HIGH-FIDELITY REAL TIME DYNAMIC PROCESS SIMULATION with OBJECT-ORIENTED PROGRAMMING
PART 1: THE TOOL

W.P. Meincer
Simons-Eastern Consultants, Inc.
Decatur, Georgia
U.S.A.

M.S. Ma
Simons-Eastern Consultants, Inc.
Decatur, Georgia
U.S.A.

F.G. Powell
H.A. Simons Ltd.
Vancouver, B. C.
Canada

M.F. McGarry
IDEAS Simulation Inc.
Vancouver, B. C.
Canada

ABSTRACT

This is the first of a 3 part series of papers published at the Control Systems '92 Conference, Whistler, B.C. describing the development of an advanced control application from concept to startup. Part 1 describes the tool and development environment. Part 2, ***Fixed Time Zone Methodology for Plug Flow Simulations as Applied to an Oxygen Delignification Reactor*** describes the process model and Part 3, ***Adaptive Predictive Nonlinear Control*** describes the resulting advanced control strategy.

This paper describes a dynamic process modeling tool which was developed using object-oriented programming techniques. High fidelity is achieved using mathematical relationships from the first principles of physics and chemistry as well as empirical data. The resultant tool is easy for designers to use. It can be applied to the modeling needs of virtually every aspect of an engineering project from concept, through process and control design to control system checkout, to operations training, and ongoing operations troubleshooting and optimization.

INTRODUCTION

Engineers and operations managers have traditionally been unable to use one modeling platform for multiple purposes such as process design, training, Distributed Control System (DCS) checkout, advanced control design and testing, training and process optimization. Because of the mathematical rigor required for high fidelity dynamic models, process design has been dominated by steady-state simulation. Where high fidelity dynamic models were developed, they were usually unsuitable for DCS checkout or training because they were too complex to run in real time with affordable computer resources. Embedded DCS simulations and many external operator trainers usually required compromises in the models to obtain real time performance, resulting in low to medium-high fidelity process simulations. The cost of developing and running large scale high fidelity simulations for control system checkout and operator training could not be justified except in extreme cases.

Recognizing the need to combine high model fidelity suitable for process and control design and real time operation into a single package, the authors participated in a development effort to address these issues. The objective was to design a tool for **I**ntegrated **D**esign **E**ngineering with **A**dvanced **S**imulation (**IDEAS™**) that would be a key contributor to the the complete life cycle of a process plant, including:

- Initial Process Conceptual Design
- Detailed Process Design
- Advanced Control Design
- DCS Checkout
- Operator and Maintenance Training
- Process Optimization

General Description

The resulting development is a graphical design tool which uses object-oriented programming techniques (see Appendix A) based on a modified form of the C programming language.

Object-oriented programming techniques result in a software system that is well organized and is especially suited to high performance dynamic simulation. The objects become analogous to pieces of process equipment and instrumentation. The messages passed between objects are analogous to the information carried through a process pipe from one piece of equipment to another.

The object-oriented approach requires no communications or subroutine calls other than the message passing. Furthermore, the message structure between objects is transparent to the user, since it is managed by graphically connecting objects on the screen.

The object-oriented approach also lends itself especially well to an icon-based graphical interface that is essential to ensure acceptance by the process and process control design engineers and practical utilization by plant personnel. An icon-based graphical programming environment ensures that the graphics are an integral part of the model building process.

This has productivity advantages over traditional graphics front-ends that require building of the model through non-graphical techniques, then building a graphic and linking it to the appropriate parts of the model.

The model is built starting with a blank computer screen upon which objects are placed from various object libraries (refer to Object Type Descriptions below). Figure 1 illustrates an example. These objects are connected using a mouse to build a model analogous to the corresponding flowsheet (P&ID or P&C). Each object is given specific data to give it the same characteristics as its real world counterpart. This is done by opening a dialog box for each object and filling in information. This dialog box appears when the object is selected and double-clicked by the mouse. Figure 2 shows a typical dialog box with information required from the user.

Model Fidelity. The fidelity of a model is a measure of how closely the model parallels the real process. The higher the fidelity the closer the model matches the real process under the same conditions. High fidelity is achieved by using the first principles of physics and chemistry to describe process operations where it is possible and empirical data from field and laboratory testing where it is not possible. The piping network pressure and flow relationship also contributes to the model fidelity. Nearly all dynamic modeling platforms use the physical laws of conservation of energy and mass in solving flows through pipes. The further use of conservation of momentum, however, is not as universal but is really necessary to achieve high fidelity in many applications. This simulator does include the conservation of momentum in its piping network solution.

