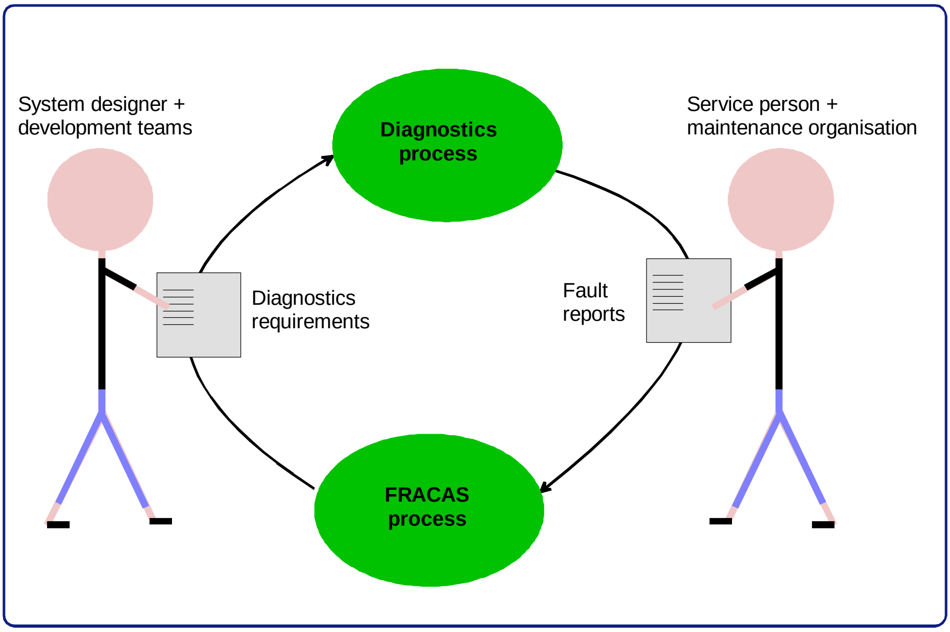
ESPOO 2006 VTT RESEARCH NOTES 2343



Jarmo Alanen, Kari Haataja, Otto Laurila,
Jukka Peltola & Isto Aho

Diagnostics of mobile work machines

VTT TIEDOTTEITA – RESEARCH NOTES 2343

Diagnostics of mobile work machines

Jarmo Alanen & Kari Haataja
VTT

Otto Laurila & Jukka Peltola
Helsinki University of Technology

Isto Aho
Plenware Oy

ISBN 951–38–6798–6 (soft back ed.)
ISSN 1235–0605 (soft back ed.)

ISBN 951–38–6799–4 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455–0865 (URL: http://www.vtt.fi/publications/index.jsp)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 3, P.O.Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

VTT, Tekniikankatu 1, PL 1300, 33101 TAMPERE
puh. vaihde 020 722 111, faksi 020 722 3282

VTT, Tekniikankatu 1, PB 1300, 33101 TAMMERFORS
tel. växel 020 722 111, fax 020 722 3282

VTT Technical Research Centre of Finland, Tekniikankatu 1, P.O. Box 1300, FI-33101 TAMPERE, Finland
phone internat. +358 20 722 111, fax +358 20 722 3282

Technical editing Maini Manninen

Valopaino Oy, Helsinki 2006

Alanen, Jarmo, Haataja, Kari, Laurila, Otto, Peltola, Jukka & Aho, Isto. Diagnostics of mobile work machines. Espoo 2006. VTT Tiedotteita – Research Notes 2343. 122 p.

Keywords mobile work machines, diagnostics, automotive industry, cars, heavy-duty vehicles, diagnostics architecture, MIMOSA, data analysis, data mining, sensor fault modes

Abstract

In this research note, we take a look at the field of mobile work machine diagnostics. The perspective by the authors is limited by the research project (KODIE) behind this research. However, we have tried to provide generic guidelines for machine builders to set up their diagnostics strategy. Building blocks, like SAE J1939/73, ISO 15765 and ODX, from automotive industry are exhibited to prevent machine manufactures from reinventing diagnostics protocols and practices. Furthermore, examples of diagnostics architectures are presented, with OSA-CBM among others. To make the most of diagnostics data, an extensive set of data analysis methods are introduced. And in order to help engineers to design diagnostics feature for the sensor system, hints and examples are supplied as to how to establish the fault modes of sensors; a good knowledge about the (relevant) sensor and actuator fault modes is a prerequisite for comprehensive fault detection.

Preface

This report is an outcome of the national KODIE project (Diagnostics and remote control of machines) carried out through 2003–2005. The project belonged to the MASINA technology programme organised by Tekes (Finnish Funding Agency for Technology and Innovation). The project was financed by Tekes, VTT, Kalmar Industries Oy, Plustech Oy, and Plenware Oy. The research work was carried out by Helsinki University of Technology and VTT.

We would like to thank all those who participated in this project.

Tampere, April 2006

Authors

Contents

List of acronyms

ABS	Antilock Braking System
AHM	Airplane Health Management
AI	Artifical Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AREMA	American Railway Engineering and Maintenance-of-way Association
ASAM	Association for Standardisation of Automation and Measuring Systems
CAN	Controller Area Network
CBA	Cost and Benefit Analysis
CBM	Condition Based Maintenance
CBR	Condition Based Reasoning
CBS	Condition Based Service
CCP	CAN Calibration Protocol
CCW	Counter clockwise
CRIS	Common Relational Information Schema
CW	Clockwise
DTC	Diagnostic Trouble Code
EOBD	Euro On-Board Diagnosis
EPA	Environmental protection agency
ESP	Electronic Stability Program
FCM	Fuzzy Cognitive Maps
FDA	Fisher Discriminant Analysis
FDI	Fault Detection and Identification
FIPA	Foundation for Intelligent Physical Agents
FMI	Failure Mode Identifier

FRACAS	Failure Reporting, Analysis and Corrective Action System
GOA	Generic Open Architecture
IEEE	Institute of Electrical and Electronics Engineers
IPG	Information Power Grid
ISO	International Organisation for Standardisation
KDD	Knowledge Discovery in Databases
LIN	Local Interconnect Network
MIMOSA	Machinery Information Management Open Systems Alliance
OBD	On-Board Diagnostics
ODX	Open Diagnostics Exchange
OMS	Onboard Maintenance System
OSA-CBM	Open System Architecture - Condition Based Maintenance
OSI	Open System Interconnection
PCA	Principal Component Analysis
PLS	Partial Least Squares
RAMS	Reliability, Availability, Maintainability and Safety
RUL	Remaining Useful Life
SAE	Society of Automotive Engineers
SDG	Signed Directed Graph
SOM	Self Organizing Maps
SPC	Statistical Process Control
SPN	Suspect Parameter Number
STIM	Smart Transducer Interface Module
TEDS	Transducer Electronic Data Sheet
VEG	Vehicle Expert Group
WWH	Worldwide Harmonisation group

1. Introduction

Diagnostics is not an easy task. The following issues make implementation of comprehensive diagnostics of work machines laborious:

- It is difficult to parse the whole picture of diagnostics into manageable pieces. Matters to be settled are, for example, the partition between on-board and off-board diagnostics, strategy for on duty and off duty diagnostics, creating corporate infrastructure to support telediagnosis and field service, providing service personnel training, producing diagnostics manuals on paper and in electronic formats (multimedia manuals), balancing between reactive, schedule based and proactive diagnostics (prognosis) and fulfilling both the legislative requirements and proprietary requirements. Creating an effective diagnostics strategy and a suitable set of requirements for the diagnostics features and services requires a significant amount of skills and experience.
- Designing of the diagnostics methods and algorithms is time consuming. If a good diagnostic method is found to detect faults of a certain measurement function, a different method may be needed for a different measurement even though it uses exactly the same sensor type. To learn the effectiveness of the invented diagnostic method may take a considerable time, as the number of machines in the field using the particular method may be low. Hence, the field feedback is slow. Furthermore, to minimise the risks posed by a fresh diagnostic method necessitates performing thorough theoretical evaluations and massive laboratory tests.
- Implementation of diagnostics software is more difficult than implementation of the actual application software. Testing and tuning of the diagnostic features especially is time consuming. Setting the fault reporting thresholds correctly is critical to prevent unnecessary fault reports, but a fault indication must always be ensured when a fault really prevails. However, the thresholds are not generic, but the machines are individuals and their working environments may vary from the cold climate of Nordic countries to the tropical climate. Hence, the fault reporting thresholds, and even the diagnostic methods, must be adapted to the specific machine and its working conditions.

- The complexity of the machine control systems may grow faster than the diagnostics skills, engineering resources and infrastructure support of the machine manufacturer company.

In the early 90's, the manager of an electronics sub-contractor of a Finnish machine manufacturer said that about 50% of the software development effort is put into programming and especially testing and tuning of the diagnostic features. Ogawa & Morozumi [2002] (from Toyota) report that (in year 2002) one third of the CPU capacity of the electronic modules of cars was spent on on-board diagnostics. Although to implement extensive diagnostics on work machines is such a tedious task, there is enough motivation to do it and to do it well:

- Diagnostics is one major way to improve the availability performance of the machine. Especially now (in year 2004) when the trend is to shift from machine sales to capacity trading, the availability performance is the quantitative measure that sets the price tag. In more traditional trade, better availability saves the customer money and keeps the customers happy. It should be noted, however, that not all diagnostics brings added value, but is a must to keep the availability at least at the same level as with the traditional, more mechanical, machines. In the context of mobile machines, the importance of availability performance may be higher than with e.g. personal cars due to the fact that a work machine does productive work that somebody pays for.
- In the case of emission related diagnostics, the legislation may require certain diagnostic services and a standardised access to those services.
- With diagnostics it is possible to defend functions against safety critical fault modes and signal deviations.

Diagnostics of work machines is thus clearly an issue meriting greater emphasis, now and even more in the future.

2. Clarifying the diagnostics picture

Diagnostics is a messy playground. Here we try to clarify the picture by posing questions: what, where, when, who and how. But before posing the questions, we define the term 'diagnostics' and also take a look at the relationship between diagnostics and the failure reporting, analysis and corrective action system (FRACAS). Finally, we summarise the relevant questions as a checklist to help create a diagnostics strategy for a modern work machine.

2.1 Definition of the term diagnostics

Diagnostics is not an easy word. The definition and sense of the word may vary depending on the context and the people using the word. IEC 60050-191 (Electrotechnical vocabulary. Dependability and quality of service) defines the phrase 'fault diagnosis' as follows: "*Actions taken for fault recognition, fault localisation and cause identification*". In this context, we use a more narrow definition for the word 'diagnostics':

Diagnostics = Actions taken for fault recognition and fault localisation

The reason to omit 'cause identification' from the above definition is the fact that in the case of machine diagnostics, the cause identification is considered to be included in the failure analysis process performed in the failure analysis laboratory. Hence, cause identification is part of the failure reporting, analysis and corrective action system (FRACAS). To make the relation between diagnostics and FRACAS clear, we will discuss FRACAS in more detail in Chapter 2.3.

2.2 Costs and benefits of diagnostics

The obvious reason why condition monitoring and diagnostic systems are developed is to:

Automate human diagnostic knowledge

A human is very good at diagnosing whether things are going well or wrong. By automating human reasoning we try to smoothen out the good days and the bad days, and work for 24 hours per day and seven days per week, but can hardly ever be as good as the best human. But, building up a reasonable set of diagnostics facilities involves a lot of human effort and often very dedicated, and thus expensive, expertise. It may be difficult to get motivated to spend huge sums of money on extensive diagnostics tasks, especially as the development project leaders know that diagnostics has little impact on the direct operative functions of their machines. But if we think of machine operations in a broader sense and in the longer term, the effect can be extremely valuable. Decisions that have effect after a long period of time are not easy to make and are certainly not straightforward. Questions that arise at the initial development stage are:

- When, where and how much condition monitoring capability is needed?
- What is the current situation? What data is collected now and is it sufficient?
- Who uses the data? How can we help him?

Some analysis and calculations can be used to get some guidance for decision making. Cost and benefit analysis (CBA) might be one tool with which to answer these questions and to determine the value of advanced monitoring systems in the long run. CBA considers costs from the development, investment and maintenance perspectives. Benefits cover both monetary and non-monetary issues. Monetary benefits include increased sales, extended lifetime, and reduced costs in service and operation. Non-monetary benefits include things like the readiness and ability of the machine to perform the task it has been planned for, better machine image and so on.

Decision making is not difficult: if the return of investment (ROI) and non-monetary benefits are high, there should be no doubt about making the investment decision. A big question mark is that the diagnostics technology is continuously evolving; this may affect the timetables.

The US Army has been investigating the extensive use of diagnostics and prognostics on weapons platforms, vehicles, etc. They have used systematic guidelines to evaluate and make cost and benefit analyses to set a target level for the availability performance [CEAC 1995]. The ability to increase real-time situation awareness and rapid reaction capabilities has wide effects, e.g. in logistics, proactive logistics, development of supporting infrastructure, and an implementation strategy that achieves maximum benefit with the resources available. The next table provides an example of how the US Army has evaluated its actions in various CBA outcomes.

Table 1. Action Matrix for CBA Results [Greitzer et al. 2001].

Non-Monetary Benefits

Return on Investment

	HIGH	MED	LOW
HIGH	Implement as soon as possible	High priority	"Harvest" to reduce budget impacts
MED	High priority	Medium priority	Low priority
LOW	Medium priority	Low priority	Abandon

2.3 Diagnostics and FRACAS

There is a clear distinction between diagnostics and FRACAS: when the service person has found the faulty component, i.e. he has completed the diagnostics process, he writes a failure report, i.e. he starts the FRACAS procedure. This is a practical distinction as the service person does not care why a sensor or other component has failed; he is happy when the machine works with the replacement component. The reliability engineers, instead, are interested in finding the root cause for the problem to incorporate corrective actions to the design, manufacturing and component selection processes if necessary. And enhanced diagnostics may be one of the items on the list of necessary corrective actions. Hence, diagnostics provides input to the FRACAS process and FRACAS sets requirements for the diagnostics (see Figure 1).

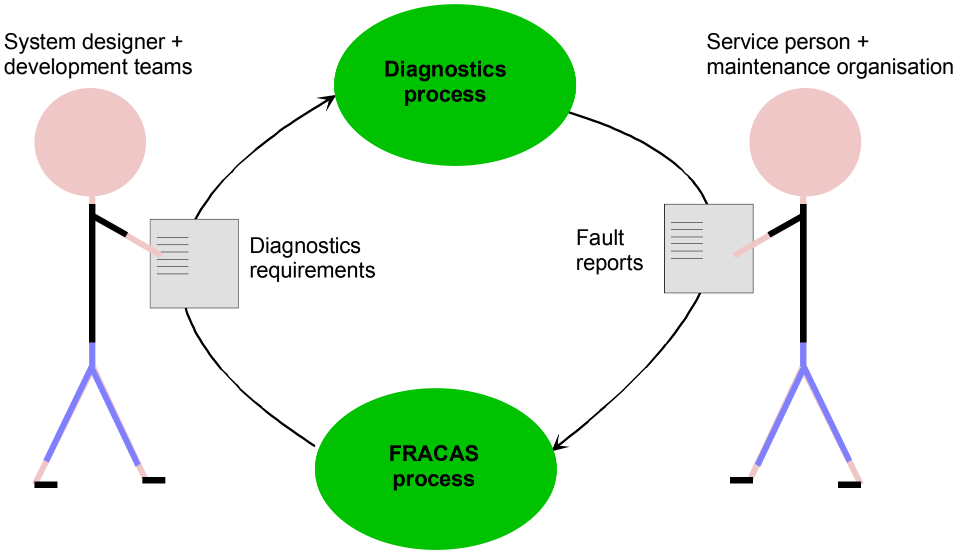


Figure 1. Connection between diagnostics and FRACAS.

But which one comes first, the chicken or the egg? FRACAS clearly sets the pace here, as we may manage well without any diagnostics whatsoever if the FRACAS process reports adequate availability performance without any corrective actions. But as soon as the FRACAS process starts to indicate too high failure rates or too long maintenance times, it is time to start to plan diagnostic features. As a consequence, we state the following:

A well functioning failure reporting, analysis and corrective action system (FRACAS) is a cornerstone to set up a reasonable set of diagnostics requirements

It should be noted that FRACAS is only a part of the dependability programme of the company (machine manufacturer) and diagnostics is only a part of the maintenance programme of the company. The dependability programme, and hence also the FRACAS process, is owned by the dependability engineer, but who owns the diagnostics process? Currently there is normally no evident owner of the diagnostics process, but the system designer designs the diagnostics features while designing the actual application. However, as the requirements for more elaborate diagnostics in work machines increase, it may be necessary to appoint a diagnostics engineer, or the dependability engineer could also take over the diagnostics process.

Note also that FRACAS is not the only source of requirements for diagnostics. There may also be legislative requirements as well as requirements due to safety issues. For a fresh design with no FRACAS based history information, the initial diagnostic requirements are set by former experience of similar systems.

2.4 What is diagnosed

What is the object to be diagnosed? As the on-board diagnostics is mainly embedded into the electronic control system, we discuss this issue from that point of view.

The electronic control system consists of sensors, controllers, actuators and communication links. The controllers are normally programmable. The electronic control system controls the mechanical system including hydraulic, pneumatic and electro-mechanical sub-systems to make the machine do the desired work. We can identify three diagnosis levels:

1. Diagnosis of the control system (sensors, connectors, cables, CPU, RAM, I/O electronics, actuators and communication sub-systems)
2. Diagnosis of the mechanical system (fluid quality, fluid levels, fluid pressure, temperatures, bearing wear-out, etc.)
3. Monitoring of the efficiency of the work and quality of the output produced (like sizes of crushed stones, volumes of timber or manoeuvring accuracy).

2.5 Where do the diagnostics procedures reside

Diagnostic procedures may reside either on-board the machine or off-board. Compared to cars, work machines are better equipped with computing power. Normally there is also a display module included in the control system. This helps a lot to implement sophisticated diagnostic features with informative fault reports, even electronic fault localisation and repair manuals. Very often a work machine is equipped with a full blown PC and with a professional operating system like Windows, Linux or QNX. This enables using commercial off-the-shelf software tools to provide flexible data analysis facilities. Hence, there is practically nothing that could not be implemented on-board, at least from the technological point of view. However, licensing of the commercial software makes it impossible to put all diagnostics software onto all machines. Nevertheless, in the case of work machines, the line between on-board and traditional off-board diagnostics is vanishing. With traditional off-board diagnostics we mean handheld testers or laptops carried by the service technician. On the other hand, the role of remote diagnostics, telediagnosis, is growing. Hence, we define three location dependent levels of diagnostics services:

1. On-board
2. Near off-board
3. Far off-board (remote diagnostics, telediagnosis).

2.6 When is the diagnosis performed?

The diagnosis can be performed either on-duty or off-duty. Continuous sensor signal monitoring is an example of on-duty diagnostics. Such diagnostics can be part of the application software, or the system may include stand-alone diagnostic agents to monitor the condition of the system. Off-duty diagnostics typically include special tests that require special circumstances and a lot of processing power. Hence the work is stopped for the duration of the diagnostics and the machine will enter a diagnostics mode.

Together with the location-based levels presented in Chapter 2.5, we can build a where-when matrix to facilitate the allocation of diagnostics features. Table 2 supplies some examples of diagnostics features allocated to such a matrix. Note that the 'near off-board' level is omitted as, within the context of work machines, near off-board diagnostics (if such is implemented) is a mixture of on-board and far off-board diagnostics services.

Table 2. Examples of diagnostics features allocated to a where-when matrix.

	On-board	Far off-board
On-duty	• Sensor signal range checking • Monitoring of the diagnostics outputs of smart FETs • Communication error checking • Logging of fault codes • Sub-system diagnostics agents	• Remote monitoring and logging of system state variables
Off-duty	• Power-on self-test of the controller • Sensor signal tests by forcing a well-defined action • Forced mode actuator tests • Browsing the fault log • Browsing the electronic manual	• Gathering large sets of data and producing long-term trends and analysing the long-term behaviour

Power-on self-test is an exceptional case of off-duty diagnostics. Although the control system is not in a diagnostics mode during the power-on self-test, the system is virtually in a diagnostics mode as the machine will not start its on-duty operation if the self-test fails.

2.7 Who performs the diagnosis?

There are basically three levels of persons who can perform the diagnosis:

1. Customer level (the operator or the owner of the machine)
2. Service technician level
3. System expert level.

The comprehensiveness of the diagnostics tools and the skill level are lowest at the customer level and highest at the system expert level. The reason is simple: it is expensive to provide all customers with a highly extensive set of diagnostics tools and it is also expensive to train all the customers to use sophisticated diagnostics tools.

2.8 How to partition diagnostics into more manageable chunks

The OSA-CBM organisation (http://www.osacbm.org [Referenced 27.01.2004]) has defined an open system architecture (OSA-CBM) for condition-based maintenance. The OSA-CBM architecture defines a seven-layer architecture (similar to the Open System Interconnection [OSI] model of communication sub-systems) to partition the diagnostics sub-system into modular and well interfacing entities. The seven layers are the following (in bottom-up order) [Lebold & Thurston 2001]:

1. Data acquisition (sensor module that outputs calibrated sensor signal values)
2. Data manipulation (signal processing; e.g. mean value calculation or frequency spectra)
3. Condition monitoring (e.g. range checking, alerts)
4. Health assessment (diagnostic processing; e.g. fault condition evaluation)
5. Prognostics (e.g. estimation of remaining useful life)
6. Decision support (e.g. 'limp home' instructions and automatic reconfiguration)
7. Presentation (user interface).

If we think about on-board diagnostics on a work machine, the simplest diagnostics activity is to implement the data acquisition layer to provide data for remote off-board diagnostics. If the amount of data to be transferred is large, it may be better to include the data manipulation layer to provide only some characteristic parameters of the measured signals. The more sophisticated on-board diagnostics and problem solving we want, the more layers we have to implement. The CPU, storage and communications capacities set the restraints and criteria for the level of sophistication and for the on-board/off-board partitioning.

OSA-CBM is presented more closely in Chapter 4.2.

2.9 Application-specific or generic

Some of the diagnostics facilities are reusable. For example, diagnostic protocols and error logs can be embedded into the operating system of the electronic control modules to constitute diagnostics-aware extended operating system services. Figure 2 depicts how such extended operating system services relate to the core operating system, to the application and to the device drivers. The example is from the SAE Generic Open Architecture (GOA) model.

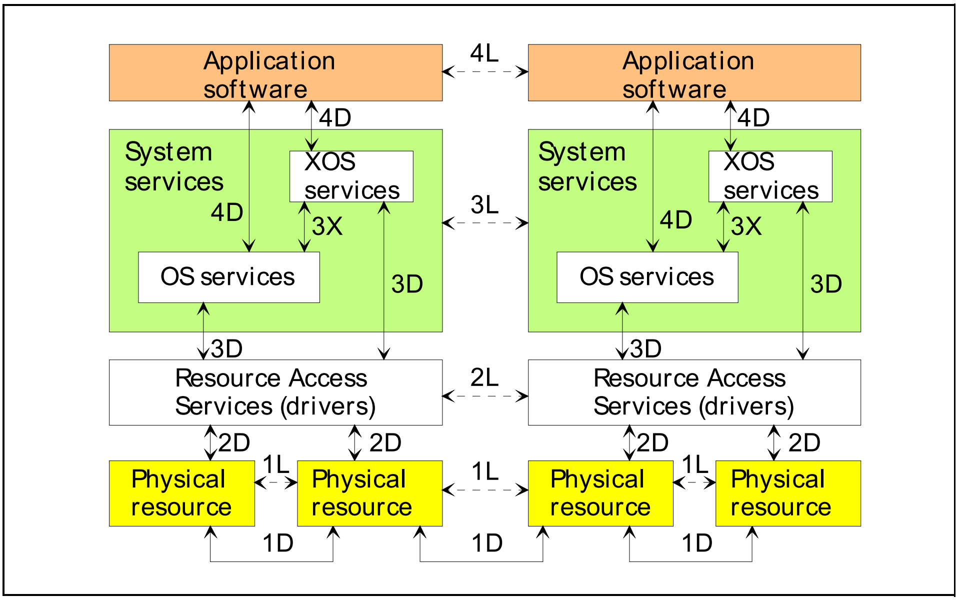


Figure 2. GOA interface reference model (SAE AS 4893. 1996) (OS = Operating System; XOS = eXtended Operating System).

4L: Application Logical Peer Interface
4D: Application to System Services Direct Interface
3L: System Services Logical Peer Interface
3D: System Services Software to Resource Access Services Direct Interface
3X: OS Services to XOS Services Direct Interface
2L: Resource Access Services Logical Peer Interface
2D: Resource Access Services to Physical Resources Direct Interface
1L: Physical Resources Logical Peer Interface
1D: Physical Resources to Physical Resource Direct Interface

Furthermore, some components or sub-systems can be considered reusable objects that also include relevant diagnostics. A potentiometer sensor could be an example of such an object: when a new potentiometer-based measurement is to be incorporated into the system, its devices drivers and diagnostics routines could be picked from a software components library. If we compare this type of object-based approach with the layered architecture of OSA-CBM presented in Chapter 2.8, we can recognise a vertical architecture instead of the horizontal architecture of OSA-CBM (see Figure 3).

