ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Closed-loop PLM for intelligent products in the era of the internet of things

Dimitris Kiritsis *

EPFL, STI-IGM-LICP, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 30 May 2009
Accepted 9 March 2010

Keywords:
Intelligent products
PLM
Closed-loop PLM
CL2M®
Product lifecycle management
Sensors
RFID
PEID
Product embedded information devices
STEP
Semantic web
Ontology
Internet of things

ABSTRACT

With the advent of the information and related emerging technologies, such as RFID, small size sensors and sensor networks or, more generally, product embedded information devices (PEID), a new generation of products called smart or intelligent products is available in the market.

Although various definitions of intelligent products have been proposed, we introduce a new definition of the notion of Intelligent Product inspired by what happens in nature with us as human beings and the way we develop intelligence and knowledge. We see an intelligent product as a product system which contains sensing, memory, data processing, reasoning and communication capabilities at four intelligence levels. This future generations of Intelligent Products will need new Product Data Technologies allowing the seamless interoperability of systems and exchange of not only Static but of Dynamic Product Data as well. Actual standards for PDT cover only lowest intelligence of today's products. In this context, we try to shape the actual state and a possible future of the Product Data Technologies from a Closed-Loop Product Lifecycle Management (C-L PLM) perspective.

Our approach is founded in recent findings of the FP6 IP 507100 project PROMISE and follow-up research work. Standards of the STEP family, covering the product lifecycle to a certain extend (PLCS) as well as MIMOSA and ISO 15926 are discussed together with more recent technologies for the management of ID and sensor data such as EPCglobal, OGC-SWE and relevant PROMISE propositions for standards.

Finally, the first efforts towards ontology based semantic standards for product lifecycle management and associated knowledge management and sharing are presented and discussed.

1. Introduction

With the rapid evolution of communication technologies such as Bluetooth, WiFi, WiMax, WiBro and relevant communication protocols and the family of IEEE 802 standards and the advent of the information and related emerging technologies, such as RFID, small size sensors and sensor networks or, more generally, product embedded information devices (PEID), a new generation of products called smart or intelligent products is available in the market. Among other innovative features, these products allow monitoring new parameters of the product and its environment along its whole lifecycle. Unlike the actual approach of data creation, management and use which focuses on product type, new emerging technologies allow focusing on product item. This is a new paradigm in the way that it is possible to monitor each single item of a product type. Instead of gathering information for the next version of a product, the gathered data can be analysed and transformed to information to knowledge which can then be used to optimise the whole lifecycle including end-of-life of a product. This is allowed by the track and trace capabilities of the RFID technologies and the condition (health) monitoring capabilities provided by the sensor systems. New business opportunities and new technological challenges have been created by this new generation of products.

For many applications of mobile and wireless condition monitoring, the deployment of RFID or sensor system alone is insufficient and the integration of all data sources is required. Moreover, because of the large amount of data and in order to guarantee the compatibility with the biggest amount of existing systems, an appropriate use of standards is needed. Standards are the common "language" to share data between organisations located at different places. Actual gaps concern the integration of the sensor data with the RFID data which is not supported by the EPCglobal standards, the de facto standards for RFID and the integration of these data in product development and the PLM systems used for that purpose. The conceptual sensor data integration has been well defined by the academic research, for example in the classification approach of RFID based sensor integration [1].

In parallel with the development of RFID standards and sensor standards, some research works have been done to cover the specific requirements of the intelligent products. This is the

* Tel.: +41 21 6935163, +41 79 5938711 (Mobile); fax: +41 21 6933553.
E-mail address: dimitris.kiritsis@epfl.ch.

0010-4485/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2010.03.002

case of PROMISE, an EU-funded FP6 project, which considers the product lifecycle and the new business opportunities relative to the technological improvements. In particular, PROMISE provides a mean to transfer critical information about a product back to the earlier design and forward to appropriate intervention area, e.g. a dismantler [2–6].

Up to now, there is a lack in the exploitation of the data that can be gathered through the whole lifecycle of an intelligent product. There are standards that can support either RFID based identification or sensor systems but there are gaps remaining for the sensor data integration and for the data analysis and their integration in product development and PLM solutions.

Relevant Product Data Technologies (PDT) are usually discussed in Special Issues of the Journal of Computer-Aided Design. The most recent one [7] surveys and evaluates two to three decades of PDT developments, giving a prominent place to STEP. Advanced functional applications of STEP and PDT are presented, with emphasis on processes to be PDT-supported. Some of the papers are discussing required extensions to existing standards. But, "remarkably enough, none of the papers casts off a clear methodological landscape or architecture of future PDT, and none outlines exactly how STEP, with Internet, is to evolve from being application-bound and document based to computer interpretable, self-contained and semantic".

In this paper, and with the advent of the more and more development of the so called "intelligent products", we will try to explore these missing elements.

2. Intelligent products and the internet of things (IoT)

The so called "Intelligent Products" and "Smart Products" – these two terms can be used interchangeably – are meant to be used in the context of the new era of the Internet of Things (IoT) which may be defined as a global network infrastructure where physical and virtual objects with unique ID are discovered and integrated seamlessly (taking into account security and privacy issues) in the associated information network where they are able to offer and receive services which are elements of business processes defined in the environment they become active [8].

The notion of intelligent product is still rather undecided. A first question to answer is what we consider to be a product. Is it simply an item of commercial value — whether consumable or reusable (asset)? Then what are the elements of intelligence that may be associated to a product?

Answers to these questions may be found in a recent Special Issue on Intelligent Products of the Computers in Industry journal [9]. As the guest editors of this special issue say "in this area, science fiction is way ahead of the research world, with talking cars, humanoid robots and smart vacuum cleaners setting the pace for the intelligent product field. Although many academic and technical challenges remain, perhaps the greatest challenge in this space is in demonstrating that intelligent products are not simply gimmicks but that they have a valuable and useful role to play in a more energy- and material-efficient, cleaner society".

Let us start with the most recent definitions of intelligent products.

2.1. Recent definitions of intelligent products [9]

In the above mentioned special issue on Intelligent Products of the journal Computers in Industry, a thorough review of various issues around the subject of intelligent products are discussed and the following three definitions are presented:

2.1.1. Definition 1 by [McFarlane et al.]

McFarlane et al. define an Intelligent Product as a physical and information based representation of a product: the physical product, the information based representation of the product is stored in the database, and the intelligence is provided by the decision making agent. The connection between the physical product and the information based representation is made using a tag and a reader.

According to McFarlane et al., an Intelligent Product has the following properties:

1. Possesses a unique identification.
2. Is capable of communicating effectively with its environment.
3. Can retain or store data about itself.
4. Deploys a language to display its features, production requirements, etc.
5. Is capable of participating in or making decisions relevant to its own destiny.

2.1.2. Definition 2 by [Kärkkäinen et al.]

The fundamental idea behind an Intelligent Product according to Kärkkäinen et al. is the inside-out control of the supply chain deliverables and of products during their lifecycle. In other words, the product individuals in the supply chain themselves are in control of where they are going, and how they should be handled. To move to inside-out control of products, the products should possess the following properties:

1. Globally unique identification code.
2. Links to information sources about the product across organisational borders, either included in the identification code itself or accessible by some look-up mechanism.
3. Can communicate what needs to be done with them to information systems and users when needed (even pro-actively).

2.1.3. Definition 3 by [Ventä]

Ventä refers by intelligence to products and systems that:

1. Continuously monitor their status and environment.
2. React and adapt to environmental and operational conditions.
3. Maintain optimal performance in variable circumstances, also in exceptional cases.
4. Actively communicate with the user, environment or with other products and systems.

Having the above in mind, we introduce here a new definition of the notion of Intelligent Product inspired by what happens in nature with us as human beings and the way we develop intelligence and knowledge. We see an intelligent product as a product system which contains sensing, memory, data processing, reasoning and communication capabilities at various intelligence levels as follows:

Intelligence Level 1: physical products without any embedded system (device or software). They do not interact with their environment. Examples: an axis of a machine, a bottle in its present form.

Intelligence Level 2: physical products with embedded simple sensors. Their embedded sensors allow them to interact with their environment. Example: a refrigerator with a thermostat. The thermostat allows the refrigerator to adapt its internal temperature to a desired level.

Intelligence Level 3: physical products with embedded sensors, memory and data processing capabilities. Such system of embedded devices and software (data processing) allows a product to adapt quickly to sophisticated changing environments. Example: a car with an ETS system. Such a car is able to adapt its trajectory to suddenly changing road conditions such as ice on the road, a sudden obstacle etc.

Intelligence Level 4: physical products with Product Embedded Information Devices (PEID). Examples of PEIDs are ID devices such as RFID tags, sensors or sensor networks, on board computers etc. The additional element at this level is that identification and communication capabilities are now added to the product characteristics, thanks to RFID, NFC (Near Field Communication) Wireless Sensor Networks and related emerging IoT technologies. Examples: airplanes with tagged components and sensor networks that allow for Predictive Maintenance.

Products of Intelligence Level 1 are the ones whose design and development needs are covered by actual PLM systems and related PDT standards such as STEP. Exchange of Static Data (dimensions, BOM, other design parameters) only is possible at this level.

In some science fiction stories and movies we may observe that products of Intelligence Level 4 (anthropoid robots) may develop "wisdom" capabilities, since they have (i) "identity" and, in addition to that, with their computing capabilities they may (ii) develop reasoning at various levels of decision making, from local to distributed, (iii) communicate to each other and with their environment and (iv) keep track of their history. These four elements are similar to the elements which allow us humans to develop our own "intelligence" and "wisdom": from the instance we are born, we first learn with our senses and then we develop our own identity (personality) through our advanced communication capabilities (first with signals, then with language) with our close family and our societal environment in general. We are capable of remembering past events (memory) and we are able to associate them with actual facts (field data) and reason on them (processing of data and information). This is the way we develop knowledge and this process is of course amplified through the actual educational systems. The whole environment allows us developing what we call wisdom.

In an analogy to that, intelligent product of Intelligence Level 4 having sensing (with sensors and wireless sensor networks), memory (with micro–nano memory chips), processing (with micro–nano processors), communication (with multi-agent system technologies) and identity (with PEID technologies) capabilities all along their lifecycles would be able to assess changes to themselves and to their environment and thanks to IoT technologies communicate with their peers (other products) and their environment develop own knowledge and, to a certain extent, self reasoning and decision making capabilities. Certainly, it would be quite provocative to say that future intelligent products will develop wisdom capabilities. Let us leave this aspect to science fiction movies.

The future generations of Intelligent Products will need new Product Data Technologies allowing the seamless interoperability of systems and exchange of not only Static but of Dynamic Product Data as well.

Dynamic Product Data concern characteristics of products that change along time and through the lifecycle of a product through a series of "events". Examples are wear after use, repair or replacement of components after functional degradation and associated failures and breakdowns, functional changes due to degradation or other environmental phenomena, etc.

In this context, we will try to shape the actual state and a possible future of the Product Data Technologies from a Closed-Loop Product Lifecycle Management perspective.

3. Closed-loop PLM

Product Lifecycle Management (PLM) aims to manage the product related information efficiently during the whole product lifecycle. PLM expands Product Data Management's (PDM) scope to provide more product-related information to the extended enterprise. Product Data Management has been developed to improve the management of data and documented knowledge for the design of new products and focus on the design and production phases of a product. Moreover, information is related to product type and not to product items.

PLM is a strategic approach and has three fundamental dimensions: (i) universal, secure, managed access and use of product definition information, (ii) maintaining the integrity of that product definition and related information throughout the life of the product or plant and (iii) managing and maintaining business processes used to create, manage, disseminate, share and use the information. In order to face the new industrial challenges such as environmental changes (mass customization, short product development lead time, high complexity of product functions), increasing collaboration (intra-enterprise, extended supply chain), importance of growth of Web based paradigms and technology, digital manufacturing, customer service as well as decision support services, the information loops along the whole product lifecycle have to be closed. The stakeholders have a general desire to enable the seamless flow, tracing and updating of information about a product, after its delivery to the customer and up to its final destiny (decommissioning, deregistration and EOL) and back to the designer and producer.

We may categorise a product lifecycle in the following three major phases:

- BOL including conceptualisation, definition and realisation
- MOL including use, service and maintenance
- EOL characterised by various scenarios such as: reuse of the product with refurbishing, reuse of components with disassembly and refurbishing, material reclamation without disassembly, material reclamation with disassembly and, finally, disposal with or without incineration.

As mentioned above, data, information and knowledge are created in all these phases. Between the first two stages of BOL (i.e. design and production), creation of data, information and knowledge is supported by intelligent systems such as CAD/CAM/CAE and other simulation software. Product Data Management (PDM) is effectively and efficiently used by many OEMs and, through their influence, by their suppliers. Such effective use becomes less and less frequent and complete moving from the MOL phase to the final EOL scenario. For the majority of today's technological products and especially for the 'hi-tech' ones (such as consumer electronics, household 'white' machines and vehicles), it is fair to say that the data, information and knowledge flows and associated transformations break down after the delivery of the product to the customer.

3.1. Closing the information loops

The fact that these flows are in most cases interrupted shortly after product sale prevents the feedback of data, information and knowledge, from service and maintenance and recycling experts, back to designers and producers. Design methodologies such as: conceptual design; design for use, design-for-manufacturing, design for assembly, design-for-service and design-for-environment – generally, collectively defined as design for X – depend upon reverse information flows to produce more competitive and sustainable products.

There is a general desire of many stakeholders in the product supply and value chain (from designers to users and recyclers) to enable the seamless flow, tracing and updating of information about a product, after its delivery to the customer and up to its final destiny (decommissioning, deregistration and EOL) and back to the designer and producer. This is illustrated in Fig. 1, where dashed thick lines represent material flow along the product

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

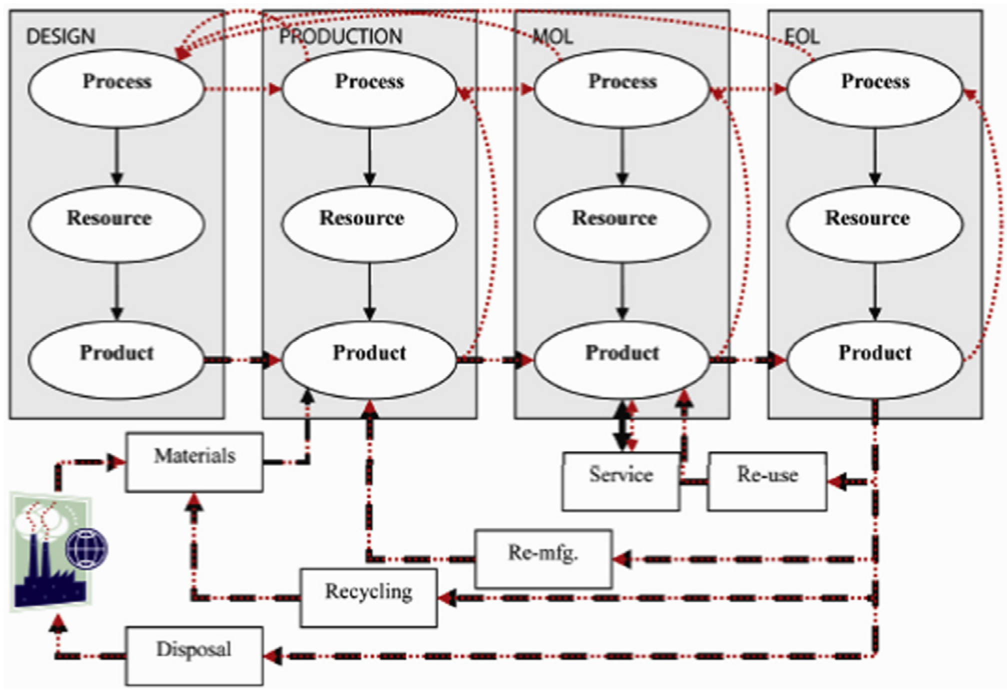


Fig. 1. Closing the information loops.

lifecycle including 'recycling' loops, while dotted lines represent information loops [10].

