NASA Technical Memorandum 102852

Automated Simulation as Part of a Design Workstation

E. Cantwell, T. Shenk, P. Robinson, and R. Upadhye

August 1990

National Aeronautics and
Space Administration

NASA Technical Memorandum 102852

Automated Simulation as Part of a Design Workstation

E. Cantwell, Ames Research Center, Moffett Field, California
T. Shenk and P. Robinson, Recom Technologies, Inc., San Jose, California
R. Upadhye, Lawrence Livermore National Laboratory, Livermore, California

August 1990

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

Automated Simulation as Part of a Design Workstation

E. Cantwell
NASA Ames Research Center

T. Shenk and P. Robinson
Recom Technologies, Inc.

and

R.Upadhye
Lawrence Livermore National Laboratory

ABSTRACT

A development project for a design workstation for advanced life-support systems (called the DAWN Project, for Design Assistant Workstation), incorporating qualitative simulation, required the implementation of a useful qualitative simulation capability and the integration of qualitative and quantitative simulations such that simulation capabilities are maximized without duplication. The reason is that to produce design solutions to a system goal, the behavior of the system in both a steady and perturbed state must be represented. We report here on the Qualitative Simulation Tool (QST), on an expert-system-like model building and simulation interface toll called ScratchPad (SP), and on the integration of QST and SP with more conventional, commercially available simulation packages now being applied in the evaluation of life-support system processes and components.

A VARIETY OF LEVELS OF NUMERICAL PROCESSING will be necessary in the design of advanced life-support systems for planetary exploration. It will frequently be the case that empirical relations alone will not be sufficient to approach all design issues and to integrate life-support system designs over the range of technologies that will be necessary to produce closed-loop systems. In addition, it is clear that a computational design tool will need a broad range of simulation capabilities with which to determine the system behaviors of alternative scenarios or specific designs. The Design Assistant Workstation (DAWN) project was undertaken to meet these computational tool requirements. Initial attempts to develop a design workstation for life-support systems concentrated on a preliminary integration of artificial intelligence (AI) techniques (expert systems) with conventional quantitative modeling tools for process-level design of physical/ chemical life-support systems (some examples of these are ASPEN+, PROCESS, Chemshare).* AI-based software techniques allow better definition of models, assist a user with complex codes, and ultimately assist in the capture of design knowledge. Previous reports on this preliminary work have been issued (1).**

*Our use of ASPEN+ as the chemical simulator does not imply an endorsement of this commercial numerical processing package. DAWN can be configured to interface with any conventional code desired.

**Numbers in parentheses designate references at end of paper.

During the past year, emphasis has been solely on DAWN's simulation capabilities. Evaluation of a first prototype (1) led to the conclusion that a fairly complete qualitative, as well as quantitative simulation, capability was needed if we were to be able to truly address implementation needs for actual design problems. Qualitative models are often simpler to understand and easier to use than mathematical models, yet they can retain all important relationships between parameters and behavior states. Qualitative simulations may be run with incomplete information, and thus may be executable much earlier in the design process. Yet quantitative modeling will be necessary when exact numbers are required, or where qualitative simulation yields ambiguous results. The development of a qualitative simulator will be addressed here, as well as the integration of qualitative and quantitative simulation components and a knowledge-based support system for creating and running simulation models with complex chemical process simulation codes.

The idea of applying AI in developing a "front-end" for numerical codes is not new (2). However, this application requires not only information about the inputs for the codes but information about the physics and chemistry we are attempting to simulate as well. We have developed front-end concepts for simulators like ASPEN+ (see Fig. 2), and have also developed and implemented concepts for "back-ends" to these simulators. Both of these are embodied in a software module called Scratchpad (SP), which will be discussed in detail in this paper.

The idea of using qualitative simulation as a stand-alone predictor of system behavior as well as an adjunct to numerical modeling is also not new (3). We have developed this concept from the chemical engineering point of view, integrating the qualitative and quantitative in such a way so that unambiguous predictions can be made qualitatively and so that parametric sensitivities and influences can be learned—that is, retained by the system for future use. This software module is called QST (Qualitative Simulation Tool) and will also be discussed herein.

SP, QST, and the simulators have been integrated as outlined in Figure 1. A future paper will detail the underlying implementation of each module and their integration. The purpose of this paper is to discuss the concepts, architectural strategies, and results of our work in developing a simulation component for the DAWN project. First, the ScratchPad component is discussed, then the QST component. As can be seen from Figure 1, in its current configuration, the user controls simulation for the purpose of analyzing design goals. Future development will allow DAWN to reason independently with simulations of scenarios and come back to the user with overall evaluations and recommendations. Other follow-on work is discussed in the next to last section of the paper, with conclusions following that.

THE SCRATCHPAD COMPONENT

The approach we have used for developing the ScratchPad (SP) component was that it should supplement or complement the simulation rather than duplicate a simulation method. Therefore, the following guidelines are used in deciding what heuristics to include in the SP:

1) Identify processes used in life-support applications in which a simulation has given incorrect or misleading results. These results have been traced to "bugs" specific to some simulation programs, bugs in the model constructed by a naive user, and inherent limitations of the methods used by the simulations.

2) Identify, based on the above, heuristics incorporating expert knowledge for performing simple calculations or rule-evaluations as a check on a model or on simulation results. Figure 2 diagrams the DAWN architecture. The components of this architecture encompassed by SP are model libraries, model validation and comparison of results, an understanding of model units, and alarms and warnings and unit solutions.

Model libraries consist of simple models with known results for verifying more complex models, complex models of certain relevant chemical processes like incineration or supercritical oxidation, and life-support system specific data (such as waste sludge compositions) for constructing models' feed streams (Fig. 3).

Model validation and comparison of results consist of qualitative or quantitative comparisons between results from SP, QST, simple models, or simulators. Most of these comparisons are made at the level of solutions to individual units or streams (Fig. 4), though comparisons over a range of states can also be made (Fig. 5). Alarms can be sounded when differences in results exceed a user-specified threshold.

