N91-20683

INTELLIGENT MONITORING AND DIAGNOSIS SYSTEMS FOR THE SPACE STATION FREEDOM ECLSS

Brandon S. Dewberry
NASA/MSFC/EB42
Systems Software Branch
MSFC, AL 35812
(205) 544-4247
NASAMail: BDEWBERRY

James R. Carnes
Boeing AI Center
P.O. Box 240002, M/S JA-74
Huntsville, AL 35824-6402
(205) 461-2348
email: ray@huntsai.boeing.com

ABSTRACT

This paper describes specific activities in NASA's Environmental Control and Life Support System (ECLSS) Advanced Automation Project designed to minimize the crew and ground manpower needed for operations. We will describe various analyses and the development of intelligent software for the initial and evolutionary Space Station Freedom (SSF) ECLSS. The paper describes: (1) intelligent monitoring and diagnostics applications under development for the ECLSS domain, (2) integration into the MSFC ECLSS hardware testbed, (3) an evolutionary path from the baseline ECLSS automation to the more advanced ECLSS automation processes.

The Environmental Control and Life Support System is a Space Station Freedom distributed system with inherent applicability to extensive automation primarily due to its comparatively long control system latencies. These allow longer contemplation times in which to form a more intelligent control strategy and to prevent and diagnose faults. The regenerative nature of the Space Station Freedom ECLSS will contribute closed loop complexities never before encountered in life support systems.

1. INTRODUCTION

The Environmental Control and Life Support System (ECLSS) aboard Space Station Freedom will sustain a safe shirt sleeve environment for its crew and payloads. Development has been divided into six functionally interconnected subsystems (Figure 1): Temperature and Humidity Control (THC), Waste Management (WM), Fire Detection and Suppression (FDS), Atmosphere Control and Supply (ACS), Water Recovery Management (WRM), and Air Revitalization (AR). The last two subsystems, WRM and AR, close air and water environmental loops to an extent never before attempted in space, and will require new technologies which are now undergoing extensive test and analysis.

1.1 ECLSS Background

Evaluation of the baselined and evolutionary ECLSS water recovery and air revitalization subsystems is continuing in NASA's Core Module Integration Facility (CMIF) and in several SSFP Work Package One development testbeds, all in Building 4755 at Marshall Space Flight Center (MSFC). These testbeds provide an enclosed environment in which regenerative ECLSS components are developed and tested for extended durations, while data is gathered and distributed to various analysis computers and personnel. Component and system tests are specifically designed to help engineers, biologists, and medical experts refine the technical specifications for the regenerative systems, WRM and AR.

The ECLSS is required to be as autonomous as possible to free crew for less mundane activities and to promote system growth. New components and procedures will be introduced to the system as if evolves. The baseline ECLSS is quite dynamic and will have a variety subsystems either functioning, or in a state of reserve, maintenance or repair.

Managing the operation of any one ECLSS subsystem is a formidable task taxing the current state of practice in software engineering. Although, as stated earlier, the ECLSS is a set of highly interactive subsystems, whose interaction has been isolated to a set of well-controlled water and gas buffers. However, these interactions, and the operations of the system as a whole, cannot necessarily be expressed in engineering terms given the atmospheric, chemical and biological processes defined across the various interfaces[5]. In the baseline system, crew members will be required to "tune" the system to achieve specific performance parameters dependent on two or more ECLSS subsystems.

As knowledge based systems are well-suited for controlled searches of large amounts of reconfigurable data, the use of knowledge-based system processes may provide enhanced capabilities to meet these needs within the baseline Space Station Freedom computing environment. The knowledge structures used by these systems may also serve to store important data for future reference and training.

1.2 Project Objectives

Preliminary study limited the scope of the domain to the potable water, hygiene water, and air revitalization problem analysis since they are functionally complex, yet still amenable to

C-5

knowledge-based solution. While a detailed ECLSS automation system evaluation revealed several viable applications within that domain, the potable water recovery fault detection, isolation and recovery (FDIR) functions were used in early prototyping. This limited the effort to a reasonable size, while providing a proof-of-concept and driving a detailed requirements derivation process. The early prototyping domain was selected based on: the abundance of knowledge, high level of visibility, and need to accelerate advanced functionality for advanced automation. The potable water system was prototyped by knowledge engineers working with ECLSS technical experts.

