A new closed-loop PLM Standard for mass products

Citation for published version:
Cassina, J, Tomasella, M, Taisch, M & Matta, A 2009, 'A new closed-loop PLM Standard for mass products', *International Journal of Product Development*, vol. 8, no. 2, pp. 141-161. https://doi.org/10.1504/IJPD.2009.024185

Digital Object Identifier (DOI):
10.1504/IJPD.2009.024185

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
International Journal of Product Development

A new closed-loop PLM Standard for mass products

Jacopo Cassina, Maurizio Tomasella,
Marco Taisch* and Andrea Matta

Politecnico di Milano,
Piazza Leonardo da Vinci,
20133 Milano, Italy
Email: jacopo.cassina@polimi.it
Email: maurizio.tomasella@polimi.it
Email: marco.taisch@polimi.it
Email: andrea.matta@polimi.it
*Corresponding author

Abstract: This work presents the product object model developed in the context of the EU-funded FP6-IST project named PROduct lifecycle Management and Information tracking using Smart Embedded systems (PROMISE), aiming at the closure of the whole set of information loops throughout the product lifecycle. The ultimate goals of the project are indeed to integrate product data from the entire lifecycle via different sources, to support comprehensive analysis on this data and to enhance by this the enterprise operational businesses. To achieve these goals, a set of hardware and software tools were developed. This paper presents the conceptual model behind one of the software components of this infrastructure, called the Product Data and Knowledge Management (PDKM) system, which is responsible for the integration and management of both product data and knowledge from all lifecycle phases, on a logically consistent basis. The same model also became the foundation for the proposal of a closed-loop PLM standard.

Keywords: Product Lifecycle Management; PLM; closed-loop PLM; PLM standards; UML class diagram; Product Data and Knowledge Management; PDKM.

Reference to this paper should be made as follows: Cassina, J., Tomasella, M., Taisch, M. and Matta, A. (2009) 'A new closed-loop PLM Standard for mass products', *Int. J. Product Development*, Vol. 8, No. 2, pp.141–161.

Biographical notes: Jacopo Cassina received his Laurea in Mechanical Engineering from the Politecnico of Milano, Italy in 2004, discussing a thesis on holonic traceability systems. He got his PhD at the Department of Management, Economics and Industrial Engineering of Politecnico di Milano. He is author of several technical papers, and is also teaching assistant in university and master courses. He has been working in a national Italian project on the Marketing of Technologies, as well as in the eu-funded FP6 PROMISE and PRIME projects. He is also collaborating with MIP, the business school of Politecnico di Milano, on the development of new teaching methodologies using games.

Maurizio Tomasella received his Laurea in Industrial Engineering from Politecnico of Milano in 2004, discussing a thesis on distributed object-oriented modelling and simulation of manufacturing systems, and then started working as a research collaborator at the Department of Mechanical Engineering of the same athenaeum. At the moment he is a PhD candidate, working on optimal

reconfiguration of production systems, and also as a teaching assistant in the course of Integrated Production Systems. Since 2004, he has been working in the eu-funded FP6 PROMISE project and VRL-KCiP NoE, as well as in a national Italian project on distributed object-oriented simulation.

Marco Taisch is a Professor of Operations and Supply Chain Management at Politecnico of Milano. He currently teaches Management of Logistic and Production Systems and Automated Production Systems at graduate level, as well as Operations Management at the MIP, the Business School of Politecnico di Milano. He has been Director of the Executive MBA and of the full-time MBA at MIP. He is member of the Working Group on Integrated Production Management of the International Federation for Information Processing (IFIP), of the International Federation for Automatic Control (IFAC), senior member of the IIE and member of the IEEE. He is or has been involved in several international research projects namely PLANTFABER, PROMISE, PRIME, SOCRADES, GEM, INCOCO-S and coordinator of the Network of Excellence on Intelligent Manufacturing Systems (IMS NoE) with more that 350 partners worldwide.

Andrea Matta is Researcher at the Department of Mechanical Engineering of Politecnico of Milano. He received his Laurea in Industrial Engineering from Politecnico di Milano in 1997. Since then, he develops his research activity at the Department of Mechanical Engineering of the same university. In the period between October 1999 and March 2000 he has been visitor researcher at the Laboratory Productique et Logistique of the Ecole Centrale Paris, France. He is member of the ISCS, the Italian Society for Computer Simulation and of the AITeM the Italian Society for Manufacturing. He is and has been involved in several international and national projects, such as PROMISE, and he currently teaches the following courses at Politecnico of Milano: Manufacturing, Integrated Production Systems and Virtual Manufacturing.

