Proceedings of ASME Turbo Expo 2004
Power for Land, Sea, and Air
June 14–17, 2004, Vienna, Austria

Paper Number: GT2004-54135

PROGNOSTIC ENHANCEMENTS TO DIAGNOSTIC SYSTEMS (PEDS) APPLIED TO SHIPBOARD POWER GENERATION SYSTEMS

Carl S. Byington
Michael J. Roemer
Matthew J. Watson
Impact Technologies, LLC
2571 Park Center Blvd.
State College, PA 16801
carl.byington@impact-tek.com

Thomas R. Galie
Christopher Savage
Naval Surface Warfare Center
Carderock Division
Philadelphia, PA, 19112-5083

ABSTRACT

Numerous advancements have been made in gas turbine health monitoring technologies focused on detection, classification, and prediction of developing machinery faults and performance degradation. Existing monitoring systems such as ICAS (Integrated Condition Assessment System), which is the Navy's program of record and is deployed on many US Navy ships, employ alarm thresholds and event detection using rule-based algorithms. Adding the capability to predict the future condition (prognostics) of a machine would add significant benefit to the Navy practice. The current paper describes a framework and development process that allows more "plug 'n play" integration of new diagnostic and prognostic technologies using evolving Open System Architecture (OSA) standards. Although many modules were developed in the PEDS framework, specific gas turbine modules that focus on compressor and nozzle degradation algorithms are discussed. The modules use statistical prediction algorithms and were developed using seeded fault data generated by the Navy engineering station. The modules are fully automated, interact with the existing monitoring system, process real-time data, and utilize advanced forecasting techniques. Such an advanced prognostic capability can enable a higher level of condition-based maintenance for optimally managing total Life Cycle Costs (LCC) and readiness of assets.

NOMENCLATURE

API - Application Protocol Interface
CBM – Condition Based Maintenance
DDI - Demand Data Interface
DLL - Dynamic Linked Library
DTD - Document Type Definition
FADC - Full Authority Digital Engine Controller
GTP - Gas Turbine Performance
ICAS - Integrated Condition Assessment System
JSP - JAVA Server Page
LCC - Life Cycle Costs
MIMOSA - Machinery Information Management Open Systems Alliance
OSA - Open System Architecture
PEDS – Prognostic Enhancements to Diagnostic Systems
PDF - Probability Density Function
PHM - Prognostics and Health Management
SGML - Standard Generalized Markup Language
SXL - eXtensible Stylesheet Language
TOC – Total Ownership Cost
W3C - World Wide Web Consortium
XML - eXtensible Markup Language

C, F – Normal Distributions
N – Speed
P – Pressure
Q – Volumetric Flow
S – Weighted Coefficients
T – Temperature
y - Predicted Value

CIT – Compressor Inlet Temperature
CDT – Compressor Discharge Temperature
CDP – Compressor Discharge Pressure
TIT – Turbine Inlet Temperature

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Prognostic Enhancements to Diagnostic Systems (PEDS) Applied to Shipboard Power Generation Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Impact Technologies LLC,2571 Park Center Blvd Ste 1,State College,PA,16801**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**9**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

FMP - Fuel Manifold Pressure
FF - Fuel Flow

Φ - Normalized Cumulative Distribution
α -- Weighting Factor
γ -- Ratio of Specific Heats
σ – Standard Deviations
τ – Prediction Interval

INTRODUCTION

Various prognostics and health monitoring technologies have been developed that aid in the detection and classification of developing system faults. However, these technologies have traditionally focused on fault detection and isolation within an individual subsystem. Machinery health management system developers are just beginning to address the concepts of prognostics and the integration of anomaly, diagnostic and prognostic technologies across subsystems and systems. [1-3] Hence, the ability to detect and isolate impending faults or to predict the future condition of a component or subsystem based on its current diagnostic state and available operating data is currently a high priority research topic.

In general, health management technologies will observe features associated with anomalous system behavior and then relate these features to useful information about the system's condition. In the case of prognostics, this information relates to the condition at some future time. Inherently probabilistic or uncertain in nature, prognostics can be applied to system/component failure modes governed by material condition or by functional loss. Like diagnostic algorithms, prognostic algorithms can be generic in design but specific in terms of application. Various approaches to prognostics have been developed that range in fidelity from simple historical failure rate models to high-fidelity physics-based models.

This paper will discuss some generic prognostic implementation approaches and provide specific applications to various mechanical systems. The ability to predict the time to conditional or mechanical failure (on a real-time basis) is of enormous benefit and health management systems that can effectively implement the capabilities presented herein offer a great opportunity in terms of reducing the operations/maintenance logistics footprint and overall Total Ownership Costs (TOC) of operating systems.

Monitoring Systems and New Prognostics

The Navy's Integrated Condition Assessment System (ICAS) [4] is a tool to enable maintenance troubleshooting and planning for shipboard machinery systems. It provides data acquisition, data display, equipment analysis, diagnostic recommendations, and decision support information to operators and maintenance personnel. Additionally, ICAS links to other maintenance-related software programs to provide a fully integrated maintenance system. ICAS assesses equipment and system condition for maintenance of machinery and equipment. Through the use of permanently installed sensors, the ICAS system monitors vital machinery parameters on a continuous basis. ICAS can diagnose the operational condition of a particular piece of machinery using customer-supplied performance data linked to a logical diagnostic process.