Often the solution techniques used to solve the piping flows and pressures only work with flow going in the positive direction. The technique used in this platform, however, allows reverse flow. If, for example, the piping objects are connected together without check valves, the simulator allows for flow in either direction for all connections and provides for the appropriate mixing calculations to cover these situations for all stream components. The pressure dynamics of the model determine the direction of flow, just like they would in an operating plant.

Object Type Descriptions. The objects used in this simulator are classified into seven types:

1. Process Objects
2. Pressure/Flow Network Objects
3. Material Properties/Stream Definition Objects
4. Control Objects
5. Communication Objects
6. Data Collection and Display Objects.
7. Training Objects

The **Process Objects** have a one to one correspondence to real world equipment. This allows the simulation to be built directly from a P&ID or flow sheet by retrieving objects from a library and connecting them together in the proper configuration on a worksheet using a mouse. Figure 1 illustrates a portion of a model showing these objects and the manner in which they are connected. First principles are used where possible to provide the modeling equations and when not possible empirical relationships are used. The process objects are modeled independently of the fluid being processed. For example, a heat exchanger would function properly regardless of whether it is fed water or oil. It merely processes the materials which arrive in the streams connected to it. The physical properties for each material in the model are available to each object on the worksheet.

The **Pressure/Flow Objects** consist of pipes, valves, pumps, pressure nodes (pipe junctions) and network solver objects for compressible and incompressible fluid flow. The person building the simulation enters the following information in the dialog boxes for each of these items:

OBJECT	PARAMETERS
Pipe	•Pipe Diameter •Pipe Length •Pipe Roughness •Fitting Resistances •Compressible Or Incompressible Fluid
Valve & Pipe	Above plus: •Cv at 100% opening •Linear or Equal Percentage Trim •Percentage open and delta pressure for automatic sizing mode
Pump/Suction Pipe	•All standard pipe parameters for suction piping (above) Curve fit coefficients for: •Pump Curve (two-dimensional for variable speed or selectable impeller pumps) •NPSH Curve •Maximum Flow Curve •Elevation of pump centerline •Flow for sizing mode
Pressure Node	•Elevation •Compressible/Incompressible Switch
Solver	•Convergence Limit •Maximum No. of Iterations.

The solver is capable of reaching a fully converged piping network solution for mass, energy, and momentum every simulation step. The flow in each piping branch is calculated either in the conventional manner using the Colebrook equation for Newtonian fluids or using empirical coefficients for non-Newtonian fluids.

