

# SELF-EVOLVING, ADVANCED TEST STAND REASONING FOR CLOSED LOOP DIAGNOSTICS

Patrick W. Kalgren, Carl S. Byington  
Impact Technologies, LLC  
2571 Park Center Blvd.  
State College, PA 16801  
814.861.6273  
patrick.kalgren@impact-tek.com

**Abstract - Advanced test stand reasoning can yield dramatically increased Line Replaceable Unit (LRU) operational availability through reduction of Mean Time to Repair (MTTR) while offering better utilization of related maintenance and test equipment resources. However, technical challenges related to development of reasoning systems, typically requiring expert LRU domain knowledge, present hurdles that can be prohibitive. Reasoning methods developed using simple pattern recognition and repair instance statistics offer a first order approach that is functional from a proof of concept perspective. Unfortunately, the small numbers of statistics available historically provide reduced reasoner reliability and effectiveness when confronted with dynamic and complex avionic systems. This paper discusses the issues, trade-offs, and potential benefits to be gained through the application of robust, self-evolving, hybrid reasoning techniques. A reasoner that can utilize and leverage the constraints found in typical test stand procedures to provide a best, safe path to diagnosis, while learning and optimizing in-situ, may offer an ideal, scalable solution for optimizing test stand operations. Related information management and diagnostic model visualization techniques are also presented in the context of diagnostic and avionic system evaluation and improvement.**

## INTRODUCTION

Traditional approaches to test and repair avionics equipment provide a sub-optimal path to quick diagnosis and repair. With a design typically centered on a primary need for verification of

functionality for flight readiness certification and little built-in ability to utilize operational or system-level evidence, current systems are limited. Emerging concepts for improvement of these approaches strive to maintain the safety and reliability provided by the traditional approach while recognizing the achievability of faster repair times. Recent developments that use statistics to relate patterns of evidence to optimal testing paths, developed in feasibility demonstrations, have shown promise and demonstrated achievable benefits through optimized testing. Although statistical methods provide a necessary first step at assembling a body of learned material correlating evidence with failures, the diversity of weapons platform operational modes, relative small number of statistics, and the changing nature of electronics platforms indicate the need for a more robust approach.

When considering the development of innovative test station reasoning systems, a large advantage provided by statistical approaches must not be ignored. The ability to provide improved diagnostics without investing heavily in obtaining an engineering level understanding of operational detail, functionality, and failure mode effects must remain a primary motivator for the development of improved reasoning systems. The approach described here, while permitting the inclusion of engineering knowledge, can be implemented using a purely symbolic representation of all evidence and maintenance process elements. The clear advantage to legacy systems where engineering design knowledge is difficult, at best, to obtain and the possibility to provide enhancements to the maintenance and repair of classified systems are points that should be emphasized.

## TODAY'S STATE OF THE ART

Normal avionics test procedures typically involve the use of Automated Test Equipment (ATE) following a test and diagnosis process described by a Test Program Set (TPS). The time consuming procedure creates a bottleneck in operational availability of critical weapons platforms. During this process, testing always begins at step one and proceeds in a linear fashion through a tree-based logical structure. Upon detection of an anomaly, the test procedure halts and provides the test station operator a list of maintenance actions (callouts). After performance of the actions prescribed, the operator then restarts the TPS at an entry point just prior to the anomaly detection. The process proceeds until all tests have passed. In many cases, completion of an end-to-end run, without failure provides final flight recertification.

A new technician will always follow the "safe" diagnostic path. Beginning at the first entry point of the TPS and performing the actions described by all callouts and proceeding until the TPS is complete. An experienced technician may be successful at "short-circuiting" the diagnostic and fault isolation process by using the experience base gained while working in his current position. Some components may exhibit familiar patterns of failure and some callouts may be consistently unnecessary while others are consistently effective. The technician may also gain useful insight when given details about the maintenance action and decision process that precipitated the LRU removal from the aircraft. However, this ad-hoc method to execution of speedy repairs breaks down on two levels.

1. Information typically does not have a consistent path from at-wing maintenance to the test station.
2. Promotions and advancements move a technician out of his current position and his experience base leaves with him.

