The Benefits of a Model-Based Approach to Diagnostics and Health Management

Many companies are using databases full of historical diagnostics and maintenance information (cases) to support field service operations. However, today's modern diagnostic and health management technology has been refined and is able to use failure information about the system (a model) to automatically generate intelligent, dynamic maintenance procedures. These commercial solutions use current symptoms from the machine or observed by the operator/technician and optimization algorithms to generate a procedure "on-the-fly" that adapts to the exact situation faced by the field service engineer (FSE). The dynamic procedure is able to consider cost, time, current symptoms(s), resource constraints and use information gain algorithms (like looking ahead a few moves in a chess game) to generate a more optimal procedure. The results have shown over 75% reduction in fault isolation time, 30% or better increase in machine availability, and most importantly, the ability to elevate every FSE to an expert level of troubleshooting and maintenance capability with minimal training and experience. Qualtech Systems, Inc. (QSI) (www.teamqsi.com) is a leading provider of model-based, intelligent diagnostics, prognostics, and health management (DPHM) software. The TEAMS (Testability Engineering and Maintenance System) software tool set provided by QSI is a "best in class" (award winning) model-based reasoning (MBR) solution available for immediate implementation in large field service operations. To help illustrate the power and significant benefits from this state-of-the-art solution, a QSI customer, Orbotech (Yavne, Israel) demonstrated and measured TEAMS capabilities. Orbotech modeled one of their machines in TEAMS, deployed the solution, introduced actual faults in the machine, and asked highly experienced technicians and average technicians to troubleshoot the problem – with and without TEAMS. The results are dramatic:

During TEAMATE trials on actual machine faults, Orbotech (Yavne, Israel), demonstrated that their junior field service engineers could resolve the faults in about the same time required by an experienced expert - only 15 minutes! The trials proved that their FSEs, operating around the world, could benefit from the troubleshooting knowledge contained in the TEAMS software.

The positive effects on training costs, downtime, maintenance costs, service call rates, reduced false pulls, and customer satisfaction are enormous. We do not believe that database solutions (or even more sophisticated case based solutions) alone can achieve this level of performance.

The remainder of this white paper discusses how your current database system can be leveraged, enhanced, and expanded to implement TEAMS. Based on our considerable experience in this arena, we have collected and summarized a few of the more significant differences between our model-based DPHM solution, TEAMS, and those that use database (or case based) methods.

- **Symptom or machine error code with only one possible faulty part**: In that scenario the stored case/record, once recorded, will help the technician resolve the problem every time. The only issue here is the failure of the database system to make sure the case is resolved in the most efficient way in terms of number of tests performed, the time spent performing a test, cost of the test, etc. Unlike data based tools, the MBR tools, when using their dynamic reasoning algorithms, will always guide the technician to the most efficient and effective way possible. In other words, if the recorded event was resolved in a less than optimal manner, then all you are doing is repeating the same inefficient process again and again.

- **Symptom or machine error code with multi-suspected parts**: Database systems work in the following way: the cases are recorded as they are received from the field, so a symptom that has multi-suspected parts will have several records, one for every faulty part. Then, when a technician is searching the database for that symptom he/she will see several records with different resolutions. Unless there is a pre-questionnaire that can help the technician select the symptom that can lead him/her to the right resolution, the technician will have to try each part individually until the problem is resolved. Not only is this process time consuming, but there is also a chance that good parts may be unnecessarily replaced. However, if the symptom with multi-suspected parts is treated as such from the beginning, then resolution procedure can be organized so that the technician will be guided, with a set of leading questions/procedures, to the right solution. But as mentioned in the beginning of this paragraph, the database systems usually record the cases as they are gathered from the field. Therefore, the best asset of the database system, namely not having to put any upfront effort into building the database as MBR may require, is lost. Dealing with symptoms with multiple faults is one of the best features of MBR tools. Its dynamic reasoning will guide the technician to the right part in the most efficient test-sequencing every time, all the time.