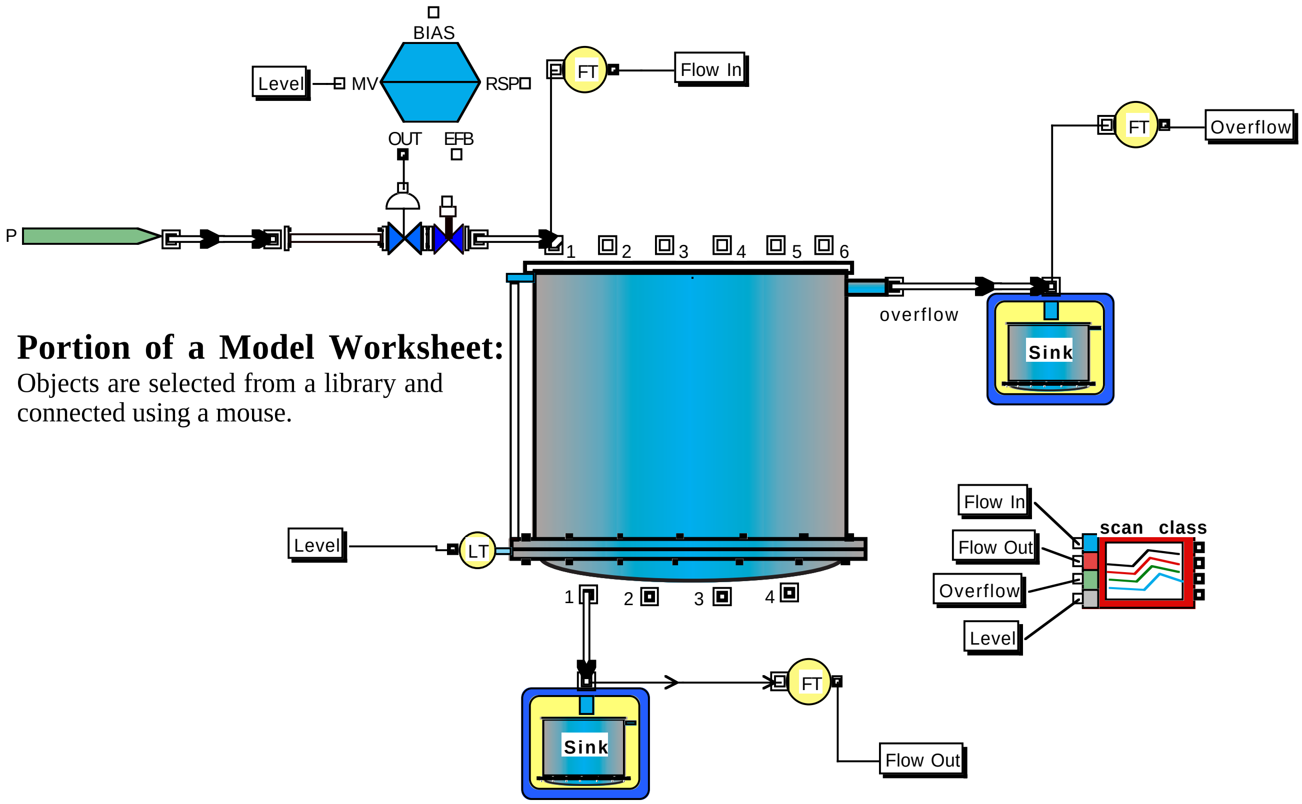


FIGURE 1. A Portion of an IDEAS™ Model Worksheet

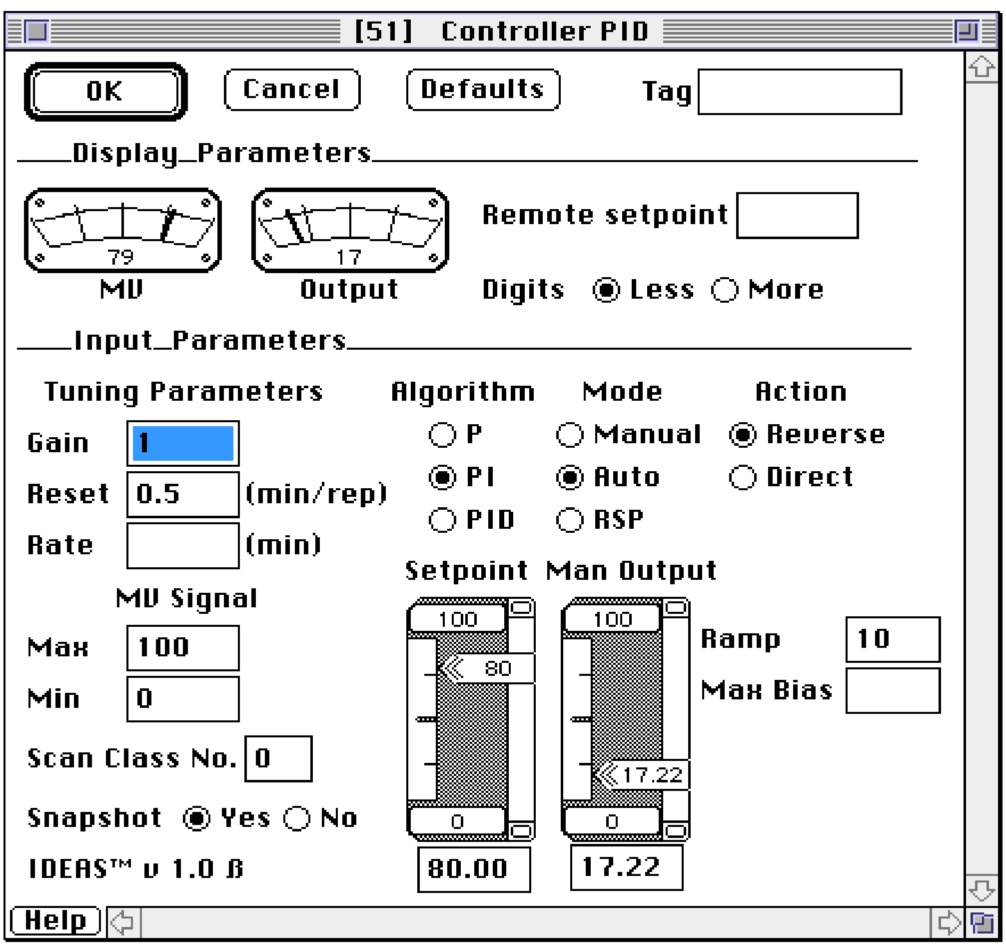


FIGURE 2. An IDEAS™Dialog Box for a PID Controller

The **Material Properties/Stream Definition Objects** allow the user to define the various materials which will make up a fluid stream in the model. Flowing through each pipe, valve pump, etc., is an array (the stream array) which is comprised of general information about the stream such as temperature and pressure as well as specific information about each component material in the stream such as mass fractions and component flows. Properties for these materials are also available to each object in the model. These material properties include but are not limited to:

- Density as a function of temperature and pressure
- Viscosity as a function of temperature
- Enthalpy as a function of temperature and pressure
- Heat Capacity as a function of temperature

The mass fraction proportions of the various materials may be specified at each source location in the model. These proportions, as well as source pressure and temperature, can be dynamically varied during the simulation run. The variations available include random noise, ramps and sine waves. These variations can be programmed to occur from another object outside the source object.