Model units are flow-sheet building blocks like stoichiometric reactor, flash, and feed stream. SP uses groups of simple rules to solve things like vapor/liquid equilibrium, and uses the same input parameters as does the simulator (in our case, mostly ASPEN+) specification.

Alarms and warnings are mechanisms which allow SP to predict or detect when the model/simulation may give incorrect or misleading results (Fig. 6). This raises an alarm to the user. Alarms implemented thus far can be subdivided into three classes: 1) violations of first principles, for example, the phase rule applied to vapor/liquid equilibrium; 2) mathematical artifacts of the simulation implementation, for example, a flash calculation may become unstable if the specified temperature is at or near the boiling point of a component; and 3) sanity checks, for example, checking that proportionately less of the higher-boiling-point component in a flash operation is vaporized than of a lower-boiling-point component.

The following is an example, in which heuristics is used to check vapor fraction for a flash operation:

Problem: Calculate the vapor fraction in a flash operation.

Assumption: Vapor fraction is more a function of temperature (rather than heat duty); that is, the system is a "wide boiler."

Heuristics: If the liquid solution can be classified as *ideal* or even *slightly nonideal*, we can scale the temperature specification to estimate the vapor fraction.

Rule: Vapor.fraction = $(T_{flash} - T_{bubble})/(T_{dew} - T_{bubble})$. This will give us an approximate result to compare with the simulations. Furthermore, we can attach a measure of belief to the vapor fraction based on the probability the system can be classified as a "wide boiler."

How SP works: SP classifies a solution as *ideal*, *slightly nonideal*, etc. by referencing a set of miscibility rules. These predict simple binary interactions between classes of compounds. An example rule for this is

If H_2O and (compound in alcohol class) are
both in liquid phase and $MW_{alcohol} < 100$
THEN alcohol miscible with H_2O and
attractive interaction of alcohol/H_2O
because of "H bonding."

SP also uses heuristics in the determination of T_{bubble} and T_{dew}. Compounds classified as *very heavy* are excluded from the T_{dew} calculation. Including such compounds results in abnormally small vapor fractions when the temperature is scaled. Similarly, *very light* compounds are excluded from T_{bubble} calculations. A compound is classified as heavy or light by a correlation with its vapor pressure.

Sanity check: A sanity check for this example, to verify that the results and assumptions of SP are consistent, is

IF solution is *ideal* or *slightly nonideal* and binary interaction are attractive
THEN T_{bubble} > boiling point of minimum boiling liquid component
ELSE "Inconsistent"

THE QUALITATIVE SIMULATION COMPONENT

There are a number of advantages to using qualitative simulation over numeric simulation. Qualitative models can be used when numeric information is not available or when the system being modeled is not understood well enough to have complete, known numeric equations for modeling it. Qualitative models can be used to explain what is happening within a system, something that cannot be easily done with a numeric simulation. In addition, if the system being modeled is subject to change, qualitative simulations are easily modified. Qualitative and quantitative modeling components need to be well integrated, however, so that quantitative information can be used when available, but the system still allowed to fall back gracefully on qualitative reasoning in the presence of uncertainty or missing information.

QST has been developed to begin to address these issues. It is structured to model the qualitative reasoning processes used by chemical engineers. Chemical engineers reason qualitatively about the trends of a process within and outside of defined operational limits. If influences are competing, qualitative reasoning is often insufficient, and the chemical engineer will resort to numerical calculations to resolve ambiguities. As a chemical engineer gains experience with a process, the sensitivities of that process to competing influences are learned (4).

QST is a qualitative chemical process simulator. It uses the same model units as the conventional quantitative process simulators such as ASPEN+. The main difference between these is that QST uses qualitative information (trends: increasing, decreasing or steady) as opposed to quantitative information (numeric values: +2.5, −3, or 0). Trends reflect the direction of process variables toward limits. When a process is not at equilibrium, an imbalance of forces or tendencies exists. Trends are a qualitative measure of this imbalance. Limits represent bounds on process variables. These limits may be natural, such as those due to the laws of physics; or they may arise out of external considerations such as safety (i.e., the temperature and pressure in a reactor must not exceed the design specifications). Although QST could be used to model either steady-state or transient behavior of processes, we will describe here the use of QST for steady-state process analysis in light of the fact that most commercially available simulation packages do not calculate transients.

Using the problem-solving strategies of a chemical engineer as a guideline, we have designed QST to handle varying degrees of model description. If a query to a model can be answered qualitatively, QST will provide an unambiguous prediction. If, however, ambiguities are generated in the qualitative simulation, the ambiguities must be resolved before simulation proceeds. Several levels of quantitative information are used to resolve ambiguities. Often, if the sensitivities of the influences are known, order-of-magnitude calculations are sufficient for providing unambiguous prediction. If all influences have similar sensitivities, however, then the magnitude of the influences is required as well. Consider a model of an ammonia reactor (Fig. 7). It has input streams (denoted by X) and output streams (Y), containing material and energy. Associated with this process are design variables (D) and operating variables (O). In this case, O defines pressure and temperature, and D defines an equilibrium reactor operating adiabatically. The qualitative state of this or any component defines a region of parameter space, with a steady state denoting a point within the parameter space. Given these definitions, an intrastate transition is a transition from a steady state to another steady state, both within the same qualitative state. Our attention is currently restricted to intrastate transitions, and this example illustrates that.

An *S_matrix* represents the behavior of a component in the neighborhood of a base steady state. Each element in the *S_matrix* represents the sensitivity of an output variable y_i to an input variable x_j, namely $\partial y_i/\partial x_j$. An example of a qualitative *S_matrix* for the ammonia reactor case is shown in Table 1. Qualitative process rules (5) were used to fill the *S_matrix*. We will show later that if the values (including magnitude) of the sensitivities are known, the *S_matrices* can be used to represent the behavior of the component quantitatively. Each row of the *S_matrix* defines the net effect of an output parameter with respect to all input perturbations. If, as an example using the ammonia reactor model, we wish to investigate how the output temperature varies with respect to a simultaneous increase in both input nitrogen flow and hydrogen flow, we can formalize this query to QST as follow: if $\partial f_1\_N_2 = +$ and $\partial f_1\_H_2 = +$ then $\partial T_2 = ?$ (where ∂f_1 is defined as "the change is amount of substance on the 1, or input side of the process"). Since the effects of these two perturbations were complementary, an unambiguous qualitative prediction of $\partial T_2 = -$ can be made using modified confluences (6).

