# DA2.6: Implementation of the PLM Process model for the Demonstrator

**Written by:**
**Lutz Rabe (BIBA)**
**Cécile Corcelle, Jaime Krull (Caterpillar)**
**Falk Brauer (SAP)**
**Ajith Parlikad (Cambridge IfM)**

| | |
|---|---|
| **DELIVERABLE NO** | DA2.6: Implementation of the PLM Process model for the Demonstrator |
| **DISSEMINATION LEVEL** | **CONFIDENTIAL** |
| **DATE** | 05.06.2008 |
| **WORK PACKAGE NO** | WP A2: PROMISE |
| **VERSION NO.** | V12 |
| **ELECTRONIC FILE CODE** | DA2 6 v13.doc |
| **CONTRACT NO** | 507100 PROMISE<br>A Project of the 6th Framework Programme Information Society Technologies (IST) |
| **ABSTRACT** | This deliverable (DA2.6) summarises the implementation of the PLM process model for the demonstrator, in terms of scenes, PROMISE components and technology implemented, as described in DA2.3 and DA2.4. The motivation for eventual discrepancies is given, together with the detailed results of the activities performed for the implementation. |

| **STATUS OF DELIVERABLE** | | |
|---|---|---|
| **ACTION** | **BY** | **DATE** (dd.mm.yyyy) |
| **SUBMITTED** (author(s)) | Cécile Corcelle | 05.06.2008 |
| **VU** (WP Leader) | Cécile Corcelle | 05.06.2008 |
| **APPROVED** (QIM) | Dimitris Kiritsis | 06.06.2008 |

## Revision History

| Date (dd.mm.yyyy) | Version | Author | Comments |
|---|---|---|---|
| 01.04.2008 | 0.1 | Jaime Krull | Initial structure definition |
| 02.04.2008 | 0.2 | Jaime Krull | Structure modification & data inputs |
| 07.04.2008 | 0.3 | Cecile Corcelle | Modification of all chapters |
| 14.04.2008 | 0.4 | Lutz Rabe | Initial input of technical descriptions |
| 23.04.2008 | 0.5 | Cecile Corcelle | Modification of structure |
| 05.05.2008 | 0.6 | Lutz Rabe | Modification of technical content |
| 21.05.2008 | 0.7 | Falk Brauer | PDKM communication description added |
| 24.05.2008 | 0.8 | Ajith Parlikard | DSS information added |
| 26.05.2008 | 0.9 | Lutz Rabe | Modification of technical aspects |
| 28.05.2008 | 0.10 | Jaime Krull | Minor modification |
| 03.06.2008 | 0.11 | Cécile Corcelle | Add information and refine |
| 05.06.2008 | 0.12 | Cécile Corcelle | Final version with conclusion |
| 05.06.2008 | 0.13 | Lutz Rabe | Final version with conclusion with annex |

## Author(s)' contact information

| Name | Organisation | E-mail | Tel | Fax |
|---|---|---|---|---|
| Lutz Rabe | BIBA | rab@biba.uni-bremen.de | +494212185519 | |
| Cecile Corcelle | Caterpillar | Corcelle_Cecile@cat.com | +33 476 238 213 | |
| Jaime Krull | Caterpillar | Krull_Jaime_A@cat.com | 1-309-578-5778 | |
| Falk Brauer | SAP | falk.brauer@sap.com | +49 351 4811 6138 | |

# Table of Contents

**1 INTRODUCTION....3**

1.1 Purpose of this deliverable....3
1.2 Objective of demonstrator....3

**2 TECHNICAL DESCRIPTION OF THE DEMONSTRATOR....4**

2.1 System architecture and components of the demonstrator....4
*2.1.1 General description....4*
*2.1.2 PEID and CorePAC concept....5*
*2.1.3 PEID and CorePAC implementation....6*
*2.1.4 Device Controller implementation....9*
*2.1.5 Request Handling Layer....10*
*2.1.6 PROMISE Messaging Interface....11*
*2.1.7 Product Data and Knowledge Management....11*
*2.1.8 Decision Support System....15*
2.2 Data structure....16
2.3 Scenes included in the PLM model....20

**3 IMPLEMENTATION OF THE DEMONSTRATOR....21**

3.1 Implementation Step 1: BOL Initialization of PEID & PDKM....21
*3.1.1 General description....21*
*3.1.2 PROMISE components used....21*
*3.1.3 Functionalities demonstrated....22*
*3.1.4 Current and future refinement....22*
3.2 Implementation Step 2: MOL Maintenance Data Up-date on PDKM....23
*3.2.2 General description....23*
*3.2.3 PROMISE components used....24*
*3.2.4 Functionalities demonstrated....25*
*3.2.5 Current and future refinement....25*
3.3 Implementation Step 3/4: EOL of Component / Part Replacement....25
*3.3.1 General description....26*
*3.3.2 PROMISE components used....26*
*3.3.3 Functionalities demonstrated....27*
*3.3.4 Current and future refinement....28*
3.4 Implementation Step 5: DSS Use to make EOL Decision....29
*3.4.1 General description....29*
*3.4.2 PROMISE components used....29*
*3.4.3 Functionalities demonstrated....29*
*3.4.4 Current and future refinement....31*

**4 CONCLUSION....32**

**5 REFERENCES....33**

**6 ANNEX A....34**

6.1 RFID System....34
6.2 Computer maintenance / part exchange activities....35

# List of figures

Figure 1: Configuration of PROMISE Components for the A2 Application Scenario....4
Figure 2: PROMISE Architecture as Implemented for A2 Demonstrator....4
Figure 3: A2 Product structure....5
Figure 4: A2 Components, information sources, and related PEIDs....5
Figure 5: CorePAC Application for initialization of component PEIDs....7
Figure 6: CorePAC application for MOL and EOL processes....8
Figure 7: Intel UPnP browser displaying CorePAC and access by getUPIs and getKeys....9
Figure 8: PROMISE DC GUI displaying CorePAC services and results for a read service....10
Figure 9: Distributed middleware architecture....11

Figure 10: Monitoring and management GUI for the entire Middleware environment .......... 11
Figure 11: PDKM view to Engine and related components .......... 12
Figure 12: PDKM view to Characteristics of an Engine .......... 12
Figure 13: PDKM view to Measuring Points of an Engine .......... 13
Figure 14: PDKM Portal view on engine details .......... 14
Figure 15: PDKM Portal (front-end) view on engine Measuring Points and related mapping to InfoItemIDs .......... 15
Figure 16: DSS GUIs implemented for end-of-life of heavy machinery .......... 16
Figure 17: Mapping of CorePAC, PMI, SOM and PDKM classes and attributes .......... 17
Figure 18: Product data model (SOM / PDKM) .......... 19
Figure 19: Initialization process and related components .......... 22
Figure 20: Connecting ET tool to ECM .......... 23
Figure 21: ECM Connection Kit .......... 23
Figure 22: Cylinder Block .......... 24
Figure 23: Crankshaft .......... 24
Figure 24: Cylinder block .......... 24
Figure 25: Mainenance process and related components .......... 24
Figure 26: PDKM view on the product structure .......... 26
Figure 27: Part exchange process and related components .......... 27
Figure 28: Decision Support System GUI .......... 30
Figure 29: Output interface (DSS) .......... 31
Figure 2: Board Computer .......... 36

# Abbreviations

Abbreviations used in this document:

| | |
|---|---|
| BOL | Beginning of Life |
| BOM | Bill of Material |
| CorePAC | Core PEID Access Container |
| DC | Device Controller |
| DSS | Decision Support System |
| ECM | Electronic Control Module |
| EOL | End of Life |
| ET tool | Electronic Technician tool |
| GUI | Graphical User Interface |
| MES | Manufacturing Execution System |
| MOL | Middle of Life |
| MP | Measuring Points |
| PDKM | Product Data and Knowledge Management (System) |
| PEID | Product Embedded Information Device |
| PMI | PROMISE Messaging Interface |
| PPC | Production Planning and Control |
| RHL | Request Handling Layer |
| SOM | Symantec Object Model |
| SQL | Standard Query Language |
| TTL | Track Type Loader |
| UPI | Unique Product Identifier |

# 1 Introduction

## 1.1 Purpose of this deliverable

Aim of this deliverable is to describe the PROMISE customized solution for the A2 application in term of technologies and methodologies.

Based on previous scenes described in DA2.3, demonstration activities have been conducted and performed to demonstrate information flow between PROMISE components and therefore, the ability of the customized IT system to support the A2 PLM objectives, in term of tracking historical information and decision-making at EOL of an engine or/and engine components.

From what has been demonstrated in chapter 3, we will then be able to specify in DA2.7 further tasks for implementation steps, allowing the technology to be moved from demonstrator scale to large-scale products.

## 1.2 Objective of demonstrator

The main objective of the demonstrator is the automation of historical engine data capture. The customized PROMISE architecture allows the record, up date and analysis of engine components lifecycle by using a RFID system, CorePAC and PMI Middle ware, back-end database called PDKM and Decision Support System.

# 2 Technical description of the demonstrator

## 2.1 System architecture and components of the demonstrator

### 2.1.1 General description

Chapter 2 provides an overview of the overall architecture as depicted in Figure 1 and Figure 2, and related PROMISE components as they are used for the technical realization of the A2 scenarios.

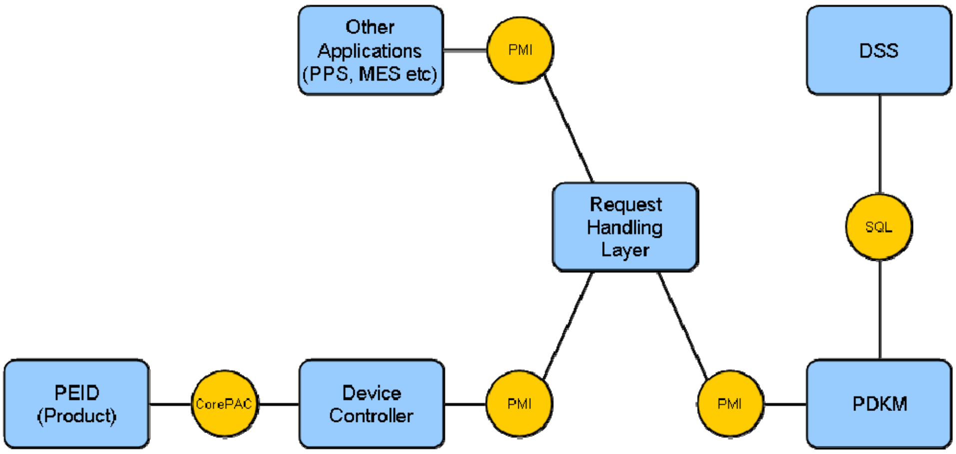


Figure 1: Configuration of PROMISE Components for the A2 Application Scenario

Main components of the A2 architecture are the Product Embedded Information Device (PEID), the Device Controller (DC), the Request Handling Layer (RHL) also referred to as Data Services, the Product Data and Knowledge Management (PDKM) System and the Decision Support System (DSS). Interfaces utilised in the configuration are the Core PEID Access Container (CorePAC), the PROMISE Messaging Interface (PMI) and Standard Query Language (SQL).

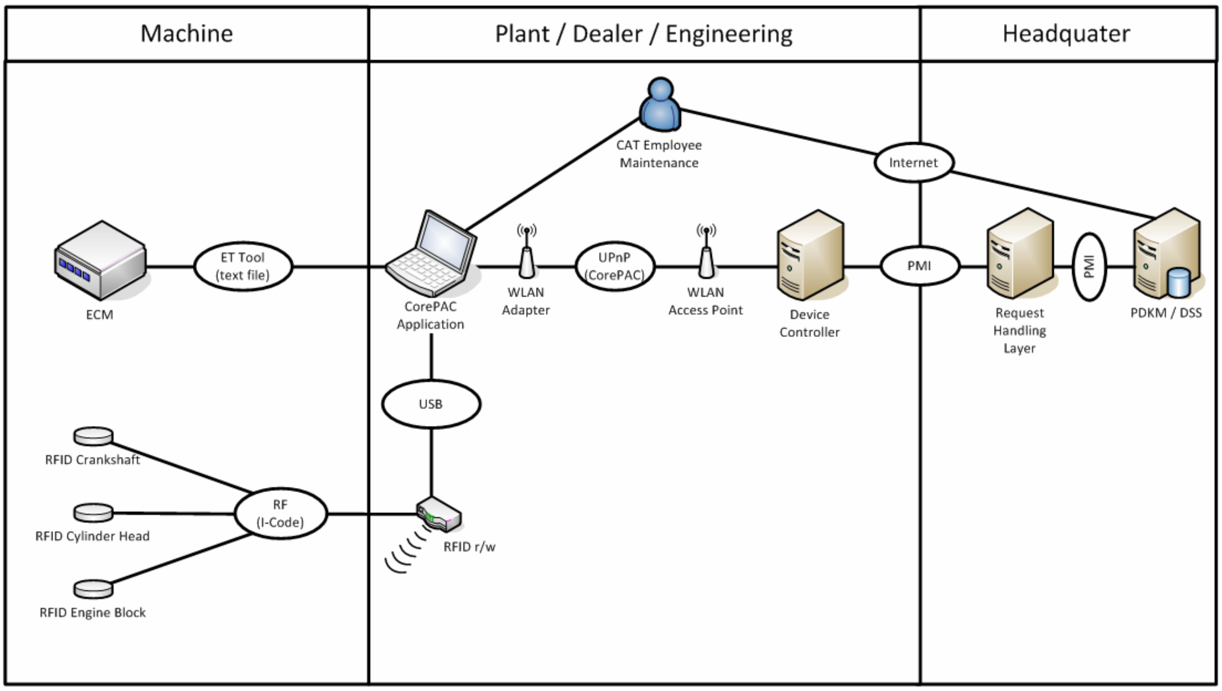


Figure 2: PROMISE Architecture as Implemented for A2 Demonstrator

### 2.1.2 PEID and CorePAC concept

PEIDs can be considered as the logical information sources for the product, which product in this demonstrator is a Caterpillar Track Type Loader (TTL). This TTL of the type 953D consists of various components. The components and related component structure which is under consideration for the A2, is depicted in Figure 3.

