Rogue components: their effect and control using logical analysis of data

**Mohamad-Ali Mortada · Thomas Carroll III ·
Soumaya Yacout · Aouni Lakis**

Received: 5 September 2009 / Accepted: 19 October 2009 / Published online: 1 November 2009

Abstract There is a small subset of any repairable component population that can develop a failure mode outside the scope of the standard repair and overhaul procedures, which makes them "rogue". When this happens, a Darwinian-like "natural selection" phenomenon ensures that they will be placed in the most disadvantageous position in the asset management program, negatively affecting multiple aspects of the operational and maintenance organizations. Rogue components have long plagued the airline industry and created havoc in their asset management programs. In this paper, we describe how these rogues develop, outline the natural selection process that leads to their hampering the asset management program, and examine some of the negative impacts that ensue. Then we propose a Condition based maintenance approach to control the development of these components. We explore the use of a supervised learning data mining technique called Logical analysis of data (LAD) in CBM for the purpose of detecting rogues within a population of repairable components. We apply the resulting LAD based decision model on an inventory of turbo compressors belonging to an airline fleet. Finally, we evaluate the applicability of LAD to the rogue component detection problem and review its efficiency as a decision model for this type of problem.

M.-A. Mortada (✉) · S. Yacout · A. Lakis
École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada
e-mail: mohamad-ali.mortada@polymtl.ca

S. Yacout
e-mail: soumaya.yacout@polymtl.ca

A. Lakis
e-mail: aouni.lakis@polymtl.ca

T. Carroll III
NetJets Inc., 4111 Bridgeway Ave, Columbus, OH 43219, USA
e-mail: tcarroll@netjets.com

Keywords Asset management · Repairable components · Rogue · Condition based maintenance · Logical analysis of data · Decision model

Introduction

Aircraft maintenance and reliability programs are essential for the safety and airworthiness of airplanes. Maintenance, repair and overhaul (MRO) operations take up a large portion of aviation companies' spending. *Aerostrategy* estimated total air transport MRO costs at $40.8 billion in 2006 (Flint 2007). According to OAG (Official Airline Guide), global MRO spending on military aviation will witness a 14.9% increase over the next decade to reach $67.3 billion a year in 2018 (OAG 2008). The performance of an aircraft operator often hinges on its capability to provide fast and efficient replacement of defective components in its fleet. For that reason, operators may either carry in-house maintaining capability or subcontract component availability (Kilpi and Vepsalainen 2004). In both cases the operator's inventory must be composed of ready-to-replace components.

As mandated by many civil aviation authorities, maintenance programs rely on condition monitoring (CM) to track the performance of the different parts and components of an aircraft.

An aircraft operator‘s program usually handles many repairable components of the same type. Such components could be in one of three places within the system:

1. In service on one of the aircraft
2. Undergoing repairs in the maintenance shop
3. In the spare part inventory

Each component has a unique serial number S/N$_i$. Most components have exhibited many installation and removal instances throughout their lifetime within the population. These instances are noted and logged in records (removal records) which are kept for every single component.

The integrity of components installed as replacements to failing parts is essential to the viability of the operator's asset management program. Typically, a repairable component works as expected for its designed lifecycle or between scheduled events (Carroll 2008). In some cases, a component fails to fulfill these expectations for three possible reasons:

1. The component has a manufacturing flaw which can be detected as it exhibits a failure or a series of failures in its early service life.
2. The component is aging and is thus suffering from consecutive failures towards the end of its service life.
3. The component is classified as rogue. Rogue components are repairable parts that develop a failure mode outside the scope of the standard repair and overhaul procedures.

Each of these reasons has identifiable characteristics. Rogue components, however are extremely difficult to identify and can spread throughout the component population.

If an asset management program includes repaired, reconditioned or overhauled parts, there is an ever-present risk of "rogue" components developing in the population. When this happens, there is a compounding negative effect across these aspects of operational and maintenance organizations:

- Operational Reliability
- Asset Management Programs
- Maintenance Effectiveness
- Preventive Maintenance Programs
- Maintenance Support & Training Programs
- Component Repair Facility
- Components Themselves
- Mechanical System Hardware
- Operator/OEM Engineering

The main problem occurs when rogue components slip into the asset management system through the operator's spare parts inventory. The detection of such components is important to ensure the reliability of the system.

A study was performed in 1995 to calculate the financial impact to an airline when a rogue component develops. It was determined that on the average, a single rogue component will cost $50,000 (US) over its life. This number pertains only to the maintenance burden and does not include flight delays and / or cancellations or flight restrictions because of the perpetuated system problems when installed to correct an aircraft system problem. Additionally, due to the high usage of spares as a result of multiple installations to resolve the perpetuated system problem, the asset management program may procure additional spare inventory, resulting in abnormally high inventory levels and an increased manpower hours. In an effort to resolve the unconfirmed failure in the shop, the OEM may elect to modify the component. Most times, these modifications are ineffective, and the airline bears the brunt of the cost. This is an extremely costly failure mode and impacts the effectiveness of many different aspects within the airline maintenance organization.

Condition Based Maintenance (CBM) is defined as perpetual monitoring of a system‘s health such that maintenance is performed when an intervention is deemed necessary. Rogue component detection is an example of how CBM can be used in aviation to detect faults. Today commercial airline operators have adopted several aspects of CBM; and with the advancement of technology, they will be able to adopt its full benefits (Teal and Sorensen 2001).

Current practice in rogue component detection

Current condition monitoring methods rely generally on statistical analysis tools or different combinations of parametric and non-parametric tools in order to evaluate the performance of aircraft components. The main drawback to the use of statistical tools is the precondition that the collected failure data are homogeneous and independent and identically-distributed (i.i.d). Many statistical analysis methods assume that the data belong to a certain probability distribution; such assumptions are not always true. An example is presented in Leung et al. (2007) where a hybrid parametric and statistical technique is used to classify aircraft components according to their maintenance status using their removal records as input data. The classification decision is done manually based on a visual evaluation of the output charts. In Carroll (2008), a set of indicators where proposed that would identify whether such a component is rogue or not by assessing its installation and removal history.

This paper is organized as follows: first we define rogue components and explain how they develop by outlining the "natural selection" phenomenon. Next we examine some of the negative impacts caused by rogue components by recounting two possible real life scenarios. Then we propose the use of Logical Analysis of Data (LAD) as a decision model for detecting rogue components within a population and describe the indicators involved in rogue component detection. We explain LAD methodology and explain its implementation in rogue component detection. Finally, we test the LAD technique on data obtained from the industry and study the results.

