Application Challenges: System Health Management for Complex Systems

George D. Hadden [1], Peter Bergstrom[1], Tariq Samad[1] ,Bonnie Holte Bennett[2] , George J. Vachtsevanos[3] , and Joe Van Dyke[4]

[1]Honeywell Technology Center, 3660 Technology Drive, Minneapolis, MN 55418
george.d.hadden@htc.honeywell.com
[2]Knowledge Partners of Minnesota, Inc., 9 Salem Lane, Suite 100,
St. Paul, MN 55118-4700
bbennett@kpmi.com
[3]The Georgia Institute of Technology, School of Electrical and Computer Engineering,
Atlanta, Georgia 30332-0250
gjv@ece.gatech.edu
[4]Systems Analysis and Software Engineering, 253 Winslow Way West, Bainbridge Island,
Washington, 98110
joevandyke@Predict-DLI.com

Abstract. System Health Management (SHM) is an example of the types of challenging applications facing embedding high-performance computing environments. SHM systems monitor real-time sensors to determine system health and performance. Performance, economics, and safety are all at stake in SHM, and the emphasis on health management technology is motivated by all these considerations. This paper describes a project focusing on condition-based maintenance (CBM) for naval ships. *Condition-based maintenance* refers to the identification of maintenance needs based on current operational conditions. In this project, system architectures and diagnostic and prognostic algorithms are being developed that can efficiently undertake real-time data analysis from appropriately instrumented machinery aboard naval ships and, based on the analysis, provide feedback to human users regarding the state of the machinery – such as its expected time to failure, the criticality of the equipment for current operation.

1 Introduction

Although some aspects of system operation, such as feedback control, are by now widely automated, others such as the broad area of system health management (SHM) still rely heavily on human operators, engineers, and supervisors. In many industries, SHM is viewed as the next frontier in automation.

System health management has always been a topic of significant interest to industry. Only relatively recently, however, have the numerous aspects of health management begun to be viewed as facets of one overall problem. The term itself has gained currency only recently. We now understand SHM as encompassing all issues related to off-nominal operations of systems – including equipment, process/plant, and enterprise. As for the capabilities that fall under the SHM label, the following are particularly notable:

- Fault detection: identifying that some element or component of a system has failed.
- Fault identification: identifying *which* element has failed.
- Failure prediction: identifying elements for which failure may be imminent and estimating their time to failure.
- Modelling and tracking degradation: quantifying gradual degradation in a component or the system.
- Maintenance scheduling: determining appropriate times for preventive or corrective operations on components.
- Error correction: estimating 'correct' values for parameters, the measurements of which have been corrupted.

Technologists are seeking to exploit advances in diverse fields for developing SHM solutions. As might be expected, the variety and complexity of problems that SHM encompasses preclude any single-technology answers. Hardware, software, and algorithmic technologies are all required and are being explored. An SHM solution can require a hardware architecture design, integrating sensors, actuators, computational processors, and communication networks. Different algorithmic techniques may be needed for signal processing, including Fourier and wavelet transforms and time series models. Artificial intelligence methods such as expert systems and fuzzy logic can be helpful in allowing human expertise and intuition to be captured. There is also increasing interest in fundamental modelling, especially in failure mode effects analysis (FMEA), a systematic approach for identifying what problems can potentially occur with products and processes. Finally, software architectures are required to manage the multiple devices, data streams, and algorithms. With Internet-enabled architectures, an SHM system can be physically distributed across large distances.

1.1 Challenges in system health management

Our successes in capturing common failure mechanisms has resulted in safer, more reliable, and more available systems. An interesting corollary is that we are now seeing failure modes that were rarely seen before. The lack of empirical data or experiential knowledge in such cases renders many methods unusable. Other types of knowledge must be relied upon in such cases, generally based on a human expert's understanding of system operation.

Another failing with many conventional methods for fault identification is that they assume that faults occur singly. Surprising relationships can occur among various failure modes. A fault in one device may cause problems in otherwise unrelated machines that depend on it for their input (perhaps separated by several intervening devices). Compound faults often do not have independent symptoms, and predicting or diagnosing multiple faults is not simply a matter of dealing with each separately.

Even when there is a single fault, its symptoms will be masked by any number of additional symptoms generated by logically upstream and downstream subsystems. Also, SHM must deal with the large differences in the time scales. Vibration data from a motor may need to be collected at nearly a megahertz for shaft balance problems to be detectable, whereas flooding in a distillation column is a phenomenon that occurs on a time scale of many minutes. System architectures and algorithms, that can deal with these extremes of sampling rates, are needed and not readily available.

1.2 Condition-Based Maintenance for Naval Ships

This project, supported by the Office of Naval Research of the U. S. Department of Defence, is focusing on condition-based maintenance (CBM) for naval ships. *Condition-based maintenance* refers to the identification of maintenance needs based on current operational conditions. In this project, system architectures and diagnostic and prognostic algorithms are being developed that can efficiently undertake real-time data analysis from appropriately instrumented machinery aboard naval ships and, based on the analysis, provide feedback to human users regarding the state of the machinery – such as its expected time to failure. Using these analyses, ship maintenance officers can determine which equipment is critical to repair before embarking on their next mission – a mission that could take the better part of a year.