- **Multiple symptoms or machine error codes**: Database systems are not designed to deal effectively with such a scenario. As mentioned before, cases are stored as they occur in the field. Usually one record is processed for every symptom and every faulty part. One may argue that you can create a case that is a combination of several symptoms. This is true, but not practical. For example, you can have 15 combinations of cases for 4 symptoms, or 31 cases for 5 symptoms, etc. Again, one may argue that for 4 symptoms not all the 15 combinations are relevant. That is true, but still not practical. Even if half of the combinations are likely to happen it is still a lot of work to sort them out and it is a nightmare to build the fault tree that represents them. We all know that machines very often report more than one error code and additional symptoms can be observed at any time. With MBR systems, the technician just needs to click on the symptoms or on all the error codes he/see sees and the dynamic reasoning takes it from there.

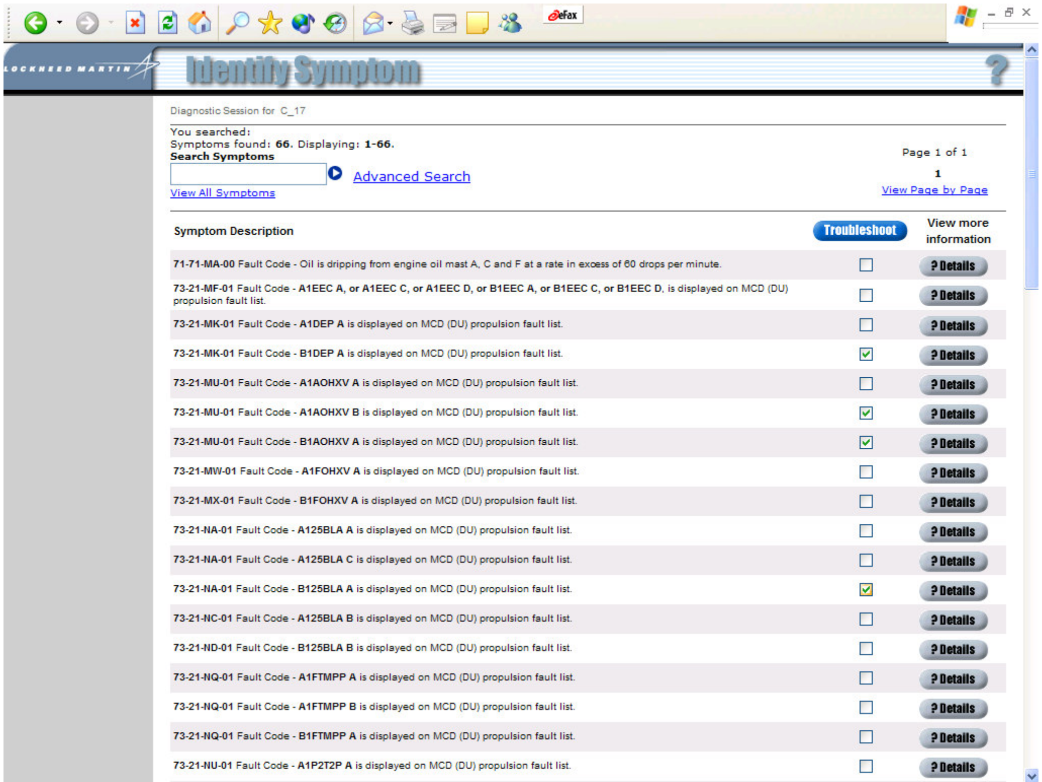


Figure 1: Tech selects the symptoms from the list (has advance search)

- **Scenario where certain tools or resources are needed to resolve the problem but not all of them are available**: This is a very real scenario. In many cases the technician may need a DVM, an Oscilloscope, Jumpers, Gages, etc. to troubleshoot the problem. Data based systems were not designed to deal efficiently with such scenarios. The stored resolutions are based on the tools that were available to the technician at that time. However, what would a technician do if one or more of the necessary resources (tools, testers, etc.) are not available? The troubleshooting would have to resort to an ad hoc process. Again, one may argue that it is possible to store cases for every possible resource combination. This may be true, but again not practical. (By the way, we have yet to encounter anyone who is willing to make these arguments on any logical/technical basis). If you need 4 tools you will have to store 15 combinations, etc… etc….. With TEAMS, the issue is resolved very nicely. After the technician selects the list of symptoms and/or error codes he/she will be prompted with a form that lists the tools and resources that may be necessary during the course of the troubleshooting session. The technician can indicate that specific resources are not available. Subsequently, the TEAMS algorithms will automatically generate a most optimal maintenance procedure that avoids any steps or tests that require those "unavailable" resources. Figure 2 shows this form.