The ICAS workstation is used for data acquisition, conditioning, performance analysis, trend and logsheet capture, and expert evaluation. Several types of data acquisition devices that process sensor output signals augment the workstation. The ICAS workstation is also responsible for providing all user interface functions and long-term data storage.

The Navy has formed an open-forum working group to establish Gas Turbine CBM [5], with the goal of planning and executing integration of CBM technologies into gas turbines on all CG & DDG class ships. Installation of FADC (Full Authority Digital engine Controller) controllers on all gas turbines in the CG & DDG classes by the Life Cycle Managers over the next 8 years will provide the hardware and computing power required for equipment health assessment and monitoring. ICAS will provide the necessary connection allowing gas turbine health monitoring systems to provide assessments and recommendations to ships crew. New algorithms developed by the Navy, industry or the other programs will be incorporated as part of the FADC. The system-wide development will incorporate ongoing and new R&D efforts into the development plan and complete system integration with ICAS. Most of the phases run concurrently and have parallel timelines.

Within these program developments and evolving environment, the Prognostic Enhancements to Diagnostics Systems (PEDS) program is focused on demonstrating prognostic enhancements using demand data interface protocols and displays using pseudo sensor inputs or simple web-based interfaces.

The approach for the PEDS program is to develop prognostic software that is modular and possesses the capability for multiple transition opportunities. The PEDS module communicates between existing elements and system enhancements using controlled proprietary interfaces or open middleware that "glue and hook" items together. The PEDS module should have the ability to interface directly with the existing database and data monitoring system, its user interface, and the decision support and logistics system using the pre-negotiated interfaces defined by the existing system. This is accomplished using system specifications, such as a Demand Data Interface (DDI) and TCP-IP as in the case of ICAS. In addition, the PEDS module should have the ability to interface directly with any system that uses OSA-CBM specifications (i.e. OSA-CBM Compliant Sensors and Processing modules) or systems that are enhanced to include the OSA-CBM specifications.

Evolving Open Systems Standards

Openness is a general concept that denotes free and unconstrained sharing of information. In its broadest interpretation, the term "open systems" applies to a systems design approach that facilitates the integration and interchangeability of components from a variety of sources. For a particular system integration task, an open systems approach requires a set of public component interface standards and may also require a separate set of public specifications for the functional behavior of the components. The development of the open-systems standards relevant to Condition-based Maintenance (CBM) and Prognostics and Health Management (PHM) development has been pursued by an International Standards Organization (ISO/TC 108/SC 5) committee, a consortium of condition monitoring companies - Machinery Information Management Open Systems Alliance (MIMOSA), and a DoD Dual-Use Science and Technology program (OSA-CBM) led by The Boeing Company. [6-8]

The MIMOSA interface standards define open data exchange conventions for sharing of static information between CBM systems (openness at the intra-system level). The goal of the OSA-CBM project was the development of an architecture (and data exchange conventions) that enables interoperability of CBM components (openness at the inter-system level). The interface set allows external clients open access to the information generated within the proprietary system solution. Alternatively, a CBM system can operate openly at the inter-system and intra-system levels. In this case the individual components are exposed at the functional component interfaces. These component interfaces offer access to the data and services supplied by the component, and provide for open information flow between components during system operation. In addition, components may be readily replaced by components with improved capability as long as they follow the same public interface standards. The architectural constraints are applied to the structure of the public interface and to the behavior of the modules. This approach allows complete encapsulation of proprietary algorithms and software and is a key enabler to prognostic module implementation.

Prognostics Approaches Considered

A health management or CBM system that possess prognostics implies the ability to predict a future condition or capability. Inherently probabilistic or uncertain in nature, prognostics can be applied to system/component failure modes governed by material condition or by functional loss. Similar to diagnostic algorithms, prognostic algorithms can be generic in design but specific in terms of application. Within the health management system architecture, the Prognostic Module function is to intelligently utilize diagnostic results, experienced-based information and statistically estimated future conditions to determine the remaining useful life or failure probability of a component or subsystem. Prognostic reasoners can range from reliability-based to empirical feature-based to completely model-based.

Some of the information that may be required depending on the type of prognostics approach used in the system include:

- Engineering Model and Data
- Failure History
- Past Operating Conditions
- Current Conditions
- Identified Fault Patterns
- Transitional Failure Trajectories
- Maintenance History
- System Degradation Modes
- Mechanical Failure Modes

Examples of prognostics approaches that have been successfully applied for different types of problems include:

Experience-Based Prognostics: Use statistical reliability to predict probability of failure at any point in time. May be augmented by operational usage information.

Evolutionary/Statistical Trending Prognostics: Multi-variable analysis of system response and error patterns compared to known fault patterns.

Artificial Intelligence Based Prognostics: Mechanical failure prediction using reasoners trained with failure data.

State Estimator Prognostics: System degradation or diagnostic feature tracking using Kalman filters and other predictor-corrector schemes.

Model-Based or Physics of Failure Based Prognostics: Fully developed functional and physics-of-failure models to predict degradation rates given loads and conditions.