1 Introduction

Within the globally scaled scenario, the product and its related management has unavoidably become a key-aspect, thus creating a product centric or product-driven problem. The customers are becoming more and more demanding, asking for better products, ‘extended’ with related services. Even if the services may not be part of the core business of manufacturers, and can be provided by different enterprises, the product’s value perceived by the customer is a sum of both aspects. Among the main existing approaches, *Product Lifecycle Management (PLM)* and *Product Extensions* are considered to be the most promising. The former has in particular emerged as an enterprise solution and implies that all software systems/methods/tools, such as Computer-Aided Design (CAD), Product Data Management (PDM), Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), etc., used by the various departments throughout the product lifecycle, have to be integrated, in a way that the information managed by these systems can be promptly and correctly tracked and shared among different people and application packages. Nevertheless, PLM is not primarily an Information Technology (IT) problem but, at first, it represents a strategic business orientation of the whole enterprise, being one of the main challenges of this decade (Garetti and Terzi, 2004).

At the same time, the explosion of information technologies has created a new kind of concept, defined as Extended Product, where the product is more than a simple artefact, but it is a complex result of tangible and intangible components. The extension is usually related to the functionality or a new business process around the product. According to Jansson and Thoben (2002) and Hirsch et al. (2001), a tangible extended product can be intelligent, highly customised and user-friendly; whereas an intangible product is mostly the business process itself.

The approach of the PROduct lifecycle Management and Information tracking using Smart Embedded systems (PROMISE) project (http://www.promise-plm.com, http://www.promise.no) to PLM aims at developing a new-generation product information tracking and flow management system. This system will allow all actors playing a role in 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 life, at any time and any place in the world (Jun et al., 2007a).

This paper describes the conceptual model behind one of the main components of this new type of PLM system, the so-called PROMISE Product Data and Knowledge Management (PDKM) system, which is devoted to the integration and management of product lifecycle data from different sources and to the creation, update and management of product knowledge. This in the viewpoint of improving future generations of products, starting from the data on the current products, collected directly from the field.

The model is explicitly object-oriented, and was implemented using Unified Modelling Language (UML), the de facto standard language for graphical modelling purposes.

An existing approach (Terzi et al., 2004; Cassina et al., 2005) was used as a major reference for the modelling work, which is based on the *holon* concept (Morel et al., 2003).

A review of the literature on standards for product design, manufacturing control and product support proved that only very few existing standards are capable of supporting the whole set of product lifecycle phases, and moreover, that where this is possible, the use of such standards is fairly complex. The conceptual model outlined in this paper was developed to overcome this complexity drawback. Nevertheless, existing standards proved to be a useful reference during the model development phase (Cassina et al., 2006).

The model was successfully tested on ten application cases, in the context of the PROMISE project, and is now being proposed as the basis for a new-generation (closed-loop) PLM standard.

2 The PROMISE project

The PROMISE project is developing appropriate technologies, including product lifecycle models, Product Embedded Information Devices (PEIDs such as Radio Frequency Identification – RFID systems) with associated firmware and software components, and tools for decision making based on data gathered during a product's life (Kiritsis et al., 2003). The aim is to enable and exploit the seamless flow, tracing and updating of information about a product, after its delivery to the customer and up to its final destiny (deregistration, decommissioning) and then back to the designer and producer. In the project viewpoint, product lifecycle was divided into three main portions. The first one is the Beginning of Life (BOL), which is composed of product development and product supply, i.e. the manufacturing process, the logistics, etc. Then, the second phase is the Middle of Life (MOL), where the product is used and maintained. Finally, it comes the End of Life (EOL) where the product is destroyed or rather dismantled and recycled (Jun et al., 2007b).

The breakthrough contribution of PROMISE is, in the long term, to allow information flow management to go beyond the customer, to close the product lifecycle information loops and to enable the seamless transformation of product lifecycle information to knowledge. The PROMISE R&D implementation plan includes fundamental and applied research activities in the disciplines of information systems modelling, smart embedded systems, short and long distance wireless communication technologies, data management and modelling, statistical methods for predictive maintenance, EOL planning, adaptive production management and Design for X (where the 'X' stands for maintenance, manufacturing, remanufacturing, etc., depending on the specific application).

The prototypical PROMISE system is being applied to ten application scenarios, covering the whole set of product lifecycle phases in the automotive, railway, heavy-load vehicles, Electrical and Electronic Equipment (EEE), instrumental and white goods sectors. The different software and hardware systems, together with the related infrastructures, building up the PROMISE system, are briefly presented in the following.

