The Negative Information Problem in Mechanical Diagnostics

David L. Hall
Robert J. Hansen
Derek C. Lang

The Pennsylvania State University
Applied Research Laboratory
Post Office Box 30
State College, Pennsylvania 16804-0030
United States of America

Abstract
Condition-based maintenance (CBM) is an emerging technology which seeks to develop sensors and processing systems aimed at monitoring the operation of complex machinery such as turbine engines, rotor craft drives trains, or industrial equipment. The goal of CBM systems is to determine the state of the equipment (i.e., the mechanical health and status), and to predict the remaining useful life for the system being monitored. The success of such systems depends upon a number of factors including: (1) the ability to design or use robust sensors for measuring relevant phenomena such as vibration, acoustic spectra, infrared emissions, oil debris, etc.; (2) real time processing of the sensor data to extract useful information (such as features or data characteristics) in a noisy environment and to detect parametric changes which might be indicative of impending failure conditions; (3) fusion of multi-sensor data to obtain improved information beyond that available to a single sensor; (4) micro and macro level models which predict the temporal evolution of failure phenomena; and finally, (5) the capability to perform automated approximate reasoning to interpret the results of the sensor measurements, processed data, and model predictions in the context of an operational environment. The latter capability is the focus of this paper. Although numerous techniques have emerged from the discipline of artificial intelligence for automated reasoning (e.g., rule-based expert systems, blackboard systems, case-based reasoning, neural networks, etc.), none of these techniques are able to satisfy all of the requirements for reasoning about condition-based maintenance. This paper provides an assessment of automated reasoning techniques for CBM and identifies a particular problem for CBM, namely, the ability to reason with negative information (viz., data which by its absence is indicative of mechanical status and health). A general architecture is introduced for CBM automated reasoning, which hierarchically combines implicit and explicit reasoning techniques. Initial experiments with fuzzy logic are also described.

Introduction
The emerging technology of condition-based maintenance (CBM) of mechanical equipment is aimed at developing systems which are capable of monitoring the operation of a complex piece of machinery and providing an accurate characterization of the current system state, and ideally, an accurate prediction of the remaining safe or useful life span. This approach to maintenance represents a departure from traditional maintenance approaches. Historically, maintenance philosophies involved one of two approaches: (1) run until mechanical failure occurs, then repair, or (2) schedule maintenance and inspection based on conservative probabilistic failure models (i.e., using mortality statistics). Either of these approaches can be disruptive to the operation of a factory, commercial operation, or a military mission which depends upon critical equipment. Moreover, the first approach (run until failure) may threaten the safety of humans by waiting until a failure actually occurs, while the second approach (scheduled maintenance) may be expensive, and may actually increase failure rates by the implementation of unnecessary maintenance. This latter effect is analogous to the problem of iatropic illness in medicine, in which treatment for one condition inadvertently induces another illness or problem. Thus, even if properly performed, preventive maintenance may induce subsequent failures in equipment because of the disruption of structural integrity, or other factors.

The advent of sophisticated sensors, rapid evolution of microprocessors, and new algorithms permits a third philosophy, condition-based maintenance, to be implemented for real systems. Condition-based maintenance (CBM) offers a significant cost savings and system availability increases for some applications, e.g., Rao (1986) and Pandian and Rao (1989). This philosophy and approach, described by Hansen et al. (1995), involves monitoring a system with robust sensors, and processing the sensor data to ultimately achieve intelligent inferences regarding the current mechanical state of a system, recommendations regarding the need for maintenance actions.
and predictions of remaining useful life (assuming current operating loads and conditions or a profile of expected system utilization). The general concept of a CBM monitoring system is shown conceptually in Fig. 1.

Functional elements of a CBM monitoring system include: (1) a suite of active or passive sensors to observe phenomena such as temperature, pressure, existence of oil debris, vibrations, acoustic spectra, etc.; (2) signal processing techniques to characterize the signal data and to extract significant features from the data; (3) multi-sensor data fusion techniques to combine the sensor data to obtain a more accurate characterization of the system being monitored than is available from a single sensor; (4) predictive models at the micro and macro levels to predict the onset and temporal evolution of failure phenomena; and finally (5) automated reasoning techniques to interpret the results of the sensor data, signal processing, fused multi-sensor data, and predictive models. Ancillary functions include supporting models and data bases and a human computer interface. Examples of the techniques associated with these functional components are shown in Fig. 1. Depending upon the specific CBM application, these functions may be implemented on a single smart sensor processing chip to monitor a single mechanical component, or in a distributed environment which uses multiple sensors and a centralized computing resource for platform level CBM, e.g., for a rotor craft, multi-engine commercial aircraft, etc.

