

Using CONFIG for Simulation of Operation of Water Recovery Subsystems for Advanced Control Software Evaluation

Jane T. Malin

NASA Johnson Space Center

Luis Flores

Lockheed Martin Space Operations

Land Fleming

Hernandez Engineering, Inc.

David Throop

The Boeing Company

DRAFT

ABSTRACT

A hybrid discrete/continuous simulation tool, CONFIG, has been developed to support evaluation of the operability life support systems. CONFIG simulates operations scenarios in which flows and pressures change continuously while system reconfigurations occur as discrete events. In simulations, intelligent control software can interact dynamically with hardware system models. CONFIG simulations have been used to evaluate control software and intelligent agents for automating life support systems operations. A CONFIG model of an advanced biological water recovery system has been developed to interact with intelligent control software that is being used in a water system test at NASA Johnson Space Center.

INTRODUCTION

There has been significant progress in development of object-oriented and graphical modeling and simulation tools that can be used for analysis of environmental control and life support systems [9, 12, 13, 14]. In these tools, system models are composed of a set of connected component models. Simulation is typically used to support sizing and optimization of system performance. CONFIG is a hybrid discrete/continuous simulation tool that, in addition to these more traditional uses, has been designed to support evaluation of system safety, operability, and operations.

CONFIG has been used for testing control software by interactive simulation. This provides a means of evaluating the software's responses to simulated failures

or off-nominal system states that would be too costly, dangerous, or otherwise infeasible to induce in the real hardware system for test purposes. In simulation, we have uncovered deficiencies in software function and requirements definition that went unnoticed even after thorough testing by more conventional means [9]. In this evaluation, unanticipated system configurations were uncovered as the simulation interacted dynamically with the control software.

In reviewing designs or testing code, humans depend on their own mental models of the system. Mental models may be incomplete or incompletely recalled at some critical juncture in design, implementation, or testing. This is especially a problem when the adverse consequences of a software action are either spatially or temporally remote from the immediate object of the action. For example, a designer may verify that, given the appropriate sensor data, the software will properly command a normally closed valve to open to permit flow into a tank. However, it may not account for a second normally open valve, also on the path of flow that, in some scenarios, may have previously been commanded closed. It must also be commanded open, in order for flow to commence. In a system accident [6], such an oversight might not be discovered until the tank has reached a critically low level and some serious failure has occurred in a device being fed by the tank. It is preferable to uncover such problems in a simulation rather than in the real hardware system.

CONFIG has been used to model gas and water processing systems, a thermal control system and a data network for space subsystem designers. A number of CONFIG models of life support systems have been

developed to support a testbed for designing and evaluating advanced software control technology [11].

In this paper, we describe a model of an advanced biological water processing (BWP) system for wastewater. This model interacts dynamically with intelligent control software. The BWP model illustrates CONFIG features and capabilities.

WATER RECOVERY SYSTEMS AND CONTROL

THE SYSTEM

The BWP is a subsystem of a Water Recovery System that has been designed, built and tested by the Crew & Thermal Systems Division (CTSD), Life Support and Habitability Test Branch at NASA Johnson Space Center. Figure 1 shows the four subsystems of the WRS [4]. The system is currently being tested to demonstrate the feasibility to produce potable quality water.

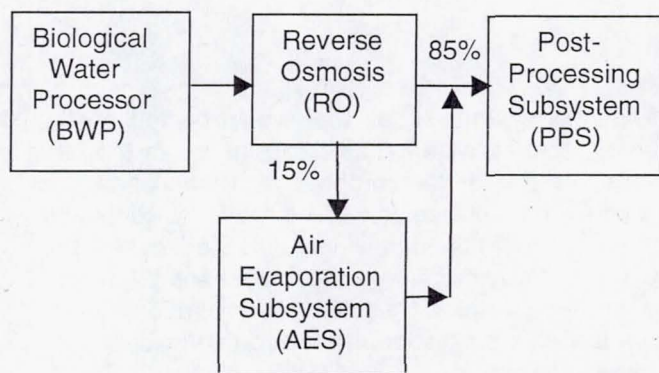


Figure 1. WRS Subsystem Flow Diagram

The Biological Water Processor (BWP) takes in raw gray water and reduces the organic carbons and the ammonium. The Reverse Osmosis (RO) unit reduces inorganic substances by forcing the BWP water through a membrane. The Air Evaporation Subsystem (AES), extracts water out of brine, a byproduct of the RO. Hot air evaporates the water in the wick. The air is chilled in a heat exchanger, and the condensate has only trace inorganic substances. The condensate from the AES and purified water from the RO go to the Post Processing Subsystem (PPS). A single water path from the feed flows through a bank of ion exchange beds that remove trace inorganic compounds, and then flows through a bank of ultraviolet lamps that remove trace organic carbons.

BIOLOGICAL WATER PROCESSOR

The BWP consists of a continuous loop with two biological processors and a Gas Liquid Separator (GLS) in a single path. A recycle pump keeps the flow through the system at the desired level. Figure 2 shows a

snapshot of the CONFIG model of the BWP during simulation.