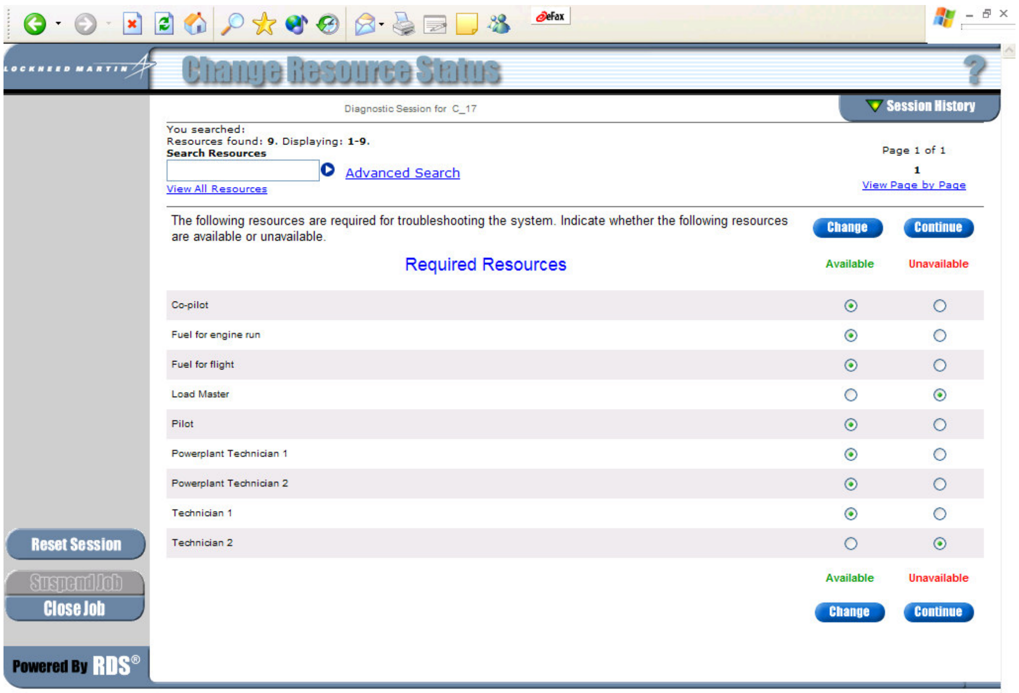


Figure 2: Tech selects what resources or tools available

- **Flexibility with test executions:** The TEAMS MBR approach to dynamic, intelligent troubleshooting provides maximum flexibility for the FSE. In contrast, database methods simply display the steps that were taken in a recorded (past) case requiring the FSE to either follow the same procedure or start ad hoc troubleshooting/repair. Suppose the technician gets to the machine, selects the symptom/s, looks at the reported suspected list and starts performing the first suggested test to narrow down the list. But, to streamline the procedure, the technician may also want to indicate to the database system that he/she already knows that some of the parts on the suspected list are good so the system will only focus on the other suspected parts and suggest the relevant tests that will isolate the faulty components. Clearly, this option would save the technician a great deal of time. Database systems are not equipped to deal with such requests. The technician must follow the instructions the way they were recorded. No options. With TEAMS, this issue is resolved by allowing the technician the option of selecting from the suspected list those parts that he/she believes are good (or bad, in case of multiple faults), at any time during the troubleshooting process. The MBR reasoner will automatically take the good parts from the suspected list and focus only on what is left.