The objective here is to allow information flow management to go beyond the customer, to close the product lifecycle information loops, and to enable the seamless e-transformation of product lifecycle information to knowledge. Closing of the product lifecycle information loops will have the following consequences [11]:

1. producers will be provided with complete data about the modes of use and conditions of retirement and disposal of their products
2. service and maintenance and recycling experts will be assisted in their work by having:
   - a complete and always up-to-date report about the status of the product
   - real-time assistance and advice through the internet
3. designers will be able to exploit expertise and know-how of the other players in the product's lifecycle and thus improve product designs towards product lifecycle quality goals
4. recyclers/reusers will be able to obtain accurate information about 'value materials' arriving via EOL routes.

3.2. *The concept of closed-loop PLM*

A closed-loop PLM system will allow all the actors who play a role during the lifecycle of a product (managers, designers, service and maintenance operators, recyclers, etc.) to track, manage and control product information at any phase of its lifecycle (design, manufacturing, MOL and EOL), at any time and any place in the world. The closed-loop PLM concept shown in Fig. 2 shows the requirements for the technologies to be investigated and developed [53].

The main elements of the closed-loop PLM concept and requirements shown in Fig. 6 are:

- local (short distance) connection mode for product data and information exchange
- internet (long distance) product information and knowledge retrieval
- data and information flows
- decision support software.

The above concepts and requirements compose what is referred to as seamless e-transformation of data to information to knowledge.

Fig. 2 explains the basic principle of business operations in closed-loop PLM. Although there are a lot of information flows and inter-organisational workflows, the business operations in closed-loop PLM are based on the interactions among three organisations: PLM agent, PLM system, and Product. The PLM agent can gather product lifecycle information from each product at a fast speed with a mobile device like a personal digital assistant or a laptop computer with a Product Embedded Information Device (PEID) reader. He sends information gathered at each site (e.g. retail sites, distribution sites and disposal plants) to a PLM system through the internet. The PLM system provides lifecycle information or knowledge created by PLM agents whenever requested by individuals or organisations. The above concept can be used for product lifecycle KM and can be partitioned into three phases: BOL, MOL and EOL.

More recently the concept of closed-loop PLM was further generalised to the concept of CL2M® (Closed-Loop Lifecycle Management) [12].

4. Enabling technologies for the development and lifecycle management of intelligent products

In this section, concepts and technologies used or to be used by Intelligent Products as defined above developed in PROMISE are presented. The details are available in the PROMISE Architecture Series, volume 1 to 5 [2–6].

4.1. *Product identification technologies*

There are several technologies to identify products. In this chapter, the focus is on RFID systems and Barcode technology.

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

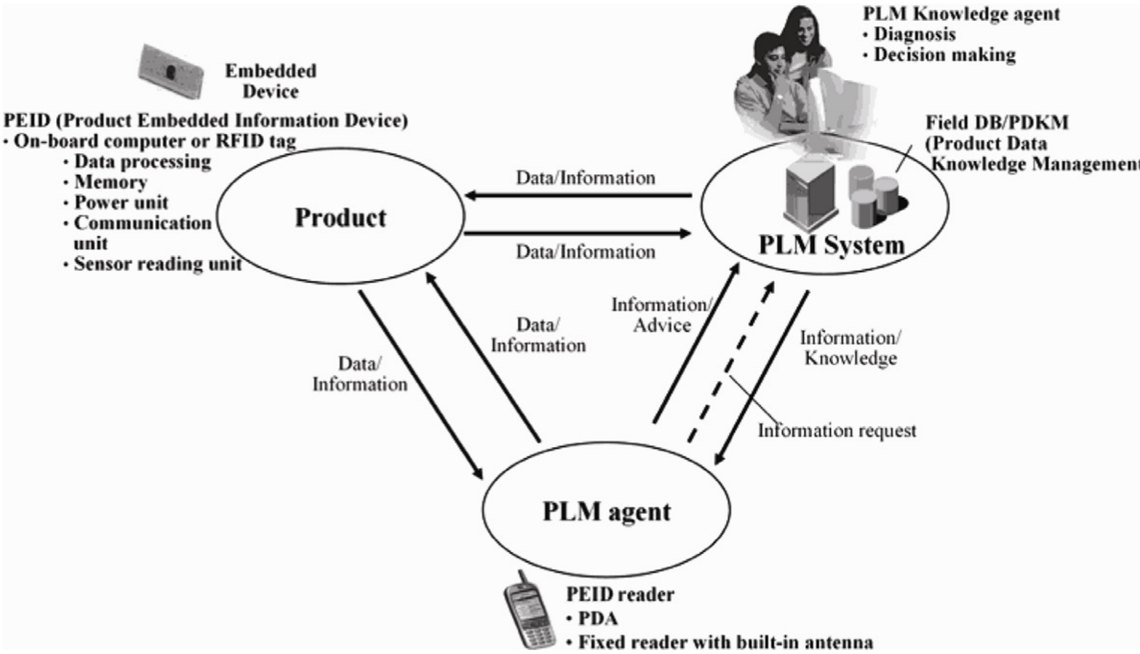


Fig. 2. The closed-loop Product Lifecycle Management concept.

0 36000 29145 2

Fig. 3. 1D barcode [http://en.wikipedia.org/wiki/Radio-frequency_identification].

Fig. 4. 2D barcode [http://en.wikipedia.org/wiki/Radio-frequency_identification].

4.1.1. Barcode technology

Developed during the 1970s, barcode technology is an optical technology to store identification data. One-dimensional barcodes (Fig. 3) store data in the widths and spaces of the parallel lines whereas two-dimensional barcodes (Fig. 4) store data in patterns of dots or concentric circles.

Barcodes are read by scanners and are supported by standards such as UPC-A or UPC-E in the USA and EAN-13 in the rest of the world [http://en.wikipedia.org/wiki/Radio-frequency_identification].

The main limitations of barcodes are the read-only property, which means that once printed, the data cannot be changed, and the need of direct line-of-sight access by scanners.

4.1.2. RFID systems

Radio Frequency Identification (RFID) systems provide direct object identity sensing. They are composed by three main elements that are RFID Tags, Readers and data processing subsystems. A RFID Tag is basically a memory chip, a processor and an antenna attached to the object to be identified. Several frequencies can be used to transfer that data and there is a wide variety in RFID tags, notably concerning power, memory or processing capabilities. The communication is done by radio waves between the tag and the reader. Data are then processed in a subsystem such as software applications.

The simplest and cheapest Tag has no built-in battery and it uses the energy of the radiation emitted by the reader. It is called a passive tag. Unlike passive tags, active tags have a built-in battery and can provide optional features such as environment sensing.

RFID systems allow tracking and tracing objects within a supply chain. For example, it is especially useful to notify the location change or the transhipment processes. The technology is supported by the EPCglobal family of standards. Some limitations of RFID include limited read range or detuning and attenuation effects due to materials such as water and metals [13].

4.2. Lifecycle monitoring

Lifecycle monitoring is needed to allow closed-loop PLM. Sensor systems are used to measure several parameters such as environment (temperature, pressure, humidity) or movement (velocity, acceleration, shock). The quality of this data is limited by the properties of the sensors. Their spatial coverage (range) and measurement accuracy (resolution) is subject to physical limitations, they can measure a restricted number of state variables and sensors might fail and are subject to noise [15]. Sensor systems have typically two layers, the sensor which produce

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

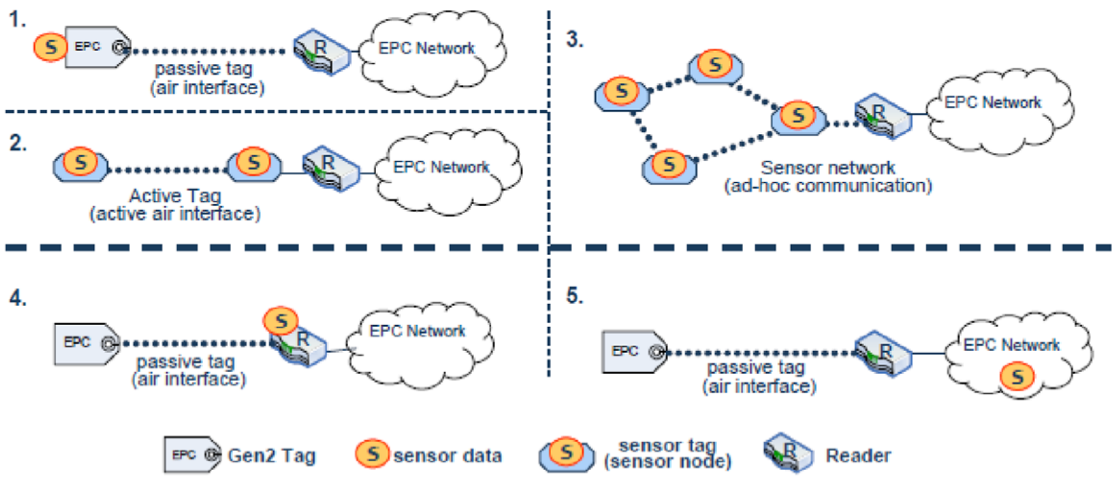


Fig. 5. Modes of identification and sensor data integration [14].

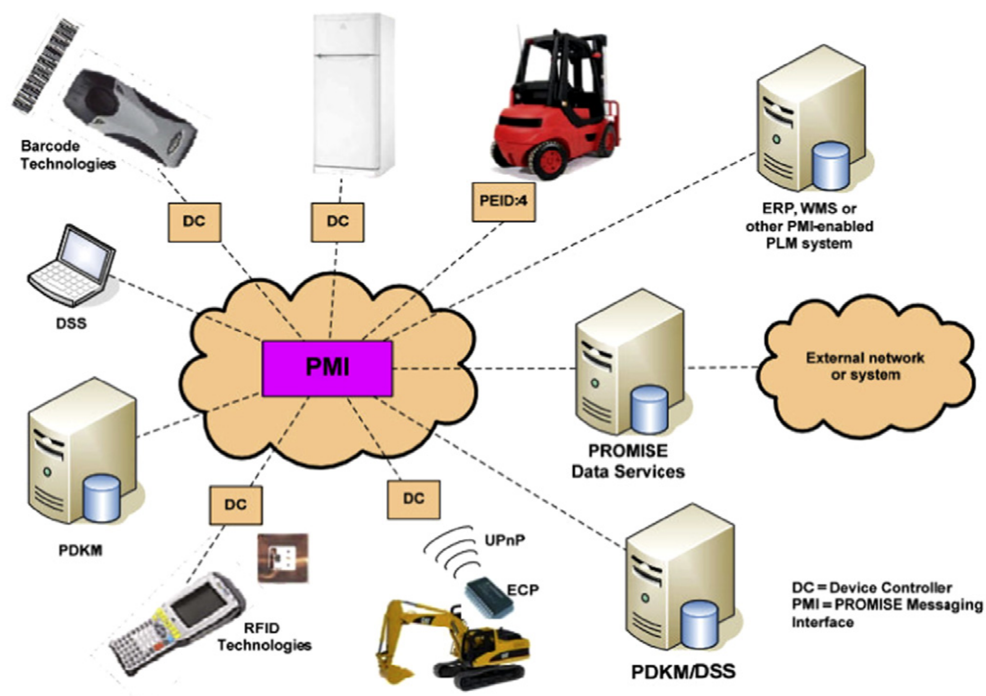


Fig. 6. PROMISE connectivity [2].

measurement and an interpretation layer which transform raw data into information. The processing of sensor data depends of the type of data, for example scalar or two-dimensional. This processing may be done schematically closely-coupled if the sensor has computing capabilities or raw data can be transmitted toward a processing unit [1].

4.3. Integration of sensor data to RFID data

The integration between sensor data and RFID data is a requirement of closed-loop PLM. The decision making process of PLM application requires to know the product history and in many applications sensors and RFID systems are not closely-coupled. The way sensor data and RFID data are integrated depends on the application. Former research works have classified the integration of sensor data. The Auto-ID Labs white paper [16] provides a fundamental classification of sensor data integration and architectures of sensor integrated EPCglobal network. In Fig. 5, five modes to integrate sensor data and RFID data which are accessed via the EPCglobal architecture are presented.

There are several hardware solution features to the design of RFID based sensor integration systems. A classification approach in [1] provides a means to classify these solutions. Locality of sensing (On-board or Off-board condition monitoring), deployment of sensor tags or independent tag and sensor devices, number and spatial distribution of sensor devices of the same type, degree of object localization and type and role of RFID reading systems are the different hardware solution features.

4.4. PROMISE technologies

PROMISE architecture is a middleware centric architecture (Fig. 6) that includes several components. These components can be represented in a layered schema as shown in Fig. 7.

This architecture and the concept of Closed-Loop PLM shown in Fig. 2 have been implemented in ten demonstrators using the

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

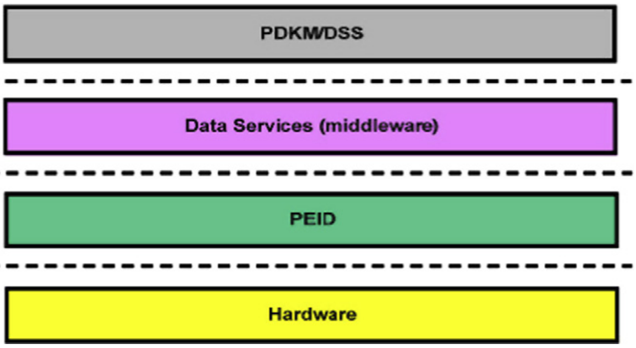


Fig. 7. PROMISE components [2].

components presented below. Web service technologies have been used with success in these prototype realisations. One of the conclusions of the experience gathered with these demonstrators is that "internet of things" technologies including the so called "discovery services" will facilitate significantly industrial implementations of these concepts and the further deployment of the proposed technologies in extended applications.

A PEID (Product Embedded Information Device) is the link between a product and its electronic representation in an information system. Depending on the PEID type, the function can be either product identification (a PEID that contain only the Global Unique ID) or data sensing, capturing and storage. There are 5 types of PEID, according to [2]. Type 0 refers to identifier only PEIDs, such as barcodes whereas type 1 refers to identifiers with data storage capabilities. Type 2 and 3 refer to PEIDs with low or medium computation power, respectively. Finally, type 4 PEIDs have sufficient computation power to connect to the middleware of PROMISE through web services [2].

PROMISE Data Services is the middleware component of the PROMISE architecture that connects together the different systems in a PROMISE infrastructure, and provides the means for communicating and gathering product data between them. The following tasks are handled by PROMISE Data services: (i) location of correct data sources; for example, where will a PEID connect next time or in which Product Data and Knowledge Management (PDKM) system is the data stored, (ii) buffering and aggregating requests; for example, one request from the PDKM may be split into several requests destined for several different nodes, and then aggregated back into a single response, (iii) device management, integrating proprietary Device Controllers (DC) and PEID communication methods, (iv) locating metadata and (v) event communication; events such as PLM events, alarm events, etc. To do that, PROMISE Data Services issues PROMISE Messaging Interface (PMI) requests such as Device Requests, various PLM events etc. It provides also a means for routing messages between Data Services nodes, a means of discovering other Data Services nodes. It keeps track of device connectivity status and device location status to make sure device requests can reach devices/PEIDs, keeps track of system connectivity status and system roles in the Data Services network [3].

