

# Distributed Multi-Algorithm Diagnostics and Prognostics for US Navy Ships

**George D. Hadden**

**Peter Bergstrom**

Honeywell Laboratories  
3660 Technology Drive  
Minneapolis, MN 55418

**Bonnie Holte Bennett**

Knowledge Partners of Minnesota  
9 Salem Lane, Suite 100  
Saint Paul, MN 55118-4700

**George Vachtsevanos**

School of Electrical and Computer Engineering  
The Georgia Institute of Technology  
Atlanta, GA 30332-0250

**Joe Van Dyke**

(formerly at PredictDLI)  
Systems Analysis and Software Engineering  
9665 Timberlane Place  
Bainbridge Island, WA 98110

## Abstract

Honeywell and its teammates (PredictDLI, Knowledge Partners of Minnesota, the Georgia Institute of Technology, York International, and WM Engineering) have developed a distributed shipboard system to perform diagnostics and prognostics on mechanical equipment (e.g. engines, generators, and chilled water systems) for the Office of Naval Research (ONR). This Condition Based Maintenance (CBM) system (called *MPROS* for *Machinery Prognostics/Diagnostics System*) consists of MEMS and conventional sensors on the machinery, local intelligent devices (called *Data Concentrators*), and a centrally located subsystem (called the *PDME* for *Prognostics, Diagnostics, Monitoring Engine*) which is designed so that it can run under shipboard monitoring systems such as ICAS (Integrated Condition Assessment System). The system uses an open, object-oriented approach with a well-defined API so that additional diagnostic and prognostic algorithms can be incorporated in a “plug and play” manner.

MPROS includes and augments periodic vibration analysis by collecting data continuously from vibration and other sensors, including temperature, pressure, current, voltage, and others. These data streams are integrated as necessary in the Data Concentrators (data fusion). Individual prognostic and diagnostic algorithms can reside in either the Data Concentrators or the PDME. A second level of integration (Knowledge Fusion) occurs in the PDME. At this level, using both Dempster Shafer Evidence Combination and a mechanism to fuse time-to-failure estimates, the conclusions of the diagnostic and prognostic reasoning algorithms are fused to yield the best possible analysis.

The authors gratefully acknowledge the support of the Office of Naval Research, grant number N00014-96-C-0373.

## 1. INTRODUCTION

The ONR CBM system, called MPROS (for Machinery Prognostics and Diagnostics System), is a distributed, open, extensible architecture for hosting multiple on-line diagnostic and prognostic algorithms.

Since these algorithms have overlapping areas of expertise, they may sometimes disagree about what is wrong with the machine. They may also reinforce each other by reaching the same conclusions from similar data. In these cases, another subsystem, called *Knowledge Fusion* (KF), is invoked to make some sense of these conclusions.

MPROS is distributed in the following sense: Devices called *Data Concentrators* (DC) are placed near the ship's machinery. Each of these is a computer in its own right and has the major responsibility for diagnostics and prognostics. The prognostic and diagnostic algorithms run on the DC. Conclusions reached by these algorithms are then sent over the ship's network to a centrally located machine containing the other part of our system – the *Prognostic/Diagnostic/Monitoring Engine* (PDME). KF is located in the PDME. Also in the PDME is the *Object Oriented Ships Model* (OOSM).

The MPROS program had two phases. The first phase had MPROS installed and running in the lab. During the second phase, we extended MPROS's capability somewhat and installed it on the Navy Hospital Ship, the USNS MERCY, in San Diego.