The first processor is the Packed Bed Biological Water Processor (PBBWP). This reactor consists of a tank packed with ceramic saddles that support the growth of anaerobic microorganisms within the reactor. These microorganisms use nitrates and nitrites provided by the nitrification process, next in the loop, to reduce the organic carbon in the wastewater.

The Nitrifier consists of four pairs of polypropylene tubes that are populated with aerobic microorganisms along the inside walls. These microorganisms convert the ammonium in the wastewater to nitrates and nitrites. The oxygen needed by the microorganisms is provided by air pumped into the tubes. An airflow controller regulates the airflow to each tube. Pumps force water through each pair of Nitrifier tubes at the appropriate flow rate. There is a bypass around the Nitrifier to ensure that the pressure at the head of the Nitrifier is kept at the correct level.

The GLS separates the Nitrifier effluent gases and liquids, and vents the gases. In steady state mode, wastewater is fed back to the PBBWP while the stream is tapped at the GLS to feed the RO.

About 95% of the liquid is recycled back into the PBBWP and 5% is directed to the RO. The inflow and the outflow are balanced to maintain a constant flow through the two biological processors and the Nitrifier bypass. An outside loop (not shown) provides fluid paths so that each processor can operate stand-alone, and can be inoculated with the appropriate microorganisms.

CONTROL SYSTEM

An advanced autonomous control system controls the WRS testing. The control system is implemented in the 3T intelligent software architecture for autonomous systems control [2]. This three-tiered architecture includes a planner, a reactive sequencer and a low-level control tier. The planner tier has not been used in controlling the WRS. The bottom tier of "skills" consists of small modules, each of which is responsible for controlling a specific device or reading a specific sensor.

The middle tier is a conditional task sequencer that controls the modules of the skills tier. The sequencer automatically executes procedures for both nominal and off-nominal operations of the integrated WRS and its subsystems. These procedures include system startup, shutdown, standby, mode changes, nominal monitoring procedures, and the detection and either recovery or implementation of safe modes from emergency situations.

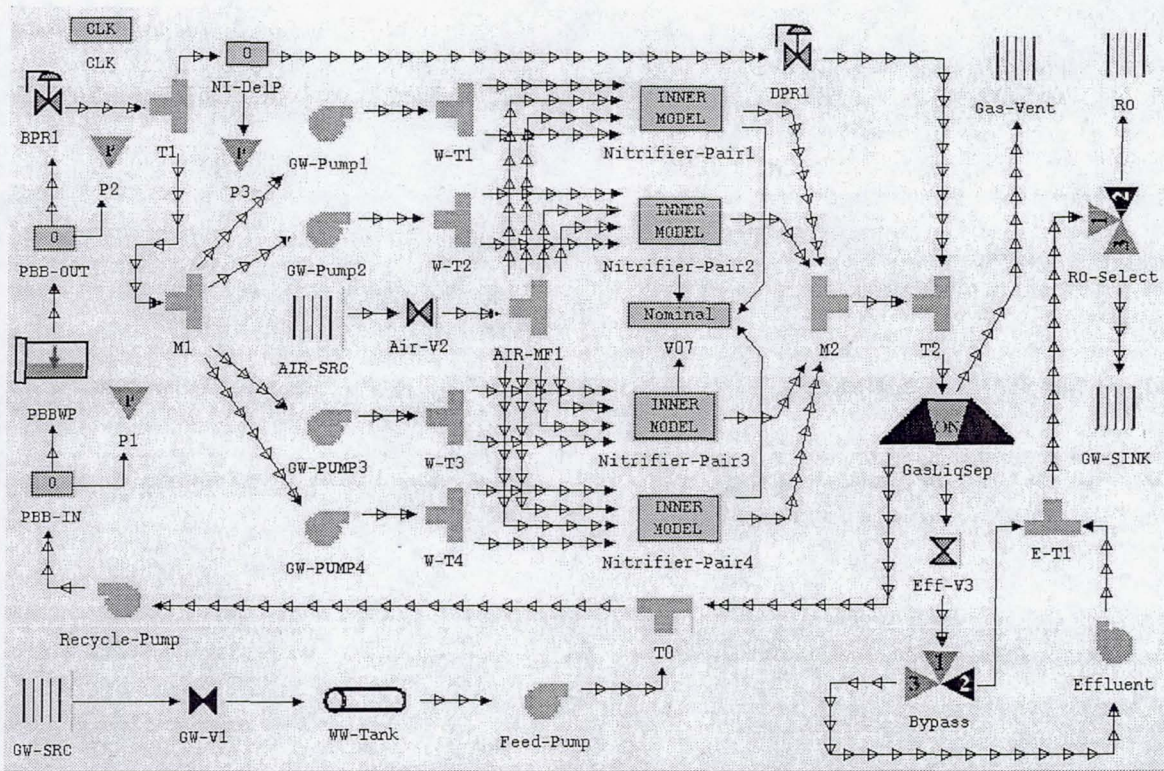


Figure 2. The Biological Water Processor Model

The primary control activity of the task sequencer is the maintenance of the water level of the GLS within a specified range by regulating the wastewater feed and effluent outflow rates. Should the GLS water level become too high or too low, the software responds by putting the system in a safe mode, by shutting down the BWP.