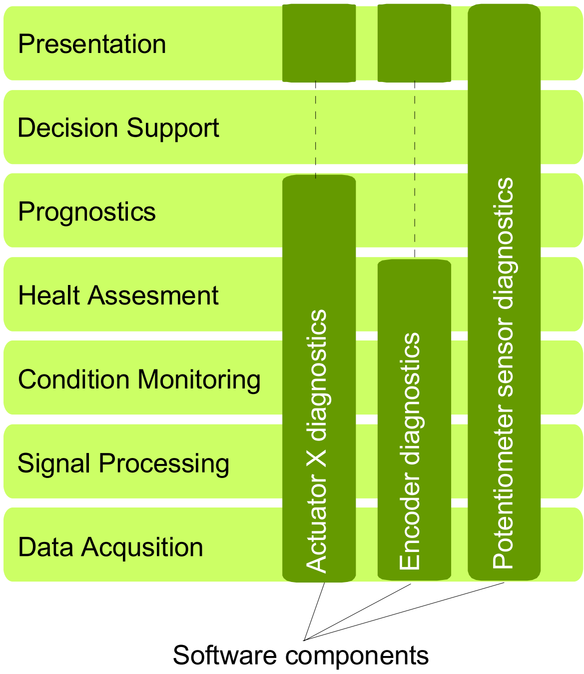


Figure 3. Horizontal OSA-CBM architecture vs. object or component-based vertical architecture.

The goal is to be able to make generic diagnostics software components, but there is always an application-specific realm for diagnostics. In the above potentiometer sensor example, the generic diagnostics can detect an electrical fault but not mechanical faults, e.g. if the shaft of the potentiometer slips, the electrical signals do not directly reveal the failure. Hence the mechanical failure cannot be detected by the generic diagnostics software, so application-specific plausibility checking must be applied. Figure 4 illustrates the partitioning between application-specific and generic diagnostics.

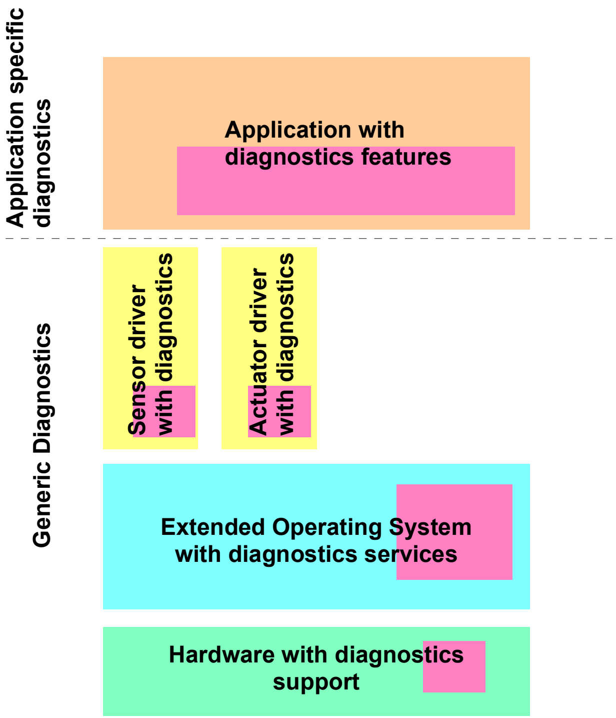


Figure 4. Partition of application-specific and generic diagnostics.

2.10 Diagnostics strategy

The diagnostics strategy is manufacturer and machine type-specific. In this chapter we will supply a list of questions that should be answered by the diagnostics strategy created for the particular work machine type. The list is presented below:

1. What is the current situation? What data is collected now and is it sufficient?
2. Who uses the data? How can we help him?
3. Who or what sets the request to implement a diagnostic feature? (FRACAS perhaps?)
4. Who (and by what criteria) decides whether the requested diagnostic feature is implemented? (Is a diagnostics engineer needed?)

5. How much effort is put into software-based intelligence and how much into support for human-based diagnostics (with, e.g., training and multimedia manuals)?
6. What are the (on-board) communications, CPU and storage resources available for diagnostics?
7. What part of the diagnostics services is embedded on-board the machine and what is left off-board?
8. Will all the off-board diagnostics be remote diagnostics or is near off-board diagnostics needed also or only?
9. What parameters, components and sub-systems are diagnosed on-duty during operation and which off-duty in a diagnostics mode?
10. Which diagnostics services should be accessible to the customer, which to the service technicians and which to only the system experts?
11. What is the level of on-board diagnostics to be implemented? (data acquisition and condition monitoring only or also health assessment and perhaps prognostics?)
12. What parts of the diagnostics hardware, software and algorithms could be generalised for use in future projects?

3. Automotive and heavy-duty vehicles diagnostics standards

Have you heard stories about modern cars left beside the road with awkward problems that have been very difficult to resolve? The automotive control, safety, information and entertainment systems are becoming too complex to be managed perfectly. The complexity makes the maintenance of cars a headache. There is practically no single person who can master all the control system functions of a car. One of the reasons for this is the fact that the control system is a composition of sub-systems from different vendors and thus from different experts. Hence the control system must be equipped with a reasonable set of diagnostics services to facilitate pertinent maintenance by non-experts and to keep the availability performance of modern cars at least at the same level as traditional cars.

Off-road work machine manufacturers could well learn from the automotive manufacturers and exploit the diagnostics tools available on the market for cars and trucks, and apply the available diagnostics standards. Some of the diagnostics features are required by the authorities, especially those concerning exhaust gas emissions. In the following we will take a look at the automotive and heavy-duty diagnostics standards.

3.1 ISO and SAE diagnostics standards

The standardisation of diagnostics for the automotive and utility vehicle sector is driven by the emissions legislation. Heavy-duty vehicles are soon to be equipped with an on-board diagnostics interface for inspection purposes similar to that in cars. However, such legislation for off-road work machines is still far ahead in the future. Nevertheless, such legislation will surely come. As a consequence, it is good to take a look at the diagnostics standards applied in the automotive and utility vehicle sector in order to make the next-generation diagnostics architectures of the work machines flexibly adaptable for the upcoming standards.

The main issues of the diagnostics standards are as follows:

- Diagnostic connector and other physical layer issues
- Communication layers (data link layer and transport layer)
- Diagnostics services, including Freeze Frame service, to support capturing of circumstances at the occurrence of a trouble code.
- Diagnostic Trouble Codes (DTCs)
- Security
- Off-board scan tool specification.

The diagnostics standards may also support non-emission related, i.e. general, diagnostic services and program downloading, as well as calibration and configuration services.

ISO 9141 [1989] is the most famous diagnostics standard and defines the physical layer of a diagnostics bus or a point-to-point connection, but it also defines issues concerning the data link layer. It does not define any diagnostics services and is thus basically generic and not emissions related. It covers both 12 V and 24 V systems. **ISO 9141-2** [1994] is a supplementary standard to ISO 9141 to support **SAE J1978** [2002]-based SAE OBD II scan tools. ISO 9141-2 only covers 12 V systems. However, both ISO 9141 and ISO 9141-2 are obsolete and uninteresting within the context of the future trends of work machine diagnostics. In Europe, the **ISO 14230** set of standards was adopted to amend the ISO 9141 standard. ISO 14230 consists of four parts:

- **ISO 14230-1** Road vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 1: Physical layer
- **ISO 14230-2** Road vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 2: Data link layer
- **ISO 14230-3** Road vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 3: Application layer
- **ISO 14230-4** Road vehicles - Diagnostic systems - Keyword Protocol 2000 - Part 4: Requirements for emission-related systems

ISO 14230-1 is based on ISO 9141-2, but also includes 24 V systems. ISO 14230-2 defines the data link layer and ISO 14230-3 defines the Keyword Protocol 2000 implementation of the unified diagnostic services defined in **ISO**

14229 [1998][1]. The ISO 14230 set of standards is generic and not application-related, except part 4, which defines requirements for emission-related systems. Hence the standards could be used within the context of work machines as well. However, Keyword Protocol 2000 is still a so-called K-line protocol, as was ISO 9141, and does not fit the contemporary control systems of work machines, which normally include a Controller Area Network (CAN)-based communication bus. Fortunately, modern cars include the CAN bus as well. As a consequence, a new set of ISO standards is being published: the **ISO 15765** set of standards. The set includes the following parts:

- **ISO 15765-1** Road vehicles - Diagnostics on Controller Area Networks (CAN) - Part 1: General information
- **ISO 15765-2** Road vehicles - Diagnostics on Controller Area Networks (CAN) - Part 2: Network layer services
- **ISO 15765-3** Road vehicles - Diagnostics on Controller Area Networks (CAN) - Part 3: Implementation of unified diagnostic services (UDS on CAN)
- **ISO 15765-4** Road vehicles - Diagnostics on Controller Area Networks (CAN) - Part 4: Requirements for emissions-related systems

The ISO 15765 set of standards is similar to Keyword Protocol 2000 (i.e. ISO 14230), except that no special physical layer or data link layer is specified as the CAN protocol is used. Instead, a network layer is specified to support data transfers larger than a single CAN frame can carry (i.e. eight bytes). The diagnostics services are practically the same as in ISO 14230-3 and, in fact, ISO 15765-3 refers to ISO 14230-3 and does not repeat the service definitions but does provide some guidelines on the use of ISO 14230-3 services in cases of CAN. Hence the potential user of the ISO 15765 standards has to acquire a copy of the ISO 14230-3 standard as well.

[1] ISO 14229 is under revision and will soon be published as ISO 14229 Part 1: Road vehicles -- Unified diagnostic services (UDS) -- Part 1: Specification and requirements.

In the USA, the Society of Automotive Engineers (SAE) has issued a set of regulatory diagnostics standards for cars, light-duty trucks and medium-duty vehicles[2]. They are as follows [Stepper et al. 1995]:

- **SAE J1930** Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations, and Acronyms - Equivalent to ISO/TR 15031-2
- **SAE J1962** Diagnostic Connector - Equivalent to ISO 15031-3
- **SAE J1978** OBD II Scan Tool - Equivalent to ISO 15031-4
- **SAE J1979** E/E Diagnostic Test Modes - Equivalent to ISO 15031-5
- **SAE J2012** Diagnostic Trouble Code Definitions - Equivalent to ISO 15031-6
- **SAE J2186** E/E Data Link Security[3]
- **SAE J2008** Recommended Organization of Vehicle Service Information for Interchange

The communication layer for these can be either ISO 9141-2 or SAE J1850 [2001]. J1939 is an alternative as a communications layer in the case of medium and heavy-duty vehicles [Stepper et al. 1995][Anon 1999]. J1939 also includes a diagnostics layer standard (J1939-73) and a diagnostics connector specification (J1939-13). J1939-73 defines diagnostic services as well as the structure (and, partly, the contents) of the Diagnostic Trouble Codes (DTCs). These are different from J1979 services and J2012 DTCs. J1939 will be presented in Chapter 3.1.1 in more detail.

In Europe, Directive 2005/78/EC stipulates that either ISO 15765 or SAE J1939 must be used in heavy-duty vehicles.

The Worldwide Harmonisation group (WWH) is working together with ISO TC22/SC3/WG1 to produce a globally harmonised set of diagnostics standards to support light, medium and heavy-duty vehicles. The WWH group proposes use of TCP/IP over Ethernet protocol (see http://www.unece.org/trans/doc/2006/wp29grpe/ECE-TRANS-WP29-GRPE-2006-08r1e.pdf [Referenced 10.04.2006]). The particular diagnostics

---

[2] The environmental protection agency (EPA) proposes these for heavy-duty vehicles starting from the year 2004 [Anon. 1999]. The EPA also allows use of J1939 where applicable.

[3] This is quite probably equivalent to ISO 15031-7:2001, although the title of the SAE standard does not imply it.

standard will later be issued as an ISO standard (ISO 27415). However, ISO 15765 SAE J1939 will be used during the transition period.

Yet another example of a diagnostics standard can be found in the Local Interconnect Network (LIN) specification (starting from its version 2.0). LIN is a low-cost multiplex bus to be used mostly as a sub-net to a CAN bus. Hence the LIN 2.0 specification adopts the CAN based on the ISO 15765-2 and -3 diagnostics standards. The frame structure of LIN, with a maximum eight data bytes, is similar enough to the CAN frame structure to make practically direct adoption of the CAN-based standards possible.

There is also a set of standards for the measurement, calibration and diagnostics of automotive application created by an organisation called Association for Standardisation of Automation and Measuring Systems (ASAM e.V.). The particular set for calibration, measurement and diagnostics is called ASAM-MCD and includes the CAN Calibration Protocol (CCP), which has been succeeded by the more general XCP that can use any relevant communication protocol including CAN. XCP defines data acquisition and stimulation services, calibration services and Flash programming services. The ASAM standards are mainly aimed at the development phase of the automotive systems, such as measurements on a test rig, and not for providing standardised on-board diagnostics for the field service. Hence XCP may not be interesting within the context of work machines. Interesting, however, may be the ASAM-MCD2 version 2.0 that will include an enhanced Open Diagnostics Exchange (ODX) format specification to support the definition of the diagnostics services of electronic control units (ECU). An ODX database is used to generate a diagnostics software template for the ECUs and to port the ECU diagnostic capability descriptions to service technician tools in field service. ODX supports the whole process cycle, including development, production and service. Tools supporting ODX are emerging onto the market.

The work machine on-board diagnostics legislation will most probably follow the tracks of heavy-duty legislation or the anticipated globally harmonised set of diagnostics standards discussed above. SAE J1939/73 and ISO 15765 are relevant in both cases, but the upcoming ISO 27415 must be considered as well. We will take a closer look at these two standards in the following two chapters (Chapters 3.1.1 and 3.1.2 respectively). ODX may also be interesting for work machine manufacturers to facilitate the management and porting of diagnostics information. Hence ODX is also presented more closely in Chapter 3.2.

Two additional J1939-based diagnostic standards are also emerging, one for tractors, (ISO 11783-12) and one for the communication between towing and towed vehicles (ISO 11992-4). These standards are not presented here.

3.1.1 J1939/73

J1939/73 belongs to the family of J1939 standards (Recommended Practice for Truck and Bus Control and Communications Network). J1939 defines CAN-based in-vehicle real-time communications in buses and trucks, including off-road vehicles. The particular standard, J1939/73 (Application Layer-Diagnostics), defines diagnostic messages and protocols over the J1939 bus. J1939/73 specifies the following issues:

- **Diagnostic connector**. In fact, J1939/73 refers to J1939/13, which specifies a special 9-pin circular connector.
- A set of **diagnostic messages** to provide the diagnostic services.
- **Direct Memory Access with security control.** This service is provided to read and manipulate the memory of a module in the network. This service can be used also for boot-loading and calibration.

The diagnostic messages defined by J1939/73 are as described in Table 3.

Table 3. J1939/73 diagnostic messages (DM).

DM 1	Active diagnostic trouble codes	Carries the statuses of warning lamps and trouble codes of active faults. This message is sent when a fault becomes active and when it becomes inactive. Furthermore, DM1 is sent once per second while the fault is active. DM1 always includes all the active trouble codes. If no active faults are prevailing, no DM1 is sent, except as a response to a request. The trouble code consists of a Suspect Parameter Number (SPN, 19 bits), Failure Mode Identifier (FMI, 5 bits), Occurrence Count (OC, 7 bits) and SPN Conversion Method (CM, 1 bit). CM is needed to allow legacy SPNs (with different encoding styles) to be used. Together, these make 32 bits - i.e. four bytes. A DM1 with more than one diagnostic trouble code is sent using a multipacket transport protocol.
DM 2	Previously active diagnostic trouble codes	Carries the history of active trouble codes. Sent as a response to a request.
DM 3	Diagnostic data clear/Reset for previously active DTCs	Erases the history of active trouble codes but does not affect currently active trouble codes.
DM 4	Freeze frame parameters	Contains values of predefined parameters (and possible manufacturer-specific parameters) at the time of trouble code occurrence. The predefined parameters are: Engine torque mode, Boost, Engine speed, Engine Load, Engine coolant temperature and Vehicle speed. DM4 is sent as a response to a request.
DM 5	Diagnostic readiness	Provides information on the number of active trouble codes and previously active trouble codes, and the OBD compliance and other diagnostics readiness. DM5 is sent as a response to a request.
DM 6	Continuously monitored systems test results	Provides diagnostics information about the emission-related components for a service technician after clearing the trouble codes and after driving the first test drive.
DM 7	Command non-continuously monitored systems	Allows commanding of manufacturer-specific on-board tests by referring to a test identifier (1–64).

DM 8	Test results for non-continuously monitored systems	Returns test results for the tests commanded by DM7.
DM 9	Oxygen sensor results	Not defined in the current version of J1939/73.
DM 10	Non-continuously monitored systems test identifier support	Provides a list of manufacturer-specific on-board tests (1–64) supported.
DM 11	Diagnostic data clear/reset for active DTCs	Erases all active trouble code information.
DM 12	Emission-related active DTCs	Same as active trouble codes, except that only emission-related active trouble codes are reported. Sent as a response to a request.
DM 13	Stop/start broadcast	Can be used to stop (and restart) transmission of broadcast messages and periodic requests. Vital broadcasts are allowed in a stop state. Is used to minimise the traffic during diagnostic procedures, e.g. during calibration or I/O-module traffic emulation.
DM 14	Memory access request	Is used to initiate direct memory access by a diagnostics tool. Includes security control.
DM 15	Memory access response	Sent as a response to DM14.
DM 16	Binary data transfer	Is used to do the actual data transfer to perform the requested memory access.
DM 17	Boot load data	Is used to download boot code into a device.
DM 18	Data security	Is used to provide a security facility during memory access.
DM 19	Calibration information	Provides information about the calibration to a diagnostics tool. The information includes a calibration checksum over the non-volatile calibration information and an ASCII identifier of the calibration entity.

In many cases the message length of the diagnostics messages described above exceeds the CAN frame length, i.e. eight bytes. In those cases the J1939-21[4] transport protocol is used to transfer the multipacket messages.

J1937/73 pre-defines the failure mode identifiers (FMI) reported within the diagnostic trouble codes (DTCs). The FMIs are defined as listed in Table 4 and the signal ranges referred to in the table are depicted in Figure 5.

Table 4. J1939/73 Failure Mode Identifiers.

FMI=0	Data valid but above normal operational range - most severe level; *range e* in Figure 5
FMI=1	Data valid but below normal operational range - most severe level; *range d* in Figure 5
FMI=2	Data erratic, intermittent or incorrect
FMI=3	Voltage above normal, or shorted to high source; *range g* in Figure 5
FMI=4	Voltage below normal, or shorted to low source; *range f* in Figure 5
FMI=5	Current below normal or open circuit; *range f* in Figure 5
FMI=6	Current above normal or grounded circuit; *range* g in Figure 5
FMI=7	Mechanical system not responding or out of adjustment
FMI=8	Abnormal frequency or pulse width or period
FMI=9	Abnormal update rate
FMI=10	Abnormal rate of change
FMI=11	Root cause not known
FMI=12	Bad intelligent device or component
FMI=13	Out of calibration
FMI=14	Special instructions
FMI=15	Data valid but above normal operating range - least severe level; *range i* in Figure 5
FMI=16	Data valid but above normal operating range - moderately severe level; *range k* in Figure 5

[4] SAE J1939-21 Data Link Layer

FMI=17	Data valid but below normal operating range - least severe level; *range h* in Figure 5
FMI=18	Data valid but below normal operating range - moderately severe level; *range j* in Figure 5
FMI=19	Received network data in error
FMI=20-30	Reserved for SAE assignment
FMI=31	Not available or condition identified by the SPN exists

Figure 5. J1939/73 FMI ranges.

J1939/73 does not pre-define Suspect Parameter Numbers (SPNs). However, it suggests mapping J1587[5] Parameter Identifiers (PIDs) one to one as SPNs. For example, the engine oil pressure parameter has a PID number 100 in J1587 and should therefore be mapped to SPN 100 in a J1939 system. An SPN greater than 511 shall be used where no PID for the J1939 exists.

3.1.2 ISO 15765-3

ISO 15765-3[6] is a CAN-based diagnostic services standard. In fact, it defines a CAN-based version of Keyword Protocol 2000 services (ISO 14230-3[7]). The diagnostic services defined in ISO 15765-3 are defined in Table 5.

---

[5] SAE J 1587 Joint SAE/TMC Electronic Data Interchange Between microcomputer systems in Heavy-Duty Vehicle Applications.

[6] ISO 15765-3 Road vehicles -- Diagnostics on Controller Area Networks (CAN) -- Part 3: Implementation of unified diagnostic services (UDS on CAN).

[7] Road vehicles -- Diagnostic systems -- Keyword Protocol 2000 -- Part 3: Application layer.

Table 5. ISO 15765-3 diagnostic services.

SERVICE NAME	DESCRIPTION
NetworkConfiguration	The client reads information about the system from a server.
DisableNormalMessageTransmission	The client requests to stop non-diagnostic message transmission.
EnabteNormalMessageTransmission	The client requests to resume non-diagnostic message transmission.
ControlDTCSetting	The client starts and stops setting of DTCs in the server.
startDiagnosticSession	The client requests to start a diagnostic session with a server(s).
securityAccess	The client requests to unlock a secured server.
ecuReset	The client forces the server(s) to perform a reset.
readEcuIdentification	The client requests identification data from the server(s).
ReadDataByLocalIdentifier	The client requests the transmission of the current value of a record with access by a local identifier.
ReadDataByCommonIdentfier	The client requests the transmission of the current value of a record with access by a common identifier.
ReadMemoryByAddress	The client requests the transmission of a memory area.
DynamicallyDefineLocalIdentifier	The client requests to dynamically define local identifiers that may subsequently be accessed by a local identifier.
WriteDataByLocalIdentifier	The client requests to write a record accessed by a local identifier.
WriteDataByCommonIdentifier	The client requests to write a record accessed by a common identifier.
WriteMemoryByAddress	The client requests to overwrite a memory area.
SetDataRates	The client changes the data rates for periodic transmissions.

ReadDiagnosticTroubleCodes	The client requests from the server the transmission of both the number of the DTC and values of the diagnostic trouble codes.
ReadDiagnosticTroubleCodesBy-Status	The client requests from the server the transmission of both the number of the DTC and values of the diagnostic trouble codes, depending on their status.
ReadStatusOfDiagnosticTroubleCodes	The client requests from the server the transmission of the number of the DTC, values and status of the diagnostic trouble codes.
ReadFreezeFrameData	The client requests from the server the transmission of the value of a record stored in a freeze frame.
ClearDiagnosticInformation	The client requests the server to clear all or a group of the diagnostic information stored.
InputOutputControlByLocalIdentifier	The client requests the control of an input/output specific to the server
InputOutputControlByCommonIdentifier	The client requests the control of a common input/output.
StartRoutineByLocalIdentifier	The client requests to start a routine in the ECU of the server using the local identifier of the routine.
StartRoutineByAddress	The client requests to start a routine in the ECU of the server using the address of the routine.
StopRoutineByLocalIdentifier	The client requests to stop a routine in the ECU of the server using the local identifier of the routine.
StopRoutineByAddress	The client requests to stop a routine in the ECU of the server using the address of the routine.
RequestRoutineResultsByLocalIdentifer	The client requests the results of a routine by the local identifier of the routine.
RequestRoutineResultsByAddress	The client requests the results of a routine by the address of the routine.
RequestDownload	The client requests the negotiation of a data transfer from the client to the server.

RequestUpload	The client requests the negotiation of a data transfer from the server to the client.
TransferData	The client transmits data to the server (download) or requests data from the server (upload).
RequestTransferExit	The client requests the termination of a data transfer.

ISO 15765-3 makes references to ISO 14230-3 and specifies only what is different from or added to ISO 14230-3. Therefore, one has to acquire both standards to implement the diagnostic services in the software. Furthermore, ISO 15763-2[8] is needed; it includes the transport protocol specification (multipacket transfer) and the use of the CAN identifier field and the data bytes to address the network nodes locally or over gateways.

ISO 15763-3 does not define failure mode identifiers or suspect parameter numbers, but is a pure service specification.

ISO 15763-3 provides a well structured and easy to comprehend set of diagnostic services. It is suggested here that if, within the context of work machines, there is no compelling reason to use SAE J1939/73, ISO 15765-3 is used instead.

3.2 Open Diagnostic data eXchange (ODX)

ODX is an effort by a group of automotive manufacturers to standardise the diagnostic services specification format. The group is called Association for Standardisation of Automation and Measuring systems (ASAM). The ODX standard is maintained by ASAM e.V. and is numbered ASAM MCD-2D.