In general, however, qualitative simulation is ambiguous. For the ammonia reactor example, simultaneous input perturbations of $\partial T_1 = +$ and $\partial f_1\_N_2 = +$ are competing (see Table 1). When an ambiguity is noted, domain-independent methods of qualitative simulation (5,6) split the prediction, failing to guarantee that the final results represent physically realizable behaviors. It is clear that the use of qualitative simulation can generate ambiguities, and that these ambiguities are not useful for the simulation, design, or operation of chemical processes. If there are n components in a flow sheet and if qualitative simulation predicts three possible behaviors for each component, the total set of final behaviors for the flow sheet could be $O(3^n)$. Given that the average flow sheet contains between 20 and 100 components, this combinatorial explosion is simply unacceptable. Therefore, when an ambiguity is detected, it must be resolved immediately.

To resolve competing effects requires additional information. That additional information can come from numerical simulators. We have chosen to use sensitivities to resolve ambiguities. An example of this would be that one *could* obtain the net effects of an output variable with respect to multiple simultaneous input perturbations by performing a single quantitative simulation run that requires all values of input perturbations. This is useful, however, only in to answering the current query. If, instead, we run the quantitative simulator for each of the input perturbations, we can define the sensitivities of all of the output variables with respect to each individual input perturbation. The effect of this is that now QST has learned the individual sensitivities, and annotated the *S_matix* accordingly. This information can be used in answering future queries without having to resort to numerical simulations. Note that the *S_matrix* gets annotated as it is necessary. At some point, an *S_matrix* will become completely annotated, and no further quantitative simulations need be performed for that particular model. A detailed example of this theory is implemented, and discussed in Ref. 4. Table 2 shows what a partially annotated *S_matrix* for the ammonia reactor looks like. In the case of qualitatively competing influences from several input perturbations, we can now see which input perturbations the process is most sensitive to. The system then represents that information so that it can be used qualitatively, in future simulations.

IMPLEMENTATION AND FOLLOW-ON WORK

DAWN has been implemented on a Symbolics workstation, networked to a VAX 8800 on which the conventional process simulations are run. An example of the DAWN user screen is shown in Figure 8. The vast majority of our simulation work has been developed using MTK and XTK (7), software developed in the Information Sciences Division at Ames Research Center for qualitative representation of physical systems. We have conducted an end-user survey (8), for the purpose of identifying hardware and user interface needs on the part of the DAWN's potential user community. As a result of this, we have produced a report on porting the DAWN to an end-user workstation (9). Implementation on an end-user system will proceed later this year. An area that still has to be fully addressed is the expansion of the DAWN's knowledge bases with specific and sufficient domain knowledge about life-support systems.

There is considerable interest in developing advanced life-support system automation strategies that can allow for high degrees of autonomy (10). Model-based reasoning, utilizing both qualitative and quantitative simulation may provide high levels of autonomy for life-support systems, which operate such that relatively long time lags for control operations are tolerable. Model-based monitoring of sensors will allow for comparisons of sensor data with model outputs generated by reasoning from a qualitative description of expected system behavior. Such comparisons can then allow fault recognition (outputs do not match), fault diagnosis and automatic handling of sensor failures (generate model outputs that *do* match those of working sensors), and component control (which command values will generate expected behavior in the absence of a true fault). The simulation capabilities we have developed for life-support system design in DAWN may be straightforwardly expanded as a basis for autonomous control of advanced life-support systems. NASA has already demonstrated a prototype for model-based thermal subsystem control (11). Some of that work has already been applied in developing DAWN to date (MTK (7) was developed for that project). As advanced life-support system test hardware becomes available, we would like to begin adapting DAWN to test simple autonomous control strategies.

CONCLUSION

An integrated qualitative, rule-based and quantitative simulation/modeling system has been developed for the domain of advanced life-support systems. The interactions of qualitative and quantitative simulation have been examined and implemented such that each is used in an appropriate and complementary manner. This system, once it has been tested and used in real-world life-support system design, could be used as a primary modeling support tool in advanced life-support system analysis.

Two primary issues must be resolved before DAWN is made into a more generally available tool: 1) end-user workstation/ user interface needs must be fully identified, documented, and implemented; and 2) knowledge base expansion needs to be addressed in a formal manner. Longer term goals for the application of this research and development work for advanced life-support systems include its use as the basis for alternative design scenario evaluations and its expansion into the model component of model-based autonomous control strategies. The advantages of both of these to NASA can include increased productivity, both in design and operation phases, and in design knowledge capture for the future.

REFERENCES

1. Rudokas, M., Cantwell, E. R., Robinson, P., and Shenk, T., DAWN for Advanced Physical-Chemical Life-Support Systems. SAE Technical Paper 891481 July 1989.

2. Chalfan, K., A Knowledge System that Integrates Heterogeneous Software for a Design Application. AI Magazine, Vol. 7, No. 2, Summer 1986.

3. Karp, P. and Friedland, P., Coordinating the Use of Qualitative and Quantitative Knowledge in Declarative Device Modeling. Artificial Intelligence, Simulation and Modeling, edited by Widman et al., John Wiley & Sons, New York, 1989.

4. Robinson, P. and Upadhye, R., Learning to Resolve Competing Influences in a Qualitative Simulator. Submitted to Applied Mathematical Modelling, 1990.

5. Forbus, Ken, Qualitative Process Theory. Qualitative Reasoning About Physical Systems, MIT Press, Cambridge, MA, 1985.

6. deKleer, J. and Brown, J. A, Qualitative Physics Based on Confluence. Qualitative Reasoning about Physical Systems, MIT Press, Cambridge, MA, 1985.