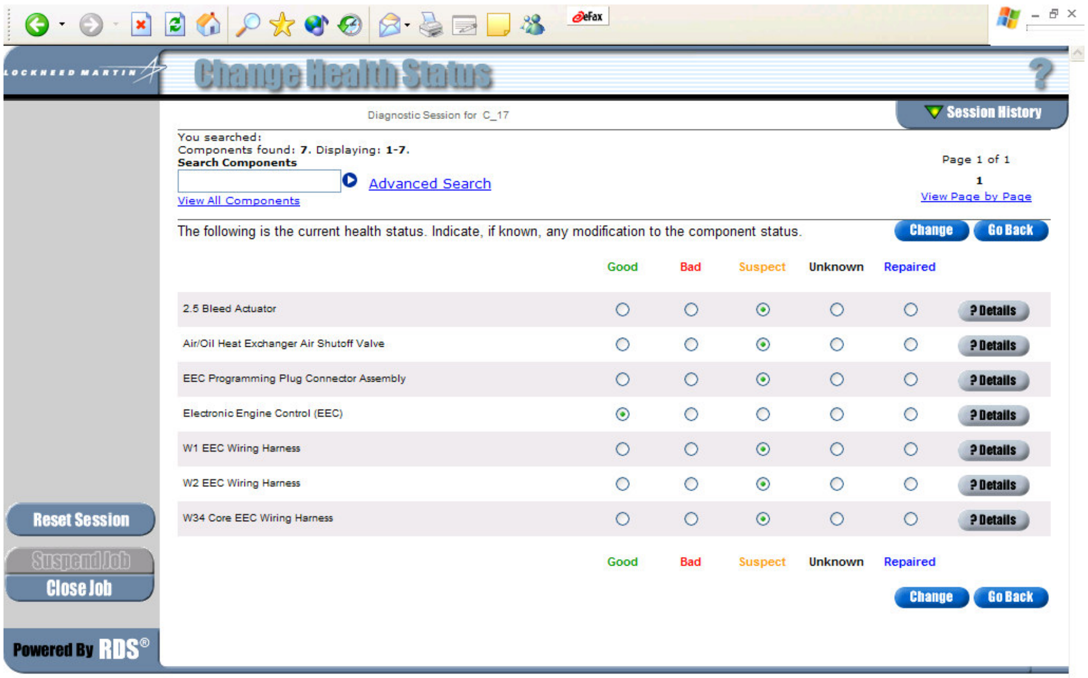
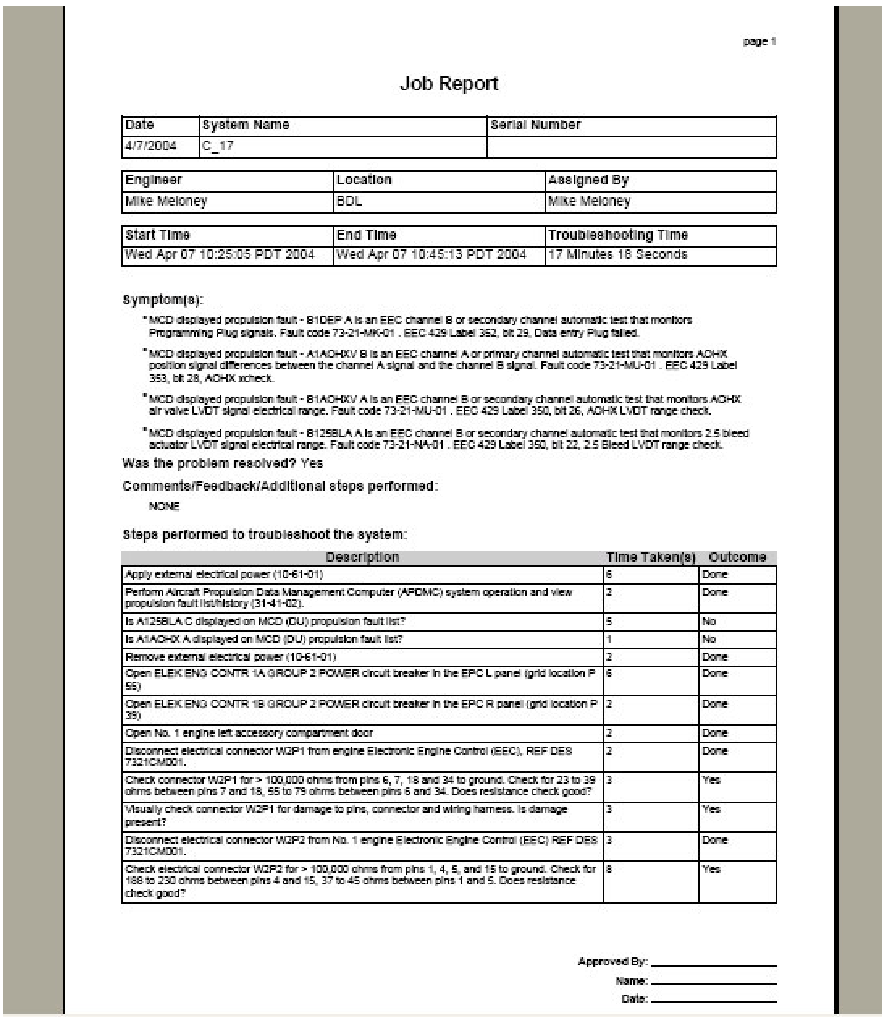