*Mission* – Our central mission in this project was to design a *shipwide* CBM system to predict remaining life of all shipboard mechanical equipment. However, implementation of such a system in its entirety would have been much too ambitious. In light of this, we chose to illustrate the general principles of our design by implementing it in a specific way on the Centrifugal Chilled Water System. The result of this

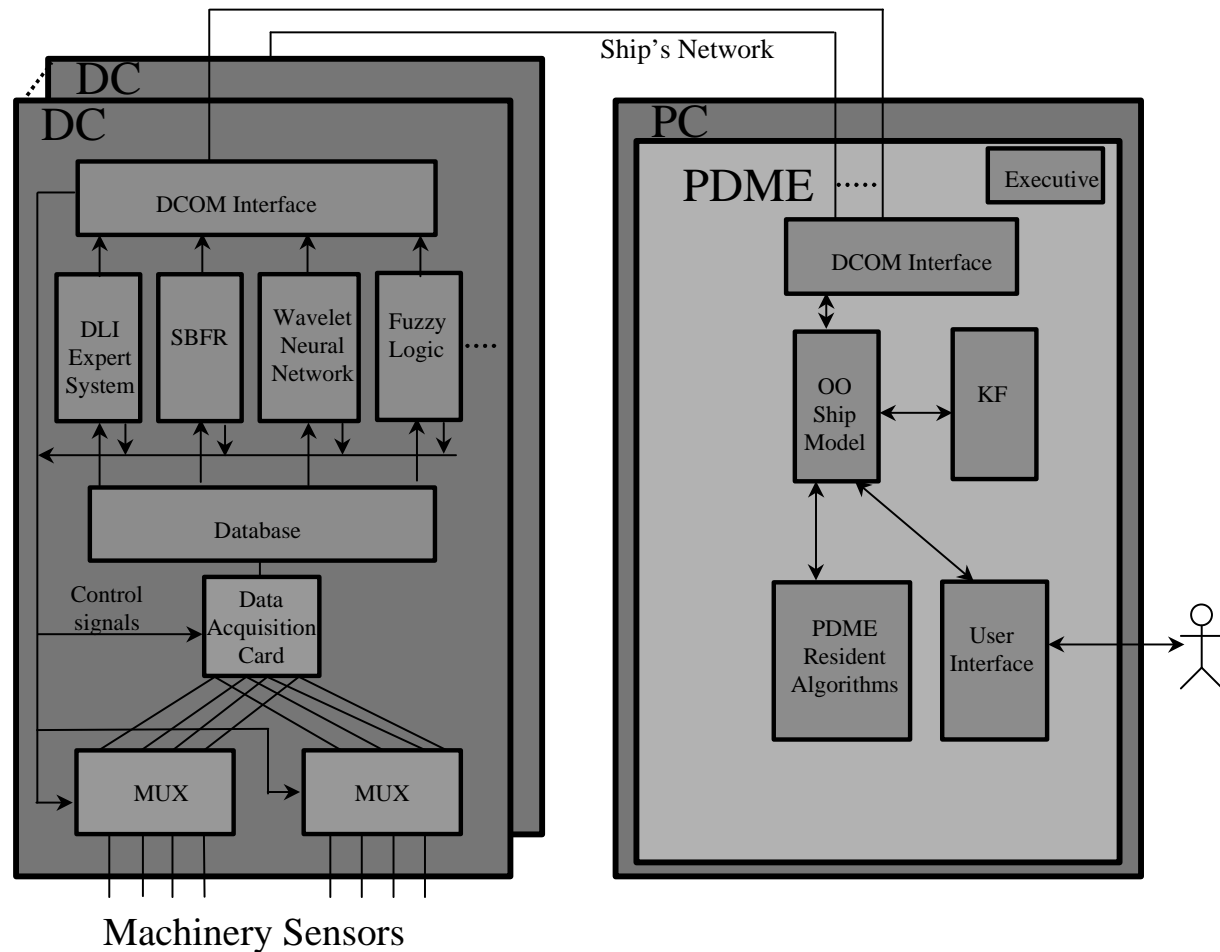


Figure 1 The MPROS System

philosophy is that occasionally we chose a more general way of solving a problem over a “centrifugal chiller-specific” solution.