Rogue component definition

A rogue component is defined as an individual repairable component, which repeatedly experiences consecutive short in-service periods, manifests the same mechanical system fault each time it is installed, and when it is removed from service, the mechanical system fault is corrected.

The reason a component develops a rogue failure is because its repair and/or overhaul tests do not *address* 100% of the component's operating functions, characteristics or environment. Interviews with various Original Equipment Manufacturers (OEM) revealed the test coverage is typically about 85% of the component's complete functionality. Even if all the functions were covered, the operating environment of the component when it is installed in the mechanical system is usually quite different than the repair facility, so if a failure is dependent upon a particular in-service environmental condition, it is unlikely that it will be duplicated during test.

Additionally, the repair and / or overhaul tests are developed to identify *anticipated* failures, focused on testing things that are expected to fail. For example, it would not make sense to check all the screws or electrical ground straps each time a component comes into the shop, since the chance of failure for those pieces is practically zero and the cost of performing such extensive testing during each shop visit would be exorbitant.

When a component experiences a failure that was either *unaddressed* or *unanticipated* by the testing procedures, a rogue is born. Since every test that is performed misses that specific aspect of the component's functionality, the fault will never be identified and resolved (Leung et al. 2007).

The rogue failure cannot be predicted if, when, and where it will occur. It is a random failure that develops and will remain until definitive action is taken to resolve it. Not every part number population will develop rogue failures. Also, when a rogue failure occurs, not all the individual components within that part number population will necessarily develop that failure. However, any part number population has the potential for individuals to develop rogue failures, regardless of how simple or complex the design and functionality.

Natural selection phenomenon

There is a Darwinian-like "natural selection" process that ensures the rogue components will be positioned in the most disadvantageous places in the asset management program. The following depiction demonstrates the mechanics of this "natural selection" phenomenon.

Initially, it starts with a spare inventory and in-service population that are comprised of serviceable (Good) components that function as expected. As a part fails in service, it is removed and replaced with a good part from the spare pool in order to solve the mechanical system problem. The component repair facility tests and duplicates the problem with the failed unit, repairs and returns it to the spare pool.

The "natural selection" process begins when a rogue failure develops in one of the in-service components. When this occurs, the component is removed and sent to the repair facility. It typically tests normally, as "No fault found" (NFF), and returns to the spare pool with no corrective action taken to resolve that failure.

As long as there are no failures in the in-service population, the rogue component will remain in the spare pool. If the unique rogue failure mode is not recognized and resolved, then other components may develop the same condition.

Every new rogue component is removed from service and sent to the repair facility where it tests as NFF, and is returned to the spare pool. As such, the potential negative effect of the rogues is multiplied.

Though these rogue components make up a very small part of the general population, the "natural selection" process ensures that they are sorted out to the most critical place in the asset management process—the spare inventory. According to accounts from experts in the industry, there are documented cases where the entire spare pool had been comprised of rogues.

The effect of rogue components

When rogue components develop within a part number group, there are significant detrimental impacts to various aspects of the operational and maintenance organizations. These impacts will below.

Maintenance effectiveness

Mechanical system problem resolution relies on the spare inventory being comprised of serviceable components. When a component is installed from the spare inventory and the system problem continues, it is illogical to assume that the replacement was a defective part. When a rogue component is installed, it severely compromises maintenance effectiveness. The following scenario describes an actual case:

Case study

There is a system that maintains a constant air pressure by adjusting the opening of a vent valve to react to operational and environmental changes. This system is comprised of an electronic control unit, various sensing units, and a vent valve.

A system malfunction occurred that caused the vent valve to intermittently lock up in mid-position during high operational demands. The maintenance technicians could not duplicate the fault, so they replaced the control unit as the most likely component to cause this problem.

The problem repeated. Since the control unit did not resolve the problem, the vent valve was replaced, which required considerable system down time and maintenance resources. Now when the system operated during high demand periods, the valve intermittently oscillated open and closed, when it should remain in a fixed position. This problem could not be duplicated by maintenance.

Since this new issue surfaced immediately after the installation of the valve, it was replaced again in the assumption that it was defective from stock. The system was down again for a considerable amount of time during this second replacement. However, the oscillation problem continued.

All the wiring was checked leading to the valve, and after a number of additional repeat complaints, all the valve electrical connectors and sensors were replaced, with no result. The control unit was replaced again and the oscillation problem was resolved.

Root cause analysis

The root cause of the initial system malfunction (when the valve would stop during operation) was a faulty vent valve. The control unit first installed was a rogue component, which had an existing failure that would cause the valve to intermittently oscillate during high operational demands.

However, this rogue failure could not manifest itself until a serviceable vent valve was installed, since the original defective valve would lock up during operation, thus preventing the oscillation from occurring.

This type of compound problem is not common. Usually the introduction of a rogue component causes the original system problem to continue, which results in the replacement of the associated system components, extensive system troubleshooting and repeat replacement of the rogue component until a "good" spare is installed.

Mechanical support

When a chronic system problem that is caused by the introduction of a rogue component persists after all the components have been replaced, the next logical step is to troubleshoot the interconnecting wiring or plumbing.

It is very likely that much of this hardware is located in areas that are very difficult or time consuming to access, possibly requiring special tooling or OEM expertise to disassemble and reassemble. In some cases, OEM engineering drawings or wiring schematics are also needed in order to proceed with the next phase of troubleshooting, which can take a considerable amount of time and / or expense to acquire.

Since the root cause of the continuing problem is actually rogue component, this in-depth troubleshooting and extensive maintenance support will not resolve the system malfunction.

Operational reliability

When the maintenance effectiveness is compromised by the presence of rogue components, the mechanical system operational reliability naturally suffers. There are repeat events of system failures and associated down time, along with extended periods of in-depth troubleshooting.

Case study

An Auxiliary Power Unit (APU) provides electrical and pneumatic power, comprised of a turbine and a generator, with a main electronic control, and a number of external sensors. One of these sensors is located in an actuator that opens a door to allow air to enter the APU during operation. It is a switch that provides a signal to the electronic control unit that the door is open, so the APU can be started and allowed to run.

If the door should start to close at any time, the switch will immediately signal the electronic control unit to shut the APU down to prevent catastrophic damage. The electronic control unit also has a monitoring circuit to record which stage of the start or run cycle had failed, providing direction for system troubleshooting.