7. Erikson, W. K. and Schwartz, M. R., MTK: An AI Tool for Model Based Reasoning. Proceedings of the 3rd Annual Conference on AI for Space Applications, Huntsville, AL, Nov 1987.

8. Robinson, P., Natarajan, R., and Cantwell, E., DAWN User Survey Report. DAWN Report 001, NASA Ames Research Center, Moffett Field, CA, Sept. 1989. (Available from the authors.)

9. Davis, E. and Cantwell, E., Delivering the Design Assistant Workstation. DAWN Report 002, NASA Ames Research Center, Moffett Field, CA, May 1990. (Available from the authors.)

10. Dewberry, B. S., Space Station Freedom ECLSS: A Step Toward Autonomous Regenerative Life-Support Systems. Proceedings of the 5th Annual Conference on AI for Space Applications, May 1990.

11. Glass, B. J., A Model-Based Approach to the Symbolic Control of Space Subsystems, AIAA Paper 90-340, Portland, OR, 1990.

Table 1 Example of a qualitative S_matrix for the ammonia reactor

S_matrix	Inputs				
Outputs	T_1	P_1	f_1_H_2	f_1_N_2	f_1_NH_3
T_2	+	0	–	–	–
P_2	0	0	0	0	0
f_2_H_2	+	0	+	–	+
f_2_N_2	+	0	–	+	+
f_2_NH_3	–	0	+	+	+

Table 2 Partially annotated S_matrix

S_matrix	Inputs				
Outputs	T_1	P_1	f_1_H_2	f_1_N_2	f_1_NH_3
T_2	+0.022	0	–	–0.370	–
P_2	0	0	0	0	0
f_2_H_2	+0.0503	0	+	–0.158	+
f_2_N_2	+0.018	0	–	+0.948	+
f_2_NH_3	–0.035	0	+	+0.106	+

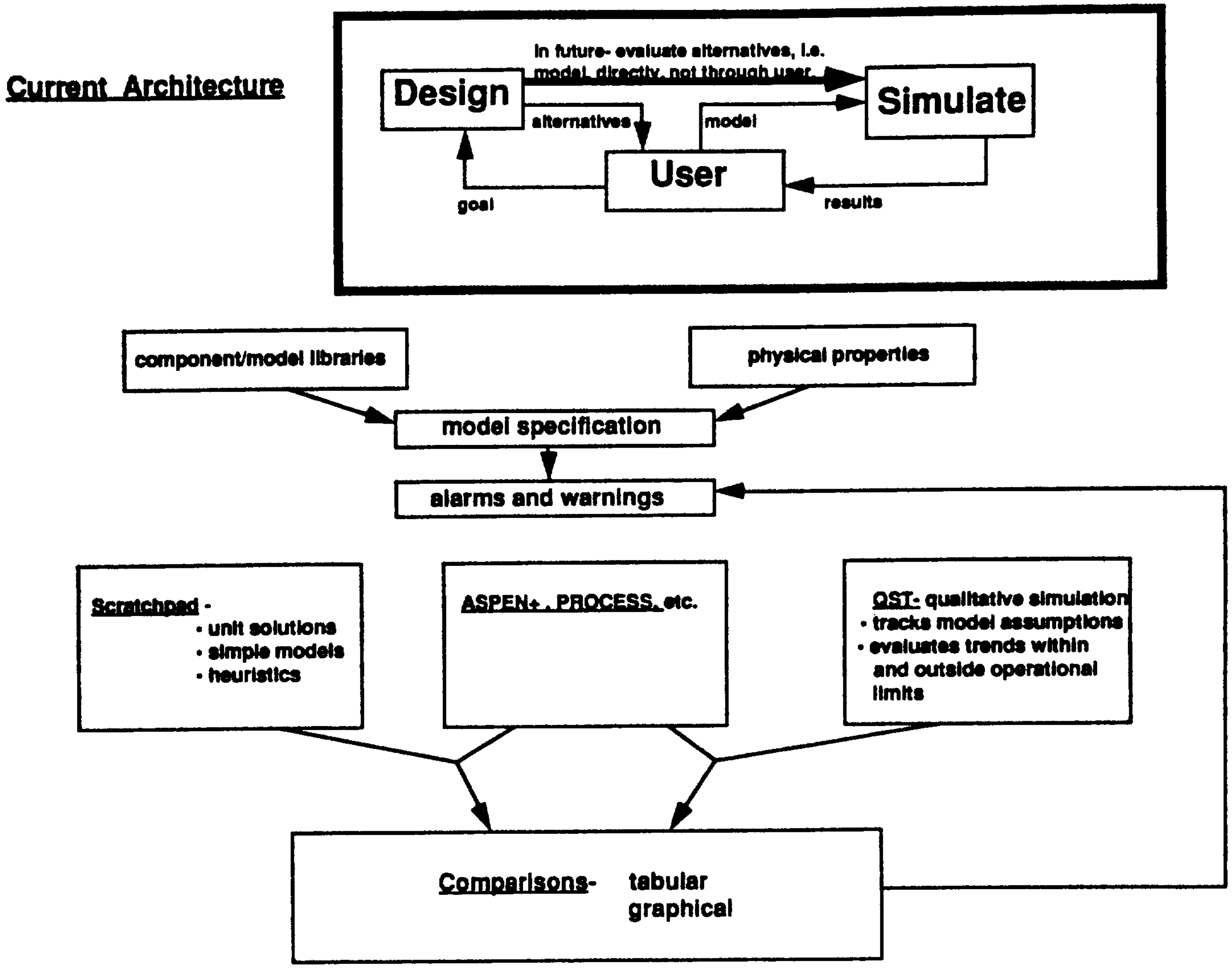

Current Architecture
In future- evaluate alternatives, i.e. model directly, not through user.
Design
Simulate
alternatives
model
User
goal
results
component/model libraries
physical properties
model specification
alarms and warnings
Scratchpad -
• unit solutions
• simple models
• heuristics
ASPEN+ , PROCESS, etc.
QST- qualitative simulation
• tracks model assumptions
• evaluates trends within and outside operational limits
Comparisons- tabular graphical

Fig. 1 Current architecture.