*Why Centrifugals?* – There were two main reasons for our choice of centrifugal chillers: System complexity and commercial applicability. These A/C systems combine several rotating machinery equipment types (i.e. induction motors, gear transmissions, pumps, and centrifugal compressors) with a fluid power cycle to form a complex system with several different parameters to monitor. This dictated the requirement for a correspondingly complex and versatile monitoring system. Dynamic vibration signals must be acquired using high sampling rates and complex spectrum and waveform analysis. Slower changing parameters such as temperatures and pressures must also be monitored, but at a lower frequency and can be treated as scalars rather than vectors as with vibration spectra. All of these monitored parameters and analysis techniques are combined using a versatile diagnostic system. The final product has the inherent capability of diagnosing not just the whole A/C system, but each of its parts as well, making it a potentially very useful tool for monitoring any pump, motor,

gearset, or centrifugal compressor in the fleet.

Secondly, the selection of A/C system as the subject will provide a high probability of commercial applicability of the resultant monitoring system. There are a great deal of facilities industrial, military, commercial, and institutional that use large centrifugal chiller based A/C systems throughout the US and the world.

## 2. SOFTWARE

Figure 1 shows a diagram of the MPROS system. Here we describe the various parts.

### *PDME Software*

The Prognostic, Diagnostic, Monitoring Engine (PDME) is the logical center of the MPROS system. Diagnostic and prognostic conclusions are collected from DC-resident

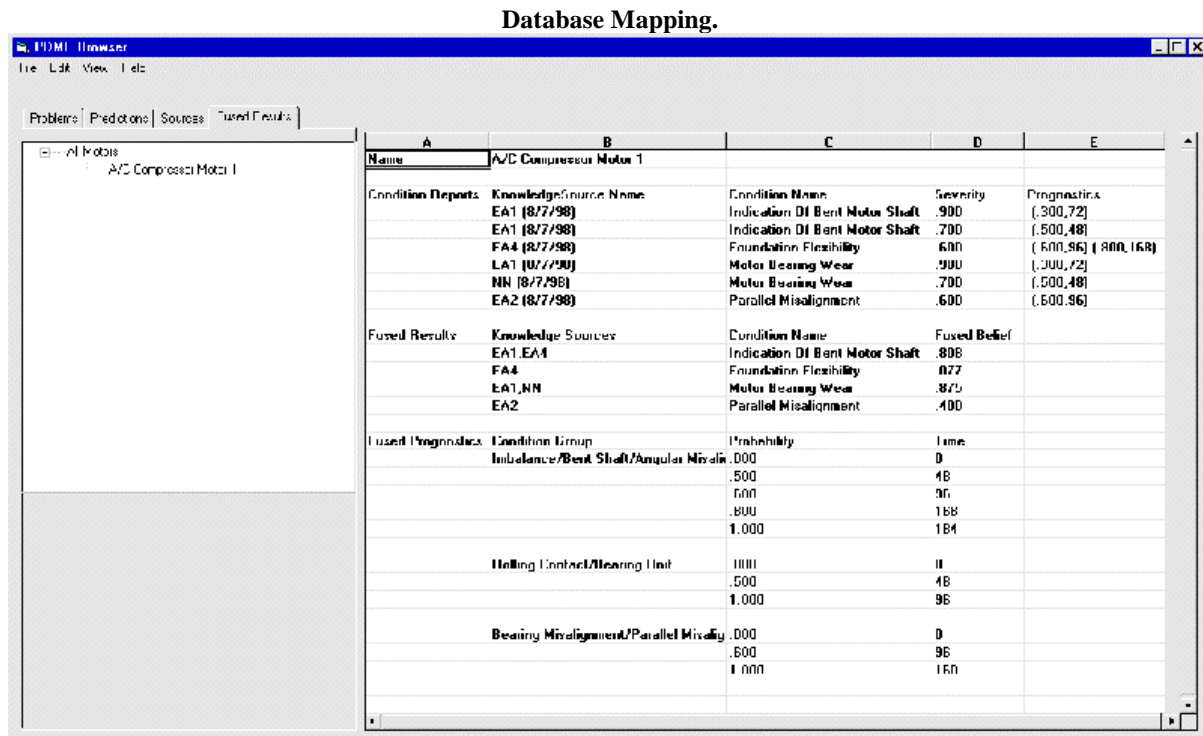


Figure 2 MPROS User Interface

algorithms as well as PDME-resident algorithms. Fusion of conflicting and reinforcing source conclusions is performed to form a prioritized list for the use of maintenance personnel.