COMPRESSOR PERFORMANCE PROGNOSTICS

Fouling degradation of gas turbine engine compressors causes significant efficiency loss, which incurs operational costs through increased fuel usage or reduced power output. Scheduling maintenance actions based upon predicted condition minimizes unnecessary washes and saves maintenance dollars. The effect of the various maintenance tasks (washing and overhaul) on gas turbine engine efficiency is shown in the figure below.

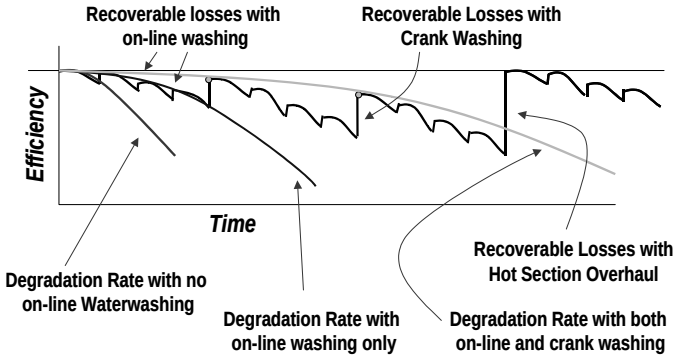

Figure 1 - Effects of Washing on Efficiency and Overhaul

Currently, washes are performed on a preventative schedule of 50 hours for on-line washes and 500 hours for crank washes. This maintenance task is performed with no engineering assessment of conditional need or optimal time to perform. In addition to the loss of availability and maintenance time incurred, unnecessary washes generate an environmental impact with the disposal of used detergent. Clearly, operating with a module that assesses condition and predicts the time to wash more appropriately would benefit the Navy.

Data and Symptoms for Development

The compressor wash prognostic model was developed using data from fouling tests taken at NSWCC in Philadelphia, PA and is an example of evolutionary prognostics approach. It is based upon specific system features and models for compressor efficiency. In order to simulate the amount of salt the typical Navy gas turbine is exposed to on a normal deployment, a 9% salt solution was injected into the engine intake. Over the course of the entire test (3 days) approximately $0.0057m^3$ of salt was used to induce compressor degradation at four different load levels (1/3, 2/3, standard and full load levels or "bells"). This method of testing was performed on both Allison 501 and LM2500 Units. Figure 2 shows a borescope image of the salt deposits on the LM2500 1st stage blading.

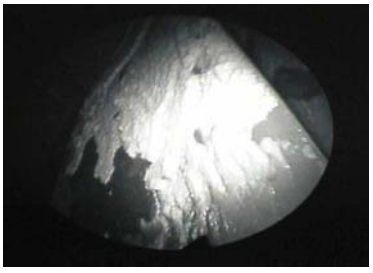

Figure 2 - Borescopic Image of Salt Deposits: 1st stage

During the testing, several critical parameters were monitored and their response to degradation was tended. Table 1 contains the measured parameters with their units and ranges (Shaft RPM and N_{gg} are for the LM2500 testing only)

Parameter	Units	Ranges
N_{gg}	RPM	0→9575
Q_{fuel}	GPM	0 → 97
TIT	°F	0 →2000
CDT	°F	0 →1468
CDP	psig	0 →300
CIT	°F	0 →500
Load	k-lbf	0 →300
Shaft RPM	RPM	0 → 274
$P_{barometric}$	inHg	

Table 1 – Recorded Parameters from Digital Control System

When a compressor undergoes fouling, several key performance factors are affected. The most sensitive of these factors is the compressor capacity or referred mass flow. (Peltier et al, [9]) This occurs due to loss of capacity that comes from throat blockage and increases in roughness on the suction side of the blading. Unfortunately, in most practical naval applications, compressor capacity is not reliably determinable. The compressor inlet temperature (CIT), outlet temperature (CDT), inlet total pressure (CIP_T) and discharge total pressure (CDP_T) can typically be used to find compressor efficiency. (Boyce [8])

$$\eta_{adb} = \frac{\left[\left(\frac{CDP_T}{CIP_T}\right)^{\gamma-1/\gamma} - 1\right]}{\left[\left(\frac{CDT}{CIT}\right) - 1\right]} \quad (1)$$

Within the developed approach, data preprocessor algorithms examine the unit's operating data and automatically calculate key corrected performance parameters such as pressure ratios and efficiencies at specific load levels. The techniques employed and processing in the module are shown in detail in Figure 3.

A probabilistic-based technique was developed that utilizes the known information on how measured parameters degrade over time to assess the current severity of parameter distribution shifts and project their future state. The parameter space is populated by two main components. These are the current condition and the expected degradation path. Both are multi-variate Probability Density Function (PDFs) or 3-D statistical distributions. As shown, the consideration of uncertainty is carried through the entire process to produce a confidence in the prediction.

