Distributed Decision Support in a PLM scenario

Jürgen Anke, SAP Research CEC Dresden, Germany
Kary Främling, Helsinki University of Technology, Finland

1. Abstract

The use of product embedded information devices (PEID) can help to close information loops between the various phases in the product life cycle. Data recorded by PEIDs allows monitoring of the product performance and reliability as well as of the environment it is operated in. Exchanging this data with external systems offers the chance to collect field data from a large number of product items to analyse the reliability of products in real-world operations. This knowledge can be utilised for product design improvements. Further decisions can be supported when this knowledge is combined with data from single product items: Maintenance plans can be made flexible by using more accurate prediction of remaining life time and recycling becomes more effective due to residual value estimation.
In this paper we propose a distributed architecture that supports decision making for maintenance based on dynamic creation of knowledge. After a short introduction we briefly discuss the different types of product information. Based on a real-world scenario of a major European truck-maker we demonstrate where this information can be acquired and derive the requirements for the architecture. Afterwards we explain the different parts of the system and their roles in the overall analysis chain. Finally, we discuss open issues and further research challenges that have to be solved to enable distributed decision support for predictive maintenance scenarios.

2. Introduction

Current product lifecycle management (PLM) systems provide necessary solutions for managing product data during the design and manufacturing phases. However, much of the information associated with the products is often irrecoverably lost after the point of sale. The use of product embedded devices (PEID) has been proposed as a potential solution to maintain the product information also during the rest of its lifecycle. In addition to being able to store information locally, PEIDs can also contain an identifier that makes it possible to find corresponding product information stored elsewhere, typically in a product data management (PDM) system.

Current PDM systems tend to represent product information on the level of the product type while PEIDs handle information on the product item level. This means that it may be difficult to store and update item-level information in the PDM system even though such information may be essential for after sales activities such as maintenance and End-of-Life (EOL) decision making. In this paper we will explain the principles of a product data and knowledge management (PDKM) system that builds up on a traditional PDM system. The PDM system enables the management of product structures, configurations and related documents. The PDKM system adds the ability to handle item-level product information and dynamic updates on product behaviour in the field. In this paper we will focus on the last point and propose a solution that is able to dynamically create knowledge on product behaviour in the field.

The structure of the paper is the following: after this introduction, section 3 describes different levels of product information and how they are used in product maintenance, section 3 relates these concepts to two industrial case applications and section 4 shows a proposal for an information system architecture that address the needs of these applications, followed by concluding remarks.

3. Background on product information and its use in maintenance

In the area of post-sale maintenance, product information is often separated into static information and dynamic information [7]. Static information means the specification of the product type as manufactured, e.g. materials, components, and suppliers used, configuration and options, servicing instructions and end-of-life information. Contrary to the static information, dynamic information comprises data that is collected during the use of a product. This includes patterns of use, environmental conditions, servicing actions, and part replacements.

Different applications have their own characteristics in how product information is created, gathered, maintained and provided to the ones needing it. *Marsh and Finch* introduce the concepts of centralized and localized (or decentralized) storage [6]. The localized solution involves storing information on the component itself. The opposite of a localized solution is the centralized model, where information is transmitted and collected to one place. In the centralized solution, it is easy to perform data analysis on entire product populations and detect general rules that can be used for improving product design and perform predictive maintenance, for instance. If dynamic information on the product item level can be continually transmitted to the centralised system, then the centralised system can also be used for real-time monitoring of every product item. In practice, communication of dynamic product information is often intermittent so real-time processing of it becomes impossible with a purely centralised solution.

It should also be pointed out that a centralized solution does not necessarily mean that product information would be located in only one place. As enterprises become increasingly global and networked (the “virtual enterprise”), product information tends to become spread on computer systems of multiple companies. This means that accessing product information may require communication between information systems of many different organizations [4].

3.1 Condition-based maintenance

Condition-based maintenance (CBM) is the key concept behind our architecture. It is based on the idea of preventive maintenance where parts are replaced before they fail. CBM extends this concept by making the points in time for replacements dependent on the observed condition of parts. Therefore assets are better utilized as an economical point in time can be chosen for replacements. Moreover, spare parts can be saved and the number of maintenance interventions is reduced.