Currently, implementation of CBM systems is limited to monitoring systems (or mechanical components or subsystems), with a primary ability to accurately characterize the current state of a system, but with limited ability to predict remaining life (not withstanding the claims of monitoring system manufacturers whose predictive results are based on simple trend techniques). Recent CBM systems have been reported for a variety of mechanical systems such as the space shuttle main engine [Guo and Merrill (1991) and Lin and Wu, (1991)], diesel engines [Hansen et al. (1994)], helicopter gearboxes and drivetrains [Jamre, Danai and Lewicki (1995) and Dousis (1994)], and gas turbine engines [Loukis, Mathioudakis, and Papailiou (1994) and Merrington (1994)]. Techniques applied to CBM span a broad range including (but not limited to); rule-based expert systems [Hansen et al. (1994)], adaptive neural networks [Ganesan, Jiouhua, and Sankar (1995)], classic pattern recognition techniques [Upadhyaya (1988)], non-linear difference equations [Neon and Li (1995)], fuzzy logic reasoning systems [Isermann, and Ulieru (1993)], and stochastic modeling techniques [Ray and Wu (1994)]. There is much on-going research in this area, and new techniques are appearing, including application of new signal processing techniques [Young and Gordon (1996)], new macro-mechanical prediction models [Harris (1996)], non-linear dynamical models [Cusumano and Harris (1995)], and application of multi-sensor data fusion techniques [Hall, (1994)].

At this time there are many remaining issues which prevent the implementation of effective intelligent CBM monitoring systems. Figure 2 provides a brief summary of these issues and limitations for each of the CBM functional components shown in Fig. 1. The primary focus of this paper is on the challenges for automated reasoning for CBM.

**Concept of Approximate Reasoning for Condition-Based Maintenance**

The concept of automated approximate reasoning systems has an extensive history. The development of so-called expert systems has been a leading goal of artificial intelligence (AI) systems. These systems attempt to emulate the type of reasoning about real-world problems normally performed by human specialists or experts. At this time several thousand expert (or knowledge based) systems have been reported in the literature, and there are numerous books [Charniak et al. (1980) and Hopgood (1992)] and commercially available tools to assist in their development. A guide to the extensive AI literature is provided by Gelfand (1992). Approximate reasoning systems have been implemented for a wide variety of applications including medical diagnosis, chemical analysis, computer architecture design, planning systems, tactical threat assessment and situation assessment for military applications, and many others. It is natural then that these types of systems should be built to address condition-based maintenance.

The structure of expert systems explicitly separates a knowledge base (which represents facts, algorithms, heuristic logical relationships, etc.), about the domain of application (i.e., about the condition-based maintenance system), and the control structure or inference engine. The knowledge about the domain of interest may be represented via rules, frames, scripts, networks or direct (analogue) structures. The most popular form of representation techniques is the use of production rules of the form: If (there is evidence of condition [X]) then do (action [Y]).

An example of a rule associated with CBM is the following: If there is evidence of increasing debris in the lubricant, then update the dynamic data base to indicate that there may be excessive mechanical wear in the monitored mechanical subsystem. Rules can be used to represent a collection of facts or procedures associated with a domain of interest. Regardless of the form of the representation technique (viz., rules, frames, etc.), uncertainty can be incorporated both in the antecedent (or left hand side of the rule) and also in the consequence (or right hand side of the rule). In the above CBM example, uncertainty could be allocated to the evidence of debris (uncertainty about the existence of debris observed by a chip detector), as well as uncertainty in the conclusion (i.e., even if there are observed chips in the lubricant, this may not be conclusive evidence for excessive wear). The specification of a knowledge representation scheme also entails the specification of the representation technique for uncertainty, and the associated calculus of uncertainty (i.e., the rules by which uncertainty measures are combined and propagated in an inference process).