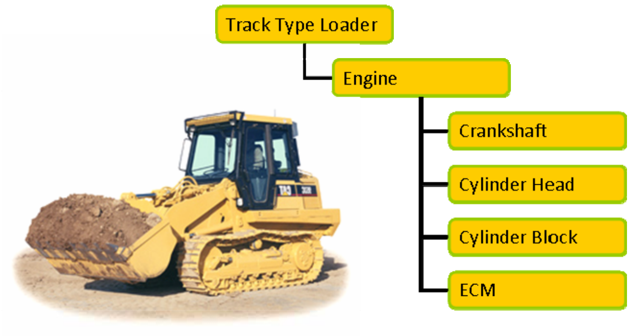
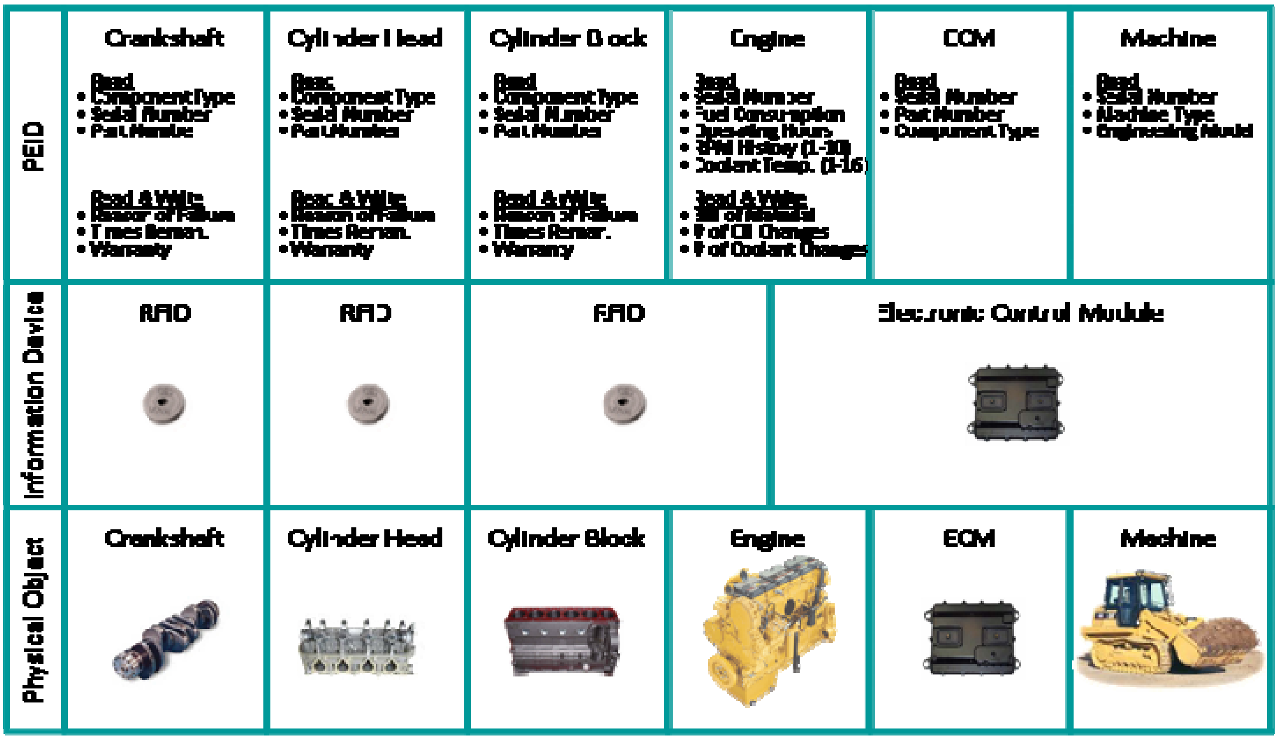


Figure 3: A2 Product structure

Each of the listed components is associated with at least some information which is relevant for the decision making process on the engine components at their EOL. To reflect this, each of these six components (including the machine) is represented by one PEID, which is then connected to a DC via CorePAC interface. An overview of considered components related information sources and associated PEID is given in Figure 4.



| | Crankshaft | Cylinder Head | Cylinder Block | Engine | ECM | Machine |
|---|---|---|---|---|---|---|
| PEID | Read<br>• Component Type<br>• Serial Number<br>• Part Number<br>Read & Write<br>• Reason of Failure<br>• Times Reman.<br>• Warranty | Read<br>• Component Type<br>• Serial Number<br>• Part Number<br>Read & Write<br>• Reason of Failure<br>• Times Reman.<br>• Warranty | Read<br>• Component Type<br>• Serial Number<br>• Part Number<br>Read & Write<br>• Reason of Failure<br>• Times Reman.<br>• Warranty | Read<br>• Serial Number<br>• Fuel Consumption<br>• Operating Hours<br>• RPM History (1-10)<br>• Coolant Temp. (1-16)<br>Read & Write<br>• Bill of Material<br>• # of Oil Changes<br>• # of Coolant Changes | Read<br>• Serial Number<br>• Part Number<br>• Component Type | Read<br>• Serial Number<br>• Machine Type<br>• Engineering Model |
| Information Device | RFID | RFID | RFID | Electronic Control Module | | |
| Physical Object | Crankshaft | Cylinder Head | Cylinder Block | Engine | ECM | Machine |

Figure 4: A2 Components, information sources, and related PEIDs

The three engine components which are under consideration for the decision making are each equipped with an RFID-Tag for unique identification of the component and for carrying a portion of the component specific data. Some of the data can only be written once to the tag, other data, such as information on warranty of the component, can be change (rewritten) during components life.

It is worth to mention that within this conception two specifics are realized.
First one is about the modelling principle: each information source should be also considered as an individual PEID. Nevertheless, with having the ECM as main information source for the products providing data on different components, it was a decision to partition this source into three PEIDs namely for the Machine, Engine and ECM.

Secondly, having an Engine PEID representing the data provided by the ECM about the engine restricts the overall solution in terms of writing back information from the PDKM to the product (Engine) because data cannot be written in standard processes (such as maintenance) onto the ECM. This is why it was decided to assign a portion of the engine related data to the component engine block, which in some sense represents the engine because it holds a unique serial number. Consequently RFID-tag and PEID of the cylinder block component is holding / representing a portion of the data that is considered as engine specific data e.g about maintenance events.

### 2.1.3 PEID and CorePAC implementation

PEID implementation for the A2 consists of hard- and software.

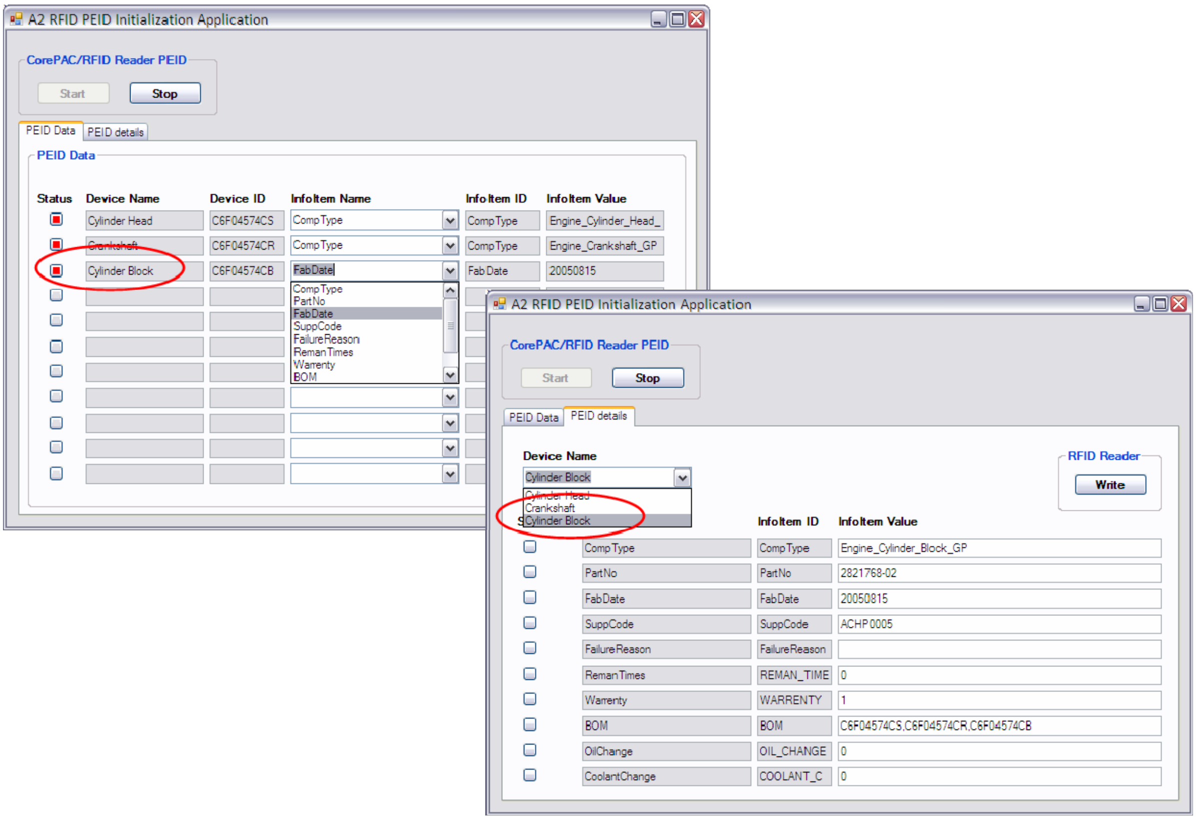
PEID hardware utilized for the demonstrator is as depicted above, where first the ECM and RFID tags are attached to the components of the Engine. For connecting to this hardware, additional hardware such as an RFID interrogator and a connection set for the ECM are required. Detailed information on the hardware used for the implementation is listed in Annex of this document.

On the software side, members of the A2 team decided to develop and implement two different software components to fulfil the required functionalities. Commonality of both software applications is their conformance to CorePAC gateway to the (hardware) PEIDs, which means that both software applications can and have to be accessed via the PROMISE infrastructure in a standardised way.

**CorePAC application for BOL process (scene 1)**
For the initialization process, it is required to write via the PROMISE infrastructure an initial set of data to the RFID-tag attached to a physical component.
As the PEID (RFID-tag) has no Unique Product Identifier (UPI) to address the write message, the PEID has to connect to a node in the PROMISE network, which is, in case of A2, a Device Controller. The DC announces the accessibility of the PEID by using the PEID UPI called TargetDevice ID to the network. After being connected and announced, read or write actions for this TargetDevice ID can be performed.

But, due to the fact that no UPI is existent on the RFID tag at the beginning messages, this cannot be addressed via the PROMISE infrastructure.

To overcome this hurdle, the software application for this scenario holds its own UPI and can be addressed directly. For the initialization of a component, the structured data describing UPI, Keys and values of the component can be communicated to the software application. After receiving data, the application creates and displays logical PEIDs as shown in Figure 5 for the engine components crankshaft, cylinder head and cylinder block. Initialization data can now be written to the PEID, which is now able to connect to the PROMISE infrastructure.


A2 RFID PEID Initialization Application

**CorePAC/RFID Reader PEID**

**PEID Data**

| Status | Device Name | Device ID | InfoItem Name | InfoItem ID | InfoItem Value |
|---|---|---|---|---|---|
| ☒ | Cylinder Head | C6F04574CS | CompType | CompType | Engine_Cylinder_Head_ |
| ☒ | ~~Crankshaft~~ | C6F04574CR | CompType | CompType | Engine_Crankshaft_GP |
| ☒ | Cylinder Block | C6F04574CB | FabDate | FabDate | 20050815 |

A2 RFID PEID Initialization Application

**Device Name** Cylinder Block

| | | InfoItem ID | InfoItem Value |
|---|---|---|---|
| ☐ | CompType | CompType | Engine_Cylinder_Block_GP |
| ☐ | PartNo | PartNo | 2821768-02 |
| ☐ | FabDate | FabDate | 20050815 |
| ☐ | SuppCode | SuppCode | ACHP0005 |
| ☐ | FailureReason | FailureReason | |
| ☐ | RemanTimes | REMAN_TIME | 0 |
| ☐ | Warrenty | WARRENTY | 1 |
| ☐ | BOM | BOM | C6F04574CS,C6F04574CR,C6F04574CB |
| ☐ | OilChange | OIL_CHANGE | 0 |
| ☐ | CoolantChange | COOLANT_C | 0 |

**Figure 5: CorePAC Application for initialization of component PEIDs**

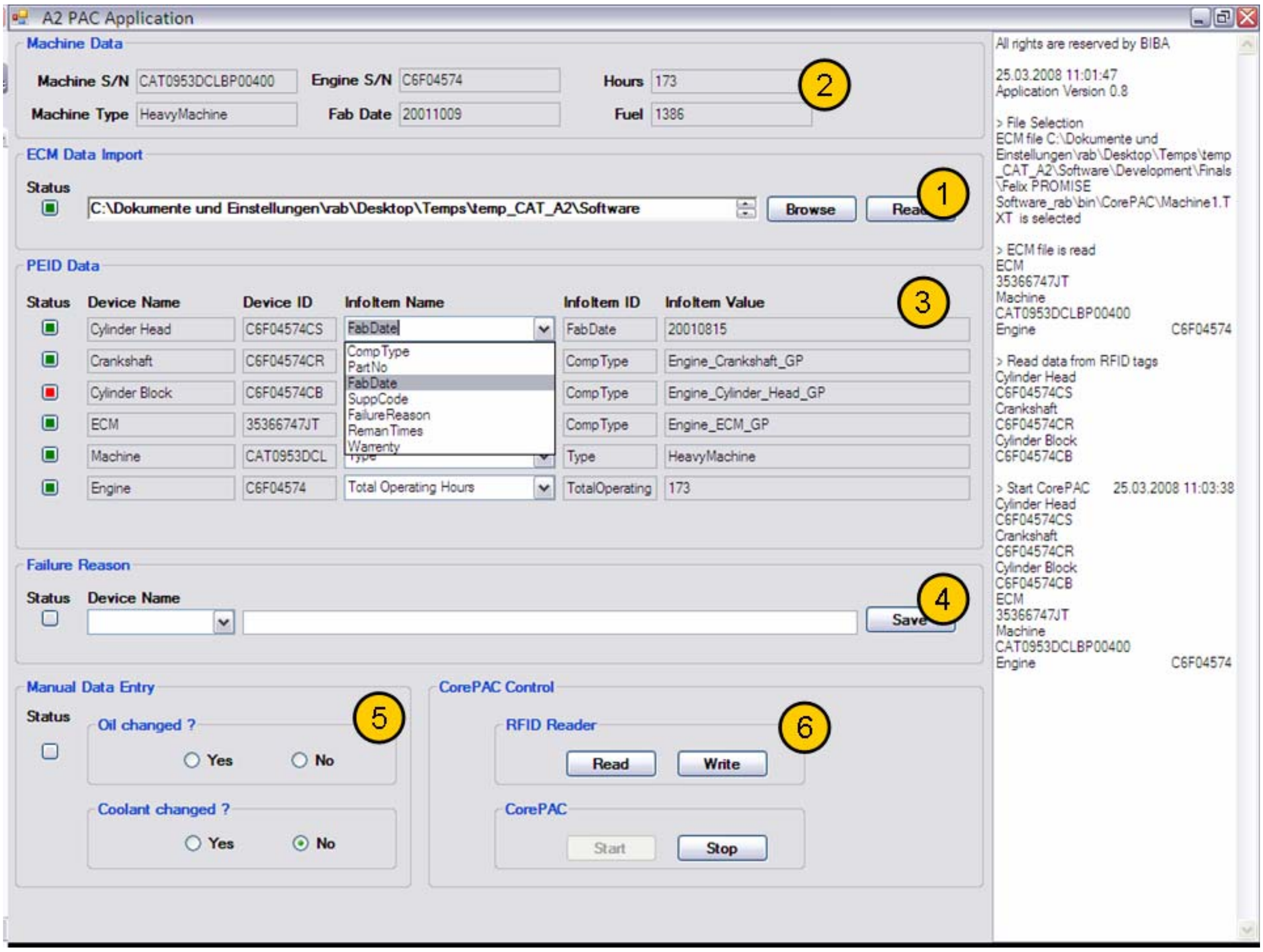
### CorePAC application for MOL and EOL processes (scenes 2-4)

Another CorePAC application was developed to support a set of processes for the MOL and EOL phases of the product (see scenes descriptions in chapter 3). The main technical requirements for the conception, design and implementation of the software are to provide functionalities to:

- Read and write data from / to the RFID-tags attached to crankshaft, cylinder head and cylinder block.
- Read information from a text file which is provided by maintenance software named Electronic Technician (ET) tool. This tool connects to the ECM and reads out all data relevant for the physical components Machine and ECM as well as some of the relevant data for the Engine.
- Enable service personnel to enter some additional data about failure reason of a component or whether engine oil or coolant fluid has been changed manually.
- Communicate product data via CorePAC interface to a DC
- Provide overall support to the user by e.g. displaying recent product data, indicating whether values have been changed and need to be updated on the PEID etc

The technical implementation of the software is based on Microsoft .net, C# technology and Intels open source UPnP SDK (for detailed information http://www3.intel.com/cd/ids/developer/asmo-na/eng/downloads/upnp/index.htm).

A screen of the application is given in
Figure 6.