The current objectives for this project are to demonstrate fault detection, isolation and recovery capabilities at the subsystem level for the Potable Water, Hygiene Water, CO2 Reduction and CO2 Removal Processes, and ECLSS system level control, diagnostics, and trends. To accomplish these objectives we are integrating different advanced technologies such as knowledge acquisition, model-based reasoning, distributed computing to support software development and problem solution. We are leveraging our software development process with knowledge engineering tools from other NASA or Boeing projects including: Aquinas for knowledge acquisition, ART/Ada Automated Reasoning Tool shell for associational reasoning, KATE for model-based reasoning, and Erasmus for distributed blackboard operations. One of the strong goals for this project is to demonstrate and document a growth path for baseline software functions into intelligent systems. This paper provides background description of the Space Station ECLSS then focuses on the diagnosis methodology and implementation.

2. ECLSS DESIGN

Life Support Systems are required to provide the habitable environment for the crew and life sciences payloads. This environment includes water for drinking and washing, and atmospheric gasses. Previous life support systems have typically met these requirements by maintaining sufficient supplies of pressurized gasses and fluids, through closed loop options have been investigated[6].

2.1 Baseline Process Description

The Temperature and Humidity Control, Water Recovery Management, and Air Revitalization Subsystems aboard the Space Station combine to meet the water and air supply requirements as in Figure 1. These requirements are met by closing the air and water loops to an extent never before implemented in space. Even so, the control system is essentially open loop, a batch filtering process. Little chemical or microbial data is fed back into the control system for use in adjusting flexible processes for maximum efficiency.

The system is tested on the ground for sufficient cleaning and recycling set types and levels of fluids in the air and water, and is periodically verified on orbit using batch laboratory analysis procedures. This alone, the actual integration of these multiple interacting subsystems to specified requirements, will be a great achievement. Lessons learned in the on-orbit integration of these batch processing systems will be invaluable in determining micro-gravity interactions and recombinations of chemical and microbial constituents throughout the revitalization systems.

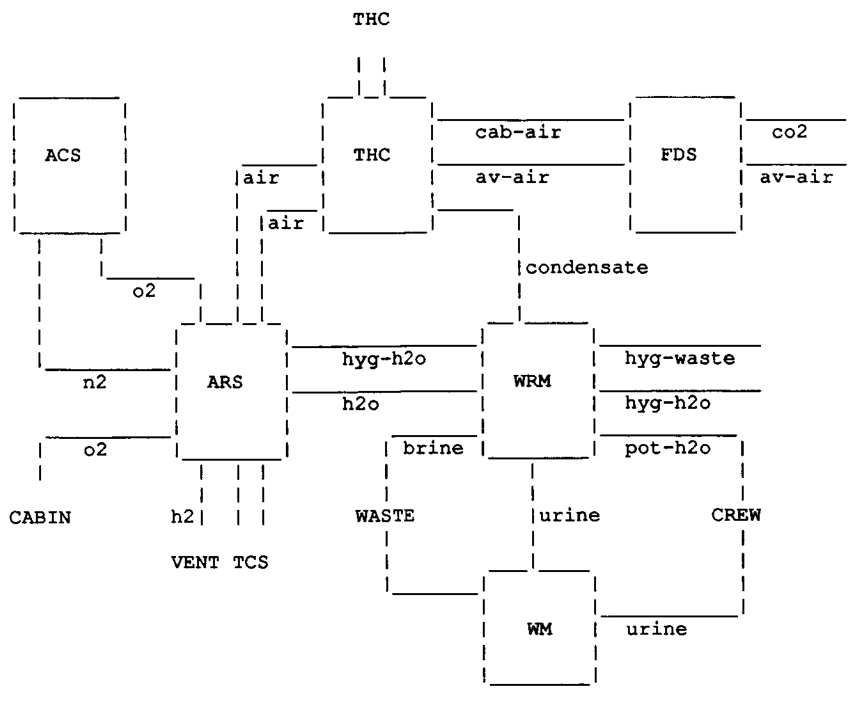

FIGURE 1 - ECLSS FUNCTIONALLY INTERCONNECTED SUBSYSTEMS

Two software processes which were determined prime candidates for automation are Real-time and Off-line Subsystem FDIR (Fault Detection, Isolation, and Recovery), and Component Performance and Trend Analysis. Both of these processes will contain parts initially in the ECLSS Ground Sustaining Engineering, with migration on-board when flight data management resources permit. An overview of the software architecture for the ECLSS can be found in reference [5].