Once the statistical performance degradation path is realized, along with the capability to assess current degradation severity, we needed to implement the predictive capability. The actual unit-specific fouling rate is combined with historical fouling rates with a double exponential smoothing method. This time series technique weights the two most recent data points over past observations. The following equations give the general formulation. (Bowerman [10])

$$S_T = \alpha y_T + (1-\alpha) S_{T-1} \quad (2)$$

$$S^{[2]}_T = \alpha S_T + (1-\alpha) S^{[2]}_{T-1} \quad (3)$$

$$\hat{y}_{T+\tau}(T) = \left(2 + \frac{\alpha\tau}{(1-\alpha)}\right) S_T - \left(1 + \frac{\alpha\tau}{1-\alpha}\right) S^{[2]}_T \quad (4)$$

Analysis of the degradation requires simulation to predict the range of conditions that might exist given measurement and modeling uncertainties. This is accomplished using a Monte Carlo simulation, which cycles through parameter PDFs created from mean values and 2-sigma uncertainties. The resulting distribution is the range of Time-to-Wash predictions. Appropriate statistical confidence intervals can be applied to identify the mean predicted value. This estimate can also be updated with a weighted fusion of the predicted value and the historical degradation level derived from fouling data.

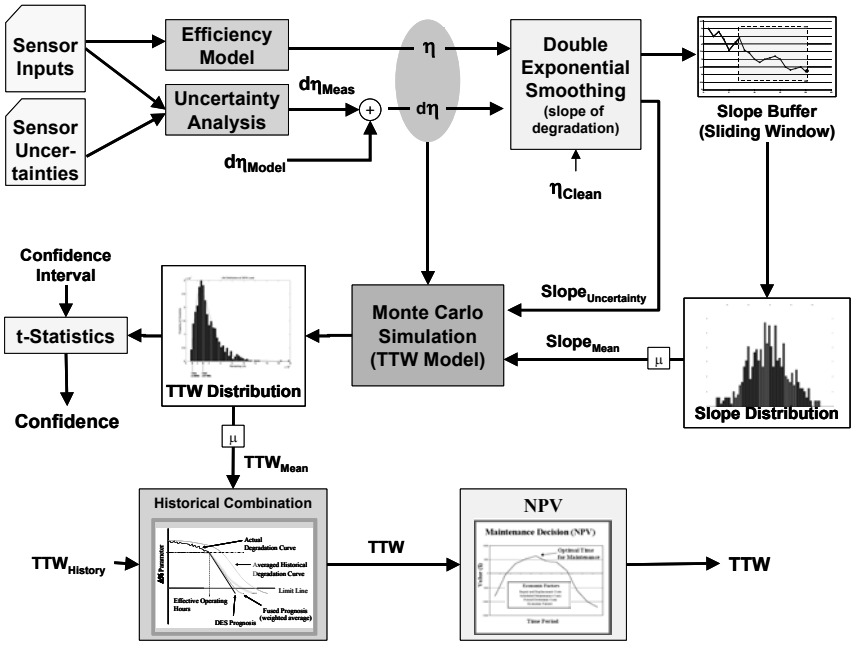

Figure 3 - Data Processing Flow Chart for Compressor Performance Prognostics

FUEL NOZZLE CLOGGING/START PROGNOSIS MODULE

The fleet of US Navy Allison 501 K-17 and K-34 gas turbines uses either pilot type or air assist fuel nozzles. A clogged fuel nozzle reduces the efficiency of the combustion process and can create potentially damaging hot spots in the combustor and turbine sections. At startup, this is especially true to the extent that "hot starts" or "no starts" may be produced. Experience has shown that the pilot-type nozzles have a tendency to accrue carbon deposits (coking) in the pilot tube and the nozzle orifice causing improper spray patterns that contribute to hot and slow gas turbine starts and combustor liner damage. Due to high turbine inlet temperatures, hot starts can also cause damage to turbine hot section components. While apparently more reliable, degraded air assist fuel nozzles can also lead to the same consequences. Figure 4 shows examples of clean and severely clogged fuel nozzles.

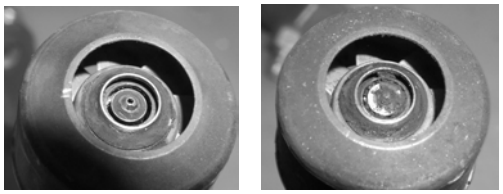

Figure 4 - Clean and Clogged (right) Delavan Nozzles

Data and Symptoms for Development

The diagnosis of fuel nozzle clogging was demonstrated using an analysis of gas turbine sensor values. Features were identified from the Fuel Manifold Pressure (FMP), Turbine Inlet Temperature (TIT), Engine speed (RPM), and Fuel Flow (Wf). The baseline data, in which the nozzles were known to be clean, was the basis signature. Multiple other indicative data sets were collected with progressive clogging conditions. A number of diagnostic indicators were developed to diagnose nozzle clogging. The first indicator captured the time delay between the end of the FMP increase and the start of the TIT increase, as defined by the baseline (multiplied by 100 for scaling purposes). The average difference between the actual FMP values at a given RPM and the expected FMP values for that RPM was also used. This value was only calculated for the FMP points associated with a start event. Another indicator was the max FF/FMP ratio, which is the maximum Fuel Flow (FF) to FMP ratio for the start. Finally, the slope of the TIT line was also used. These features appear to be reliable indicators of fuel nozzle clogging that can provide ample warning prior to full start.