Machine S/N: CAT0953DCLBP00400
Engine S/N: C6F04574
Hours: 173
Machine Type: HeavyMachine
Fab Date: 20011009
Fuel: 1386

Status 🟩 C:\Dokumente und Einstellungen\rab\Desktop\Temps\temp_CAT_A2\Software

| Status | Device Name | Device ID | InfoItem Name | InfoItem ID | InfoItem Value |
|---|---|---|---|---|---|
| 🟩 | Cylinder Head | C6F04574CS | FabDate | FabDate | 20010815 |
| 🟩 | Crankshaft | C6F04574CR | | CompType | Engine_Crankshaft_GP |
| 🟥 | Cylinder Block | C6F04574CB | | CompType | Engine_Cylinder_Head_GP |
| 🟩 | ECM | 35366747JT | | CompType | Engine_ECM_GP |
| 🟩 | Machine | CAT0953DCL | Type | Type | HeavyMachine |
| 🟩 | Engine | C6F04574 | Total Operating Hours | TotalOperating | 173 |

CompType
PartNo
FabDate
SuppCode
FailureReason
RemanTimes
Warrenty

Failure Reason — Status ☐ Device Name

Manual Data Entry — Status ☐
Oil changed ? ○ Yes ○ No
Coolant changed ? ○ Yes ● No

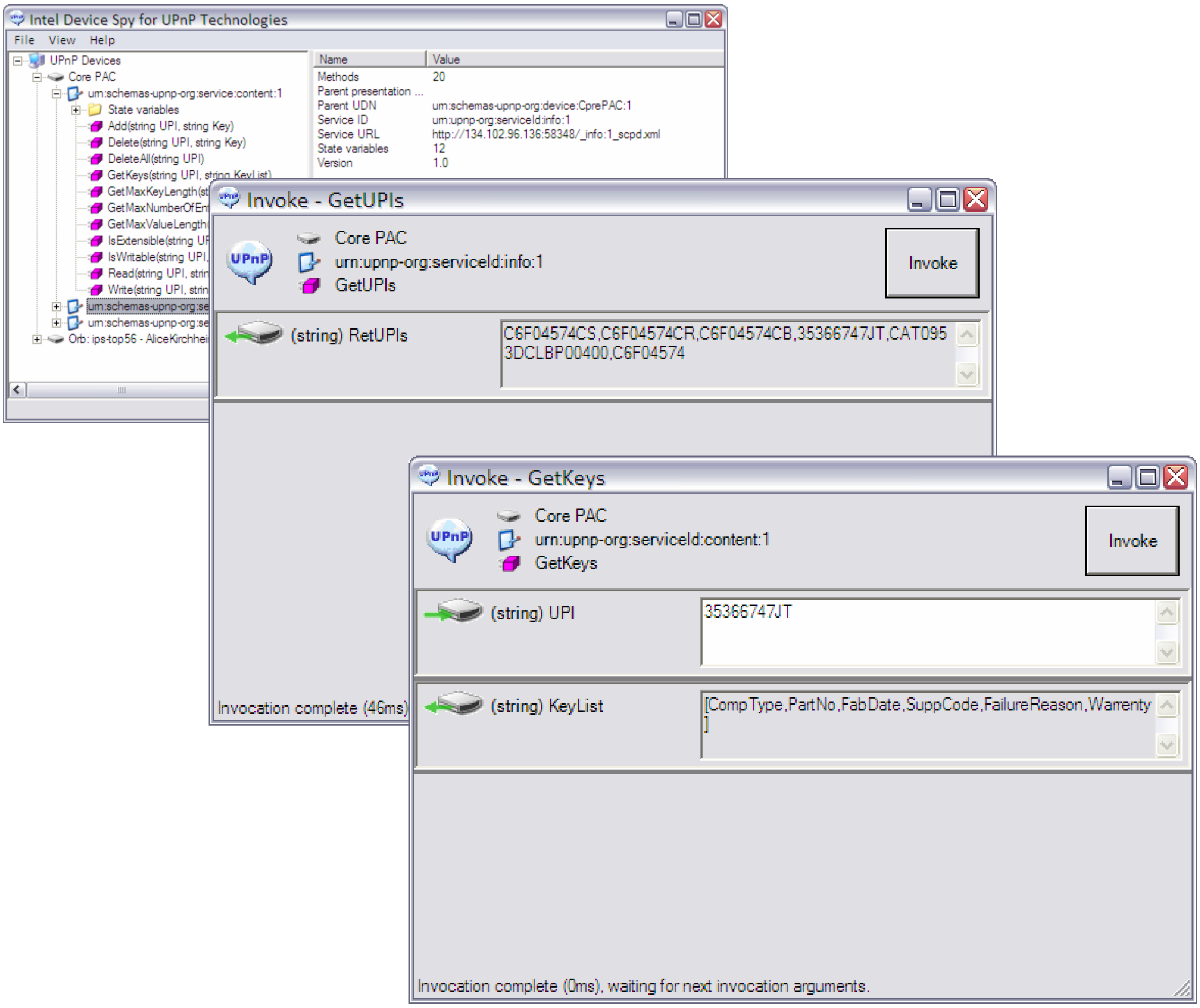
```
All rights are reserved by BIBA

25.03.2008 11:01:47
Application Version 0.8

> File Selection
ECM file C:\Dokumente und
Einstellungen\rab\Desktop\Temps\temp
_CAT_A2\Software\Development\Finals
\Felix PROMISE
Software_rab\bin\CorePAC\Machine1.T
XT  is selected

> ECM file is read
ECM
35366747JT
Machine
CAT0953DCLBP00400
Engine                    C6F04574

> Read data from RFID tags
Cylinder Head
C6F04574CS
Crankshaft
C6F04574CR
Cylinder Block
C6F04574CB

> Start CorePAC     25.03.2008 11:03:38
Cylinder Head
C6F04574CS
Crankshaft
C6F04574CR
Cylinder Block
C6F04574CB
ECM
35366747JT
Machine
CAT0953DCLBP00400
Engine                    C6F04574
```

**Figure 6: CorePAC application for MOL and EOL processes**

The following describes how to capture and communicate information through CorePAC user interface and how to update information on the PEIDs.

- Step 1: First step in the process is to read information which is extracted by the ET tool from the ECM by selecting a file from the file system of the computer.
- Step 2: Basic information about the machine and engine are then displayed in the header of the Graphical User Interface (see 2).
- Step 3: RFID tag information is captured by using the RFID interrogator (see 3).
- Step 4: Information on failure can be entered manually by the user in a limited free text field (see 4)
- Step 5: Information on oil and coolant fluid changes can be entered manually by the user (see 5)
- Step 6: Information on the PEIDs is displayed in a structured format. Beside the name and identifier (UPI) of the PEID, names, identifiers and values of the related keys, as well as the PEID status are displayed (see 3). A red colour indicates that either the PEID data is not available (not read so far) or data has been updated from the PDKM side.
- Step 7: To send PEID data to the PDKM, an UPnP device has to be started (see 6). This software device provides fully functional CorePAC confirm access to all PEID data.
- Step 8: After quitting the CorePAC UPnP device, updated data has to be written back to the PEIDs. PEIDs, which have to be updated, are indicated by the red colour in the PEID data field.

Basically, communication with the CorePAC UPnP device is achievable with every UPnP enabled device browser / explorer. Figure 7 shows a screen of UPnP browser (Intel Device Spy) displaying available services of the A2 CorePAC application.



| Name | Value |
|---|---|
| Methods | 20 |
| Parent presentation ... | |
| Parent UDN | urn:schemas-upnp-org:device:CprePAC:1 |
| Service ID | urn:upnp-org:serviceId:info:1 |
| Service URL | http://134.102.96.136:58348/_info:1_scpd.xml |
| State variables | 12 |
| Version | 1.0 |

Core PAC
urn:upnp-org:serviceId:info:1
GetUPIs

(string) RetUPIs: C6F04574CS,C6F04574CR,C6F04574CB,35366747JT,CAT0953DCLBP00400,C6F04574

Invocation complete (46ms)

Core PAC
urn:upnp-org:serviceId:content:1
GetKeys

(string) UPI: 35366747JT

(string) KeyList: [CompType,PartNo,FabDate,SuppCode,FailureReason,Warrenty]

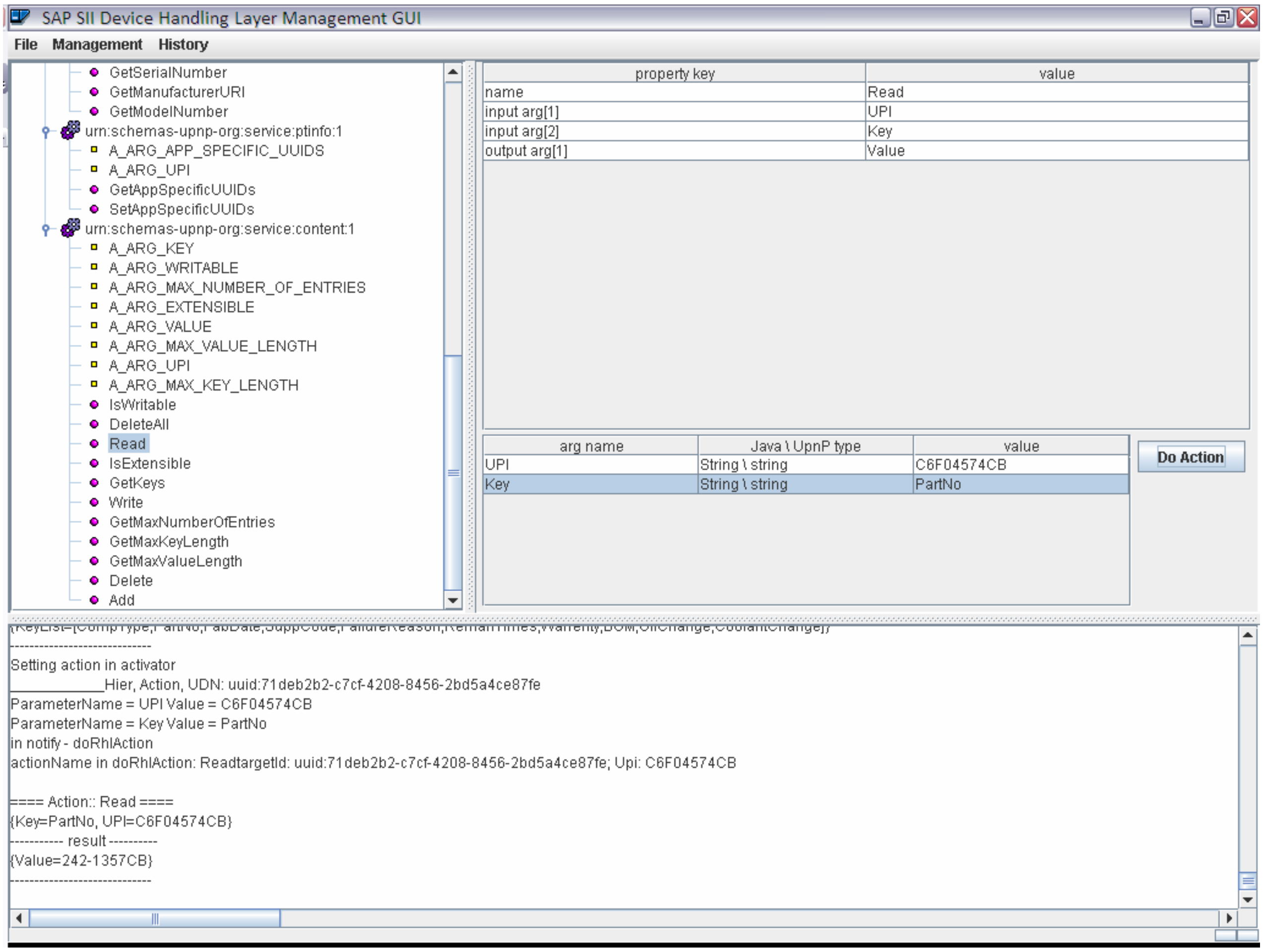
Invocation complete (0ms), waiting for next invocation arguments.

**Figure 7: Intel UPnP browser displaying CorePAC and access by getUPIs and getKeys**

### 2.1.4 Device Controller implementation

PEID data communication is realised by the Device Controller (DC) using the CorePAC UPnP application.

Figure 8 provides a screen of the DC GUI displaying the DC services as well as UPIs (TargetDeviceID), Keys (InfoItemID) and Values (InfoItem Value) provided by the A2 CorePAC UPnP device.



| property key | value |
|---|---|
| name | Read |
| input arg[1] | UPI |
| input arg[2] | Key |
| output arg[1] | Value |

| arg name | Java \ UpnP type | value |
|---|---|---|
| UPI | String \ string | C6F04574CB |
| Key | String \ string | PartNo |

```
----------------------------
Setting action in activator
____________Hier, Action, UDN: uuid:71deb2b2-c7cf-4208-8456-2bd5a4ce87fe
ParameterName = UPI Value = C6F04574CB
ParameterName = Key Value = PartNo
in notify - doRhlAction
actionName in doRhlAction: ReadtargetId: uuid:71deb2b2-c7cf-4208-8456-2bd5a4ce87fe; Upi: C6F04574CB

==== Action:: Read ====
{Key=PartNo, UPI=C6F04574CB}
---------- result ----------
{Value=242-1357CB}
----------------------------
```

Figure 8: PROMISE DC GUI displaying CorePAC services and results for a read service

### 2.1.5 Request Handling Layer

SAP Research has developed an architecture concept for a fully functional stand-alone device controller that is able to transfer any kind of data from the PEIDs to backend systems in a secure and flexible way. The implemented system is able to communicate with heterogeneous devices (sensors, embedded systems, RFID) using standard UpnP-based interfaces. Moreover, intermittently connected devices, which are common in PLM applications, are also supported. The details on the middleware concepts are described in the Deliverables of Workpackage R6, in particular DR6.1, DR6.2, DR6.3, and DR6.6.

Based on the concepts, a prototype device controller (see Figures 8 and 9) has been successfully developed and deployed for the PROMISE application scenarios. In order to achieve a high scalability and deal with geological distribution of PEIDs, a distributed implementation has been realized: The Request Handling Layer is to be deployed next to PLM application, while the Device Handling Layers can be installed in the close environment of the PEIDs. In PROMISE, a central instance of RHL is installed, supported, and hosted by SAP. Several instances of the Device Handling Layer are installed at the site of the application partners, Caterpillar, but also of other technology partners, namely BIBA, for A2 testing purposes.