The goal behind ODX is to provide a single source of diagnostic information for the different parties involved in the car diagnostics. These parties include the software development team that designs and implements the diagnostics

8 Road vehicles -- Diagnostics on Controller Area Networks (CAN) -- Part 2: Network layer services

software, manufacturing staff and service personnel (see Figure 6). ODX defines the diagnostics services of a car type in an XML file. A diagnostic or testing tool from any vendor can read this XML file to configure itself to be able to perform diagnostic tests embedded into the car control system. Furthermore, the XML file can be fed as an input to a code generation tool to produce the framework for the diagnostic software of the electronic control units, thus speeding up the diagnostic software production process.

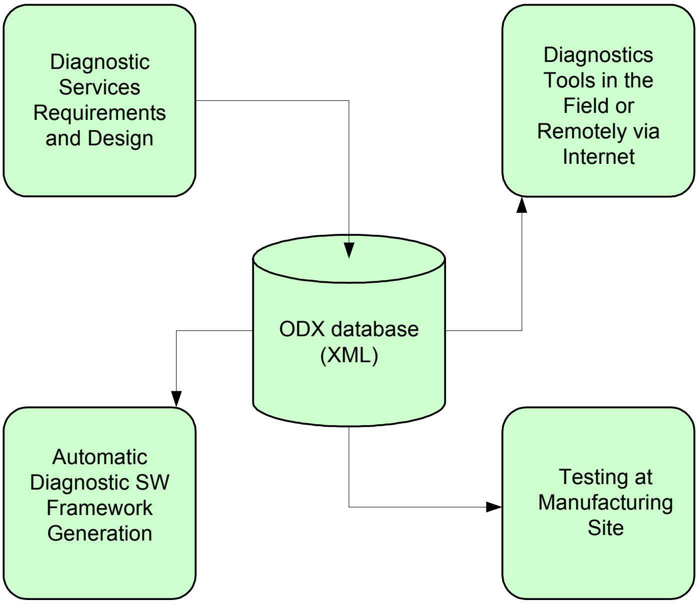


Figure 6. ODX context.

ODX is independent of the diagnostic protocol used. Hence it can be used to describe J1939/73, ISO 15763-3, etc., diagnostic services. Tools that support the ODX format are already available from different vendors. However, a quick Internet search does not find too many such vendors. Hence broad acceptance of ODX cannot be verified at the time of writing this report.

ODX is also being standardised by the International Organisation for Standardisation (ISO); the standard will be numbered ISO 22901 (with several parts). Meanwhile, the standard is available from ASAM e.V. at http://www.asam.net/ (Referenced 09.02.2005).

4. Diagnostics architectures

This chapter is made with the point in mind that the main focus of this report is to take a glimpse at the diagnostics of machinery that is not stationary. This gives us some kind of boundaries that can be followed. At the same time, we must get a good full picture of what these diagnostics will mean in more years to come.

The current situation of machine manufacturers is that more and more information technology is being added into machines [Lind et al. 1999, Jameel et al. 1998, Rohmert 2002]. The main reason is that every maintenance organization wants to take good care of the machinery they have. Every breakdown means interrupted operation and lost profit. In some instances the breakdown can be very costly. Usually, there is no spare machine to substitute the original machine.

Every machinery manufacturer has some estimates of component run time and other values that influence the run time cost for making the purchase. Usually, these values include some estimates of service speed. If service support could be done by better machine self-diagnostics or by remote service, it would mean a great reduction in the time spent wondering what is wrong and what to do. This usually reduces the changing of parts that are ok.

The constraints that come with machinery that is not static are basically limited to communication issues. The limitations on moving a large amount of data through wireless networks are still a bit problematic. Also, the computers on a work machine are getting more powerful and are thus capable of hosting software components for condition-based maintenance (CBM).

It also has to be noted that the purpose of a CBM system is basically to save money or to protect from accidents. A CBM monitoring system makes sense if the cost of constructing and maintaining the system is less than the cost due to system failure if the machine is left unattended.

It is recognized that condition-based maintenance and machine diagnostics became a focused research area in the United States at the end of the 1990s and early 2000. Big global companies like Boeing and Caterpillar with military partners like US Navy have focused on making machinery and monitoring systems more open so that development costs can be reduced and a better service can be provided. With other partners, their ideas are coming to come to the stage of standardization.

Machines that have very a limited capability of providing measurement information are hard to diagnose until some faults have come up with them. Usually, this is called the run-to-failure maintenance approach. The basic idea of condition-based maintenance is to monitor a machine and to react when some faults have been recognized. To take this a step further, we could make an estimate of when is the right time to take maintenance action before something is actually broken. This means that some estimate or prognosis has to be made about the remaining useful life of machine. At the same time, this helps to prepare the service action and to make the down time shorter. Usually, this is done by maintenance with periodic service and is based on experience of the lifetime of machine parts.

In order to build a sophisticated condition-based maintenance system, some things are essential: without sensing and data acquisition it would be impossible to build a good CBM system.

The reasons why the people that are involved in CBM are trying to standardise the systems and define an architecture and framework are the benefits that can be reached in the long term in solving the problem of integration of heterogeneous systems (see Figure 7). This could lead to plug-and-play systems, saving the burden of system integration and allowing more freedom to choose the best technology from the supplier.

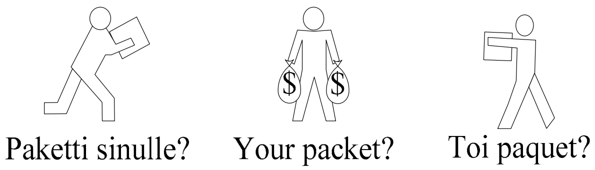


Figure 7. Do we understand each other?

Integrating systems from different soft-/hardware vendors is usually very difficult. There are four different options for solving this problem:

- Avoid bridge-building (purchase as many systems as possible from one vendor)
- By a custom bridge (buy a pre-designed gateway offered by a supplier)
- Build a custom bridge (build your own gateway through an integration company or internal IT group)
- Use an open systems bridge (build or buy an industry-standard gateway).

Every option has its pros and cons. The last option requires that the whole architecture and all interfaces have been accurately defined and the information is available.

Architectures that are used in condition-based maintenance have evolved into a stage of standardisation like ISO 13374. This chapter will describe some of the research in this area.

The common feature of the following is an idea to put the required software into the components and let them communicate with each other through a TCP/IP network.

4.1 MIMOSA

The Machinery Information Management Open Systems Alliance (MIMOSA) was set up to address problems that come up when setting up a machine condition monitoring and diagnostic system. The aim is to establish an open architecture and a set of protocols for exchanging complex sensor information between CBM systems. Currently, the direction of MIMOSA is to think of management in a larger perspective and to focus on asset management-related information standards. The next figure (Figure 8) presents a view on how various enterprise-level software systems can use MIMOSA. These are Human-Machine Interfaces (HMI), Manufacturing Execution Systems (MES), Plant Asset Management (PAM) systems, Enterprise Resource Planning (ERP), Enterprise Asset Management (EAM) systems, Operational Data Historian Systems (ODHS), and Condition Monitoring (CM) systems.

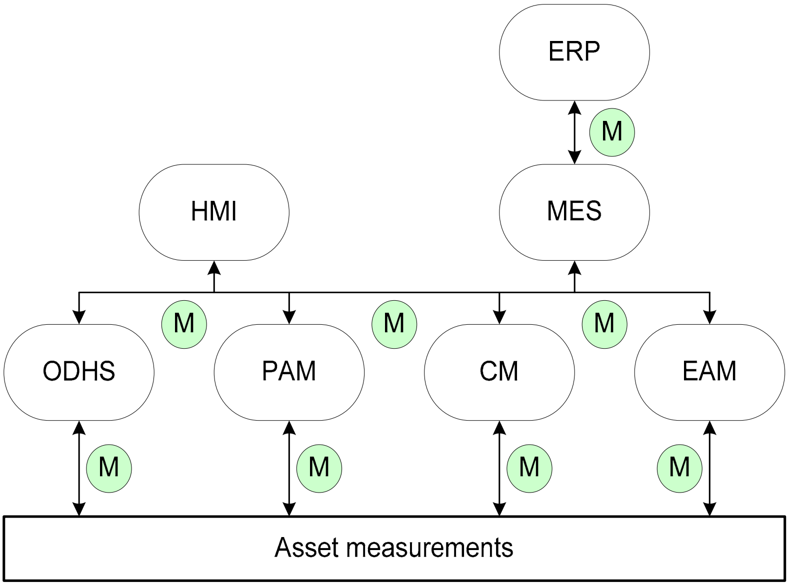


Figure 8. Integration with MIMOSA standards.

The core behind MIMOSA is the definition of the CRIS and Tech-XML schema that describe document sharing between various systems. The Common Relational Information Schema (CRIS) protocol is a common language for transferring and exchanging machine condition monitoring and assessment data between a client's database and another remotely located user. This data transfer is done by data communication conventions such as Java, XML and DCOM.

The CRIS schema covers a broad range of CBM information requirements from machinery data acquisition to trend setting and condition assessment [Lee et al. 2002]. The various types of data are classified into eight categories: trend, machine dynamic, diagnostics, thermography, test samples, asset registration, reliability data and work management. A language-independent description of these data elements is provided in the form of MIMOSA Tech File specifications. As illustrated in Figure 9, various vendors can exchange CBM data through the MIMOSA Tech File architecture, which serves as an information gateway. Thus MIMOSA can be viewed as an architecture that provides archiving and storage of static CBM data [Lee et al. 2002].

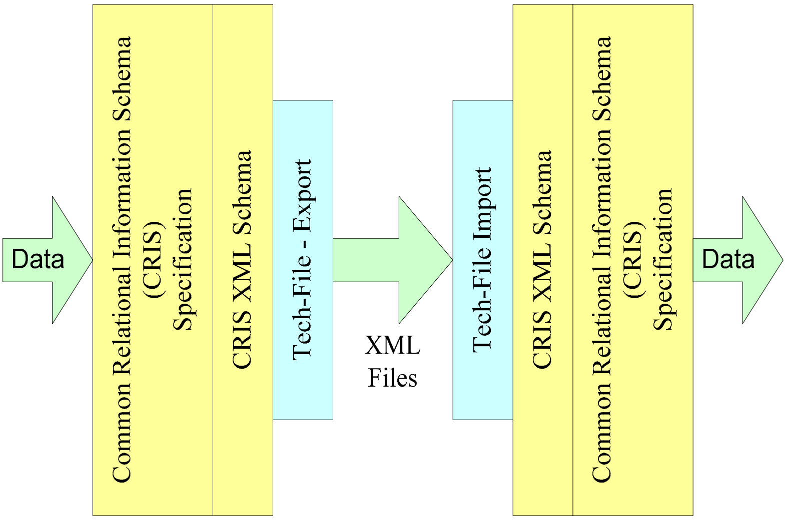


Figure 9. MIMOSA Tech-File architecture for information exchange.

The latest version of CRIS, V3.0, was released in 2005. The schema provides a broad coverage of the types of data that need to be managed within the CBM domain [Lebold & Thurston 2001]:

- A description of the configuration of the system/equipment being monitored
- A list of specific assets being tracked, and their detailed characteristics
- A description of equipment functions, failure modes and failure mode effects
- A record of logged operational events
- A description of the monitoring/measurement system (sensors, data acquisition, measurement locations, etc.) and the characteristics of the monitoring components (calibration history, model number, serial number, etc.)
- A record of sensor data (and its characteristics) whether acquired on-line, manually logged or manually acquired using hand-held roving instrumentation
- A means of describing signal processing algorithms and the resulting output data

- A record of alarm limits and triggered alarms
- A means of describing diagnoses of evolving equipment faults and projections of equipment health trends
- A record of recommended actions and the basis of those recommendations
- A record of work requests from initiation through completion.

4.2 Condition-based maintenance

Condition-based maintenance (CBM) refers to a maintenance strategy that is based on machine health rather than on fixed operating hours or kilometres or other duty cycle-based schedule. Maintenance cost savings are anticipated, not only from avoiding unnecessary servicing but also by servicing the machine prior to unexpected wearing out or an upcoming fault. This requires diagnostic and prognostic actions to assess the expected health of the machine, not only today but also in the coming days or weeks.

Condition-based maintenance is coming to the car servicing business. BMW calls their diagnostic concept of their BMW 5 and 7 series models by name Condition-Based Service (CBS) [Deicke 2002]. The idea is not to return to the old days (before the schedule-based maintenance paradigm came into vogue), where the car was only brought for service when it needed repair, but the idea is to monitor the relevant car parts to predict the "kilometres to go" for the particular part. This simple example illustrates that reactive diagnostics do not make the system a CBM system; rather, that proactive diagnostics must be incorporated. Prognosis is thus the key ingredient of CBM, and the cornerstone of prognosis is diagnostics – i.e. the evaluation of the current health of the machine.

An organisation called the Open System Alliance (including e.g. Caterpillar) has defined an Open System Architecture for Condition-Based Maintenance (OSA-CBM). The target was to develop an architecture that is not exclusive to any hardware implementation, operating system or software technology; to define the distribution of CBM functions to modules; to define the interface to function modules. This architecture consists of seven functional layers, as depicted in Figure 10.

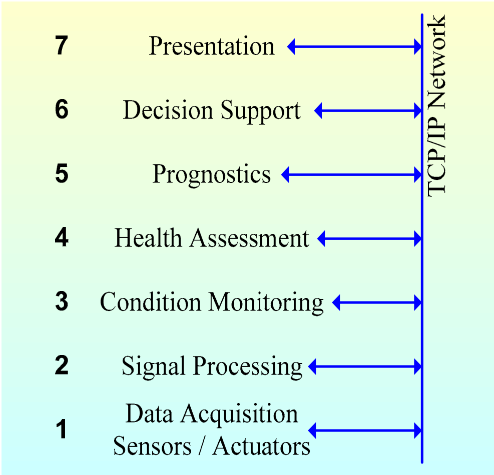


Figure 10. OSA-CBM seven-layer architecture.

The seven layers are described by Lebold & Thurston [2001] as follows (note that the Health Assessment layer is called Diagnostic Processing in the text below):

"Layer 1 – Sensor Module: The sensor module has been generalized to represent the software module that provides system access to digitized sensor or transducer data. The sensor module may represent a specialized data acquisition module that has analog feeds from legacy sensors, or it may collect and consolidate sensor signals from a data bus. Alternately, it might represent the software interface to a smart sensor (e.g. IEEE 1451 compliant sensor). The sensor module is a server of calibrated digitized sensor data records.

Layer 2 – Signal Processing: The signal processing module acquires input data from sensor modules or from other signal processing modules and performs single and multi-channel signal transformations and CBM feature extraction. The outputs of the signal processing layer include: digitally filtered sensor data, frequency spectra, virtual sensor signals, and CBM features.

Layer 3 – Condition Monitor: The condition monitor acquires input data from sensor modules, signal processing modules, and from other condition monitors.

The primary function of the condition monitor is to compare CBM features against expected values or operational limits and output enumerated condition indicators (e.g. level low, level normal, level high, etc). The condition monitor also generates alerts based on defined operational limits. When appropriate data is available, the condition monitor may generate assessments of operational context (current operational state or operational environment). Context assessments are treated, and output, as condition indicators. The condition monitor may schedule the reporting of the sensor, signal processing, or other condition monitors based on condition or context indicators, in this role it acts as a test coordinator. The condition monitor also archives data from the Signal Processing and Sensor Modules, which may be required for downstream processing.

Layer 4 – Diagnostic Processing: The diagnostic processing layer acquires input data from condition monitors or from other diagnostic processing modules. The primary function of the diagnostic processing layer is to determine if the health of a monitored system, subsystem, or piece of equipment is degraded. If the health is degraded, the diagnostic processing layer may generate a diagnostic record that proposes one or more possible fault conditions with an associated confidence. The diagnostic processing module should take into account trends in the health history, operational status and loading, and the maintenance history. The diagnostic processing module should maintain its own archive of required historical data.

Layer 5 – Prognostic Processing: Depending on the modeling approach that is used for prognostics, the prognostic layer may need to acquire data from any of the lower layers within the architecture. The primary function of the prognostic layer is to project the current health state of equipment into the future taking into account estimates of future usage profiles. The prognostics layer may report health status at a future time, or may estimate the remaining useful life (RUL) of an asset given its projected usage profile. Assessments of future health or RUL may have an associated diagnosis of the projected fault condition. The prognostic module should maintain its own archive of required historical data.

Layer 6 – Decision Reasoning: The decision reasoning module acquires data primarily from the diagnostic and prognostics layers. The primary function of the decision reasoning module is to provide recommended actions and *alternatives and the implications of each recommended action. Recommendations include maintenance action schedules, modifying the operational configuration of equipment in order to accomplish mission objectives, or modifying mission profiles to allow mission completion. The decision reasoning module needs to take into account operational history (including usage and maintenance), current and future mission profiles, high-level unit objectives, and resource constraints.*

Layer 7 – Human Interface (Presentation Layer): The human interface layer may access data from any of the other layers within the architecture. Typically high-level status (health assessments, prognostic assessments, or decision reasoning recommendations) and alerts would be displayed, with the ability to drill down when anomalies are reported. In many cases the human interface layer will have multiple layers of access depending on the information needs of the user. This layer may also be implemented as an integrated user interface that takes into account information needs of the users other than CBM."

The main focus of OSA/CBM is on layers 1 through 5.

The core of the OSA/CBM architecture is an object-oriented data model, which has been developed from a mapping of the Mimosa relational schema to the OSA/CBM layers. The data model does not describe all the object classes that would be required for a software implementation. The focus is on describing the structure of the information that is of interest to the clients of that layer.

Here is a list of features of the OSA/CBM components [Lebold et al. 2003]:

- Implement the functionality of individual layers of the architecture
- Communicate in a client/server relationship
- Have EntryPoints that serve the information needs of specific clients
- Provide access to synchronized data channel sets and to background information through their interfaces.

The interface descriptions of each component are [Lebold et al. 2003]:

- Request Data: prompts a measurement or calculation update
- Get Data: returns dynamic measurement data or a calculated result to the client
- Get Explanation: returns a data structure that describes the input data and data transformation processes used in the calculation of the associated output data set
- Get Config.: returns static information on the monitoring system and the monitored system configuration.

4.2.1 Prognostics

Prognostics is an essential function in any CBM system. Its purpose is to help us determine the condition of a machine in the future. More precisely, it means determining the remaining useful life (RUL) of the inspected system. As an operational point, this means the system will give an estimate of the time when, with the current workload, the machine cannot function as expected.

Prognostics can be performed in several different ways. The basic structure is to use the prognostic function as an input – output mapping (Figure 11). Inputs can include current and archived data on the workload, failures, and previous maintenance. Output estimates can include the current efficiency, RUL, trouble spots and recommendations.

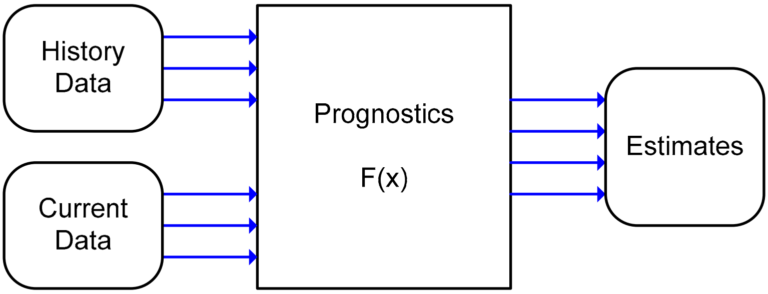


Figure 11. Prognostic function produces estimates.

In [Lebold & Thurston 2001], prognostic approaches are categorized into three groups: model-based, estimation-based and experience-based. The ISO 13381-1

(2004) standard provides guidance for the development of prognosis processes for machine manufacturers.

4.2.2 Other component-based CBM systems

One example is presented in [Wolfram & Isermann 2002], where a component-based tele-diagnosis approach has been developed. Their idea is to equip each sensor and actuator of an overall system with an appropriate signal processing unit and thus create intelligent devices. These are to perform tele-service tasks. So, monitoring can take place in the components independently from the process structure as a whole. A common field bus system connects the components to the control centre PC.

4.2.3 Interpretation of OSA-CBM in machine automation

In this context we will develop the OSA-CBM architecture from the perspective of contemporary work machine control systems. Work machines very often include a powerful PC computer capable of executing most of the tasks dedicated to the OSA-CBM seven-layer architecture. The tailored architecture is depicted in Figure 12. The sixth layer (Decision Support) is omitted.

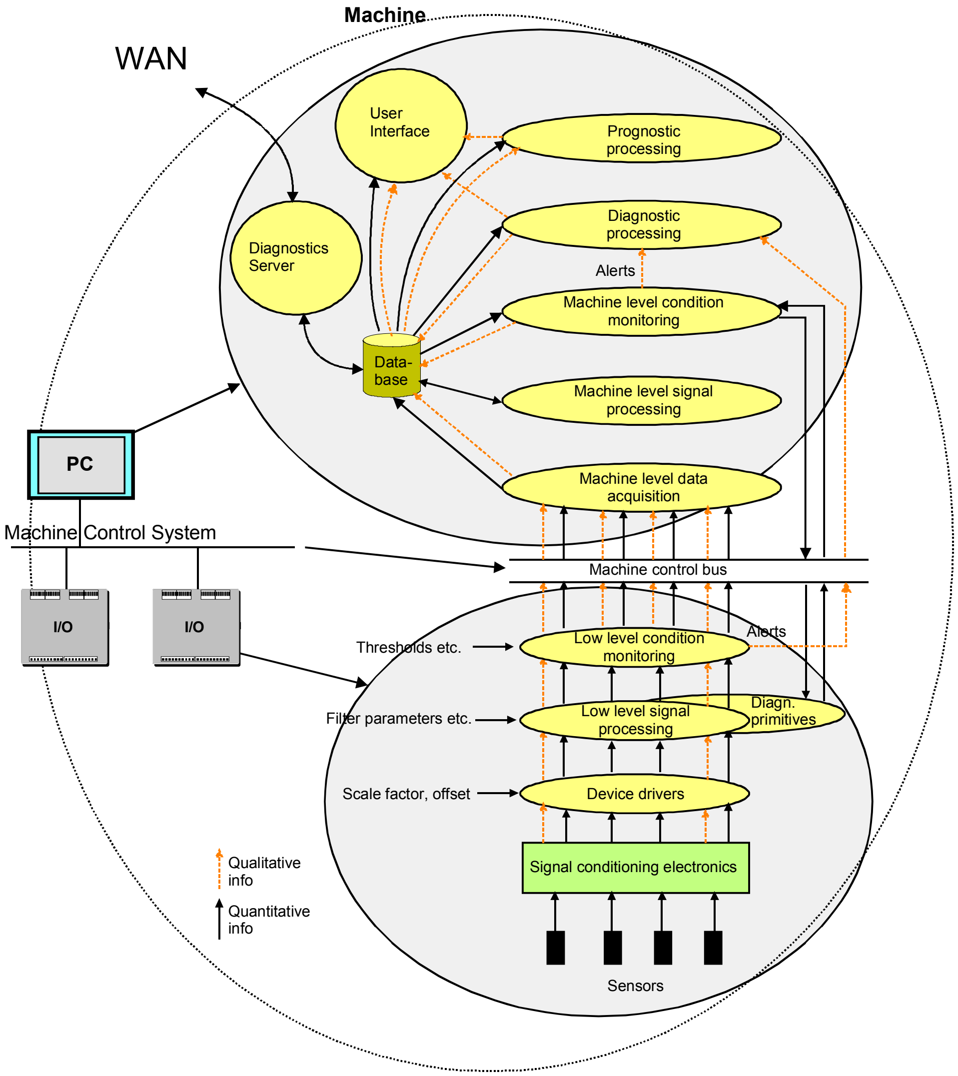


Figure 12. OSA-CBM model tailored to contemporary machine control systems.

4.3 IEEE 1451

The Institute of Electrical and Electronics Engineers (IEEE) 1451 [Lee 2000] is the standard for the smart transducer interface for sensors and actuators. It was developed to solve the problem of sensor interfacing. The common problem is that a large number of sensor networks are currently in the market place, each designed for a specific application class, with its own custom protocols.

IEEE 1451 has defined a set of common communication interfaces for connecting transducers (sensors or actuators) to control networks and instruments in a network-independent environment. It can be placed between the sensor and OSA/CBM's first layer module. The standard defines a framework consisting of data and object models for getting sensor data to the network. This framework is displayed in Figure 13.

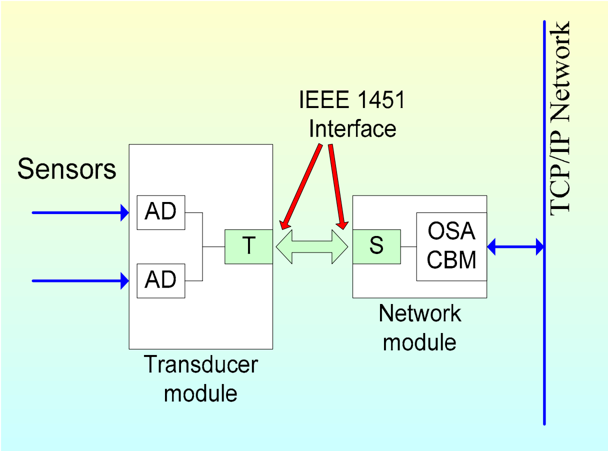


Figure 13. Framework of IEEE between sensors and OSA/CBM's first layer.