In order to set up CBM, data from the products has to be collected and evaluated. This serves the assessment of condition that is called “diagnosis”. It is followed by the “prognosis” where data analysis algorithms are employed to predict when a product or one of its parts will fail. Both diagnosis and prognosis are operations where field data needs to be transformed into relevant information that can be used as input for a knowledge-based decision as described in the next section.

3.2 Transformation of data to knowledge

When dealing with terms like “knowledge creation”, it has to be clear what is meant by knowledge and how it differs from related terms like “data” and “information”. Unfortunately there is no commonly agreed definition of these terms. Therefore we have to find one definition that is useful for the scope of the addressed problem. We found the model of *Hicks et al* [3] particularly suitable for our purpose as it not only defines the terms but also explains how data can be transformed to information and how information can be transformed into knowledge, respectively.

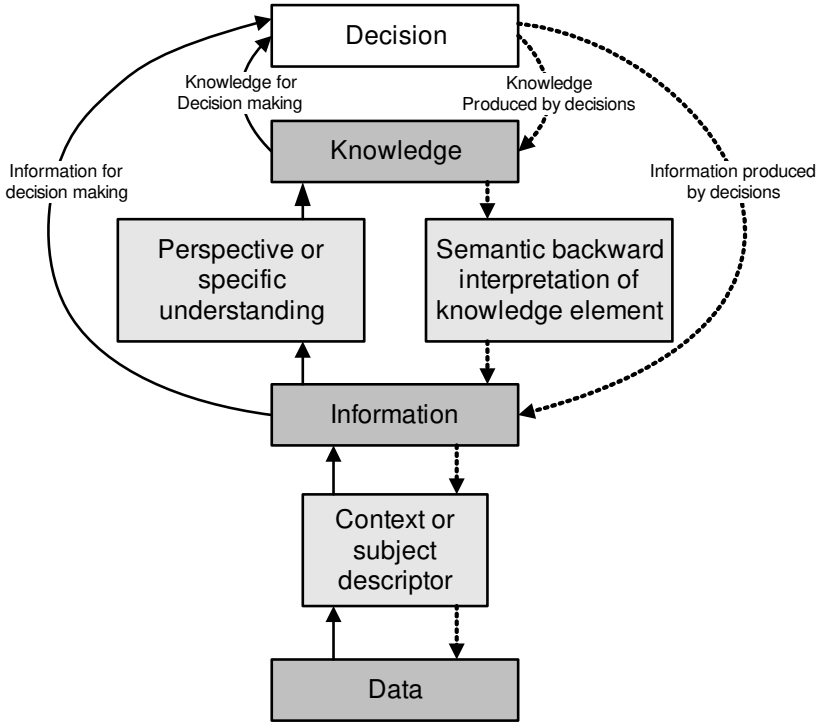

Fig. 3.1 : Bi-directional info. & knowledge transformation processes for decision making (Hicks et al., 2002)

According to *Hicks et al*, data refers to a structured single measurement, whereas information denotes a "fact". To transform data to information, some meaning has to be added to the data. In other words, information is data with semantics. Knowledge is defined as the ability to understand given information. It is created by inference from relevant information, combination of information, or by combining information with previously existing knowledge.

In the case of predictive maintenance, the transformation of data to knowledge can be understood as transforming recorded data from the product's operation into the knowledge of an optimal point of time for maintenance interventions.

4. Related work

Related work aiming at architectures for supporting condition-based maintenance has been done in the area of e-maintenance. The "Watchdog agent" of *Koc & Lee* attempts to predict failures and remaining lifetime of different parts by advanced analysis methods [4]. As pointed out in *Lee et al*, the most appropriate analysis methods to use depend very much on the requirements of the application [5]. Therefore the watchdog agent implementation presented proposes a toolbox of different signal processing and statistical methods, e.g. Fourier analysis, artificial neural networks, and fuzzy logic. They also point out the need for communication between different watchdog agents and other agents that represent machines or processes, for instance. Such inter-agent communications are typically performed over a computer network such as the Internet. In order to enable agents to understand messages sent by other agents there is a great need for established standards on how to represent and send information. In the area of CBM, the OSA-CBM initiative (www.osacbm.org) attempts to define such standards.