The PDME is implemented on a Windows NT platform as a set of communicating servers built using Microsoft's Component Object Model (COM) libraries and services. Choosing COM as the interface design technique has allowed us to build some components in C++ and others in Visual Basic, with an expected improvement in development productivity as the outcome. Some components were prototyped using Microsoft Excel and we continue to use Excel worksheets and macros to drive some testing of the system. Communications between DC components and PDME components depend on Distributed COM (DCOM) services built into Microsoft's operating systems.

**User interface** – As shown in Figure 2, an interface to the MPROS conclusions has been built. The sample screen shown indicates that for machine A/C Compressor Motor 1, six condition reports from four different knowledge sources (expert systems) have been received, some conflicting and some reinforcing.

After these reports are processed by the Knowledge Fusion component, the predictions of failure for each machine condition group are shown at the bottom of the screen.

This display is updated as new reports arrive at the PDME and are accumulated in the OOSM.

**Object Oriented Ship Model** – Entities in the OOSM are modeled as objects with properties and relationships to other entities. Some of the OOSM objects represent physical entities such as sensors, motors, compressors, decks, and ships while other OOSM objects represent more abstract items such as a failure prediction report or a knowledge source. Some common properties include name, manufacturer, energy usage, capacity, and location. Common relationships include part-of, kind-of, connected-to, and energy flow.

Diagnostic and prognostic conclusions are stored in the OOSM – both those of the individual algorithms and those reached by KF. It also serves as blackboard by providing a means of communication among the individual algorithms.

**Knowledge Fusion** – Knowledge fusion is the coordination of individual data reports from a variety of sensors. It is higher level than pure "data fusion" which generally seeks to correlate common-platform data. Knowledge fusion, for example, seeks to integrate reports from acoustic, vibration, oil analysis, and other sources, and eventually to incorporate trend data, histories, and other components necessary for true prognostics.

The knowledge fusion components must be able to accommodate inputs that are incomplete, time-disordered, fragmentary, and that have gaps, inconsistencies, and contradictions. In addition, knowledge fusion components must be able to collate, compare, integrate, and interpret

data from a variety of sources. To do this, it must provide both inference control that accommodates a variety of input data and fusion algorithms with the ability to deal with disparate inputs.

Knowledge fusion follows this procedure:

1. New reports arriving to the PDME are posted in the OOSM.
2. New reports posted in the OOSM generate “new data” messages to the knowledge fusion components.
3. The knowledge fusion components access the newly arrived data from the OOSM. They perform knowledge fusion of diagnostic reports and knowledge fusion of prognostic reports.
4. Conclusions from the knowledge fusion components are posted to the OOSM and presented in user displays in the graphical user interface.

To date, two levels of knowledge fusion have been implemented: one for diagnostics and an extension for prognostics (remaining life).

Our approach for implementing knowledge fusion for diagnostics uses Dempster-Shafer belief maintenance for correlating incoming reports. This is facilitated by use of a heuristic that groups similar failures into logical groups.

Dempster-Shafer theory is a calculus for qualifying beliefs using numerical expressions. For example, given a belief of 40% that *A* will occur and another belief of 75% that *B* or *C* will occur, it will conclude that *A* is 14% likely, *B* or *C* is 64% likely, and assign 22% of belief to unknown possibilities. This maintenance of the likelihood of unknown possibilities is both a differentiator and a strength of Dempster-Shafer theory. It was chosen over other approaches (e.g., Bayes nets) because the others require prior estimates of the conditional probability relating two failures – data not yet available for the shipboard domain.