The implementation is logically divided in two parts: a lower layer, the Device Controller (DC), that handles the hardware interface towards readers, PEIDs and other ID devices such as RFID, and an upper layer, called Inter System Communication (ISC) that connects different DCs, backend systems, PDKM systems, etc and handles the communication between these systems [2,4].

PROMISE Messaging Interface The PROMISE Messaging Interface (PMI) is a generic interface that may be used by any component of the PROMISE architecture, except PEIDs from a lower class than class 4; see Fig. 8.

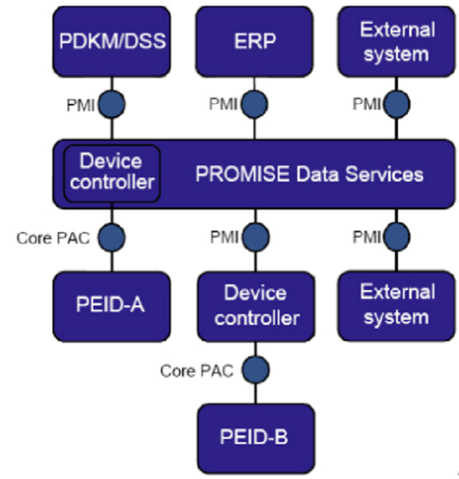


Fig. 8. PROMISE components and interfaces [2].

PMI is a flexible interface that allows data exchange between any pair of PMI users. Different message types are used in PMI communication: Field Data Request, PLM Events, Metadata Request, System Management Events, Device Management Events and Alarm Events [2].

PROMISE PDKM/DSS (Product Data and Knowledge Management system/Decision Support System) incorporates data from several different software systems, e.g. legacy CAD, CRM and/or SCM systems as part of a company's IT infrastructure. Consequently, if data from these systems is required for generating specific information to support a decision, it should be made available through the PDKM and associated DSS system.

The PDKM aims to systematically integrate and manage data from all product lifecycle phases. The ultimate goal is to integrate product data throughout the entire lifecycle from different sources, to support comprehensive analysis on such data so enabling the enhancement of operational businesses through the more detailed insight on products thus obtained [5].

A PDKM System Object Model (semantic model) has been designed with the aim of providing the system with a basis for representing product data throughout the whole lifecycle. This model can be separated in two main areas, see Fig. 9. A first one, within the solid-line frame, contains information such as serial number, product type, product structure, etc. The loop of this basis information has to be closed within the PROMISE architecture. In addition, this area also describes the product as a product type. The second area, within the dashed-line frame, aims at modeling the pieces of information connected to the different lifecycle phases [2,5].

5. Standards for closed-loop PLM of intelligent products

In this chapter we present the main existing standards or open solution to cover the lifecycle management of a product. Some of them propose only a data transfer protocol whereas some others are complete solutions.

5.1. Requirements for closed-loop Product Lifecycle Management

Product information is generated during all the lifecycle phases. At the design stage, information can be distributed between several organisations and is generally of different types of documents, that is, for example brain storming reports, design documents or bill of materials. In the manufacturing process, parts and subassemblies can be produced by different manufacturers. During the product transit, there might be some relevant data (e.g. sensor

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

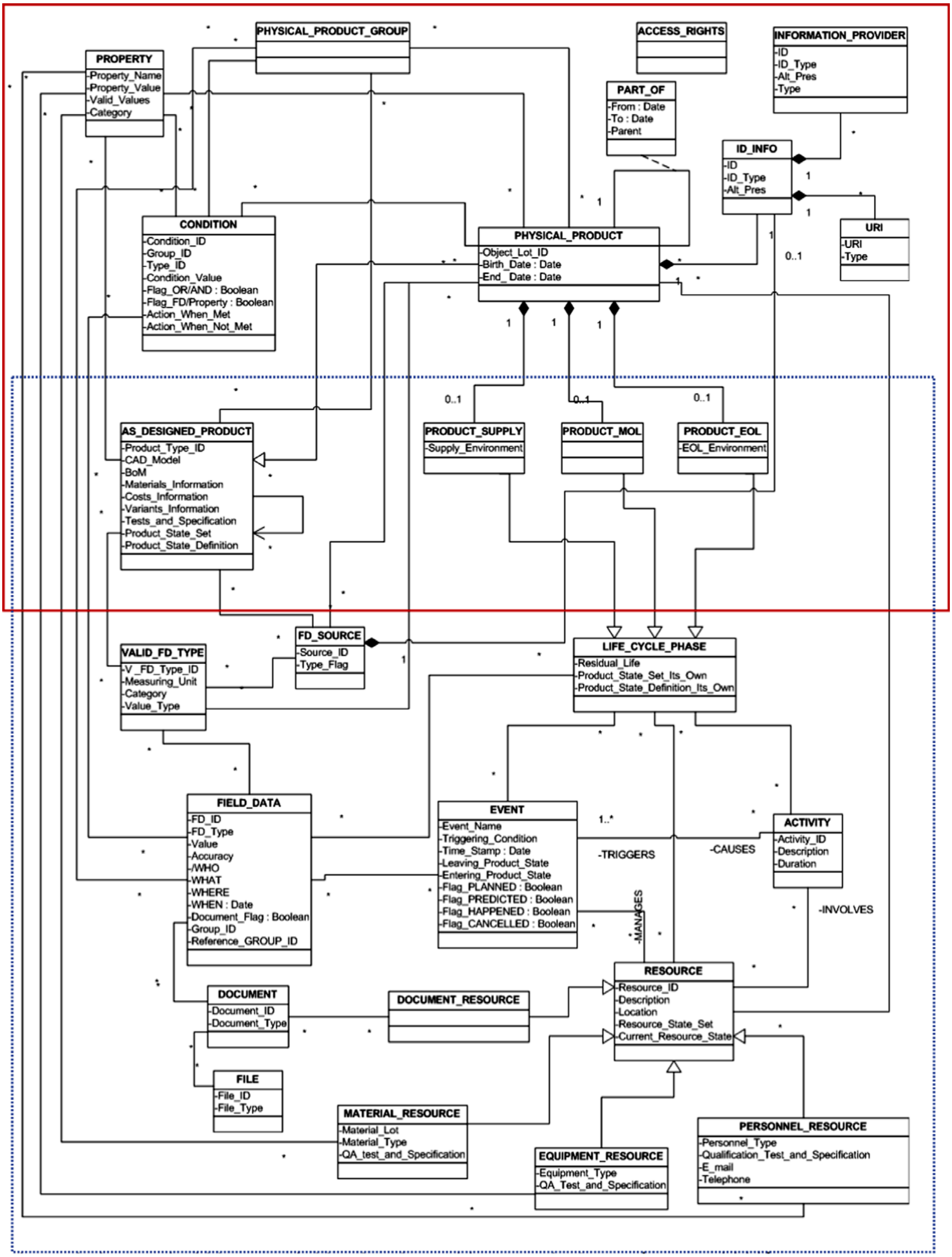


Fig. 9. Semantic object model [5].

Table 1
Summary of functional requirements [extended from [17]].

Lifecycle phase	Functional PLM requirements
BOL	Globally unique identifier for objects and the storage of object related quasi-static data Access to previous MOL lifecycle information on product usage Access to previous EOL lifecycle information on recycled parts Product design improvements using MOL and EOL data New product design generation using MOL and EOL data
MOL	Object usage information capture Access to component relevant information from BOL Data analysis functions — DSS for predictive maintenance Capture maintenance events
EOL	Track and trace Access to component relevant information from BOL and MOL Data analysis functions — DSS for EOL management

data about vibrations) collected while the product is between the manufacturer and the customer. When the product is sold to a customer, the usage phase begins and the product alternates between work phases and maintenance or repair phases. The product reaches then its EOL and the gathered information is needed to take the EOL decision such as reuse or recycle a component.

Based on the analysis of the scope of the PROMISE project and of the objectives of its demonstrators, the requirements for PLM have been identified. Functional requirement are summarized in Table 1.

Based on this functional analysis, requirements for information management can be extracted. Seven requirements have been identified, that is, globally unique identifier, routing or lookup mechanisms, information resources, timely information, synchronisation, reconfigurability and data analysis. A brief description of these requirements is given below.

To link a product with its related product information, the use of a unique identifier is required. It allows discovering and accessing the information to multiple organisations. It is necessary to provide a means to identify each single item and not only a product type. This is particularly needed at the EOL in order to decide which parts are worth reusing or recycling and which are not.

Discovery, routing and lookup mechanisms are needed to bridge the gap between a product and unique identifier and the information resources linked to that object. Discovery refers to discovering what is in the neighborhood, routing refers to the transmission of data to a well-defined recipient and lookup is used to retrieve the source of the data.

Information resources may fall into various categories such as BOL information (e.g. geometry, assembly information or CAD files), human readable such as photos, lifecycle monitoring data of a product (e.g. historical records of its location or temperature) or information services related to the object (e.g. interactive instruction manual, diagnostics tools).

A PLM system is expected to provide information in a timely manner. This is necessary to reflect in the information world the changes that happen in the physical world. Predictive maintenance and MOL information based design are two applications which require timely information.

Due to intermittent access of products to network resources, synchronisation of remote events is needed. Conflicts of validity should be resolved seamlessly and the correct sequencing of data and event updates has to be guaranteed.

Reconfigurability should be possible to manage the architecture of the information system. For example, hardware or databases changes are needed and should be supported, while ideally maintaining the same interfaces to internal/external clients. The existing standards will be described and analysed in the next paragraphs regarding the requirements exposed above.

5.2. Existing standards and architectures

A first observation that can be concluded is that actual STEP standards cover product development for Intelligence Level 1 type of products because it is about the static product definition data only.

On the other hand, several standards cover partly the requirements of the closed-loop PLM: PLCS and ISO 15926, MIMOSA, EPCglobal standards, etc. To make it clearer, a hierarchical typology of PLM support standards is presented [18]. Type Zero standards are standards for implementation languages, for example FORTRAN, C++ or Java. They are not considered in this thesis. Then, Type One standards are information modeling standards. EXPRESS, UML, XML are three examples. Type Two includes content standards. This hierarchical step can be separated into product information modeling standards (e.g. ISO 10303 (STEP, PLCS), MIMOSA), information exchange standards (e.g. SOAP), product visualisation standards, E-business and value chain support standards and finally security standards. The last three categories are less relevant regarding the scope of this project. The last type of PLM support standards is Type Three. It includes Architectural frameworks standards, for example EPCglobal.

The analysis focuses on type two and three. Type three standards or open solutions are EPCglobal architecture, DIALOG system, WWAI and OGC SWE and the new PROMISE propositions. Type two standards which are analysed are PLCS and MIMOSA OSA-EAI and CBM.

5.2.1. EPCglobal network architecture

The EPCglobal network is based on Auto-ID technologies. It is an infrastructure that aims to link each physical object to the global Internet to enable visibility within supply chains. The architecture has got five layers, that is, RFID tags, RFID tag readers, the filtering middleware (ALE), EPC Information services (EPCIS) and lookup services such as Object Name Services (ONS) and Discovery Services (which are currently under development); see Fig. 10.

The EPCglobal network uses Electronic Product Code as a unique identifier. Each company receives an EPC Manager number while a central management ensure the uniqueness of each EPC manager number allocation. The Object Name Service (ONS) and the Discovery services provide the lookup mechanisms. The EPCIS standardised interfaces allows uniform access to the information resources and the product-related information. Timely information is provided by the push mechanism of the EPC architecture (the filtering of information into events and the publishing process). The architecture can be reconfigured easily. The layered architecture provides the flexibility in insulating each layer from each other. Data gathering is enabled via Discovery Services. In conclusion, EPC network architecture matches almost every requirement but the synchronisation of data and the management of sensor data.

5.2.2. DIALOG system

DIALOG system is an open source solution developed at the Helsinki University of Technology. It was originally developed for tracking of shipments. The system supports the transmission of other data such as sensor measurements, as well (Fig. 11).

Two components, a unique ID string and a URI (uniform resource identifier), compose the unique identifier. This system (ID@URI) is similar to the e-mail address system. The routing mechanisms depend on the type of the URI. In case of a URL (uniform resource locator), DNS, web services or web server could provide routing mechanisms. If the URI is a URN (uniform resource name), no specific routing mechanism is provided. Information resources are accessed through an agent hosted at the URL. No data

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

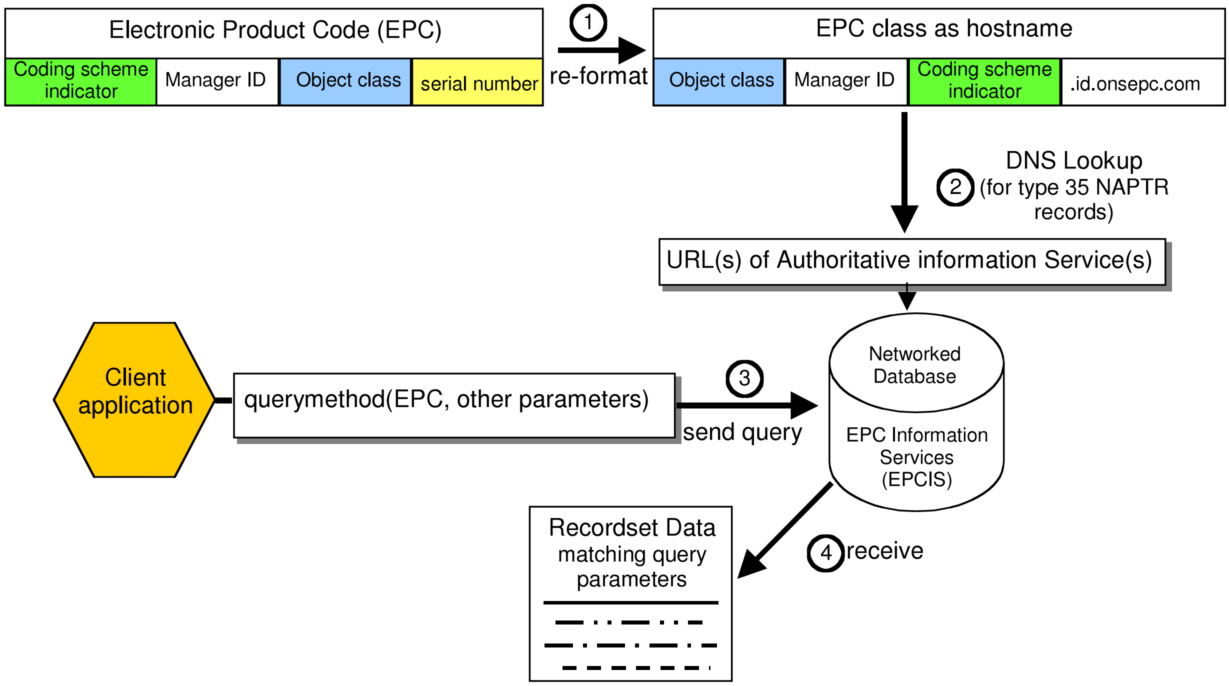


Fig. 10. EPC network architecture [19].