CONFIG CAPABILITIES

The CONFIG hybrid simulator has been used to model the BWP. CONFIG extends discrete event simulation with capabilities for approximate and qualitative modeling of continuous system behavior [8]. The tool supports investigation of the sequencing of system events in fast scenario-based simulation of operations, and is well suited to analysis of hazards and effects of failure modes and fault tolerance.

A communications interface may be established between a CONFIG hardware simulation and external control software applications so that the asynchronous interactions between the simulation and the software may be observed and evaluated. CONFIG also supports asynchronous interaction with human operators by means of instrument displays that can be custom-built to resemble simplified versions of the physical system displays and controls. For such interactive tests involving external agents, the simulation is run in a real-time approximation mode.

To evaluate operation concepts prior to implementation of actual control software, or to evaluate manual

procedures that unfold over very long periods of time, the procedures themselves may be modeled as components of the larger system model in such a way that the procedure models and hardware component models interact synchronously, thus permitting the simulation to be run at speeds many times faster than real time. CONFIG capabilities for scripting and for modeling activities (control, procedures, schedules) can be used for early dynamic analysis of operational problems.

The discrete event simulation base provides a framework for causal modeling of states and outcomes, and for specifying transition functions that are internal or triggered by external inputs. The continuous time base of discrete event simulation [16] supports both event-stepped time advances and discrete-time-step approaches to continuous simulation. In CONFIG, a discrete-time-step approach, with either linear or exponential approximation [15], can be used to periodically update continuous variables in a component. In discrete event simulation, generation of events and time-advances can be random, supporting probabilistic analysis. CONFIG has been used for deterministic analysis of specific state configurations and inputs.

Discrete event models of systems are composed of coupled interacting component models. In CONFIG, components have multiple behavior modes. Each mode has state equations that generate behavioral effects, and conditions that govern mode transitions. System behavior emerges from the coupled behavior of the components. In CONFIG, the model structure can be

"recomposed" during a simulation as the direction and activation of the couplings changes. Parts of the model are activated or deactivated as operating modes of components change or closed off areas of the system are brought into the working system configuration. Recently, the coupled model approach has been used in a capability for selecting and mixing of simple and complex subsystem models in simulation experiments. Selecting versions of subsystem models can focus and speed up simulations.

CONFIG OBJECT-ORIENTED LIBRARIES

Object-oriented classes support component modeling. Figure 3 shows the hierarchy of component classes for the BWP system model. Subclasses of devices inherit

the data structures and behavior descriptions of their super classes. The inherited structure and behavior are then modified or extended for the subclass.

CONFIG activity models support modeling of controllers, human operator procedures, actions and schedules. A CONFIG library contains a class hierarchy of activities similar to the device class hierarchy. Data structures and behavior descriptions are defined in nearly identical ways for both devices and activities.

Parts of system models are selected and instantiated from palettes of device and activity classes (see Figure 4). Parts are connected to each other on a graphical canvas (see Figure 2) by means of device and activity "relations."

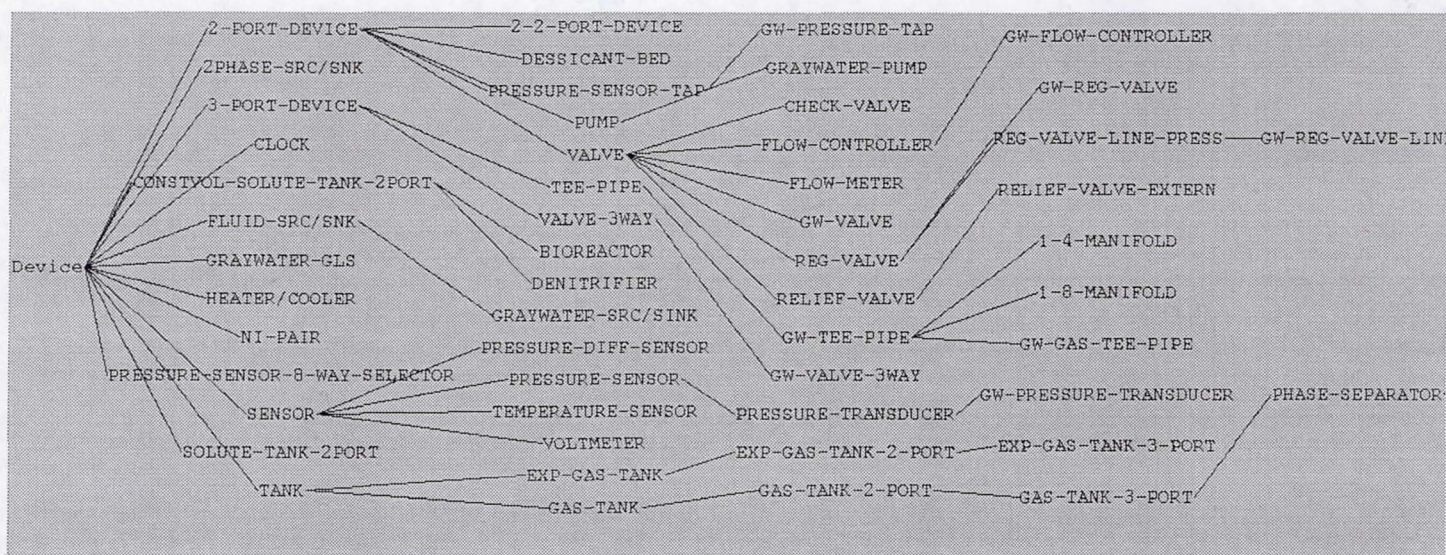


Figure 3. Device Classes available in CONFIG including those designed for the BWP system simulator