The system was augmented by heuristically collecting similar failures into logical groups. This facilitates processing and streamlines operation because Dempster-Shafer analysis looks at each failure in light of every other possible failure and is required to produce the likelihood of unknown possibilities. In the MPROS case, this is inadequate because it would assume mutual exclusivity of failures. However, this is not a realistic assumption. There can, in fact, be several failures at one time, and two or more of them might be independent of one another. Thus, we developed the concept of logical groups of failures. Failures that are all part of the same logical groups are related to each other (for example, one group might be electrical failures, while another group would be lubricant failures, etc.). Moreover, failures within a group might be mistaken for one another, so any two of them are logically related and should share probabilities when they are both under consideration. Note that this does not preclude multiple failures within a group all being suspected concurrently; it

simply ensures that they are tracked and weighted correctly.

The second level of knowledge fusion combines time to failure estimates. Time to failure is represented in our system as a list of one or more time points, probability pairs, called the ‘prognostic vector’. For example, the prognostic vector with the single member ‘((3 months, .1))’ indicates that the system has a 10% likelihood of failure within 3 months. The prognostic vector ‘((2 weeks, .1) (1 month, .5) (2 months, .9))’ indicates a likelihood of failure of 10% within 2 weeks, 50% within 1 month, and 90% within 2 months.

Our approach to the fusion of prognostics information is to combine the lists, taking the most conservative estimate at any given time period, interpolating a smooth curve from point to point, and extrapolating the worse case from the entry with the longest interval. For example, suppose we have a prognostic for a given component that indicates it will perform well for 3 months, and then experience some trouble making it as likely to fail as not by 4 months and almost surely to fail within 5 months. The prognostic vector for this case is ((3 months, .01) (4 months, .5) (5 months, .99)). Suppose further that we need to combine this with another report showing that the same component will experience some small trouble at 4-1/2 months. This prognostic vector is ((4.5 months, .12)). Under our current approach, we ignore the second report and stick with the first, which is more conservative. If, however, the second report indicates a much higher likelihood of failure, say ((4.5 months, .95)), then this report would dominate and the extrapolation of the curve beyond this point would indicate an even earlier demise of the component than the first prognostic vector.

**Interfaces provided.** One of the goals of the MPROS system is to encourage the incorporation of the appropriate set of algorithms supplying diagnostic and prognostic conclusions based on similar, overlapping, or entirely disjoint sensor readings. At the same time, we recognized that these diverse results must be unified into a meaningful report to the system’s users. To this end, a standard protocol has been defined for reporting failure predictions to the PDME for fusion and display.

The general incoming report format may contain the following data fields (not all reports need use all fields):

1. KnowledgeSourceID: The unique MPROS object ID for the instance of the diagnostic/prognostic algorithm generating the report.
2. SensedObjectID: The unique MPROS object ID for the sensed object to which this report applies.
3. MachineConditionID: The unique MPROS object ID for the diagnosed machine condition (usually a failure mode).
4. Severity: Numeric value in the range 0.0 to 1.0 indicating relative severity of machine condition to operation. Maximal severity is 1.0.

5. **Belief:** Numeric value in the range 0.0 to 1.0 indicating belief that this diagnosis is true. Maximal belief is 1.0.
6. **Explanation:** An optional text string providing a human-readable description of the diagnosis.
7. **Recommendations:** An optional text string providing a human-readable description of the recommended actions to take.
8. **Timestamp:** The timestamp for when this report should be considered effective.
9. **Additional Information:** An optional text string providing human-readable additional information.
10. **Prognostic vector:** A vector of time point, probability pairs indicating projected likelihood of failure (as described above).

Diagnostic knowledge fusion generates a new fused belief whenever a diagnostic report arrives for a suspect component. This updates the belief for that suspect component and for every other failure in the logical group for that component. It also updates the belief of 'unknown' failure for the logical group for that component.

Prognostic knowledge fusion generates a new prognostic vector for each suspect component whenever a new prognostic report arrives.

**Future directions for knowledge fusion.** Several high-level control extensions are under consideration for future extensions. First, multilevel data are represented by the OOSM. We are not currently exploiting this fully. For example, we could reason about the health of a system based on the health of a constituent part. Currently, only the parts are tracked. Second, spatial reasoning using the OOSM could lead us to fuse information about spatially related components. One example of a spatial relation is proximity. For example, a device might be vibrating because a component next to it is broken and vibrating wildly. Another example is flow. Flows are relationships that represent fluid flow through the system (one component passing fouled fluids on to other components downstream), electrical flow, or mechanical flow of physical energy. Third, temporal reasoning components could be implemented to scrutinize failure histories and provide better projections of future faults as they develop.