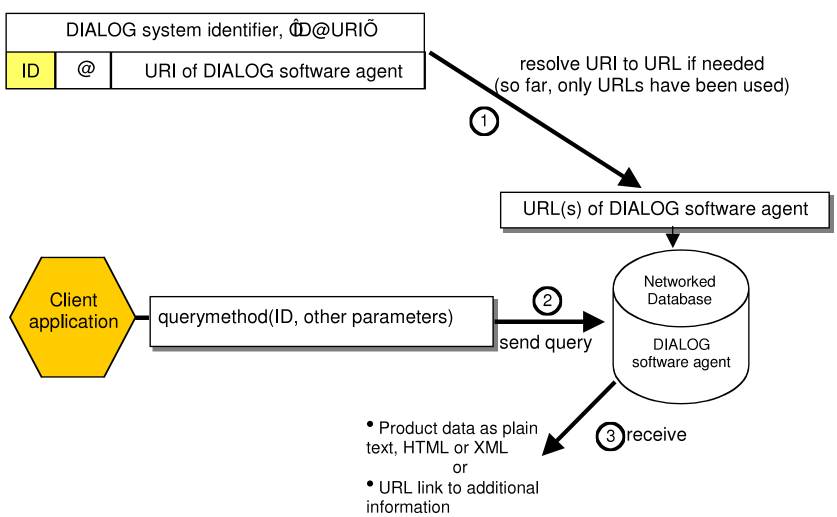


Fig. 11. DIALOG architecture overview [19].

model is defined. Timely information is supported through pushed events. Synchronisation is not supported and reconfigurability is limited by the fact that the network is not resilient to changes in the URL of the information provider [19].

5.2.3. *WWAI network*

WWAI is a peer-to-peer RFID middleware solution. It is owned by Trackway Oy of Finland.

The unique identifier is the WWAI identity code. It consists of a prefix (which identifies the company) ant the remainder. There is no central storage of the information but storage by each information provider. A dynamic virtual map is used instead of a centralised Object Name Service (ONS). A search and a joining mechanism provide a means to access relevant information. The system is reconfigurable and accepts changes in information provider address. Events subscription and transfer are supported. The synchronisation is not supported and there is no formal data analysis layer [19] (Fig. 12).

5.2.4. *OGC SWE*

OGC SWE stands for the standards for Sensor Web Enablement (SWE) developed by the Open Geospatial Consortium (OGC). The scope of these standards is wide, from data encoding to sensor alert services. Fig. 13 presents all standards and their relationships, followed by a description of the most important standards of the family, which are SensorML, TML, O&M and SOS.

Sensor Model Language (SensorML) defines processes and processing components associated with the measurement of observation. Several tools are available for the description of sensors, the processing and analysis of the sensors observations. Sensors and other components (e.g. detectors, transmitters, actuators and filters) are modeled as processes and SensorML provides an XML model for the basic process called atomic process [20].

The scope of the SensorML is very wide and other conceptual schemas encoding or phenomenon description are available.

Furthermore, SWE provides conceptual models for Simple Data types or Aggregate Data types.

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

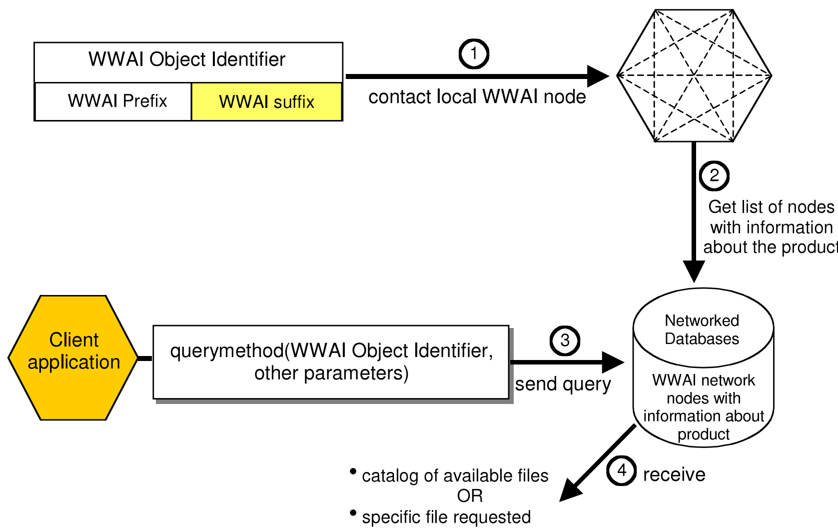


Fig. 12. WWAI architecture overview [19].

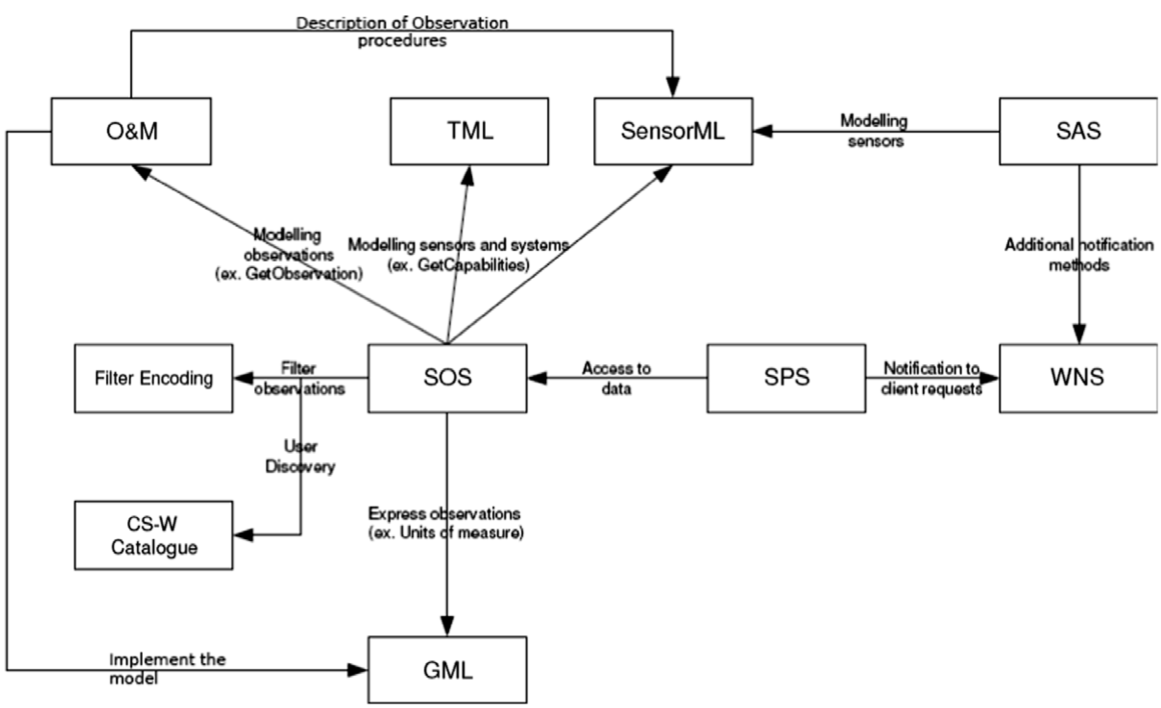


Fig. 13. Relationship between OGC SWE standards [20].

Transducer Markup Language (TML) defines a language for the information necessary to understand and process transducer data. Conceptual models and XML schemas for describing these transducers and supporting real-time streaming of data to and from sensor systems are provided. A TML description includes several attributes, such as calibration, operational condition or transducer properties. Moreover, critical information to the processing of data such as logical models or transfer function is captured within TML.

The TML documentation contains a generic TML implementation. This is the result of a general "plug and play" philosophy. The aim is to be independent of the object and of the process design methodology [21].

Observation and Measurement (O&M) provides definition to terms used for measurements and observations. The main elements are observation, measurement, result, procedure, feature of interest, observed property, property type and coverage. Other related terms are described, as well.

The basic observation model is given in Fig. 14. The key properties of an observation are its feature of interest, the observed property, the procedure and the result.

The formats given by the O&M documentation are to be used within the Sensor Observation Service (SOS).

Sensor Observation Service (SOS) is the central element of the OGC standards family. It provides a means to manage the sensors and retrieve sensor data. Three basic operations, GetObservation, DescribeSensor and GetCapabilities, provide filtered access to sensor observation, a means to retrieve detailed information about the sensors and access to metadata. Other operations support transactions or secondary tasks, such as description of the feature of interest. The principle of filtering is the definition of property-types to describe the phenomena that interest us. These property-types

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

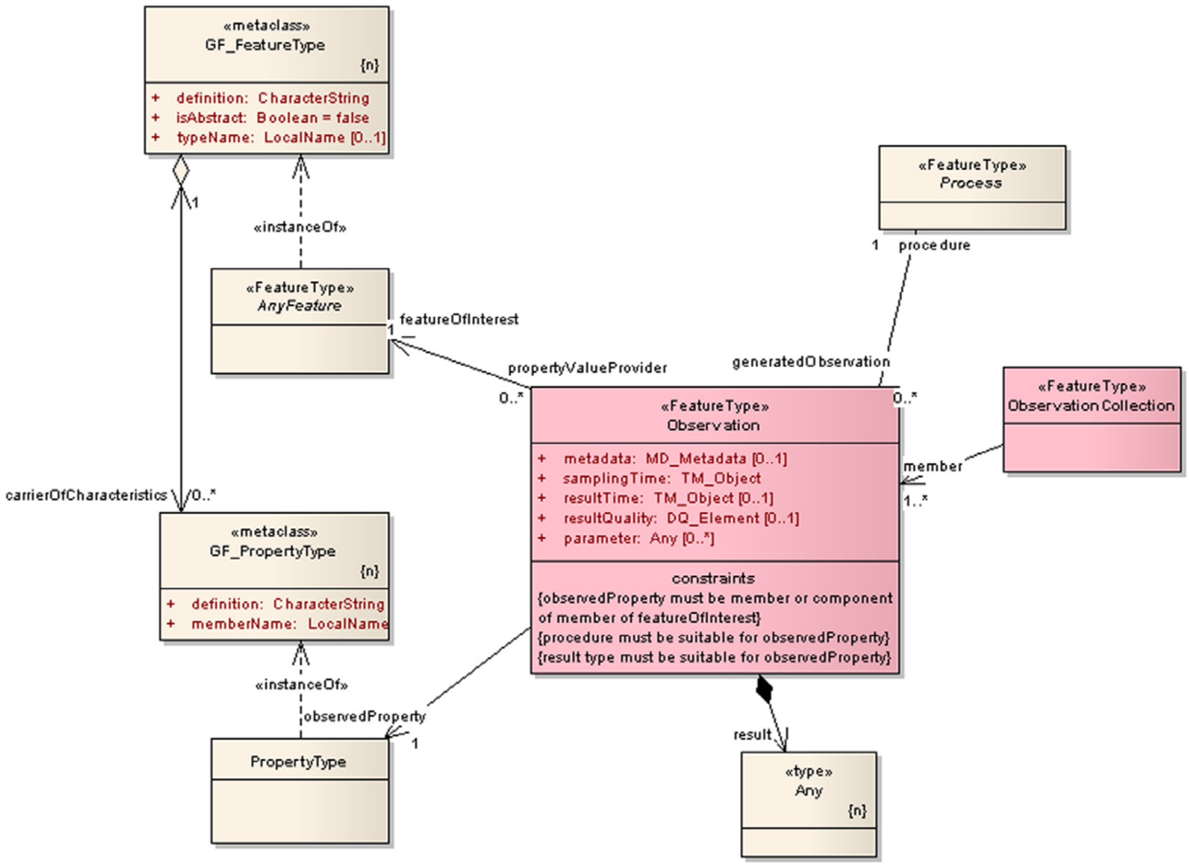


Fig. 14. Basic observation model [22].

can be CompositePropertyType or CompoundPropertyType to describe more complex phenomena [22].

As described, OGC family of standards provides a means to manage sensor data and sensor system and its filtering capabilities might be used to filter complex events. There is no unique identifier provided.

5.2.5. PLCS

Product lifecycle support (PLCS) is an application protocol of STEP (ISO 10303 AP 239). PLCS has been designed to support the maintenance of complex products such as aircrafts or ships in operational conditions [23].

The functionalities of PLCS are supported by Data Exchange Specifications (DEX). They cover a wide scope of activities, such as maintenance plan, operational feedback or system requirements. Each of these DEXs defines business processes and specific reference data. XML schemas derived from STEP are available, as well.

The focus of PLCS is on product types and not on product items. PLCS is a very complex standard and its implementation requires high efforts, because it is not only a way of administrating data. It standardises also the processes and activities during the whole lifecycle. PLCS is not designed to manage a large amount of simple field data and its complexity involves that a use of this standards is less relevant for few complex products.

A PLCS implementation of a Ship Product Model Data for lifecycle support is reported in [24] where an interoperability model is implemented for ship damage control and maintenance using STEP (ISE information models), PLCS and S100D standards (Fig. 15).

In this work, the entire range of product model for design, engineering, production, logistics, lifecycle support (maintenance and repair) and technical publications is used during the processes of ship operation, damage control, maintenance and repair.

5.2.6. ISO 15926 — lifecycle data for process plant

ISO 15926 defines a format for the representation of information about a process plant based on STEP standards [25]. It does not only record the process plant as it exists at a given point of time but also represents how the plant changes after lifecycle activities such as maintenance.

ISO 15926 is named "Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities". It is in fact a fusion of STEP AP 221 and relevant POSC CAESAR Association results on process plant modeling. Although initially designed for the process industry, the standard matured into a widely applicable generic modeling approach that spreads across other industry areas as well. ISO 15926 embraces XML and web services and now seeks to move into the direction of an upper ontology. ISO 15926 consists of 7 parts. Each part has a unique function:

- ISO 15926-1 provides an overview of ISO 15926.
- ISO 15926-2 specifies a generic, conceptual data model that supports representation of all lifecycle aspects of a process plant.
- ISO 15926-4 defines a reference data library that can be periodically updated by a competent body, designated by ISO as a registration authority, which has the requisite infrastructure to ensure the effective use of the reference data library.
- ISO 15926-5 specifies the procedures to be followed by a registration authority for reference data.
- ISO 15926-6 specifies the information required when defining additions to the reference data specified in ISO 15926-4.

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

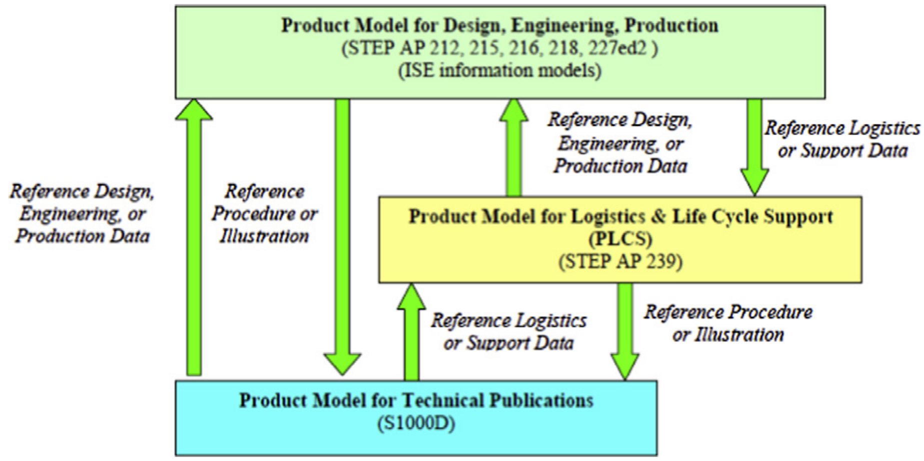


Fig. 15. Interoperability model for ship damage control and maintenance [24].