In this case, a door actuator had failed, which caused the APU to shut down when it was running. It was replaced with a rogue door actuator. Now the APU would intermittently shut down during various stages of the start cycle. This problem could not be duplicated during system troubleshooting, so maintenance reacted to the fault codes recorded by the electronic control unit.

Since there were different fault codes each time, a considerable amount of various components were replaced and the interconnecting wiring was checked a number of times. New wires were strung between the electronic control unit and the APU.

When another door actuator was installed, the problem stopped. This recurring problem generated 45 complaints that spanned a period of 344 days, with a total of 46 days of complete system shut down.

Root cause analysis

The first door actuator that was installed was a rogue component that had an intermittent failure of the sensor, which would indicate the APU door was closing when it was open.

Because this malfunction intermittently happened during different stages of the APU start cycle, the electronic control unit's fault recording system would record the each stage of the start sequence that was interrupted and list the most likely device that could be responsible for causing that failure at that particular time in the start cycle.

Unfortunately, the monitoring system would not record that the door actuator switch had signaled the door was closing during the start cycle.

Asset management

When a significant portion of the spare inventory is comprised of rogue components, traditional asset management models are no longer effective.

Typically, multiple spares must be withdrawn to resolve in-service problems, resulting in sporadically high spare usage and low spare levels. If the available spare inventory repeatedly reaches critically low levels, then more spares will be added. As more rogue components develop, this process will repeat until there is an abnormally high number of spares, which cannot be managed effectively.

Case study

An operator had a fleet of 40 aircraft, each having an autopilot system comprised of a control panel, pitch computer, roll computer, and a number of servomotors and sensors. The asset management program determined that 6 pitch computers were needed for the spare inventory to maintain a satisfactory level of support.

After a number of years, it was difficult to keep the spare levels up, so more computers were procured. It was assumed that the equipment was getting older, so the increased usage of the spares was a natural progression. This chain of events repeated as the years went on until there were 28 spare computers to support the 40 that were in service

Root cause analysis

Initially, the pitch computer population developed a small number of rogue components, which was a substantial percentage of the spare population. The result was a recurring low spare level, so more computers were procured to offset the demand. As new computers were added to the spare inventory, the percentage of rogue to non-rogue spares was reduced, so it was possible to maintain a satisfactory spare level, despite the rogue component presence.

Over time more rogue components developed, again increasing the rogue to non-rogue percentage in the spare inventory, with the same reaction from the asset management program, which diluted the rogue component impact to the asset management program. This cycle continued with the incremental increases to the spare inventory until extremely high levels were obtained.

After an analysis of the pitch computer population's in-service performance, it was discovered that 20 of the 28 spare computers were rogue components. Once these were identified and resolved, it was possible to surplus 20 of the spares. Each computer was valued at approximately $12,000 (US), so the cost of acquiring the excess inventory totaled around $240,000 (US). When the excess components were sold on the surplus market, only a small fraction of the initial expense was recovered.

Preventive maintenance programs

Some major components receive regularly scheduled preventive maintenance to ensure they operate through their designed life cycle, such as oil and filter changes. If rogue failures develop in these components, then an increasing number of in-service failures will occur despite these preventive maintenance actions. In an effort to eliminate these failures, typically the interval between preventive maintenance actions will be reduced from what was originally set. This is a very expensive action to take, as it could double or triple the recurring maintenance burden and cost. If the rogue failure mode is not corrected, then the failures will still continue despite the additional preventive maintenance.

Case study

On a turbine engine, the Constant Speed Drive (CSD) gearbox drives the electrical power generator at a constant RPM, regardless of the engine RPM. This gearbox has a preventive maintenance program in place to replace the oil and filter every 1,000 operating hours. After several years of operating these engines, several CSDs exhibited failure mode that resulted in oil starvation and catastrophic failure.

The immediate plan to resolve this situation was to change the oil and filter every 500 h, instead of the original 1,000 h.

With a CSD population of 240 units that had the filter and oil replaced about three times a year at a cost of $150 and 2 man-hours labor, the total annual maintenance burden was approximately $108,000 (US) and 1440 man-hours per year. Reducing the preventive maintenance interval to 500 h would double the cost and man-hour consumption.

Root cause analysis

Of the total CSD population, only 10 had exhibited this fault, but had done so repeatedly. These individuals developed a rogue failure that caused the oil pressure to fluctuate and damage an oil pressure relief valve, which then starved the CSD of oil.

An analysis of the rogue components revealed the unusual failure, which was resolved. The oil and filter interval remained the same and the reliability returned to the previous level.

Maintenance training programs

If mechanical system problems become chronic, it appears that maintenance efforts are ineffective, so the formal maintenance training programs are typically reassessed in an effort to raise the technical expertise.

When no formal technical training exists for those troublesome mechanical systems, then courses may be created to improve the overall understanding of system description, operation, troubleshooting and repair. If a formal technical training program exists, then the course material must be lacking, so a great deal of time and effort is spent to amplify the various aspects of the training to provide more detail.

When the maintenance effectiveness still does not improve, then the technical personnel may be required to attend recurrent training, assuming that repeated exposure to the same information will improve their expertise.

In all these situations, the expanded / additional / recurrent training will typically have little positive effect, since the root cause of the issue is not system knowledge, but rogue components. In addition, the maintenance personnel generally have a good understanding of the systems they work with, so subjecting them to additional training can convey the impression that management believes they are technically deficient, rather than taking action to identify and resolve the root cause of the problem. This can create or compound a division between management and the technical workforce.

Component repair facility

Rogue components cause a sporadic rate of removals, so the component repair facility has a correspondingly sporadic workload. Typically, there are periods of relative inactivity that are punctuated by high demands, can exceed the repair facility's manpower and testing capability.

The resulting low spare inventory levels, high repair backlog and extended lead times can force a selective type of testing that centers on the components that require the least amount of work, as satisfying the demand for serviceable spares outweighs the need to perform the necessary in-depth analysis of rogue failures. This tactical approach perpetuates the existing rogue component population and allows more to develop, which amplifies the demands and difficulties for the repair facility.

Case study

A certain component required three elapsed hours to perform a serviceability test, 12–14 h to calibrate, and 20–24 h to overhaul. Any failure that was above and beyond the typical overhaul could take upwards of 40 h to repair.