INNER MODELS

Composite model parts can be constructed and copied. Such structures are referred to as "inner models." An inner model may be constructed first as a top-level "stand-alone" model for testing its behavior in simulation. After the model has been verified, copies can be made of the top-level model and spliced into another model. Inner models may be nested to any arbitrary depth in a model parts hierarchy. In CONFIG, a set of one or more "instrument panels" for display and manual setting of model variables may be constructed. An instrument panel for the BWP is shown in Figure 5. Each time a model is duplicated, a copy of its instrument panels is also automatically generated and associated with the model copy.

The inner model capability shortens model development time. It also makes the model's graphical displays more manageable during simulations; the graphical inner

model displays and their instrument panels may be opened and closed as needed.

The BWP Nitrifier model uses the inner model representation because it can be partitioned into four identical subassemblies. Each of the Nitrifier tube pair subassemblies in the BWP is a copy of the same original model, and each includes its own instrument panel.

PROCESS DESCRIPTION LANGUAGE

The CONFIG process description language is designed to specify data-driven changes in device states during simulations. For example, consider a "Gas-Transfer-Device" with an Input that receives a mixture of carbon dioxide (CO_2) and hydrogen (H_2) gases and merely propagates the mixture of the gases through an Output to some device downstream. Assuming that the gases

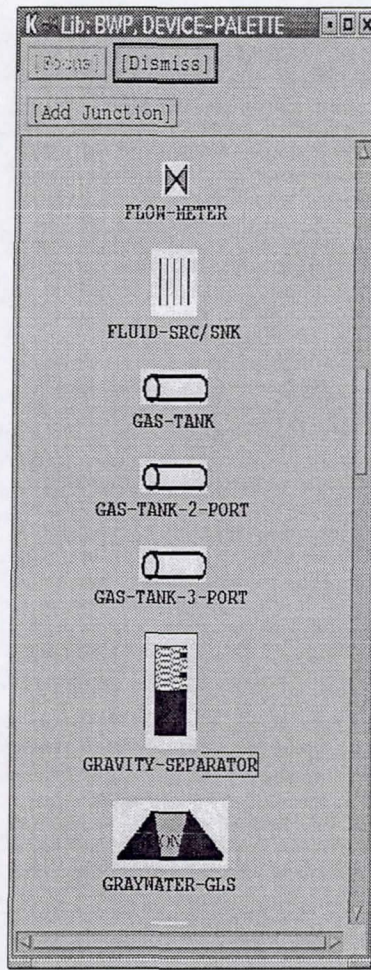


Figure 4. Section of the WRS Device Palette.

will always be some mixture of CO_2 and H_2 , the propagation behavior could be expressed in two statements:

```
(<- output:co2-fraction input:co2-fraction)
```

```
(<- output:h2-fraction input:h2-fraction)
```

The Input and Output would be represented by data structures referred to as "variable clusters." In this case, both the Input and Output variable clusters would be instances of the same class of variable cluster, defined to contain the two variables *h2-fraction* and *co2-fraction*. Whenever the relative proportions of CO_2 and H_2 changed in the course of a simulation, the process would be activated, setting the Output proportions of both gases to their new Input proportions.