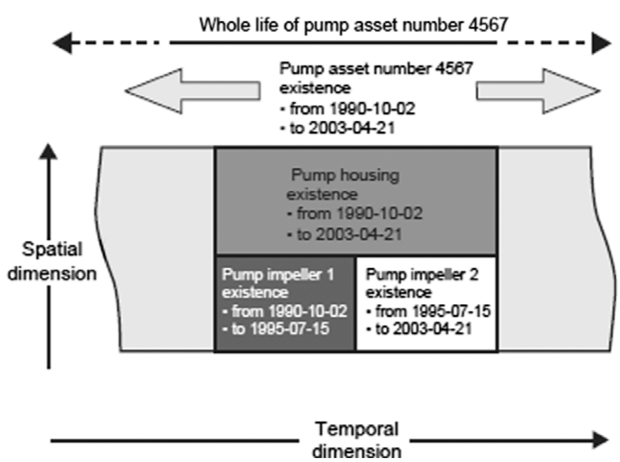


Fig. 16. Spatial and temporal decomposition of a pump [25].

- ISO 15926-7 (old) provides implementation methods for the integration of distributed systems (currently not available at the ISO website).
- ISO 15926-7 (old) has been revised and will be spitted into 4 parts:
- ISO 15926-7 Template Methodology.
- ISO 15926-8 OWL.
- ISO 15926-9 Façade Implementation.
- ISO 15926-10 Abstract Test Methods.

ISO 15926 implements the so called 4D approach which recognises that physical objects can have a temporal as well as a spatial decomposition as illustrated in the example of a pump in Fig. 16.

5.2.7. MIMOSA OSA-EAI

MIMOSA OSA-EAI (Open System Architecture for Enterprise Application Integration) aims to enable organisations to optimise the utilisation of their assets by filling the gaps between the different islands of information. MIMOSA specifications allow organisations to cost-effectively develop and maintain products. Among other functionalities, MIMOSA proposes asset lifecycle management solutions, which will be assessed in this chapter regarding the requirements for PLM.

MIMOSA is part of the OpenO&M initiative which aims to provide a set of information standards for the exchange of Operations and Maintenance (O&M) data (Fig. 17).

The core technology of MIMOSA OSA-EAI is the Common Conceptual Model (CCOM). It is then converted into a Relational Implementation Model called Common Relational Information Schema (CRIS). Interoperability is specified for work management systems, diagnostic/health assessment systems, process data historian systems, dynamic vibration systems, sample test data historian systems, binary/thermography data historian systems and reliability database systems. The technology is designed to transfer archived data and no middleware is required.

5.3. Globally unique identifier

The assignment of Unique Integration Codes (UICs) is required to guarantee "plug-and-play" interoperable OSA-EAI systems. It is defined as a 4-byte, non-negative integer. The four registration Authorities are, at the OpenO&M level, MIMOSA which provides an enterprise UIC, at the Enterprise level the enterprise administrator, which provides site UICs, then at the Site level the site administrator, which provides segment UICs, agent UICs, database UICs and measurement location UICs. At the data base level, the database Administrator provides manufacturer, model, reference type, ordered list and work management UICs. This system provides universal asset type taxonomy.

5.4. Routing and lookup mechanism

MIMOSA provides a series of information exchange standards using XML and SQL. (Tech–XML client–server application specification V3-1) Several technology types are supported and XML server interfaces have been designed. Information exchange interfaces for product lifecycle management are included in the following technology type: Registry Management Information (REG-XML Server interfaces), Reliability Management information (REL-XML Server interfaces), Trendable Scalar Data (Trend-XML interfaces). Despite the existence of these interfaces, there is no proper lookup mechanism, i.e. the address of client and resource must already be known.

5.5. Information resources

Access to information on server are from three types: Read-Only (Query interface), Write-Only and Read/Write. MIMOSA provides the tools for writing, updating, or deleting information for example, but there is no mechanism to link information with product instances. Because of the wide spectrum of application of MIMOSA OSA-EAI, many supported features are not needed. The method used to stored data limits the amount of asset monitoring [26].

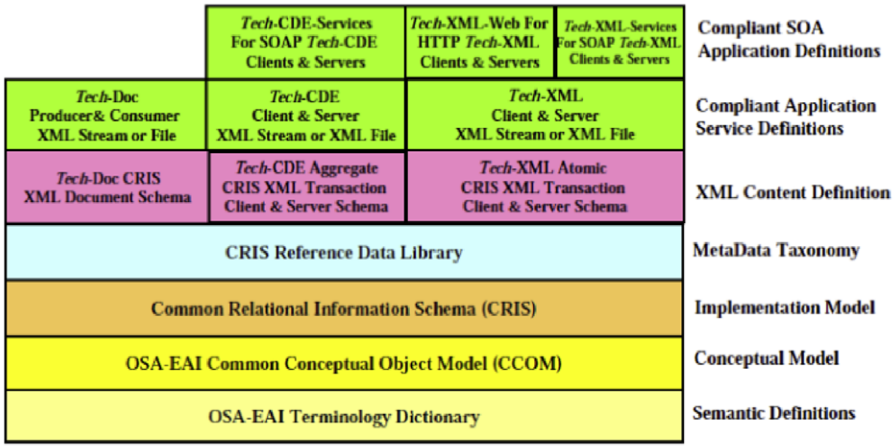


Fig. 17. MIMOSA Open system Architecture for Enterprise Application Integration (OSA-EAI) (www.mimosa.org).

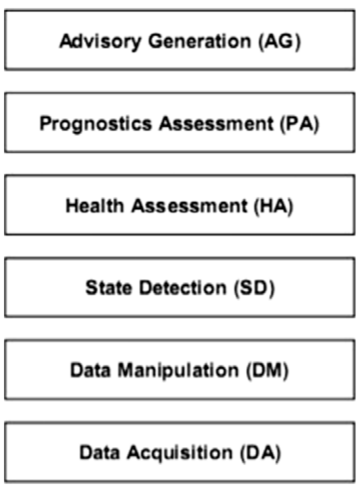


Fig. 18. OSA-CBM functional blocks (www.mimosa.org).

5.5.1. MIMOSA OSA-CBM

MIMOSA OSA-CBM (Condition Based Monitoring) aims to allow the integration of a wide variety of software and hardware in a condition based maintenance system. OSA-CBM specifies a standard architecture and framework for implementing condition based maintenance systems. It is an implementation of the ISO-13374, which defines the six blocks of functionality; see Fig. 18, and the inputs and outputs of those blocks. Data structures and interface methods is the value added by MIMOSA.

Among others, an advantage of OSA-CBM is the competition between solutions providers and the resulting freedom to choose the best solution at each functional level.

The architecture is divided into the information specifications and the interface specification to separate information that is moved, stored and processed from the mechanism that accomplishes these tasks.

Subscription request are supported, as well as synchronous, asynchronous and service type of communication. To implement the architecture, several technologies can be used, e.g. Web services. Note that two systems implemented with different technologies are not compatible, even though the information they are communicating is the same.

OSA-CBM provides only interfaces and UML class diagrams. Algorithms remain hidden and vendors can develop and sell algorithms. The consequence is that OSA-CBM is not very useful regarding the requirements from a lifecycle management point of view.

5.6. PROMISE architecture propositions

The PROMISE architecture has been already presented before and has been developed to cover the requirements for Closed-Loop PLM. The different elements of the architecture are not standardised yet. The main information exchange mechanism is the Web Service based PROMISE Messaging Interface (PMI), Fig. 8.

The PROMISE architecture matches all the requirements of PLM. It is capable of using any identifiers such as EPC or WWAI identifier. Callback address in subscription requests is the routing mechanism provided. Information resources can be any node that implements the PMI. A PDKM Semantic Object Model, Fig. 9, stores product related information and a Decision Support System can process this data. Timely information is provided by the PMI and its subscription functionality. The subscription interval provides a kind of synchronisation. The flexibility of the PROMISE architecture (no restrictions on how nodes should be connected, Fig. 6) insures the reconfigurability. Data analysis can be processed within a DSS. PROMISE specifications provide a conceptual model for the hardware layer, the PEID, but do not provide any specific standard in that area.

Although PROMISE propositions cover all the requirements of the closed-loop PLM, they are not standards and in order to avoid duplications of standards, this work aims to take advantage of the existing standards. Hence, the assessment of standards is relative to the PROMISE architecture.

Two candidates for alignment with standards have been defined within the PROMISE project, that is PMI (PROMISE Messaging Interface) and the PDKM SOM (Product Data and Knowledge Management System Object Model, Fig. 9). Parlikad et al. [17], identified two main gaps to this respect: (i) Product lifecycle event notification and (ii) management of item-level field data as shown in Table 2. The authors underline in this report that Tables 2 and 3 are derived from the observation that the use of many low-level standards such as Unicode or XML is inevitable. Table 3 is designed to focus attention on higher layer standards, especially where these imply the lower layer ones. For example, the Web Services standard builds on standards such as XML, TCP/IP, and HTTP.

5.6.1. PROMISE architectural elements — recommended standards gaps to be filled

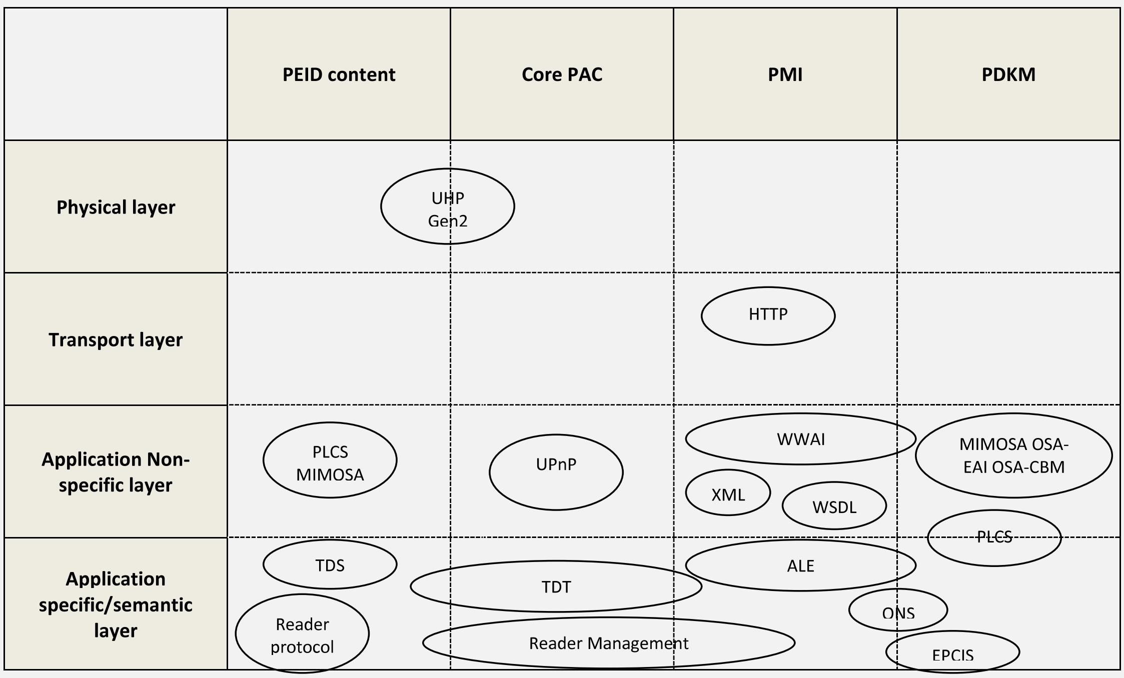
A mapping of the existing standards facing PROMISE architecture is given in Table 3 [23].

The columns of the table correspond to different components of the PROMISE architecture while the rows correspond to different protocol layers. Each layer is described in detail below:

Table 2
Summary of the gaps based on the recommendations of Parlikad et al. [17].

PROMISE architectural element	Recommended standards	Gaps to be filled
PEID	OGC SWE, EPCglobal standards (RFID)	None
Core PAC	UPnP	None
PMI	HTTP, SOAP, WWAI, XML, HTML, ID@URI, EPCglobal standards	Product Lifecycle event notification (only partly supported by EPCglobal standards: ALE and EPCIS)
Middleware	J2EE OSGi, Web Services	None
PDKM	PLCS	Semantic management of item-level field data corresponding to the product

Table 3
Role of the standards in PROMISE [adapted from [23]].



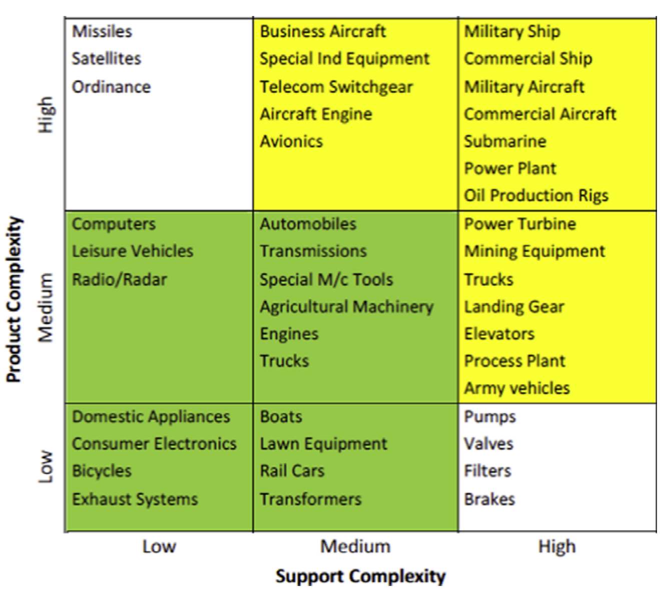


Fig. 19. Applicable sectors for PLCS (yellow) and PROMISE (green) [17]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Physical: The physical layer roughly corresponds to the physical layer of the OSI reference model. This layer corresponds to such things as the radio communication mechanism used to communicate with the PEID.

Transport: The transport layer roughly corresponds to the transport layer of the OSI reference model. Standards or protocols such as HTTP that ensure correct and robust communication over a physical medium fall into this layer.

Application non-specific: The "application non-specific" layer refers to high-level protocols that are not specific to a particular application. For example, Web Services is a generic protocol that can be used in a wide variety of applications.

Application specific: The "application specific" or "semantic" layer refers to specific uses of high-level protocols that assign particular semantics or meaning to the communication. For example, EPCIS is an example of an application specific protocol as it builds on the Web Services protocol and defines a specific set of method calls.

There is a gap in the application specific domain within the PMI. In particular, complex product lifecycle event notification is not supported. EPCIS supports event notification to clients with standing queries but extension of the vocabularies is required.