The main parts of this standard are a common object model for smart transducers with interface specifications for the components of the model (P1451.1). This contains an STIM (smart transducer interface module) component for the network module, a TEDS (transducer electronic data sheet) component for the transducer module and a digital interface to access the data (P1451.2) between them. This means that the interface is independent of the transducer. Other communication protocols are planned for the family of 1451 standards.

4.4 ISO 13374

The intention with ISO 13374 is to provide the basic requirements for open software specifications that allow machine condition monitoring data and information to be processed, communicated and displayed by various software packages without platform-specific or hardware-specific protocols [ISO 13374-1 2003].

The standard has a slightly modified OSA/CBM approach to data flow. The diagnostic layers and the information flow are presented in Figure 14.

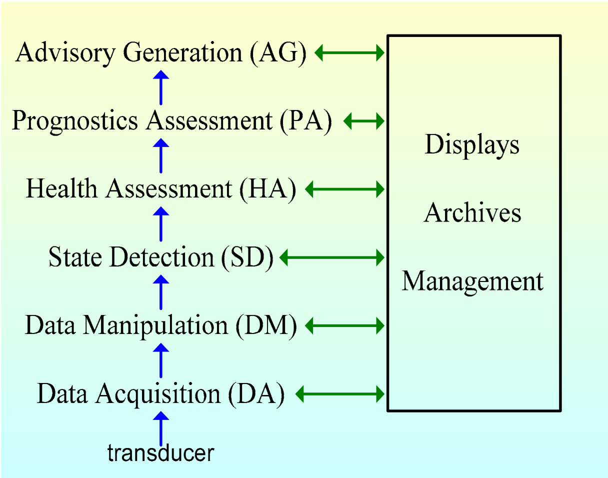


Figure 14. Data processing and information flow in ISO 13374.

The data flow begins at the bottom and finally results in the actions to be taken being presented at the top. As the information flow progresses from data acquisition to advisory generation, data from earlier processing blocks needs to be transferred to the next processing block. Additional information can be acquired from or sent to external systems.

This standardization is directed by the MIMOSA OSA/CBM alliance.

4.5 The OSGi Alliance

The OSGi Alliance (http://www.osgi.org/) is a forum that specifies, creates, advances, and promotes an open service platform. This platform is for the delivery and management of multiple applications and services to all types of networked devices in home, vehicle, mobile and other environments.

The OSGi Alliance tries to put together the needs and ideas that come from communities like service providers, technology, industrial, consumers and automotive electronics. As an independent non-profit corporation, the OSGi

Alliance wants to keep distribution of the information fair between all its members.

The idea of the OSGi service platform is to offer tools to develop, deploy and manage services in a coordinated fashion. The target is to enable flexible and managed services for an entirely new category of smart devices. These include devices such as set top boxes, service gateways, cable modems, consumer electronics, PCs, industrial computers, cars, smart handhelds and more. Service providers like mobile phone operators, cable operators and others are enabled to deliver differentiated and valuable services over their networks. With the OSGi specification, different service provider networks can be seen as one window on existing telephony networks, broadband connections, high-speed wireless data networks to the home, car, mobile and other device environments, and cable entertainment services.

Services in the Home:

- Communication
- Information/entertainment
- Safety and security monitoring
- Energy management and metering
- Appliance diagnostics and servicing
- Telemedicine and healthcare monitoring.

Services in the Car:

- Navigation
- Emergency assistance
- Mobile commerce
- Information/entertainment
- Vehicle diagnostics
- Location-based services.

The OSGi Service Platform (see Figure 15) consists of APIs that define framework standards for service platform devices like a service platform server. These APIs are divided into to a set of core and optional APIs that together define an OSGi-compliant service platform. The core APIs are focused on service delivery, dependency and life cycle management, resource management, and remote service administration. The optional APIs define mechanisms for exporting resources to an HTTP-based web server, client interaction with the service platform and data management.

The OSGi Platform leverages existing Java technology whenever possible. Java is used because it is a portable technology that can run on multiple platforms, including multiple types of service platform devices such as automotive and consumer electronics equipment, household appliances, communications appliances, computers, smart handhelds and more.

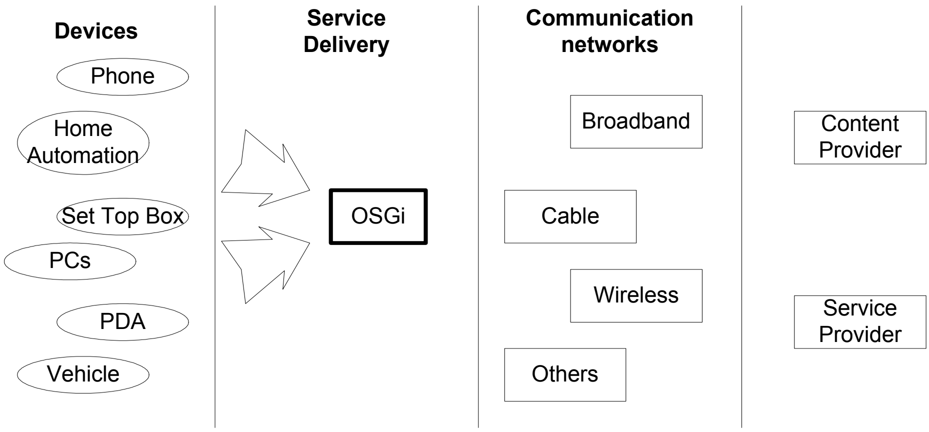


Figure 15. OSGi Service Platform.

The key role in the OSGi alliance is in technical expert groups. All of the core APIs are either contributed by a member or developed by these groups.

The following companies are included in OSGi's Members list (in 2006): *BMW, IBM, Motorola, Nokia, Oracle, Panasonic, ProSyst, Siemens and Sun Microsystem.*

4.5.1 Vehicle Expert Group Charter

OSGi specifications are developed by an Expert Group comprised of the participants. One such a group is the Vehicle Expert Group Charter.

The Vehicle Expert Group (VEG) has a task to focus on the vehicle-specific requirements and to tailor and extend the OSGi specification for this purpose. To achieve this, the VEG defines a list of topics that cover vehicle-specific issues. These topics are put into the requirement documentation and each topic is analysed with the Architecture Expert Group. The idea of this analysis is to define if the topic is relevant to other existing OSGi groups or only to the VEG.

The deliverables of the VEG are requirement documentation and APIs with a reference implementation and a test suite. According to the OSGi policy, the requirements for VEG comes from automotive, transport and telematics companies, as well as from other OSGi groups and other standardization organizations.

In the next list a few vehicle manufacturers are listed with their connection to OSGi:

- BMW Research uses ProSyst's (www.prosyst.com) mBedded Server as an enabling technology for the development and deployment of applications to their vehicle infotainment platform and other devices.
- Bombardier Transportation utilizes ProSyst's mBedded Server and IBM's JVM J9 for its remote diagnosis system (RDS), a wireless remote data transmission system for rail vehicles.
- DaimlerChrysler uses Jentro's and Sun Microsystem's Java and OSGi-based telematics solutions in a pilot showcase.
- GM has made a deal with OnStar Europe to implement the ACUNIA (www.acunia.com) Open Telematics Framework in new car models. Also cooperation with some car rental companies.
- Volvo – VTD is planning to use Gatespace's OSGi Service Platform-based service platform for building infotronic solutions into future generations of Volvo vehicles.

4.6 Agent systems

The definition of an agent is not that simple because there are various things people want to see in agent platforms. One basic definition argues that agents should have the following properties [Woolridge & Jennings 1995]:

- Autonomy
- Proactiveness
- Reactivity
- Social ability.

In that sense, an agent can independently observe the surroundings with its ’sensors’ and act on that information. It is not merely an observer of its environment or a passive recipient of actions performed by other entities. Human beings are the most obvious examples of agents in the real world. The research into multi-agent systems can be thought to have started in 1980 when a small number of AI researchers gathered at MIT for the First Workshop on Distributed AI [Sycara 1998].

It is important to understand the main difference between traditional ’functional’ software systems and agent-based systems. The functional system simply takes input, performs some computation over this input, and finally produces the result. This can be viewed as function from a set of inputs to a set of outputs. In agent systems the basic way of achieving some goal is to negotiate and co-operate with other agents. While doing this, we must understand the basic beliefs and goals of agents.

The Foundation for Intelligent Physical Agents (FIPA) is an international non-profit association of companies and organisations sharing the effort to produce the specifications of generic agent technologies. It began its work in 1996 and is registered in Geneva, Switzerland. One active company from Finland has been Telia Sonera. FIPA’s target is to evolve a set of basic technologies, not just for one application but as generic technologies for various application areas. These technologies can be integrated by developers to make complex systems with a high degree of interoperability.

The main feature and the actual power of agent systems is the ability of agents to communicate with each other. Agents do not just engage in single message exchanges; they have conversations – task-oriented, shared sequences of messages that they follow, such as a negotiation or an auction. At the same time, this is the main feature that differentiates agent systems and client/server type systems.

The agent communication is based on a speech act. This means that when somebody sends a message the other party has to know how to respond to that message, just like a normal conversation between humans. FIPA has specified the FIPA Communicative Act Library, which states the minimum level an agent communicative act must satisfy in order to be FIPA-compliant.

One example of an agent-based diagnostic system is the PEDA system [Hossack et al. 2003]. This was built to access the data from multiple data source and to make diagnostic issues from that. Its functioning agents are:

- Incident and event identification (IEI)
- Fault record retrieval (FRR)
- Fault record interpretation (FRI).

For communication, PEDA agents use ACL-type interactions like subscribe, inform and request. The ontology is also FIPA type and can be extended in future if needed.

4.7 Client - Server

Some vendors have developed their own system for CBM purposes. One good example is Qualtech Systems, who have a server that uses an agent-type structure [Deb et al. 2000, Deb & Ghoshal 2001]. Figure 16 presents the framework built by Qualtech Systems.

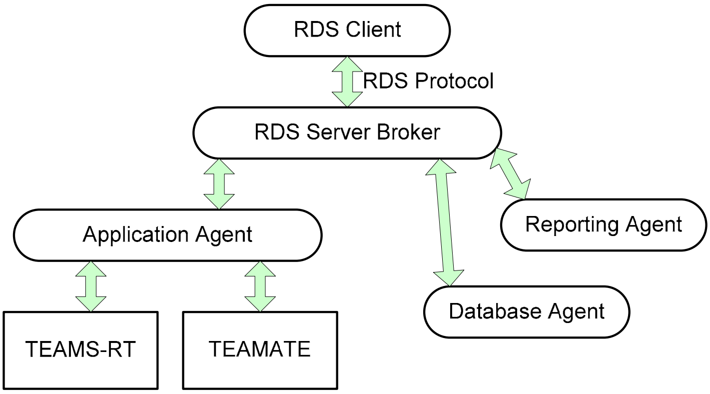


Figure 16. QSI's integrated diagnostics modules.

The modules use their own message structure and protocol, called the RDS Protocol, to communicate. The protocol is based on the ToolTalk architecture, which works very much like an agent system [Deb et al. 2000].

4.8 Example applications

This chapter describes some solutions relating to the condition monitoring of mobile machines. Passenger cars are covered first because many of the technical solutions used in them are similar to mobile work machines. After that, some examples from aviation will be described. Due to the possible risks involved in aviation, fault detection and condition monitoring is very important. Because of this, the use of reactive maintenance is usually out of the question

4.8.1 Passenger cars

Because the technology used in mobile work machines and passenger cars is at least partially very similar, a brief description of the fault detection and maintenance used in them is in place. The high volume of passenger cars makes the technology used in them relatively cheap, which of course is a great benefit for the manufacturers of the work machines. The technology is transferred to the work machines with a little delay and is at that time already tested thoroughly. Naturally, only part of the technology can be directly used but it is always better to reuse something already approved if it is possible.

Some kind of fault detection is nowadays present in most of passenger cars. When faults occur the driver is informed with a signal light or a message on the display. At least, this is the case with the most critical faults. These faults can usually be detected with one sensor, e.g. motor temperature, but more sophisticated methods are also used or at least are being developed.

Most cars have CAN buses to which all the measurements are transferred. Usually, cars have several different buses, divided according to the functions of the devices in them. For example, the devices with a more critical function, such as ABS (Antilock Braking System), ESP (Electronic Stability Program) or airbags, are in a bus with a higher speed and have higher priorities than the other less important devices, e.g. entertainment electronics. Using a CAN bus has an effect on the weight and complexity of the wiring, making it much simpler and, at the same time, reducing the weight of the car by as much as 60 kilograms. A lot of space is also saved by using a CAN bus. The number of different buses varies depending on the car manufacturer. Skoda Octavia, for example, have four different CAN buses and 28-37 devices connected to them: CAN-Drive, CAN-Comfort, CAN-Infotainment and CAN-Diagnostics [Helkama 2004].

Every new car has an On-Board Diagnostics (OBD) connection according to European Unions regulations. A similar system is also used in the U.S. The system used in Europe, Euro On-Board Diagnosis (EOBD), differs slightly from the OBD II used in the U.S.. The EOBD connection allows reading the diagnostic information from the car with the appropriate equipment. The connection and the fault codes are standardized and even the warning light is the same with every manufacturer. The EOBD is mostly used for monitoring the car’s exhaust, but because the concentration of the pollutant components can’t be directly measured it has to be done by monitoring the condition of the components that affect the quality of the exhaust. These components are:

1. catalytic converter
2. lambda probes
3. combustion system
4. secondary air system
5. exhaust gas recirculation system
6. fuel tank ventilation system

7. fuel distribution system
8. CAN databus
9. electronic power control

Even though the EOBD is only used to monitor the quality of the exhaust, the condition of many different components in the car is also checked at the same time [Helkama 2004].

The car’s on-board electronics allows constant monitoring of all the measurements transferred in the CAN bus. Many manufacturers use them in estimating the need for service. This is a good example of how CBM is implemented. Previously, the cars had a fixed service schedule (e.g. every 20000 km or once a year) but nowadays the service can be done when it is needed. This can be done by monitoring several different measurements that mostly differ according to the driving style. Some of the monitored values are fuel consumption, quality and quantity of oil, number of cold starts, etc. The car automatically notifies the driver before the car should be taken to service. This kind of service is now present in cars from many different manufacturers.

Wireless communication is becoming more common in cars every day and in the future almost every car will probably have some kind of wireless communication possibilities. This will allow car manufacturers to collect data from the cars in almost real time, which could then be used for fault detection and identification (FDI). It is of course possible that the customers won’t allow this, even though it is done for their benefit. DaimlerChrysler, for example, has been researching the possibility of transferring information from the car to be used for FDI and as a help for maintenance (see Figure 17) [DaimlerChrysler 2002].

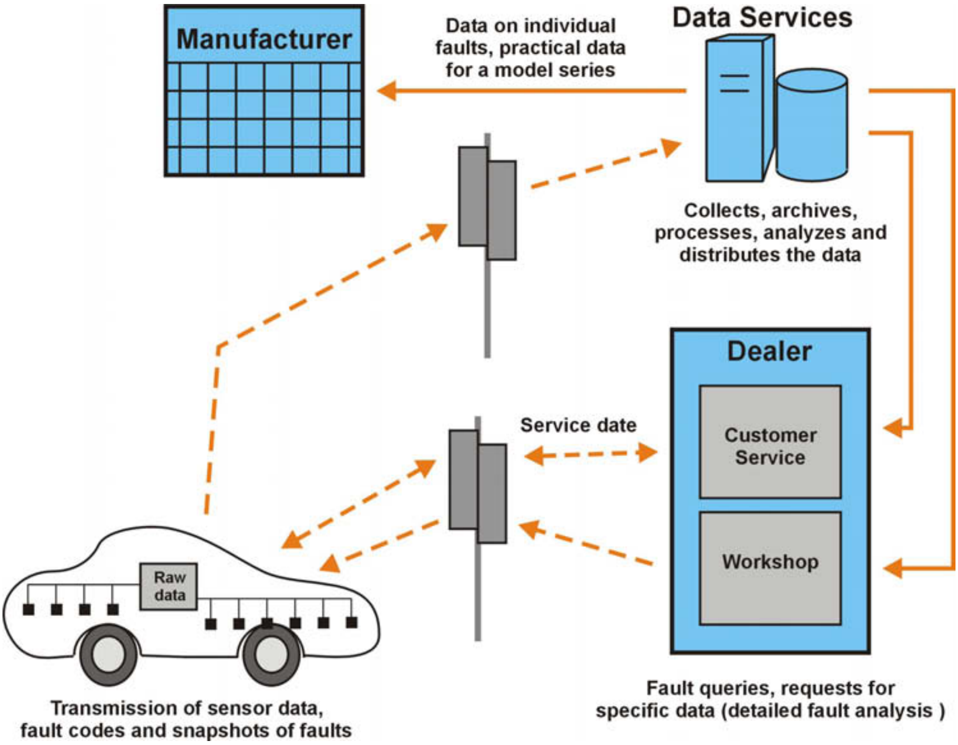


Figure 17. CBM concept of DaimlerChrysler [DaimlerChrysler 2002].

Wireless communications can also be used to provide information and help for the drivers. OnStar is a service available in the U. S. for some vehicle models that provides a variety of different functions for the driver using wireless communication. These include air bag deployment notification, driving directions, emergency services and stolen vehicle tracking. In addition to OnStar, a similar service is provided by ATX technologies [Onstar 2004].

4.8.2 Airbus – Aircraft Maintenance Analysis

In addition to the fault diagnosis used on aircraft, Airbus has developed a ground-based maintenance software: Aircraft Maintenance Analysis (AIRMAN). This software was developed to further enhance the maintenance efficiency by interpreting the ECAM warning (Electronic Centralized Aircraft Monitor) and fault messages downloaded from the aeroplane. The data is transmitted to the ground by radio frequency or satellite communication. The fault monitoring and diagnostic data is used together with the data available on the ground (e.g. aircraft documentation, service information) to provide maintenance actions and analyzed maintenance data for engineering. This improves aircraft dispatch, simplifies maintenance and reduces maintenance cost. The features provided by AIRMAN include gate maintenance, predictive maintenance and data analysis.

The gate maintenance feature allows operators to access information on each aircraft consisting of leg or post-flight reports, technical documents and analyzed fault messages. The predictive maintenance feature uses statistical analysis to classify fault messages from the aeroplane onto a job list. These classifications can be “new today”, ”still open” or “long lasting”. This allows maintenance personnel to take the required maintenance actions before the fault leads to a malfunctioning of the system. It also allows planning the maintenance actions according to the aircraft’s schedule. The data analysis feature processes the data from the Onboard Maintenance System (OMS) to improve the ability of the maintenance personnel to take the most appropriate maintenance actions.

The architecture of AIRMAN consists of a real-time data acquisition module, an Oracle database and the core software application (AIRMAN executable) [Aircraft 2004].

4.8.3 Boeing – Aeroplane Health Management

Like Airbus, Boeing has also developed a service to help with the maintenance of its aircraft: Airplane Health Management (AHM). The in-flight data is transmitted to the ground with radio frequency technology (ACARS). This data is then analyzed with diagnostic and prognostic algorithms during flight and the maintenance crew on the ground can be notified if needed. The tools needed to process the data are hosted by Boeing. Customers get alerts through a fax, PDA, e-mail or pager system. The information can be accessed on the web at MyBoeingFleet.com. In this way the customers (airlines) don’t have to manage and store the data themselves. With AHM the airlines can reduce delays, cancellations and air turnbacks.

AHM was first tested with Air France and British Airways in 2003; Japan Airlines was also included in the test at the beginning of 2004. The first customer of AHM was Singapore Airlines.

4.8.4 The National AirSpace System-Wide Simulation

National AirSpace System-Wide Simulation (NAS Sim) is a program with a goal to develop and implement a comprehensive, integrated health management system for the national (US) aviation system. In addition to health management of individual aircraft, the system also monitors, models and evaluates risks in air traffic over the whole of the United States.

The system consists of simulation models for the different subsystems (engine, wings, etc.) of an aeroplane and a large amount of input data (weather, airline schedules). This data is transferred through a secure network under the Information Power Grid (IPG). The simulation of the subsystems produces modelled values for different parameters, which allow checks for anomalies. This approach is similar to the methods described in Chapter 5.3. For the engine, these values include shaft rpm, inlet and outlet temperatures, and pressures in the compressor and turbine and fuel burn rate. Other subsystem models can be seen in Figure 18 [Bardina et al. 2000].

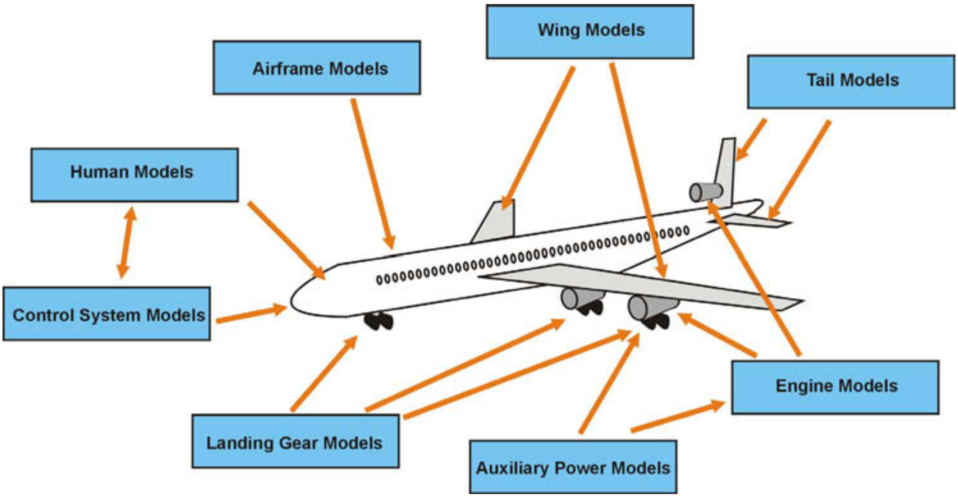


Figure 18. NAS Sim subsystem models [Bardina et al. 2000].

5. Methods for Data Analysis

The purpose of this chapter is to give a brief introduction to the various fault diagnosis methods. Fault detection methods can be classified in many ways. We will use five categories: Data-driven, Analytical, Knowledge-based, Data mining and Model-based. Some methods can be classified into more than one category and combinations of various methods are often used.

Fault detection and identification (FDI) consists of several stages or procedures. There is no explicit classification of these procedures and the terminology varies. For example, in [Patton et al. 2000, Patton & Chen 1992] FDI is simply divided into *fault detection* and *fault isolation*. The fault detection stage only determines whether a fault has occurred or not. The source of the fault is identified in the fault isolation stage - is it a sensor or a faulty actuator? A more profound classification is described in [Chiang et al. 2001], where FDI is divided into four stages:

- Fault detection
- Fault identification
- Fault diagnosis
- Process recovery

Fault <u>detection</u> is the monitoring activity needed to recognize abnormal operation. This may simply be monitoring process variables or trends, threshold values and handling their alarms. Fault detection may also include advanced signal processing based on statistical or model-based methods (see 5.1 and 5.3). In these scenarios the detection of a fault or an abnormal condition is not obvious from single process variables.

In the fault <u>identification</u> stage the observation variables most relevant to diagnosing the fault are identified. This helps to focus on the subsystems most relevant to the fault, making the elimination of the fault more efficient.

Fault <u>diagnosis</u> determines which fault occurred. This means determining the type, location, magnitude and time of the fault.

The effect of the fault is removed in the process recovery stage. In most situations this is not possible automatically and it might need human assistance. A term other than process recovery would be more applicable in mobile machines, but the idea is the same.

The normal FDI sequence is represented in Figure 19. If a fault is detected, fault identification, fault diagnosis and finally process recovery are employed.

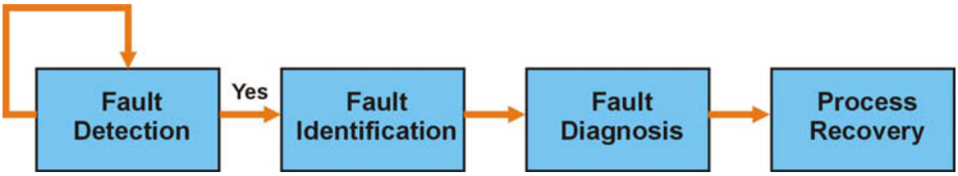


Figure 19. FDI sequence [Chiang et al. 2001]

Another division of activities in fault diagnosis is from OSA-CBM (a MIMOSA activity). They have defined the interfaces of some essential functional components of a condition-based maintenance system (CBM). These components present functionalities, which may also be regarded as phases of a 'CBM cycle':

- Data Manipulation
- Condition Monitoring
- Health Assessment
- Prognostics
- Decision Reasoning

5.1 Model-based approach

When somebody builds or uses a machine or vehicle he has a good idea of what it should do and how it should work. This view or idea is in a human's head. If the development is done in an iterative way, the output is always compared with this general view of how things should work. If the output is not what is expected, some adjustment has to be done to the machine until the required output is achieved.