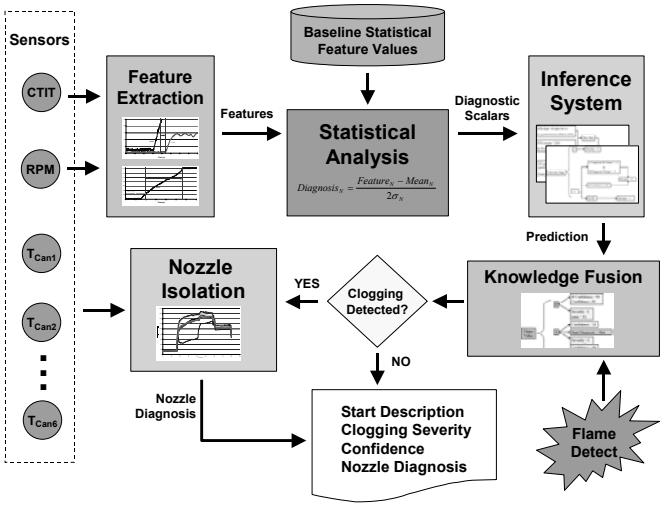

Figure 5 – Processing Flow of Gas Turbine Component Prognostics Module

Figure 5 shows the processing flow of the fuel nozzle algorithm. Using this methodology, the module analyzes the GTG during start-up to determine if the start was normal, hot, slow, or a combination of hot and slow. If the start was determined to be abnormal, a confidence and severity is computed, and an analysis is performed to determine which nozzle is clogged.

Prognostic adaptations focus on the automated interpretation of the nozzle clogging projected in time and a recommended change threshold based upon the features identified. The prognostic output should be a recommended number of starts or operational hours for a nozzle change.

PROGNOSTICS MODULE ARCHITECTURE

The requirements associated with a prognostic module's interaction with the existing system were used to identify an architecture for prognostics module developments. The generic prognostic approach that was developed, along with its interaction with existing systems, can be seen below. As seen in the figure, the module architecture consists of a number of elements, each of which performs a different function in the operation of the module. The purpose of each of these elements is discussed next. In addition, the figure demonstrates how each element interfaces with the existing system. A review of the major functions of these PEDS elements is provided.

Prognostic Director – The Prognostic Director stores configuration files for the dynamic link libraries, verifies the organization of data and inter-element communication using OSA objects and flows, and orchestrates the overall operation of the module. It also provides the API (Application Protocol Interface) and middleware information for the elements to interact with external sensors or databases. XML interfaces are anticipated to accomplish this function, but others are also possible with "bridge" software. If information is not available from an OSA data source, then the prognostics director will reformat it for inter-element use and appropriate "wrappers" would be deployed in the director.

Initialization Element - The Initialization Element initializes shared memory, provides the startup parameters, launches the other elements and updates the logging and interface.

Data Processing Element – The Data Processing Element accepts raw data acquired and processes features not available from the existing system. The data processing element will be necessary for "vertical" prognostics modules that input raw data and perform usage or failure prognostics directly. Advanced feature processing of vibration data is a typical example of the function of this element.

Diagnostic Assessment Element - The Diagnostic Assessment Element links with the fault detection and diagnostic processes external to the prognostic module. This allows the prognostic module to consider the outputs of these processes in its prediction.

Mission Upload Element - The Mission Upload Element provides a means to input data to the prognostic module on future environmental conditions and mission plans. This will enable mission planning to be considered in the prognosis.

Prognosis Elements – The Prognosis Elements use information gathered and processed by the previous elements to make a prognosis about the system or component being monitored. They will possess the run-time DLLs (dynamically linked libraries) that perform the predictions on performance degradation and component faults.

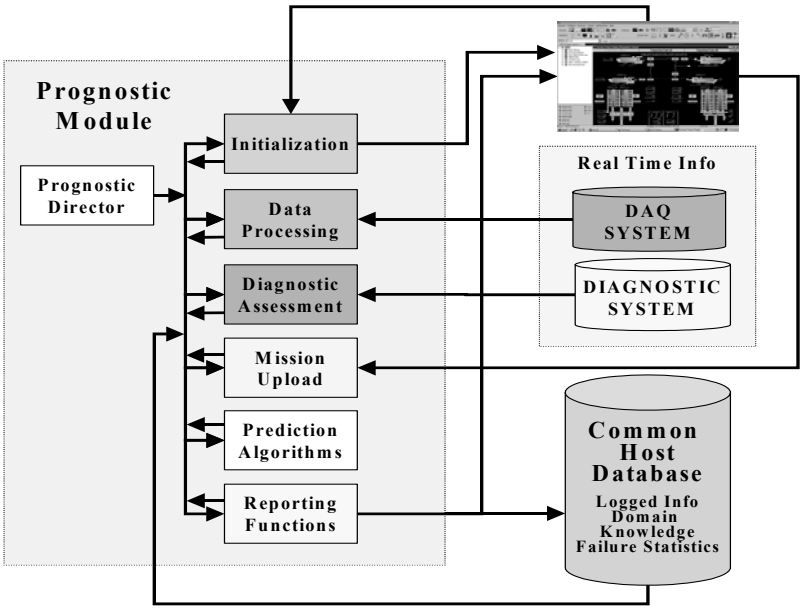

Figure 6 - Prognostic Approach and Interface Diagram

PEDS DEVELOPMENT WITH OSA

The PEDS module implementation consists of translating the engineering code (MATLAB® in this case) into an implemented "plug and play" module. The final compiling of the code is somewhat platform specific, but for Windows based applications the code can be written in C++ and compiled as a DLL (dynamic linked library).