Some current efforts are striving to bridge the gap between onboard and at-wing information sources and the test station repair facility. The NAVAIR/Boeing Smart TPS study [1] provided a proof of feasibility and a suggested information management framework to facilitate this exchange of data. The joint services have undertaken an endeavor to provide a demonstration of many

emerging concepts, including closing the information loop to take better advantage of information at all levels of repair and begin to facilitate test system interoperability. Agile Rapid Global Combat Support (ARGCS) provides a vehicle for this concept demonstration. To date, feasibility of enhanced test station processes has been demonstrated and benefits have been realized using basic statistical correlation techniques with small subsets of the evidence available. We believe that more can be achieved and that the concept presented here will provide a vehicle to improved diagnostics resulting in faster LRU repair times and improved mission readiness through greater operational availability ( $A_0$ ).

## A SELF-EVOLVING REASONING SYSTEM

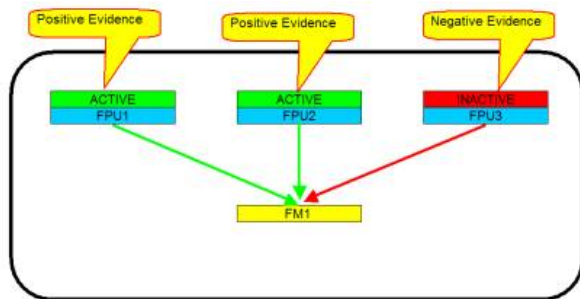
The authors have developed an approach to closed-loop diagnostics that employs advanced techniques in a hybrid architecture to leverage proven technologies and the existing maintenance infrastructure. ReasonPro – Evolution™ employs many of the techniques presented herein to achieve a robust diagnostic inference system that is portable across a large variety of applications. The following sections provide foundation for the reasoner engineering process. The authors' hope to foster discussion regarding ongoing maintenance community endeavors to achieve the benefits promised by "closed-loop" diagnostics. Components essential to a successful self-evolving approach include:

1. Core reasoning model
2. Pattern recognition and case discovery
3. Solution discovery
4. Solution path control
5. Learned solution integration
6. Learning feedback loop

### Core Reasoning Model

An apt core reasoning model choice will provide a solid foundation for the construction of a robust inference system and permit the operation of test station processes to proceed in a familiar manner. When viewed from the perspective of normal test stand operations, constraints provided by today's methods can be leveraged to provide model components useful for enhanced reasoning. Given a fixed set of TPS entry points, expected callouts, and the knowledge of a "safe" path, a reasoning system can be built to work within these

constraints to learn with an evolving experience base. In this case, a tree-based structure would ultimately be a replica of the typical TPS deployment concept and thusly insufficient to provide added value. A networked model seems to be indicated. We have chosen a commercial off the shelf reasoner developed by Impact Technologies as the core model and computational engine. ReasonPro™ utilizes a nodal structure and a positive/negative evidence schema [2] to provide a robust core inference module that provides a human decipherable model and can be visualized using existing tools.



**Figure 1. ReasonPro™ uses a nodal structure with positive and negative evidence.**

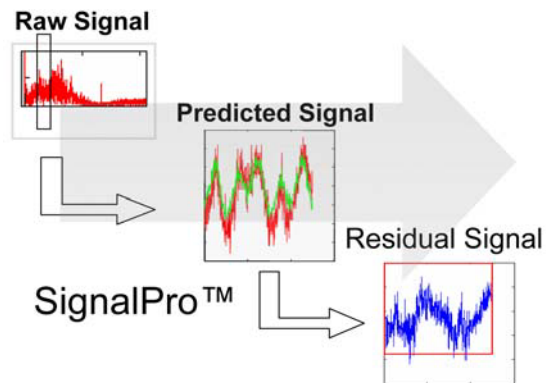
This reasoning model provides a base structure that is stored in a database for ease of access, growth, and future management. Calculation of evidence and model results, performed by a modular core computation engine, supports application environments ranging from embedded to desktop. The model elements can adapt readily to other reasoning algorithms such as Bayesian Belief Networks, permitting the incorporation of hierarchical reasoning structures for data fusion. A model represented by stable data structures and a base algorithm that provides deterministic results is key to successful implementation of an evolvable reasoner.

## Model Evolution

Pattern recognition and case discovery permit the detection of new evidence and growth of the model structure. Evidence useful to diagnostic reasoning can include all or part of onboard BIT data, all or part of at-wing data, and any other related data collected throughout operations and maintenance. The unfortunate consequence of this vast array of data is an overwhelming effect on inference systems and a subsequent dilution of the diagnostic result. Recent attempts to resolve this issue reduce the evidence considered to an

arbitrarily small set giving minimal consideration to data elements external to the central LRU under test. The resulting method may neglect potentially valuable evidence. A two step process to evidence selection yields high benefit by permitting consideration of all evidence items prior to evidence reduction and down selection of elements correlated to the failure event.