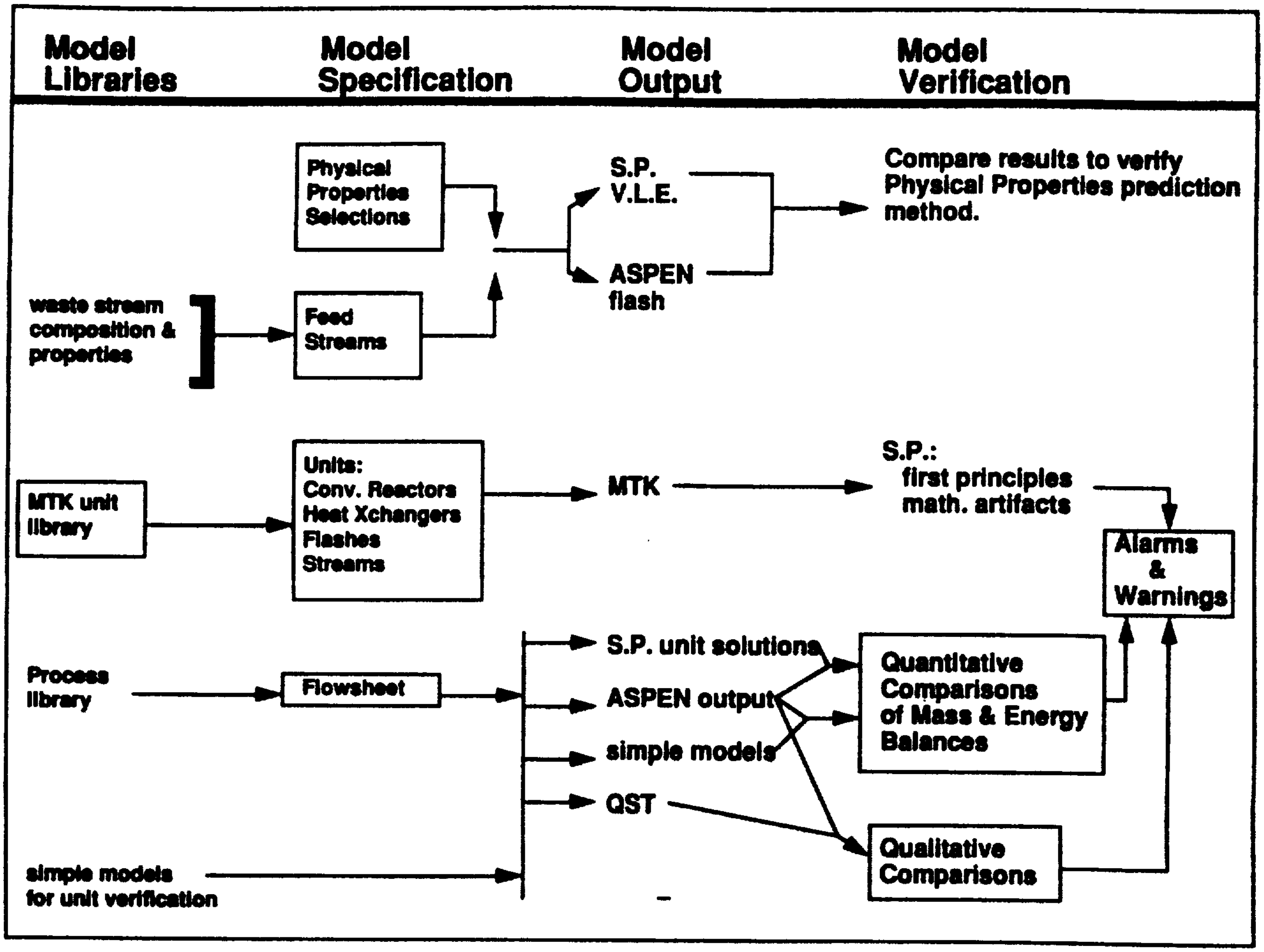


Fig. 2 Conceptual overview of DAWN.