Popular uncertainty representation schemes include probability, Dempster-Shafer evidential intervals, fuzzy membership functions, certainty factors and/or ad hoc data ambiguity indicators relating to either the collection integrity or (semantic) meaning of received data/features within a given context. An excellent review of these representation techniques is provided by Post and Sage (1990) and by Krause and Clark (1993). All of these techniques, however, are challenged by the problem of combining both positive information (e.g., symptoms of a problem) with negative information (i.e., information which by its absence or lack of symptoms is indicative of meaningful inferences).
Figure 1. Functional Components for a Condition-Based Maintenance System.

<table>
<thead>
<tr>
<th>CBM System Components</th>
<th>Description</th>
<th>Challenges</th>
</tr>
</thead>
</table>
| Sensors                     | Active and passive sensors to monitor mechanical systems and detect precursors to failure (e.g., accelerometers, temperature, pressure, etc.) | - Lack of robust sensors.  
- Non-optimal placement of sensors for existing mechanical systems  
- High false alarms rates  
- Sensor suite selection |
| Signal Processing           | Digital Signal Processing (DSP) techniques for real-time analysis and characterization of the signal data. | - Detection and characterization of rare events  
- Poor signal to noise ratios  
- Real time processing |
| Multi-sensor Data Fusion    | Combination of multi-sensor data to improve the estimate of equipment state and detection of failure precursors. | - Lack of fusion models for CBM  
- Need to fuse non-commensurate sensor data |
| Automated Decision Processing | Automated approximate reasoning to interpret the results of sensor measurements, DSP, and predictive models for context based interpretation of results. | - Need to combine implicit/explicit reasoning  
- Hierarchical reasoning  
- Need to address negative information |
| Predictive Techniques       | Micro and macro level models to predict the temporal evolution of mechanical failure phenomena. | - Fundamental lack of predictive models  
- Need to link statistical prediction with non-linear dynamic system evolution |

Figure 2. Challenges for Automated CBM Systems.
The Challenge of Negative of Reasoning

An example of negative reasoning is readily provided by the medical community. Physicians must be able to interpret and evaluate the relevancy and accuracy of information a patient gives them concerning their apparent illness in order to either make a diagnosis, or to specify additional testing if the patient supplied data is incomplete. In many cases, both serious and not serious, patient related symptoms are not sufficient for diagnosis and additional tests need to be performed.

From a physician’s point of view, the doctor cannot allow himself to miss a potentially serious disease in his patient. Unless the diagnosis is trivial or readily treatable by a general antibiotic, the doctor is likely to specify a series of tests that seem relevant to the case at hand and would quickly bring to the surface the possible presence of a life threatening condition such as a carcinoma, aneurism, or arterial blockage, etc. By a careful selection of blood and tissue tests and scans, the doctor will be able to either positively diagnose the condition or, at least, selectively eliminate candidates on the list (of hypotheses) of potential disorders. In particular, a negative test result indicates to the physician what is not the problem. This, in principle, is negative reasoning and appears to be an integral part of the medical community’s standard approach to the treatment and diagnosis of disease. So, even though the doctor doesn’t yet know exactly what is wrong with us, it may be assuring to know that whatever it is, it’s probably not life threatening.

Consider the following example of the use of negative reasoning related to condition-based maintenance. Suppose that one observes excessive wear occurring on one’s relatively new automobile tires. Possible mechanical rules-based faults leading to rapid tire wear are; (a) Improper tire pressure, (b) front end misalignment, (c) damaged suspension arms, (d) worn shock absorbers, or (e) wheels out of balance. If your mechanic (the mythical Mr. Holmes), discovers nothing wrong (i.e., a negative result), he may then question you on your own driving habits. These questions could eventually lead to the information that there are several eligible drivers with access to this vehicle. His suspicions (hypothesis) of premature tire wear due to hard driving are reinforced by a quick look at the front brake pads and discs, and their installation date/milage records. He finds both excessively worn pads, and heat glaze damage on the surface of the discs (positive result) for the age of the parts. Thus, by using a combination of positive and negative reasoning in conjunction with the vehicle maintenance history data base records, machinery malfunction is ruled out.