1.2.1 MPROS Architecture

The development of the CBM system, called MPROS (for Machinery Prognostic and Diagnostic System), 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 *Mercy* in San Diego.

MPROS is a distributed, open, extensible architecture for hosting multiple on-line diagnostic and prognostic algorithms. Additionally, our prototype contains four sets of algorithms aimed specifically at centrifugal chilled water plants. These are:

1. PredictDLI's (a company in Bainbridge Island, Washington, that has a Navy contract to do CBM on shipboard machinery) vibration-based expert system adapted to run in a continuous mode.
2. State-based feature recognition (SBFR), an Honeywell Technology Center (HTC)-developed embeddable technique that facilitates recognition of time-correlated events in multiple data streams.
3. Wavelet Neural Network (WNN) diagnostics and prognostics developed by Professor George Vachtsevanos and his colleagues at Georgia Tech. This technique

is aimed at vibration data; however, unlike PredictDLI's, their algorithm excels at drawing conclusions from transitory phenomena.

4. Fuzzy logic diagnostics and prognostics also developed by Georgia Tech that draws diagnostic and prognostic conclusions from nonvibrational data.

Since these algorithms (and others we may add later) have overlapping areas of expertise, they may sometimes disagree about what is ailing 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. We use a technique called *Dempster-Shafer Rules of Evidence* to combine conclusions reached by the various algorithms. It can be extended to handle any number of inputs.

MPROS is distributed in the following sense: Devices called *Data Concentrators* (DCs) 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. Except for Knowledge Fusion, the algorithms described above 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 Ship Model* (OOSM). The OOSM represents parts of the ship (e.g., compressor, chiller, pump, deck, machinery space) and a number of relationships among them (e.g., part-of, proximity, kind-of). It also serves as a repository of diagnostic conclusions – both those of the individual algorithms and those reached by KF. Communication among the DCs and the PDME is done using Distributed Common Object Module (DCOM), a standard developed by Microsoft.

1.2.2 Data Concentrator hardware

The DC hardware (Figure 1 shows the HTC-installed DC) consists of a PC104 single-board Pentium PC (about 6 in. x 6 in.) with a flat-screen LCD display monitor, a PCMCIA host board, a four-channel PCMCIA DSP card, two multiplexer (MUX) cards, and a terminal bus for sensor cable connections. The operating system is Windows 95™, and there are connections for keyboard and mouse. Data is stored via DRAM. The DC is housed in a NEMA enclosure with a transparent front door and fans for cooling. Overall dimensions are 10 in. x 12 in. x 4 in. The system was built entirely with commercial off-the-shelf components with the exception of the MUX cards, which are a PredictDLI hardware subcomponent, and the PCMCIA card, which was modified from a commercial two-channel unit to meet the needs of the project.

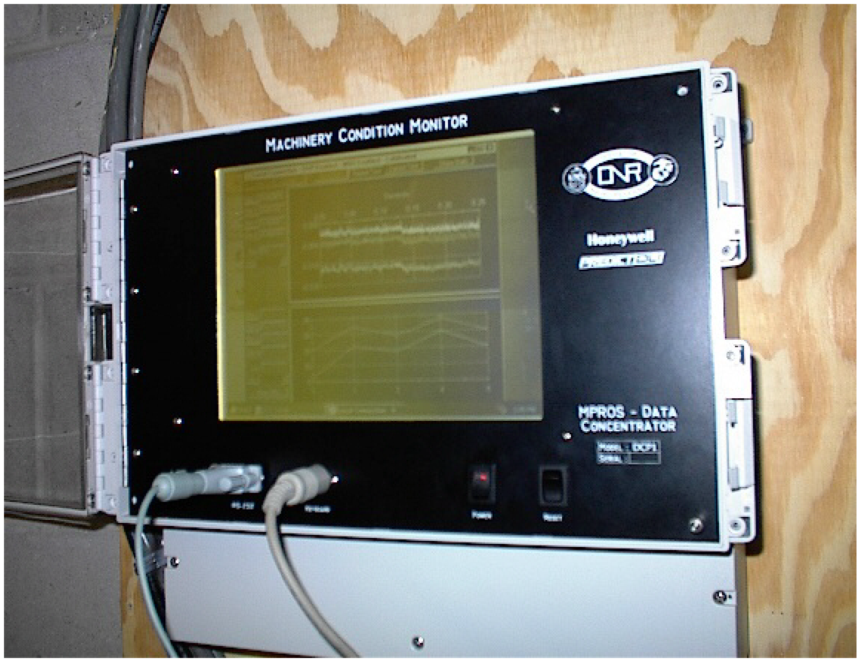

Figure 1 **Data concentrator installed at HTC**

2 MPROS Software

Figure 2 shows a diagram of the MPROS system. The PDME consists entirely of software and runs on any sufficiently powerful Windows NT machine. A potentially large number (on the order of a thousand) DCs are installed on the ship and report diagnostic and prognostic conclusions to the PDME over the ship's network. In the following, we describe the various software parts of the system.

