Dialog Box:
The user interacts with a block through it's 'Dialog Box'. The dialog box is a window shown by clicking twice on the graphic (or 'Icon') of the block. The dialog box can contain variables, buttons, checkboxes, radio control buttons, data tables, as well as text. These variables, buttons, etc. can be changed by the user by typing in new values or by pointing and clicking with the mouse. A third layer, which normally would be invisible to the user, is the source code. Variables in the dialog box as well as variables received from other blocks are all available to the source code. Output can be sent to plotters which plot while the model is running or, if needed, messages can be sent to the screen. A simulation can also write variables to a report file and/or a debugging file. So the user has all the power of compiled source code at his fingertips but with the flexibility to alter input, insert new model blocks and view results dynamically all without having to enter or even see a line of code.

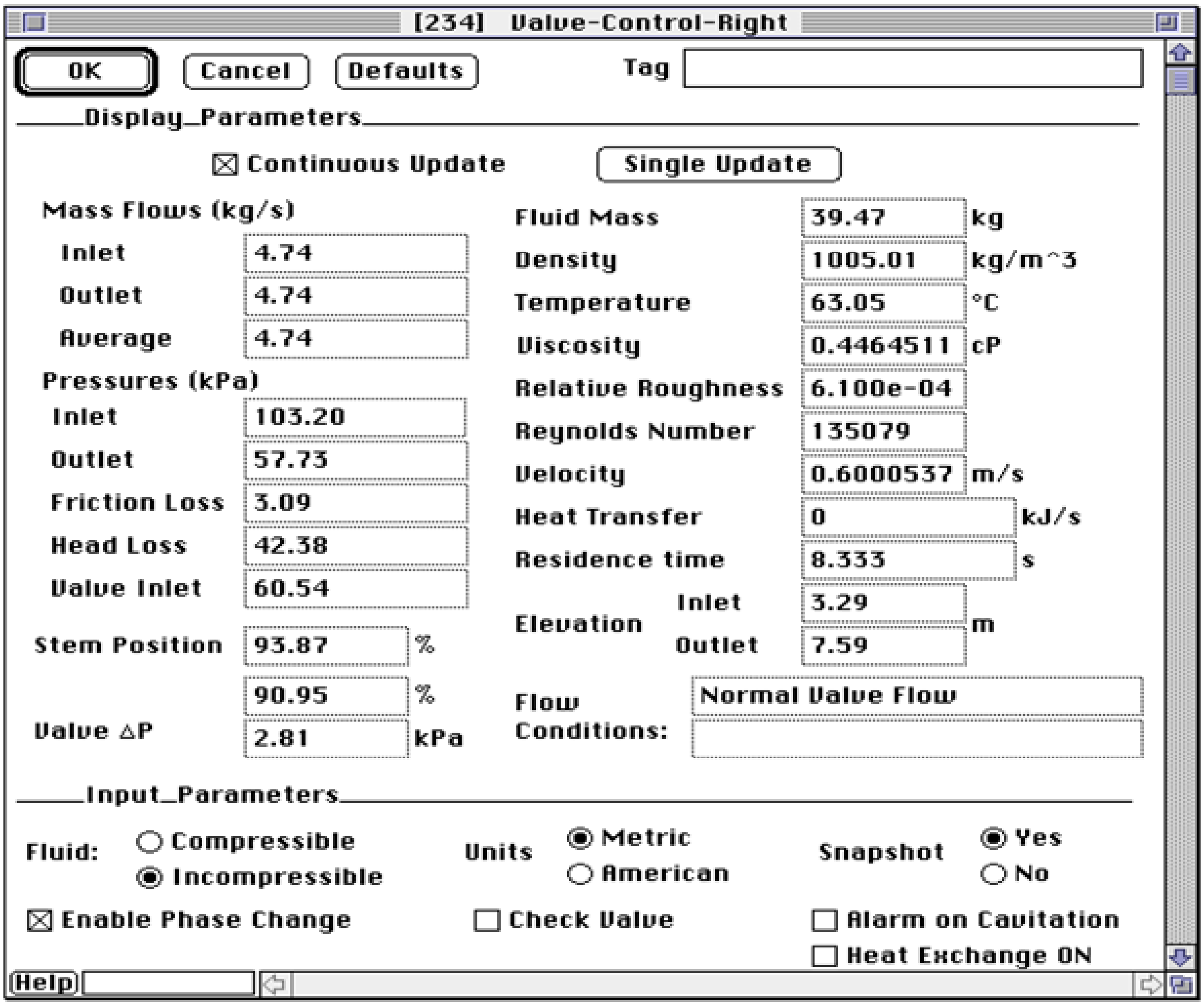


Figure 2
A Typical Control Valve Dialog Box

Object-oriented programming techniques result in a software system that is well organized and is especially suited to 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 control equipment to another.

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). 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 accurate results in many applications. Simons IDEAS™ 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 Simons IDEAS™ 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 - Incompressible Flow •Maximum No. of Iterations - Incompressible Flow. •Convergence Limit - Compressible Flow •Maximum No. of Iterations - Compressible Flow.