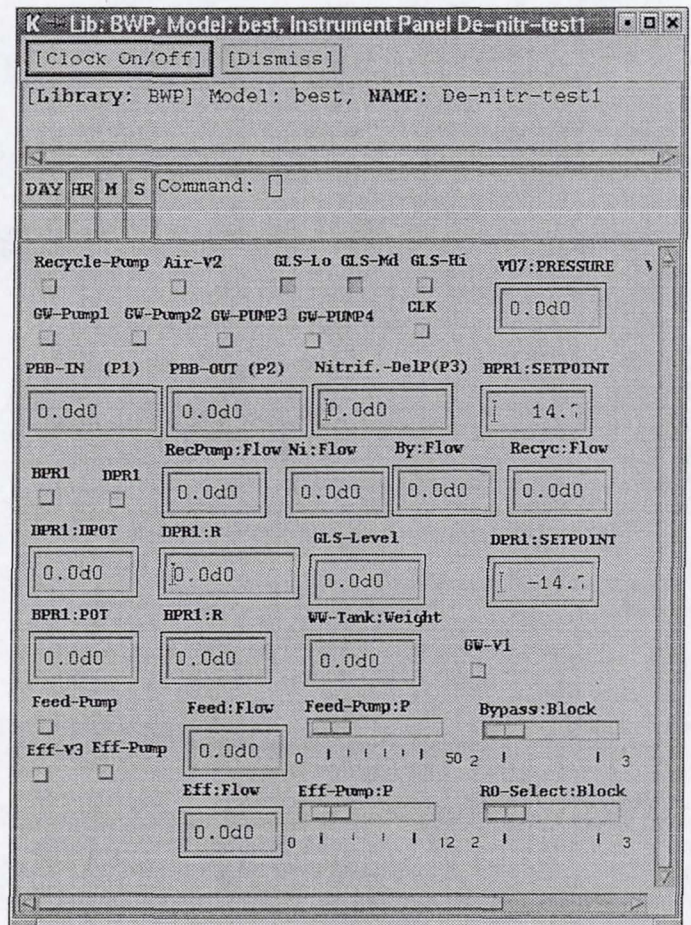


Figure 5. Instrument Panel for the WRS Simulation

PACKET OPERATIONS FOR MIXTURES

Such a general-purpose device should not be limited to transporting only two gases. It would not be sensible to include variables to accommodate all conceivable types of gases that a device of this type might transport. Therefore, CONFIG supports a more abstract form of statement for sets of similar entities such as gases. The two statements above can be replaced with the single statement:

```
(<- [output:gases mass-fraction]
    [input:gases mass-fraction])
```

The *gases* variable, rather than holding a single primitive value, holds a list of data structures called "packets," each with a "Packet-ID" and a *mass-fraction* variable. The Packet-ID of each packet would be the chemical symbol for a specific gas (e.g., CO_2 , H_2 , H_2O or N_2). The process is activated not only when the proportions of gases change, but also whenever a new chemical species of gas is propagated to the device Input.

More complex numerical operations on sets of attributes may be described using packet operations, which constitute a kind of vector processing. For example, Eulerian integration of the mass of each type of gas

contained in a two-terminal storage tank could be based on the Input and Output flow rates and on the fraction of each type of gas in the flows. The INTEGRATE operator is one of the primary means used by CONFIG to represent continuous processes in discrete event simulation. The general form of an INTEGRATE statement is:

```
(<- integrated-variable (INTEGRATE integrated-variable
                        rate -expression time-interval))
```

Changes in the rate of flow between updates are accounted for in the increment added to or subtracted from the *integrated-variable* at each regular update.

For a storage tank, the mass update statement could be written as:

```
(<- [contents:gases mass]
    (INTEGRATE [contents:gases mass]
      ( +      (* [input:gases mass-fraction]
                  in:total-flow)
        (* [output:gases mass-fraction]
          out:total-flow))
      contents:time-step))
```

Here, the Input and Output variable clusters would be defined for the Tank device as for the simpler Gas-Transfer-Device, but the Tank would also have a third variable cluster named "Contents." The Contents variable cluster would also contain a slot named "gases" in which packets are stored, but these packets would be of a different class than those in the *gases* slot of the Input and Output variable clusters. Each packet in which the mass data is stored for a particular type of gas would be associated with the packets having the same Packet-ID in the Input and Output variable clusters' *gases* slot. The increment added to the mass of each type of gas over the time interval, *time-step*, is equal to the *time-step* multiplied by the sum of the *total-flow* times the *mass-fraction* for the gas at the two respective flow terminals. The CONFIG simulator updates the mass of each gas every time the simulation clock advances by an amount equal to *time-step*.

REPRESENTATION OF SOLUTES

Water processing models requires a rich representation of the process fluids. The fluid streams are multi-component liquids and gases with dissolved and suspended solids. Packet operations are used for vector processing of solutes and gases. Physico-chemical water processing generates and consumes these components and shifts them between the liquid and gas phases.

In a vessel, fluid composition can be represented several ways. Material may be tracked on a basis of mass, moles or volume. Each may be tracked as simple values (e.g. 5 grams of CO₂) or as fractions (12 molar-

percent CO₂). For streams, such as a constant-composition inflow of wastewater, a component-fraction basis is most convenient. Therefore we represent flows (inflows, outflows and the flows between components) on a mass-fraction basis. Each fluid stream has both gas and liquid mass-flow rates and mass fractions for each phase. Suspended and dissolved solids are represented the same as with fluid components. This approach is straightforwardly extensible to multiple liquid phases, e.g. oil / water mixtures.

For vessels, all components are tracked on a simple weight basis, separately for the liquid and gas phases. This leads to the simplest and most easily understood equations. A *generation* variable tracks the rate of appearance of the component in the phase, in units of *mass / time*, providing uniform mass balance equations for all components, phases and vessels.

Actual test-bed data may be recorded on other bases. For example, the concentration of the solute composition of the bioreactor is recorded as parts per million (ppm), not the net amount of solute in the vessel. The governing equations also have their own forms. For example, the bioreactor's removal rate for NH₃ is normally written in terms of concentrations, and the gas phase is recorded as a volume or molar fraction. The mass compositions of the simulation are translated into other bases as required for display.