2.2 Advanced Regenerative Life Support System

In general, future autonomous regenerative life support systems, including the evolutionary ECLSS, will be required to supply water and air, within specific chemical and microbial limits, for extended durations without crew or ground support adjustment. The control system and plant will be intelligent and robust enough to autonomously withstand unexpected crew and payload anomalies. These requirements will be achieved with a minimal set of instrumentation and processing assemblies.

These requirements may be met by augmenting the baseline ECLSS with various technologies. Software hooks and hardware scars in the baseline will be necessary to minimize the impact of integrating these technologies after Assembly Complete. Increased automation of the ECLSS is possible, but evolution to complete automation, defined as above but requiring some simple unit replacement occasionally, may not be feasible due to the degree of fundamental process adjustments and control strategies required. But the ECLSS can be used to dramatically increase the state-of-the-art in regenerative life support systems.

There are several advantages to beginning ECLSS automation with upgrades in the automatic fault isolation and recovery and health maintenance (failure prediction and prevention) processes. These processes are software oriented and theoretically, software is the most flexible part of the system and most amenable to upgrade.

Automatic fault isolation and recovery (FDIR) and health maintenance (failure prediction and prevention) processes require the implementation of emerging software technologies. These processes can be verified in the ground support environment and migrated to the flight ECLSS to increase the Station's flight autonomy. This approach to increasing ECLSS autonomy is described in [3] and [4] and is be the focus of the ECLSS Advanced Automation Project.

3. DIAGNOSIS APPROACH

We have divided the diagnosis problem into three layers: Reactionary, Heuristic and Comprehensive. Reactionary diagnoses are easily classified at low levels in the component structure and are usually within the scope of and implemented in the control logic (i.e., look-up tables) of a single component. They are often manifest as caution and warning statements in human interfaces. Specification of these so called "reactionary diagnoses" is already built-in to the baseline ECLSS design and will not be covered in this paper.

Heuristic diagnoses are characterized with some degree of accuracy (or confidence) by selected "rules-of-thumb". These rules usually associate specific component and system state information is a diagnostic observation (e.g., a component fault warning). This type of analysis is presented below in our work on associational diagnosis.

Comprehensive diagnoses are characterized, in our system, by component-oriented models of system structure and function. These models provide a stronger, perhaps more detailed definition of each system component; however, the major distinction of this diagnosis class is the causal propagation of structural and functional data through component networks to determine (and discriminate among) a set of diagnostic hypotheses. This type of analysis is presented below in our work on model-based diagnosis.

3.1 Associational Diagnosis

The associational diagnosis applications are designed to function quickly, constructing diagnostic observations about the functioning (or malfunctioning) system through standard forward- and backward-chaining mechanisms. Component-oriented rules used here are shallow and and are quite sensitive to system configuration/mode and component state changes. However, the ECLSS environment can be partitioned into a small of major operating configurations and system modes making feasible the use of this heuristic rule-based approach to diagnosis without a combinatorial explosion of rules. The ultimate goal for this module is to construct a diagnosis approach (that is compact in size, yet broad in scope) for integration with ECLSS flight software on-board the Space Station.

We are currently working on two approaches to associational diagnoses. The first approach is strictly heuristic mapping a set of abstracted system states into a set of possible component diagnoses with confidence levels. The mapping between system states and component diagnoses are acquired and managed with a knowledge acquisition workbench, Aquinas (from Boeing). Aquinas gathers the complex relationships between system traits and diagnostic solutions from one or more experts and stores it in a hierarchical network of repertory grids[11]. This knowledge can be examined and refined using tools that do clustering, similarity analysis, implication analysis, and consultation testing. These tools use techniques to analyze the information in the grids and suggest way to refine the knowledge base. After the diagnostic knowledge in an Aquinas grid is verified by the ECLSS engineers and ready for operational use, it is encoded into a set of ART/Ada (from Inference) facts and rules.