Because of a rogue component presence, the unserviceable components arrived in batches, which severely taxed the two test stations in the repair facility and created a significant backlog. Additionally, the resulting low spare inventory levels pressured the repair facility to produce serviceable components quickly in order to support the needs of the operation. If one or more of the components required overhaul or extensive repair, then the remaining backlog was audited to determine which ones could be turned around, that is, tested with no adjustment, repair or overhaul required.

Once several components were returned to the spare inventory, the production pressure lessened and the more time-consuming repairs or overhauls could resume.

The unserviceable components that could be turned around were the ones that had been replaced as a result of poor troubleshooting and the rogue components, as they both tested normally. Since there was no in-depth analysis of the rogue failures, the rogue component population grew, increasing the volume of the sporadic returns and intensifying the pressures on the repair facility to produce serviceable spares more quickly.

The turn-around methodology became a standard operating procedure, which became a self-feeding rogue component problem.

Operator/OEM engineering

When a significant rogue population develops, the number of system complaints grows and the repair facility has a high rate of NFF. As the operational reliability continues to decrease despite all the maintenance technical expertise improvements, then the operator or OEM engineering may be tasked with identifying the root causes.

Since a definite problem cannot be identified, then the efforts turn to theorizing what could be a root cause and component or system design modifications may be developed in an attempt to resolve the assumed shortcomings. Generally, the reliability improvement modifications do not address the true root cause, which is the rogue failure.

The poor operational reliability continues, with the risk that the incorporated change can also negatively impact the reliability of the general population.

Rogue components can present another challenge when a modification is introduced to enhance the operation of in-service components, such as a functionality or performance change. When these upgrade modifications are started, the spare inventory is modified as "seed" units, and then placed into service to remove the next wave of components to be modified. This process continues until all the modifications have been accomplished.

If the spare inventory contains a significant amount of rogue components, it will critically impact the modification campaign. The "natural selection" phenomenon ensures all the rogue components are in the spare inventory. When these components are modified and placed in service at the same time, they create their natural system failures. Since multiple system faults appear coincidentally as the modification was introduced, it is logical to assume the modification was the root cause of this sudden spike in operational problems.

The engineering group will typically halt the modification, so they can analyze each aspect of the modification, looking for something that was introduced that could cause such an adverse reaction. However, since the analysis does not focus on the rogue failure, it will consume a tremendous amount of manpower and resources for nothing. In some cases, a completely new modification will be developed—with the same results.

Case study

The heart of an engine indicating system is a computer that processes all the various inputs and displays the operational parameters on a monitor. This computer has an internal testing system that continually checks its functionality. If it detects an internal or external anomaly, it will display a fault message. In this case, there were 46 of these computers in service, with excellent overall operational reliability.

A modification to the computer software was introduced that changed the display characteristics. As the first batch of modified units was placed into service, a high number of system failures immediately occurred. When the modified computers were removed from service and tested, there were no faults found.

It was assumed that something in the new software must be the cause of these anomalies. A great deal of engineering time was spent reviewing all the software changes, but nothing could be identified as a root cause. The modification was halted.

Root cause analysis

The spare inventory had a significant number of rogue components (approximately 75%). When all the modified rogue components were placed into service at the same time, there was an abnormal spike in the amount of system faults, and when the modified computers were removed, the system reliability returned to normal.

The engineering group could find no problems with the modification, so an analysis of the in-service performance of all the modified components was initiated. It revealed that approximately 25% of them did not exhibit any problems when placed into service.

If the software modification was the root cause, then all the modified components should have exhibited faults. Since a segment of the modified components had no faults, the modification was exonerated. However, a considerable amount of engineering time and resources were expended needlessly analyzing the modification.

Mechanical system hardware

For the most part, rogue components create intermittent system faults. When a system problem persists after all the components have been replaced, the next logical step is to suspect an intermittent malfunction of the interconnecting wiring and connectors that could be caused by dynamic operational conditions, such as vibration, flexing, heat, cold, water ingression, etc.

Generally, the maintenance technician will attempt to replicate these conditions by subjecting it to physical stress and environmental conditions that could immediately create a new problem or weaken the wiring or connectors so another intermittent problem will develop in the future.

Typically, an ohmmeter is used to check the continuity of the wiring, which is measured with two metal probes. In order to accomplish the checks, one probe might be inserted into the female pins of the electrical connectors, which can consist of a high number of very small gauge pins. If the probe is not the exact size of the male counterpart, when it is inserted it can spread the internal contact points of the female pin, which will create an intermittent connection when the connectors are rejoined. When this occurs, the troubleshooting of this induced fault is extremely difficult to locate and resolve.

Another method of identifying a wiring problem that is intermittently shorting to ground is to use test equipment known as a "megger", which uses a high voltage to determine if the wiring insulation is breaking down. If it is not used correctly, it could damage the insulation. Additionally, if all the interconnected electrical components are not disconnected, it will damage their internal workings, creating additional system faults.

Components themselves

As an inordinate amount of components are replaced to resolve a single system problem caused by a rogue component, damage can occur during the removal, installation, and shipping of the components to and from the repair facility. Additionally, damage can occur during installation from electrical or pressure surges during the connection / disconnection of the components, which could create another intermittent fault. All of these scenarios are very expensive and time consuming to resolve.

Control of rogue components

Rogue components cannot be prevented. It is impossible to proactively anticipate a failure that could occur and develop a new test to identify it before it happens. Therefore, the only action that can be taken is reactive, which is to detect and isolate rogue component from the population they're embedded in. Once detected and isolated, their unique failure modes can be analyzed in order to develop tests to identify them in the future.

The first step in the detection of rogue components is to develop a data collection system that captures system maintenance events and tracks the installed / removed components by part and unique serial number.

By monitoring certain indicators in the data collection system, patterns that are unique to rogue components can be discovered. Carroll (2008) reported the following patterns that are unique to rogue components after years of manually monitoring repairable component removal records:

1. Repeated short in-service installation periods. Shortness of the period is determined by comparison to a typical service life time of a component. A third consecutive short in-service time triggers the rogue flag.
2. Repeated identical reasons for removal. If the component exhibits identical system fault manifestation for the last 3 removals, then the rogue flag is triggered.
3. Shop records indicate that the failure cannot be detected by standard testing procedures: No Failure Found.
4. Removal of the failing component from the operating system resolves the system fault. If the system is still at fault even when the component is removed, then this means the rogue condition is not satisfied.

If these patterns or occurrences are found in a certain component's removal record, then that component can be classified as rogue. It takes the presence of all the above criteria to be able to classify a component as rogue.