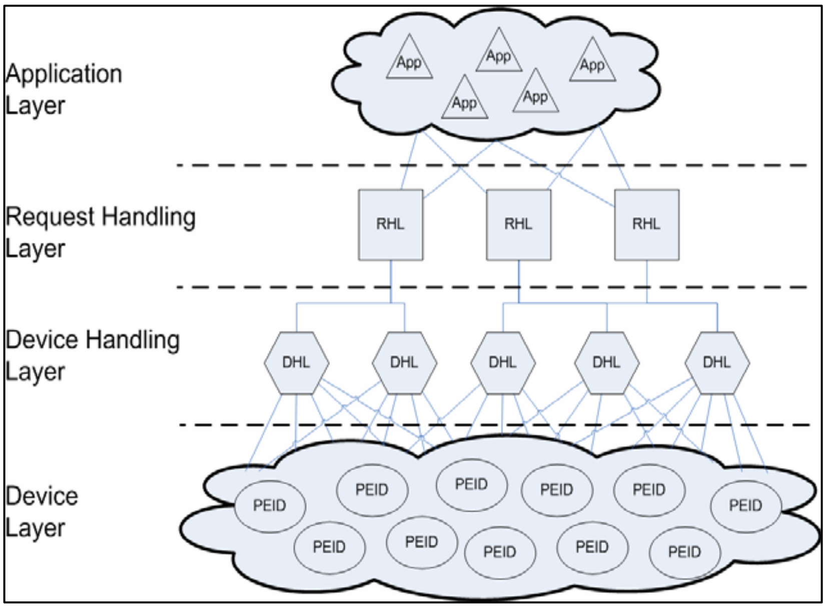
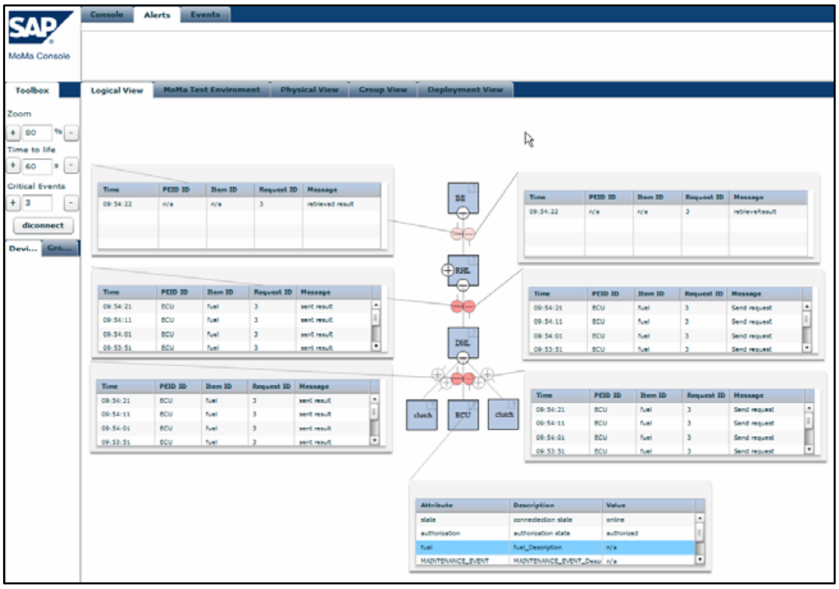


**Figure 9: Distributed middleware architecture**



| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:22 | n/a | n/a | 3 | retrieved result |

| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:21 | ECU | fuel | 3 | sent result |
| 09:54:11 | ECU | fuel | 3 | sent result |
| 09:54:01 | ECU | fuel | 3 | sent result |
| 09:53:51 | ECU | fuel | 3 | sent result |

| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:21 | ECU | fuel | 3 | sent result |
| 09:54:11 | ECU | fuel | 3 | sent result |
| 09:54:01 | ECU | fuel | 3 | sent result |
| 09:53:51 | ECU | fuel | 3 | sent result |

| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:22 | n/a | n/a | 3 | retrieveResult |

| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:21 | ECU | fuel | 3 | Send request |
| 09:54:11 | ECU | fuel | 3 | Send request |
| 09:54:01 | ECU | fuel | 3 | Send request |
| 09:53:51 | ECU | fuel | 3 | Send request |

| Time | PEID ID | Item ID | Request ID | Message |
|---|---|---|---|---|
| 09:54:21 | ECU | fuel | 3 | Send request |
| 09:54:11 | ECU | fuel | 3 | Send request |
| 09:54:01 | ECU | fuel | 3 | Send request |
| 09:53:51 | ECU | fuel | 3 | Send request |

| Attribute | Description | Value |
|---|---|---|
| state | connection state | online |
| authorization | authorization state | authorized |
| fuel | fuel_Description | n/a |
| MAINTENANCE_EVENT | MAINTENANCE_EVENT_Desc | n/a |

**Figure 10: Monitoring and management GUI for the entire Middleware environment**

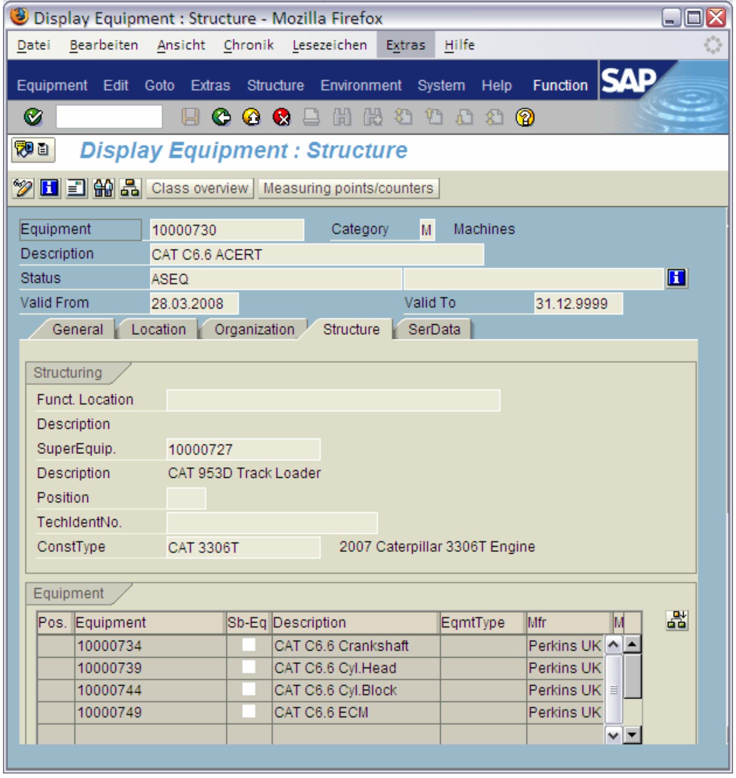
### 2.1.6 PROMISE Messaging Interface

For the implementation of the scenes the standard PROMISE Messaging Interface version 2 is used as documented in DR6.5 and the Architecture Series Volume 3.

### 2.1.7 Product Data and Knowledge Management

The Product Data and Knowledge Management (PDKM) system supports the creation of a product model and displays product data, information and knowledge related to product classes (groupings) and objects. For the creation of the A2 product model a SAP backend system was used. Access to this system was realised via SAP Netweaver / SAP Web Application Server.

Figure 11 shows a screen of the PDKM GUI used for the creation of the product model, displaying the details of a “CAT C6.6 ACERT” engine with attached components, and the machine to which the engine is assembled to.



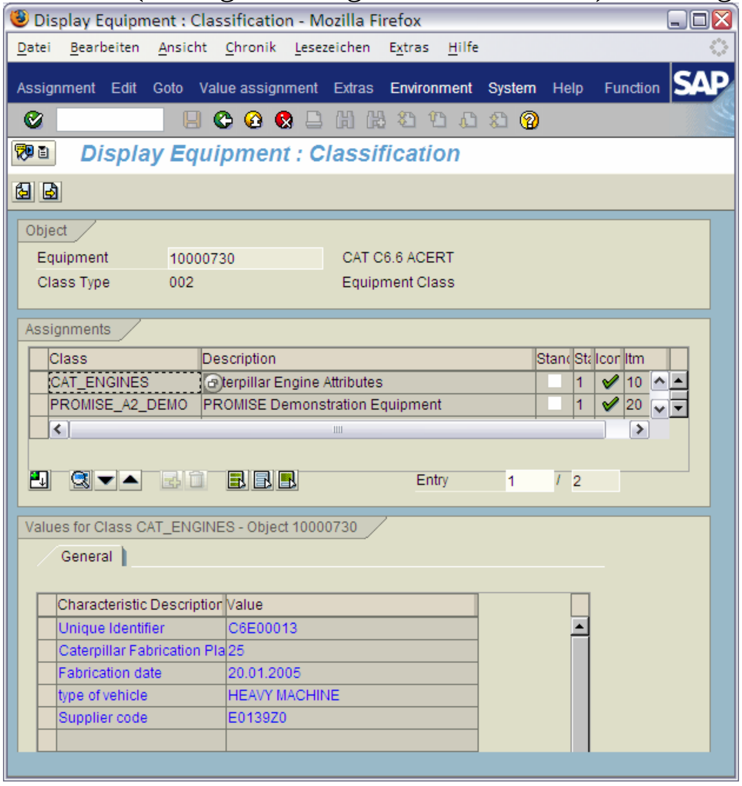
Equipment 10000730 Category M Machines
Description CAT C6.6 ACERT
Status ASEQ
Valid From 28.03.2008 Valid To 31.12.9999

Funct. Location
Description
SuperEquip. 10000727
Description CAT 953D Track Loader
Position
TechIdentNo.
ConstType CAT 3306T 2007 Caterpillar 3306T Engine

| Pos. | Equipment | Sb-Eq | Description | EqmtType | Mfr | M |
|---|---|---|---|---|---|---|
| | 10000734 | ☐ | CAT C6.6 Crankshaft | | Perkins UK | |
| | 10000739 | ☐ | CAT C6.6 Cyl.Head | | Perkins UK | |
| | 10000744 | ☐ | CAT C6.6 Cyl.Block | | Perkins UK | |
| | 10000749 | ☐ | CAT C6.6 ECM | | Perkins UK | |

Figure 11: PDKM view to Engine and related components

Engine attributes are described by using characteristics (e.g. fabrication date), which are organised in classes (see Figure 12 e.g. CAT_ENGINES) and assigned to particular equipment.



Equipment 10000730 CAT C6.6 ACERT
Class Type 002 Equipment Class

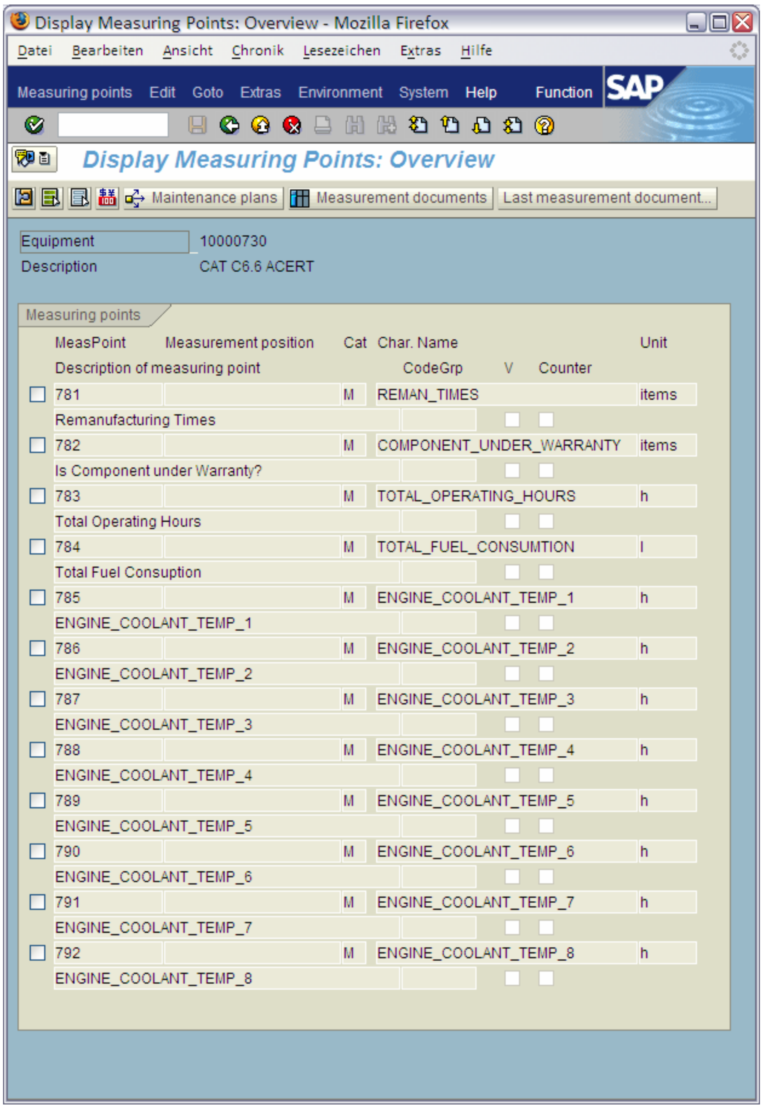
| Class | Description | Stan( | Sta | Icor | Itm |
|---|---|---|---|---|---|
| CAT_ENGINES | terpillar Engine Attributes | ☐ | 1 | ✓ | 10 |
| PROMISE_A2_DEMO | PROMISE Demonstration Equipment | ☐ | 1 | ✓ | 20 |

Entry 1 / 2

| Characteristic Descriptior | Value |
|---|---|
| Unique Identifier | C6E00013 |
| Caterpillar Fabrication Pla | 25 |
| Fabrication date | 20.01.2005 |
| type of vehicle | HEAVY MACHINE |
| Supplier code | E0139Z0 |

Figure 12: PDKM view to Characteristics of an Engine

Dynamic (field) data are defined as measuring points that are attached to particular equipment. The following Figure 13 provides a view to 12 out of 33 MPs defined for an engine.



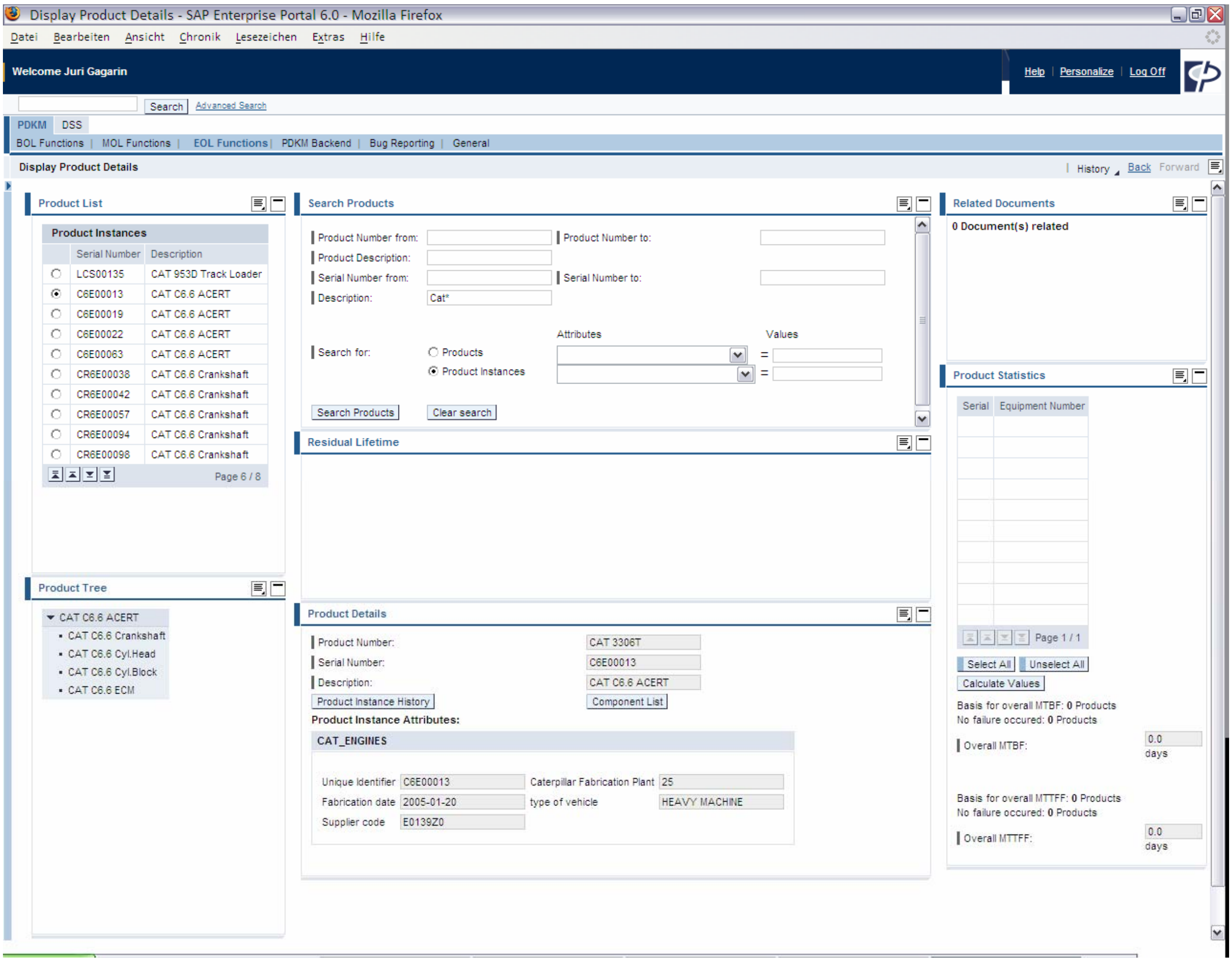
Display Measuring Points: Overview

Equipment 10000730
Description CAT C6.6 ACERT

| | MeasPoint | Description of measuring point | Measurement position | Cat | Char. Name | CodeGrp | V | Counter | Unit |
|---|---|---|---|---|---|---|---|---|---|
| ☐ | 781 | Remanufacturing Times | | M | REMAN_TIMES | | ☐ | ☐ | items |
| ☐ | 782 | Is Component under Warranty? | | M | COMPONENT_UNDER_WARRANTY | | ☐ | ☐ | items |
| ☐ | 783 | Total Operating Hours | | M | TOTAL_OPERATING_HOURS | | ☐ | ☐ | h |
| ☐ | 784 | Total Fuel Consuption | | M | TOTAL_FUEL_CONSUMTION | | ☐ | ☐ | l |
| ☐ | 785 | ENGINE_COOLANT_TEMP_1 | | M | ENGINE_COOLANT_TEMP_1 | | ☐ | ☐ | h |
| ☐ | 786 | ENGINE_COOLANT_TEMP_2 | | M | ENGINE_COOLANT_TEMP_2 | | ☐ | ☐ | h |
| ☐ | 787 | ENGINE_COOLANT_TEMP_3 | | M | ENGINE_COOLANT_TEMP_3 | | ☐ | ☐ | h |
| ☐ | 788 | ENGINE_COOLANT_TEMP_4 | | M | ENGINE_COOLANT_TEMP_4 | | ☐ | ☐ | h |
| ☐ | 789 | ENGINE_COOLANT_TEMP_5 | | M | ENGINE_COOLANT_TEMP_5 | | ☐ | ☐ | h |
| ☐ | 790 | ENGINE_COOLANT_TEMP_6 | | M | ENGINE_COOLANT_TEMP_6 | | ☐ | ☐ | h |
| ☐ | 791 | ENGINE_COOLANT_TEMP_7 | | M | ENGINE_COOLANT_TEMP_7 | | ☐ | ☐ | h |
| ☐ | 792 | ENGINE_COOLANT_TEMP_8 | | M | ENGINE_COOLANT_TEMP_8 | | ☐ | ☐ | h |

**Figure 13: PDKM view to Measuring Points of an Engine**

For accessing information about products defined in the back-end system in an appropriate format, the PDKM Portal (PDKM front-end) is used. Figure 14 shows details on the engine in the Portal GUI.