The second approach employs a component-oriented model-base written in G2 (from Gensym), a real-time expert system shell. G2 can be used to develop the same type of heuristic diagnosis model as described above, however, causal models can also be defined and used with the G2's built-in simulation engine to propagate functional properties through a network of components. When the simulation engine is run in parallel with the real-time system, the simulated property values in each component can be compared with the analogous observed values. If a discrepancy is noted diagnostic rules that reference the anomalous values are activated and diagnostic reasoning runs through matching and resolution mechanisms to produce forward- and backward-chaining effects. This approach provides an excellent architecture for system monitoring and a stronger (than the first approach) to diagnosis with an approach for focussing the inference engine on specific diagnosis rule, increasing the manageability of the rule set and decreasing the response time required for diagnostic analysis. However, a major weakness in this approach, as in the first approach, is that a specific diagnosis can be rendered only if it was preconceive and programmed into the rule set.

3.2 Model-Based Diagnosis

Component-oriented definitions are also used in the model-based approach but are in more detail and are quite robust in their reaction to system configuration/mode and component state changes. The near-term goal for this module is to construct a diagnosis approach to appraise the overall system health from a ground-based site integrated with ECLSS ground support software.

The model-based diagnosis application in this project is accomplished with NASA's KATE (Knowledge-based Autonomous Test Engineer) software. Models of the ECLSS subsystem processes are being constructed and refined using the KATE definition language. The KATE knowledge base use a frame representation to model the system processes. Specifically each component's interfaces, functions, measurement and command structure defined within the slots of selected frames. An example of a component definition in KATE's declarative-style definition language is shown in Figure 2.

```
(DEFRAME PUMP
  (NOMENCLATURE "a pump")
  (AKO ANALOG-OBJECT)
  (INSTANCES PUMP1)
  (INPUTS (IN1))
  (OUTPUTS (OUT1) (OUT2))
  (OUTPUT-FUNCTIONS
    (OUT1 (* IN1 PUMP-OUT-SELECT))
    (OUT2 (- IN1 (* IN1 PUMP-OUT-SELECT))))
  (PARAMETERS (PUMP-OUT-SELECT 0.5))
  (DELAY (OUT1 2) (OUT2 2))
  (TOLERANCE (OUT1 0.1) (OUT2 0.2))
  (UNITS "ml/min"))
```

FIGURE 2 - SAMPLE KATE OBJECT DEFINITION

The diagnosis algorithm in KATE scans a set of observed measurements comparing them to a set of simulated values obtained by propagating commands forward through the network of components models. Once some measurement has been noticed to be discrepant, the diagnoser is invoked to localize the fault to the extent possible. Faults are perturbations from a system's expected functionality. Diagnosis, in this case, is the search for one or more faults that can explain the system's observed behavior. The strength of component-oriented modeling lies in its ability to hypothesize faults from the information given by discrepant sensor readings[7].

A general analysis generates possible fault hypotheses for possible faulty objects, these fault hypotheses predict hypothesized sensor measurements, and those measurement hypotheses are tested against observed sensor readings. An agreement of fault hypothesis with observation lends support to (but does not prove) the hypothesis, while a contradiction would rule out that hypothesis. If all fault hypotheses for a particular faulty object are ruled out, then that object is no longer suspect.

Fortunately, one does not need to test all suspects (potentially faultly objects) against all related sensors. Search is anchored by the discrepant sensor or sensors. Only objects which are connected in controlling relationships to a discrepant sensor-object are considered as potential suspects. The diagnostic algorithm is discussed in more detail in [10]. An earlier, more structurally-oriented diagnostic algorithm is discussed in [9].

The greatest savings comes from using discrepant sensors to reduce or even eliminate the search for hypothetical faults by transmitting the information in their readings to the suspects. This means effectively inverting the dependency of sensor upon suspect which is known through the interface and function expressions. Such inversion generates all hypotheses consistent with discrepant sensor readings, and even these are eliminated where possible by contradiction with other sensors.

4. PROJECT ARCHITECTURE

The major goal of this project is to produce intelligent system software to monitor, control, and diagnosis the Space Station ECLSS. Five major components of this software system are under development in a distributed environment: console interface, model management, data acquisition, associational reasoning, and model-based reasoning. The applications for diagnostic reasoning have already been discussed in detail in the previous section. Figure 3 illustrates the relationships between the software subsystems in this project.