The PROMISE project of the European Union's IST 6th Framework program has defined a number of CBM application scenarios to implement [2]. Agent-based architectures have been studied as a strong candidate solution also in this project [1]. As for the watchdog agent, it is assumed that different analysis methods are needed for different applications. For instance the Agent Building and Learning Environment (ABLE) toolbox (www.alphaworks.ibm.com/tech/able) contains similar analysis methods as the watchdog

agent toolbox. The ABLE toolbox is written in Java and should be sufficiently lightweight to make it possible to embed the agents into PEIDs, but this remains to be verified in practice for the different application scenarios.

5. Application scenario

5.1 Business goals

The application scenario we describe here is a real world example of a major European truck-maker. We are presenting this case here for two reasons: First, we want to illustrate the problem we are addressing and second, it serves as a basis to derive the requirements for the architecture from it.

The goal of truck manufacturer is to create flexible maintenance plans for trucks. In a pilot phase the optimal time to change of oil, air filter, and brakes shall be determined based on the principles of condition-based maintenance. Data from PEIDs has to be analysed to determine the remaining lifetime for these components in order to predict failures before they occur. Thus, a replacement can be performed at an economically optimal point of time. At the same time, the availability of trucks will be improved.

Besides the planning of maintenance for a single vehicle, the scenario also includes management of maintenance interventions for truck fleets. Thus, the following kinds of decisions have to be supported:

1. When do specific parts of a truck have to be replaced?
2. When will a particular truck of the fleet be sent to the garage?
3. What truck should be sent on what mission so that it can perform the mission safely?

5.2 Required knowledge

In order to support these decisions, the remaining lifetime of different parts has to be estimated. For some parts the estimated remaining lifetime may be calculated by the on-board diary and sent to the PDKM system located at the ground station, which manages the truck fleet. For other parts, the remaining lifetime can be estimated in the PDKM based on collected field data. It would be an advantage to be able to calculate the remaining lifetime on the PEID for as many parts as possible. Calculating the remaining lifetime on the PEID would make it possible to react in real time on unexpected events and reduce the amount of data to be transmitted to the PDKM. In practice, the amount of processing that can be performed locally depends on the storage and computing capacities of the PEID. On the other hand, since the knowledge used for making the decisions may change as more field data is collected from similar vehicles, there also has to be a way to update the decision support models in the PEID.

In which way can a novel architecture improve the decision support? We argue that feeding back the created knowledge into the PEIDs will help to continually improve the accuracy of prediction for the remaining lifetime of a component. The knowledge to be provided to the PEIDs can be in the form of adjusted thresholds for raising events, interpretation rules for condition assessment, or class limits for summary statistics upon which analysis is performed. In the knowledge creation process depicted in Fig. 3.1 this signifies changing the understanding of information to improve the quality of generated knowledge. A system architecture that allows this kind of feedback will be presented in the next section.

5.3 Requirements for the architecture

A system architecture which supports the outline scenario has to fulfil the following requirements:

- **Collect field data:** Data on the product use and status has to be determined and recorded. Furthermore, this data has to be made accessible to external systems. This can be achieved either by providing querying capabilities or make the products send the data themselves.

- **Transform data into knowledge:** The recorded data has to be transformed into the knowledge of how probable a part failure is under a given workload.

- **Provide knowledge to Decision Support Systems (DSS) and PEID:** The created knowledge has to be provided to the DSS and PEID. If the probability of failure under given conditions is known, the optimal time for replacement can be determined, which is a precondition for appropriate alerts and notifications.

- **Feasible to embedded decision support:** It has to be possible to embed functionalities for decision support into products, in order to enable the driver/operator to respond timely to potential problems.