These examples suggest that negative reasoning seems to be a natural part of the human reasoning process. Although, J. B. Evans et al. (1993) point out that there are numerous cognitive problems exhibited by humans which can cause erroneous reasoning. It appears however, that negative reasoning can be invoked by humans when a situation presents itself that does not necessarily fit a given template and/or admits multiple hypotheses. The paradigm then shifts at some point from identifying what is wrong, to enumerating problems that cannot exist based on the observed data. In this way, one can at least possibly eliminate machine life critical problems that might lead to expensive operations shutdown or mission abort. In addition, within the negative reasoning context, fuzzy set theory can be used to measure degrees of set membership, e.g., machine oil may only be partially contaminated by the small addition of the wrong oil or the presence of debris vs a more concentrated degree of contamination, etc. Thus, even the degrees of negativeness instead of 0/1 binary states can be incorporated into the overall analysis scheme of a diagnosis.

A Health Usage Monitoring System (HUMS) Example

In the case of an on-board helicopter Health Usage Monitoring System (HUMS), suppose that the low-oil transmission system pressure-sensor trips several hours into a flight mission. Any problems in the transmission, of course, get the pilot’s attention quickly, and may necessitate a mission abort. Possible fault conditions leading to low transmission oil pressure are; (a) loss of fluid/seat leakage, (b) broken shaft in pump, (c) other oil pump failure, (d) clogged screens, or (e) faulty transducer/indicator system. The HUMS controller now has to decide what to do, i.e., what does it relate to the flight crew. Negative reasoning could be used to effectively identify and eliminate possible causal mechanisms.

The HUMS system should be able to cross-correlate the low oil condition with; (1) transmission oil temperature - normal temperature may suggest a false alarm/FAULT sensor/electrical fault; (2) tripped oil bypass valves - if not tripped then the screens are probably fine; (3) engine oil level - if not overfull, then the dividing labyrinth seal between transmission and engine is probably not leaking; (4) check last time oil was added to the transmission; (5) check vibration signatures/levels of accelerometers in the area; and (6) check conditions of the transmission chip detectors and, when technology permits, perform an on-board oil spectroscopy analysis for ppm metal counts and rates. If no supporting evidence is found for the low oil pressure alarm, then the HUMS should: (a) record the event and relevant sensor information; (b) alert the pilot that a probable oil pressure sensor false alarm has occurred in the transmission, and that he should, if possible, perform an acceleration or change of speed maneuver carefully so HUMS can double check; (c) inform the pilot that the system has increased vigilance on the transmission, but to continue mission normally until further notice.

Figure 3 shows a positive matrix mapping of observations (O1, O2, ...) across the top to causes (C1, C2, ...), down the left side. Similarly, Fig. 4 shows the negative matrix mapping of non-observations in X across the top to causes (C1, C2, ...), down the left side. Thus, for example, if the set of positive observations from figure 3 are {01} => problems (C1 or C3) and the set of negative observations from figure 4 are {—03, —04, —05}, then fusion of the two positive and negative observation sets leaves C3 as the diagnosis. On the other hand, if the positive observation set is {01 and 02} and the negative observation set is {—03 and —04 and —05}, then there is/are either a faulty sensor(s), or the knowledge base has a logical inconsistency.

Logically speaking, negative reasoning is closely akin to what is sometimes called the contra-positive, i.e., P => Q <= > —P, where P means not P. The statement (—Q) => —P is the contra-positive of P = > Q. The proof of equivalence is simple. P = > Q has the same truth table as (—P) or Q. Likewise, (—Q) => —P has the same truth table as (—Q) or —P = (Q or —P) = (—P or Q) which is that of the original P => Q truth table. Consequently, negative reasoning approaches are naturally built in to first order logic (FOL) system theorem provers in which P is shown to follow from a set of propositions S by first assuming —P and then deriving a contradiction (resolution refutation). Thus, negative reasoning could be utilized with
first order and fuzzy logics or other inference schemes to help resolve cases of both uncertain and incomplete knowledge.

In practice, the combined positive/negative reasoning illustrated above by Figs. 3 and 4, can be implemented using a technique called *logical templating*, originally developed for automated reasoning about tactical military situation and threat assessment. This technique combines logical relationships with the ability to identify information which must be present (or absent) in order to establish the existence of an activity, event, or condition. Logical templating may be implemented in a number of ways [e.g., Hall and Linn (1989) and Noble (1987)]. More recently, these methods have begun to be implemented using knowledge-based system tools providing frame-based or case-based reasoning capabilities. Logical templating can be implemented very efficiently. Hall and Linn (1989) provide a description of the knowledge representation techniques and inference mechanisms.