Two other future directions for knowledge fusion are the refinement of specific knowledge fusion components for diagnostics and for prognostics. For example, Bayes nets seem to be a promising approach to diagnostic knowledge fusion when causal relations and a priori relationships can be teased out of historical data. Prognostic knowledge fusion could be improved with the addition of techniques from the analysis of hazard and survival data. These approaches scrutinize history data to refine the estimates of life-cycle performance for failures, and the refined inputs to the prognostic analysis should yield better projections of future failures.

*Resident Algorithms* – As the reader can see from Figure 1, the PDME has the capability to host prognostic and diagnostic algorithms. Some reasons for placing the algorithms in the PDME rather than the DC include: the algorithm requires data from widely separate parts of the ship, the algorithm can reason from PDME resident components (a model-based diagnostic and prognostic system, for instance, might use only the OOSM), and so on.

Although we provide this capability in our general architecture, our system as currently implemented does not place any diagnostic/prognostic algorithms in the PDME – all of them run in the data concentrators.

### *Data Concentrator (DC) Software*

*Scheduler* – At the heart of the DC software is an event scheduler. This software component runs the show by organizing all necessary events. For example, the standard vibration test and analysis is executed routinely by the scheduler. To do a standard vibration test, the scheduler first triggers execution of vibration data acquisition component, and when that operation is complete the scheduler fires off the vibration analysis component and then triggers communication of the results. In similar fashion, the scheduler conducts wavelet and neural network testing and analysis and state based feature recognition and fuzzy logic routines to collect and analyze process variables.

Each of the components extracts information from and store data in the DC database, which is configured as a database server and can be accessed by client PC's on the network. In this way, the PDME or any other client can command the scheduler to conduct another test and analysis routine.

*Data Base* – Central to the operation of the DC is an open architecture ODBC compliant relational database designed to store all of the instrumentation configuration information, machinery configuration information, test schedules, resultant measurements, diagnostic results, and condition reports. The database design is a commercially available database already field tested and proven effective in many industrial facilities.

*Prognostic/Diagnostic Algorithms* – The current version of MPROS has four sets of prognostic/diagnostic algorithms: the PredictDLI Expert System, Wavelet Neural Networks, Fuzzy Logic, and State Based Feature Recognition. A brief description of each of these follows:

PredictDLI's (a company in Bainbridge Island, Washington that has a Navy contract to do CBM on shipboard machinery) vibration based expert system has been adapted to run in a continuous mode. These algorithms excel when the Chiller is performing in steady state.

Wavelet-Neural Network (WNN) diagnostics and prognostics developed by Professor George Vachtsevanos

and his colleagues at Georgia Tech. This technique is, like PredictDLI's, aimed at vibration data, however, complementing the PredictDLI algorithm, their algorithm excels at drawing conclusions from transitory phenomena rather than steady state data.

Fuzzy Logic diagnostics and prognostics also developed by Georgia Tech which draws diagnostic and prognostic conclusions from non-vibrational data.

State Based Feature Recognition (SBFR), a Honeywell-developed embeddable technique that facilitates recognition of time-correlated events in multiple data streams. Originally developed for Space Station Freedom, this technique has been used in a number of NASA related programs.