6. Proposed information system architecture

Here we present a system architecture that addresses the requirements elaborated in the previous section. We describe what the relevant parts are responsible for and show how they interact to allow for dynamic creation and updates of knowledge. Fig. 6.1 gives an overview of the proposed architecture.

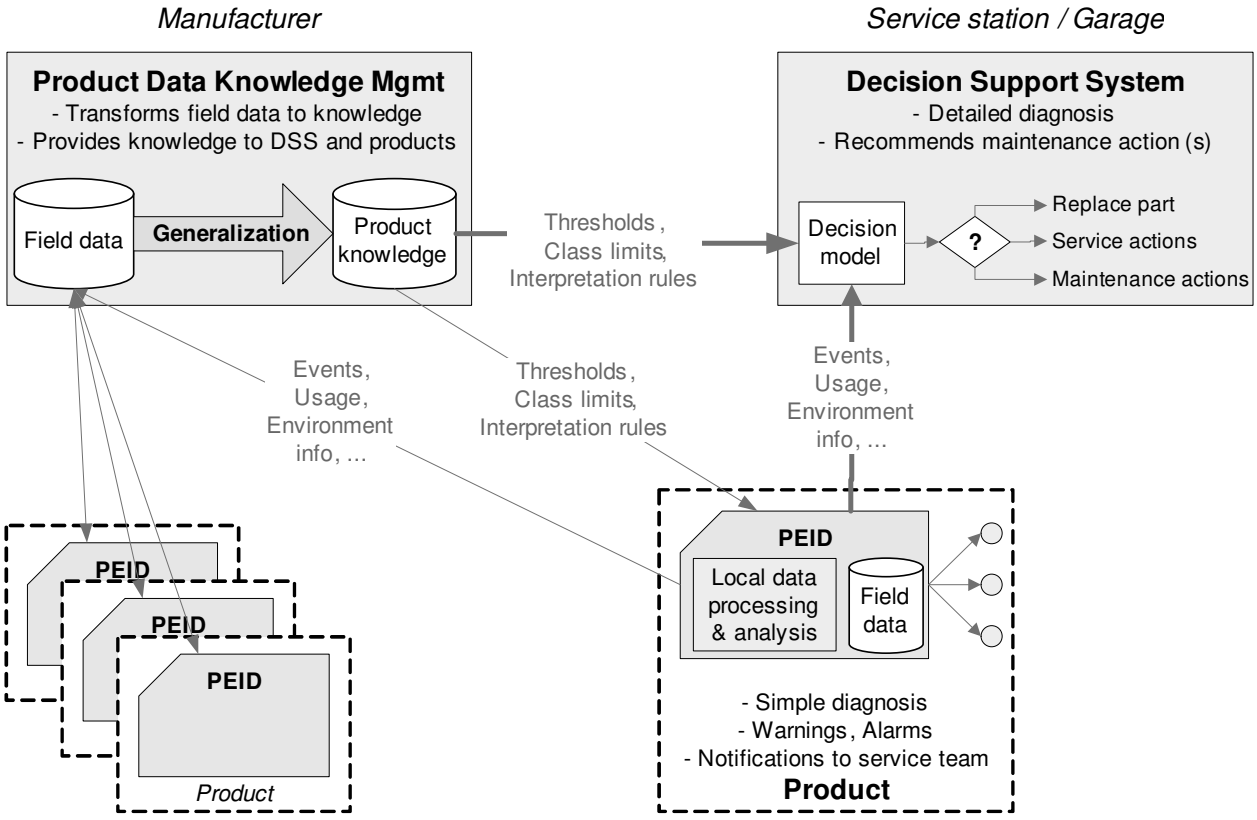

Fig. 6.1 : Overview of proposed architecture

6.2 PEID

The product embedded information device (PEID) is an embedded system that provides at minimum the ability to store and communicate the product identification. From that perspective, even a passive RFID tag can be considered as a PEID. In order to provide decision support on the product, a more sophisticated PEID is needed, e. g. an embedded system with reasonable memory capacity, processing capabilities and attached sensors.