The module currently supports OSA-CBM compliant XML (eXtensible Markup Language) and other documented data structures. Figure 7 shows the two possible deployment opportunities for the Gas Turbine Performance (GTP) module: ICAS and Tiger™, and the elements of the module that are re-usable between the two approaches. This is possible because the code has been written to allow for a number of different input possibilities. Flags are set in the initialization element that tells the module which inputs to expect for the current implementation. Therefore, this modularity of design allowed easy modification of the GTP module to interface with Sermatech's Tiger™ gas turbine software. This resulted in faster development time and lower costs.

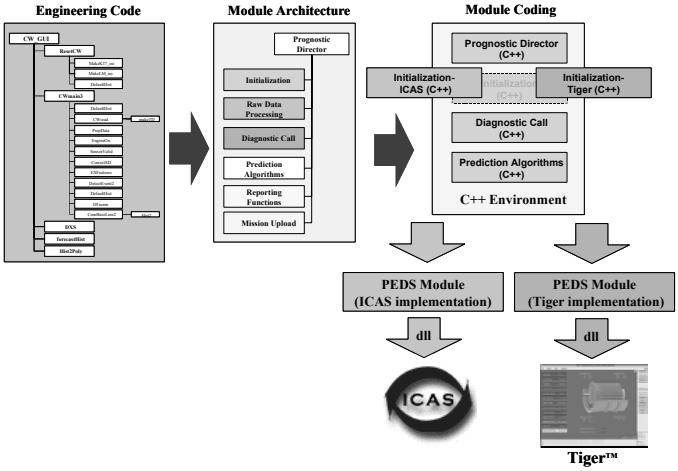

Figure 7 – Producing PEDS modules from Engineering Code

The use of XML is considered a significant enabler to the open systems development. XML is an extension of Standard Generalized Markup Language (SGML) and has been a World Wide Web Consortium (W3C) recommendation since February 1998. XML describes information content and information relationships using a metadata structure. The structure of the XML document is defined by a user-generated Document Type Definition (DTD) or schema. The display format of an XML document is also specified by the user/generator of the document using eXtensible Stylesheet Language (XSL) and transformation sheets. Thus, the same document can be displayed in multiple ways depending on the consumer of the information.

A web-based demonstration interface was created to demonstrate the operation of the GTP module (Figure 8). The HSI and design concept of the stand-alone module demonstration provide a means to illustrate the functionality of the GTP module and explore the decision variables that affect the TTW recommendation. It also provides an example of a web-based implementation of a PEDS module with separable prognostics code and HSI display. The primary decision variables affecting the TTW cost benefit are: recommended wash time, allowable efficiency degradation, total fuel cost, and the cost of the maintenance action.

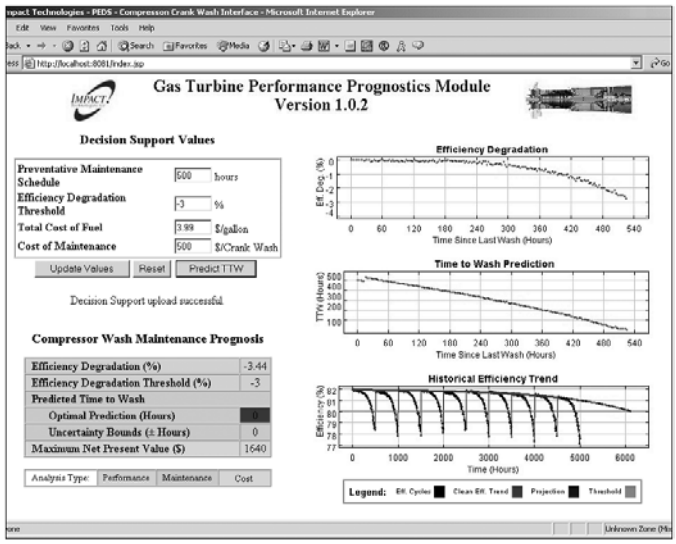

Figure 8– Web-based Gas Turbine Performance Prognostics

The Gas Turbine Performance interface operates using a combination of JAVA Server Pages (JSP), XML, and XSL. By linking a series of JSP pages using frames, the HSI allows the user to input cost variables to be used in the GTP module. After the user inputs cost variables in the text boxes, the *Update Values* button generates an XML files to hold these values, outputs a status message to the user, and activates the *Run Module* button. This button is used to activate the GTP module, the XML output of which is displayed in the right-hand frame of the HSI using an XSL stylesheet. This HSI and design concept provides a means to illustrate the functionality of this performance prognostics module and explore the decision variables that affect the TTW recommendation.

In order to facilitate the web-demonstration and to improve cross platform reusability of the module, a Java wrapper and JNI interface were developed for the GTP module. This interface enabled the execution of the module from the Java Server Page and controlled operation of the module for demonstration. A data simulation program was also developed to stimulate the module for demonstration and integration testing. The Java wrapper was used to read data from the simulation and call the GTP module at the appropriate time.