The number of components in the stream is adjustable. A recent model of a wood pulp bleaching process, for example, used 15 total stream components.

The **Control Objects** have a one to one correspondence with physical instruments such as transmitters, analyzers, stand-alone controllers, etc. or with DCS functions such as PID control algorithms, high/low select blocks, etc. They are used to provide control functions to the process model. The objects can be either generic control objects or objects corresponding to a specific equipment vendor. Figure 3 shows part of the control logic which was reproduced from the actual DCS configuration documents (in this case Bailey™ Cadews™). If the end use of the model is as a stand-alone simulator, such as an operator trainer, the control will be part of the model. If, however, the model will be used for DCS checkout, the control will reside in the actual DCS hardware/software and the process model will interface to this control via the I/O sub-system discussed later.

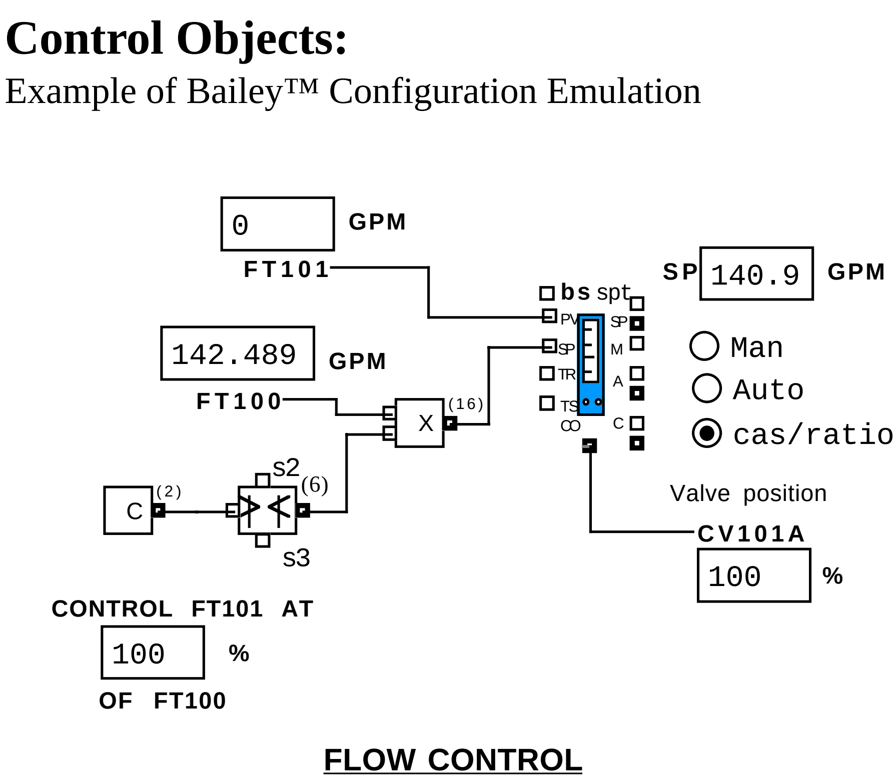


FIGURE 3. IDEAS™ Control Objects

The **Communication Objects** are used to interface control signals or stream array information to other computers, to the I/O subsystem, or to a DCS.
These objects operate in conjunction with a communication driver object. DCS interface objects which allow direct communication with Bailey Infi-90™ and Rosemount System 3™ consoles have been developed and others are in development.