The basic idea of model-based diagnostics is that some machine operations are modelled to describe its functions. Generally, this model is first in the head of person that is developing the machine and finally in the head of the machine operator. This information can be put into a form that a computer can use as a reference model of how the system should work. If there is difference between this reference model and the actual response of the machine, there is an error somewhere (see Figure 20).

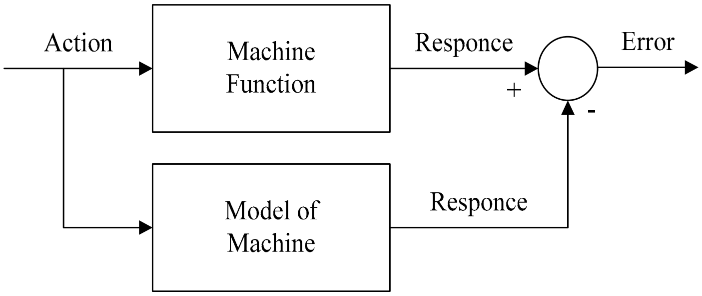


Figure 20. Basic fault detection.

Machine functions can be divided [Kurki 1995] into structural, functional and behavioural models. Functional models are impractical, expensive and difficult to update. Structural and behavioural information is generally available in some form.

The topology of a machine is generally well known and documented. This component hierarchy, and their relationship information can be used when a structural model is derived. The reuse of this information is dependent on the applied method. Methods like rule-based systems, expert systems and fuzzy systems belong to this category.

Behavioural models represent information on how the machine responds to dynamic control actions in the state the machine is in. Generally, these models include inputs, controls, variables and parameters. Knowing the context, dynamic behavioural can be monitored. An example of a machine's low-level control functions can be found in the pressure control in a vehicle's braking system. If the pressure doesn't reach the target, a fault is detected. A high-level example of this system is when bringing the vehicle to a stop from its current speed. We all know that the speed is reduced slowly, depending on the force put into the brakes and some other factors that might be affecting the vehicle. In this case the context and dynamic of the system is very important. There are various methods for detecting threshold overshoots, analogue signals for peaks, slopes and variations, durations of sequences and events. These are described in more detail in this chapter.

Some requirements should be considered when modelling a machine function in practice [Kurki 1995]:

- Model development has to be cost-effective and flexible.
- Models have to be easy to customize and tailor.
- Updating of the models has to be flexible.
- Domain experts have to be able to configure and update the models.

The most problematic reason for a model-based system not working is that the model is out of date. Usually, this means that it has worked fine until some modifications have been made to the machine. Such modifications can be either mechanical or electrical. Some procedures are required after any modifications to keep any possible model parameters correct. One way to automate this is to add adaptive functions to fix the required parameters.

Basic control system models are mostly stable or easily modified. More complicated systems can use an adaptive approach, like parameter estimation techniques [Isermann 2004] or case-based reasoning techniques [Kolodner 1993]. Adaptive systems can be very good if parameter modification is working properly. They can also be useless if the parameters are totally wrong.

Fault diagnosis is generally divided into fault detection, fault localization and recovery. In fault detection the actual input output data is compared with the model's output, as in Figure 20. Models perform very well when detecting exceptions in system behaviour. But when is the system behaving in the wrong way? How big is the error? To make the detection accurate and reliable might require some threshold parameter tuning.

It is easier to tell that something is wrong but harder to say what is actually causing the fault. Fault localization is tasked with giving the information on what caused the fault. The basic problem is that it might find zero, one or too many causes for the fault. Structural models are good in fault localization because they generally exist and are quite reliable. Fuzzy rule-based systems and case-based methods are also common. One interesting way of getting fault localization information is to use inverted models [Rauma 1999].

The following chapters will present some methods that are generally used in diagnostics. Most are used to model the system or a part of it. The kind of system the model is based on is vague. Methods for clustering, e.g., expect some particular features of events, or methods for causality expect causal connections.

5.2 Data-driven methods

Data-driven methods directly use the data measured from the system. The high-dimensional data (each measurement signal presents a "dimension") can be transformed into a lower dimension with the aid of statistical computing. Interesting states in the process (usually faults) cannot always be seen directly from just a few measurement signals. It may well be possible to detect such states by analyzing the combined variability of a large number of signals in a reduced dimension (e.g. in two dimensions on an x-y chart). These methods are highly dependent on the quantity and quality of the measured data.

Some typical data-based methods are principal component analysis (PCA), Fisher Discriminant Analysis (FDA) and partial least squares (PLS). PCA and FDA will be briefly described next. Statistical Process Control (SPC) is mathematically less complicated than PCA and FDA but it is more widely used in industrial applications. Typical application areas of data-based methods are complicated systems with a high number (e.g. hundreds) of continuous measurement signals. These are commonly found in the process industries.

SPC is not a dimension reduction technique and it has a wider scope of application due to its simplicity and adaptability.

5.2.1 Principal Component Analysis

Principal component analysis (PCA) is one of the dimension reduction techniques used in FDI. The basic idea of PCA is to represent the data in a lower-dimensional format while still preserving the correlation between the variables. This can be done because usually all the variables measured from the system are not the principal driving forces behind the system's behaviour. PCA produces new variables that are linear combinations of the original measurements. These variables are presented with orthogonal axes. The first axis captures most of the variation from the original data. The full set of principal components has the same size as the original data but usually only two or three axes are enough to capture most of the variation.

The number of principal components needed for this can be determined by several different methods. These methods include the percentage variance test, the screen test, parallel analysis and the PRESS (prediction residual sum of squares) statistic. Fault detection can be done using a T^2 or Q statistic for the lower-dimensional PCA representation. Differences from the normal operating state usually have large effects on the Q and T^2 statistics. Faults can be detected by using appropriate thresholds for the statistics. The T^2 and Q statistics tend to detect faults differently and the best results can be achieved by using both statistics together. An example of this is presented in Figure 21, where 'x' is data collected during normal operation, '+' is a fault that can be detected with the Q statistic and 'o' is a fault detectable with the T^2 statistic.

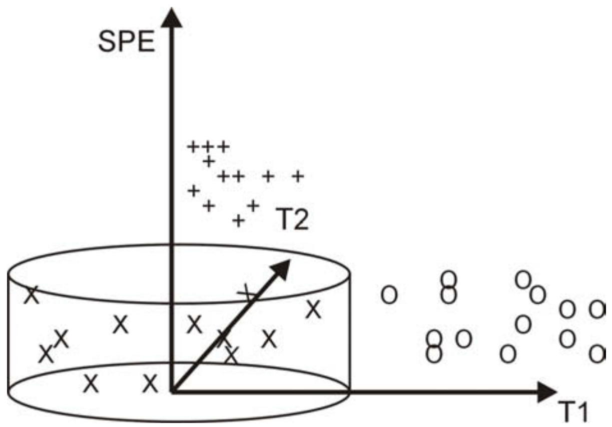


Figure 21. Fault detection with Q and T^2 statistics [Chiang et al. 2001].

The simplest way of doing fault diagnosis with PCA is to define regions in the lower-dimensional space to classify different faults. Another method is to construct separate PCA models for each fault and then use some statistic (e.g. Q or T^2) to predict which fault has most likely occurred. More information on using PCA and T^2 or Q statistics in FDI can be found in [Chiang et al. 2001].

[Mattila 2003, Chiang et al. 2001, Matlab]

5.2.2 Fisher Discriminant Analysis

Fisher Discriminant Analysis (FDA) is a dimension reduction technique well suited to classification of data. While PCA is optimal in capturing the variance of the data, FDA separates the data into different classes maximizing the scatter between the classes and minimizing the scatter within each class. FDA can be applied to data collected during normal operation and when different faults have occurred. Fault detection can be done easily if the data during normal operation can be separated from the different fault classes. Similarly, fault diagnosis is possible if the fault classes are reasonably separated.

In Figure 22 presents the classification of the same testing data in three classes using FDA and PCA. It can be seen that FDA separates the classes better.

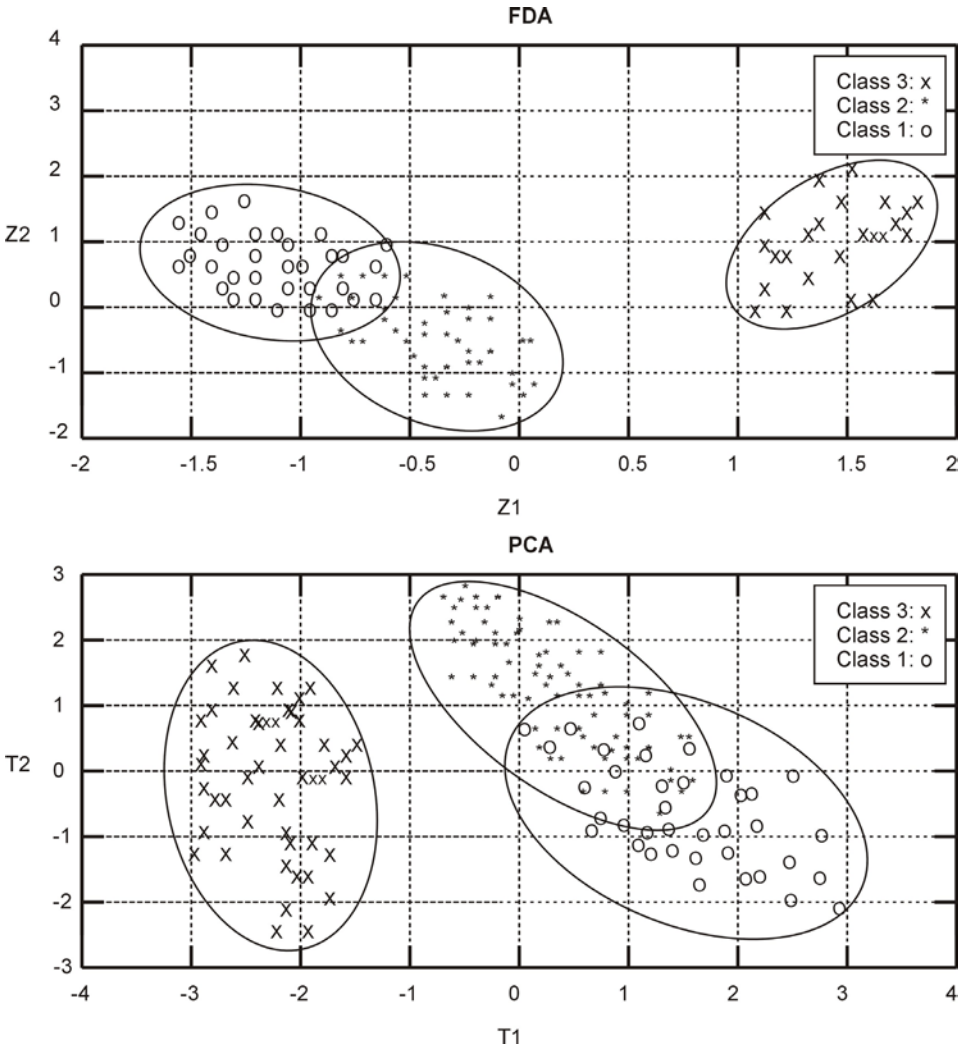


Figure 22. Classification of data with FDA and PCA [Chiang et al. 2001].

5.2.3 Statistical Process Control

Statistical Process Control (SPC) is a collection of methods suitable for a wide range of process control and monitoring applications and quality management. Here we describe some of SPC's features applicable to simple machine condition monitoring and fault detection.

The most common visual presentation of an SPC system is a process control chart showing a trend plot of measurement samples and levels for upper and lower warnings and alarms. The limits for the warnings and alarms may be based on the standard deviation of the variable, measured during a certain warm-up period. Examples of commonly used limits are $\pm 3\sigma$ and $\pm 5\sigma$ ('three times sigma or five times sigma'), sigma being the measured variation of the process variable. SPC methods assume that the input data is normally distributed [Oakland 1990],

Figure 23 shows how the means of measured samples can be used in process control. The distribution of the sample means is on the left. The lower warning limit is $2\sigma / \sqrt{n}$ and approximately 1 in 40 samples will be over the limit. In FDI, similar limits for different measurements (e.g. pressure) can be calculated from historical data. Measured means exceeding the predefined limits indicate faults or degrading of condition. Several limits can be used to classify the severity of the fault.

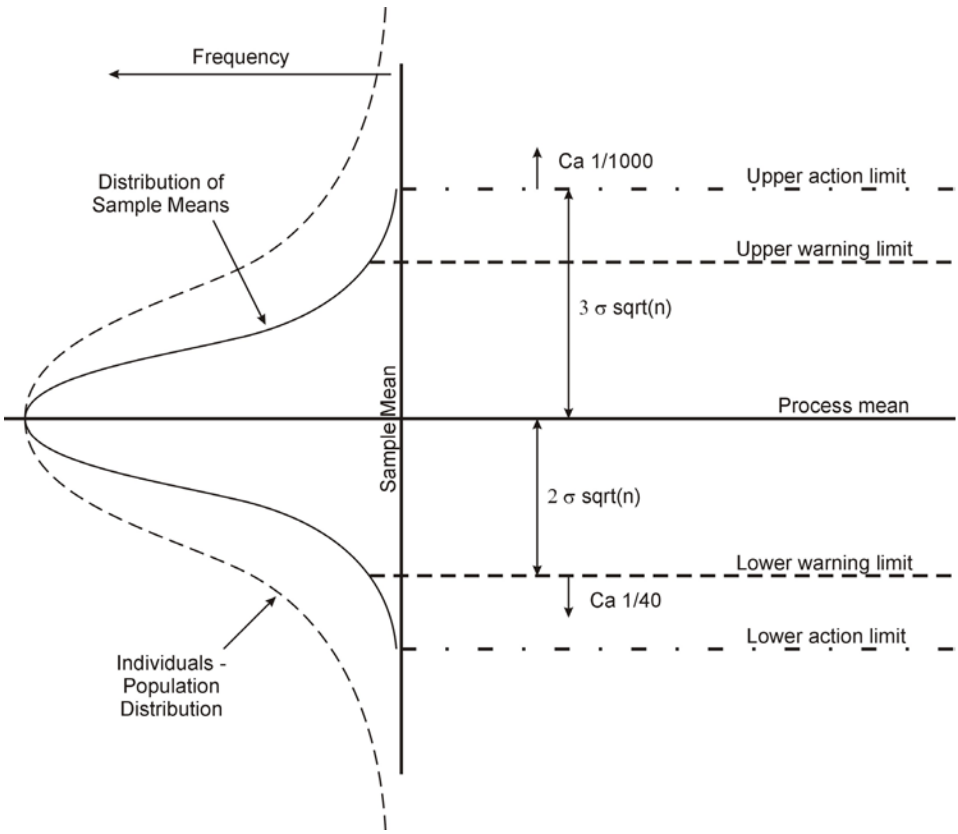


Figure 23. Mean control chart [Oakland & Followell 1990].

5.3 Analytical methods

Analytical methods use features generated from the input and output using detailed mathematical models. Fault detection and diagnosis is done by comparing the features from the models with the features associated with normal operating conditions. The features normally used include residuals, parameter estimates and state estimates [Chiang et al. 2001, Mattila 2003]. An essential requirement for the case is that a model needs to be created. Here it helps if the system and its dynamic behaviour are well known by the engineers. The model needs to be precise enough to produce reliable fault diagnosis, but, on the other hand, modelling is very time consuming and it may have to be repeated whenever the system changes even a little. This causes major model management efforts.

Residual-based methods are most commonly used and are referred to as *analytical redundancy* methods. The residuals are the differences between the measured signals and the corresponding signals of the mathematical model. Ideally the residuals should be zero when the system operates normally and non-zero in the presence of faults and disturbances. Non-zero residuals are also generated by noise, disturbances and modelling errors, making distinguishing of the residuals caused by faults very challenging. The faults can be detected by using appropriate thresholds for the residuals if the residuals caused by the faults are larger than those caused by noise, disturbances and modelling errors [Chiang et al. 2001, Mattila 2003, Simani et al. 2003].

The residuals are usually generated by one of the following methods [Chiang et al. 2001, Simani et al. 2003]:

- Parameter estimation
- Observers
- Parity relations.

Figure 24 shows a typical structure of an analytical FDI system. Residuals are generated from the measured inputs and outputs. The residuals should be close to zero when no faults have occurred. Next, the residuals are evaluated to determine if a fault has occurred and, possibly, diagnose the fault more precisely.

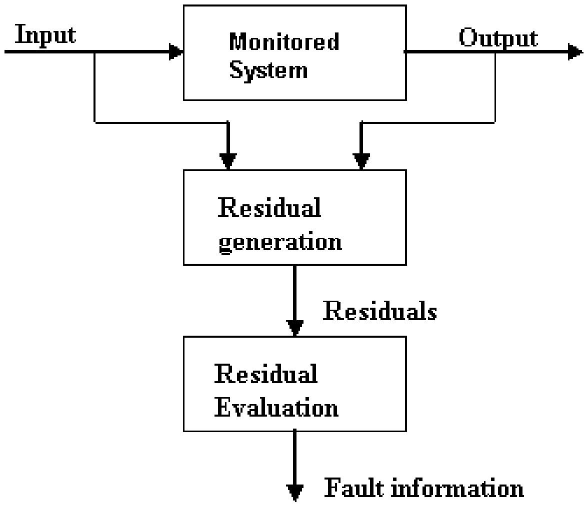


Figure 24. Typical structure of an analytical FDI system [Simani et al. 2003].

For analytical methods the faults can be seen as either *additive* or *multiplicative*. The additive faults are usually caused by changes in actuators, sensors or immeasurable state variables. The multiplicative faults may be caused by parametric faults or modelling errors. The modelling error may be due to inaccuracies in physical parameters, simplification of a higher-order model or because of approximation of a non-linear model with a linear model. Additive faults affect the output by summation, whereas the multiplicative faults affect the output by multiplication. If the input is doubled, the effect of the faults also doubles for the multiplicative faults. For additive faults the effect would remain the same.

Analytical FDI methods are discussed in [Tan & Sepehri 2002, Patton & Chen 1992 and Patton 1991].

5.3.1 Parameter estimation

Parameter estimation is applicable when the faults affecting the system are multiplicative and the basic mathematical structure of the model is known. The parameters of the models can be estimated with standard parameter estimation techniques, which are not discussed here. When the model is constructed from first-principles (physical facts, laws of nature) the model parameters have direct physical meanings. Thresholds can then be set between nominal and estimated parameters for fault detection.

The parameter estimation follows the following sequence:

1. Define the process equations that relate the input and the physical model parameters to the output
2. If necessary, simplify the model or combine the physical parameters so that the model parameters can be determined uniquely
3. Estimate the nominal model parameters from historical training data
4. Calculate estimates of the physical parameters from the estimated model parameters
5. Faults can be detected during operation by comparing the estimated physical parameters with those obtained from the training data. Fault isolation can be done by comparing changes in the parameters with historical observations in a database

[Chiang et al. 2001, Mattila 2003]

5.3.2 Observer-based methods

Observer-based methods are best suited to situations where the faults are additive - i.e. faults affecting sensors, actuators or immeasurable state variables. These methods need detailed mathematical models that are derived from first-principles so that the states have a physical meaning.

Fault detection can be done based on residuals. For the measurable states the residual is the difference between the measured and the estimated state. The immeasurable states can be constructed from measurable inputs and outputs with a Luenberger observer or Kalman filter. The residuals used with these are generated from the difference between measured and estimated output. Usually, the states are not measurable and a Luenberger observer or Kalman filter should be used for FDI.

[Chiang et al. 2001, Mattila 2003 and Simani et al. 2003]

5.3.3 Parity relations

Another analytical method for residual generation is parity relations. When using parity relations for FDI the residual vector is generated from the observations. A transfer function matrix, W, is used to separate the faults from other disturbances in the residual. The transfer function matrix W should be designed so that non-zero residuals only occur when faults are present. In real-life situations the residual is also affected by noise, modelling errors and disturbances.

The effect of the noise on the residual can be reduced by using low-pass filtering; Kalman filtering can be used on more complex noise signals. Although the effect of the noise can be rather easily reduced, some effect on the residual will still remain and some threshold on fault detection must be used. The effect of the modelling error on the residual is more difficult to handle. According to [Chiang et al. 2001], there are two mainly used methods in the literature for this. These are robust residual generators and structured residuals with an unknown input observer. The modelling errors can be modelled as disturbances and the transformation matrix W can be designed so that the residual is insensitive to these disturbances as well. This approach requires the number of disturbances and model uncertainties to be small.

Ideally, the residual generated with the transformation matrix W will be sensitive to each type of fault. A triggering limit can be used to measure the sensitivity to each fault. The exact definition of the triggering limit is not considered here but can be found in [Chiang et al. 2001] or [Mattila 2003].

[Chiang et al. 2001, Mattila 2003]

5.4 Knowledge-based methods

Knowledge-based methods use qualitative models instead of the highly mathematical models used in analytical fault detection methods. Unlike many other method groups, knowledge-based methods are not used for fault detection. Instead, they provide human-like “expert” help for identifying faults detected by other means. These models can be constructed with causal modelling, expert interviews, non-mathematical system modelling and case-based reasoning (CBR). Using qualitative models is applicable when the mathematical model is impossible or very difficult to construct. Experienced engineers can usually diagnose the system's, e.g. the motor's, condition without knowing the exact mathematical model. As experienced engineers are expensive and often not available all the time, some other approach is needed for FDI. Knowledge-based methods try to remove the need for an engineer to monitor the system all the time [Chiang et al. 2001, Chow 1997, Mattila 2003].

5.4.1 Causal Analysis

Causal analysis uses causal modelling of fault-symptom relationships for FDI. These methods are best suited to fault diagnosis. Signed directed graph and expert systems are some examples of methods based on causal analysis.

Signed directed graph (SDG) consists of nodes and signed branches that represent system variables and faults, and the relationships between them. The nodes represent the state or process variables, such as flow rate. First, thresholds for high and low values are assigned for each variable. A node takes a value of 0, + or – according to the variable's value - i.e. normal, high or low. The nodes can also represent system faults or component failures, such as a stuck valve or a leak. The signed branches represent the cause-effect relationships between the nodes. A branch has a value of + if the cause and effect change in the same direction. Respectively, the branch will have a value of – if the change is in the opposite direction.

Fault diagnosis can be done by tracing through the net to the root node that is causing the abnormal behaviour. "The goal of utilizing a SDG for diagnosing faults is to locate the root node(s) representing the system faults based on the observed symptoms. To achieve this, the measured node deviations are propagated from effect nodes to cause nodes via consistent arcs until the root nodes are identified. An arc is consistent if the sign of the cause node times the sign of the arc times the sign of the effect node is positive." [Chiang et al. 2001],

The use of SDG for FDI is covered in [Shiozaki et al. 1989].

5.4.2 Expert Systems

Human-like problem solving can be applied to FDI with expert systems. A well-developed expert system is able to:

- represent existing expert knowledge
- accommodate existing databases
- accumulate new knowledge
- make logical inferences
- make recommendations
- make decisions with reasoning

[Chiang et al. 2001]

Usually, the expert systems consist of a knowledge base, an inference machine and a human system interface.

The knowledge base contains the required knowledge of the system. Expert systems can be seen as being either shallow-knowledge or deep-knowledge systems, depending on how the knowledge base is built. Various types of representations can be used for the stored knowledge. The quality and quantity of the information in the knowledge base is essential for the system to work properly. The knowledge base has to be updateable to benefit from the new knowledge and experience received during operation.

The knowledge base is used by the inference engine for evaluation of the current situation. The most common inference mechanism is forward/backward chaining. Other inference mechanisms include hypothesis test, heuristic search or artificial neural networks [Chiang et al. 2001, Mattila 2003].

The ability of an experienced engineer to diagnose faults in shallow-knowledge expert systems is formulated into IF-THEN rules. These rules can be obtained without a deep understanding of the system mechanisms or physics. Shallow-knowledge expert systems are flexible and, usually, the conclusions on the system can be easily verified. The results from the system are strongly dependent on adequate knowledge. When faced with a novel situation, a solution for the problem might not be found.

Deep-knowledge expert systems are based on engineering fundamentals, a structural description of the system or a description of the system's components' behaviour during faults and normal operation. As is the case with analytical methods, a mathematical model of the system is usually needed for building deep-knowledge expert systems.

More information on expert systems is available from [Correcher et al. 2002] and [Ready 1991].

5.4.3 Artificial Neural Networks

An artificial neural network (ANN) is an FDI method that is difficult to place into just one of the described three method categories. Here it is placed under knowledge-based methods since it is in a way similar to human thinking. ANN tries to emulate the structure of the human brain. Neural networks are also used for hybrid FDI systems that use more than one method. Some examples of this are described in Chapter 3.4.

Several different network architectures have been developed. According to [Kohonen 1990], the network architectures can be categorized into three main types: Feed forward networks, Recurrent networks and Self-organizing networks.