**Product Instances**

| | Serial Number | Description |
|---|---|---|
| ○ | LCS00135 | CAT 953D Track Loader |
| ● | C6E00013 | CAT C6.6 ACERT |
| ○ | C6E00019 | CAT C6.6 ACERT |
| ○ | C6E00022 | CAT C6.6 ACERT |
| ○ | C6E00063 | CAT C6.6 ACERT |
| ○ | CR6E00038 | CAT C6.6 Crankshaft |
| ○ | CR6E00042 | CAT C6.6 Crankshaft |
| ○ | CR6E00057 | CAT C6.6 Crankshaft |
| ○ | CR6E00094 | CAT C6.6 Crankshaft |
| ○ | CR6E00098 | CAT C6.6 Crankshaft |

Page 6 / 8

Product Number from:
Product Number to:
Product Description:
Serial Number from:
Serial Number to:
Description: Cat*

Search for: ○ Products ● Product Instances

Attributes Values

Search Products  Clear search

0 Document(s) related

- CAT C6.6 ACERT
  - CAT C6.6 Crankshaft
  - CAT C6.6 Cyl.Head
  - CAT C6.6 Cyl.Block
  - CAT C6.6 ECM

Product Number: CAT 3306T
Serial Number: C6E00013
Description: CAT C6.6 ACERT

Product Instance History  Component List

**Product Instance Attributes:**

CAT_ENGINES

Unique Identifier C6E00013
Caterpillar Fabrication Plant 25
Fabrication date 2005-01-20
type of vehicle HEAVY MACHINE
Supplier code E0139Z0

| Serial | Equipment Number |
|---|---|

Page 1 / 1

Select All  Unselect All  Calculate Values

Basis for overall MTBF: **0** Products
No failure occured: **0** Products

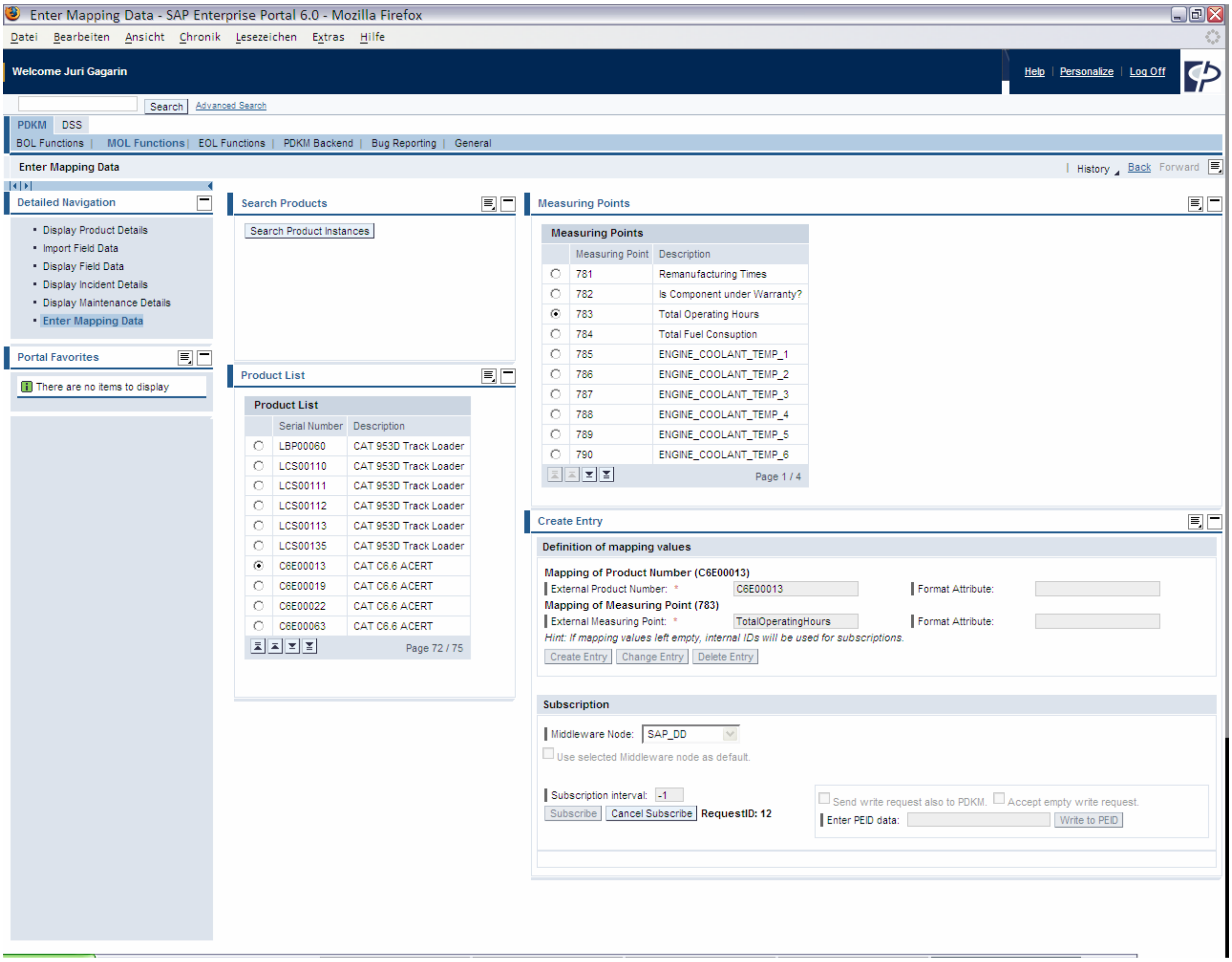
Overall MTBF: 0.0 days

Basis for overall MTTFF: **0** Products
No failure occured: **0** Products

Overall MTTFF: 0.0 days

**Figure 14: PDKM Portal view on engine details**

Figure 15 shows the Portal GUI with details on existent MP for a particular engine. In addition, functionality for mapping MP identifiers to InforItem IDs is displayed (see chapter 2.2 for details).



**Product List**

| | Serial Number | Description |
|---|---|---|
| ○ | LBP00060 | CAT 953D Track Loader |
| ○ | LCS00110 | CAT 953D Track Loader |
| ○ | LCS00111 | CAT 953D Track Loader |
| ○ | LCS00112 | CAT 953D Track Loader |
| ○ | LCS00113 | CAT 953D Track Loader |
| ○ | LCS00135 | CAT 953D Track Loader |
| ● | C6E00013 | CAT C6.6 ACERT |
| ○ | C6E00019 | CAT C6.6 ACERT |
| ○ | C6E00022 | CAT C6.6 ACERT |
| ○ | C6E00063 | CAT C6.6 ACERT |

Page 72 / 75

**Measuring Points**

| | Measuring Point | Description |
|---|---|---|
| ○ | 781 | Remanufacturing Times |
| ○ | 782 | Is Component under Warranty? |
| ● | 783 | Total Operating Hours |
| ○ | 784 | Total Fuel Consuption |
| ○ | 785 | ENGINE_COOLANT_TEMP_1 |
| ○ | 786 | ENGINE_COOLANT_TEMP_2 |
| ○ | 787 | ENGINE_COOLANT_TEMP_3 |
| ○ | 788 | ENGINE_COOLANT_TEMP_4 |
| ○ | 789 | ENGINE_COOLANT_TEMP_5 |
| ○ | 790 | ENGINE_COOLANT_TEMP_6 |

Page 1 / 4

**Definition of mapping values**

**Mapping of Product Number (C6E00013)**
External Product Number: * C6E00013
Format Attribute:
**Mapping of Measuring Point (783)**
External Measuring Point: * TotalOperatingHours
Format Attribute:
*Hint: If mapping values left empty, internal IDs will be used for subscriptions.*

**Subscription**

Middleware Node: SAP_DD
☐ Use selected Middleware node as default.
Subscription interval: -1
**RequestID: 12**
☐ Send write request also to PDKM. ☐ Accept empty write request.
Enter PEID data:

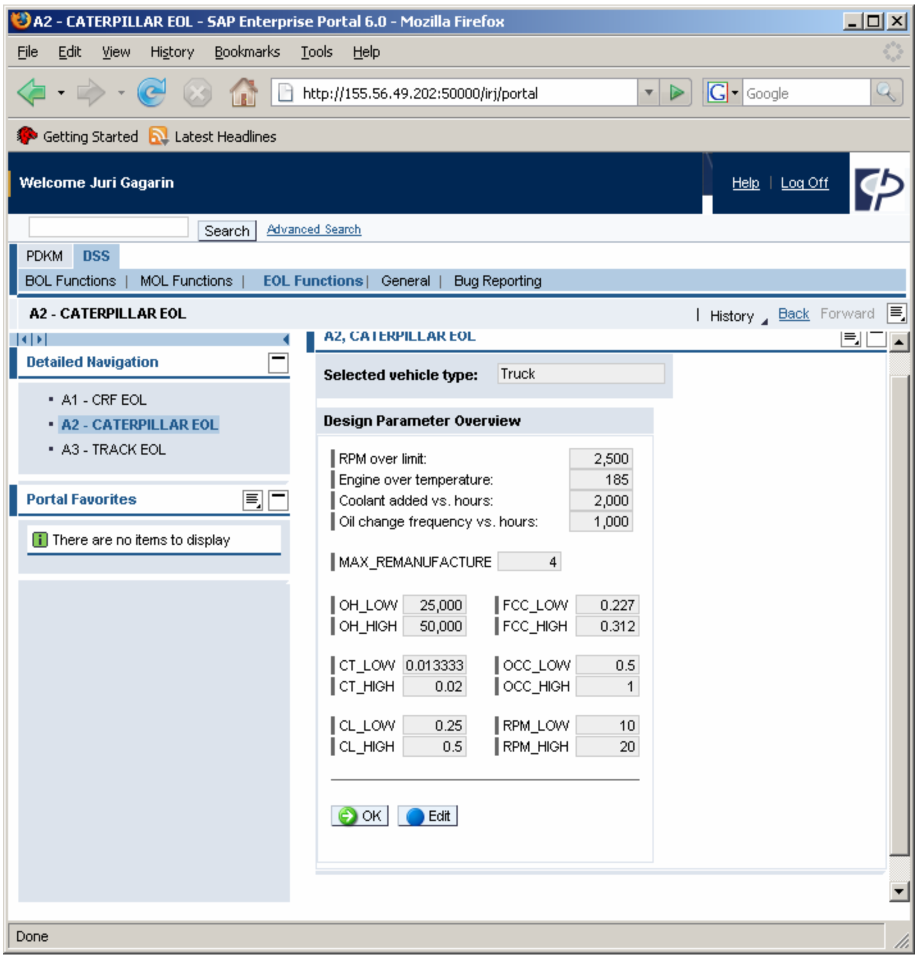
**Figure 15: PDKM Portal (front-end) view on engine Measuring Points and related mapping to InfoItemIDs**

### 2.1.8 Decision Support System

SAP Research has developed overall concepts for the implementation of the DSS GUI for all application scenarios in PROMISE. Using SAP technologies (WebDynpro and Enterprise Portal), the DSS GUIs of all application scenarios can be implemented in a uniform way and integrated with the PDKM GUI to provided the same look and feel. They are all deployed in the central server of SAP hosting the SAP PLM instance and accessible for the partners of the PROMISE consortium.

A sample DSS GUI for the A2 Caterpillar scenario is shown figure 15.

Details on the concepts and implementation of the DSS GUIs are described in previous deliverables, namely DR9.2 (User Interface Design) and DR8.10 (DSS Handbook).



Selected vehicle type: Truck

RPM over limit: 2,500
Engine over temperature: 185
Coolant added vs. hours: 2,000
Oil change frequency vs. hours: 1,000

MAX_REMANUFACTURE 4

OH_LOW 25,000
OH_HIGH 50,000
FCC_LOW 0.227
FCC_HIGH 0.312

CT_LOW 0.013333
CT_HIGH 0.02
OCC_LOW 0.5
OCC_HIGH 1

CL_LOW 0.25
CL_HIGH 0.5
RPM_LOW 10
RPM_HIGH 20

Figure 16: DSS GUIs implemented for end-of-life of heavy machinery

## 2.2 Data structure

Data to be considered by the A2 demonstrator are defined by the objectives of the demonstrator resulting in the algorithms applied for decision making. Physical objects and the product model are represented by the product structure depicted in Figure 3. Having initially three components under consideration creates a need to model not just those components but also objects related to these components. The main reason for this is that data, which is required for decision-making, is often not directly related to the considered components but to the engine or to the machine where components are assembled.

After defining an abstract product model that carries all relevant information based on the concepts of attributes and relations defined in the SOM, the technical models for the CorePAC and the PDKM have been created. The process of implementing a specific product data model for the A2 demonstrator underlies the restrictions given by the PDKM-System and the PMI and CorePAC interfaces definitions.

A pragmatically approach to implement the conceptual model into the PROMISE framework is given in Figure 18 where CorePAC, PMI and SOM concepts are mapped to the concepts provided by the PDKM, which is, in the case of A2, a SAP PLM-System.

Let’s consider as an example a physical item such as the crankshaft. This engine component is represented in different ways on the different levels of the architecture.

**On the CorePAC level**, a UPI with no name identifies this crankshaft. Related to this UPI there is a set of keys, which represents the existing field data types, related to the crankshaft. Keys are identified by a key name and a value.

**On the PMI level**, a target device ID and a set of InfoItems represent this crankshaft. ID and name basically describe InfoItems. InfoItems can carry a value, which corresponds to a field data type of the crankshaft.

**In the SOM**, an object (instance) of the physicalProduct class represents the crankshaft.
An object of the class ID_Info that is related to the physicalProduct object gives the identifier of this object. Values of field data or product attributes can be represented either as attribute of the crankshaft object or as an object of the FieldData class which is related to the crankshaft object.