The **Data Collection and Display Objects** allow the user to observe and record various information in the model. Plotter objects can be connected to control signals or to stream variables (by using transmitters to extract the proper variable from the stream array).

These objects graphically plot the information as the model runs as well as present the information in tabular form (see Figure 4).

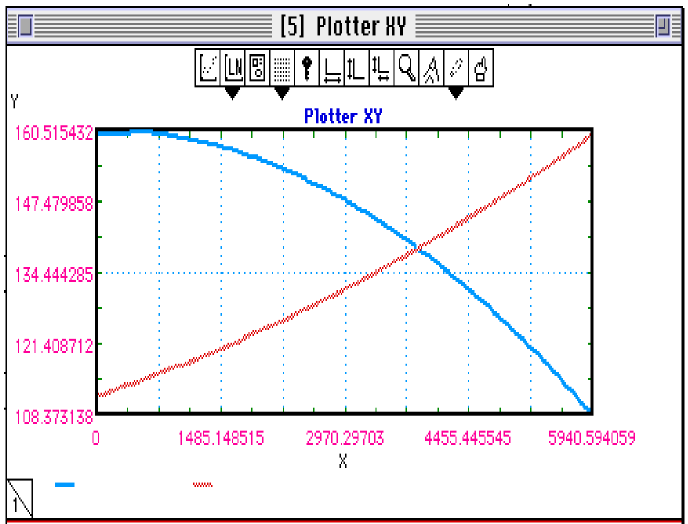
The Snapshot object allows worksheet conditions to be stored as a file for later retrieval and replay.

The **Training Objects** allow the creation and execution of preprogrammed plant operations scenarios. When the model is configured for training it includes all control functions and communicates directly with a DCS console so that the operator can control the simulated process from the console just as he would the real process. Using the Scenario Manager™ object, the trainer can initiate one or more failure/anomaly scenarios for the operator. Other objects work in concert with the Scenario Manager™ to create changes in setpoints, trip pumps, fail transmitters either high or low, etc. The plotter objects can then be used to record and review the operator's responses. For example, one scenario could cause the consistency in a stock feeder to increase by reducing dilution flow because of a flow transmitter "failure". The operator should notice the gradually increasing feeder amperage and manually increase the dilution flow. If the operator failed to do this the internal control logic would, as would the real control system, trip the feeder. Figure 5 shows a typical Scenario Manager for a bleach plant model.

Advanced Control Design

The high fidelity of this simulator allows it to be used to develop and check-out advanced control strategies. An example of using this modeling method for advanced control design and check-out is with an O_2 delignification reactor control strategy. The proprietary strategy uses a combination of an embedded kinetic model with an adaptive one, an adaptive/predictive controller and an on-line parameter estimation technique that "learns" a process' idiosyncrasies. It is designed to successfully cope with varying dead-times and control outlet Kappa for minimum variation based upon accurate predictions of process reactions to disturbances. This strategy is described in the companion paper, Part 2, referenced in the abstract.

The process objects are as high in fidelity as the available information allows. For example the high fidelity O_2 delignification system model has a reactor that is divided into



Point	1	1	2	2
0	0	160.51057372	0	111.6899077
1	59.405940594	160.51543177	59.405940594	111.99587594
2	118.81188118	160.50965887	118.81188118	112.30532338
3	178.21782178	160.49325501	178.21782178	112.61825001
4	237.62376237	160.46622019	237.62376237	112.93465585
5	297.02970297	160.42855442	297.02970297	113.25454089
6	356.43564356	160.38025768	356.43564356	113.57790512
7	415.84158415	160.32132999	415.84158415	113.90474856
8	475.24752475	160.25177134	475.24752475	114.23507119

FIGURE 4. Plotter Object

143 zones where material, energy, momentum balances and reaction kinetics are computed in each zone in real-time. This model is described in the companion paper, Part 3, referenced in the abstract.

Operations Management

When used during the original process design phase this simulator will result in an optimized plant design with fewer dynamic problems and operational restrictions.