There is no existing standard that is suitable to reach all the objectives of the Closed-Loop PLM. Some of the standards can partly fill the requirements and some others are too complicated for the products in the scope of this work. In the case of PLCS, for example, the applicable sectors of PLCS are not the same as those from PROMISE (Fig. 19). PLCS has been designed with major engineering assets producers whereas PROMISE focuses on less

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

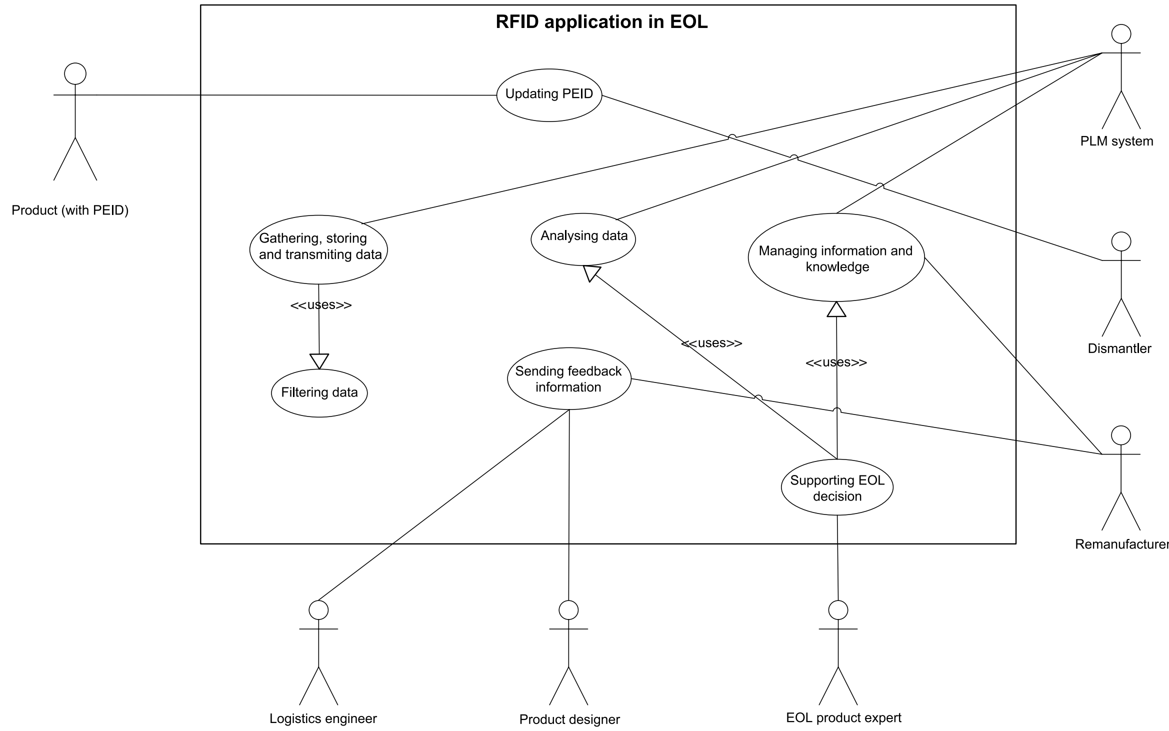


Fig. 20. Use case for an EOL application.

complex products. It is reflected by the simplicity of the PDKM SOM.

MIMOSA OSA-EAI and MIMOSA OSA CBM have been developed in order to link the islands of information within a company but they do not provide a means to focus on lifecycle monitoring.

The main gap identified during the PROMISE project can be filled by the OGC family of standards. There is no gap within the sensor data standards but when we consider single asset lifecycle monitoring, integration of sensor data with RFID could be needed. For the latter, there is no standard.

Recently, a very new initiative has been launched by The Open Group in order to establish a consortium to work along the direction of the PMI propositions under the term of "Quantum Lifecycle Management" [27].

Based on the results of the assessment of existing standards, it was recommended using existing standards or extensions to these existing standards to cover the requirements of PLM. In particular, we will focus on extension to EPCglobal standards and see how the PROMISE propositions can be used to extend EPCglobal's scope. For a more detailed assessment and discussion on the relevant existing standards please see the works of Parlikad et al. [17,23].

Former research works about extension to EPCglobal are presented in the next chapter and a preliminary case study is carried out to illustrate the standards analysis.

6. Case study

In this chapter, the PROMISE A1 demonstrator is analysed to better understand how the existing standards can support the EOL management of a passenger vehicle [28–30]. The description of the case study is followed by the analysis of the impact of the standards and by propositions.

6.1. Description of the case study

Two main objectives are defined for the EOL: firstly, the identification of components which are worth re-using and, secondly, the transfer on the component of relevant information about its post-deregistering life. To address the first of these two objectives, several tasks have to be done. These tasks are: tracking relevant component info from BOM/Production phase, tracking relevant component info about its usage/mission, updating/resetting info about component mission in case of component substitution, identify economical information about component re-use and decide whether to re-use a component or not, using all the information gathered during the previous tasks. The overall closed-loop PLM methodology is presented in a recent publication by the author in the new Springer Automation Handbook [31]. Fig. 20 shows the use case model, Fig. 21 the overall workflow and Fig. 22 the general physical description of the case study.

6.1.1. Use case model

The use case model for an EOL application is illustrated in Fig. 20.

6.1.2. Overall workflow

Two different technologies have been assessed with success by this demonstrator. The first realisation combines RFID tags and normal production sensors of measurement (e.g. temperature) whereas the second uses Zigbee tags and wireless sensors. Because of the specific research question we address in this work, the focus is on RFID data. Data are gathered in an ECU, RFID Tag data through a tag reader and sensor data by a wired or wireless connection. At this stage, data are stored and processed in the on-board diary (a component inside the ECU). The previous components are inside the vehicle. The communication with backend systems (DSS, PDKM, OEM backend IMDS and other database) is operated through middleware. There are several pieces of information to be transmitted, that is, synthesis, statistics of lifecycle and vehicles and components mission profile description (Tables 4 and 5).

Integration of the following components (in bold the components specifically developed in PROMISE) has been performed:

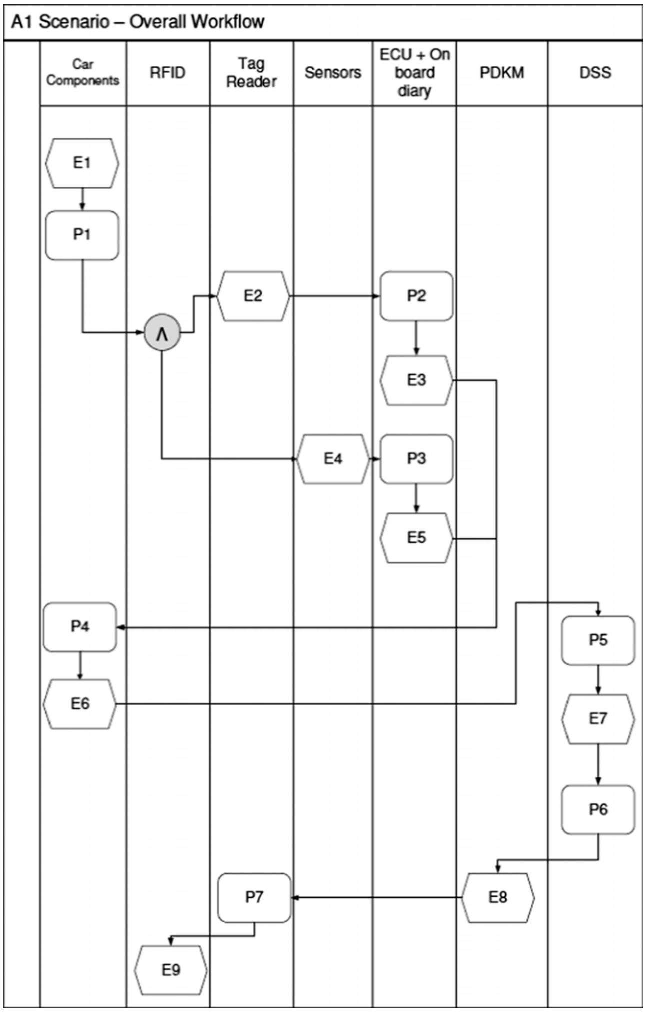


Fig. 21. A1 scenario — overall workflow [29,30].

1. Sensors and Embedded devices on-board of a vehicle: the hardware is off-the-shelf and the **software** has been developed to create the network and interface with a telematics platform (see component 4 below);
2. Data buses on-board of vehicles: collect on-board data on the real use of the vehicle, which enable computation of the residual life of the components;
3. Demonstrator-specific **algorithms for residual life**;
4. Telematics platform: acts as a repository for lifecycle data; aggregates data; calculates residual life of components using the algorithms mentioned in point 3 above; updates the list of on-board components;
5. Demonstrator-specific **database**: collects data coming from the vehicle at End-of-Life;
6. Demonstrator-specific **PMI**: transfers data towards PDKM;
7. **PDKM**: acts as a repository of lifecycle data, enables computation in DSS;
8. **Algorithms for decision making at End-of-Life**: calculates the optimal decision for recycling/remanufacturing/reusing;
9. **DSS**: integrates the algorithms mentioned in point 8 above;
10. DSS and PDKM **GUI**: creates the interface with the user.

The architecture of this demonstrator is PROMISE compliant.

6.2. How can standards support the realisation of the objectives of the demonstrator?

The following functionalities are required [29,30]:

Table 4
A1 overall workflow [29,30].

Modeling components		Description
Process	P1	Customer uses the car
	P2	ECU memorises list of components
	P3	ECU stores data
	P4	The CAR reaches EOL
	P5	DSS analyses data
	P6	DSS updates list of components on PDKM
	P7	Tag Reader writes residual life and other information on RFID
Event	E1	Car assembly
	E2	List of components defined
	E3	List of components is written on On Board Diary
	E4	Data are acquired
	E5	Map of data updated
	E6	Car EOL
	E7	List of dismantling actions to be performed
	E8	List of information updated on PDKM
	E9	Residual life and other info written on RFID

1. Update component mission profile.
2. Reset Component mission profile, update age of component.
3. Initialization of component mission profile, update age of component.
4. Transfer Component Ident. Nr, substitution date from Tag to DSS.
5. Transfer Comp. Mission profile and mileage from on board computer to DSS.
6. Get off-line info from OEM backend systems.
7. Decide what component is worth regenerating.
8. Transfer Info from DSS to Component TAG.
9. Transfer Info from DSS to OEM backend system.

These tasks can be ordered in three types: internal (1, 2, 3), external (6, 7, 9) and crossing tasks (4, 5, 8), depending if they can be achieved inside the car or if they need the backend system.

The support of these functionalities by standards has been analysed.

As show in Table 6, some of the tasks are not supported by standards. Propositions are made in the following paragraphs.

1. Update component mission profile

Proposition: The real-time measurement and the storage of the data can be supported by the OGC SWE architecture. The use of this family of standards provides us with several tools. SensorML can be used to describe sensor data model and metadata. A filtering mechanism is provided by SOS and SPS. The principle of filtering is the definition of property-types to describe the phenomena that interest us. These property-types can be CompositePropertyType or CompoundPropertyType to describe more complex phenomena [22]. The phenomena to be measured (i.e. number of clutch usage, number of vehicle starts and working time against a certain temperature and number of engine start-ups depending on water temperature) can be supported by the property-types. The On-board diary can access the data by querying the discovery services (CS-W). The Sensor Alert Service (SAS) allows a "push" principle of delivering measurements.

2. Reset Component mission profile, update age of component and

3. Initialization of component mission profile, update age of component

Proposition: EPCglobal network supports automatic identification of component replacement. An AggregationEvent that includes the ID of the new component will be created and the business context can be added at the EPCIS level. The reset of the component mission profile is done during a time where the vehicle can be connected to the internet and for this reason, a direct access to the On-board diary is possible. No standard supports the format of the "component mission profile".

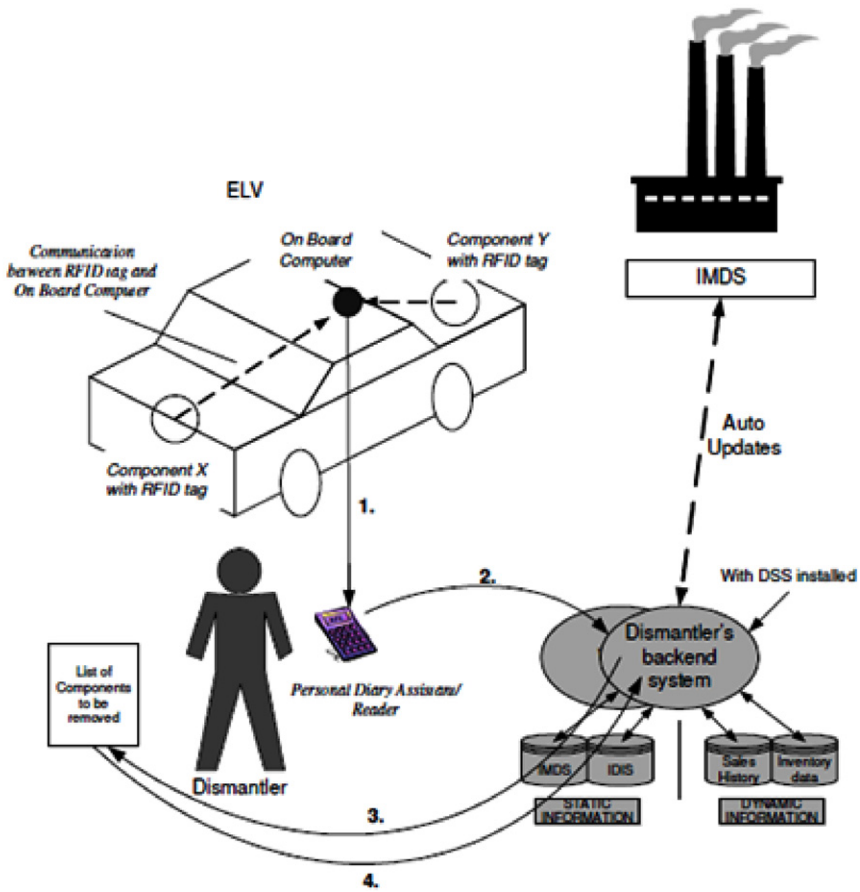


Fig. 22. General description of A1 strategy [28].

Table 5
Physical Components (PC) and Software/Support Systems (SS) of the demonstrator [adapted from [29,30]].

Name	Physical Component Software Support System	Functionality	Interfaces
PC1	Tag on identified component X	Uniquely identify component X	Wireless communication with SS1 included in PC4, using PC5
PC2	Sensor measurements	Continuously measure some vehicle parameters (e.g.: outside temperature)	Links with PC1
PC3	Component X	Depending on X	Supports PC1
PC4	Electronic Control Unit (ECU)	Manage in real-time vehicle electronics	Wireless link with PC1 through PC5, wired link with PC2, hosts SS1, wired link with SS2
PC5	Tag reader	Get info from PC1	Link between PC1 and SS1
SS1	On board diary	Store X statistics. Perform basic calculation on sensor data	Communication with PC5, PC2 and SS2
SS2	Backend decision support system	Download info from SS1, get component age from PC4	Retrieve data from SS1, SS3, SS4
SS3	Backend international material data system	Manage info about X from its ID	
SS4	Other database	Contain the components selling prices, availability of products, destination	

The use of the PDKM SOM instead of the mission profile is possible.

4. Transfer Component Ident. Nr, substitution date from Tag to DSS and

5. Transfer Comp. Mission profile and mileage from on board computer to DSS

Proposition: The transfer from the tag component can be supported by the EPCglobal framework and the transfer from the On-board diary can be done through Webservices or by a wired connection. The centralisation of the data can be supported by the EPCglobal standards, using discovery services and standing requests.

6. Get off-line info from OEM backend systems and

9. Transfer Info from DSS to OEM backend system

Proposition: This can be done by a wired connection or through Web-services.

6.3. Propositions of how to use the standards in the case study

The real-time measurement and the storage of the data can be supported by the OGC SWE architecture. In parallel with the OGC architecture, EPCglobal standards can be used to manage the RFID tags and Tag data. To fill the lack of centralised data storage and analysis, the two elements of the PROMISE Project PDKM and DSS are added "at the top" of the architecture. They communicate through the Web and discovery services and may subscribe to standing queries registered with multiple information services (e.g. EPCIS repositories).

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

Table 6
Analysis of the supported tasks of the case study.