Current practice in the identification of rogue components involves searching through thousands of removal records manually and detecting visually the above mentioned patterns in order to extract these outlier components. The automation of this process through an automatic decision model that classifies repairable components into two classes: (1) Rogue and (2) non-rogue, provides a better solution to this problem. LAD, as a decision model that is capable of automatically generating patterns from input data, is an ideal method to automate the above process.

In what follows is a description of the LAD methodology and its implementation in rogue component detection.

LAD methodology

LAD is a data mining technique that classifies observations into the categories they are associated to. The history of this method goes back to 1988 where it was first proposed in Cama et al. (1988) as a method for classifying binary data. LAD has been proven to give comparable and even superior results in some cases to the traditional decision models used in CBM, such as neural networks and support vector machines (Salamanca 2008; Boros and Hammer 2000). The main advantages of LAD are:

1. It is not based on statistical analysis. Consequently, it does not assume that the data belong to a specific statistical distribution. The method therefore does not require statistical analysis of data prior to its use.
2. LAD automatically extracts features and generates patterns from the indicators collected from the observations and, accordingly, sorts the components into separate classes based on the patterns generated.
3. Unlike other data analysis techniques, such as neural networks and support vector machines, LAD is a transparent method; the output of LAD can be traced back to the specific root causes that resulted in the categorization of a specific observation into a certain class. This explanatory power, a potential asset to maintenance experts, is attributed to the patterns that LAD can generate from the observation and analysis of criteria that are pertinent to the classification problem.

As LAD is a supervised learning technique, it relies on the presence of training data, already sorted into the existing classes, in order to generate the patterns. Training data are a learning set of pre-classified observations based on which the algorithm develops its decision function. In the case of rogue component detection, these observations are the records of installation and removal (removal records) of some components in the population whose *rogueness* or *nonrogeness* is already confirmed. A typical training set is composed of two subsets: a positive observation subset composed of rogue observations and a negative observation subset composed of non-rogue observations.

After the acquisition of training data, the LAD algorithm can be divided into three steps:

1. Data binarization
2. Pattern generation
3. Theory formation

Data binarization

The information extracted from the training observations is binarized prior to analysis. Each observation can be considered as a vector of m indicators. As LAD is based on discrete mathematics and combinatorial enumeration, its input, the observation vectors formed by the non-binary indicators, are transformed into Boolean observation vectors of n binary attributes.

The binarization of non-binary indicators depends on their type. Indicators can be divided to two categories: descriptive indicators (e.g. code type) and numerical indicators (e.g. time, temperature, etc. ...). Descriptive indicators can take up many possible values. Binarization, in this case, occurs by allocating to each value v_n of the indicator x a Boolean variable $b(x, v_n)$such that (Boros and Hammer 2000):

$$b(x, v_n) = \begin{cases} 1 & \text{if } x = v_n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Numerical indicators are binarized using two types of binary variables: Level variables and Interval variables. Level binary variables are obtained by first sorting the values of the numerical indicator in the observation set in descending order and then introducing 1 cut-point between each interval $v_n < v_{n-1}$ such that $v_n \in S^+$ and $v_{n-1} \in S^-$ or vice versa, where S^+ and S^- represent the positive and negative observation subsets respectively and the cut-point t is calculated as (Boros and Hammer 2000):

$$t = 0.5(v_n + v_{n-1}) \quad (2)$$

The resulting binary attributes are Boolean variables defined by each cut-point t such that (Boros and Hammer 2000):

$$b(x, t) = \begin{cases} 1 & \text{if } x \geq t \\ 0 & \text{if } x < t \end{cases} \quad (3)$$

Interval binary variables, as the name implies, take the value of 1 when the value of the numerical indicator is within a certain interval and 0 otherwise. These intervals are formed by the cut-points calculated for the level variables. An interval binary variable of a numerical indicator is therefore obtained from every two cut-points found for that indicator while calculating the level variables, and would have the following form (Boros and Hammer 2000):

$$b(x, t_1, t_2) = \begin{cases} 1 & \text{if } t_1 \leq x \leq t_2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The cut-points t_1 and t_2 belong to the level binary attributes obtained for the numerical indicator.

The outcome of the binarization of an observation set is a set of Boolean observation vectors with a number of attributes n exceeding the initial number of non-binary indicators $m(m > n)$. For a total of R observations, we thus obtain R Boolean observation vectors $O_1, O_2, \ldots, O_R$ of dimension n. A Boolean observation vector has the form $O_r = y_1 y_2 y_3 \cdots y_n$ where y_i is a binary digit.

Pattern generation

After transforming the observation set into Boolean observation vectors of dimension n, a bottom-up pattern generation approach is implemented to generate the patterns. This approach starts by finding a binary variable that *covers* one or more observations. Such a variable is called a *literal* in algebraic terms. A literal that covers an observation O_r has the form b_i if the value of y_i in O_r is 1 and the form $\overline{b}_i$ if the value of y_i is 0. A combination of literals is referred to as a *term*. A term is said to cover a certain observation when all the literals of that term cover the Boolean observation vector. For example, the term $\overline{b}_2 b_3 b_4$ covers the observation $O_r = 1011101$ since the value of that binary observation vector at the digits $y_2 y_3 y_4$ is 011. Similarly, the term $\overline{b}_1 b_2 \overline{b}_3 b_4$ covers an observation $O_r = 0101011$. A term is said to be of degree k if it is composed of k literals. For example, the terms in the examples above are of degree 3 and 4, respectively.

If a literal covers both positive and negative observations, then it is considered a *candidate*. More literals are added to it progressively, each time checking whether it still covers observations. If by adding more literals, the number of observations covered becomes zero, then that particular term is discarded. Otherwise, the term keeps its candidate status as long as it covers at least one positive observation and one negative observation or vice versa. If, by adding another literal, the resulting term covers only positive (negative) observations, then it is considered a positive (negative) prime pattern. This methodology favors the generation of small patterns, thus following the simplicity principle (Boros and Hammer 2000). In order to reduce the amount of computations necessary, the lexicographic order is followed in generating the patterns (Boros and Hammer 2000):

$$b_1 < \overline{b_1} < b_2 < \overline{b_2} < \cdots \quad (5)$$

The number of terms to be searched for patterns increases exponentially with the number of binary attributes that constitute a Boolean observation vector. For n attributes, the total number of terms is given as:

$$\sum_{i=1}^{n} 2^i \cdot \binom{n}{i} \quad (6)$$

For example, for a number of attributes $n = 45$ which is typical of a problem of this nature, the total number of terms to be searched is 2.95431×10^{21}. Therefore, due to computational and time constraints, a limit is set on the maximum degree of terms to be searched for patterns.