Anomaly detection and correlation using autonomous signal prediction provides an indication of relevant evidence changing with the onset of a failure (see Figure 2). Such a pre-filtering step can reduce the available evidence to a manageable and high value subset.



**Figure 2. SignalPro™ Signal prediction and error tracking**

## Solution Discovery

The second stage, case discovery, will assemble available evidence and compare to the existing diagnostic model. Adding new nodes or modifying previously existing evidence nodes as experience grows permits the automated adaptation of the model to new experience and can incorporate adaptation to changing or upgraded systems.

Figure 3 is an example demonstration constructed around a mission critical LRU for the Apache, AH-64D, attack helicopter. This platform implements a maintenance data recorder to maintain a history of greater than 160 data elements. This data, including time series, BIT discretes, and data bus errors provide a system-level view of the operational state of the platform prior to, during, and after LRU failures.

SignalPro™ identified nine highly correlated evidence elements, as illustrated in Figure 3, by simultaneously considering all available evidence. Assembled as a case this reduced evidence set provides a model node for comparison and

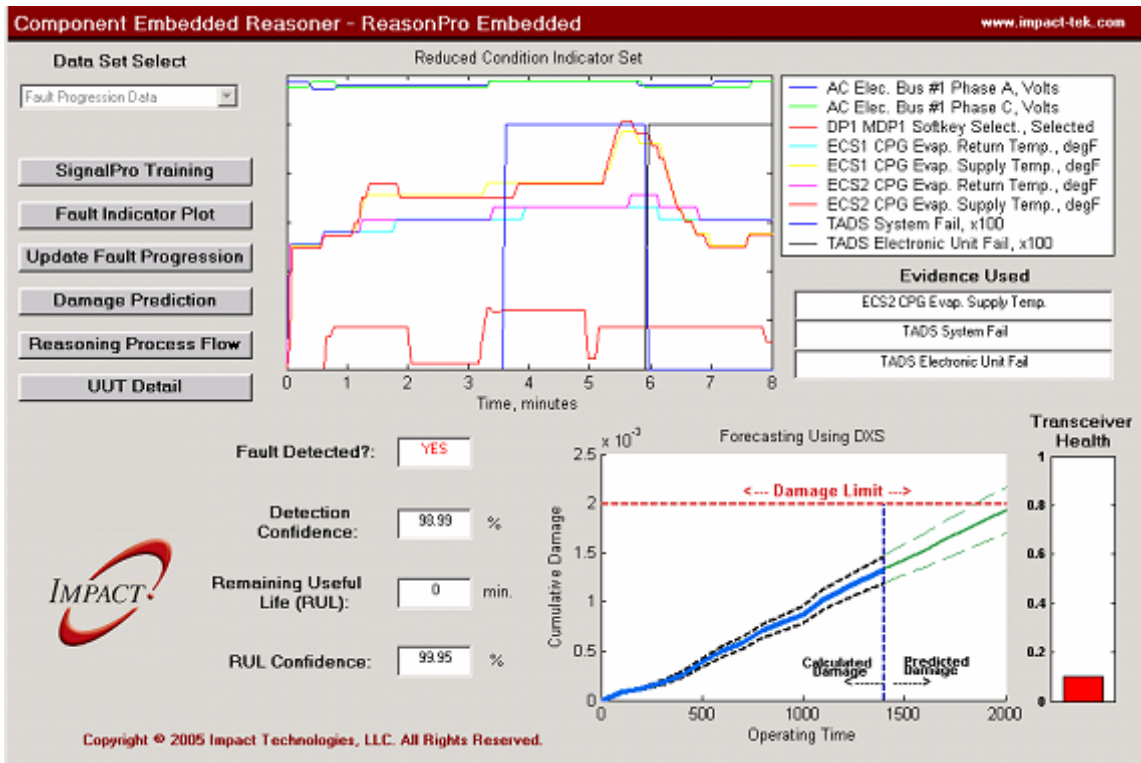


Figure 3. Evidence correlation with SignalPro™

addition to the model providing reasoning robustness for selection of an optimized test station repair path.