Figure 3: Tech can indicate, up-front, what parts are definitely good or bad

- **Learning:** The TEAMS model contains basic relationships between component faults and tests (all types of tests – built-in, manual, support equipment, etc.). The model also contains parameters characterizing component failure rates; the cost and time to execute a test and test setup; and the cost and time to remove/replace components. The relationships in the model (faults to tests) and these parameters are used to generate diagnostics and optimal troubleshooting/maintenance procedures. During active troubleshooting/maintenance sessions, TEAMS logs symptom(s), diagnostics, steps, outcomes, actions, and time stamps. TEAMS also provides a mechanism for the FSE to document any new information or discovery such as a new failure mode, new fault to test relationship, or new test (if he/she resorted to ad hoc troubleshooting). This information logged in the TEAMS knowledge base can be used to update parameters and/or update the model. Subsequent troubleshooting/maintenance sessions will then utilize the "new" knowledge to more efficiently perform the maintenance. This solution uses a model and associated parameters to "inference" its way through the problem in the most efficient manner possible – regardless if the problem were experienced in the past or not. It is our experience that this "combination" of methods (model-based reasoner "refined" with actual field data/results) is by far the most effective and capable solution available in this industry.

  By contrast, a database method may display the symptom's suspected list in descending order of failure probability. Based on recorded data, which may or may not be statistically significant. The database tools have no inference engine that can re-calculate the tests sequencing that can lead to faster and/or cheaper sessions. There is little or no "adaptation" or flexibility and very poor performance on less frequent faults – the faults that typically give FSEs the most problems since they have much less experience handling this symptoms (or combinations of symptoms!).

- **Reports:** TEAMS keeps a log of every repair session. The log contains some logistic information (tech name, customer information, time stamp, etc), the symptom/s that initiated the service call and all the steps performed by the tech including the part/s replaced to fix the problem. There are all sorts of benefits you can reap from this report. For example, you can:
  a. Send a copy to your customer for billing or review and comments
  b. Generate management and engineering reports, as a bar or pie chart, about most frequent symptoms, replaced parts, time to resolution, etc.
  c. Use the data for refining the model and/or parameters (knowledge base)



page 1

Job Report

Date	System Name	Serial Number
4/7/2004	C_17	

Engineer	Location	Assigned By
Mike Meloney	BDL	Mike Meloney

Start Time	End Time	Troubleshooting Time
Wed Apr 07 10:25:05 PDT 2004	Wed Apr 07 10:45:13 PDT 2004	17 Minutes 18 Seconds

Symptom(s):

- MCD displayed propulsion fault - B1DEP A is an EEC channel B or secondary channel automatic test that monitors Programming Plug signals. Fault code 73-21-MK-01 . EEC 429 Label 352, bit 29, Data entry Plug failed.
- MCD displayed propulsion fault - A1AOHXV B is an EEC channel A or primary channel automatic test that monitors AOHX position signal differences between the channel A signal and the channel B signal. Fault code 73-21-MU-01 . EEC 429 Label 353, bit 28, AOHX xcheck.
- MCD displayed propulsion fault - B1AOHXV A is an EEC channel B or secondary channel automatic test that monitors AOHX air valve LVDT signal electrical range. Fault code 73-21-MU-01 . EEC 429 Label 350, bit 26, AOHX LVDT range check.
- MCD displayed propulsion fault - B125BLA A is an EEC channel B or secondary channel automatic test that monitors 2.5 bleed actuator LVDT signal electrical range. Fault code 73-21-NA-01 . EEC 429 Label 350, bit 22, 2.5 Bleed LVDT range check.