### A General Approximate Reasoning Architecture for CBM

Of course, the caveat of negative reasoning in diagnosis is that one needs a finite list of symptoms or hypotheses to work on, i.e., the doctor cannot specify an unlimited number of tests for his patients, but selects from a set of both general and system specific tests based on the reported symptoms. Unfortunately, combinatorial explosion of hypotheses is a general problem for logic systems since the more knowledge you have, the more hypotheses you are able to generate.

Due to the intricacies of real-world problems, a combination or hybridization of reasoning techniques is required to achieve some basic acceptable level of performance. These systems may be structured heuristically as a hierarchical blackboard/whiteboard architecture with intelligent numerical, parametric, and/or symbolic reasoning subsystem experts. However, in a context where either missing or uncertain knowledge is the rule rather than the exception, so-called monotonic reasoning methods appear inadequate. Due to the addition or change of input information over time, conclusions previously arrived at along the way within a diagnosis procedure may need to be revised or extended in order to accommodate the new information. For this reason, nonmonotonic methodologies are often considered, i.e., those methodologies for which a conclusion can be retracted.

For example, first order logic is monotonic since a new axiom may not invalidate any of the previous conclusions. Monotonicity, however, is generally viewed as a deficiency both in representation and reasoning for actual system applications (because it causes the inference process to become computationally intractable). On the other hand, Dempster-Shafer's method of evidential reasoning [Dempster (1968) and Buede and Martin (1987)] does permit nonmonotonicity. Post and Sage (1990) provide an extensive discussion of knowledge based methods and the issue of monotonic vs nonmonotonic reasoning.
In a basic sense, automated decision making may be reduced to a (directed) search. The search may begin at the meta-level whereby the intelligent controller plans its attack against the problem, and then proceed afterwards in a top-down/bottom-up real-time control strategy. The search is generating and examining various hypotheses concerning the meaning of the data from its storehouse of knowledge. Unfortunately, search consumes time which for many real-time processors is a constraining factor which must be carefully monitored. Any method that can effectively and validly reduce search time is then by definition good.

Consequently, negative reasoning is one tool that may be of help in reducing the overall number of hypotheses at various levels of abstraction for a given intelligent controller (I.C.) application. We believe that no one single knowledge representation scheme and inference technique is suitable for condition-based maintenance to span the inference hierarchy from the materials through platform level. Figure 5 shows a general knowledge-based system (KBS) architecture for CBM. The architecture uses a hierarchical set of reasoning approaches with an overall blackboard type of structure [see Englemore and Morgan (1988) for a discussion of blackboard systems]. In particular, at the platform level, three separate blackboards represent the current equipment status, predictions of near-term conditions, and a long-range (far-term) prediction. Various different knowledge representation techniques are used at the materials, element, component, subsystem, system, and platform levels to represent knowledge about mechanical faults, conditions, and predictions of the future evolution of failure mechanisms. A summary of these techniques at each level of the hierarchy is shown in Fig. 6. A particular challenge is to develop effective predictive models to predict the progression of a system state in time (e.g., the prediction of the progression of an identified failure mechanism).

Combined rules-based and negative reasoning strategies can be encoded and implemented in object-oriented languages such as C++ for a variety of applications. Such languages with their symbolic object structures and associated procedures seem, in particular, to enjoy success in the areas of signal and image processing. However, some specialized languages such as Prolog may experience some difficulty in exploiting negative reasoning to full advantage if they operate entirely in closed-world mode, i.e., if a relation cannot be proved with the existing knowledge, it is considered false.

We are currently experimenting with different techniques for condition-based maintenance. One component of this automated reasoning involves the use of fuzzy logic at the platform level. In particular, we have adapted a fuzzy logic controller system, originally developed for real-time control of the sensors, propulsion, and guidance of an autonomous underwater vehicle [see Gibson et al. (1994)]. This architecture uses a new formulation of fuzzy logic calculus, and a continuous, differentiable fuzzy membership function to map observables into fuzzy membership values for use in the explicit fuzzy reasoning. Such a control system allows integration of the evolving information on mechanical components, subsystems, systems, and at the platform level for a contextual interpretation and recommendations to platform users or crew.