### Solution Path Control

Beginning with a base model, knowledge of a “safe” diagnostic path and building a foundation of experience forms a sensible approach to learning. Unfortunately, a naïve approach to reasoning and providing decision support to the maintainer may result in dramatic swings in reliability as the experience base grows. A sound approach requires careful consideration of the implementation and effects of new knowledge. This permits the solution path to mature reliably and establishes a trust relationship with the user base. An automated model growth development and solution path control mechanism is required. Learning techniques utilizing cost metrics and configurable growth parameters can permit this automated, safe growth.

### Reinforcement Learning

Q-Learning is among a series of Reinforcement Learning (RL) algorithms with a common objective to successfully solve problems that require complex decision processing. Reinforcement Learning is a combination of fields such as dynamic programming and supervised learning, which lead to powerful intelligent machine learning systems. One of the biggest advantages of these techniques is the model-free aspect. The applicability of reinforcement learning algorithms is independent from the structure of the system taken into consideration. This makes Q-Learning a good choice to modify the diagnostic model as new information is incorporated.

The Q-Learning process has been chosen to support the model growth and evolution aspect of the ReasonPro – Evolution™ core algorithm. Q-Learning, integrated in the model environment evolution process, reinforces the nodal decision processes and address the evolvable requirements of the program.

The following relationships formally introduce the reinforcement-learning model.

- S: A discrete set of environment states
- A: A discrete set of agent actions
- R: A set of scalar reinforcement signals

The advantage of Q-Learning is that initially, when the optimal state to action relationships are not entirely known, the agent may have the option of randomly selecting an action. Q-Learning allows arbitrary experimentation and simultaneously it preserves the current best estimate state values.

## Solution Integration

The final results in Q-Learning are not contaminated by experimental actions, which will only ultimately reward or penalize the state action pairs. In order to obtain meaningful values, it is necessary for the agent to try out each action in every state multiple times allowing the reinforcement updates to assign penalties or rewards based on the outcome. A stated previously, test station processes provide a safe and certain path for testing, Q-Learning can provide the impetus for migration from the “safe” path to the optimal path as the reasoning systems gains experience.

The Q-Learning equation is presented as follows [3]:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a_{t+1}))$$

where:

Q is defined as the sum of all reinforcements for the state action pair (s,a) at a given time t.

.  $\alpha$  is the learning rate parameter [0, 1].

.  $\gamma$  is the discount parameter [0, 1].

The learning rate parameter  $\alpha$  carries a significant importance in adjusting the algorithms learning behavior.  $\alpha$  controls the weighting on current and previous knowledge represented in the state-action pair values. As the learning rate parameter approaches 1, new experience outweighs previous experience. As it approaches 0, previous experience outweighs new experience. For the state – action pairs, representing task effectiveness of a maintenance action relative to a

fault,  $\alpha$  is set closer to 0, outweighing previous experience to new, which translates in assigning more importance to the maintenance task history performance.

The discount parameter  $\gamma$  controls the weighting on the new state action pair. For the current application, the parameter is fixed to a constant and provides scaling to the state-pair values, which does not affect the overall results of the Q-Learning algorithm [4].

The mapping between state and action pairs can be represented in a  $n \times m$  matrix where

$n$  represents the number of possible states

$m$  represents the number of possible actions

## Learning Feedback

Feedback provides the critical component to growth in fidelity and robustness of the self-evolving reasoner. Without explicit knowledge of maintenance action effects and test path results, little growth can be achieved. As a student learns, his progress is measured and feedback is a constant source of clarification. So too, must an evolutionary system be motivated to improve. Clever systems will strive to incorporate feedback into the normal workflow and derive benefit from process knowledge. However, input as to the cost of one decision versus another; success or failure of a recommendation; and overall usefulness of the session result will provide valuable insight for the reinforcement learning algorithm and case consolidation methods.

## REASONER VISUALIZATION

Visualization of the reasoning model provides an opportunity for “human in the loop” model growth control. The self-evolving application described here utilizes Impact’s PHM Design Tool for model viewing. Developed for the engineering of Prognostic and Health Management (PHM) systems, this tool’s XML-based model interface provides a convenient link to the reasoner models. Figure 4 illustrates a subset set of PHM component attributes contained in the PHM design software package. Evidence Sources, Maintenance Tasks and Fault Failure Modes are the most common among all other components.