Theory formation

The generated positive patterns which cover the positive observations are denoted by $P_1, P_2, \ldots, P_k, \ldots, P_K$, whereas the negative patterns are denoted by $N_1, N_2, \ldots, N_l, \ldots, N_L$. These positive and negative patterns are used to produce a discriminant function which, in the context of this paper, can separate rogue components from non-rogue ones. This function is of the form (Boros and Hammer 2000):

$$\Delta\left(O_r\right)=\sum_{k=1}^{K} w_k^{+} P_k\left(O_r\right)-\sum_{l=1}^{L} w_l^{-} N_l\left(O_r\right) \tag{7}$$

where the value $P_k O_\gamma$ is one if the positive pattern P_k covers observation $\left(O_\gamma\right)$ and zero otherwise. Similarly, the value $N_l\left(O_\gamma\right)$ is one if the negative pattern N_l covers observation O_γ and zero otherwise. The resulting discriminant Δ is thus the weighted sum of the values of all the generated positive and negative patterns for a certain observation O_r. The weights w_k^+ and w_l^- can be calculated in multiple ways. The method used here is to compute the weight of a pattern as a normalized function of the number of observations it covers (Salamanca 2008):

$$w_k^{+}=\frac{\sum_{r=1}^{R} P_k\left(O_r\right)}{\sum_{k=1}^{K} \sum_{r=1}^{R} P_k\left(O_r\right)} \tag{8}$$

The negative weights are calculated similarly.

The output of the discriminant function shown in (7) is therefore a value between −1 and +1. If the value of Δ is closer to +1, then the observation is classified as positive (Rogue). If the value of Δ is closer to −1, then the observation is classified as negative (non-Rogue). A value close to 0 indicates that the results are inconclusive, therefore no classification takes place. A threshold $\pm\tau$, set by the user, is the smallest value beyond which the observation is regarded as unclassified. The LAD algorithm decision function can therefore be formulated as:

$$f=\begin{cases}\text{Positive} & \text{if } \Delta \geq +\tau \\ \text{Negative} & \text{if } \Delta \leq -\tau \\ \text{Unclassified} & \text{if } -\tau<\Delta<+\tau\end{cases} \tag{9}$$

The above decision function can be used to test for the rogueness of any new observation:

The necessary indicators are extracted from its removal records and binarized. Then, the resulting Boolean attributes are plugged into the discriminant function to get Δ. The decision function then reveals to what class that specific component belongs to.

Implementation

In CBM, the detection of a fault can only be achieved if there exists a set of indicators that can reveal information about the status of the asset by monitoring them.

LAD, as a supervised learning decision model, has only recently been adopted in CBM in Salamanca (2008). Implementing LAD for the purpose of detecting rogue components requires the preparation of training data in the form of observation vectors before binarization can occur. As explained in the previous section, these observation vectors are formed by the indicators used to monitor the component's status in CBM. The binarization step then transforms these observation vectors to Boolean observation vectors.

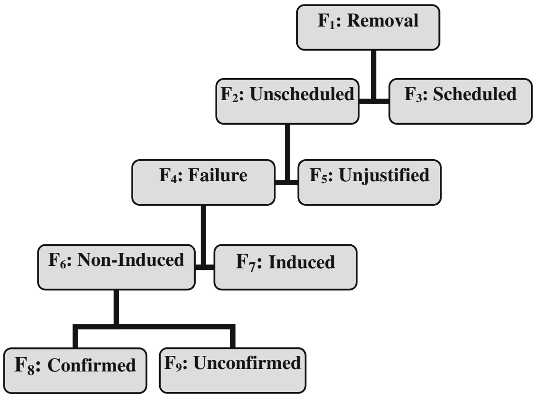

Fig. 1 Fault confirmation codes as presented in Leung et al. (2007) describe the nature of the removal

In the case of repairable components of an aircraft fleet, the indicators that form the observation vectors are extracted from the indicators in the removal records of these components. Judging from the criteria that characterize rogue components, the following indicators found in the components' removal records can be extracted and used to form the LAD observation vectors:

1. Fault Confirmation Codes (FCC): When a component is removed, it is taken to shop for check-up and repair. After each repair, a "Fault Confirmation Code" is added to the component's record. There are 9 possible removal confirmation codes: F_1, F_2 . . . and F_9. As shown in Fig. 1, a these codes describe what kind of removal had occurred, whether the removal was scheduled or not, whether a failure was justified or not, whether it was induced or not, etc. . . . A combination of those codes will describe the removal (Leung et al. 2007).
2. Reason for Removal Codes (RRC): These codes describe the cause or mode of failure of the component (e.g. leak in sealing area, wear in bearing, etc. . . .). One component can have a mixture of reasons for removal describing the same failure incident. For a given component type, q known possible RRC codes may exist.
3. Time-to-Removal (TTR): This is the amount of time (i.e. number flight hours) the component spent in service before it was removed. This is measured as the time between installation and removal. This number is sometimes multiplied by a constant d between 0.5 and 1 that is chosen based on some known utilisation characteristics of the aircraft the component was used in Leung et al. (2007).

Classification within the maintenance process

The ability to use the indicators mentioned above depends on where, in the maintenance process, rogue component detection occurs. Implementation of the LAD algorithm can take place at one of two points in the process: before or after the component enters the repair shop.

By performing the detection before the repair stage, any unnecessary resources that may be expended on a rogue component can be saved. However, the disadvantage of detecting rogue components at this point is that FCC cannot be used as indicators. Consequently, the LAD algorithm would have to rely on the two remaining indicators to come up with a decision about the rogueness of a certain component.

Performing classification after the component undergoes repairs allows for the utilization of the FCC codes as inputs to the LAD algorithm. The presence of additional evidence leads to a more educated judgment of the components' maintenance status. The disadvantage, however, is that these codes are hard to procure given the current structure of the aircraft maintenance process. In many cases aircraft component maintenance is administered by the OEMs themselves. Communication between the aircraft operator and the OEMs on maintenance matters is usually minimal. Consequently, obtaining information regarding what occurs in the repair shop may not always be possible.

It is worth mentioning that the extraction of maintenance data from an aircraft operator's logs is in many cases a tedious task. This is largely due to the fact that most maintenance data is generated for the goal of record keeping and not for utilization as an asset for the purpose of CBM.