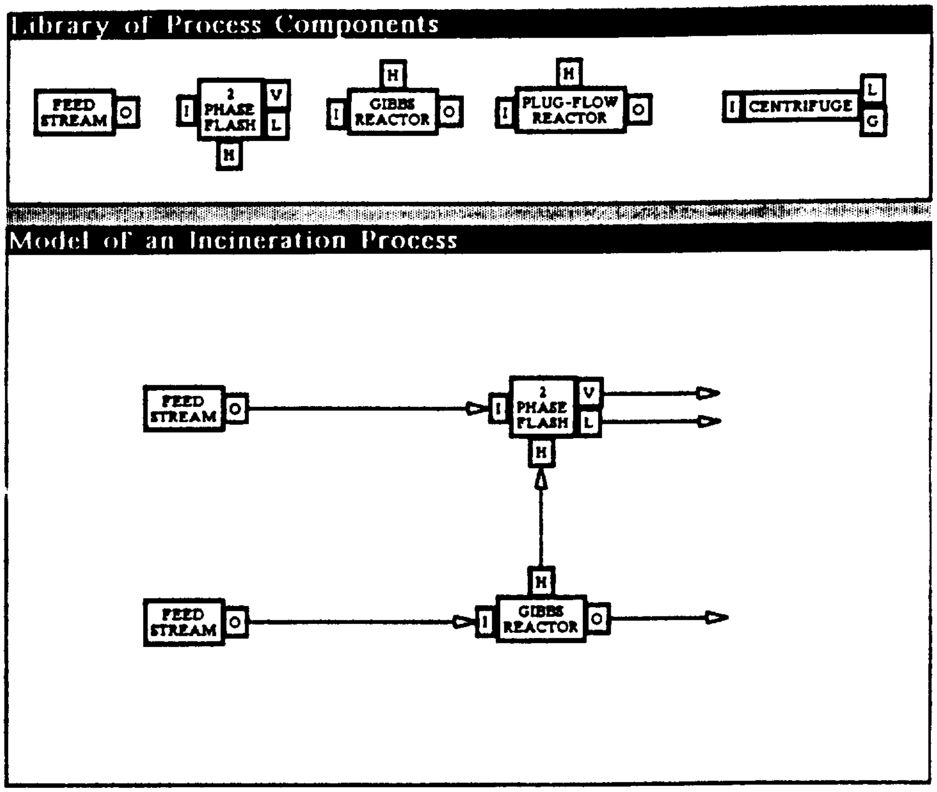
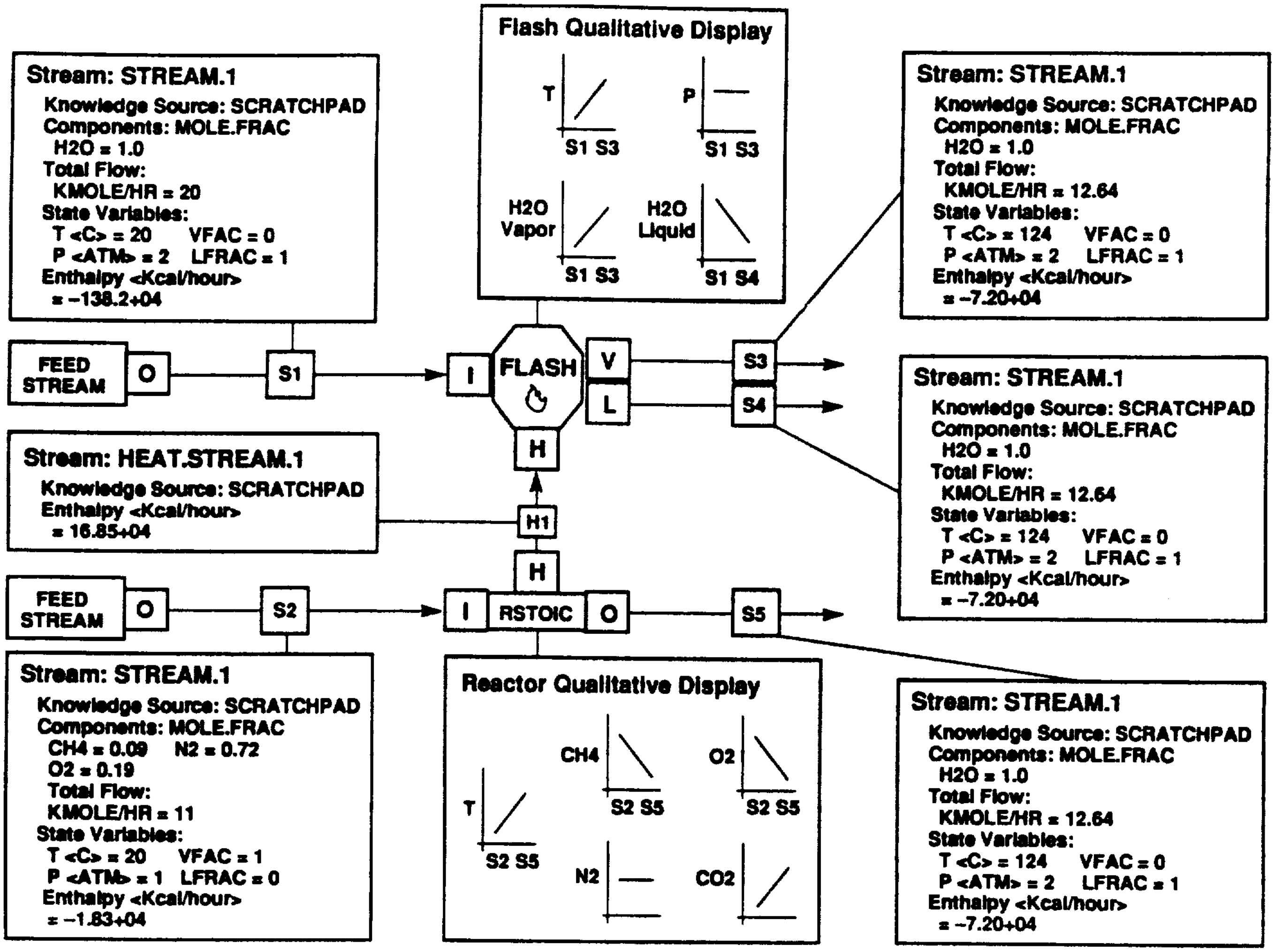


Fig. 3 Model libraries.

QUALITATIVE/QUANTITATIVE DISPLAY OF INCINERATION PROCESS



Stream: STREAM.1
Knowledge Source: SCRATCHPAD
Components: MOLE.FRAC
H2O = 1.0
Total Flow:
KMOLE/HR = 20
State Variables:
T <C> = 20 VFAC = 0
P <ATM> = 2 LFRAC = 1
Enthalpy <Kcal/hour>
= -138.2+04

Stream: HEAT.STREAM.1
Knowledge Source: SCRATCHPAD
Enthalpy <Kcal/hour>
= 16.85+04