SIMULATION OF FLUID AND ELECTRICAL FLOWS

CONFIG simulates complex flow regimes – multi-component mixtures, mixed-phase flows, and variations in pressure, temperature and fluid density. A global flow and pressure/potential analysis capability, the Flow-Path Management Module (FPMM) tracks dynamic changes in system configurations and model structure during simulation [7]. The same underlying facility, using graph analysis and linear circuit analysis, supports computation of flow rates and the associated potential drops across resistive elements for both fluid and electrical flow at the abstracted level of effort, flow and resistance. This facility computes "static potentials" generated by flows at points where they are in contact with blockages such as closed valves. The "dynamic potentials" across resistive elements carrying flow in a linear circuit are completely determined at any point in time by the magnitudes of resistances and efforts in the circuit and are, in effect, state variables. The distribution of static potentials, however, is also strongly dependent upon the sequence of operations that led to the current state of the system flows (e.g., the order in which valve and pumps are operated). The dependence of the values and distributions of static potentials upon the history of system operations can make them difficult for human operators to understand and for the system to anticipate. By tracking the changes in static potentials with system reconfiguration events as well as with changes in system efforts and resistances, serious problems may be

uncovered that would otherwise go unnoticed. In a hydraulic system, for example, simulations could reveal that two sequences of operations lead to the same nominal system flows and dynamic potentials, but one of them may also produce an undesirable distribution of static potentials with effects such as the unintended opening of a relief valve.

In each simulation step, both local and global calculations produce the system behavior. Locally, the component behavior is calculated from its inputs and its state. Externally triggered and internally driven transitions result in changes in time-delayed value assignments. The capability to alternate global analysis of flow paths with local computation of events during simulation supports predicting indirect global effects of local operations.

FAILURE MODELING

CONFIG models and simulations can represent the effects of failures and other problems. CONFIG provides capabilities to model failures of configuration, input, capacity, performance, control and operations. Certain types of failures and problem inputs may trigger discrete changes in behavior. These can change the control regime, the system configuration, or the capacity or performance of a system component. For example, discrete changes can result from bursts, shorts, errors or uncommanded actions. Changes in control, capacity or performance can also be continuous, gradual and nonlinear. These types of changes can result from buildups, wear, leaks and drifts. Some failures such as stuck components result in component states that cannot change when they should. Some failures involve random variation in measurements or inputs. For any failure, the magnitude of the effect may be determined by the magnitude of the failure. Component failures can produce problem inputs elsewhere in the system. These result in cascades of failures and off-nominal component states. In complex systems, these cascades can be difficult to anticipate during design.

SOFTWARE INTERFACE

The approach to interfacing simulations to control software minimizes the need for a special "simulation version" of the control software; the software is, in effect, "plugged in" to the simulation via local area network connections in a manner analogous to the way it is to be interfaced to the real hardware. Well-designed software has clearly defined communications interfaces to the external world, so this approach has generally proven to be quite straightforward. The simulation and control software communicate over the local area network, with a general-purpose message server application as an intermediary.

The software data formats and the data channels to the hardware actuators and sensors are used to write a

simulation software interface specification. This specification declares the correspondences between the data channels and device variables in the CONFIG model of the system. Conversion functions may be given as part of the interface specification so that the model itself can be constructed independently of the details of the software data communications.

Prior to interactive simulation, the simulation software interface specification is automatically converted by CONFIG. Message-handlers process command input from the control software, and message constructors marshal sensor data to be sent to the control software at regular intervals (or whenever a sensor value has changed).

BWP SIMULATION

THE MODEL

The purpose of the model is to dynamically simulate the operation of the BWP including the behavior of the state variables, fluid flows and solute concentrations. Decisions made in constructing the model of the BWP were guided primarily by what was needed to provide the appropriate sensor data to the 3T control software in response to its commands to the hardware.

Construction of device models was guided by engineering documents describing the behavior of system components. Engineering schematics guided the flow connections between the model components. Sources of information included Project Planning Documents [4] and Schematics from the CTSD, Life Support and Habitability Test Branch at JSC [5] and information about the automatic control system [1]. The scope of the model did not include the inoculation phases (initial growth of microorganisms), since it was thought to be of less interest to the software developers.

Figure 2 and Figure 6 show a CONFIG-generated dynamic schematic of the BWP in a "stand-alone, drain" mode. The arrowheads along the lines connecting components show the directions of flows at any given point in the simulation and are redrawn by the CONFIG graphical user interface as system configuration events such as valve operations affect system flows.

The Nitrifier tubes were modeled using Inner Models, by creating a subassembly model and using copies in the higher-level system model. The Inner Model of a Nitrifier consists of two parallel paths, where each path consists of an incoming air line, an airflow controller, the incoming wastewater pipe, a T-pipe where the air and the water join together and a Nitrifier tube. After the Nitrifier tube the two water/air paths join together with a T pipe. Figure 6 shows one Nitrifier pair, for an Inner Model depicted in Figure 2.

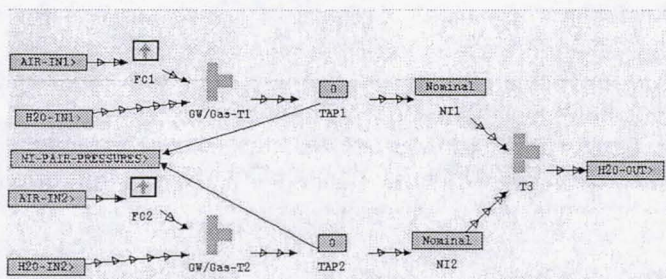


Figure 6. One Nitrifier pair in the BWP Model

The phenomena represented in the BWP model include:

- A linear circuit approximation of fluid flow
- Biological conversion of organic carbons and nitrites to carbon dioxide, oxygen and nitrogen
- Biological conversion of ammonium and air into nitrites, nitrates and nitrogen
- Separation of gases from liquids
- System-wide mass balances for fluids, solutes and gases, taking into account physical and chemical conversion processes

Packet operations were used to represent the quantitative changes in the small concentrations of solute impurities in the wastewater at various points in the BWP processing loop. The model includes three of these solutes: total organic carbons, ammonia, and nitrites and nitrates.