LAD training table

To our knowledge, previous uses of LAD did not require taking into account historical values of the same indicator in generating the patterns and decision functions. However, in this situation, the nature of the observations from which a classification decision is obtained necessitates the incorporation of historical data into the set of LAD attributes.

Values for the three indicators mentioned above are recorded for every single removal instance of a single component. In the case of rogue detection, each component in the population has exhibited many removals in its life time. Therefore, the removal records of a certain component contain values for these indicators for every removal instance. Additionally, some components are older than others, and some have exhibited more failures than others. Therefore not all removal records contain the same amount of data.

In view of the above, it is difficult to obtain input observation vectors having a unique form if all the available information for each component is used. As such, the observation vector used to train the LAD algorithm is limited to nine non-binary indicators representing the three most recent FCC, RRC, and TTR values of a component. Example: we are given a training set of four rogue components and four non-rogue components, where each component has a recorded number of removals ranging between 3 and 9. We choose to limit the removal data we are going to look at to the three most recent removal incidents.

The reasoning behind this is that whatever pattern we would find will be clear to us by looking at the most recent removal data. This reasoning is deduced from Leung et al. (2007) where, in the visual graph obtained through the CH-method, the most relevant and pertinent data are the ones found in the top right corner, which actually represent the data obtained from the three most recent removals of the components.

The number 3 (i.e. the last three removals) is used in many cases in Leung et al. (2007) and Carroll (2008) when calculating factors or triggering rogue flags. While we will use this for illustration purposes throughout this paper, this number can be modified within the algorithm without any major structural change. Ultimately, the goal is to be able to consider the entire history of a certain component in the classification process.

The LAD methodology explained above has been adapted into a software program called CBM-LAD written in C++ at École Polytechnique de Montréal. This software is capable of treating the rogue detection problem explained above.

Results

The CBM-LAD software was used on real component data obtained from the maintenance department of NetJets Inc. The data was extracted from the maintenance records of 61 airplanes during a period stretching from March 28, 1999 to June 20, 2009. These records consist of 576 removal instances belonging to 150 turbo compressors. From the records of each component an observation vector was obtained as explained in the sections above. Of the available 150, 68 were used to train the LAD decision model and 74 to test the resulting model; the rest were discarded as incomplete records. The data shown in Table 1 show a portion of the training data. Two of the components shown in the table were judged as rogue by maintenance professionals. There are, in all, 13 negative observations representing normal components (grey) and two positive observations representing rogue components (white). Each observation represents information obtained from the removal records of one component with a unique serial number S/N_i. It is assumed here that the LAD algorithm is implemented before the component enters the repair shop. FCC codes are consequently absent from the table.

The 150 components did not enter into service at the same time, thus not all components exhibited 3 removals within their lifespan as most components exhibit one or two removals per 3 years for this type of part. This phenomenon is dealt with in Table 3 by placing close to infinity Time-to-removal values (99,999 days) and the 0 code for reason-for-removal

Table 1 Non-binarized training data

	Reason-for-removal code			Time-to-removal codes			
	Last	2nd Last	3rd Last	Last removal	2nd Last removal	3rd Last removal	
1	2	0	0	413.73	99,999	99,999	Negative (non-rogue)
2	2	0	0	21.99	99,999	99,999	
3	2	0	0	366.81	99,999	99,999	
4	3	2	0	194.72	477.67	99,999	
5	2	3	2	1288.99	196.70	125.15	
6	2	0	0	266.76	99,999	99,999	
7	2	0	0	1503.23	99,999	99,999	
8	2	0	0	0	99,999	99,999	
9	5	2	2	1045.42	1451.63	133.41	
10	2	0	0	212.47	99,999	99,999	
11	3	0	0	616	99,999	99,999	
12	2	3	0	284.08	539.97	99,999	
13	3	0	0	304	99,999	99,999	
14	2	2	2	144.08	57.6	132.7	Positive (rogue)
15	2	2	2	204	281.20	83.7	

to illustrate the absence of such events. The LAD table is then used for training the algorithm and producing a decision function.

The decision model was trained three times, each time with a different maximum allowable pattern degree. The degrees used were 2, 3, and 4. The resulting 3 decision models were tested in each case using the data set composed of 74 observations reserved for that purpose. The value τ was randomly set to 0.2 for all three decision models. The number of binary attributes obtained and the number of positive and negative patterns found for each decision model are shown in Table 2.

The values of the discriminant function Δ for the 15 observations shown in the previous table are presented in Table 3 for the three decision models obtained. The table shows that the score of the discriminant function is positive for the positive observations and negative for negative observations.

The results, part of which is shown in Table 2, reveal that the detection has been done successfully. The scores of the discriminant function for all the observations of the testing set give a negative value for the normal (non-rogue) components and a positive value for rogue components. However, since the threshold for considering an observation unclassified is ± 0.2, the result was not 100% successful for all pattern decision models.

In order to evaluate the performance of the resulting decision models, a number of performance measures are calculated using the proportions shown in Table 4. Each observation classified by the LAD decision model can be in one of the 6 situations shown in the table. The letters a, b, c, d, e, and f represent the proportions of classified observations found in each of these six situations.

The values a and d represent the proportion of positive and negative observations that are correctly classified, respectively. The values c and b are the proportion of positive and negative observations that are falsely classified, respectively. The values e and f represent the proportion of positive and negative observations that remain unclassified, respectively. The performance measures obtained from these values are:

$$\text{Quality of classification:} \quad Q = \frac{a+d}{2} + \frac{e+f}{4} \tag{10}$$

$$\text{The true positive rate:} \quad TP = \frac{a}{a+c+e} \tag{11}$$

$$\text{The false positive rate:} \quad FP = \frac{b}{b+d+f} \tag{12}$$

$$\text{The true negative rate:} \quad TN = \frac{d}{b+d+f} \tag{13}$$

$$\text{The false negative rate:} FN = \frac{c}{a+c+e} \tag{14}$$

The results for the three decision models obtained are shown in Table 5.