Was the problem resolved? Yes

Comments/Feedback/Additional steps performed:

NONE

Steps performed to troubleshoot the system:

Description	Time Taken(s)	Outcome
Apply external electrical power (10-61-01)	6	Done
Perform Aircraft Propulsion Data Management Computer (APDMC) system operation and view propulsion fault list/history (31-41-02).	2	Done
Is A125BLA C displayed on MCD (DU) propulsion fault list?	5	No
Is A1AOHX A displayed on MCD (DU) propulsion fault list?	1	No
Remove external electrical power (10-61-01)	2	Done
Open ELEK ENG CONTR 1A GROUP 2 POWER circuit breaker in the EPC L panel (grid location P 55)	6	Done
Open ELEK ENG CONTR 1B GROUP 2 POWER circuit breaker in the EPC R panel (grid location P 39)	2	Done
Open No. 1 engine left accessory compartment door	2	Done
Disconnect electrical connector W2P1 from engine Electronic Engine Control (EEC), REF DES 7321CM001.	2	Done
Check connector W2P1 for > 100,000 ohms from pins 6, 7, 18 and 34 to ground. Check for 23 to 39 ohms between pins 7 and 18, 55 to 79 ohms between pins 6 and 34. Does resistance check good?	3	Yes
Visually check connector W2P1 for damage to pins, connector and wiring harness. Is damage present?	3	Yes
Disconnect electrical connector W2P2 from No. 1 engine Electronic Engine Control (EEC) REF DES 7321CM001.	3	Done
Check electrical connector W2P2 for > 100,000 ohms from pins 1, 4, 5, and 15 to ground. Check for 188 to 230 ohms between pins 4 and 15, 37 to 45 ohms between pins 1 and 5. Does resistance check good?	8	Yes

Approved By: ____________

Name: ____________

Date: ____________

Figure 4: Sample of job report (symptoms, tests performed, time, etc.)

- **Equipment Health Management:** TEAMS was designed not only as a troubleshooting tool but also as a comprehensive health management tool. Service managers are using TEAMS to generate reports of equipment status such as: which systems are down and for what reasons, which systems are in service and by whom, and which systems are waiting for service. It also generates reports showing cost/time, diagnostic distributions, technician performance, etc. Many database solutions have no such capability**.**