Similar to the GTP module, a web-based demonstration was constructed for the GTC module (Figure 9). The HSI and design concept of this demonstration illustrates the functionality of the module and provides a means for pursuing transition of the module. It also serves as another example of a web-based implementation of a PEDS module with separable prognostics code and HSI display.

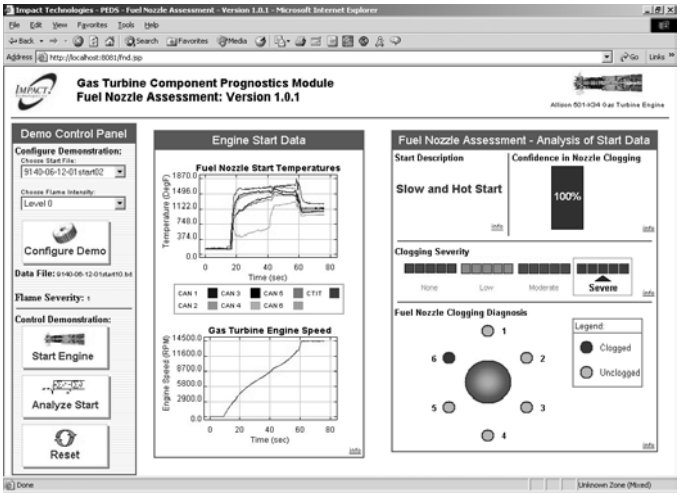

Figure 9 – GTC Module Web-Demonstration Interface

The Gas Turbine Performance interface also operates using a combination of JAVA Server Pages (JSP) and XML/XSL documents. The HSI allows the simulation of a generator start and subsequent analysis by the module using a series of linked JSP pages embedded in frames. The demonstration begins by pressing the *Start Engine* button, located in far left side of the HSI. This simulates the start of the gas turbine generator and start data, including temperatures and engine speed, are displayed in the plots in the middle of the HSI. The user can click on these plots to show a larger representation of the plot for easier viewing. The start can be analyzed by pressing the *Analyze Start* button, which executes the GTC module. Upon completion of its analysis, the module will produce a number of XML documents containing the results of its assessment. These analysis results are displayed in the right frame using a number of XSL stylesheets. As seen in the figure, the HSI displays the results of the Start Description, Confidence, Clogging Severity, and Nozzle Isolation analyses described previously.

PEDS DEMONSTRATION IN ICAS

A major advantage of the PEDS architecture is its modularity and code re-usability. This was evident in the adaptation of the module for use with the Navy's ICAS system. In this case, the legacy system did not support the OSA-CBM schema for data transfer, but Navy personnel have developed several other means to directly interface to the inference engine. These were implemented and the same basic modules were "wrapped" to provide a direct means to perform prognosis in ICAS. Impact developed a JNI interface to act as a buffer between ICAS and the PEDS modules and permit the use of these modules within Java. This interface was developed to control data streams and calls to the PEDS module.

In order to enhance the ICAS capability, a means to pull data out of ICAS and write prognostic results was needed. This was accomplished using the Demand Data Interface and Transmission Control Protocol (TCP/IP).

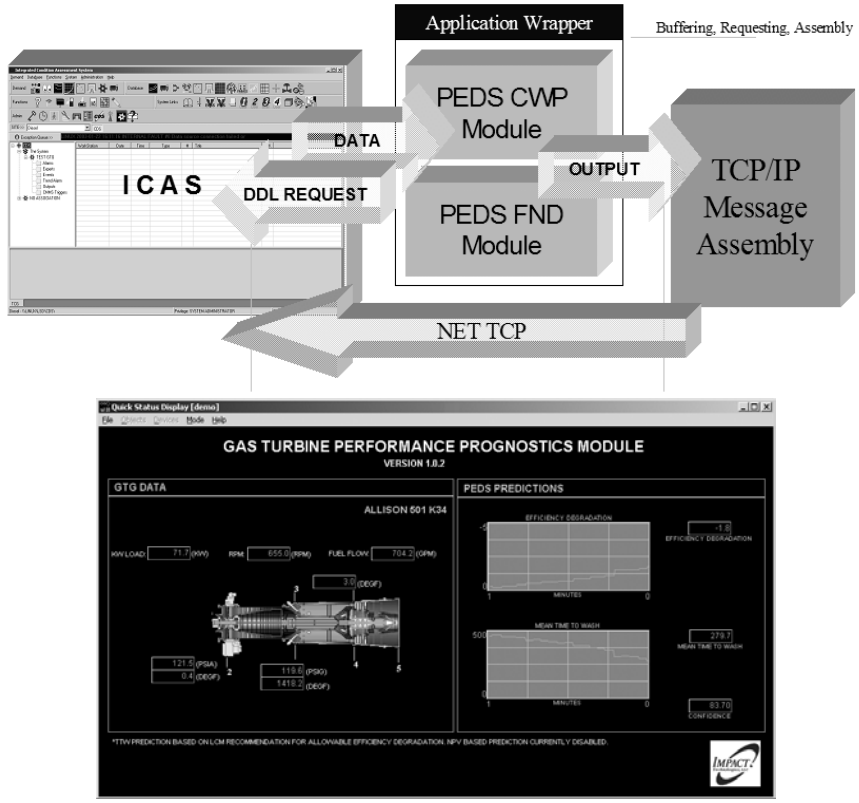

Figure 10 – PEDS Wrapper Implementation with ICAS and Quick Display Page for Prognostic Results

