditional control code development life-cycle is the validation and testing of the control code. One method of validation uses the control code on the actual production equipment. Although the control code must ultimately be tested on the actual production equipment, premature testing can be a costly alternative for two reasons. First, production equipment must be taken out of production during testing, second there is a cost due to potential equipment or part damage. An alternative approach is to construct hardware simulators consisting of switches used to simulate system inputs and light or meters used to simulate system outputs. These hardware devices are connected to the actual controlling computer and the control logic is tested by pressing the switches and observing the outputs. A drawback to this approach is that system inputs must be manually timed and outputs must be observed. Rapid state changes in the logic produce rapid changes in the output light sequences. Likewise, the hardware simulator must be constructed to represent the control system and then interfaced with the control computer.

A third option is to use software simulation. Approaches to software simulation of control code (Mecker 1989) generally consist of developing a software model of the manufacturing system to be controlled and executing the control program against the software model. This approach has to date been plagued with obstacles. The development of a realistic manufacturing system model against which a control program can be executed can take a large amount of time. Validation of the simulation model can be difficult. When errors occur from the execution of the logic it is difficult to ascertain if the error resulted from the control logic or from the simulation model. Finally, the model must be interfaced with the control logic. This requirement can result in additional coding for a software interface or the development of a hardware interface to couple the systems. These obstacles have made it generally impractical to perform software simulation of control code. The object oriented domain provides an environment to overcome some of these impediments.

8 CONCLUSIONS

The same principles that are being used to drive flexible manufacturing are needed to give flexibility to the process industry. A software based virtual plant that is operated on by a suite of expert system assistants is a potential solution to the flexibility needs. A completely robust set of system tools may involve a prolonged effort in gathering and maintaining a set of knowledge bases. Simulation offers a viable mechanism for handling incomplete knowledge, for extending knowledge, and for validating system outcomes.

Two simulation assistants have been proposed for a complete batch processes management architecture. These include a batch sequencing simulation assistant and a control specification simulation assistant. These assistants also have the intended capability of using the plant specifications and modeling knowledge to construct simulation models for the purposes previously mentioned. It is felt that these tools will significantly increase the batch process management life-cycle time and increase process plant flexibility.

9 PROJECT STATUS

At present, the development of the batch process management environment is in its first stage. Completion of the first stage involves the representation of the physical and information elements of the virtual plant, along with the capabilities to add or remove virtual plant elements (equipment or processes). In addition, the first stage development includes the ability to take an order input and match process specifications with appropriate equipment in the facility which will generate several potential production plans for an order. Based on the desired time frame and the current schedules of equipment, the software can claim available equipment for production or indicate that production is not possible within the rime frame under consideration.

In a separate project, a generic control specification is

being developed and simulated. The tools developed by this separate project will be incorporated into the batch management architecture.

Further development activities are being conducted which will add the ability to generate control schemes for the production code will be performed through plant simulation using information contained in the virtual plant specification. This information will be a data and functional specification of equipment allowing the devices to be modeled and exercised through combined next event and continuous simulation.

REFERENCES

Baudel, B. Cantegrit, E., and Toulette, J. M. 1988 Smalltalk and Simulation of Batch Processes, *Artificial Intelligence, Expert Systems and Languages in Modelling and Simulation,* Elsevier Science, pp. 295-299.

Faught, W. 1986. Applications of AI in Engineering, *Computer,* July 1986, Vol. 19, pp. 17-27.

Barr, A. and Feigenbaum, E. A., *The Handbook of Artificial Intelligence,* Vol. 2, William Kayfmann Inc., Los Altos, CA.

Freeman, D. D. 1985. Artificial Intelligence Applications in Process Control." *Proceedings of the American Control Conference,* 1985.

Ford, D. and Schroer, B. 1987. An Expert Manufacturing Simulation System, *Simulation*, May 1987, vol. 48, pp.193-200.

Ford, W. R. 1990. Object Oriented Programming: Will it work for real-time systems?, *Computer Design*, Vol. 29., pp. 44-45, March .

Gidwani, K. K. 1985. The Role of Artificial Intelligence Systems in Process Control. *Proceedings of the American Control Conference.*

Hatley, D. J. and Pirbhai, I. A. 1987. *Strategies for Real-Time System Specification*, Dorset House, NY.

Massey, L. E. and McCarthy, J. J. 1990. Batch Process Automation for the 90's. Honeywell, Inc. I.A.C.D. Phoenix Az.

McCarthy, J. J. 1990. The CIM Reference Model as a Tool for Plant Information and Control Systems Development. Honeywell, Inc. I.A.C.D. Phoenix Az.

Musier, R. F. H. Evans, L. B. 1990. Batch Process Management. *Chemical Engineering Progress,* June.

Shannon, R. E. Models and Artificial Intelligence. *Proceedings of the 1987 Winter Simulation Conference*, Atlanta, GA, 1987, pp. 16-24.

Suydam, W. 1989 CASE Makes Strides Toward Automated Software Development, *Computer Design*, January.

The Information Structure in a CIM Architecture, A *Report of the Advanced Technical Planning Committee of Computer Aided Manufacturing-International Inc, CIM Review*, Spring 1989, 5-4.

Vick, C. R. 1984. Introduction: A Software Engineering Environment, in *Handbook of Software Engineering*, Van Nostrand Reinhold, NY.

Waters, R. C. 1985. The Programmers Apprentice: A Session with KBEmacs, *IEEE Transactions on Software Engineering*, November.

Wilson, R. 1987. Object Oriented Languages Reorient Programming Techniques. *Computer Design*, November pp. 52-62.

AUTHOR BIOGRAPHIES

DR. CHELL A. ROBERTS is an Assistant Professor in the Department of Industrial and Management Systems Engineering at Arizona State University. He holds a BS (University of Utah, Math), MS (University of Utah, Industrial Engineering), and Ph.D. (Virginia Polytechnic and State University, Industrial Engineering) degrees. His interests are in simulation, manufacturing control, and artificial intelligence. He is a member of IIE, SCS, ASEE, and SME.

DR. TERRENCE G. BEAUMARIAGE is an Assistant Professor in the Department of Industrial and Management Systems Engineering at Arizona State University. He holds a BS (Rochester Institute of Technology), MS (Oklahoma State University), and Ph.D (Oklahoma State University) degrees, all in Industrial Engineering. His interests are in simulation, manufacturing, artificial intelligence, and quality control. He is a member of IIE, SCS, ASQC, and ASEE.

YASSER DESSOUKY is currently a Ph.D. candidate in the Department of I&MSE at ASU. He holds a BS (University of Wisconsin-Madison) and MS (Arizona State University) degrees, both in Industrial Engineering. He has worked as a systems analyst at TRW and at Pritsker and Associates, and is a member of IIE.

MICHAEL K. OGLE is an Assistant Professor in the Department of Mechanical and Industrial Engineering at Louisiana Tech University. He holds a BS in Mechanical Engineering and an MS in Industrial Engineering, both from the University of Arkansas. He is currently pursuing a Ph.D. from the ASU Department of Industrial and Management Systems Engineering. His interests are in information systems in manufacturing, object oriented analysis, and simulation. He is a member of IIE, NSPE,ACM, ASEE and SCS.