**On the PDKM backend level** an equipment object represents a physical item such as the crankshaft. This object possesses a set of attributes, which are defined as characteristics. These characteristics can be grouped into classes (not related to the term classes from the object oriented modelling concepts). A class (see also Figure 12 CAT_ENGINE) can be assigned to an equipment, which means that all characteristics contained in this class can be used for describing the equipment. A particular characteristic of the A2 is the fabrication date that is contained in the CAT_ENGINE class. Field data related to the crankshaft are defined as Measuring Point (MP). MPs carry a description, an identifier valid for the particular equipment (NOT system wide). Attached to this MP are Measuring Documents (MD) carrying the actual field data values together with a time and date.

One of the main restrictions of the technology as implemented was the need to model all types of information that has to be communicated during the lifetime of the product via PMI as field data (Key / InfoItem / FieldDate / Measuring Point). Figure 17 displays the mapping of SOM / PDKM concepts to the concepts of the PMI / CorePAC.

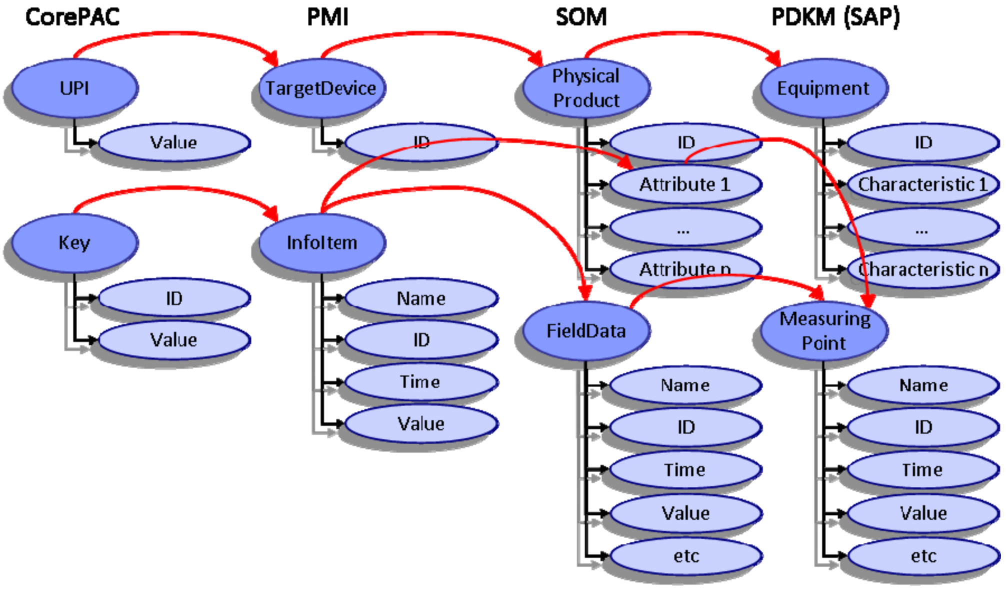


**Figure 17: Mapping of CorePAC, PMI, SOM and PDKM classes and attributes**

With these restrictions and the additional limitation that a PEID with its UPI has to have a 1-1 relation to an Equipment (e.g. an engine might require more than one PEID), the structure of the data on the product side was defined as follows (see also Figure 4).

- Each of the six physical objects into consideration, has its representation as PEID.
- Mapping of the PEID to the PDKM object has been realised by the usage of the serial number as UPI.
- Mapping of the Keys / InfoItems to the Measuring Points (MP) of the PDKM has been done by either utilising the PDKM internal identifiers of the MP on the product side or by filling in a mapping table on the PDKM side.
- First approach was not applicable for PEIDs which are based on data coming from the ECM because access to modify data identifiers on the ECM cannot be facilitated in the context of the demonstrator.

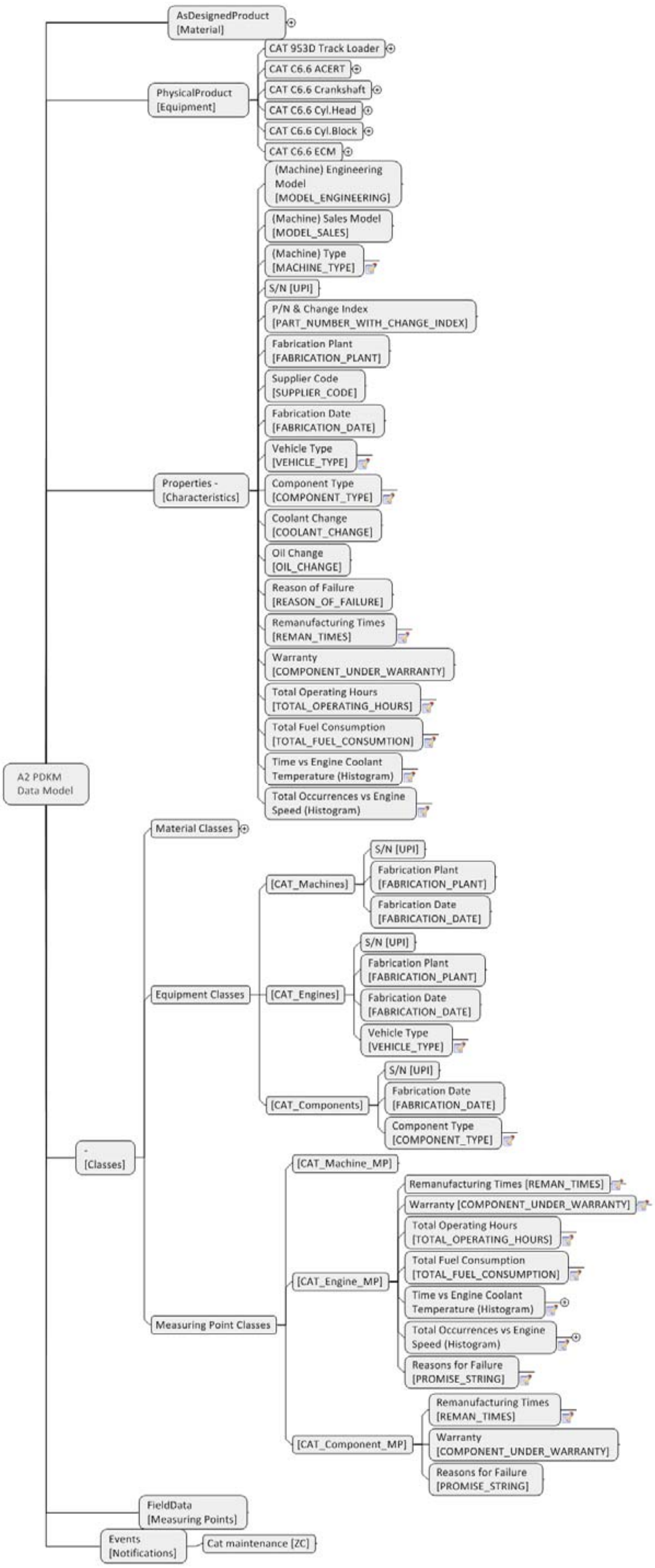


**Figure 18: Product data model (SOM / PDKM)**

## 2.3 Scenes included in the PLM model

Considering the PLM objectives of the A2 application described in DA2.4, the following demonstration activities are required:

1. At BOL, initialization of the PEID information on the PDKM database and on the PEID itself
2. During MOL, maintenance data up-dated on the PDKM database
3. At EOL of an engine component, part replacement process with BOM up-date onto both PDKM and engine tag
4. At EOL, ECM data recorded onto PDKM and onto RFID tags
5. At EOL of an engine component, possibility to use the DSS to take EOL decision on this PROMISE compliant engine component

From these scenes, we can extract functionalities that were implemented and validated using the PROMISE architecture:

1. Initialization of PEID & PDKM Equipment object
2. Data capture from PEIDs (RFID and ECM)
3. Data representation (CorePAC application)
4. Manual data entry (CorePAC application)
5. Data transfer to PDKM via CorePAC, DC, RHL
6. Data update on PEID (CorePAC)
7. Capture EOL data and enter manual data in PDKM
8. Read BOM Update from PEID
9. Update PDKM for Part Replace (exchange part process) & EOL Data
10. Update PEID for Component and Engine with new EOL information
11. Read ECM Data Read data from PDKM
12. Use DSS to make EOL Decision
13. Save DSS output in PDKM
14. Record DSS output on PEID

# 3 Implementation of the demonstrator

## 3.1 Implementation Step 1: BOL Initialization of PEID & PDKM

Setting up the demonstrator for an individual engine requires basically two things.
The first one is to create an object (instance of equipment) at the PDKM, which is the digital representation of the physical object engine.
Secondly, the physical object needs to be connected with this representation. This requires attaching a data carrier holding a unique identifier to each component on which information should be tracked later on. In the case of Caterpillars' engine components, a RFID tag is attached to the component carrying at least the serial number of the component.

### 3.1.1 General description

At manufacturing facility, once an engine is assembled and equipped with RFID tags, preliminary data will be written onto the RFID tags attached to the considered components using PROMISE infrastructure. Data to be carried by the components tag covers both static data, which will not be changed over the component life and dynamic data.
Static data for a component includes information about the serial number, part number, fabrication date and supplier code.
Dynamic data, which will be updated at least once in the lifetime of a component, includes information about its warranty, times component has been remanufactured and reason for failure of a component. In addition to this dynamic data to be stored on the tag attached to cylinder block includes information on the numbers of oil changes, numbers of coolant fluid changes and serial numbers of other components assembled to the cylinder block (see 2.1.2 for details).

For the demonstrator, physical objects such as machines, engines and engine components are listed in a MS Excel sheet. This Excel simulates the input, which is given in a real world implementation by a production system such as MES. Based on this Excel sheet, both the required input for the PEID initialization in a PMI format and the input for the batch import to the PDKM in a proprietary format can be generated. After the initialization processes, the new engine is shipped to a Caterpillar assembly facility to be put onto a machine. The same process will be followed to update the PEID and PDKM with the relevant machine information.

### 3.1.2 PROMISE components used

An overview on the components used in the initialization processes is given in Figure 19.

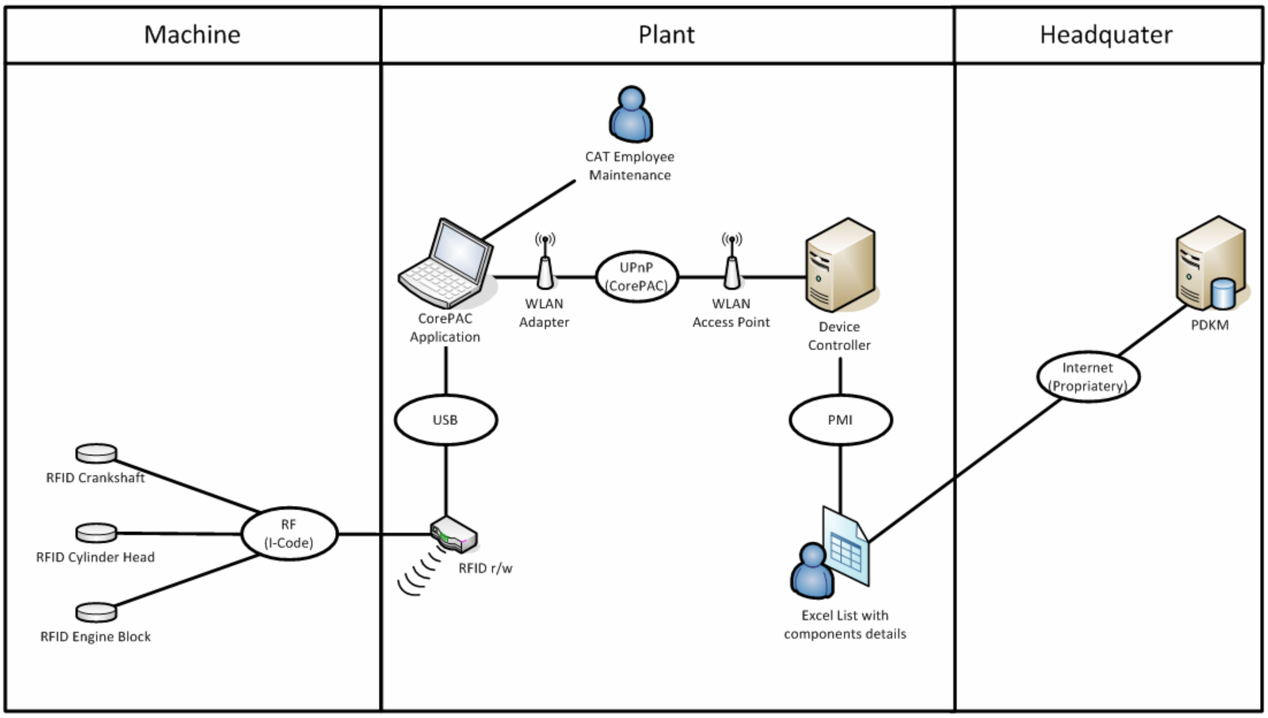


**Figure 19: Initialization process and related components**

As described above, main IT-components used in this scene are:

- Microsoft Excel simulates a production related information system (e.g. PPC, MES etc) holding the initial information for the initialization process.
- RFID tags and interrogator are used for component identification. In addition, tags carry components BOL, MOL and EOL data.
- CorePAC application (including RFID interrogator) is used for receiving and writing initialization data to component tags.
- PDKM is used for creating and holding digital representations of the considered physical objects.

### 3.1.3 Functionalities demonstrated

The functionalities that have been demonstrated with the first scene are the initialization of new objects inside the PROMISE environment. Both the initialization of the physical objects / PEID and the initialization of the digital counterpart in the PDKM (represented by an Equipment) have been performed. Technical functionalities in detail are:

1. Data transfer from PPS (Excel) to CorePAC via DC
2. Data reception and data representation (CorePAC application)
3. Initial data writing to RFID-Tag / Initialization of PEID (CorePAC)

### 3.1.4 Current and future refinement

The input in this scene is simulated by a simple list holding all information relevant for the initialization process. In a real production environment information is provided by a Production Planning and Control (PPC) system, which has at this time no interface to the PROMISE system. However, with having a defined structure for the information to be communicated, a simple list can be exported from the PPC into an Excel Sheet. This list can then be easily transformed into a PMI conform XML-File as demonstrated in this scene.

## 3.2 Implementation Step 2: MOL Maintenance Data Up-date on PDKM

Collecting MOL maintenance data involves two components.
First one is Caterpillar Electronic Technician (“ET Tool”) software that can read historic information from the engine ECMs.
Second one is a manual entry interface for data capture that is not captured in the ECM. This data will be used in the decision that will be taken at the EOL.

### 3.2.2 General description

A customer buys a machine from a Caterpillar dealership. In many cases (assumed in this scenario), the customers will return to that dealership for repairs and maintenance. When a machine is brought in, and maintenance work on the engine has been done, the maintenance personnel has to update engine and component related information on the PDKM. Update of information includes information on oil and coolant changes, dates and time of the event.

For capturing basic data about machine, engine and components, Caterpillar specific maintenance software “ET Tool” is used. While connecting physically the ECM to a notebook including “ET tool”, ECM Information is captured. Figure 20 shows the connection plug inside the driver cabin of the machine.

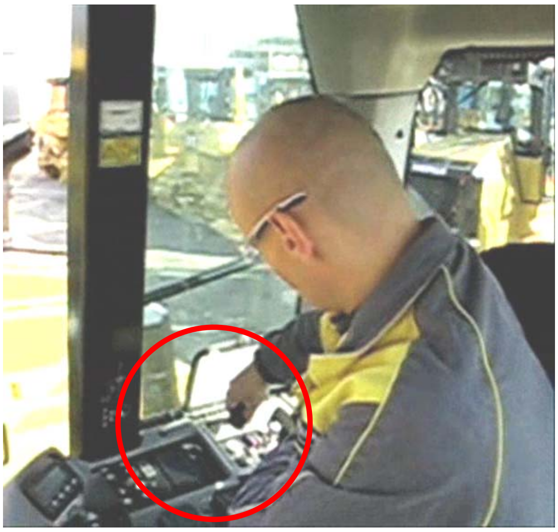


**Figure 20: Connecting ET tool to ECM**

**Figure 21: ECM Connection Kit**

After connecting to the ECM and starting the “ET tool” software, data can be downloaded from the ECM and stored into a text file into the file system of the maintenance computer. By starting the CorePAC software the data-file can be loaded into the CorePAC as described in
Figure 6 point 1. In a second step the details of the components can be read into the CorePAC by performing steps 3, 5 and 6 as described in chapter 2.1.3
Figure 6. Tagged components as used in the scenes are depicted in Figure 22, Figure 23 and Figure 24.