The feed forward network will be described more accurately since 80% of all ANN applications [Chow 1997] are of this type. Self-organizing maps (SOM) are also discussed because they are better suited to classification problems. The classification capabilities of ANNs are interesting for FDI. In addition to classification, the other task that ANNs are usually used for is function approximation [Koivo 2004].

5.4.4 Feed forward neural network

The feed forward neural network consists of several neurons in different layers. The basic structure of a neuron is shown in Figure 25:

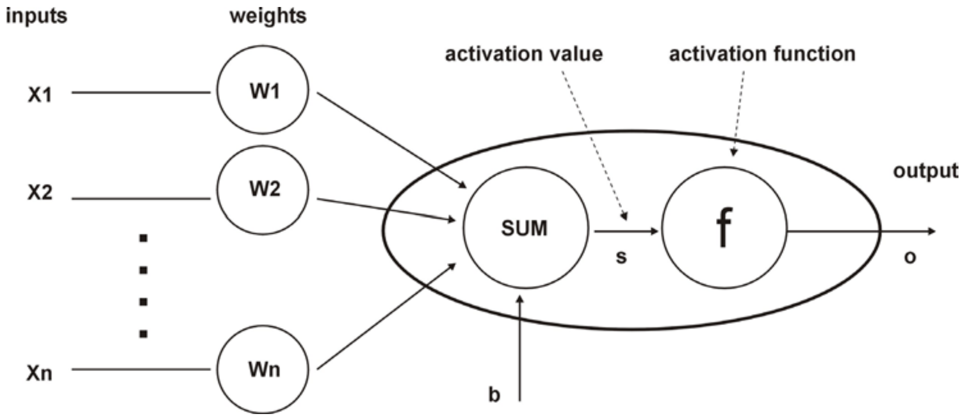


Figure 25. Structure of an artificial neuron [Chow 1997]

All the inputs $x = [x_1, x_2, \ldots, x_n]^T$ of the neuron are multiplied by weights $w = [w_1, w_2, \ldots, w_n]^T$. The neuron has also an optional bias term, b. The sum of the weighted inputs and the bias form the activation value s. The output, o, of the neuron is formed from the activation value with the activation or transfer function f: $o = f(s)$. Typical activation functions are: Threshold function, Piecewise linear and Sigmoid functions [Koivo 2004].

Of these, the sigmoid functions are most popular because they behave in the same way as the human neuron [Chow 1997]. The sigmoid functions are also bounded and monotonically decreasing, and are differentiable everywhere [Chiang et al. 2001].

The network has one input layer, one output layer and one or more hidden layers. Every neuron in the hidden layers and the output layer receives inputs from the previous layer. In a feed forward network the neurons are only connected to the neurons in the next layer and there are no cyclic connections. If every neuron in a layer is connected to every neuron in the next layer, the network is called fully connected. Otherwise, the network is partially connected. Figure 26 shows the structure of a feed forward network:

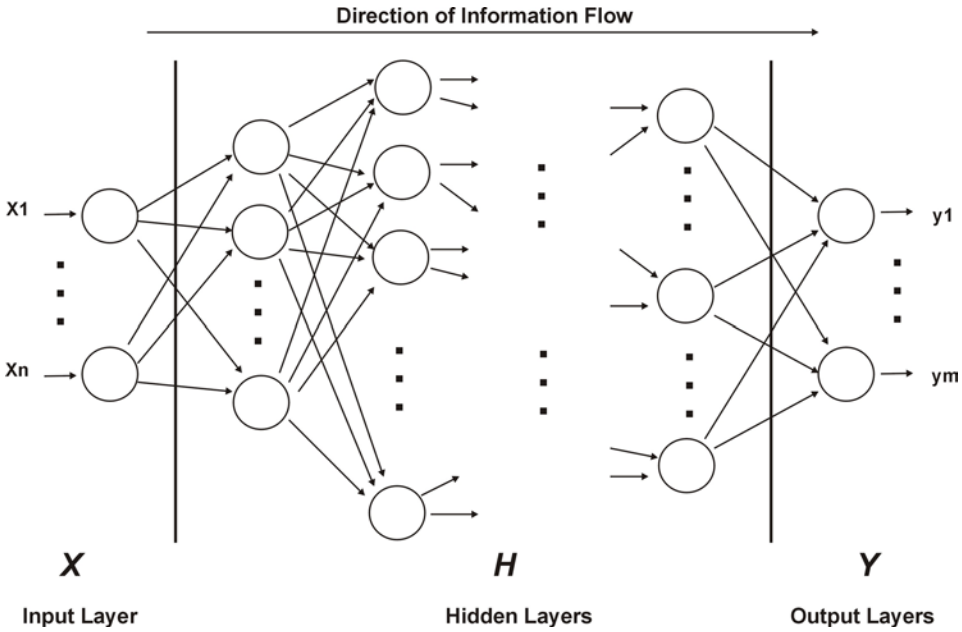


Figure 26. Basic structure of a multilayer feed forward network [Chow 1997]

A network with one hidden layer is called a three-layer network and is the most popular choice for the network structure.

A single neuron in the network is not computationally very powerful in itself. However, the numerous neurons form a complex network in which the inputs are propagated through the layers to finally get the outputs. Each connection between the neurons has a weight associated with it and it is by these weights that the network is trained. Initially, the user has to choose the structure of the network - i.e. the number of layers, the number of neurons in each layer and the initial weights. After this, only the connection weights are modified.

Training the network is done by updating the weights of the connections according to the error of the network outputs. The weights are updated until the error is within the tolerance level the user has defined. Different training algorithms are used to determine how the connection weights are adjusted to minimize the error. The most used algorithm is the back propagation algorithm.

One way of using neural networks in FDI is to classify the inputs into different fault classes. In this case the number of neurons in the input layer is equal to the number of process variables and the number of neurons in the output layer equal to the number of different fault classes. The number of outputs is then equal to the number of faults in the training data. It has to be noted that neural networks cannot operate properly with measurements outside the training data. This means that only faults that appear in the training data can be identified. The neural network has to be retrained for new faults.

Information on using neural networks in FDI can be found in [Sorsa et al. 1991]. [McCormik & Nandi 1996] have made a comparison of neural networks and statistical methods in condition classification.

5.4.5 Case-Based Reasoning

Case-Based Reasoning (CBR) is an 'intelligent systems' ('knowledge-based systems', 'expert-systems', etc.) methodology developed in the Artificial Intelligence research community during the 1990s [Watson 1997, Kolodner 1993]. Previous achievements in this research domain are the Rule-based expert systems (70s and 80s). Rule-based means that knowledge of, for example, a diagnostics task is formulated as a set of logical rules, and when a new fault occurs these rules are supposed to lead to the roots of the problem and suggest a solution. However, transforming expert knowledge of a certain domain or system into a set of rules is not always feasible or even possible. The knowledge of an area of expertise may be poorly structured or new situations simply cannot be solved by chains of abstracted (generalized) rules. Case-Based Reasoning does not involve rules; instead, it utilizes a stored base of earlier experiences or cases. In this way the CBR method tries to mimic human and animal reasoning, which is based on memorized solutions to earlier situations. The stored case base can be queried and the retrieved solutions can be reused. Important issues are how stored cases are presented and how new situations are described to the system to make queries for similar or closely matching situations.

A basic CBR cycle has the following steps (Figure 27):

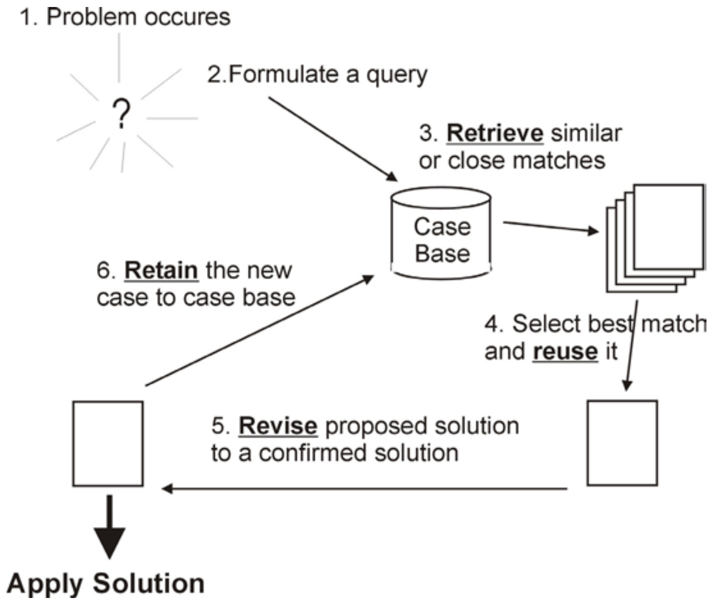


Figure 27. CBR cycle [Watson 1997].

1. A problem occurs
2. Describe the problem's features as a query
3. Retrieve similar or closely matching cases from the case base
4. Decide which case is the best representative of the current situation and reuse it as a *proposed solution*
5. Revise the proposed case (solution), if necessary, to a *confirmed solution*
6. Retain the new case and solution by storing it in the case base.

[Kolodner 1993, Watson 1997]

Kaidara's Advisor is a CBR tool [Kaidara 2004] that is used by Wärtsilä in the CBM of engines. The condition of the engine is evaluated by comparing the current maintenance data on major engine components with referenced cases. It is also used by Citroën to help the after-sales personnel diagnose faults in vehicles.

5.5 Data mining

In this section we describe data mining, how it is defined, why it is applied, what is involved in the data mining process, typical methods, etc. Much of the text below is based on Klösgen & Zytkow [2002a] and Berry & Linoff [2004]. At the end we present an example on how data mining can be applied; methods suitable for time series handling are emphasized a bit. We by-pass many important topics related to data mining, like data warehousing, knowledge representation techniques and model validation, to name a few. Often, the knowledge representation is tightly connected to the mining method, but not always.

5.5.1 What is data mining?

Some people use "Knowledge discovery in Databases (KDD)" as a synonym for data mining. Data mining refers to a broad set of different techniques - for example, the database can be understood as the Internet - but it also refers to the process of building data mining solutions. Some of the methods handled in Chapter 5 are often included in the concept of data mining. A Google search on "define:data mining" gives a long list of possible definitions for data mining. Here is one:

> "*An information extraction activity whose goal is to discover hidden facts contained in databases. Using a combination of machine learning, statistical analysis, modeling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results. Typical applications include market segmentation, customer profiling, fraud detection, evaluation of retail promotions, and credit risk analysis.*"
> (http://scianta.com/technology/datamining.htm [Referenced 10.04.2006])

We show three other definitions just to get an idea of what data mining is:

> "*The comparison and study of large databases in order to discover new data relationships. Mining a clinical database may produce new insights on outcomes, alternate treatments or effects of treatment on different races and genders.*"

(http://www.theebusinesssite.com/IT%20Terms/Health%20Terms.htm [Referenced 10.04.2006])

"*A technique to analyse data in very large databases. Analysis can reveal trends and patterns and can be used to improve vital business processes.*" (http://www.knowledgepoint.com.au/starting_out/glossary.htm [Referenced 10.04.2006])

"*A new discipline lying at the interface of statistics, data base technology, pattern recognition, and machine learning, and concerned with secondary analysis of large data bases in order to find previously unsuspected relationships, which are of interest of value to their owners.*" (http://www.alz.washington.edu/NONMEMBER/DATA2000/GERALD1/tsld003.htm [Referenced 10.04.2006])

Taking a narrow view, data mining usually involves large data repositories and utilizes a collection of tools and techniques to discover something from the data. However, data mining can be seen as an activity where actions are based on learning, decisions are informed, and measuring the results is found beneficial. In other words, data mining also refers to the process of applying the tools and techniques. Some people say that KDD is the process while data mining refers to the methods that may be applied during the process. To apply data mining, we have to be able to observe and collect data, learn from the collected data and base actions, and act on what was learned. See Klösgen & Zytkow [2002b].

5.5.2 Why data mining?

The main motivation for data mining is to efficiently gather knowledge, and then use the knowledge. At the moment, we have many knowledge discovery tools available. The knowledge can be used for *exploration*, *description*, *prediction*, *optimization* and *explanation* [Klösgen & Zytkow 2002b, Berry & Linoff 2004].

In the *exploration*, weak knowledge is used to take the first look at the data. This may give hints on dependencies between variables, thus leading to descriptions with further data mining efforts. For example, we may not yet know which variables influence the target variable and by exploration we gain background knowledge on the problem. Often, the data mining process starts with exploration and, after the initial results, the target is set to get descriptions and predictions.

In the *description*, knowledge presents dependencies among variables. Note that this is not the same as prediction, but rather an empirical equation, an understandable overview of components and their lifecycles, etc. For instance, we may have a complete but easily understandable description of the benefits and nuisances a product modification causes for different customers.

In the *prediction*, the knowledge can be used to make future forecasts, like estimating the time a component will work. Another use is to predict a property of a case, such as to classify a component as broken (e.g., based on variables that do not directly measure the component). Sometimes, confidence intervals can be given, how accurate the classifications are.

The *optimization* here is an application of the knowledge to seeking the best solution to a combinatorial optimization problem. The knowledge could be derived from a database containing many examples of production events. We may not know the exact function that describes the quality of a process or product in the terms of parameters. This feature distinguishes the optimization usage of data mining from mathematical optimization.

In the *explanation*, deep knowledge (at the level of principles) is used to derive a description that applies to a class of special situations. This is usually difficult to achieve.

Often, conventional data analysis methods can be applied too, but sometimes data mining gives better results. Data mining works well when the problem has a large number of potentially relevant variables, when data mining is applied to multidimensional relations that vary in different subpopulations, when no statistical model has been made, and when surprising results can be expected for subpopulations. Often, conventional data analysis benefits from simultaneous data mining efforts and vice verse.

Typical challenges include very large data sets - that is, a large number of tuples and variables and the data coherence and consistence compared with the real world. For instance, if the data is collected for process control, some of the assumptions needed for conventional data analysis may not hold. Other challenges include help given by external data, problems in a large-scale search through the hypothesis space, sampling versus weak relationships, complex versus simple models, robustness and accuracy, hypothesis space formations, constructive induction (e.g. derived variables), meaningfulness of hypotheses (statistical significance, novelty, simplicity, usefulness) and data mining system integration with other systems. See Klösgen & Zytkow [2002b].

5.5.3 Data mining process

The data mining process can be divided into four phases: business understanding, data transformation into information, action taking and outcome measuring [Berry & Linoff 2004]. Figure 28 shows a data mining process that is divided into four phases. Usually, the process starts when a need is recognized, such as gaining an understanding of the demographic differences in the ways the customers use machines (our products). Another target could be to derive knowledge on how the weather conditions affect the maintenance of the machines. To be able to transform data into information, we have to be able to collect the data. It may take time to find the relevant attributes to be stored in the databases. (See also [Klösgen & Zytkow 2002c].)

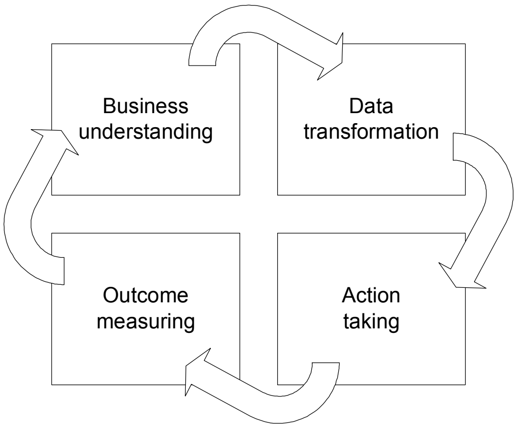


Figure 28. Data mining process.

In this example we need data typically collected from customers, data about locations where the machines are used, and data collected from the machines in use. The data is classified based on location, customer and malfunction type. Moreover, survival analysis is applied to derive typical lifetimes for components for given customer and location types. This is essentially similar to the remaining-useful-lifetime - that is, in RUL analysis. Based on this derived information, it is possible to improve the design of the machine for the environments that are both difficult and important. At the same time, the classification gives insights into the user groups and, together with the survival analysis, we are able to find the components of the machine that require additional work. These findings help to make the maintenance more cost-efficient, which is assessed in the last phase. After these phases, everything starts again. Now we want to transform the survival analysis into a predictive maintenance process to obtain quick alerts on the components showing symptoms of breaking in the near future.

Figure 29 shows an example of two survival curves. They show the likelihood that one the components will be still useful at a particular point in time. These curves represent the same components but from different vendors. The curves can be constructed from retention curves, which are, in turn, basically cumulative histograms. There are several ways of forming survival curves. We are now able to measure the difference between two components, and here the different component vendors form our initial conditions. The initial conditions could be based on different maintenance operations for the same fault, on different locations, customers, etc. This approach is called *stratification*. See Berry & Linoff [2004].

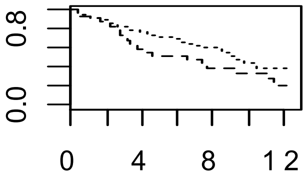


Figure 29. Survival curves for survival analysis.

We may add more phases into the above process description to further describe the data mining activities. These could be data selection and data understanding, model set creation, data problem fixing, modelling, evaluation and deployment. Each phase has own objectives, work items and deliverables. The business understanding phase includes determining the business objective, assessing the situation, determining the data mining goals and making a project plan. Data understanding includes collecting the initial data, describing the data, verifying the data quality and exploring the data. See Reinartz [2002].

Model set creation and data problem fixing includes describing the data set, selecting, cleaning, constructing, integrating and formatting the data. In the modelling we select the modelling technique, generate the test design, build and assess the models. The evaluation phase means evaluating the results, reviewing the process, listing possible actions and making decisions. Deployment may include making a deployment plan, maintenance plan, final report and project review. See Reinartz [2002].

5.5.4 Methods used in data mining

Some of the methods presented in Chapter 5 can be used when mining data. For a wider introduction to the different techniques, see Klösgen & Zytkow [2002] and Berry & Linoff [2004].

Sometimes, the application of certain techniques is either required, like data warehousing (data cleaning and administration), or almost essential, which is the case with data reduction, if there are large amounts of data available. Different data reduction techniques include sampling, feature selection, feature aggregation and discretization of numerical attributes. See Klösgen & Zytkow [2002, pp. 205–225].

Exploration activities include different visualization techniques, like interactive statistical graphics and different animation techniques. Spatial and demographic data can be explored using geographical information systems. See Klösgen and Zytkow [2002, pp. 226–253, 409–417 and 509–523].

In the classification, a method is given a pre-defined, usually well-defined, set of classes, and then the method should assign an object to one of the classes. To do the classification, we may apply decision trees, decision rules, Bayesian classification, nearest-neighbour methods, regression, neural networks (several models) and multicriteria classification methods. Clustering is related to classification, except that the pre-defined classes are not available. In clustering, a heterogeneous population is segmented into homogeneous sub-groups or clusters. Approaches to clustering include numerical and conceptual clustering, and the algorithms can be classified into partition-based, density-based and hierarchical clustering algorithms. Typical methods include discriminant analysis, decision trees, decision rules, Bayesian classification, nearest neighbour methods, minimal consistent sets, Monte Carlo Sampling, neural networks (several models), rough sets, etc. See Klösgen and Zytkow [2002, pp. 254–442].

Figure 30 shows an example of a decision tree. Each node has a condition that is used to divide the tree into sub-trees. The left branch is for false condition and the right for true. Finally, leaves present the partition of the data into classes. From the tree we obtain decision rules that are easy to interpret by specialists. If the classes are not known in advance, we may apply clustering. The right side of the figure presents clusters formed with SOM (self-organizing map), which, in this example, provides a non-linear mapping of data into two dimensions. The size of a hexagonal presents the number of similar objects found from the data - that is, they tell the densities. By looking the map we may conclude that there could be 5–9 classes available, which can further guide our classification efforts.

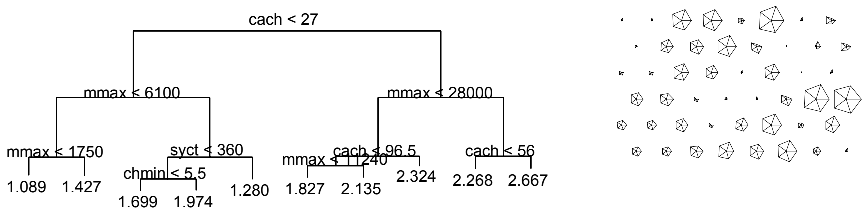


Figure 30. Classification and clustering examples.

Another set of methods is related to rule discovery. Decision trees and decision rules naturally belong here. Rough sets, characteristic rules, association rules and methods based on inductive logic programming can also be applied.

In the subgroup discovery, we may do deviation analysis or change analysis, or apply drill-down methods. Equation fitting, in turn, includes different equation finders and multidimensional regression analysis. Other methods include spatial and demographic analysis (spatial clustering, spatial classification, spatial characterization), and probabilistic and causal networks. See Klösgen & Zytkow [2002, pp. 254-442].

5.5.5 Data mining time series

In this section we pick up a few methods suitable for time series data mining. In the next section we have an (artificial) example and some of the methods presented there can also be applied to time series. Many of the above methods can be applied to time series analysis. Time series analysis is interesting from the viewpoint of process control, predictive maintenance and the like. Time series data mining includes special issues on representing time series, indexing and retrieving, and detecting changes, and on classifying time series. See Last, Kandel & Bunke [2004].

For example, the classification of a time series is a more demanding task than the classification of "static" objects, since the time series (or a sequence of data points) is a complex object when handled as a single object. It is possible to handle this conventionally: first, there could be some preprocessing and then some classification method like a decision tree. Anyhow, the classification can be hard to understand and the preprocessing is often problem-dependent. By considering the time series directly it is possible to utilize temporal concepts that are easy to understand, like permanence in a region for a certain time or temporal literals (like *increases* or *always* in the region). Note that the time series used for building the classifier and used in the classification can have different lengths and partial "examples" are possible. In this way it is possible to classify the time series before the event producing the time series has finished. See Gonzalez & Diez [2004].

Keogh et al. [2002] show how to find surprising patterns in time series. They start by defining a pattern to be surprising if the frequency of its occurrence differs substantially from that expected by change, given some previously seen data. Note that they are not requiring an explicit definition of surprise. The method is given a collection of observed data that is considered normal, and newly observed data is considered relative to this data collection. Further, the surprise of a pattern is not tied exclusively to its structure but rather to the departure of the frequency of the pattern from its expected frequency.

The outline of the method is as follows: first, the data is discretised to use a finite alphabet so that each symbol is equiprobable. To do this, the user has to select the feature window length and to provide a method of extracting features from the feature window. These features are coded in the given alphabet as the time series is processed. In the next step we build up suffix trees from the discretised time series; the suffix trees are used to construct Markov models. The stationary and transition probabilities of Markov models are used to find the probabilities of occurrences of patterns and how surprising they are. Last, when we have the probabilities, we compare them with the user-provided threshold value and the samples exceeding the threshold will be shown to the user.

A benefit of the above method is the versatility of the definition of "surprising pattern". Further, it is straightforward to apply the method as there are only a few parameters the user has to provide and the user-provided "feature extraction from window" method is often natural in the problem context.

If the period is unknown (a period has to be less than the length of the feature window in the previous method), we may follow Elfeky et al. [2004]. They also start by discretising the time series: This time the alphabet is set, and it is binary. Modified convolution is applied on the binary vector, and the resulting values are analyzed to determine the symbol periodicities and the periodic single-symbol patterns. After this, a set of candidate periodic patterns is formed and support for each pattern is estimated. The method is reported to find periodic patterns efficiently, and it tolerates certain types of noise well. However, this method has difficulties when the length of the period is not stable over time. In the next section we will present another method to overcome the difficulty of obscure periods occurring in real-world time series.

5.5.6 Example: Anomaly detection

Data mining has been applied to widely different fields. Here we present a general approach to anomaly and misuse detection. Figure 31 shows how anomaly detection relates to the OSA-CBM architecture; in this case health assessment and automatic decision reasoning levels are involved and, of course, we suppose that the lower layers supply data and that the seventh, the representation layer, is also used. Prognostics can easily be added into this example. We speculate with this later.

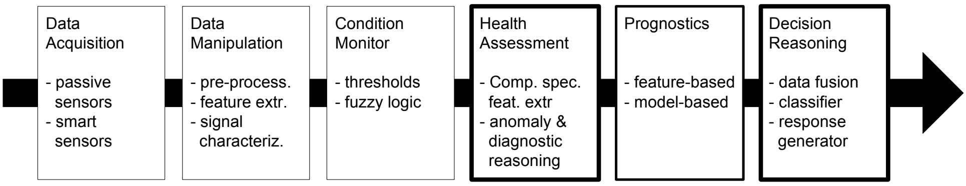


Figure 31. Part of the OSA-CBM architecture description (adopted from Lebold et al. [2003]).

Our “business problem” is to identify faulty components, and to recognize any causal relations between events or between patterns observed from data that indicate anomalies. We have microphones on the devices and signals from several machines can be collected. At the same time, we may collect data on how the machine is used, including user actions and machine state information with time-stamps.