**Comments on Implementation**

The architecture in the previous section is intended to be a general structure which is applicable to condition-based maintenance systems across the complete spectrum from mechanical components and subsystems to systems and platform level of automated inference. The selection of particular knowledge representation techniques and prediction algorithms (e.g., from those listed in Fig. 6), must be tailored to specific applications having specific requirements, sensor suites, mission requirements, etc. Until many systems have been implemented using this structure, it is difficult to evaluate the utility of this architecture compared. However, we are currently implementing two condition-based maintenance (CBM) systems using this approach: (1) an intelligent oil monitoring system as part of Allied Signal Engine's Full Authority Digital Electronic Controller (FADEC) system for an upgrade for a helicopter; and (2) an experimental system to monitor the health of the rotor drive train for a helicopter. Some general comments are provided in this section concerning our initial experiences with this architecture.

The intelligent oil monitoring system for Allied Signal Engine's FADEC system is intended to provide automated monitoring and prediction of the oil subsystem for turbine engines. This is part of an anticipated evolution of the TA 40 FADEC. Types of sensor measurements will include: engine speed, oil pump flow, oil pump pressure, gearbox oil pressure, main oil pressure, filter outlet temperature, filter inlet pressure, etc. For each of these sensors, separate signal processing will be performed with extraction of representative features to represent the current state of the system, based on single sensor data alone. Experiments are currently ongoing to establish the specific digital signal processing (DSP) to be applied to each sensor data stream. Several promising techniques appear to be the use of higher order (cyclo-stationary) spectral techniques, and use of non-stationary recursive filters [Lopatin-Skaja et al. (1995)] and wavelet techniques [Weiss (1994)]. A parametric template technique monitors whether observed features (e.g., measured temperature, oil debris metrics, etc.) exceed acceptable thresholds. In turn, these individual sensor data are fused using feature-based, and decision level, data fusion techniques. These latter include classic pattern recognition techniques, weighted decision techniques (i.e., voting techniques), and Bayesian inference methods. The data fusion processing is implemented using a special visual programming data fusion toolkit developed at ARL [Hall and Kasuma (1996)]. This toolkit was implemented in Visual C++ and allows researchers to select signal processing and data fusion algorithms from an evolving library of algorithms. The techniques can be linked together and applied to either simulated or real sensor data. Graphics utilities allow ready visualization of the processing results. The tool is analogous to the toolkit Khoros used for image processing.

A fuzzy logic inference system, based on Stover and Gibson's continuous inference network [Gibson, Hall and Stover (1994)] provides an interpretation of the results. While the fuzzy logic representation is convenient to represent explicit information about interpretation of sensor observations (e.g., based on the operation of
Figure 5. A Hierarchical Approximate Reasoning Architecture for CBM.
<table>
<thead>
<tr>
<th>Hierarchical Layer</th>
<th>Example</th>
<th>Knowledge Representation and Diagnostic Models</th>
<th>Predictive Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>Metal</td>
<td>- Finite Element Models</td>
<td>Coupled Field Damage Models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Continuous Models</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>Bearing Ball</td>
<td>- Finite Element Models</td>
<td>Ioannides-Harris Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Oil Analysis</td>
<td>Crack Growth Models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Mechanical Dynamic Models</td>
<td>Probabalistic Neural Nets (PNN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Wignerville/Stewart Figure of Merit [Stewart (1977)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time Synch, Averaging</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PNN</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Bearing</td>
<td>- Probability Models [Pomfret (1995)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Logical Templates (Case-Based)</td>
<td>Mortality Statistics [Harris]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bayesian Belief Nets [Shachter (1986)]</td>
<td>Non-Linear Dynamics [Cusumano and Harris (1995)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dempster-Shafer's Method [Dempster (1968)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Continuous Inference Nets [Gibson, Hall and Stover (1994)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PNN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Petri Nets [David and Hassane (1994)]</td>
<td></td>
</tr>
<tr>
<td>Subsystem</td>
<td>Hub/Swashplate/Lubrication</td>
<td>- Continuous Inference Nets [Gibson, Hall and Stover (1994)]</td>
<td>UMARC [Haas et al. (1995)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bayesian Belief Nets [Shachter (1986)]</td>
<td>Markov Modeling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Undirected Graph Models [Pelc (1991)]</td>
<td>Floquet Theory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dempster-Shafer's Method [Lowrance and Garvey (1982)]</td>
<td>Dynamic State Models [Frank (1990)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Logical Templates</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Petri Nets [David and Hassane (1994)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Blackboard KBS [Hall and Linn (1989)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expert Systems [Hansen, Autar and Pickles (1993)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interacting Autonoma [Peluso et al. (1994) and Phoha et al. (1993)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Syntatic Models [Eshera and Fu (1986)]</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>Main Rotor System</td>
<td>- Continuous Inference Nets [Gibson, Hall and Stover (1994)]</td>
<td>Scenario Models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bayesian Belief Nets [Shachter (1986)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Undirected Graph Models [Pelc (1991)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dempster-Shafer's Method [Lowrance and Garvey (1982)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Logical Templates</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Petri Nets [David and Hassane (1994)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Blackboard KBS [Hall and Linn (1989)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expert Systems [Hansen, Autar and Pickles (1993)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interacting Autonoma [Peluso et al. (1994) and Phoha et al. (1993)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Syntatic Models [Eshera and Fu (1986)]</td>
<td></td>
</tr>
<tr>
<td>Plant/Platform</td>
<td>Helicopter</td>
<td>- Continuous Inference Nets [Gibson, Hall and Stover (1994)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expert Systems [Hansen, Autar and Pickles (1993)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Utility Theory</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Blackboard Models [Englemanre and Morgan (1988)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Syntatic Models [Eshera and Fu (1986)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Process Models [Dvorak (1991)]</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. Examples of Knowledge Representation Techniques for CBM.