### 3. VALIDATION

One question we are often asked is "How are you going to prove that your system does what you say it does?" This question, as it turns out, is a quite difficult one. The problem is that we are developing a system that we claim will predict failures in devices, and that in real life, these devices fail relatively rarely. In fact, for any one failure mode, it is entirely possible that the failure mode (although valid) may never have occurred on any piece of equipment on any ship in the fleet! We have a number of answers to the question:

- We are still going to look for the failure modes. We have a number of installed data collectors both on land and on ships. In addition, PredictDLI is collecting time domain data for a number of parameters whenever their vibration-based expert system predicts a failure on shipboard chillers. This will give us data that we can use to test our system.
- As Honeywell upgrades its air conditioning systems to be compliant with new non-polluting refrigerant regulations, older chillers become obsolete. We have managed to acquire one of these chillers that Honeywell replaced. It was shipped to York, and we have completed a program to collect data from this chiller through a carefully choreographed set of destructive tests.
- Seeded faults are worth doing. Our partners in the Mechanical Engineering Department of Georgia Tech are seeding faults in bearings and collecting the data. These tests have the drawback that they might not exhibit the same precursors as real-world failures, especially in the case of accelerated tests.
- Honeywell, York, PredictDLI, NRL, and WM Engineering have archives of maintenance data that we will take full advantage of in constructing our prognostic and diagnostic models.

- Similarly, these partners have human expertise that we are able to tap in building our models.

Although persuasive, these answers are far from conclusive. The authors would welcome any input on further validation of a failure prediction system.

### 4. MERCY INSTALLATION

Our system is complete and has been installed in two places: in the basement of Honeywell Laboratories on a Carrier 19DK chiller and in San Diego on the USNS MERCY's (T-AH-019) #1 AC plant – a York 363 Ton chiller. The system has detected and reported a number of problems including a bad bearing on one of the seawater pumps. MPROS also diagnosed a problem on one of the chilled water pumps that led to the discovery of a previously unknown failure mode – an electrical path through the bearings that led to pitting.

The Mercy was chosen for a number of reasons:

1. It is in a climate that was more likely to require cooling during the winter shipboard test phase of our centrifugal chiller prognostics and diagnostics system.
2. It was likely to stay stationary and not put out to sea (as, for instance, a carrier would).
3. It contains pieces of equipment that are the target of our prognostics and diagnostics efforts.
4. We have a good working relationship with the crew.

Earlier versions of our system were installed on the following ships: the USNS COMFORT (T-AH-020), the USS CONSTELLATION (CV-64), and the USS ABRAHAM LINCOLN (CVN-72).

### 5. BIBLIOGRAPHY

Bennett, B.H. and Hadden, G.D. (1999) Condition-based maintenance: algorithms and applications for embedded high performance computing. *Proceedings of the 4th International Workshop on Embedded HPC Systems and Applications (EHPC'99)*.

Bristow, J., Hadden, G.D., Busch, D., Wrest, D., Kramer, K., Schoess, J., Menon, S., Lewis, S. and Gibson, P. (1999) Integrated diagnostics and prognostics systems. *Proceedings of the 53<sup>rd</sup> Meeting of the Society for Machinery Failure Prevention Technology* (invited).

Echaz, J. and Vachtsevanos, G. (1996) Elliptic and radial wavelet neural networks. *Proceedings of the 2<sup>nd</sup> World Automation Congress*, Montpellier, France, May 27–30.

Edwards, T.G. and Hadden, G.D. (1997) An autonomous diagnostic/prognostic system for shipboard chilled water plants. *Proceedings of the 51<sup>st</sup> Meeting of the Society for*

*Machinery Failure Prevention Technology.*

Hadden, G.D., Bennett, B.H., Bergstrom, P., Vachtsevanos, G. and Van Dyke, J. (1999) Machinery diagnostics and prognostics/condition based maintenance: a progress report. *Proceedings of the 53<sup>rd</sup> Meeting of the Society for Machinery Failure Prevention Technology.*

Hadden, G.D., Bennett, B.H., Bergstrom, P., Vachtsevanos, G. and Van Dyke, J. (1999) Shipboard machinery diagnostics and prognostics/condition based maintenance: a progress report. *Proceedings of the 1999 Maintenance and Reliability Conference (MARCON99).*

Hadden, G.D., Edwards, T.G. and Van Dyke, J. (1998) Shipboard machinery condition based maintenance. *Proceedings of the Maintenance and Reliability Conference (MARCON98).*

Hadden, G.D., Edwards, T.G. and Van Dyke, J. (1996) Condition based maintenance for shipboard machinery. *Proceedings of the 67<sup>th</sup> Shock and Vibration Symposium.*

Hadden, G.D., Nelson, K.S. and Edwards, T. (1994). State-based feature recognition. *Proceedings of the 17<sup>th</sup> Annual AAS Guidance and Control Conference.*

Kang, H., Cheng, J., Kim, I. and Vachtsevanos, G. (1991) An application of fuzzy logic and Dempster-Shafer theory to failure detection and identification. *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, England, 1555–1560.