In the described scenario, the PEID should be able to sense/detect data on the product's usage, operational status, and maintenance activity (see Fig. 6.2). However, it can be problematic to identify component replacements automatically. In this case, a maintenance engineer should be able to add relevant data manually, e.g. by using a PDA or a mobile RFID interrogator. In other scenarios it could be relevant to record environment parameters as well, e. g. humidity, outside temperature.

Usage	Operational status	Maintenance
Trip duration	Oil temperature	Number of brake changes
Fuel consumption	Water temperature	Energy consumption during brake action
Number of engine starts	Engine load	Component failures
Engine working hours	Air pressure at air filter	
Engine age	RPM	
	Boost Pressure	

Fig. 6.2 : Examples of different kinds of data to be recorded by the PEID

The PEID will not only record this data but also analyse it locally. The local analysis is done to alert or notify the driver/operator when overloads or abnormal conditions are detected and allow him to respond appropriately. For example, there will be a notification on the truck's dashboard if maintenance is required. Secondly, it can perform pre-processing of data before sending it to the PDKM. Such pre-processing can include the calculation of frequency of particular events, (e. g. component failure, engine starts, abnormal engine stalls), and detected trends (e. g. performance degradation, increased overheating). Pre-processing data aims at reducing the amount of data to be sent over the network. It is important though to carefully design these pre-processing operations as too much aggregation can compromise data analysis in the PDKM system.

In critical situations the PEID will send alerts to concerned parties, e. g. a service company, user, and manufacturer. Such incoming events represent the product's behaviour in the field and will form the basis for knowledge creation. If additional field data is needed from the products, it should either be possible to send a request to the PEIDs for additional data or configure them so that they send the respective data regularly.

6.3 PDKM

The Product Data and Knowledge management (PDKM) system builds up on a traditional PDM system, which enables the management of product structures, configurations and related documents. The PDKM system adds the ability of dynamic updates on product behaviour in the field. This knowledge is created based on the recorded field data that will be sent to the PDKM.

In the case discussed here, the analysis to be performed should determine a correlation of component failure and usage of products. Usage data to be included in this analysis can consist of working hours, overload, age of components and others. It is important to note that the expected remaining lifetime of a product is not determined in the PDKM. The purpose of the PDKM is to provide a correlation between component failure and usage. When a new product type is introduced, this correlation is set to the assumptions of reliability and durability made during product design. By analysing field data from products these assumptions are adjusted to reflect real behaviour of products and therefore make predictions on remaining lifetime more reliable.

The PDKM's purpose is not limited to providing knowledge for a single use-case. The knowledge of component failure is not only relevant for maintenance planning but is also highly interesting for the engineers who design new products. In fact, the knowledge created here should depend as little as possible on specific applications. Thus it can supply knowledge to various users in different scenarios. As PDKMs are most likely to be located in the manufacturer's organisation, it will have to provide versatile knowledge to a large number of product users who in exchange would allow field data to be gathered from the products they use.

6.4 DSS

Decision support systems help decision makers by giving more insight in the situation in order to help making better decisions. A prediction model will be needed to integrate the field data and estimate the remaining lifetime. This model will be based on physical properties of the material, e.g. percentage of degradation after a certain mileage etc. In general, the input data needed for the prediction model is the relevant data. How the actual support is provided can differ greatly. Simpler systems offer data analysis; more sophisticated ones provide simulation and recommendations. For the maintenance scenario two types of DSS are foreseen. One is running on the PEID and can warn the operator or give alerts to the service team. The other one is used at the ground station to manage the maintenance activities of the whole truck fleet.

- The DSS on the PEID is used to determine the remaining lifetime of the products or its components. This is done by comparing to the correlation of failure and usage (knowledge provided by the PDKM) with the usage of the product (field data available on the PEID). If defined thresholds are exceeded, the user will be notified of the situations' severity and recommendations for actions are given.

- A second DSS is used in the ground station to help planning the maintenance interventions for trucks. All notifications from the trucks are collected and scheduled according to their priority. For a detailed diagnosis additional data can be requested from the PEID. Such diagnosis can be helpful in order to decide whether spare parts or other material is needed for the scheduled maintenance.