**Figure 22: Cylinder Block**

**Figure 23: Crankshaft**

**Figure 24: Cylinder block**

Information on oil change and coolant change / added has to be entered manually as shown in chapter 2.1.3

Figure 6 point 6. When all data is in place, CorePAC device can be started to enable the communication of the CorePAC with the PDKM via DC and RHL. Having a subscription on the field data of this particular engine and its components data will allow data to be automatically transferred to the PDKM.

### 3.2.3 PROMISE components used

An overview on the components used in the maintenance processes is given in Figure 25.

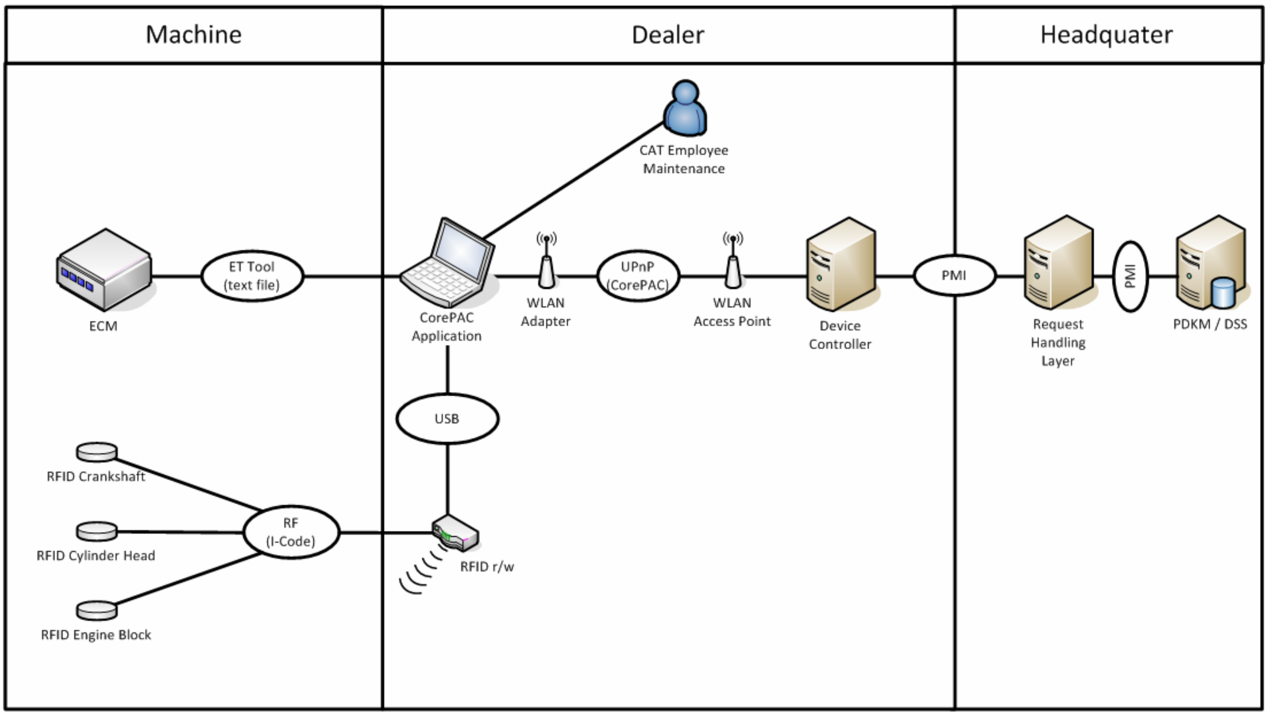


**Figure 25: Mainenance process and related components**

As depicted in Figure 25, main IT-components used in this scene are:

- “ET tool” software for extracting machine, engine and component related data into a text file.
- RFID tags and interrogator used for component identification and capture of cylinder block tag engine related data.

- CorePAC application (including RFID interrogator) is used for PEIDs data capture, manual data entry, data presentation (GUI) and data communication from and to the PDKM.
- DC for connecting the CorePAC software to the PROMISE system
- RHL for handling subscriptions and managing read and write requests to the CorePAC
- PDKM for making subscriptions on field data related to physical components and for creating new field data entries.

If the maintenance activity takes place in the field, information captured by the CorePAC software is stored in a PMI conform XML-file. This information can be transferred to the PDKM using a connection to a DC later on.

### 3.2.4 Functionalities demonstrated

The functionalities that have been demonstrated with the second implementation step are the field data capture from the various data sources as described in Figure 4, the manual data entry related to the maintenance event and the communication of the data to the PDKM-System.
More detailed technical functionalities are:

1. Data capture from PEIDs (RFID and ECM)
2. Data representation (CorePAC application)
3. Manual data entry (CorePAC application)
4. Data transfer to PDKM via CorePAC, DC, RHL
5. Data update on PEID (CorePAC)

### 3.2.5 Current and future refinement

Several avenues can be investigated in the short and longer term.

First improvement would be the development of a PDKM GUI where information related to the maintenance actions performed would be implemented. Additionally, a medium-term solution could be the integration of data captured in Caterpillar's internal system called Service Information System (SiS). This is a system where maintenance operations are recorded. The main concern when using this system is that usually only maintenance events are captured for warranty purposes and all dealers do not uniformly capture the information in this system.

Therefore, a longer-term solution is to implement physical solution for automatic update of maintenance events to avoid human interactions with the systems and ensure the automation of maintenance capture. As technologies to monitor oil level, oil or water pressure with specific sensors exist in the market, it is realistic to consider that we could develop a physical solution with associated processes that automatically would capture oil change or coolant change (or added) operation. More over, refined version of the DSS algorithms would try to be based on engine maintenance parameters that can be tracked automatically.

## 3.3 Implementation Step 3/4: EOL of Component / Part Replacement

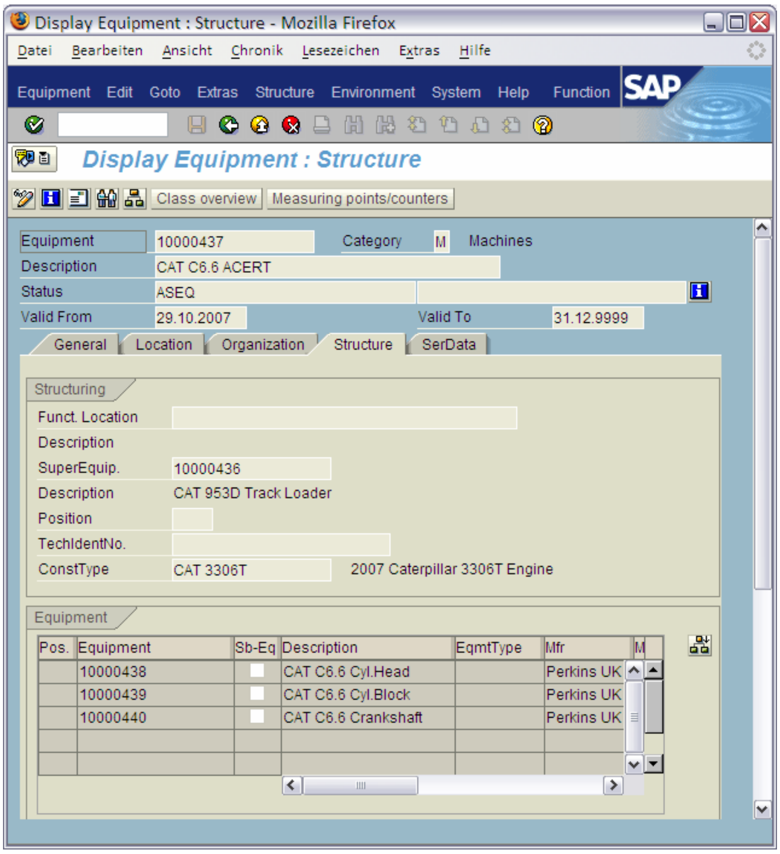
The two main processes in this implementation step are the capture of EOL data for an EOL component and BOM update related to the replaced component. The implementation step takes place at the dealer for a scheduled maintenance event or emergency maintenance in case of a failure or a problem.

### 3.3.1 General description

Maintenance actions as described above might also include the exchange of engine components. At the EOL of an engine component, the component tag will be up-dated with its EOL data and information on the engine tag will be updated with respect to the new BOL. Concurrently updated information will be stored in the PDKM. The process steps in detail are described in the following.

Data for both engine and component to be exchanged will be captured and transferred to the PDKM thanks to the use of the CorePAC application as described in chapter 3.2.
New component will be initialized as described in chapter 3.1.
After the new component is assembled to the engine, CorePAC software needs to be started again. By reading the component data from the attached RFID-tag / PEID, the CorePAC software informs the maintenance personnel that data on the cylinder block needs to be updated. Writing of BOM details to the cylinder block PEID can be performed directly from the CorePAC software.

Finally, disassembly of the old component and assembly of the new component needs to be done manually for the "digital counterpart" in the PDKM. Figure 26 shows the structure of an engine with four components (cylinder head, block and crankshaft) assembled to a machine.



Equipment 10000437
Category M Machines
Description CAT C6.6 ACERT
Status ASEQ
Valid From 29.10.2007
Valid To 31.12.9999

Funct. Location
Description
SuperEquip. 10000436
Description CAT 953D Track Loader
Position
TechIdentNo.
ConstType CAT 3306T 2007 Caterpillar 3306T Engine

| Pos. | Equipment | Sb-Eq | Description | EqmtType | Mfr | M |
|---|---|---|---|---|---|---|
| | 10000438 | ☐ | CAT C6.6 Cyl.Head | | Perkins UK | |
| | 10000439 | ☐ | CAT C6.6 Cyl.Block | | Perkins UK | |
| | 10000440 | ☐ | CAT C6.6 Crankshaft | | Perkins UK | |
| | | | | | | |
| | | | | | | |

**Figure 26: PDKM view on the product structure**

### 3.3.2 PROMISE components used

An overview on the components used in the part exchange processes is given in Figure 27.

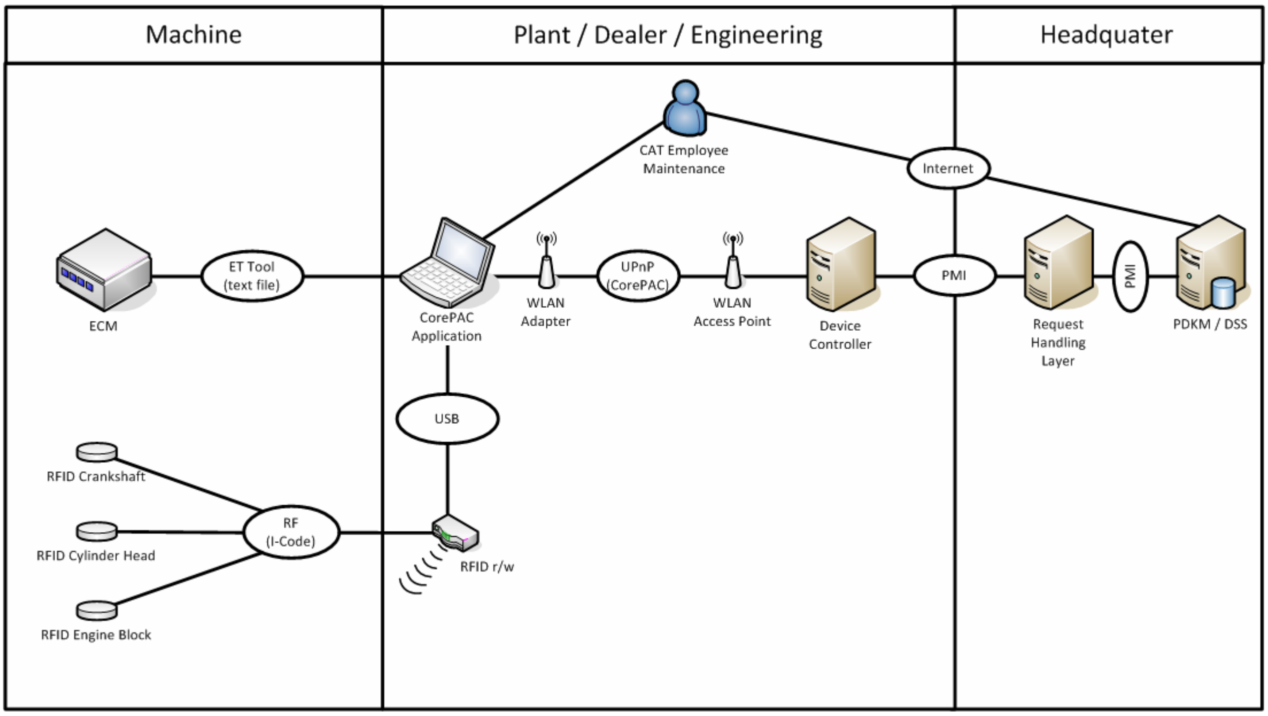


Figure 27: Part exchange process and related components

As depicted in Figure 27, main IT-components used in this scene are

- “ET tool” software for extracting machine, engine and component related data into a text file.
- RFID tags and interrogator used for component identification and capture of cylinder block tag engine related data.
- CorePAC application (including RFID interrogator) is used for PEIDs data capture, manual data entry, data presentation (GUI) and data communication from and to the PDKM.
- DC for connecting the CorePAC software to the PROMISE system
- RHL for handling subscriptions and managing read and write requests to the CorePAC
- PDKM for making subscriptions on field data related to physical components and for creating new field data entries.

Differently to the implementation of step 2 described in chapter 3.2, PDKM is also used for the modification of the product structure.

### 3.3.3 Functionalities demonstrated

The functionalities that have been demonstrated for this implementation are similar to the ones already described in chapter 3.2:

1. Data capture from PEIDs (RFID and ECM)
2. Data representation (CorePAC application)
3. Manual data entry (CorePAC application)
4. Data transfer to PDKM via CorePAC, DC, RHL
5. Data update on PEID (CorePAC)
6. Capture EOL data in PDKM
7. Read BOM Update from PEID

Main additional functionalities compared to implementation step 2 are:

8. Update PDKM for Part Replace (exchange part process)
9. Update PEID for Component and Engine with new EOL information
10. Manual update PDKM with EOL Data

### 3.3.4 Current and future refinement

Ideally, more automatic processes should be further investigated for future implementation; in particular for the ECM data extract and the BOM update processes.

For the demonstration purpose, and due to current proprietary communication protocol on-board the CAT machines, "ET tool" was used to download ECM data. In the medium term, as the CAT next generation of machines should more widely support standard CAN communication protocol, physical connexion between the ECM and the service personnel laptop equipped with CorePEID application could be developed with associated customized interfaces. The ECM data could therefore be automatically downloaded through the CorePEID and transferred to the PDKM.

The other refinement may be realized more in the long term as technologies advance, if RFID technologies could be used on-board a machine with enabled automatic Read and Write functionalities. Considering on-board R&W RFID system, the engine component being replaced could be automatically detected and BOM of engine automatically updated onto the engine block tag.

In this field of activities, preliminary tests with alternate passive RFID equipment were performed at CAT proving ground and provided optimistic results, as explained below.
The trial conducted by the Cambridge Auto-ID Lab using mount on metal passive UHF RFID tags (PEIDs) demonstrated how a master RFID tag on the engine block could be used to hold information about configuration and life span of ancillary engine components, which are also tagged with passive RFID technology.