the turbine engine), there are still challenges with combining uncertainty information about sensor reliability (in a dynamic sense), a priori information about the relative likelihood of fault conditions, and the confidence of logical relationships. This problem has not yet been solved. However, we expect that appropriate combination of probability measures, fuzzy membership functions, and utility measures will be useful to address the uncertainty representation. Currently, experiments are being conducted using simulated sensor
data and hypothesized profiles of system operation (viz. the operation characteristics of the system as a function of a startup, typical operation, and system shut down). In the near future, actual sensor data will be processed. The resulting signal processing, data fusion, and automated inference system will be implemented in the FADEC processing architecture. An advantage of the continuous inference network formulation for fuzzy logic is that it admits of a real-time implementation using common microprocessors. The negative information templating process, described in the previous section of this paper, is used to reduce the number of potential causes for maintenance problems.

The second system currently under development is an automated inference system to monitor and diagnose problems with the rotor drive train for a Huey type helicopter. The concept involves use of accelerometers and related sensors to monitor the condition of the rotor system including the rotor assembly, mechanical drive linkages, and transmissions [see Hall and Smith (1996)]. The focus on this research involves the higher level inferences for prediction and data interpretation (rather than the details of the processing of individual sensor data). Current developments use a rule-based fuzzy logic reasoning structure. Experiments are ongoing to compare standard fuzzy reasoning systems (e.g., such as those available from the commercial package, MATLAB), to the Stover and Gibson Continuous Inference Network (CIN) structure [Gibson et al. (1994)]. It appears straightforward to develop fuzzy logic rules for diagnosis and prediction based on observed conditions. For example, only 30 rules are required to analyze vibration, temperature and pressure data to diagnose bearing seizure and pump failures in a helicopter drive train. The negative reasoning template structure described in the previous section is again used to reduce the number of potential causes based on observed symptoms.

A general comment involves the adoption of predictive models. Currently, truly predictive models do not exist. Other research at Penn State is focusing on this area [Cusumano and Harris (1995)]. In the mean time, we are utilizing trending models based on several types of neural networks to process sensor time series data. In addition, the evolution of feature-based data are tracked to provide predictions of failure trends. As more physically realistic models become available we will incorporate them into our automated CBM model.

**Summary**

In this paper we have discussed the challenges of approximate reasoning for condition-based maintenance systems. At this time there does not appear to be a single knowledge representation and reasoning approach which will perform automated context-based reasoning for CBM, meeting the unique requirements which include the ability to reason with negative information, the need to reason about a hierarchy of components, and the need for nonmonotonicity. The architecture described here utilizes a wide set of techniques to link microscopic failure phenomena, macroscale observations, and human decision factors such as mission constraints. The structure introduced here is very general, but can be readily adapted to individual CBM problems. In particular, the use of a blackboard structure and a hierarchical set of inference methods is well suited to mechanical systems because these systems can be partitioned based on physical interconnections and hierarchy. Experience with two sample CBM systems appears promising. Much research still remains, especially in the area of predictive models. Nevertheless, progress is being achieved to attain true predictive intelligent CBM systems.

**References**