Greenfield installations or plant expansions benefit from reduced start-up time and off-specification product due to operator familarization with the new process and controls.

This simulator is useful for managing the day to day operations of a process plant. It provides a powerful "what-if" tool for use in plant process troubleshooting and maintaining optimal process performance. It can be used to schedule production to minimize product change-over waste, optimize, raw material usage and minimize power consumption. Maintenance outages and start-ups can be scheduled more effectively. Process and control system modifications can be tested prior to implementation to minimize upsets and optimize results.

Communications Capabilities

Serial Communications with DCS Consoles. In order to accomplish operator training it is necessary for the model to communicate with a DCS console of the exact type the operator will use on the real process. The model and the controls would reside in the simulator. Special communication objects (described above) would be placed on the model worksheet to provide communications with a DCS console. A special serial driver object then communicates the appropriate control signal to and from the console via a serial link.

I/O Subsystem Connections. In order to use the model in DCS checkout the model must be able to interface to the actual DCS control hardware. Other systems in use for DCS checkout often use a "gateway" technique to communicate the simulation parameters directly to the DCS controller without going through the DCS I/O wiring. This allows the testing of the DCS configuration but does not check out the I/O hardware and various switch settings, dip shunts, etc. used to configure output channels on many DCS types. This simulator allows the user to interface to a DCS under test, directly through the I/O terminals which will be used in the field. The device which emulates the I/O signals is called the Advanced Systems Test Simulator (ASTS, Figure 6). TCP/IP protocol is used to communicate between this subsystem and the simulator. This high fidelity simulation approach allows testing of not only basic controls but also higher level supervisory and advanced control applications. The high speed of the I/O subsystem also allows testing of drive control systems requiring process response to speed control loops within a 10 millisecond scan rate.

Multiple Computers Used for Large Models. Sometimes the scope of a specific modeling project requires a model that is too large to run in real time on one computer. When this occurs the model is split into logical process areas and is run on two or more computers. The computers communicate with each other also using TCP/IP protocol. The model is synchronized from one process area to another by using a modified form of the standard pipe object. This pipe object allows the user to specify the pipe diameter, length, roughness, number of fittings and other parameters for this inter-computer process link.

BENEFITS

The benefits from the perspective of those most likely to use this simulator are:

Control Designer's Perspective:

- Process dynamics are modeled accurately enough to allow development and testing of complex control strategies
- Complex logic can be tested before startup
- Control design can be optimized for startup, shutdown and upset conditons
- Can size control valves, pumps and piping systems simultaneously

Process Designer's Perspective:

- Improved communications with the client
- Improved coordination of design between process engineers and control engineers
- Dynamic capability results in design of the process as a <u>system</u>
- Optimized design for startups, shutdowns, upset conditions
- Allow troubleshooting of very complex process problems
- What-if scenarios allows optimization of process design and equipment sizing
- Increase in quality by "doing the right thing right the first time"

Plant Perspective:

Operator Training
- Realistic process response to control input
- Hands-on experience without the potential equipment damage and schedule restrictions
- Can test what-if equipment/operator failure scenarios
- Allows familiarization with control equipment and strategies before startup
- Can spot weakness in operator skills before startup

Operations Management
- Results in an optimized plant design from both a process and control standpoint
- Minimized down-time and start-up
- Optimized ongoing operation
- Effective maintenance training
- Planning of production changes and maintenance outages
- Subsequent process changes can be made with confidence

CONCLUSION

The object-oriented icon-based software approach results in a tool that is easy for process and process control designers to use. Armed with a P&ID the user can build detailed simulations significantly faster than other approaches of similar fidelity.

The benefits to designers and operating plants are significant and will revolutionize the way process and process control engineering is done on industrial projects.

The high fidelity objects allow this platform to be used across a wide range of uses throughout the life-span of a project.