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.*

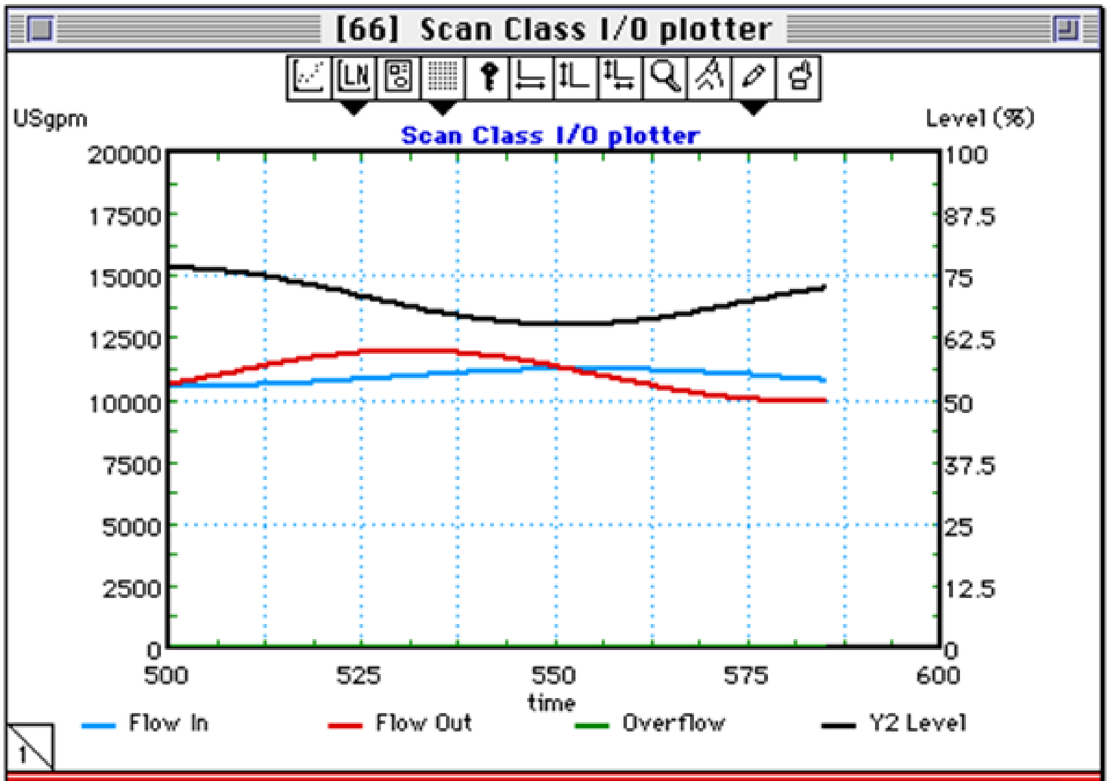
*The **Material Properties/Stream Definition Objects** allow the user to define the various materials which will make up a fluid stream in the model. Flowing through each pipe, valve pump, etc., is an array (the stream array) which is comprised of general information about the stream such as temperature 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
- Constants such as Melting Point, Boiling Point, Molecular Weight, etc.

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 metallurgical process, for example, used 54 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. If the end use of the model is as a stand-alone simulator, the control will be part of the model. If 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 a virtual I/O communication card on the DCS.

The **Communication Objects** are used to interface control signals or stream array information to other computers, to the PLC, or to a DCS. These objects operate in conjunction with a communication driver object on the worksheet. DCS interface objects which allow direct communication with Bailey Infi 90™, ABB Advant, Rosemount System 3™and Allen Bradley PLC have been developed and others can be made available within two month's notice.

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). The Snapshot object allows worksheet conditions to be stored as a file for later retrieval and replay. Plotter objects graphically plot the information as the model runs as well as present the information in tabular form (see Figure 3).



time	1 Flow In	2 Flow Out	3 Overflow	4 Level
0	10505.713762	10658.893005	0	64.726796966
1	10505.713762	10658.893005	0	64.578285913
2	10630.512861	10719.220112	0	64.549781922
3	10695.090577	10780.528684	0	64.522329138
4	10926.432849	10842.576765	0	64.549273244
5	11002.030280	10905.119479	0	64.580412393
6	11077.881531	10967.909999	0	64.615748186

Figure 3
A Plotter Recording Tanks Levels and Flows

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.

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.

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

Simons IDEAS™ is useful for managing the day to day operations of a process plant. It is 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.

BENEFITS

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 conditions
- 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 driving forces of lower operating costs, lower process emissions coupled with higher quality products will push the application of dynamic simulation toward the front-end of process design. Ease of use and transportation of dynamic models will link process designers with process operations. The same dynamic models that are used in the front-end design work can migrate through to the detailed design process and on to the DCS configuration check-out. Once the process is put on-line, dynamic models can be applied to on-going plant evaluation and optimization. Simons IDEAS™ puts the power of dynamic process and control models into a form suitable for users who are not computer experts or dynamic simulation experts.

REFERENCES

(1) J.P. FLETCHER, J.E. OGBONDA, 'A Modular Equation-Oriented Approach to Dynamic Simulation of Chemical Processes', Comput. Chem. Engng, Vol.12, No. 5, pp. 401-405, 1988.