Figure 32 presents an overview of a base for a system to be used in anomaly and misuse detection. This follows the OSA-CBM architecture, including the data repositories. The detection modules belong to the health assessment layer, while the data and knowledge repositories are the glue needed for making decisions. Data and knowledge can be used to generate new knowledge automatically with methods for discovering fuzzy rules, for example [Luo & Bridges 2000]. They are also used to make inferences and later decisions (e.g. alerts) based on the current observed behaviour of the system.

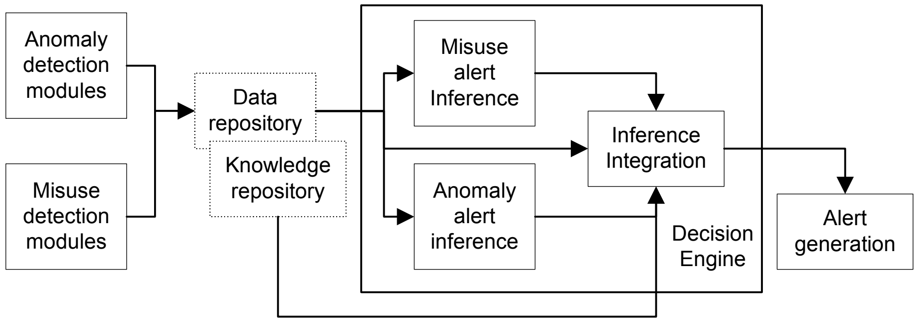


Figure 32. An architecture of anomaly and misuse detection system (adopted from Siraj et al. [2001]).

Siraj *et al.* [2001] describe how to use a fuzzy cognitive maps (FCMs) to support the decision making when detecting anomalies and misuses in a network. We mimic their classification example, but in a different environment. Anomaly detection modules detect the abnormal behaviour of a component in a machine. We want to recognize when a component is breaking in a machine by analyzing a sound signal from a microphone. Anomaly and misuse detection modules have several FCMs for different types of "suspicious events" - that is, for different kinds of sound patterns. These patterns are formed in the Data Manipulation layer. The anomaly detection module uses the pattern data and triggers the events.

Figure 33 shows a simple example of how FCM is used to trigger an event. Specialists and their knowledge are used to construct these kinds of FCMs. In this example FCM is used to identify the component breakdown by observing the sound signal (noise) and the machine state (e.g., orientation, position, velocity, acceleration of a component) during an operation. The activation of the event "Noise_Same_state_Same_machine" can be captured with a one fuzzy rule:

If *duration of operation* is *long* **and** *duration of noise* is *short* **and** this happened for the *same machine* **and** *during the same operation (state)* **then** *Noise_Same_state_Same_machine* is activated *highly*.

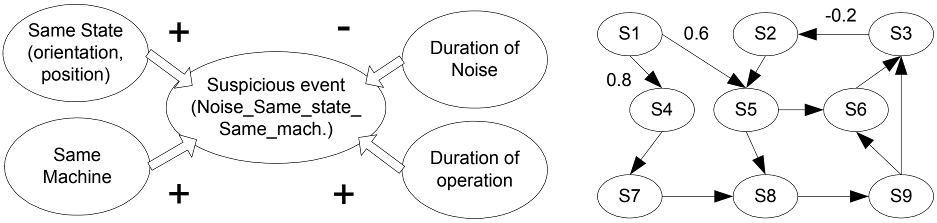


Figure 33. Example FCM for capturing suspicious events - that is, capturing the concept "Noise_Same_state_ Same_machine". The right side represents a network of concepts and a few weights on how concepts affect each other.

The nodes of the FCM correspond to different concepts and the edges denote the causality between the concepts. We could have several rules for one FCM. In this example the inverted influence of the duration of the noise is used because, occasionally, the environment generates noise lasting longer and we have several FCMs for capturing other types of noise patterns as "suspicious events". We combine concepts related to the "suspicious events" from different FCMs at the inference modules into the "main" FCM, in which the individual concepts have their own weights when generating the alert levels on the inference processes. Figure 33 depicts a network of concepts. The structure resembles recurrent neural networks. For details, see Siraj *et al.* [2001].

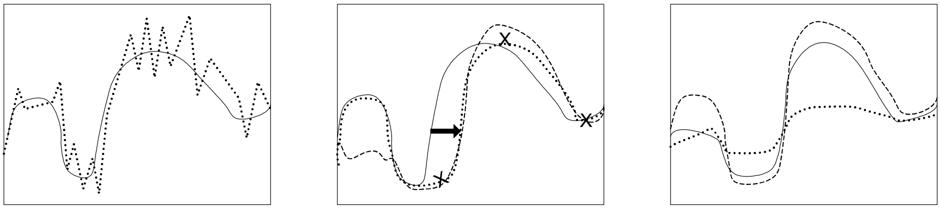
The above method is suitable when there is expertise available. FCMs are simple, and are thus easy to build and deploy. In addition to fault detection, FCMs can be used to model and simulate complex systems, control processes, etc. Aquilar [2005] provides a survey.

However, if we do not have expertise - for instance, if we want to deepen our knowledge of the typical patterns that occur in the sound signals - we have to revert to other methods. This can also be seen as work for the data manipulation layer (or as a second iteration through data mining process). For example, we may wish to determine the main clusters of typical sound signals. Furthermore, we may have a lot of data available that is related to a large set of components and we want to identify the patterns occurring over time. In the rest of this section we first discuss an approach to analyzing signals, and then discuss a method of classifying time series.

Ramsay & Silverman [2002] analyze different time series by applying functional data analysis. They give methods and expose a way of thinking on how to (effectively) represent time series (or functional data), to smooth them, to investigate variability and mean characteristics, and to build models. In functional data analysis, conceptually, functional data are continuously defined; the individual datum is the whole function and smoothness or other regularity is often a key aspect of the analysis.

Let us assume that we have observed the sound signal from 50 devices, repeating the measurement 20 times on similar usage (state, condition). First, we want to find the mean characteristics for individual devices and then compare the devices. Figure 34 (a) shows a signal for one device. We are mainly interested in the larger "hill". Our first action is to *represent* the curve (signal) in an appropriate base, and we use B-splines in this case. After that, we *smooth* the curve - that is, we construct a new curve that approximates the old one and in which the curvature (the second derivate of the curve) is penalized a bit.

When every curve is smoothed, we *register* them. This is shown in Figure 34 (b). The curves may vary in *phase* or in *amplitude* and in this example we are not interested in phase because we think that user actions have little influence on it. Thus the solid line is registered to the *landmarks* marked with Xs, shown as a dotted line. Now we see that there is a small variation between the dotted and dashed lines in amplitude in the region of interest. The aim is to construct a set of curves that only vary in amplitude for further analysis. Registration is therefore a time transformation, called *time warping* in the literature. (The use of landmarks is not always necessary.)



(a) (b) (c)

Figure 34. Smoothing, registering, and applying functional principal component analysis.

Last, Figure 34 (c) shows a mean of the individual signals for one device in the solid line. Further, the other lines are the two first *smoothed functional principal components*, explaining part of the variation for this particular device. An interpretation could be that the dotted line is for devices that are "normal" or fully functional while the dashed curve is for machines that work somehow abnormally. Similarly, we carry out analysis between different devices. Figure 34 (c) could represent this too.

Note that we could have carried out principal component analysis normally and smoothed the results, while Ramsay & Silverman [2002] present a smoothed functional principal component analysis technique with reasons why their approach should be followed. (This example is modified from one given by Ramsay & Silverman [2002]. The data and curves are not from any real application.)

Now, as we have obtained our first insights into the data, we are ready to apply other data mining techniques. Bengtsson *et al.* [2004] show how to detect faults in the gearboxes of industrial robots by comparing sound signals and by using case-based reasoning (CBR). Figure 35 shows the architecture needed for this system. After the feature vector has been assembled from the sound signal, it is matched with the vectors in the case library. Bengtsson *et al.* [2004] use the nearest neighbour algorithm as a similarity measurement.

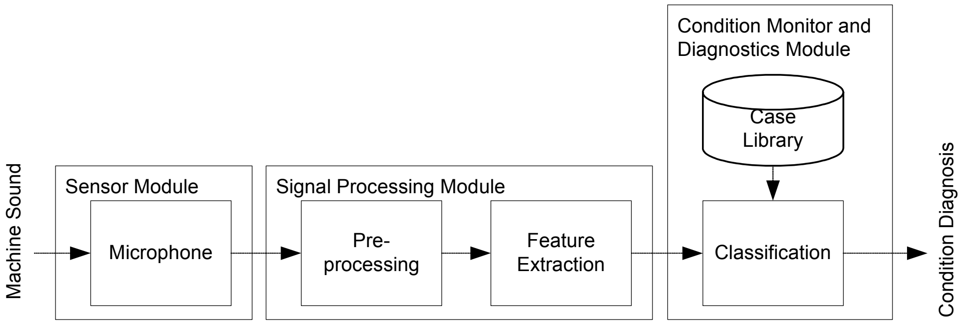


Figure 35. Architecture for case-based fault diagnosis system (from Bengtsson et al. [2004]).

When the case library contains both faulty and healthy cases the similarity measure finds feature vectors that are analogous to the new sample. Earlier samples divide the space of all samples, and as more samples are collected the system learns and the performance increases continuously. If a sample does not fit into the existing set of samples, the sample can be shown to a specialist and can then be added into the database. Other benefits reported by Bengtsson [2004] include: the method is easily accepted by engineers; it transfers experience, it is possible to make manual comparisons between sounds and the system does not have to be complete from start. When applying CBR we have to choose an appropriate set of training records, choose an efficient way of representing the records, and choose the distance function.

If we do not have specialists available to do the initial classification of devices into faulty and healthy, we have to resort to other methods. We want to quickly recognize surprising and meaningful patterns from each sound signal. Further, we want to recognize surprising and meaningful patterns from a collection of sound signals so that we will be able to see how the patterns change over time for a particular device. There might be typical patterns over time that can be used in prediction. The approaches presented in Section 5.5.5 can be applied. Other recent approaches to time series data mining are provided by Aref *et al.* [2004] and Pei *et al.* [2004].

5.6 Combinations of various methods and techniques

Because some methods are better suited to fault detection and others to fault diagnosis it is sometimes useful to combine several different methods. Some methods also work better with certain applications than others. Some typical ways of combining different methods are represented next.

5.6.1 Neurofuzzy networks

Neurofuzzy networks try to combine the advantages of artificial neural networks and fuzzy logic. This is a rather new field of study since little effort was put into research prior to the early 1990s. Different kinds of approaches to combining these two techniques can be used. One common way is to use fuzzy logic in the learning algorithm of the neural network. The idea is to use fuzzy rules in updating the weights of the connections. Another common approach is to use fuzzy neurons in the network. These neurons have some or all of their components and parameters described by fuzzy logic. Neural networks can also be used to process the data before a fuzzy model. This helps to reduce the size of the model, which can often be a real problem if the number of inputs is large. The previously described dimension reduction techniques, especially PCA, can also be used to pre-process the data in fuzzy models [Koivo 2004].

5.6.2 Fuzzy logic and analytical methods

Fuzzy logic can also be used in combination with analytical methods. Most analytical methods, including those described in Chapter 5.3, use residuals for FDI. Fuzzy logic can be used to transform the residuals into qualitative knowledge - i.e. fault indications. First the residuals are fuzzified into fuzzy sets. Different faults can then be identified by IF-THEN rules. Another approach is to use fuzzy logic in determining the threshold used for the residuals. Due to noise and other disturbances, the residual is not zero, even if faults have not occurred, and some threshold for the residual must be used. If the residual is too small, false alarms will occur. On the contrary, too high a residual will leave some faults undetected. With fuzzy logic the threshold can be adjusted to the situation at hand.

5.6.3 Neural networks and expert systems

Neural networks can be used in expert systems to serve as the knowledge base of the expert system. This can improve the data acquisition, which often is the bottleneck in expert systems. On the contrary, expert systems can also be used to improve neural networks. As the results from neural networks are quantitative and possibly hard to understand for humans, expert systems can be used to interpret these results and execute fault diagnosis. Retraining the neural network for new situations is another way of using expert systems with neural networks [Chiang et al. 2001].

6. Establishing sensor fault modes

In order to develop comprehensive diagnostic services for components (like sensors and actuators) and sub-systems, we need to know the relevant fault modes of the devices or sub-systems under diagnosis. In the following we will provide an example of how to establish the fault modes of sensors with the special cases of potentiometer sensors, encoders and proximity switches.

6.1 General fault modes

A typical sensor system consists of a sensor element with internal or external signal conditioning (and possibly an internal communications interface) and a programmable microcontroller, which involves software to read the value from the digital output interface of the signal conditioning electronics and to convert the binary value to engineering units. Such a sensor system may introduce faults in different phases along the signal path from the physical interface to the application process. Figure 36 illustrates the potential points of faults.

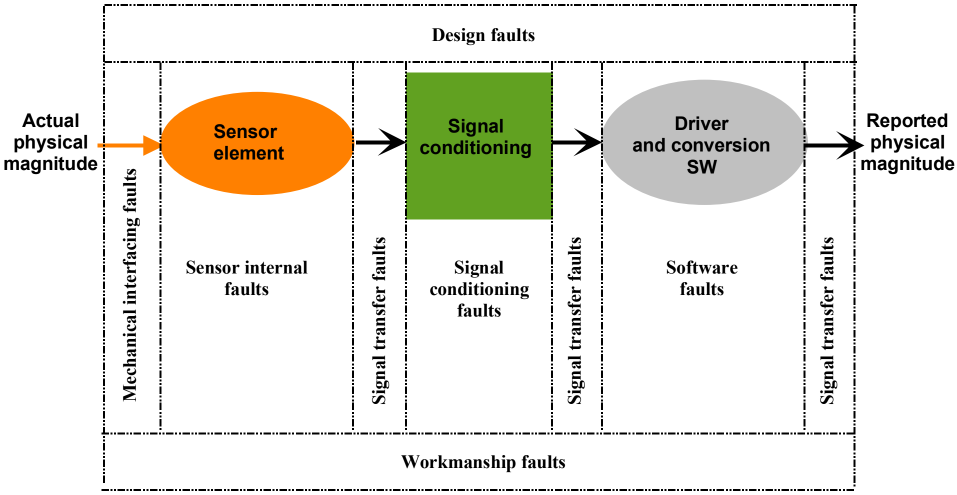


Figure 36. Potential points of faults in a sensor system.

As illustrated in the figure above, the following faults are possible (Table 6):

Table 6. Fault categories.

Fault category	**Fault type**	**Remarks**
A	Design faults	
B	Mechanical interfacing faults	E.g. the sensor is accidentally detached from the physical object
C	Sensor internal faults	
D	Signal transfer faults	Short-circuits, breaks or noise in signal wires; communication errors
E	Signal conditioning faults	Typically, hardware faults
F	Software faults	
G	Workmanship faults	During manufacturing, packaging, shipping, storage, installation, operation and maintenance; includes software update faults and calibration faults

The interface between the physical transducer and the rest of the system varies. For a simple sensor element, like a potentiometer sensor, the sensor wires are attached to the controller module, which includes the signal conditioning electronics. Some transducers include transmitters to convert the measured magnitude to a standardised voltage or current range. For more sophisticated sensors, a communication protocol is applied to transmit the measured value in a digital format. In the latter case the sensors are normally more intelligent in other respects as well, including, e.g., linearization curves and diagnostic features. In the case of an intelligent sensor, all the ingredients depicted in Figure 36 (sensor element, signal conditioning and software manipulation) are included in the transducer.

Any fault in any of the fault categories presented in Table 6 may cause deviations in the reported physical magnitude. We may here apply the guide words presented in the hazard and operability study (HAZOP) standard IEC 61882 [2001] to express the set of relevant deviations in this context (see Table 7).

Table 7. Possible deviations in the reported physical magnitude.

HAZOP guide word	Example interpretation of the deviation in the context of a sensor signal
No	Signal is stuck at a fixed value, like ground potential; or, in a case of data communication, no messages arrive
More	Signal value is higher than the real physical magnitude
Less	Signal value is lower than the real physical magnitude
As well as	Signal includes noise or other accidentally modulated signals
Part of	Intermittent signal
Other than	Signal value belongs to a sensor other than expected
Reverse	The absolute value of the signal is correct, but its sign is opposite to the reality
Early	Signal comes too early
Late	Delay in signal
Before	Signal events occur in the wrong order
After	Signal events occur in the wrong order
Excessive variations[1]	Excessive noise (value domain); excessive jitter (time domain)

[1] This guide word is not presented in the standardised set of HAZOP guide words.

Thus we have two views of the sensor system problems: sensor system fault modes and sensor signal deviations. When planning the diagnostics methods for the sensor types under study, both views may be useful in finding effective diagnostics methods. For simple sensor elements with no transmitters or digital communication interface, a fault mode analysis may provide a better way oof finding effective diagnostics methods, whereas in the case of sensors with transmitters or digital communication interface a deviation analysis may be more suitable.

The list of deviations in Table 7 is generic and is applicable to all the sensor types under study. The fault modes, however, are more sensor type-specific, at least for categories B and C (see Table 6). In the following three chapters the fault modes of potentiometers, encoders and inductive sensors are established. The fault modes presented below mainly include faults from categories C and D.

6.1.1 Potentiometer fault modes

The typical potentiometer fault modes are the following [Ernsberger & Kordecki 1992][Anon. 1995]:

- Excessive contact resistance between the wiper and the track
- Other wear-out or contamination phenomena causing noise and parameter drift
- Breaks in the contact points between the end terminals and the track, as well as between the wiper terminal and the wiper
- Packaging failures (e.g. with oil-filled potentiometers the oil may spill out).

The packaging failures can be regarded as a root cause of some of the actual fault modes, such as noise and wear-out. Hence packaging failures are not included later in the list of fault modes.

In the machine directive harmonised standard EN ISO 13849-2 (Safety of machinery - Safety related parts of control systems - Part 2 Validation; formerly known as prEN 954-2) the following fault modes of potentiometers are listed:

- "*Open-circuit of individual connection*
- *Short-circuit between all connections*
- *Short-circuit between any two connections*
- *Random change of value: $0{,}5\ R_p < R < 2R_p$, where R_p is the nominal value of resistance.*"

Furthermore, an organisation called AREMA (American Railway Engineering and Maintenance-of-way Association) that makes specifications for train applications lists the following potentiometer failure modes [Anon 2003]:

- Open
- Short
- Resistance increase over plus tolerance, to open
- Resistance decrease under minus tolerance, to short

- Increase in contact resistance of slider
- Change in division ratio.

Contact breaks between the wiper and the resistive element can be intermittent, caused by shocks and vibration. The reference [Antonelli et al. 1999][9] estimates such a discontinuity as lasting 0.1 ms or longer.

In the reference [Ernsberger & Kordecki 1992] the authors (from CTS Corp.) report the analysis of field returns from automotive throttle position potentiometric sensors: 87.6% of the field return sensors were found to fall into the no-trouble-found category; packaging and other failures comprised 11.2 % of the failures, and only 1.2% of the failures indicated element wear and electrical noise. Hence the authors concluded that excessive contact resistance is a minor problem. Instead, the authors suspected internal and external termination contacts to be the major cause of the field replacement of the sensors that are later judged to be no-trouble-found items in the failure analysis laboratory. However, the authors from CTS Corporation could not estimate how many of the no-trouble-found items really had terminal contact problems; it is quite possible that most of the no-trouble-found sensors were replaced in vain. Nevertheless, if we consider a 'worst case' scenario with all no-trouble-found items to be unnecessary replacements, the percentage of element wear and electrical noise fault modes only rises to about 10%.

In the reference [Saitoh & Osada 1991] the authors (from ALPS Electric Co. Ltd.) report the operational life and dither tests of their newly (in 1991) developed potentiometric sensors, which indicated the same results as Ernsberger & Kordecki [1992]; the contact resistance increase is a minor problem. During the operational life test of 10 million sliding cycles the rise in the contact resistance, compared with the initial contact resistance (which was about 2% of the total resistance), is negligible. The authors claim that contact resistance values below 15% of the total resistance are tolerable. The operational life tests up to 40 million cycles and the dither cycle test up to nearly 10^9 cycles indicated that the contact resistance value does not even cross the 10% line much.

[9] The particular reference derived the information from a precision potentiometers catalog #752 of Vernitron Motion Control Group

As a summation, the CTS and ALPS studies suggest that excessive contact resistance is a minor problem. Note also that the fault modes listed by EN 13849-2 do not include excessive wiper contact resistance. However, we hypothesise here that the variance of the contact resistance value may provide information on the quality of the resistive track.

The reference [Anon. 1995] supplies statistics on potentiometric sensors as well as normal rotary potentiometers. The statistics on potentiometric sensors are assembled from only eight fault cases and are thus neglected here. The statistics on normal potentiometers report 38.2 % of the faults to be unknown; out of specification takes the second position with 21.5%; wear-out, noise, contamination and corrosion problems take 13.2 % of the share; open contact faults have 4.5%, and the other faults take the rest, except for a fault category 'broken', which takes 17.7% of the failure distribution. This latter category is non-informative.

As a consequence, it is expected here that most of the field returns, let's say 40% - 90%, fall into the no-trouble-found category. One of the goals of the Kodie project was to minimise this number. This can be achieved by providing the service personnel with reliable information on the health of the sensors. Another conclusion is that intermittent contact problems are most probably the main fault mode to be focused on; monitoring of excessive contact resistance does not contribute much to the health assessment of a potentiometric sensor, although, depending on the measurement method, it may provide information on the other fault modes as well.

The fault modes supplied above mainly include only the electrical fault modes. Mechanical faults are another fault mode category to be considered. Such faults include shaft misalignment, slipping of the shaft mounting and bearing wear-out or contamination.

The following figure (Figure 37) and the subsequent table (Table 8) establish the relevant electrical and mechanical fault modes in potentiometers.

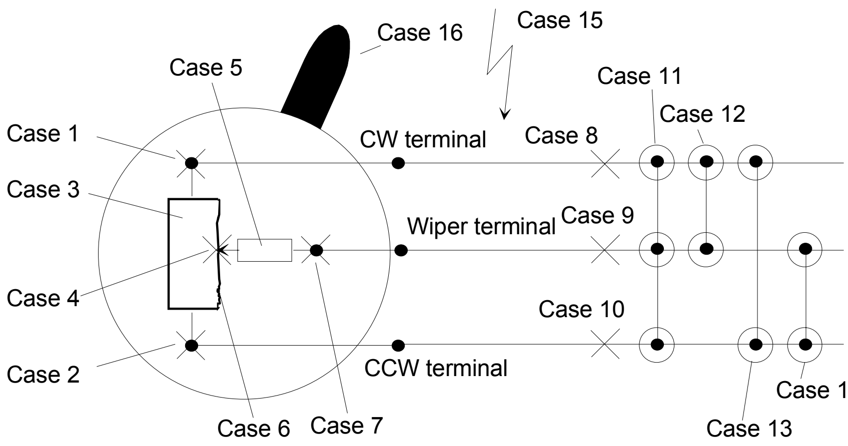


Figure 37. Potentiometer electrical and mechanical fault modes.

Table 8. Potentiometer electrical and mechanical fault modes descriptions.

Case 1	Clockwise (CW) terminal internal break
Case 2	Counter-clockwise (CCW) terminal internal break
Case 3	Change in the resistance value of the resistance element
Case 4	Intermittent break due to vibration or shocks
Case 5	Excessive wiper contact resistance
Case 6	Noise due to wear-out, contamination, etc.
Case 7	Wiper terminal internal break
Case 8	CW terminal external break
Case 9	Wiper terminal external break
Case 10	CCW terminal external break
Case 11	External short-circuit of all terminals
Case 12	External short-circuit of CW and wiper terminals
Case 13	External short-circuit of CW and CCW terminals
Case 14	External short-circuit of wiper and CCW terminals
Case 15	Noise induced in wires
Case 16	Mechanical problems with the shaft (no movement, slippage, etc.)

6.1.2 Encoder fault modes

In the case of a normal quadrature encoder, four wires are needed - power supply, ground, channel A and channel B wires - whereas in the case of a sensor with RS 422 interface, the channels A and B are differential outputs with A+ and A- wires, as well as with B+ and B- wires. However, in this case the encoder under study is a normal quadrature sensor with single-ended outputs. Furthermore, the particular sensor types facilitate totem pole outputs with active pull up and pull down.

Encoder fault modes include the typical open-circuit and short-circuit fault modes. Furthermore, the power supply voltage may be incorrect or noisy. Of course, channel A and B lines may be noisy as well, also including spikes and blips. Internal fault modes include grating cracks and fractures, LED faults and other electronic faults, which may cause unwanted pulses, intermittently lost pulses or total loss of pulses on either of the two channels, or the planned 90º phase shift is distorted. External mechanical faults include antenna slippage, for example.

The reference [Anon. 1995] presents fault mode statistics on thirty field return encoders. For thirteen of them, the fault mode is unknown; for eleven of them 'no movement' or incorrect antenna rotation is reported, indicating mechanical faults. Other fault modes include cracks, design faults, loose casing, optical assembly fault, resistor failure and incorrect marking, each with a single fault case.

The relevant fault modes of an optical quadrature encoder are presented in Figure 38. Single-ended channel lines are assumed. Short-circuits between more than two wires are not included.