Task	Functionality	Description	Standard candidates	Comments
1	Update component mission profile	Real-time measurement by the sensor and storage of data on the On board diary using "component specific mission profile synthesis"	ODC standards family (raw data to XML), MIMOSA, OSA-CBM, PLCS (information model) PROMISE PDKM SOM	No standard for the integration of sensor data to RFID Data. No information about the synthesis (proprietary system)
2	Reset component mission profile, update age of component	Automatic identification of component replacement and reset of the component mission profile	EPCglobal (UHF Class1Gen2, Tag Data standard, EPCIS) ZigBee (IEEE 802.15.4)	EPCglobal only for passive tags (Active tags not supported) and ZigBee for ZigBee network
3	Initialization of component mission profile, update age of component	Car initialization. Automatic reset to 0 of all synthesis related to car components	EPCglobal, ZigBee	EPCglobal only for passive tags (Active tags not supported) and ZigBee for ZigBee network
4	Transfer Component Ident. Nr, substitution date from Tag	At the deregistration of the vehicle, transfer from Tag or On board computer (depending on the implementation of previous functionalities) to DSS system	EPCglobal, PROMISE PMI	No lack of standards
5	Transfer Comp. Mission profile and mileage from on board computer to DSS	At the deregistration of the vehicle, transfer of component mission profile from On board diary to DSS system	PMI, PDKM SOM, MIMOSA OSA EAI	Since we do not know the format of the synthesis, there are several standards which might suit
6	Get off-line info from OEM backend systems	Component identification number, transfer of cost model and other information about the component and vehicle from OEM backend systems to DSS	MIMOSA OSA-EAI, PROMISE PMI	No lack of standards
7	Decide what component is worth regenerating	Decision making inside the DSS	No Standards for decision making	Need for standards (core competency of the company)
8	Transfer info from DSS to Component Tag	Writing on the tags of the decision taken before (wear-out level)	EPCglobal, ZigBee	Active RFID not supported
9	Transfer info from DSS to OEM backend system	Transfer of the decision taken to OEM backend system	PROMISE PMI, MIMOSA OSA-EAI	No lack of standards

The remaining issues concern the integration of sensor measurement with the RFID data. We have now two architectures (OGC SWE and EPCglobal) with similarities and differences. Both architectures have a repository, discovery services, filtering mechanisms and a single interface for querying data. The main differences are the more restricted services in EPCglobal. It does not provide alerting service and planning element due to the simpler requirements of data capturing. Furthermore, OGC SWE does not support global IDs but locally unique IDs for sensor. The management of the two separate discovery services and repositories can be done in several ways, i.e. to employ two separate services, to extend one of the discovery services or to develop an interface to manage multiple repositories and discovery services. The discovery services should be able to provide links to various kinds of information resources for a particular ID or EPC, e.g. to EPCIS interfaces, web pages, web services. Additionally organisations might operate local sensor Discovery Services about available sensor metadata (e.g. locations capabilities) (Fig. 23).

The compliance of the data format between the architecture has to be analysed. Each framework defines its own XML schema and compatibility is not guaranteed. In particular, a mapping engine is probably necessary to make the data flow (from EPCglobal or OGC family) PDKM compliant. There are standardised tools such as XSLT that can be used for transforming XML data into other formats (including other XML schema) but it is not within the scope of this thesis to analyse that tools.

7. Ontology based approaches for semantic PLM standards

As we said before, future PDT systems need to allow exchange of both Static and Dynamic Product Data and seamless interoperability of related systems. Ontology based approaches for semantic standards is a promising technology towards that objective. In this section we present an overview of the first efforts in this direction.

Ontologies are commonly used in artificial intelligence and knowledge representation. Computer programs can use an ontology for a variety of purposes including inductive reasoning, classification, a variety of problem solving techniques, as well as facilitating communication and sharing of information between different systems. Different organisations are attempting to define standards for specific domains. The 'Process Specification Language (PSL)' created by the National Institute for Standards and Technology (NIST) is one example for the ontology of process. There have been a number of modeling languages for ontologies: ontolingua KIF, OKBC, CycL, XOL, RDF, OIL, and DAML + OIL, and so on. Among them, XOL, RDF, OIL, and DAML + OIL are XML based languages that are recently being highlighted with the semantic Web. There have been several modeling tools in various domains and many research works to deal with information or ontology modeling for several objects. For example, Fangyi et al. [32] proposed a conceptual information model to define and describe the characteristics of product lifecycle environment. Furthermore, Fenves et al. [33] proposed a conceptual framework for product information interoperability to access, store, serve, and reuse whole product lifecycle information. Sim and Duffy [34] proposed an ontology for product design activities. They identified and classified a generic set of design activities. The ontological completeness, clarity and coherence of activities are evaluated through a protocol analysis and the design process in the domain of electronic design. Li et al. [35] presented an ontology-oriented approach that used description logics formally to represent concepts and roles (relations) of partner's view of domain knowledge in a virtual enterprise.

In the field of PLM there are several recent works dealing with ontologies. Fiorentini et al. [36] translated the NIST's core product model and proposed an ontology for the Open Assembly Model (OAM) implementing several OWL capabilities. Also Fiorentini et al. [37] based on the work developed for the OAM demonstrated how to implement ontologies into existing product models. Lee and Suh [38] developed a model for sharing product knowledge of the Beginning Of Life (BOL) on a web. Brandt et al. [39] apply ontologies on to knowledge management in design processes with the aim of making knowledge of the design processes understandable and accessible to all engineers. Zhang and Yin [40] make an attempt of applying ontologies in a multi-agent distributed design environment. Suh et al. [41] use ontologies for interoperability and present a model for using data of the entire life of the products as an input for the design and production of new products. Aziz et al. [42] have developed an ontological management methodology to overcome limitations of current PLM implementations.

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

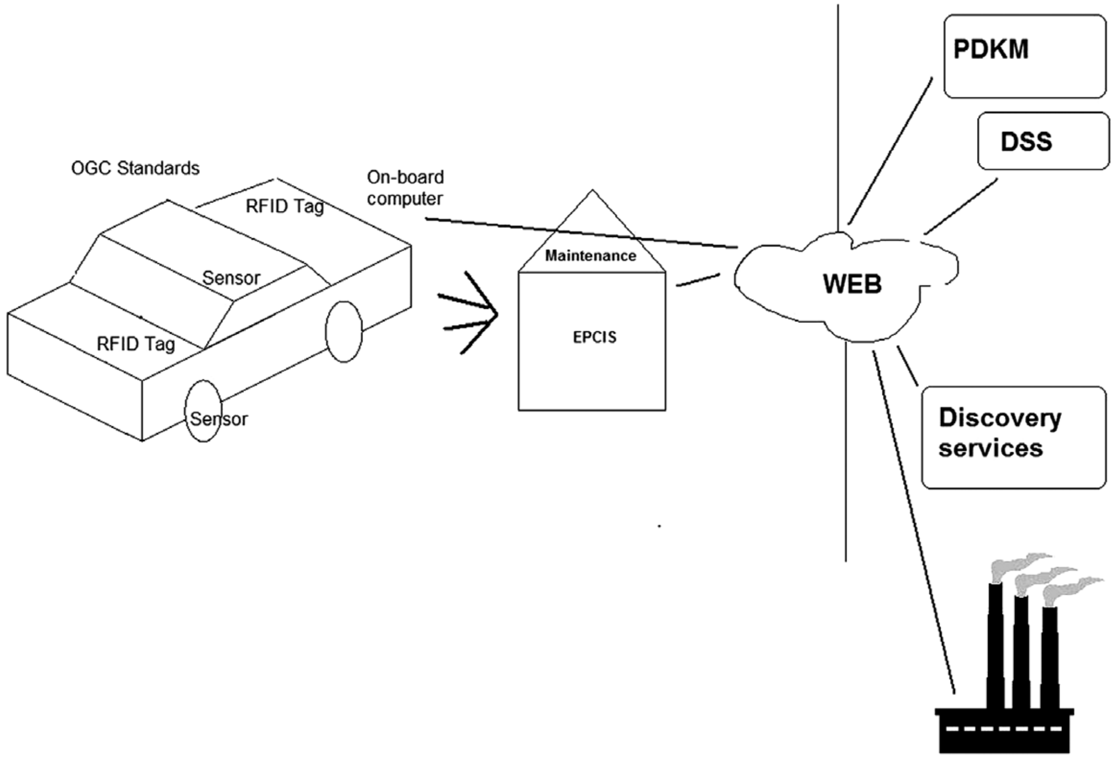


Fig. 23. Proposed standardised support for the demonstrator [28].

For PLCS, Price and Bodington [43] describe the implementation of OWL into OASIS Product LifeCycle Support (PLCS). First step is the use of Semantic Web technology for developing Reference Data which includes the reuse of Reference Data of ISO 15926.

For ISO 15926, an initiative to translate this standard to the semantic web technology OWL has been taken by the Norwegian Oil Industry [44]. The objective is to study the semantic relationship between ISO 15926 and OWL in order to facilitate in- and cross-organisational cooperation. The authors developed two transformation methods that resulted in a seemingly direct representation of ISO 15926 in OWL. In [45] the OWL approach is extended to "templates" as patterns for complex statements of the industrial domain. They describe how ISO 15926 can support data integration and exchange with automated reasoning applied to information mapping using SWRL rule language. However, it is stated that rule languages, such as SWRL, are not yet as settled as OWL itself and it is anticipated that developments around W3C's Rule Interchange Format (RIF) will contribute to a solution. Sandsmark and Mehta [46] describe the framework of the project "Integrated Information Platform for reservoir and subsea production systems" (IIP) that is supported by the Norwegian Research council. In this project the concept is to develop an information platform combined with the use of ontologies to overcome proprietary and system dependent data definition that prohibits effective exchange, sharing and integration of information. As a basis ISO 15926 is used, since its generic concept model makes it ideal as an integration platform for other standards. Gulla et al. [47] describe the work done within the IPP towards transforming and extending existing standards into OWL ontology for reservoir and subsea production systems. This ontology used for analysing data and interpreting user needs, may allow data to be related across phases and disciplines, helping people collaborate and reducing costs and risks. Tomassen et al. [48] based on the work done in IIP project propose a method to improve information retrieval quality by using ontologies. The ontology used is the one developed in IIP, which is based on ISO 13628 and it will be modeled in ISO 15926. Strasunskas [49] presents research in IIP on development of rule based notification in subsea production systems to monitor and analyse production data. The author concludes that the full expressive power of OWL (OWL Full) is needed in order to represent ISO 15926-2/4 which is a burden for reasoning (incomplete) and inference (undecidability). Moreover, a certain future work will be the alignment of the method developed in IIP with MIMOSA's open systems architecture for condition based maintenance.

In [50], a primitive ontology model is proposed for product lifecycle metadata. Because there are hundreds of different kinds of product lifecycle metadata, it is impossible to contain all things into a schema. Accordingly, it is necessary to characterize product lifecycle metadata into few generic types. In this study, we establish a schema for product lifecycle metadata based on T-PPOA (Time-Product, Process, Organisation, Agent) concept. The T-PPOA consists of the relations among product, process, organisation, and agent, which is based on a time event. The innovative particularity of this model is that it considers "time" as the focal point of PLM events and activities. Fig. 24 shows the RDF schema for product lifecycle metadata. A schema defines taxonomies of resources and properties for describing product lifecycle metadata. Schema can be referenced using the namespace mechanism of XML within the RDF syntax. In this study, the RDF schema for product lifecycle metadata uses five namespaces: product, process, organisation, agent, and time. This RDF schema contains basic relations among T-PPOA. During product lifecycle, all information can be modeled with the concept of T-PPOA. Product lifecycle operation is basically done with relations among products, processes, and organisations. Optionally, agents can be involved in this relation. Any relations among them should be done based on time events.

Matsokis and Kiritsis [51] used this time based schema and transformed the PROMISE Semantic Object Model presented in Fig. 9 in an ontology model using Protégé shown in Fig. 25.

The necessary transformations and possible uses of this model are presented in the above reference and can be summarized as follows:

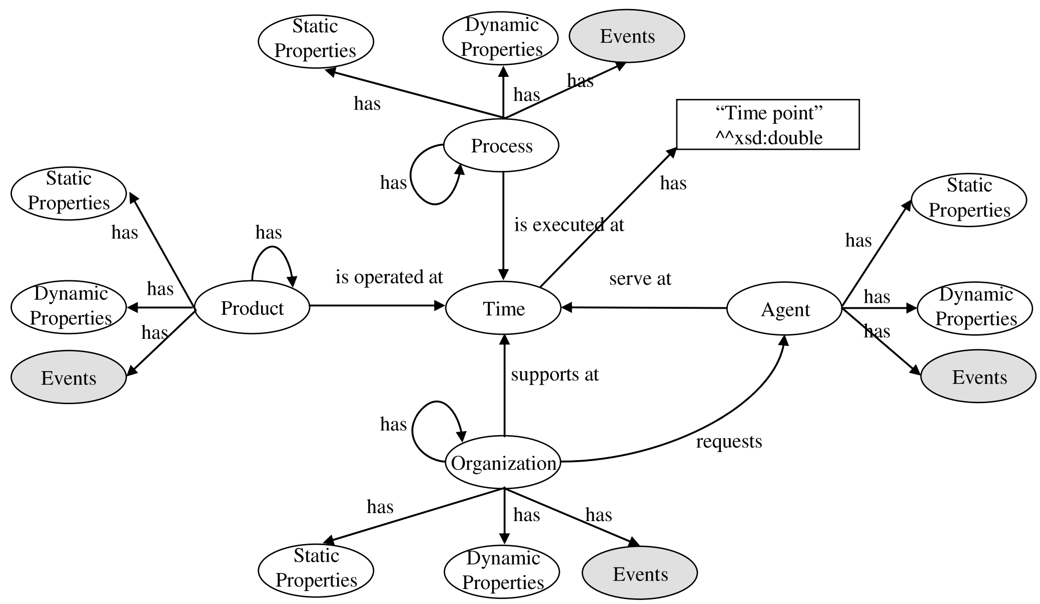


Fig. 24. RDFS for product lifecycle metadata.

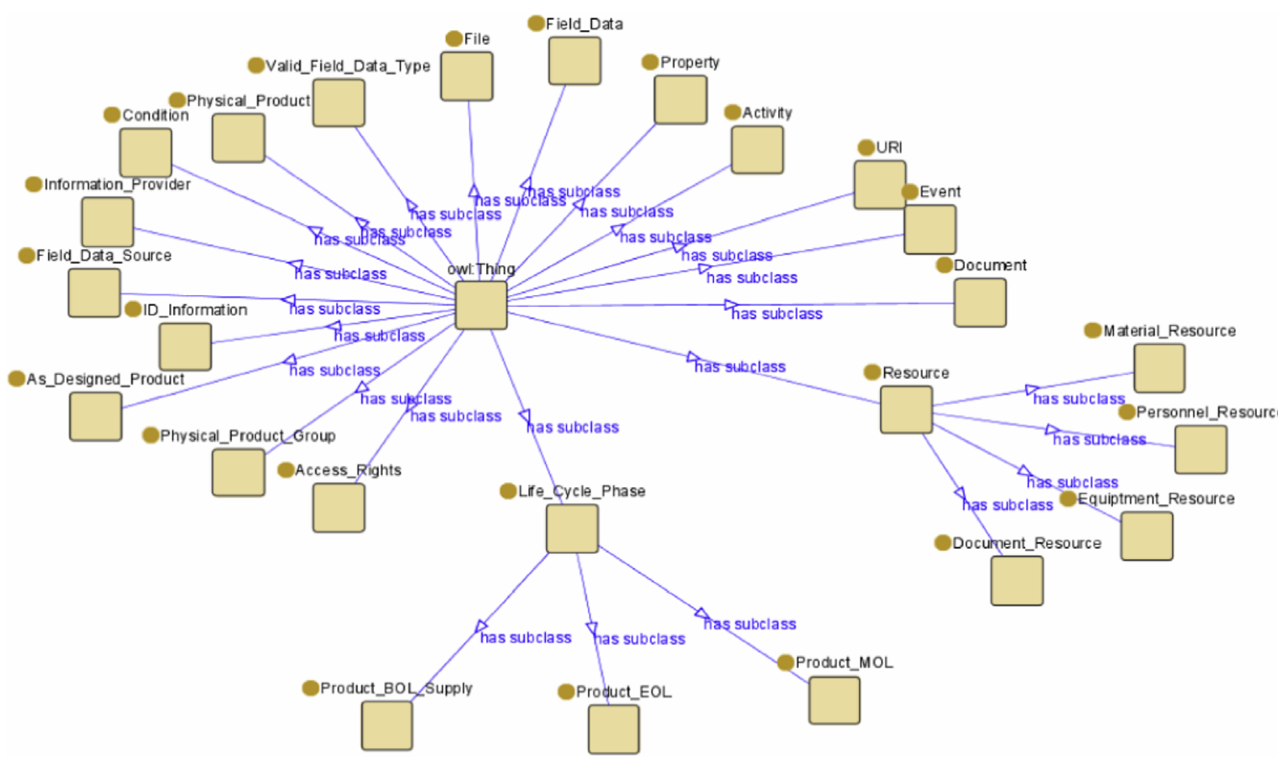


Fig. 25. Structure of the ontology based class-hierarchy of the PROMISE PDKM SOM.