This trial was intended as a 'proof of concept study' to meet the initial CAT demonstrator objectives of a fully automated solution for MOL monitoring of key engine component service and maintenance. RFID technology was utilized to tag a selected number of ancillary engine components (oil filter, fuel filters, fuel injector, value cover, exhaust manifold and fuel pump) on a 973D earth-moving machine.

The demonstration confirmed the ability to uniquely identify all the tagged components automatically by using an RFID Reader installed within the machine. Automatic updates of the ancillary engine components were demonstrated by recording their relevant hours of operation. The trial also successfully showed that it is possible to automatically detect part replacement (during a maintenance event) and record them automatically on the engine block tag. RF tests of the engine enclosure presented a challenging environment to work in for RFID technology. The demonstration supports a fully automatic solution does not involve re-training of existing CAT technicians or maintenance processes could be implemented for PLM in MOL of CAT earth moving machines.

As a conclusion, reducing human operations and manual entries in the systems will lead to more reliable and cost effective processes, but, in the context of the EOL demonstrator, not all EOL information could be updated automatically and the maintenance personnel will always need to interface with CorePEID GUI for defining the maintenance actions.

## 3.4 Implementation Step 5: DSS Use to make EOL Decision

At EOL of an engine component, the DSS is used to take EOL decision on this PROMISE compliant engine component.

### 3.4.1 General description

At the EOL of the engine (or engine component), and at the request by the dealer of an exchanged Remanufactured engine (or engine component), data from the PDKM is sent to the DSS. The DSS results are used to facilitate the transaction of the engine (or engine component) at Caterpillar Remanufacturing and can assist Caterpillar Remanufacturing in scheduling their builds.

For each engine component under the DSS, the DSS will use the engine information collected throughout the engine component life including ECM data and the maintenance that has been performed.

For the actual remanufacturing processes, personnel at the Reman facility use DSS results and physical components inspection to make decisions about the work to be done.

### 3.4.2 PROMISE components used

Assuming all engine information is recorded in the PDKM backend, the following PROMISE components are used to run the DSS at the Reman factory location:

- PDKM backend
- DSS java program
- DSS GUI

Interfaces:

- Mapping SQL statements for DSS program integration to the PDKM database
- Interface between DSS GUI and PDKM

Laptop to run the DSS should be configured with:

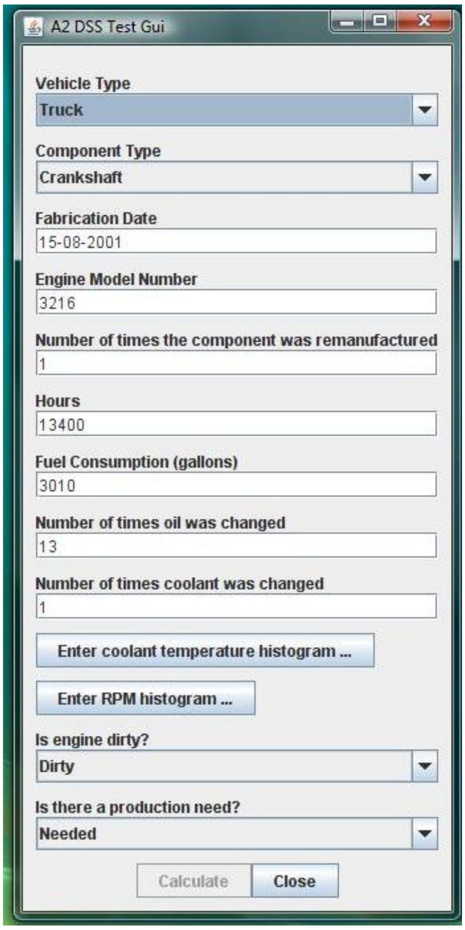
- Web access, with access to the PDKM server (located in Kalsrue)
- Java V5 installed

Due to technical issues for the Mapping SQL statements task for DSS program integration to the PDKM database; the DSS program has been validated on a standalone platform with dedicated user interface to manually enter data sets and to display DSS results.

### 3.4.3 Functionalities demonstrated

**Development and test of the stand-alone DSS (version 2)**

In order to test the functionality of the DSS algorithms, a stand-alone version of the DSS was created with a GUI in which data required by the DSS algorithms were entered manually. The DSS was successfully tested with four datasets (refer to the appendix for the datasets used). The input GUI is as shown in Figure 28.



Vehicle Type
Truck

Component Type
Crankshaft

Fabrication Date
15-08-2001

Engine Model Number
3216

Number of times the component was remanufactured
1

Hours
13400

Fuel Consumption (gallons)
3010

Number of times oil was changed
13

Number of times coolant was changed
1

Is engine dirty?
Dirty

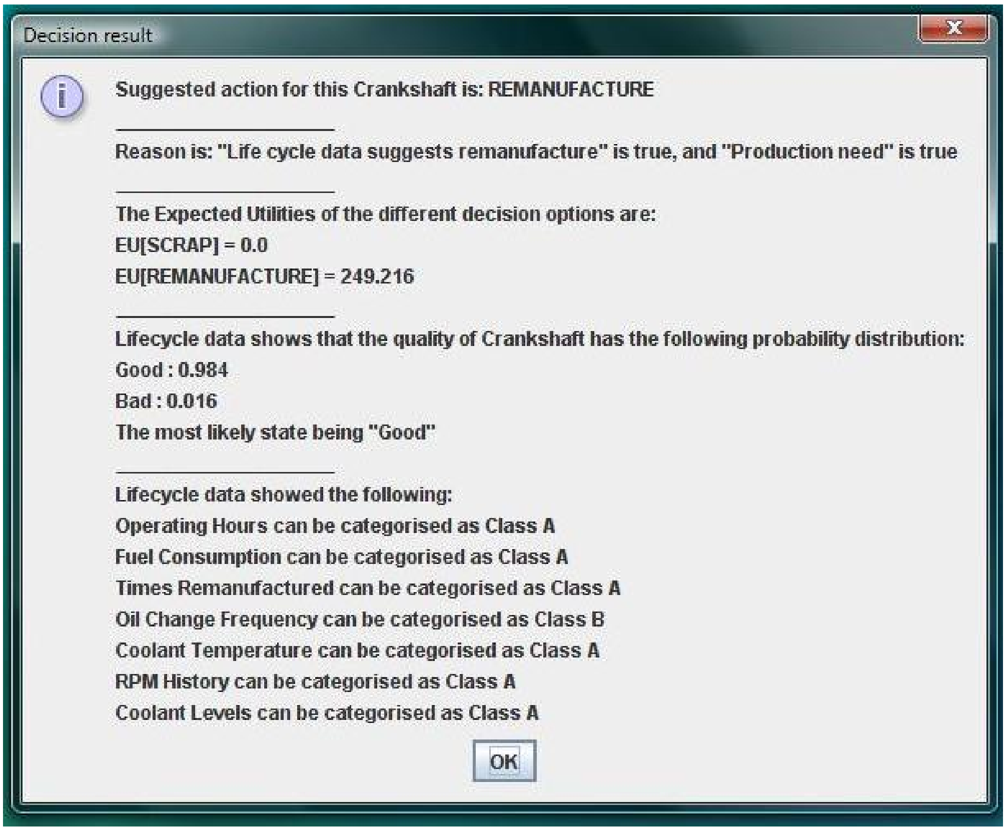
Is there a production need?
Needed

**Figure 28: Decision Support System GUI**

The key differences between version 1 and version 2 of the DSS is that the second version considers three different components (crankshaft, cylinder head, and engine block) in two different vehicle types (truck and heavy vehicle), whereas the first version only considered one component (engine block) on a heavy vehicle.

More over, Reman experts refined the probabilistic information considered in the model (definition of classes, conditional probability tables and utility tables for each engine component under the DSS)

On the basis of the given inputs, the test results showed that the DSS provided outputs that were expected. An example of the output interface is shown in Figure 29.



Suggested action for this Crankshaft is: REMANUFACTURE

Reason is: "Life cycle data suggests remanufacture" is true, and "Production need" is true

The Expected Utilities of the different decision options are:
EU[SCRAP] = 0.0
EU[REMANUFACTURE] = 249.216

Lifecycle data shows that the quality of Crankshaft has the following probability distribution:
Good : 0.984
Bad : 0.016
The most likely state being "Good"

Lifecycle data showed the following:
Operating Hours can be categorised as Class A
Fuel Consumption can be categorised as Class A
Times Remanufactured can be categorised as Class A
Oil Change Frequency can be categorised as Class B
Coolant Temperature can be categorised as Class A
RPM History can be categorised as Class A
Coolant Levels can be categorised as Class A

**Figure 29: Output interface (DSS)**

### 3.4.4 Current and future refinement

Refinement for the DSS use would be focused on the incorporation of the algorithm refinement using the DSS historical results and decisions taken by the operator based on the results. This would allow a more accurate algorithm to be built based on historical data that has been captured, versus the current algorithm that has been built using expert historical knowledge. Additionally, refinement needs to be done to examine the durability of the PEID during the remanufacturing process, and how information will be captured during remanufacturing processing.

# 4 Conclusion

Main issues during the implementation of the A2 scenes from the technical viewpoint were the simultaneous development of the PROMISE architecture and components.

We can consider that three main steps were conducted to develop, customize and implement the PROMISE architecture for building the A2 End Of Life demonstration case.

First step was to investigate and select appropriate technologies fitting the demonstrator design (DA2.3) and taking into account existing CAT systems, physical products and associated datasets.

Very intensive effort was then necessary to customize each PROMISE component (PDKM data structure, PEID and CorePAC interface, RFID UPnP enabled R&W system) to build a complete IT architecture enabling information flow between PDKM database and PEID – RFID tags.

Finally, implementation work was performed to develop methodologies and tools (CorePEID applications, DSS GUI) for the realization of the scenes happening during the lifecycle of an engine.

The scenes of tagging a new engine, recording maintenance events, replacing an engine component with a new one, up-dating an old component with end of life data or running the end of life decision support system where "programmed" and finally performed on a real CAT machine thanks to data initialization, data collection and automatic up-date.

As a conclusion, for the industrialization of the demonstrator on a larger scale, standard interfaces between PROMISE components and user-friendly process GUIs to facilitate the realization of the PLM scenes should be developed. More over, these interfaces will be required to easily up-date the PLM models.
Technically speaking, the different data models (ie PDKM, SOM and CorePAC) should be able to communicate, meaning that PDKM should support SOM data structure directly.

At last, the PROMISE IT system should communicate with the CAT systems to exchange information in common format and to have a unique and safe source of information. This goal was partly achieved with data extracted from the engine ECM using CAT ET software. For further application, easy methodology to make use of existing CAT systems and databases should be developed.

# 5 References

DA2.3 “Design of the A2 Demonstrator on information management for heavy load vehicles decommissioning”
DA2.4 “Process model workflow description for the demonstrator”
DR2.5 “PROMISE PLM models” (with EPC diagrams for A2)
DR6.5 “Interface definition and design of enterprise communication infrastructure”

# 6 Annex A

## 6.1 RFID System

RFID Interrogator
Requirements:

- Reading and writing
- 13.56 MHz (HF)
- Compact size
- low sensitivity regarding metal environment
- usability for very small TAGs

Selected System: Microsensys iID®PEN USB mini

- RFID Read Write Unit in pen design for PCs, Handhelds and PDAs
- Transponder: iID®2000, TELID®2 /3, I-CODE®, I-CODE®1, Tag-it®, EM types, my-D®, Ario (RO), ISO15693
- HOST Interface: USB 1.1
- Pen Size: app. D 12 mm, L 130 mm
- Package: metal
- Protection Class: IP 64 (without connector)

RFID Tags
Requirements:

- Applicable in harsh metal environments
- Temperature resistance up to 200 $^{O}$C
- Reduces size
- 128 kbit (BOM size increases during machine lifetime)

Selected System: Microsensys D14-TAGspecial

- Small Size Read Write Transponder in plastic packaging
- 256kbit memory
- Carrier Frequency: 13.56 MHz
- Chips: 2kbit or 16kbit read write EEPROM, based on ISO 15693, 128bit read write EEPROM up to 256kbit, based on ISO 14443B
- Size: D 14mm , T 2.5mm
- Package: lens or half lens form, epoxy or epoxy ferrite multilayer
- Certified for application in harsh environment conditions.

## 6.2 Computer maintenance / part exchange activities

Requirements for the computer to be used in the shop floor environment in the scenes are:

- Windows XP SP2 as operating system

- Microsoft .NET runtime environment for executing the CorePAC software and RFID software
- WLAN Adaptor for connecting the CorePAC to the device controller

Selected Hardware:

- Compact Size: Length: 10.6” (27.0 cm), Width: 7.2” (18.4 cm), Depth: 1.65” (4.2 cm)
- Weight: 3.7 lbs (1.7KG)
- Processor/Memory: 933 MHz Ultra Low Voltage Mobile Intel PIII with 512K L2 cache, 256 to 640 MB Low Voltage SDRAM
- Storage: Shock mounted 2.5” 40 GB hard disk drive
- Power: Smart 3600 mAH (40.0W) Lithium-Ion main battery pack, 3600 mAH (40.0W) Lithium-Ion expansion battery pack option
- Display: 8.4” SVGA TFT Outdoor Transmissive display with Digitizer Control Panel Touchscreen
- Pointing Device: Digitizer Control Panel Touchscreen
- Operating System: Microsoft Windows XP Tablet PC Edition
- Interfaces: PC Card slot for (1) Type I or II card with 32 bit Card Bus 2.1 Interface Compact Flash slot for (1) Type I or II card, Built-in RJ- 11 and RJ- 45 jacks for integrated fax/modem and Ethernet, (2) USB 2.0 connectors,26-pin docking connector, External speaker and microphone jacks,Connector for expansion Battery
- Communications: 56 Kbps V.92 fax/modem, 10/100 Mbit Base-T Ethernet LAN Integrated Wireless GPRS, CDMA, EDGE, 802.11b WLAN, Bluetooth (various combinations available depending on specific requirements)
- Specifications: 3ft drops onto plywood over concrete per MIL-STD 810F, 516.5 IV, Blowing rain/water resistance at 4” per hour at approximately 40 psi for 10 minutes on all 6 axes per MIL-STD 810F, 506.4 II, Dust per IP- 54,Vibration per MIL-STD 810F, 514.5 and ASTM 4169-99 truck assurance level II, Schedule E
- Intrinsic Safety & Non-Incendive: Class I, Division 2, Group A, B, C & D, certified by CSA International.

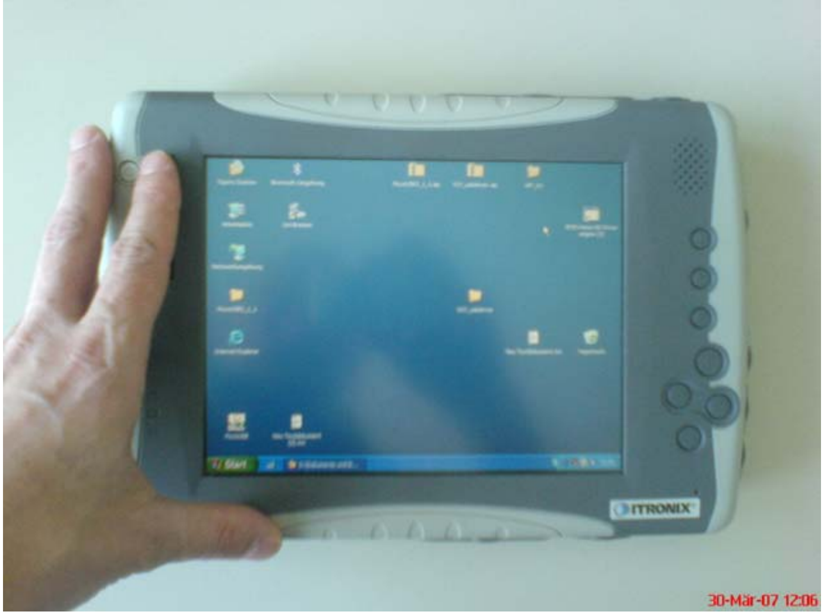


**Figure 30: Board Computer**