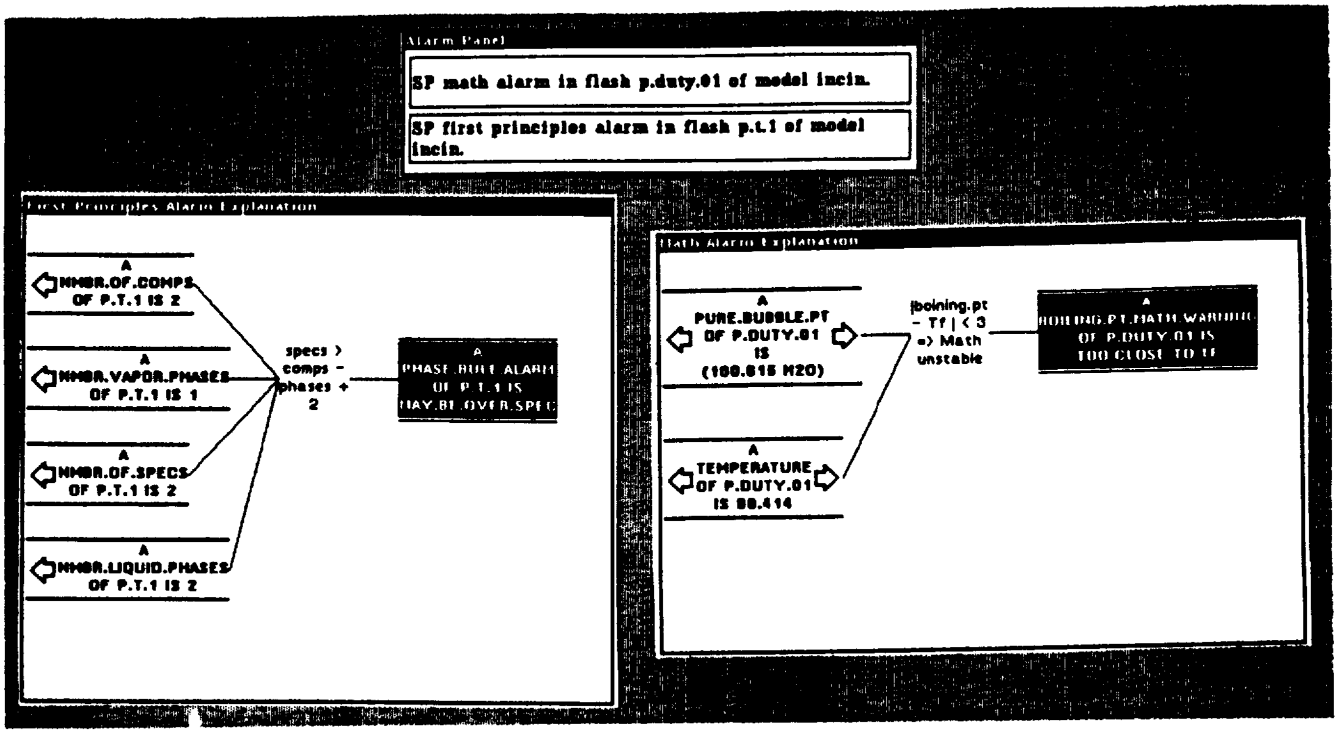
Stream: STREAM.1
Knowledge Source: SCRATCHPAD
Components: MOLE.FRAC
CH4 = 0.09 N2 = 0.72
O2 = 0.19
Total Flow:
KMOLE/HR = 11
State Variables:
T <C> = 20 VFAC = 1
P <ATM> = 1 LFRAC = 0
Enthalpy <Kcal/hour>
= -1.83+04

Stream: STREAM.1
Knowledge Source: SCRATCHPAD
Components: MOLE.FRAC
H2O = 1.0
Total Flow:
KMOLE/HR = 12.64
State Variables:
T <C> = 124 VFAC = 0
P <ATM> = 2 LFRAC = 1
Enthalpy <Kcal/hour>
= -7.20+04

Stream: STREAM.1
Knowledge Source: SCRATCHPAD
Components: MOLE.FRAC
H2O = 1.0
Total Flow:
KMOLE/HR = 12.64
State Variables:
T <C> = 124 VFAC = 0
P <ATM> = 2 LFRAC = 1
Enthalpy <Kcal/hour>
= -7.20+04

Stream: STREAM.1
Knowledge Source: SCRATCHPAD
Components: MOLE.FRAC
H2O = 1.0
Total Flow:
KMOLE/HR = 12.64
State Variables:
T <C> = 124 VFAC = 0
P <ATM> = 2 LFRAC = 1
Enthalpy <Kcal/hour>
= -7.20+04

Fig. 4 Qualitative/quantitative display of incineration process.



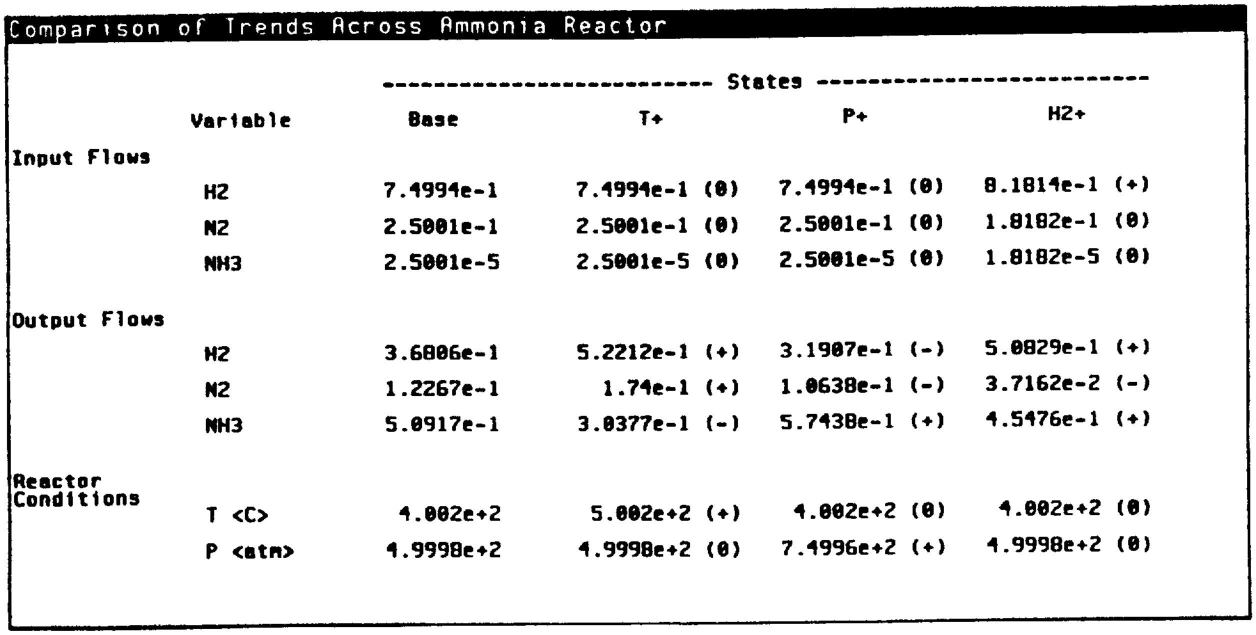
Alarm Panel

SP math alarm in flash p.duty.01 of model incin.