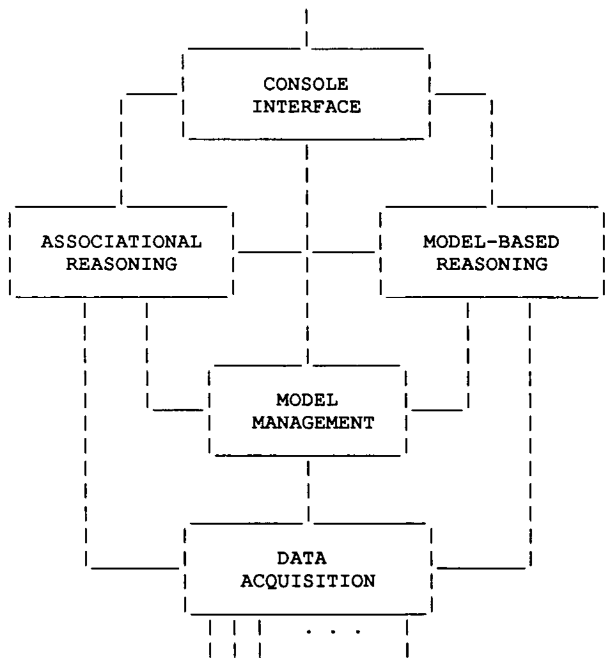

FIGURE 3 - ADVANCED ECLSS AUTOMATION ARCHITECTURE

4.1 Console Interface

The Console Interface module provides human computer interaction for monitor and control applications running in different modules. The look-and-feel of the interface conform to the Space Station Freedom Program Work Package 02 standard SY-45.1[2] and is being developed in TAE-plus, an X-window-oriented application tool for prototyping computer interfaces for both flight controls and ground consoles. The Console Interface receives information on current system state/status from sensor and actuator data (formatted by the Model Manager) and updates on monitoring and diagnostics applications for the Associational and Model Based Reasoners. Control commands for the domain system can be formed and issued through the Model Manager, while control of monitoring and diagnosis applications is fed to the appropriate reasoning module.

4.2 Model Manager

The Model Manager is a module to store and control access to the object knowledge base and run-time database. It provides a consistent definition of components/systems to the control and diagnostic algorithms running in the Associational and Model Based Reasoning modules. The Model Manager also manages the run-time collection of the ECLSS environment collecting observations of the sensors/actuators from the data acquisition module.

4.3 Data Acquisition

The Data Acquisition module for the system is provides binding to all sensors and actuators running in the hardware testbed. Data collection for the prototype software is accomplished through a modification to the existing SCATS (Systems and Components Automated Test System) data server used to supply data to the control panels test bed control room.

4.4 Diagnostic Modules

The independent diagnosis approaches described in the previous section are being integrated into the reasoning modules and coupled with the Model Manager. Structural and functional models of the ECLSS subsystem processes are used to diagnose and isolate failures. The model based approach to diagnosis is computationally intensive but performs autonomous, in-depth diagnosis of faults. The process control nature of the ECLSS allows the use of emerging model based reasoning tools in automating the system, while storing knowledge in component form[7]. The system also may be upgraded for automatic diagnosis of regeneration analysis with the future inclusion of chemical and microbial transfer equations.

We have analyzed and developed detailed models of two different processes within the WRM subsystem (Potable Water System and Hygiene Water System). KATE and G2 models have been constructed for the Hygiene Water process. The work with the process models focuses on integrating multiple-aspect models (i.e., structural, functional, thermal, etc.) as opposed to the explicit modeling of disfunction.

Associational failure models for the Potable Water process were developed using Aquinas. Current models establish relationships, in hierarchical grids, down to a level below the ORUs (Orbital Replacement Units). A functional architecture for the integration is depicted below.

4.5 Testbed Description

The following diagram (Figure 4) depicts the hardware and software configuration currently in use to support prototype development for the CMIF Testbed.

```
                                         prototype | testbed
                                         equipment   equipment
                                                <---|--->
 (system monitoring     (fault management
   and diagnosis)       performance testbed)        |
                                                          o testbed data
   o TAE-plus               o TAE-plus              |       collection
   o Gensym/G2              o ART/Ada                       distribution
                                                    |
  _____________          ______________                  ______________
 |             |        |              |                |              |
 |    SUN      |        |     IBM      |            |   |     DEC      |
 |  Sparc-1    |        |  PC-80386    |                |   VAX 6320   |
 |_____________|        |______________|            |   |______________|
        |                      |                               |
        |                      |                    |          |
==========================================================================
ethernet      |                   |                 |             |
        ______|______       ______|______                  _______|______
       |             |     |             |          |     |              |
       | Symbolics   |     | Symbolics   |                |  Subsystem   |
       |   3670      |     |   3650      |          |     |  Hardware    |
       |_____________|     |_____________|                |______________|
                                                    |
        o Gensym/G2         o KATE                     o potable water process
                                                    |  o hygiene water process
       (subsystem          (subsystem                  o co2 reduction process
        simulation          simulation              |  o co2 removal process
        and monitoring)     and diagnosis)
```