The ontology model aids into achieving various levels of interoperability, required to enable the full potential of PLM and more particularly of Closed-Loop PLM. By developing DL rules, we further extended this model to incorporate reasoning capabilities.

The ontology model is executable and able to facilitate multiple data from multiple physical products.

The ontology model was applied in a PLM case study where:

- The ontology model was populated with instances.
- The model was extended according to requirements.
- Easy to implement guidelines for sorting data to the extended model were provided.

The case study also demonstrated how and which are the advantages and benefits achieved with systematic use of OWL-DL and the new wider horizon opening for Ontology based PLM. The advantages for the PLM model are:

- The model handles multiple data from multiple physical products by applying DL rules.
- Concept equivalencies–inconsistencies are efficiently handled by applying DL rules, supporting system interoperability and data integration.

More recently, in [52] the concept of "*Duration of Time*" is introduced for improving today's PLM systems in the domains of

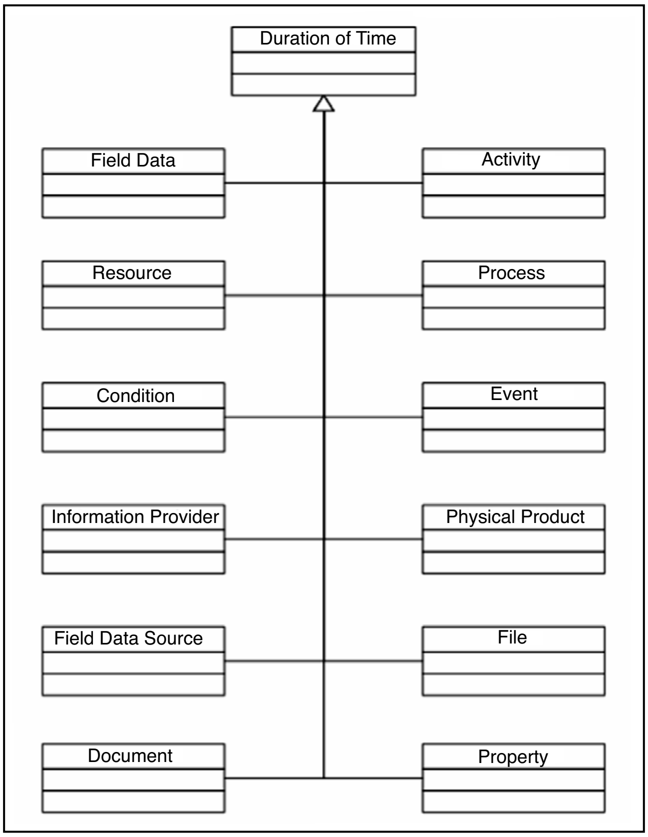


Fig. 26. Schematic Duration of Time representation example.

data visibility, data integration and system interoperability. The main element of the concept, used for improving the systems performance is time. The concept is that time should not be one part of the model, but it should be the basis of the model and all other elements should be parts of it. All aspects and elements of a model are parts of time.

The concept of "*Duration of Time*" has unique advantages over existing concepts. Systems built on this concept take advantage of the time characteristics and combined with semantics provide data visibility, data integration and system interoperability. Time is used as a basis to provide a first step system to system visibility and common understanding. Two different time based systems will certainly have in common their time attributes and therefore, they can be synchronised even though they might have been extended and used differently.

This is an easy to apply method on existing models by making a "Duration of Time" class as a super-class of all classes of the model. This class provides the unified time framework for the entire system. A schema of this model is shown in Fig. 26. The concept is protected by a patent provisional application.

8. Discussion and conclusions

Closed-Loop PLM or Closed-Loop Lifecycle Management in general is becoming a fact in various industrial sectors and business in general. The trend is for more integrated software systems including smart embedded systems that provide real-time information to higher level information management systems. Seamless and transparent interoperability is a top priority requirement for the developers of such integrated information systems. This obviously relies on appropriate standards.

Actual PDT standards focus on static product definition data models and do not cover the needs for lifecycle information and knowledge management by such integrated systems and many efforts are undertaken by industry and standardisation organisations for the identifications of gaps by the existing standards and the appropriate extension of existing or development of new standards where required.

Moreover, future generations of intelligent products having extended data sensing, processing, communication and self decision making capabilities impose new advanced requirements for interoperability and related standards covering not only product data definitions but also dynamic data that have to do with the real life of a product as it used and serviced during its whole life.

Actual efforts to cover product lifecycle data are focusing on large scale systems (PLCS) and specific applications (ISO 15926 for the Gas & Oil industry).

New efforts such as EPCglobal cover the needs for Auto-ID requirements of product items with the aim to replace barcodes.

There is no system able yet to combine together sensor and ID data, which is a strong requirement for the lifecycle support of future intelligent products.

PROMISE, a recent FP6 EU research project proposes a new concept and architecture for Closed-Loop Product Lifecycle Management covering the whole product lifecycle. A thorough study of existing standards that are usable in developing Closed-Loop PLM systems has been done in this project and identified a number of gaps in existing standards in covering these needs. The project proposed two standardisation initiatives, one at the product semantic modeling level (SOM — Semantic Object Model) and one at the middleware level (PMI — PROMISE Messaging Interface).

The development and extension of ontology models using semantics, supports both interoperability between software platforms and data integration. New ontology based approaches are promising for a new generation of semantic standards that will explore and facilitate at the same time the deployment of the emerging technologies of the era of the Internet of Things. The actual efforts undertaken at a research level by academia and at an implementation level by standardisation bodies, such as the effort to develop an upper ontology for the ISO 15926 standard, are promising and implementation of real industrial test cases are expected to move bring interoperability among intelligent products, process and systems to a new level of performance in the near future.

Acknowledgements

I would like to express my deep gratitude to all PROMISE partners for their much appreciated collaboration. I would also like to thank my Master student Mr. Baptiste Moix for his very valuable contribution in this work and Dr. Ajith Parlikad's team at IfM in University of Cambridge for their support in Baptiste's work.

References

[1] Theodorou S. A classification approach for RFID-based sensor integration, MPhil thesis, University of Cambridge. 2007.
[2] PROMISE architecture series, volume 1: architecture overview. 2008.
[3] PROMISE architecture series, volume 2: architecture reference: PROMISE Core PAC Interface. 2008.
[4] PROMISE architecture series, volume 3: architecture reference: PROMISE Messaging Interface (PMI) and Data services concepts. 2008.
[5] PROMISE architecture series, volume 4: architecture reference: PROMISE PDKM System Object Model. 2008.
[6] PROMISE architecture series, volume 5: architecture reference: PROMISE Decision Support System (DSS). 2008.
[7] Gerritsen, et al. Editorial. Computer-Aided Design 2008;40:735–7.
[8] CERP-IoT Cluster, http://www.rfid-in-action.eu/cerp-iot. 2009.
[9] Framling K, et al. Intelligent products. Computers in Industry 2009;60:135–6 [special issue].

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002

[10] Kiritsis D, Bufardi A, Xirouchakis P. Research issues on product lifecycle management and information tracking using smart embedded systems. Advanced Engineering Informatics 2003;17(3–4):189–202.
[11] Kiritsis D, Moseng B, Rolstadas A, Rostad CC. PROMISE: Product lifecycle management and information tracking using smart embedded devices. Trondheim: Tapir Academic Press; 2008.
[12] CL2M, http://promise-innovation.com/node/7. 2009.
[13] Parlikad AK. The impact of product information on the effectiveness of product recovery decisions. August 2006.
[14] Mitsugi J, Inaba T, Pátkai B. Architecture development for sensor integration in the EPCglobal network, Joined Auto-ID Lab report. Keio University, Cambridge University, ICU University. 2007.
[15] Manyica J, Durrant-Whyte H. Data fusion and sensor management: a decentralized information-theoric approach. Ellis Horwood series in electrical and electronic engineering, 1994.
[16] Auto-ID Labs. Architecture development for sensor integration in the EPCglobal network. July 2007.
[17] Parlikad A, Brusey J, Cassina J, Potter D. PROMISE DI1.5: recommendations for standards. 2006.
[18] Nagargadde A, Sridhar V, Ramamritham K. Representation and processing of related to real world events. December 2007.
[19] Forss B, Petrow J, Framling K, Dikici C, Anke J. PROMISE DR6.5: interface definition and design of enterprise communication infrastructure. 2006.
[20] Botts M. OpenGIS Sensor Model Language (SensorML) Implementation Specification, Version 1.0.0, Open Geospatial Consortium. 2007.
[21] Havens S. OpenGIS Transducer Markup Language (TML) Implementation Specification, Version 1.0.0, Open Geospatial Consortium. 2007.
[22] Cox S. Observation and measurements – Part 1, Version 1.0, Open Geospatial Consortium. 2007.
[23] Parlikad A, Brusey J, Metin A, Barisic D, Framling K, Anke J. PROMISE DI1.4: assessment of existing standards. 2006.
[24] Briggs TL, Rando TC, Daggett TA. Reuse of ship product model data for life-cycle support, In: 2005 ship production symposium, vol. 22, 4. 2005, p. 203–211.
[25] Leal D. ISO 15926 life cycle data for process plant: an overview. Oil & Gas Science and Technology – Rev. IFP 2005;60(4):629–37.
[26] Mathew A, Zhang L, Zhang S, Ma L. A review of the MIMOSA OSA-EAI database for condition monitoring systems. 2006.
[27] QLM: http://www.opengroup.org/qlm/. 2009.
[28] Francone N, Mascolo J. PROMISE DA1.1: scope, definition and description of the curent state vs. Critical issues. 2005.
[29] Francone N, Mascolo J. PROMISE DA1.2: documentation of applicability of technologies and process solutions. 2006.
[30] Francone N, Mascolo J. PROMISE DA1.3: design of the EOL demonstrator on deregistering of ELVs. 2006.
[31] Kiritsis D. Product lifecycle management and embedded information devices. Nof ShimonY, editor. Springer handbook of automation, 2009 [chapter 43].
[32] Fangyi L, et al. 2002. Green design assessing system model of products. In: Proceedings of the 2002 IEEE international symposium electronics and the environment.
[33] Fenves SJ, Sriram RD, Sudarsan R, Wang F. 2003. A product information modeling framework for product lifecycle management. In: Proceedings of international symposium product lifecycle management.
[34] Sim SK, Duffy AHB. Towards an ontology of generic engineering design activities. Research Engineering Design 2003;14:200–23.
[35] Li L, Wu B, Yang Y. An ontology-oriented approach for virtual enterprises. LNCS, vol. 3007. 2004. p. 834–43.
[36] Fiorentini X, Gambino I, Liang V-C, Foufou S, Rachuri S, Bock C, Mani M. Towards an ontology for open assembly model. In: Proceeding of the international conference on product lifecycle management. 2007. p. 445–56.
[37] Fiorentini X, Rachuri S, Mahesh M, Fenves S, Sriram Ram D. Description logic for product information models In: Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference, DETC2008-49348. 2008.
[38] Lee J-H, Suh H-W. Owl-based product ontology architecture and representation for sharing product knowledge on a web. In: 2007 proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference, 2 PART B. p. 853–61.
[39] Brandt SC, Morbach J, Miatidis M, Theißen M, Jarke M, Marquardt W. An ontology-based approach to knowledge management in design processes. Computers and Chemical Engineering 2007;32(1–2):320–42.
[40] Zhang WY, Yin JW. Exploring Semantic Web technologies for ontology-based modeling in collaborative engineering design. International Journal of Advanced Manufacturing Technology 2007;36(9–10):833–43.
[41] Suh S-H, Shin S-J, Yoon J-S, Um J-M. UbiDM: A new paradigm for product design and manufacturing via ubiquitous computing technology. International Journal of Computer Integrated Manufacturing 2008;21(5):540–9.
[42] Aziz H, Gao J, Maropoulos P, Cheung WM. Open standard, open source and peer-to-peer tools and methods for collaborative product development. Computers in Industry 2005;56(3):260–71.
[43] Price D, Bodington R. Applying semantic web technology to the life cycle support of complex engineering assets. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 3298. 2004. p. 812–22.
[44] Hakkarainen S, Hella L, Strasunskas D, Tuxen S. A semantic transformation approach for ISO 15926. In: Roddick JF, et al., editors. ER workshops 2006. LNCS, vol. 4231. Berlin (Heidelberg): Springer-Verlag; 2006. p. 281–90.
[45] Klüwer JW, Skjæveland MG, Valen-Sendsta M. ISO 15926 templates and the Semantic Web, Position paper for W3C workshop on semantic web in energy industries; Part I: Oil & Gas. December 9–10, 2008. Houston, Texas, http://www.w3.org/2008/07/ogws-cfp.
[46] Sandsmark N, Mehta S. Integrated information platform for reservoir and subsea production systems, In: Proceedings of the 13th product data technology europe symposium. 2004.
[47] Gulla JA, Tomassen SL, Strasunskas D. 2006. Semantic interoperability in the norwegian petroleum industry. In: 5th international conference on information systems. 2006.
[48] Tomassen SL, Gulla JA, Strasunskas D. Document space adapted ontology: Application in query enrichment. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). LNCS, vol. 3999. 2006. p. 46–57.
[49] Strasunskas D. Resource monitoring and rule-based notification. Applications in subsea production systems. In: Khosrow-Pour M, editor. Managing worldwide operations and communications with information technology (Proc. of the 2007 IRMA international conference). Vancouver (Canada): IDEA Group Publishing; 2007. p. 1211–3.
[50] Jun HB, Kiritsis D, Xirouchakis P. A primitive ontology model for product lifecycle meta data in the closed-loop PLM. In: Gonçalves RJ, Müller JP, Mertins K, Zelm M, editors. Enterprise interoperability II: New challenges and approaches. Springer-Verlag London Limited; 2007. p. 729–40.
[51] Matsokis A, Kiritsis D. An ontology-based approach for product lifecycle management. Computers in Industry 2009 [in press] [special issue: Semantic web computing in industry].
[52] Matsokis A, Kiritsis D. An advanced method for time treatment in product lifecycle management models. In: WCEAM 2009. 2009.
[53] Kiritsis D, Nguyen VK, Stark J. How closed-loop PLM can improve knowledge management over the complete product lifecycle? International Journal of Product Lifecycle Management 2008.

Please cite this article in press as: Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design (2010), doi:10.1016/j.cad.2010.03.002