The DDI depicted was designed to allow real-time data retrieval from ICAS at a rate of 1 Hz. It takes the form of a Windows DLL and has a standard C interface that was coded in the wrapper program. A TCP/IP protocol is one of a number of ways to write data back into ICAS for display, logging, etc. Implementation of this approach required the construction of a TCP data interface server and a database table within ICAS. The data interface is required to enable the TCP/IP client/server interface and the database table is needed to translate incoming data streams to appropriate sensor channels. Alternative means to accomplish this interface and pass multiple parameters back to ICAS are available but not discussed here.

Impact successfully implemented both the demand data and TCP/IP interfaces and demonstrated the operation of each across a Local Area Network. In order to test the ICAS interfaces described above, a Gas Turbine Generator simulation was created to generate data that can be used to populate ICAS. A custom CDS of the Allison 501 K-34 engine that was previously developed at Impact with the help of Russ Leinbach (Code 9521) was utilized in the simulation.

CONCLUSIONS

This paper discussed many concepts associated with prognostic module development under the PEDS (Prognostic Enhancements to Diagnostic Systems) program. A brief review of prognostic approaches, implementation issues (including current OSA developments), and an example of gas turbine performance prognostics was provided. Data availability, dominant failure or degradation mode of interest, modeling and system knowledge, accuracies required and criticality of the application are some of the variables that determines the choice of prognostic approach. The OSA-CBM architecture described has been successfully adapted and implemented in marine, aircraft, and industrial environments.

The prognostic compressor wash and nozzle algorithms were demonstrated successfully for the 501K34 gas turbine generators as described. The predictions have been validated with 'ground truth' degradations in the Navy land-based test facilities, and the open system integration was demonstrated and works well. The OSA implementations were developed using primarily XML and those used in current Navy monitoring applications. Preliminary work has also demonstrated the feasibility to adapt these algorithms for the LM2500 gas turbine generator. Although field experience has been limited to date, these algorithms and architecture are applicable to a range of DoD and industrial gas turbine monitoring systems and testing on more units is planned for full acceptance.

Ultimately the ability to predict the time to conditional or mechanical failure (on a real-time basis) is an enormous benefit and machinery health management systems that can effectively implement the capabilities presented herein offer a great opportunity in terms of reducing the maintenance and logistics footprint and overall Life Cycle Costs (LCC) of operating systems.

ACKNOWLEDGMENTS

The authors thank Mitch Lebold and Mike Thurston for their input regarding open systems architectures developments during earlier phases of this work. Moreover, this work benefited from the solid contributions of many colleagues at Impact Technologies and The Penn State Applied Research Lab. The assistance of Russ Leinbach (Code 9521) was integral to the ICAS interface development. Jack McGroarty, John Scharschan, and Daniel Caguiat provided valuable insight and direction to the gas turbine algorithm development. The support for this work by the DoD SBIR program office through a Phase II contract (N00167-01-C-0064) is also gratefully acknowledged. This work was supported by Naval Surface Warfare Center Carderock Division through Impact Technologies under the Prognostics Enhancement to Diagnostic Systems project. The content of the information does not necessarily reflect the position or policy of the Government, and no official endorsement should be inferred.

REFERENCES

1. Roemer, M. J., et. al, "Prognostic Enhancements to Naval Condition-Based Maintenance Systems," *Improving Productivity Through Applications of Condition Monitoring*, 55th Meeting of the Society for Machinery Failure Prevention Technology, April, 2001.
2. Roemer, M. J. and Kacprzynski, G.J., "Advanced Diagnostics and Prognostics for Gas Turbine Engine Risk Assessment," Paper 2000-GT-30, ASME and IGTI Turbo Expo 2000, Munich, Germany, May 2000.
3. Mathew, J., "Standards in Condition Monitoring", Condition Monitoring in Perspective - Proceedings of the First $(CM)^2$ Forum, Ballarat, ISBN 0-7326-0496-6, 129-134, 1994.
4. Thurston, M. and Lebold, M., Standards Developments for Condition-Based Maintenance Systems, *Improving Productivity Through Applications of Condition Monitoring*, 55th Meeting of the Society for Machinery Failure Prevention Technology, April 2001.
5. MIMOSA website [http://www.mimosa.org/]
6. OSA-CBM website [http://www.osa-cbm.org/]
7. Roemer, M. J., and Ghiocel, D. M., "A Probabilistic Approach to the Diagnosis of Gas Turbine Engine Faults" Paper 99-GT-363, ASME and IGTI Turbo Expo 1999, Indianapolis, Indiana, June 1999.
8. Boyce, Meherwan, Gas Turbine Engineering Handbook, Gulf Publishing Company, 1995
9. Peltier, R. V., Swanekamp, R. C., "LM2500 Recoverable and Non-Recoverable Power Loss" ASME Cogen-Turbo Power Conference, Vienna, Austria, August 1995.
10. Bowerman, Bruce L. and O'Connel, Richard T., Forecasting and Time Series, Duxbury Press, 1993.