Mufti, M. and Vachtsevanos, G. (1995) An intelligent approach to fault detection and identification. *Proceedings of the American Control Conference.*

Nelson, K.S. and Hadden, G.D. (1994) Real-time feature recognition in medical data. *Proceedings of the AAAI 1994 Spring Symposium, Artificial Intelligence in Medicine.*

Vachtsevanos, G., Kang, H., Cheng, J. and Kim, I. (1992) On the detection and identification of axial flow compressor instabilities. *American Institute of Aeronautics and Astronautics, Journal of Guidance, Control, and Dynamics*, **15**(5), 1216–1223.

**George D. Hadden** is a Senior Research Fellow at the Honeywell Laboratories and a recipient of the H. W. Sweatt award – Honeywell's highest technical honor – for a preventive maintenance expert system. He also serves as adjunct faculty at the University of Minnesota where he occasionally teaches a course on Advanced Artificial Intelligence Programming. Currently, he is principal investigator of a project to perform condition based maintenance on Jet Engines. Dr. Hadden received his Ph.D. and Masters Degree in Electrical Engineering from the University of Illinois where he specialized in the study of Artificial Intelligence and his Bachelors Degree in Electrical



Engineering from Purdue University. He has published over thirty-five technical papers and has been invited to give several presentations.

**Peter Bergstrom** is a Principal Research Scientist in the Software Tools Group at the Honeywell Laboratories in Minneapolis, MN. He is active in research and development in System Health Management. He received his B.A. in Computer Science from Macalester College in St. Paul, Minnesota. Since joining Honeywell Laboratories in 1991, he has developed applications in Honeywell's avionics, industrial, building and home control business units.



**Bonnie Holte Bennett** is an Associate Professor in the Graduate Programs in Software at the University of St. Thomas in St. Paul, Minnesota. She is the founder and director of the Artificial Intelligence/High Performance and Parallel Computing Lab. She is also a principal with Knowledge Partners of Minnesota, Inc. ([www.kpmi.com](http://www.kpmi.com)).



She consults extensively with Fortune 200 companies in the areas of knowledge-based systems and knowledge management. Previously, she spent 14 years at Honeywell Laboratories in Minneapolis, Minnesota. She holds Ph.D. and M.S. degrees in Computer Science from the University of Minnesota, and a Bachelors degree in Computer Science and English from the College of St. Thomas.

**George Vachtsevanos** is Professor of Electrical Engineering at the School of Electrical and Computer Engineering of the Georgia Institute of Technology. He received the BEE degree from the City College of New York, the MEE degree from New York University and a Ph.D. degree in EE from the City University of New York. He is directing the Intelligent Control Systems laboratory at Georgia Tech and teaches courses and conducts research in intelligent systems, fuzzy logic and neural networks, diagnostics and prognostics and manufacturing systems. His R&D activities are sponsored by industry and government agencies. He has published over 250 technical papers in his area of expertise and is the inventor or co-inventor of six U.S. patents. He is a senior member of IEEE and has



chaired the Association for Electronics Manufacturing Board of Advisors of SME. He serves as the associate editor of the International Journal of Intelligent and Robotic Systems.

**Joe Van Dyke**, formerly of PredictDLI, has been involved in the predictive maintenance field for 13 years managing software development projects, teaching machinery vibration analysis, and performing a broad range of mechanical and monitoring systems analysis work. Currently his software development consulting company is working to develop new and specialized vibration analysis components for use in industrial machinery vibration monitoring. Joe Van Dyke holds an M.S. degree in Mechanical Engineering from the University of Washington and a P.E. license in the State of Washington.