The PBBWP was modeled as a liquid container of volume equal to its dimensions minus the volume of the ceramic saddles. The consumption of total organic carbons (TOC) was modeled as a simple linear function of TOC concentration in the inflowing water. The effect on flows of the pressure differential across the PBBWP due to the weight of the contained water was also represented in the model.

Each of the two tubes, in the four Nitrifier inner models, was modeled as a container with a gas and a liquid mixture. The consumption of ammonia in the Nitrifier and the production of nitrites and nitrates were also modeled as a simple linear function.

The GLS was modeled as a tank with a mixture of liquid and gas at atmospheric pressure. Because the inflowing gases to the GLS are vented, the gas content of the device does not change during normal operations. The changes in the mass of the accumulated liquid are calculated from the sum of all the flow rates entering and leaving the device's three ports.

There also are various other devices that were required to model the system. The water pumps are represented as sources of effort and can be controlled by the user from an instrument panel or by the autonomous 3T software. Regulators and flow controllers are modeled as variable resistance devices that maintain pressures

and flows respectively at the desired set-point level. The 2-way and 3-way valves that determine the directions of system flows can be controlled by CONFIG or by the autonomous control system. Other devices include manifolds, the wastewater tank, and the wastewater source and air sources and sinks.

Water is extracted from the system by either the effluent pump or the RO injector pump. The only dependence on the rest of the WRS is the rate of extraction of water by either the effluent pump or the RO injector pump.

TESTING AND DEMONSTRATION

The primary objective of the model was to simulate the BWP when it is in a steady state mode ("stand-alone, drain"), and some of the departures from the steady state that may occur during normal operations. During model verification testing prior to interfacing the model to the 3T autonomous control software, the model was controlled manually from software instrument panels containing the necessary switches and sensor readings (Figure 5).

After verification testing of the model, interactive testing was performed with the 3T software interfaced to the model. The 3T control software was not revised to make it compatible with the model. Except for the removal of platform-specific user-interface code, no modifications were made to the 3T software used to control the actual BWP hardware.

In nominal operations, the BWP extracts water from the wastewater tank while partially purified water is simultaneously extracted from the system by the effluent pump. Solute concentration, using our simple representations of microbe metabolism, behaves as expected, reducing organic carbons in the PBBWP and ammonia in the Nitrifier. Nitrates are also produced in the Nitrifier, and then consumed in the PBBWP.

While running the simulated system, the 3T control software reacts correctly to the changing level of the GLS by increasing or decreasing the rate of the effluent pump to maintain a desired level. Also, the control software shuts the system down as required when the water level becomes too high or too low.

One of the side effects of nominal operations of the WBP is a buildup of microorganisms along the walls of the Nitrifier tubes, clogging the tubes and increasing the pressure at the Nitrifier inflow ports. The normal procedure to handle this process is to increase the airflow in the affected Nitrifier tube to slough off the microorganisms until a desired population is achieved. Microbial buildup in a tube can be represented as increased resistance. In the model, the 8-way sensor labeled V07 in Figure 4 detects the pressure increase in a tube that has become clogged. A new version of the 3T software that includes procedures for managing

clogged tubes has not yet been interfaced to the simulation, but the CONFIG model can be used to test this 3T function in the future.

CONCLUSION

The CONFIG tool supports developing useful models of life support systems for operability analysis. Object-oriented and graphical support and hierarchical models make the modeling and simulation effort reasonable. Capabilities to model fluid mixtures and system-wide reconfigurations make the simulation useful for identifying operability problems. The same control software that is used to control the actual hardware can control the simulated hardware. Simulations can be used to systematically simulate system perturbations and deviations from operating intentions, instead of having to recreate these conditions with the real system. "User/control-software" interaction issues can be investigated, such as the control software interaction with the system operators and interactions of automatically controlled life support systems with ground flight control.

FUTURE WORK

Additional CONFIG capabilities are needed to model biological processes that are dependent on the concentration of solutes in the system. Some of the failures that have occurred in the WRS laboratory have been related to the sensitivity of bacteria colonies to environmental factors. These can only be simulated when a more realistic model for the reduction of solutes and its influence on the microorganism populations has been implemented.

A scenario server is under development. The BWP simulation can serve as a virtual device for developing other software. The HCAAST project (Human Centered Autonomous and Assistant Systems Testbed) is developing tools to provide graduated automated responses to off-nominal conditions aboard spacecraft [11]. When fully deployed, inputs to these tools will include telemetry data streams. During development and demonstration, the BWP simulation will provide test data streams. Development involves testing the same scenario repeatedly. Rather than running the BWP model and its control software for each development run, the scenario server will store sets of data streams that have been generated by simulation runs and provide them on demand. This will provide more flexibility for restarting scenarios in mid-simulation, and for developing in computing environments where running the full simulation is inconvenient.