In most cases, the DSS needs to have access to product item-specific information, i.e. collected field data and 'static' information on the product (CAD drawings, bill of materials ...). The provision of this type of information is a typical task of traditional PDM systems and will therefore be handled by the PDKM system in our architecture.

6.5 Communication between the components of the architecture

Fig. 6.1 illustrates the main components of the proposed information system architecture, i.e. PEID, PDKM and DSS. These different components need to communicate with each other through a public interface that declares the methods or messages each component understands. Examples of information that needs to be exchanged are field data, time intervals for transmitting field data, decision support for filtering field data etc.

The PEID is typically a single computing device but it is also possible that the PEID is composed of many different computing devices that need to communicate with each other (e.g. most modern vehicles contain many computing devices for monitoring and controlling different functions of the vehicle). A PDKM may also be located on one single computer, but in many cases product information is stored in PDKMs of many different organisations. As proposed in Fig. 6.1, DSS components will usually be located both in the PEID and in the PDKM. The proposed product information system is therefore divided into many different software components that may be located on different physical computers. A challenge for designing the communication between different software components comes from that the same software components may be in the same computer or in different computers depending on the application.

Software components located in the same computer communicate by calling methods declared in the public interface of each component. For software components located in different computers, a message-based middleware solution (e.g. web service or other XML-based solutions) is proposed. In the middleware case the challenge is how to ensure that both components indeed have the same interface so that they use compatible messages even when the components are programmed by different companies or when they use different versions. Message and interface standardisation is one solution to this challenge but standards tend to lag behind business needs. This is why the middleware to be implemented here will opt for a „weakly typed“ architecture that leaves some freedom to define and use new messages or modify existing ones according to the needs ot different applications. The advantages and disadvantages of weakly typed solutions versus strongly typed solutions is well known from different programming languages, e.g. weakly type languages such as Basic, Smalltalk, JavaScript versus strongly

typed languages such as C++, Java etc. In practice, such a weakly typed solution should make it possible to use the same middleware solution for communication on all levels of the architecture, i.e. for PEID to PEID, PEID to backend and for backend to backend system communication.

7. Summary and outlook

Product embedded systems enable the collection and processing of real world data. This allows timely responses locally as well as improved planning for a group of products in backend applications. The combination of processing in the backend and locally offers the opportunity for dynamic generation and distribution of product knowledge based on the product's behaviour in the field.

We demonstrated how the accuracy of failure prediction could be improved by dynamic updates of product knowledge. This is just one example of how this approach can be applied. Other applications can be imagined, e g. observing and measuring the performance of major product components in different environments to improve the overall product design or configuration.

8. Acknowledgements

Parts of this work were funded by the European Union IST 6th Framework program, project number 507100.

9. References

[1] Främling, Kary, Rabe, Lutz (2005). DR7.2: Concept and methods for information enrichment. Public deliverable of PROMISE project of European Union IST 6th Framework. To appear.

[2] PROMISE (2004). Product Lifecycle Management and Information Tracking using Smart Embedded Systems, EU IST 507100, project web site: http://www.promise.no

[3] Hicks, B. J., Culley, S. J., Allen, R.D., Mullineux, G., A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design, International Journal of Information Management, Vol. 22, Issue 4, 2002, Pages 263-280.

[4] Koç, M. and J. Lee, A System Framework for Next-Generation E-Maintenance System, EcoDesign 2001, Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Tokyo Big Sight, Tokyo, Japan, Dec. 11–15, 2001.

[5] Lee, J., Qiu, H., Ni, J., Djurdjanovi, D: Infotronics Technologies and Predictive Tools for Next-Generation Maintenance Systems. 11th IFAC Symposium on Information Control Problems in Manufacturing. Salvador, Brasil April 5-7th, 2004

[6] Marsh, L., Finch, E. (1998). Using portable data files in facilities management. Facilities, Vol. 16, No. 1-2. pp. 21-28.

[7] Simon, M., Bee, G., Moore, P., Pu, J.S., Xie, C. (2001). Modelling of the life cycle of products with data acquisition features. Computers in Industry, Vol. 45. pp. 111-122.