SP first principles alarm in flash p.t.1 of model incin.

First Principles Alarm Explanation
A NMBR.OF.COMPS OF P.T.1 IS 2
A NMBR.VAPOR.PHASES OF P.T.1 IS 1
A NMBR.OF.SPECS OF P.T.1 IS 2
A NMBR.LIQUID.PHASES OF P.T.1 IS 2
specs > comps - phases + 2
A PHASE.RULE.ALARM OF P.T.1 IS MAY.BE.OVER.SPEC
Math Alarm Explanation
A PURE.BUBBLE.PT OF P.DUTY.01 IS (100.815 H2O)
A TEMPERATURE OF P.DUTY.01 IS 98.414
|boining.pt - Tf | < 3 => Math unstable
A BOILING.PT.MATH.WARNING OF P.DUTY.01 IS TOO CLOSE TO TF

Fig. 5 Comparison of trends across ammonia reactor.



Comparison of Trends Across Ammonia Reactor

	Variable	Base	T+	P+	H2+
		States			
Input Flows					
	H2	7.4994e-1	7.4994e-1 (0)	7.4994e-1 (0)	8.1814e-1 (+)
	N2	2.5001e-1	2.5001e-1 (0)	2.5001e-1 (0)	1.8182e-1 (0)
	NH3	2.5001e-5	2.5001e-5 (0)	2.5001e-5 (0)	1.8182e-5 (0)
Output Flows					
	H2	3.6806e-1	5.2212e-1 (+)	3.1907e-1 (-)	5.0029e-1 (+)
	N2	1.2267e-1	1.74e-1 (+)	1.0638e-1 (-)	3.7162e-2 (-)
	NH3	5.0917e-1	3.0377e-1 (-)	5.7438e-1 (+)	4.5476e-1 (+)
Reactor Conditions					
	T <C>	4.002e+2	5.002e+2 (+)	4.002e+2 (0)	4.002e+2 (0)
	P <atm>	4.9998e+2	4.9998e+2 (0)	7.4996e+2 (+)	4.9998e+2 (0)

Fig. 6 ScratchPad alarms and warnings.

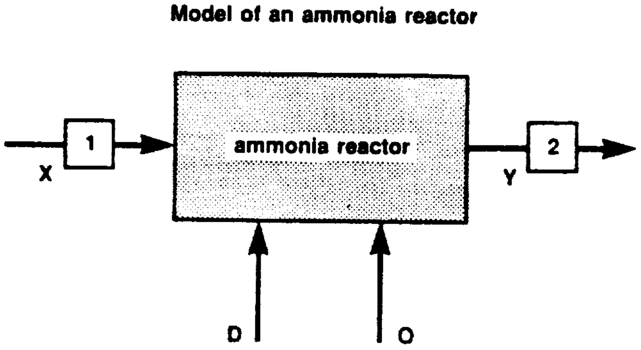


Fig. 7 Model of an ammonia reactor.

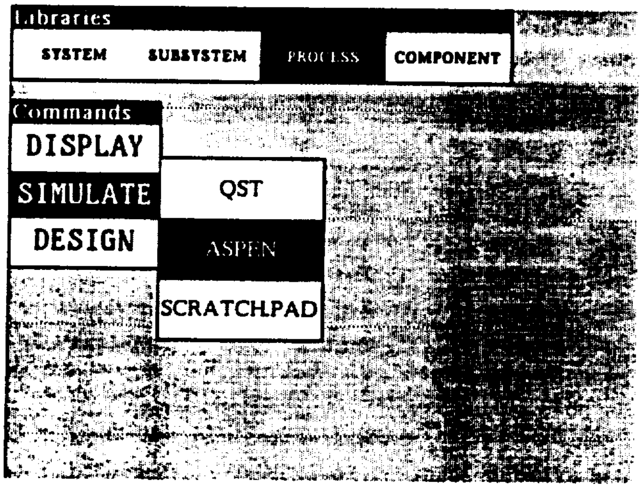


Fig. 8 DAWN user screen.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-102852	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
Automated Simulation as Part of a Design Workstation

5. Report Date
August 1990

6. Performing Organization Code

7. Author(s)
E. Cantwell, T. Shenk*, P. Robinson*, and R. Upadhye†

8. Performing Organization Report No.
A-90245

10. Work Unit No.
506-41-61

9. Performing Organization Name and Address
Ames Research Center
Moffett Field, CA 94035-1000

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes
Point of Contact: Elizabeth Cantwell, Ames Research Center, MS 244-18, Moffett Field, CA 94035-1000, (415)604-3368 or FTS 464-3368

*Recom Technologies, Inc., 1777 Saratoga Ave., San Jose, CA 95129
†Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94550

Report originally prepared for 20th Intersociety Conference on Environmental Systems, Williamsburg, VA, July 9-12, 1990

16. Abstract

A development project for a design workstation for advanced life-support systems (called the DAWN Project, for Design Assistant Workstation), incorporating qualitative simulation, required the implementation of a useful qualitative simulation capability and the integration of qualitative and quantitative simulations such that simulation capabilities are maximized without duplication. The reason is that to produce design solutions to a system goal, the behavior of the system in both a steady and perturbed state must be represented. We report here on the Qualitative Simulation Tool (QST), on an expert-system-like model building and simulation interface tool called ScratchPad (SP), and on the integration of QST and SP with more conventional, commercially available simulation packages now being applied in the evaluation of life-support system processes and components.

17. Key Words (Suggested by Author(s)) Life support systems Simulation Artificial Intelligence	18. Distribution Statement Unclassified–Unlimited Subject Category–54

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 16	22. Price A02

For sale by the National Technical Information Service, Springfield, Virginia 22161