Recently, we have begun collaborating with developers of Brahms, a tool for modeling human work and activities [3]. Integrated use of these tools can enhance analysis of operational aspects of designs. We are also beginning

a project that will use CONFIG in a system to support model-based hazard analysis for complex systems [10].

ACKNOWLEDGMENTS

TBD

REFERENCES

1. Bonasso, R. P. 2001. Intelligent control of a NASA advanced water recovery system. In *Proceedings of the Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space* (I-SAIRAS 2001).
2. Bonasso, R. P., R. J. Firby, E. Gat, D. Kotenkamp, D. P. Miller, M. Slack. 1997. Experiences with an architecture for intelligent, reactive agents, *Journal of Experimental and Theoretical Artificial Intelligence*, 9:237-256.
3. Clancey, W. J., P. Sachs, M. Sierhuis, R. van Hoof. 1998. Brahms: simulating practice for work systems design. *International Journal of Human-Computer Studies* 49: 831-865.
4. CTSD. 2000. Advanced Water Recovery System (WRS) Integrated Test Plan. Crew and Thermal Systems Division, Life Support and Habitability Systems Branch (EC3), Johnson Space Center, NASA, Houston, Texas, February 11, 2000.
5. ECLSS. 2000. Advanced WRS ECLSS Integrated Schematic 2/6/2000, CTSD internal document.
6. Leveson, N. G. 1995. *Safeware*. Addison-Wesley.
7. Malin J. T. and L. D. Fleming. 1998. Global Qualitative Flow-path Modeling for Local State Determination in Simulation and Analysis. U.S. Patent 5,732,192.
8. Malin, J. T., and L. Fleming 1999. Enhancing Discrete Event Simulation by Integrating Continuous Models. In *Hybrid Systems and AI. Working Notes for AAAI 1999 Spring Symposium Series* (Stanford, CA, March 22-24). AAAI, Menlo Park, CA.
9. Malin, J. T., L. Fleming, T. R. Hatfield. 1998. Interactive Simulation-Based Testing of Product Gas Transfer Integrated Monitoring and Control Software for the Lunar Mars Life Support Phase III Test. In *Proceedings of SAE 28th International Conference on Environmental Systems*. SAE Paper No. 981769.
10. Malin J. T., L. Fleming, D. R. Throop. 2002. Predicting System Accidents with Model Analysis during Hybrid Simulation. In *Proceedings Advanced Simulation Technologies Conference*, San Diego, CA.
11. Malin J. T., K. Johnson, A. Molin, C. Thronesbery, D. Schreckenghost. 2002. Integrated Tools for Mission Operations Teams and Software Agents. In *Proceedings of the 2002 IEEE Aerospace Conference* (CD). IEEE Aerospace Conferences, Manhattan Beach CA.
12. Osburg, J.; R. Bertrand; E. Messerschmid. 1998. MELISSA - A Graphical Environment for Life-

Support Systems Simulation. In *Proceedings of SAE 28th International Conference on Environmental Systems*. SAE Paper No. 981754.

13. Osburg, J., and E. Messerschmid. 2000. Integrated Simulation of Synergistic Space Station Subsystems During the Conceptual Design Phase. In *Proceedings of SAE 30th International Conference on Environmental Systems*. SAE Paper No. 00ICES-368.
14. Pin, O. 2001. Using EcosimPro for Thermal and Thermo-Hydraulic Analysis. In *Proceedings of SAE 31st International Conference on Environmental Systems*. SAE Paper No. 2001-01-2409.
15. Throop, D. R., J. T. Malin, L. Fleming. 2001. Automated Incremental Design FMEA. In *IEEE Aerospace Conference Proceedings 2001*, (CD). IEEE Aerospace Conferences, Manhattan Beach CA.
16. Zeigler, B. P. 1976. *Theory of Modeling and Simulation*. New York: Wiley.

CONTACT

Jane T. Malin (malin@jsc.nasa.gov) is Technical Assistant in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center, where she has led artificial intelligence research projects for 17 years. She has led development of the CONFIG simulation tool for evaluating intelligent software for operation of space systems. She has led research on intelligent user interface and intelligent systems for control and fault management of space

systems. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan.

Luis Flores (luis.flores@lmco.com) is a systems software engineer supporting the NASA Johnson Space Center Automation, Robotics, and Simulation Division since 1985. He has been involved in design and development of software using knowledge-based, intelligent control and computer simulation tools for space systems applications. His 1967 Ph.D. degree in Physics is from Texas A&M University.

Land D. Fleming (land.d.fleming1@jsc.nasa.gov) is a Computer Systems Specialist supporting the NASA Johnson Space Center Automation, Robotics, and Simulation Division since 1990. He has been involved in both the development of computer simulation tools and their application to space systems. His 1987 M. S. in Computer Science is from De Paul University.

David R. Throop (David.Throop@SW.Boeing.com) has been an Artificial Intelligence Specialist with The Boeing Company since 1992. He provides engineering software support in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center. He oversaw development of FMEA modeling software and its use for the International Space Station. His 1992 Ph.D. in Computer Science is from the University of Texas, with a dissertation on Model Based Diagnosis. His 1979 Bachelors of Chemical Engineering is from Georgia Tech.