FIGURE 4 - EXAMPLE ARCHITECTURE FOR TESTBED INTEGRATION

5. CONCLUSION

The Environmental Control and Life Support System aboard the Space Station Freedom will be a step ahead in the implementation of regenerative life support systems. The interactions of its subsystems with each other and the crew will serve to greatly increase our knowledge in low gravity regenerations complexities. The Space Station can be used as a test bed for verification of chemical and microbial, variable gravity transfer models which will prove essential in long duration regenerative life support system engineering and autonomy analysis.

The fully automated regenerative life support system described cannot be built today. Quite a few steps must be taken, and research performed in order to develop systems which can autonomously remain stable for long durations. A first step is to build and deploy the Freedom Station. The actual hands-on knowledge generated from ground and flight test will allow incremental builds upon the ECLSS toward automation and long term stability. Another step is the inclusion of the Life Sciences medical technology in Life Support engineering. Life support systems which use regenerative techniques to meet their supply requirements will have to actively worry about and control microbial recombination, and insure

To support this work future work in Aquinas will include automatic generation of ART/Ada rules from grid structures. Future work in KATE will include simultaneous equation solving and constraint suspension to provide more flexibility in modeling physical systems and more discriminatory power in diagnosis.

6. REFERENCES

[1] Carnes, J. R., "Model-Based Diagnostics," in the Intelligent Sensory Processing Workshop, IEEE International Conference on Robotics and Automation, May, 1990.

[2] Cohen, A. D., "User Interface Requirements Document (DR SY45.1)," MDC H4261 (Revision A), McDonnell Douglas Space Systems Company-Space Station Division, March 1990.

[3] Dewberry, B. S., "The Environmental Control and Life Support System Advanced Automation Project - Phase I Application Analysis," Proceeding of the Space Operations Automation and Robotics Conference, June, 1989.

[4] Dewberry, B. S., "Automation of the Environmental Control and Life Support System," Proceedings of the NASA Space Station Evolution Symposium, February, 1990.

[5] Dewberry, B. S., "Space Station Freedom ECLSS - A Step Toward Autonomous Regenerative Life Support Systems," NASA CP-3073, Proceeding of the Fifth NASA Conference on AI for Space Applications, Huntsville, AL, May, 1988, pp. 193-201.

[6] Prince, N. R., et. al., "Challenges in the Development of the Orbiter Atmospheric Revitalization Subsystem," NASA No. 2342, Part 1, NASA Space Shuttle Technical Conference, June, 1983, pp. 465-479.

[7] Scarl, E. A., Jamieson, J. R., and New, E., "Deriving Fault Location and Control from a Functional Model," Proceedings of the 3rd IEEE Symposium on Intelligent Control, Arlington, VA, August, 1988.

[8] Scarl, E. A., Jamieson, J. R., and Delaune, C. I., "A Fault Detection and Isolation Method Applied to Liquid Oxygen Loading for the Space Shuttle." In Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, August, 1985, pp. 414-416.

[9] Scarl, E. A., Jamieson, J. R., and Delaune, C. I., "Monitoring and Fault Location at the Kennedy Space Center," in Newsletter of the ACM Special Interest Group on Artificial Intelligence (SIGART), No. 93, July, 1985, pp. 38-44.

[10] Scarl, E. A., Jamieson, J. R., and Delaune, C. I., "Sensor-Based Diagnosis using Knowledge of Structure and Function," IEEE Transactions of Systems, Man, and Cybernetics, SMC-17, No. 3, May/June, 1987, pp. 360-368.

[11] Shema, D. B. and Boose, J. H., "Refining problem-solving knowledge in repertory grids using a consultation mechanism," International Journal of Man-Machine Studies, Vol. 29, No. 1, July, 1988. pp. 447-460.