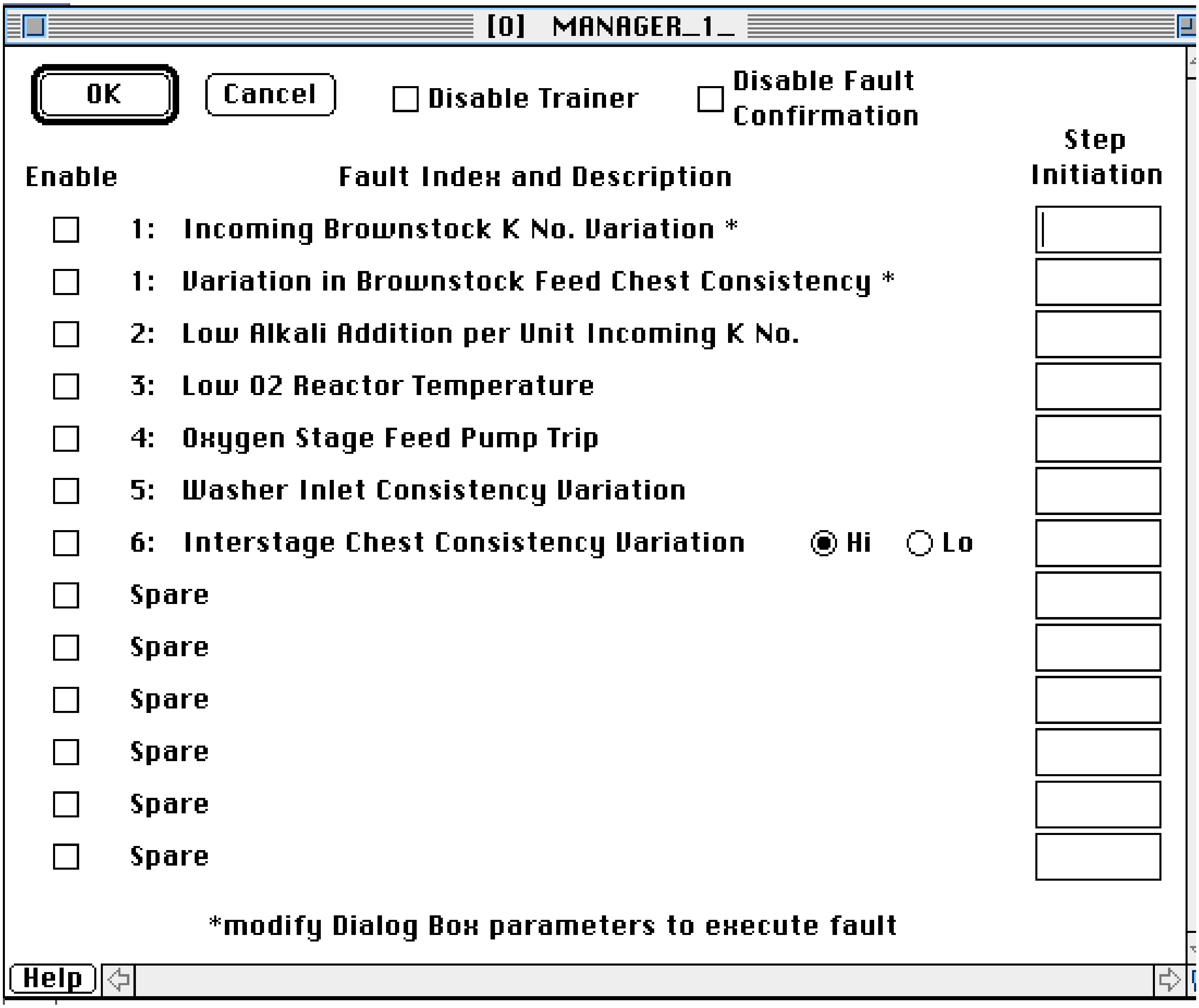


FIGURE 5. Training Scenario Manager™ Object Dialog Box

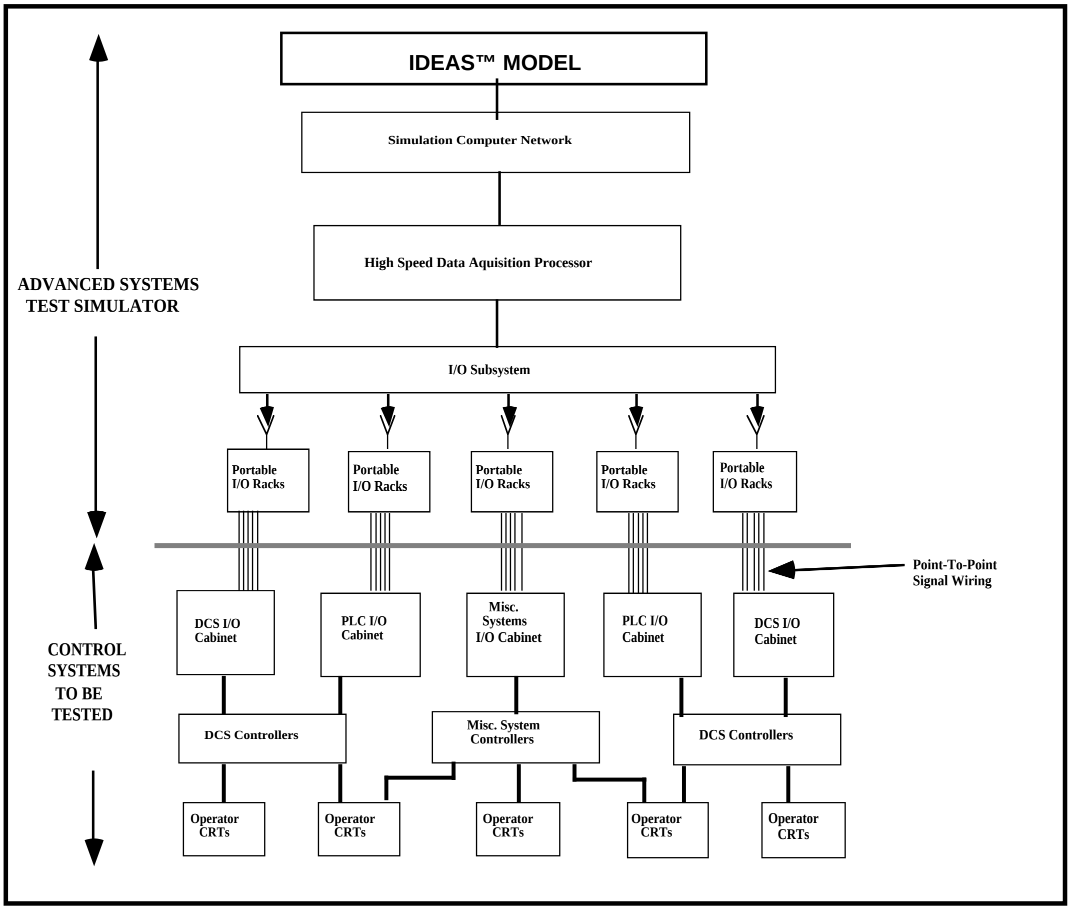


FIGURE 6. Advanced Systems I/O Subsystem

APPENDIX A.

Object-oriented programming. Object-oriented programming is characterized by the following three major software design features:

1. Data encapsulation - Both data and procedures reside within software entities called objects.

2. Polymorphism - Each object operates only on messages received from and transmitted to it from other objects. Polymorphism is the ability of one type of object to interpret an identical message differently from that of another type.

3. Inheritance - When an object is created that is a member of a class of objects it automatically inherits the characteristics of that class. (Inheritance of attributes of one object to another is not strictly supported in this platform although objects can be duplicated, edited and renamed to produce a similar but different object.)

The objects reside in a library and when placed on a model worksheet only the data unique to that instance of that object resides there. The actual programming code remains in the one object in the library. If, for example, a change must be made to a pipe object program it is made one place and will automatically be used for all instances of that pipe object.

REFERENCES:

1. DeWitt, S. A., Noy, A. H. ,“Multi-Vendor Systems Integration/Checkout: A Solution To The Problem”, 1990 TAPPI Engineering Conference, Seattle, WA., September 24-27, 1990.

2. Andersen, J. P., “Distributed Control System Testing”, ISA90, October 1990, New Orleans ISA Conference Proceedings.

3. McGarry, M.F., Ulinder J.D., Powell F.G., “The Coming Object-Oriented Revolution in Process and Process Control Design”, ISA91, October 1991, Anaheim ISA Conference Proceedings.

4. Meincer, W.P., Andersen, J.P., “Real Time Dynamic Process Simulation-A Spectrum of Choices”, 1991 TAPPI Engineering Conference, Nashville TN., October 1991.

5. Ulinder,J.D., "Fixed Time Zone Methodology for Plug Flow Simulations as Applied to an Oxygen Delignification Reactor - Part 2 Process Model Development", Control Systems '92, Whistler, B.C.September, 1992

6. DeWitt, S.A. , Wang, P.H., "Adaptive Predictive Nonlinear Control - Part 3 Advanced Control Development, Simulation Testing and Mill Performance", Control Systems '92, Whistler, B.C., September, 1992