The results in Table 5 show that the 3 decision models obtained have a high classification quality Q. The classification quality increases significantly with the increase in maximum pattern size from 2 to 3 bits. Degree 3 and degree 4 show an equal performance. Additionally, all three models resulted in zero false alarms; i.e. no rogue components were

Table 2 Pattern numbers found for each decision model

	Max. degree 1	Max. degree 2	Max. degree 3
No. binary attributes	49		
No. negative patterns	25	125	125
No. positive patterns	7	274	330

Table 3 The value of discriminant function Δ for all three decision models

Observation	Max. degree 2	Max. degree 3	Max. degree 4
1	−0.7617	−0.8542	−0.8542
2	−0.6191	−0.6093	−0.6093
3	−0.6925	−0.6194	−0.6194
4	−0.8513	−0.8664	−0.8664
5	−0.5479	−0.7999	−0.7575
6	−0.6130	−0.6793	−0.6793
7	−0.7617	−0.8542	−0.8542
8	−0.6925	−0.6194	−0.6194
9	−0.6864	−0.6893	−0.6893
10	−0.6486	−0.4982	−0.4918
11	−0.6864	−0.6893	−0.6893
12	−0.6884	−0.8441	−0.8441
13	−0.6864	−0.6893	−0.6893
14	0.3794	0.5035	0.4137
15	0.1978	0.4535	0.3958

Table 4 Calculating the quality of classification

True class	Classification result		
	Positive	Negative	Unclassified
Positive	*a*	*c* (Type II error)	*e*
Negative	*b* (Type I error)	*d*	*f*

Table 5 Performance measures of the three decision models

	Max. degree 2	Max. degree 3	Max. degree 4
Q	0.8263	0.9965	0.9965
TP	0.3333	1	1
FP	0	0	0
TN	0.9718	0.9859	0.9859
FN	0	0	0

misclassified as non-rogue and vice versa. The true positive and true negative values also increased with the increase in maximum patterns size from 2 to 3. However, these values will change if the threshold $\tau$is changed from the set value of ±0.2. If $\tau$is decreased, for example, the number of false alarms will increase and the quality of classification measure will change.

In comparing the discriminant function values obtained from the models with maximum pattern degrees 2, 3, and 4, we notice that the scores for the positive observations increase in the degree 3 model and then decrease slightly for the degree 4 model. The rise in the values of Δ can be explained by the fact that a much higher number of positive patterns was found in the degree 3 model (274) compared to the degree 2 model (7). The scores, however, decrease slightly again in the degree 4 model even though the number of positive patterns found increases to 330. This decrease can be attributed to the fact that the third model generated degree

4 positive and negative patterns which are too specific, thus leading to a decrease in the discriminatory power of Δ, as a higher degree pattern has a lower chance of covering an observation than a lower degree one. In addition, judging from the rogue component characteristics discovered manually by experts and discussed in the sections above, any pattern we expect to find must relate three consecutive events to each other, as explained in the sections above. Degree 3 patterns therefore are more meaningful than patterns of the other degrees.

The advantage of the decision models obtained through LAD, besides their accuracy, is the interpretability of the decisions obtained from it. For example, one negative pattern found in the second decision model (degree 3) is: $b_{14}b_{18}b_{22}$. This pattern translates verbally to the statement:

> *"The three last reason-for-removal codes are all of value 2"*

Such a pattern is exactly what we would expect to have given the characteristics for rogue components explained above. The ability to translate the patterns leading to the decision to logical statements that could be understood by any maintenance technician is unique to the LAD technique.

Conclusion

In this paper, we studied rogue components, which plague the asset management programs in the aviation industry. We explained how these rogues develop and discussed their impact on the entire asset management program. We then described how to control such components and proposed the use of LAD as a decision model to solve the problem of detecting them.

Testing results showed that the LAD technique is capable of detecting rogue components automatically through feeding the components' performance history into the LAD algorithm. The automatic detection of rogue components solves the problem of having to sift through thousands of removal records in order to evaluate each component visually. A major advantage of its utilization in rogue component detection is, therefore, the huge amount of time and resources that it can potentially save. LAD is capable of accomplishing in seconds something which takes days currently in the industry.

The financial benefits are also evident. By a 1995 estimate, the maintenance burden to an airline of one rogue component is $50,000 (US). If 100 rogue components are detected using the LAD decision model, an aircraft operator's asset management system saves $5 million in maintenance costs alone. In addition to saved costs, early detection of such components also increases the safety and overall performance of the operator.

In applying LAD to rogue component detection, we were capable of generating the patterns that maintenance experts expected to see. As such, the ability of LAD to reduce dependence on their subjective opinions was demonstrated. The advantage of LAD, though, is that it is capable of detecting new patterns without previous knowledge or any aid from maintenance experts.

The automation of the evaluation of records for rogue component detection is a big step towards achieving CBM in aviation. It is however apparent that for achieving full CBM implementation in the industry, maintenance records must be regarded as assets and not as mere tracking logs.

Further work is going on in developing the LAD algorithm to include more sophisticated pattern recognition techniques. Further investigation of more effective measures to deal with incomplete data is also underway.

References

Boros, E., & Hammer, P. (2000). An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering, 12*(2), 292–306.

Cama, Y., Hammer, P. L., & Ibaraki, T. (1988). Cause-effect relationships and partially defined Boolean functions. *Annals of Operations Research, 16*(1), 299–325.

Carroll, T. (2008). The statistical outliers are in control of asset management. In *The maintenance and reliability conference MARCON 2008*. Tennessee, USA.

Flint, P. (2007). Balancing act: Rising demand for MRO services occurs against a backdrop of steady market evolution. *Air Transport World*. November 2007, 46–47. http://www.atwonline.com/magazine/article.html?articleID=2115. Accessed 24 May 2009.

Kilpi, J., & Vepsalainen, A. P. J. (2004). Pooling of spare components between airlines. *Journal of Air Transport Management, 10*(2), 137–146.

Leung, T., Carroll, T., Hung, M., Tsang, A., & Chung, W. (2007). The Carroll-Hung method for component reliability mapping in aircraft maintenance. *Quality and Reliability Engineering International, 23*, 137–154.

OAG. (2008). Global MRO spend on military aviation to increase by 14.9% over the next decade, reports OAG. *Official Airline Guide.* http://www.oag.com/oagcorporate/pressreleases/08+GLOBAL+MRO+SPEND+ON+MILITARY+AVIATION+TO+INCREASE.html. Accessed 29 April 2009.

Salamanca, D. (2008). Logical analysis of data applied in condition based maintenance. *M.Sc. Thesis, Department of. Industrial Engineering, Ecole Polytechnique de Montreal.*

Teal, C., & Sorensen, D. (2001). Condition based maintenance [aircraft wiring]. In *The 20th conference on digital avionics systems, Daytona Beach, FL*, 1, 3B2/1–3B2/7.