In the case of classified systems, a separate symbol interpreter may be implemented to de-

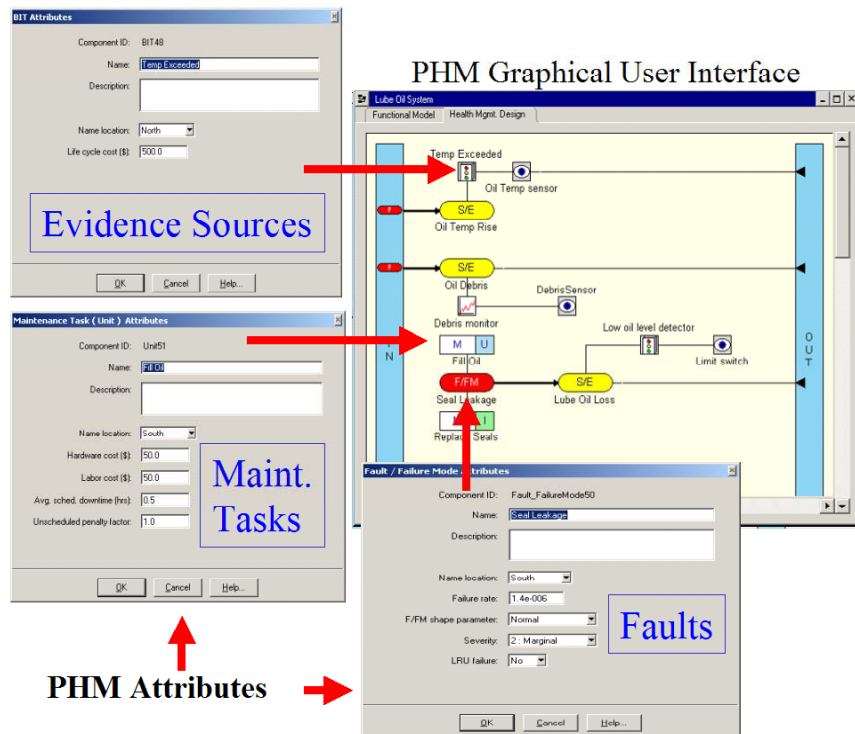


Figure 4. Visualization of the model

obfuscate the model. In all cases, human insight into case node reduction and identification of cascading failure modes may provide improvements to the overall learning system. Moreover, visualization may provide a bi-directional benefit when system developers can visualize actual failure mechanisms and related indicators to develop an understanding of unanticipated failures or complex system-level interactions.

## CONCLUSIONS

The authors have developed an approach to self-evolving closed-loop diagnostic reasoning that builds upon the sound engineering and best practices of legacy processes. Utilizing an innovative combination of existing intelligent technologies, a hybrid system has been developed that leverages current test station processes to gain a gradual understanding of system evidence and failure effect relationships. The approach has the advantage of operational capability with purely symbolic evidence representation, revealing a wide array of potential applications potentially including classified systems. Techniques, presented here, provide a

foundation for discussion and continued development of improved test and repair processes for avionics equipment.

## ACKNOWLEDGMENT

The authors wish to gratefully acknowledge the support and consultation of William Birurakis, Michael Malesich, and Bob Fox of NAVAIR; Phillip Dussault and Jay Lindsey of AMRDEC. In addition, the consultation of Dennis Hecht and Anthony Alwardt at Boeing; Richard Beers and David Goldstein at Honeywell; and Howard Savage of SCI are acknowledged and very much appreciated.

## REFERENCES

- [1] Alwardt A., et al, "Smart TPS Development and Implementation Results," AutoTestCon 2004
- [2] Byington, Kalgren, Donovan, "Portable Diagnostic Reasoning for Improved Avionics Maintenance and Information Capture and Continuity," IEEE AutoTestCon, September 2004.

- 
- [3] Harmon, M., "Reinforcement Learning – A Tutorial" WL/AAFC, WPAFB Ohio 45433
  - [4] Frank, V., "Q-Learning" IRIDIA Universite Libre de Brussels, 7-7-2003
  - [5] Borne, R.O., "Intelligent Embedded Diagnostics/Prognostics for Future Avionics Systems –An Overview," Naval Air Systems Command
  - [6] Kalgren, Byington, et. al., "Automated Data Translation for Avionics Health Management through Intelligent Evidence Streams and Databases," AHS Annual Forum, 2004