2.1 PDME

The PDME is the logical center of the MPROS system. Diagnostic and prognostic conclusions are collected from DC-resident as well as PDME-resident algorithms. Fusion of conflicting and reinforcing source conclusions is performed to form a prioritized list for use by 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 and PDME components depend on Distributed COM (DCOM) services built into Microsoft's operating systems.

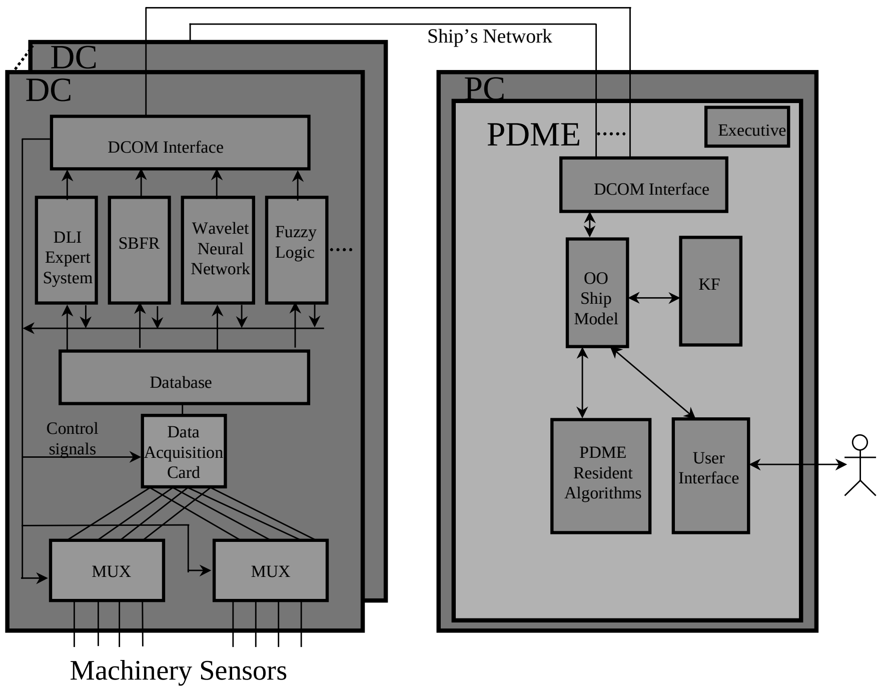

Figure 2 The MPROS system

2.2 Knowledge fusion

Knowledge fusion is the co-ordination 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.

Implementation To date, two levels of knowledge fusion have been implemented: one for diagnostics and one for prognostics.

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.

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.

3 Validation

A question we are often asked is, ‘How are you going to prove that your system can really predict failures?’ This question, as it turns out, is quite difficult to answer. The problem is that we are developing a system we claim will predict failures in devices, and that in real life, these devices fail relatively rarely. We have several answers to this 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 several parameters whenever their vibration-based expert system predicts a failure on shipboard chillers.
- As Honeywell upgrades its air conditioning systems to be compliant with new nonpolluting refrigerant regulations, older chillers become obsolete and are replaced. We have managed to acquire one of these chillers and are now constructing a test plan to collect data from this chiller.
- 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, the Naval Research Laboratory, and WM Engineering, have archived maintenance data that we will take advantage of.

Although persuasive, these answers are far from conclusive. The authors would welcome any input on how to validate a failure prediction system.

4 Conclusions

In the not too distant past, automation was employed largely to manage systems under nominal operating conditions. The realm of automation rarely extended to abnormal conditions – people were expected to handle these. Whether it was equipment failure, severe environmental disturbances, or other sorts of disruptions, the responsibility for predicting and diagnosing faults and returning the system to normal operation rested squarely on human staff. Developers of control systems and their applications were concerned about these issues only to the extent that they needed to provide the appropriate information and decision support to operators, engineers, and supervisors. The actual prognosis, diagnosis, and remedial actions were generally outside the scope of automation.

We have succeeded in our original mission almost too well, and this success has led to a broadening of our ambitions for automation and control systems. This has happened even as the scale and complexity of the physical systems – whether naval ships or commercial buildings or factories – have dramatically increased.

As might be expected, problem complexity translates to solution complexity. For instance, the more time we have to plan our response before a failure occurs, the better off we are – catastrophic failures can be avoided, human safety can be maximized, repair actions can be combined, and so on. To increase this time, we must find new ways to access data that we have not sensed before. In addition, we have to construct software that derives prognostic and diagnostic conclusions from increasingly subtle correlations among the sensed data.

5 Acknowledgment

The authors gratefully acknowledge the support of the Office of Naval Research, grant number N00014-96-C-0373. Joe Van Dyke participated in this project while employed at Predict DLI.

References

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 53rd Meeting of the Society for Machinery Failure Prevention Technology* (invited).

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 53rd 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).*