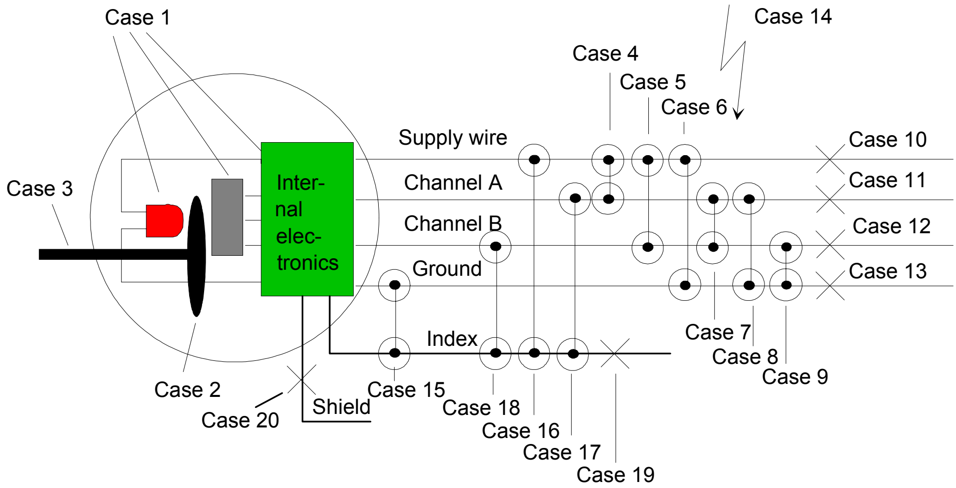


Figure 38. Encoder electrical and mechanical fault modes.

Table 9 supplies descriptions of the fault cases presented in Figure 38.

Table 9. Encoder electrical and mechanical fault modes descriptions; suggested detection methods.

Case	Fault	Symptom	Detection
1	LED and other internal electronics faults, including noise problems	Decreased current level Missing pulses	Current level measurement Pulse length check
2	Cracks, fractures or contamination in the grating	Increased current Incorrect pulse timing	Current level high Timing check
3	Antenna mechanical problems (no or incorrect rotation, bearing fault)	Missing pulses Overheating	Pulse train check Temperature measurement
4	Channel A short-circuit to supply voltage	Ch A stuck at high	State transition check Current level change
5	Channel B short-circuit to supply voltage	Ch B stuck at high	State transition check Current level change

6	Supply voltage short-circuit to ground	Increased current level	Current level measurement
7	Channel A short-circuit to Channel B	Ch A and B in the same phase	State transition check
8	Channel A short-circuit to supply ground	Ch A stuck at low	State transition check Current level change
9	Channel B short-circuit to supply ground	Ch B stuck at low	State transition check Current level change
10	Supply wire break	Dead sensor	Current level low
11	Channel A wire break	Ch A stuck at low or high	State transition check
12	Channel B wire break	Ch B stuck at low or high	State transition check
13	Ground wire break	Dead sensor	Current level low
14	Noise induced in wires	Short extra pulses	Pulse length check Pulse count check Filtering
15	Index wire short-circuit to supply ground	Index stuck at low	Pulse count check Current level change
16	Index wire short-circuit to supply voltage	Index stuck at to high	Pulse count check Current level change
17/18	Index wire short-circuit to Ch A/Ch B	Too many index pulses	Pulse count check
19	Index wire break	Index stuck at low/high	Pulse count check
20	Shield break	Disturbances	Pulse length check

In this case it may be fruitful to analyse the deviations in the encoder channel A and channel B signals as well. The possible deviations listed in Table 7 are applied to the encoder channel A signal in Table 10 (channel B signal deviations are identical).

Table 10. Encoder channel A signal deviations.

HAZOP guide word	Quadrature encoder channel A deviation
No	No pulses
More	Too many pulses
Less	Too few pulses
As well as	Too many pulses
Part of	Missing pulses
Other than	Channel A mixed with channel B
Reverse	Channel A and channel B pulses come in the wrong order
Early	Channel A pulse edge appears too early compared with channel B pulse edge
Late	Channel A pulse edge appears too late compared with channel B pulse edge
Before	Channel A and channel B pulses come in the wrong order
After	Channel A and channel B pulses come in the wrong order
Excessive variations[1]	Excessive jitter (in time domain) -- the time difference between channel A and channel B pulse edges varies excessively (only applicable in a case of constant rotation)

[1] This guide word is not presented in the standardised set of HAZOP guide words.

6.1.3 Proximity switch fault modes

In the machine directive harmonised standard EN ISO 13849-2 (Safety of machinery - Safety related parts of control systems - Part 2 Validation; formerly known as prEN 954-2) the following fault modes of proximity switches are listed:

- "*Permanently low resistance at output, see EN 60947-5-3 (IEC 60947-5-3)*
- *Permanently high resistance at output*
- *Interruption in power supply*
- *No operation of switch due to mechanical failure*
- *Short-circuit between the three connections of a change-over switch.*"

We develop here a somewhat more detailed fault mode model. Figure 39 depicts the fault model for a three-wire proximity sensor (NPN or PNP).

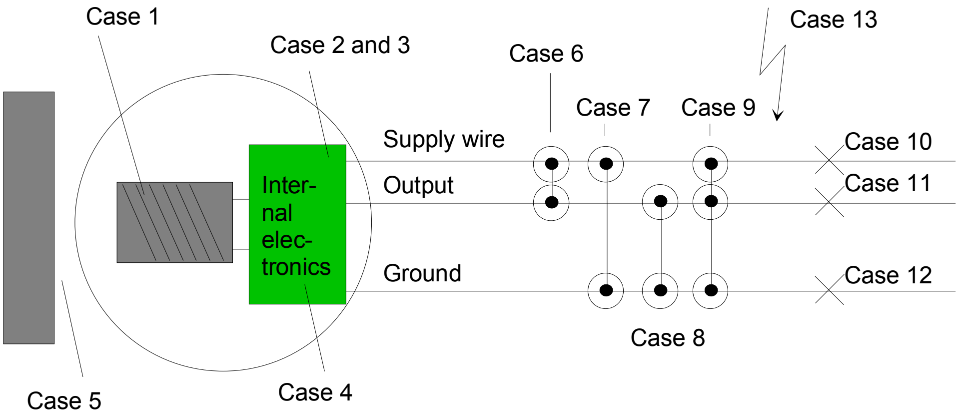


Figure 39. Three-wire proximity sensor fault modes.

Table 11 lists the fault modes depicted in Figure 39.

Table 11. Three-wire proximity sensor fault modes.

Case 1	Coil break
Case 2	Output transistor constant low resistance
Case 3	Output transistor constant high resistance
Case 4	Other internal electronics problems
Case 5	Assembly error (mechanical faults)
Case 6	Output short-circuit to supply voltage
Case 7	Supply voltage short-circuit to ground
Case 8	Output short-circuit to ground
Case 9	All wires short-circuited together
Case 10	Supply wire break
Case 11	Output wire break
Case 12	Ground wire break
Case 13	Noise induced in wires

A deviation model is also provided in Table 12.

Table 12. Three-wire proximity sensor output signal deviations.

HAZOP guide word	**Quadrature encoder channel A deviation**
No	Constant low or constant high
More	NA
Less	NA
As well as	NA
Part of	Intermittent operation
Other than	NA
Reverse	High when should be low or low when should be high
Early	NA
Late	Output level changes after too long a delay
Before	NA
After	NA
Excessive variations[1]	The output level changes at a different distance at different times.

[1] This guide word is not presented in the standardised set of HAZOP guide words.

7. Conclusions

The KODIE project behind this research work showed us that it is difficult to carry out diagnostics-related research projects so that a lot of generic results can be gained. Normally, the diagnostics issues are very application-specific and tightly baked into the actual application software. Hence the best diagnostics solutions are heavily tailored to the particular application or machine type. However, there are issues that are, or at least could be, common to several machine manufacturers, like diagnostics architectures and diagnostics protocols. In this document we have tried to bring forth such things. As a consequence, this document is a mixed bag of machine diagnostics-related issues that could be utilised by several machine automation companies. The most remarkable of these, as we see it, is the idea of a well defined diagnostics strategy that is included in the RAMS specifications that cover reliability, availability, maintainability and safety.

References

Aircraft, 2004. Realtime aircraft health monitoring – Airbus and Boeing approaches, Aircraft Technology Engineering and Maintenance – Farnborough 2004 special.

Anon. 1995. Failure Mode/Mechanism Distributions. Rome (NY): Reliability Analysis Center, 1995. 398 p. (RAC Handbook FMD-97)

Anon. 1999. Control of Emissions of Air Pollution from 2004 and Later Model Year Heavy-Duty Highway Engines and Vehicles; Revision of Light-duty Truck Definition. Environmental Protection Agency. 434 p. (40 CFR Parts 85 and 86.) Available at http://www.epa.gov/otaq/regs/hd-hwy/1999nprm/nprm-4c.pdf [Referenced 02.02.2004]

Anon. 2003. Recommended Practice for Hardware Analysis for Vital Electronic/Software-Based Equipment Used in Signal and Train Control Applications. Unapproved draft of a proposed AREMA Recommended Practice. Available at http://www.arema.org/comm/c37/AREMA%2017.3.3%20FMEA%20MP%20Rev%202.1%20Clean%20Copy.pdf [Referenced 28.10.2003]

Antonelli, K., Ko, J. & Ku, S. 1999. Displacement Measurement, Linear and Angular – Resistive Displacement Sensors. In: Measurement, Instrumentation and Sensors Handbook. CRCnetBase 1999.

Aquilar, J. 2005. A Survey about Fuzzy Cognitive Maps Papers. International Journal of Computational Cognition, Vol. 3, No. 2, June 2005. Pp. 27–33.Bengtsson, M., Olsson, E., Funk, P. & Jackson M. 2004. Technical Design of Condition Based Maintenance Systems – A Case Study using Sound Analysis and Case-Based Reasoning. Proceedings of the 8th Congress of Maintenance and Reliability Conference, May 2nd – 5th, Knoxville, TN, USA, 2004.

Aref, W.G., Elfeky, M.G. & Elmagarmid, A.K. 2004. Incremental, online, and merge mining of partial periodic patterns in time-series databases. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 2, February 2004.

Bardina, J., Follen, G., Blaser, M., Pavlik, W., Zhang, D. & Liu, X. 2000. Integrated Health Management System. https://postdoc.arc.nasa.gov/postdoc/s/document.ehtml?url_id=66975&more=1, [Referenced 13.8.2004]

Berry, M.J.A. & Linoff, G.S. 2004. Data Mining Techniques for Marketing, Sales, and Customer Relationship Management, Wiley Publishing.

CEAC. 1995. Economic Analysis Manual. U.S. Army Cost and Economic Analysis Center. Department of the Army, July 1995.

Chiang L., Russell E. & Braatz R. 2001. Fault Detection and Diagnosis in Industrial Systems. Springer.

Chow, M. 1997. Methodologies of Using Neural Network and fuzzy Logic Technologies for Motor Incipient Fault Detection. World Scientific.

Correcher, A., Morant, F., Garcia, E., Blasco-Gimenez, R. & Quiles E. 2002. Failure Diagnosis of a Cement Kiln using Expert Systems. IEEE 2002, 28th Annual Conference of the Industrial Electronics Society, Vol. 3.

Deb, S., Ghoshal, S., Malepati, V.N. & Kleinman, D.L. 2000. Tele-diagnosis: remote monitoring of large-scale systems. Aerospace Conference Proceedings. IEEE ,Volume 6 , 18–25 March 2000. Pp. 31–42.

Deb, S. & Ghoshal, S. 2001. Remote diagnosis server architecture. AUTOTESTCON (Proceedings). Pp. 988–998.

Deicke, A. 2002. The Electrical/Electronic Diagnostic Concept of the New 7 Series. Warrendale: Society of Automotive Engineers. 4 p. (SAE paper 2002-21-0022)

DaimlerChrysler 2002. Data greetings from COMO, DaimlerChrysler Hightech Report 2/2002.

Elfeky, M., Aref, W.G. & Elmagarmid, A.K. 2004. Using Convolution to Mine Obscure Periodic Patterns in One Pass. Proceedings of EDBT 2004, the 9th Int.

Conference on Extending Database Technology, Heraklion, Crete, Greece, March 2004.

Ernsberger, C.N. & Kordecki, D.L. 1992. Throttle Position Sensor Termination and Connector Reliability. Warrendale: Society of Automotive Engineers. 10 p. (SAE paper 920703)

Gonzalez, C.J.A. & Diez, J.J.R. 2004. Boosting interval-based literals: variable length and early classification. In: Last, Kandel and Bunke [2004]. Pp. 149–171.

Greitzer, F.L., Hostic, C.J., Rhoads, R.E. & Keeney, M. 2001. Determining how to do prognostics, and then determining what to do with it. Autotestcon 2001, Aug. 20-23, 2001, Valley Forge, Pennsylvania.

Helkama 2004. Skoda training material received from Helkama-Auto Oy.

Hossack, J.A, Menal, J., McArthur, S.D.J. & McDonald, J. R. Source. 2003. A multiagent architecture for protection engineering diagnostic assistance. IEEE Transactions on Power Systems, Vol. 18, No. 2, May 2003. Pp. 639–647.

IEC 61882. 2001. Hazard and Operability (HAZOP) studies – Guide word approach. Geneva: International Electrotechnical Commission. 113 p.

Isermann, R. 2004. Model-based fault detection and diagnosis – status and applicatons. 16th Symposium on Automatic Control in Aerospace, 14–18 Juni 2004, St. Petersburg, Russland.

ISO 14229. 1998. Road vehicles – Diagnostic systems – Diagnostic services specification. Geneva: International Organization for Standardization. 50 p.

ISO 13374-1. 2003. Condition monitoring and diagnostics of machines – Data processing, communication and presentation – Part 1: General guidelines. Geneva: International Organization for Standardization. 16 p.

ISO 9141. 1989. Road vehicles – Diagnostic systems – Requirements for interchange of digital information. Geneva: International Organization for Standardization. 11 p.

ISO 9141-2. 1994. Road vehicles – Diagnostic systems – Part 2: CARB requirements for interchange of digital information. Geneva: International Organization for Standardization. 14 p.

Jameel, A., Stuempfle, M., Jiang, D. & Fuchs, A. 1998. Web on wheels: Toward Internet-enabled cars. Computer, Vol. 31, No. 1, Jan, 1998, p. 69–76.

Kaidara 2004. www.kaidara.com [Referenced 7.12.2004]

Keogh, E., Lonardi, S. & Chiu, B.Y. 2002. Finding Surprising Patterns in a Time Series Database in Linear Time and Space. Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02), 2002. Pp. 550–556.

Klösgen, W. & Zytkow, J.M. (eds.). 2002a. Handbook of Data Mining and Knowledge Discovery, Oxford University Press.

Klösgen, W. & Zytkow, J.M. 2002b. Knowledge discovery in databases: the purpose, necessity, and challenges. In: Klösgen and Zytkow [2002]. Pp. 1–9.

Klösgen, W. & Zytkow, J.M. 2002c. The knowledge discovery process. In: Klösgen and Zytkow [2002]. Pp. 10–21.

Kohonen, T. 1990. The self-organizing map. Proceedings of the IEEE , Vol. 78.

Koivo, H. Educational material of course AS-74.115 Neuro-Fuzzy Computing in Automation. http://control.hut.fi/Kurssit/AS-74.115/Material/ [Referenced 25.1.2004].

Kolodner, J. 1993. Case-based reasoning. San Mateo, CA. Morgan Kaufman Publishers. 668 p.

Kurki, M. 1995. Model-based fault diagnosis for mechatronic systems. VTT Publications 223. VTT, Espoo. 116 p.

Last, M., Kandel, A., & Bunke, H. (eds.). 2004. Data Mining in Time Series Databases, World Scientific.

Lebold, M. & Thurston, M. 2001. Open Standards for Condition Based Maintenance and Prognostic Systems. 14 p. Available at: http://www.osacbm.org/reports.html [Referenced 09.02.2004].

Lebold, M., Reichard, K. & Boylan, D. 2003. Utilizing DCOM in an Open System Utilizing DCOM in an Open System Architecture Framework for Architecture Framework for Machinery Monitoring and Diagnostics. Slides for IEEE Aerospace Conference – 9. MAR 2003. Available at: http://www.osacbm.org/reports.html [Referenced 09.02.2004].

Lee, K. 2000. IEEE 1451: A Standard in Support of Smart Transducer Networking. IEEE Instrumentation and Measurement Technology Conference, USA, May 1–4. 2000. Pp. 525–528.

Lee, K., Gao, R.X. & Schneeman, R. 2002. Sensor network and information interoperability – Integrating IEEE 1451 with MIMOSA and OSA-CBM. IEEE Instrumentation and Measurement Technology Conference, Vol. 2, 2002, Pp. 1301–1305.

Lind, R., Schumacher, R., Reger, R., Olney, R., Yen, H., Laur, M. & Freeman, R. 1999. Network Vehicle – a glimpse into the future of mobile multi-media. IEEE Aerospace and Electronic Systems Magazine, Vol. 14, No. 9, Sep, 1999. Pp. 27–32.

Luo, J. & Bridges, S.M. 2000. Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection. International Journal of Intelligent Systems, Vol. 15, No. 8, 2000. Pp. 687–704.

Mattila, M. 2003. Condition Monitoring of an X-ray Analyzer. Helsinki: Helsinki University of Technology. Information and Computer Systems in Automation Laboratory. 71 p. (Report 8.)

McCormick, A.C. & Nandi, A.K. 1996. A comparison of artificial neural networks and other statistical methods for rotating machine condition classification. IEE Colloquium on Modeling and Signal Processing for Fault Diagnosis. Pp. 2/1–2/6. (Digest No: 1996/260.)

Oakland, J.S. & Followell, R. F. 1990. Statistical Process Control: A Practical Guide. Butterworth Heinemann. 431 p. ISBN 043-491-484-3.

Ogawa, T. & Morozumi, H. 2002. Diagnostics Trends for Automotive Electronic Systems. Warrendale: Society of Automotive Engineers. 8 p. (SAE paper 2002-21-0021)

Onstar 2004. www.onstar.com [Referenced 9.11.2004]

Patton, R.J. 1991. Fault detection and diagnosis in aerospace systems using analytical redundancy. Computing & Control Engineering Journal, Vol. 2, Issue 3. Pp. 127–136.

Patton, R.J. & Chen, J. 1992. Robustness in quantitative model-based fault diagnosis. IEE Colloquium on Intelligent Fault Diagnosis – Part 2: Model-Based Techniques. Pp. 4/1–417.

Patton, R.J., Frank, P. & Clark, N. 2000. Issues of Fault Diagnosis for Dynamic Systems. Springer Verlag. 597 p. ISBN 354-019-968-3.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H. Chen, Q., Dayal, U. & Hsu, M.-C. 2004. Mining sequential patterns by pattern-qrowth: the PrefixSpan approahc. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, October 2004.

Ramsay, J.O. & Silverman, B.W. 2002. Applied Functional Data Analysis, Methods and Case Studies. Springer Series in Statistics.

Rauma, T. 1999. Fuzzy modelling for industrial systems. VTT Publications 382, VTT, Espoo. 97 p. + app. 40 p.

Ready, C.M. 1991. Expert Systems for Condition Monitoring, Fault Diagnosis and Control. IEE Colloquium on Condition Monitoring for Fault Diagnosis. Pp. 1/1–1/3.

Reinartz, T. 2002. Stages of the Discovery Process. In: Klösgen and Zytkow [2002]. Pp. 185–192.

Rohmert, F. 2002. Remote maintenance and diagnosis of mobile machinery. Praxis Profiline – CANopen, January 2002, Pp. 2–4.

SAE AS 4893. 1996. Generic Open Architecture (GOA) Framework. Warrendale: Society of Automotive Engineers.

SAE J1850. 2001. Class B Data Communications Network Interface. Warrendale: Society of Automotive Engineers.

SAE J1978. 2002. OBD II Scan Tool. Warrendale: Society of Automotive Engineers. (Equivalent to ISO/DIS 15031-4:December 14, 2001)

Saitoh, M. & Osada, K. 1991. Long Life Potentiometric Sensor – Its Material And Application. Warrendale: Society of Automotive Engineers. 12 p. (SAE paper 910269)

Shiozaki, J., Shibata, B., Matsuyama, H. & O'Shima, E. 1989. Fault Diagnosis of Chemical Process Utilizing Signed Directed Graphs – Improvement by Using Temporal Information. IEEE Transactions on Industrial Electronics, Vol. 36, No. 4. Pp. 469–474.

Simani, S., Fantuzzi, C. & Patton, R.J. 2003. Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques. Springer Verlag. 282 p. ISBN 185-233-685-4.

Siraj, A., Bridges, S.M. & Vaughn, R.B. 2001. Fuzzy cognitive maps for decision support in an intelligent intrusion detection systems. Proceedings of ISFA- NAFIPS-2001, July 25–28, 2001.

Sorsa, T., Koivo, H.N. & Koivisto, H. 1991. Neural networks in process fault diagnosis. IEEE Transactions on Systems, Man and Cybernetics, Vol. 21. Pp. 815–825.

Stepper, M.R., Butler, S.R. & Zhu, G.G. 1995. On-Board Diagnostics, A Heavy Duty Perspective. Warrendale: Society of Automotive Engineers. 15 p. (SAE Paper 951947)

Sycara, K.P. 1998. The many faces of agents. AI Magazine, Vol. 19, No. 2. Pp. 11–12.

Tan, H.-Z. & Sepehri, N. 2002. Parametric Fault Diagnosis for Electrohydraulic Cylinder Drive Units. IEEE Transactions on Industrial Electronics, Vol. 49, No. 1. Pp. 96–106.

Watson, I. 1997. Applying Case-based Reasoning – Techniques for Enterprise Systems. Morgan Kaufmann Publishers. 290 p. ISBN 155-860-462-6.

Wolfram, A. & Isermann, R. 2002. Component based tele-diagnosis approach to a textile machine, Control Engineering Practice, Vol. 10, Issue 11, November 2002, pp. 1251–1257.

Woolridge, M. & Jennings, N.R. 1995. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, Vol. 10, No. 2. Pp. 115–152.

Published by	Series title, number and report code of publication
VTT	VTT Research Notes 2343 VTT–TIED–2343

Author(s)
Alanen, Jarmo, Haataja, Kari, Laurila, Otto, Peltola, Jukka & Aho, Isto

Title
Diagnostics of mobile work machines

Abstract
In this research note, we take a look at the field of mobile work machine diagnostics. The perspective by the authors is limited by the research project (KODIE) behind this research. However, we have tried to provide generic guidelines for machine builders to set up their diagnostics strategy. Building blocks, like SAE J1939/73, ISO 15765 and ODX, from automotive industry are exhibited to prevent machine manufactures from reinventing diagnostics protocols and practices. Furthermore, examples of diagnostics architectures are presented, with OSA-CBM among others. To make the most of diagnostics data, an extensive set of data analysis methods are introduced. And in order to help engineers to design diagnostics feature for the sensor system, hints and examples are supplied as to how to establish the fault modes of sensors; a good knowledge about the (relevant) sensor and actuator fault modes is a prerequisite for comprehensive fault detection.

Keywords
mobile work machines, diagnostics, automotive industry, cars, heavy-duty vehicles, diagnostics architecture, MIMOSA, data analysis, data mining, sensor fault modes

ISBN
951–38–6798–6 (soft back ed.)
951–38–6799–4 (URL:http://www.vtt.fi/publications/index.jsp)

Series title and ISSN			Project number
VTT Tiedotteita – Research Notes 1235–0605 (soft back ed.) 1455–0865 (URL: http://www.vtt.fi/ publications/index.jsp)			E3SU0021
Date	**Language**	**Pages**	**Price**
May 2006	English	122 p.	C

Name of project	Commissioned by
KODIE - Diagnostics and remote control of machines	Tekes, VTT Technical Research Centre of Finland, Plustech, Kalmar Industries, Plenware Oy
Contact	**Sold by**
VTT Technical Research Centre of Finland P.O. Box 1300, FI-33101 TAMPERE, Finland Phone internat. +358 20 722 111 Fax +358 20 722 3282	VTT Technical Research Centre of Finland P.O.Box 1000, FIN-02044 VTT, Finland Phone internat. +358 20 722 111 Fax +358 9 456 4374

VTT Tiedotteita – Research Notes

Tätä julkaisua myy	Denna publikation säljs av	This publication is available from
VTT	VTT	VTT
PL 1000	PB 1000	P.O. Box 1000
02044 VTT	02044 VTT	FI-02044 VTT, Finland
Puh. 020 722 4404	Tel. 020 722 4404	Phone internat. + 358 20 722 4404
Faksi 020 722 4374	Fax 020 722 4374	Fax + 358 20 722 4374

ISBN 951-38-6798-6 (soft back ed.) ISBN 951-38-6799-4 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1235-0605 (soft back ed.) ISSN 1455-0865 (URL: http://www.vtt.fi/inf/pdf/)