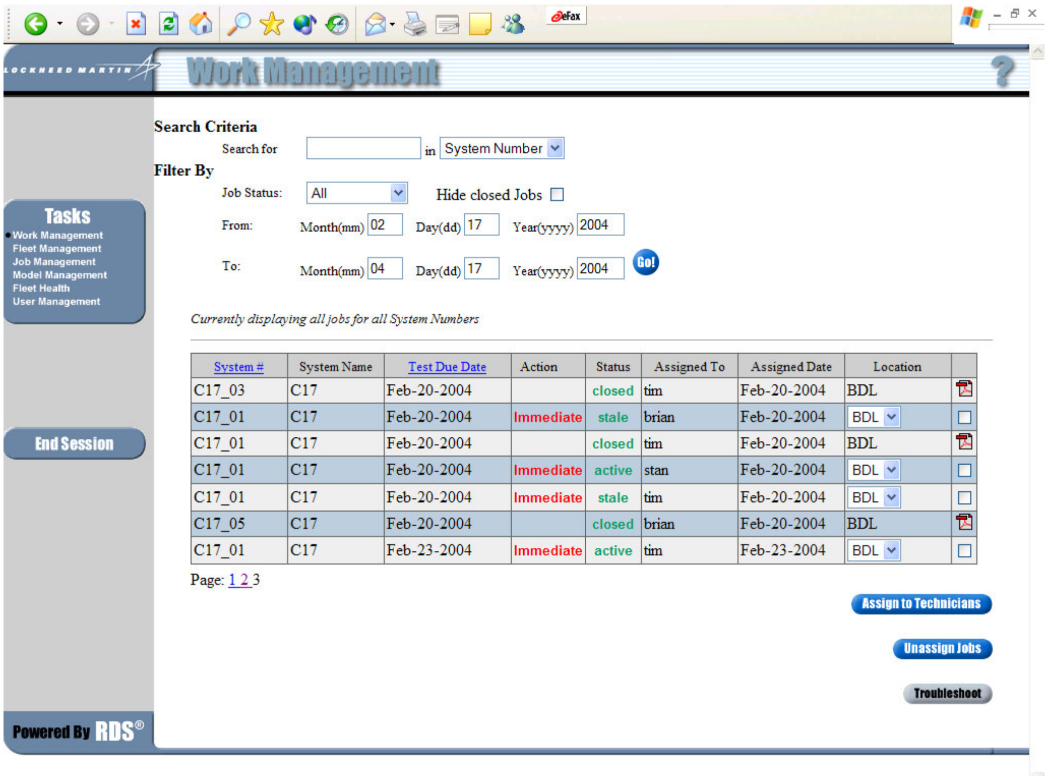


Figure 5: Managers can view the work status

- **Other Model-Based Utilities:** The TEAMS model-based approach presents other capabilities that are not possible with database approaches. The TEAMS model can be used to perform a rigorous analysis of the machine testability and diagnostic characteristics. It is therefore possible to gain an understanding – before deploying the solution – of what the performance will be in terms of fault isolation ambiguity (ambiguity group size/frequency), potential spares impacts, maintenance cost/time, etc. This allows a "baseline" to be established for your field service operations and a good set of metrics with which to evaluate performance/benefit. This is not possible with database methods. In addition, the MBR technology is very suitable to embedding on the actual machine to provide run-time diagnostics – further enhancing your overall field service capability. Again, this is not practical with database approaches.

- **Stand-alone and web-client deployment combinations:** The TEAMS solution can be deployed as a large-scale web-client and/or as a stand-alone solution. The FSE computer can be a lap-top, desk top or compact computer. Database applications may require connectivity to large repositories of field data; the TEAMS solution deploys the "knowledge" (models and parameters) from this data on the FSE's standalone PC.

The TEAMS solution (specifically TEAMS-RDS) also provides a highly scaleable server to web client capability for completely remote "telemaintenance" operations using any web browser device such as a PDA. Several customers deploy a combination of the server and standalone solution to meet all their deployment needs. The solution uses standard database synchronization techniques to ensure that standalones and servers are up to date with models and session logs.

How to leverage existing database and case-based solutions to build new, advanced DPHM capabilities

Now that the case has been made for modern model-based DPHM solutions, the objective should be to enhance/leverage existing database solutions and "build-on" or integrate the TEAMS solution. The end result will be dramatically more capable and efficient solution moving into the future of aftermarket service. We propose the following process:

1. Since TEAMS uses a "component fault" to "test" (including symptoms – which can be tests as well) relationship we believe that it is possible to extract these relationships from the type of databases that are used for service and support.
2. Generate the machine model(s) from the OEM data extracted in step 1 and from other source materials such as existing maintenance manuals, fault trees, failure mode analyses, and other relevant engineering data (if available).
3. Populate and/or link the model with maintenance procedures.
4. Integrate the TEAMS knowledge base (also a database) with the existing database and set up the required tables to extract parameters (used in the models), update existing records in the database, generate reports, support work flow, etc.
5. Deploy the TEAMS solution (specifically TEAMS-RDS and/or TEAMATE) in the appropriate combination of server and standalone – according to the deployment needs and requirements of the OEM.
6. Operate and maintain the advanced DPHM solution.

QSI is currently using processes like this to convert legacy systems, and their maintenance and diagnostics data into a TEAMS based application. These applications include embedded model-based reasoners (specifically TEAMS-RT), intelligent maintenance debrief, intelligent procedures (Interactive Electronic Technical Manuals – IETM), remote web based "telemaintenance", and intelligent, reasoner-driven test equipment.