The PROMISE PDKM system manages both product data collected from the field via smart PEIDs, and knowledge created and updated from this field data, in order to enhance, e.g. the design of new products in the future.

The PROMISE Decision Support System (DSS), which is also part of the PDKM system, is devoted to the support of lifecycle decision making activities, thus providing the analytical basis to the whole project. This can be done by having defined appropriate decision strategies to be applied in the different application scenarios, as well as the related algorithms implementing these strategies.

A set of PEIDs, in particular RFID active and passive tags, different kinds of sensors and on-board computers, with the related embedded and backend software systems, solve the application-specific product lifecycle issues around the problem of how to trace the product and collect data from the field.

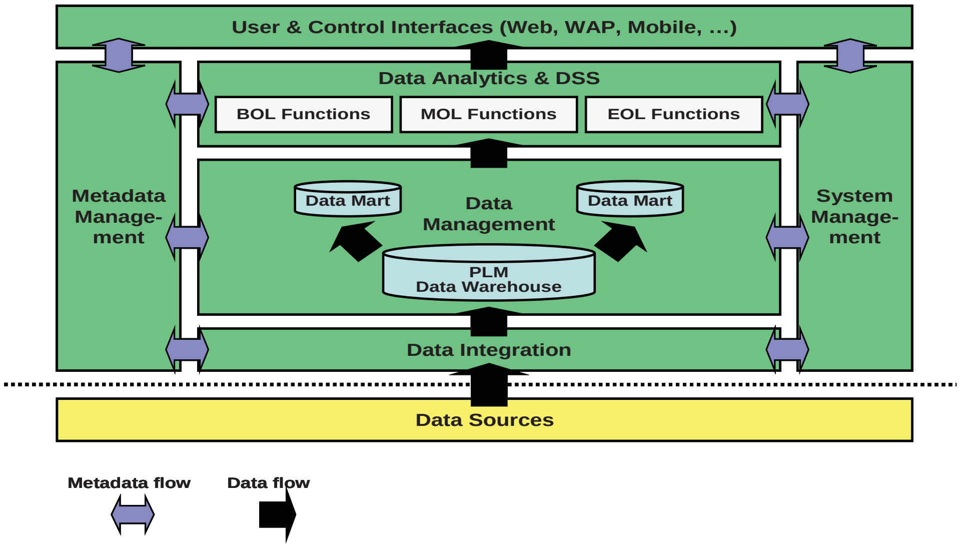
3 The PROMISE PDKM system

The PROMISE PDKM system aims at integrating and managing data from all lifecycle phases of products, in particular, from design, development, production, through use and maintenance, to recycling and finally, to the end of life, in order to support comprehensive data analysis in business intelligence applications.

Figure 1 illustrates the overall PDKM architecture with its components and the data sources that are themselves not part of the PDKM but feed it with product lifecycle data.

Concerning these data sources, it suffices to say here that the relevant data is imported from BOL, MOL and EOL operational systems. BOL-specific data sources include databases, such as PDM and Supply Chain Management (SCM) systems, which are indeed largely static. On the other side, MOL-specific and EOL-specific data is captured in a real-time manner from products either using the attached PEIDs, or in PDM and field databases as required for the specific MOL and EOL operations.

Figure 1 Architectural overview of the PROMISE PDKM system (see online version for colours)



The shown architecture is a model of abstract *layers* and *towers*, which represent logical building blocks of the architecture, coupled one with another via interface functions. The towers provide functions that concern several layers at the same time. In the following, a brief explanation of these layers and towers is provided.

Data integration layer. This layer consists of all tools, scripts and programs utilised to import source data into the PDKM system. Before moving data into the PDKM, they are transformed, cleaned and unified under a consistent global representation, also called the *global schema*. This is necessary as different sources may use different notations and formats to represent and store their data.

Data management layer. This layer aims at providing an application-independent semantic view of data. In particular, all relevant data is kept in a central database, the *PDKM data warehouse*, which implements the global schema, thereby providing a uniform data representation for queries and analyses. For supporting specific analysis tasks, usually only subsets of the data warehouse are needed, which can be provided in preformed views, the so-called *data marts*.

Data analytics layer. This layer comprises all algorithms and tools employed to perform data analysis and to support decision making/problem solving. The methods to be supported range from simple browsing and searching techniques, querying and reporting capabilities, to complex Online Analytical Processing (OLAP) and data mining algorithms. On top of the collection of such generic methods, customised analysis applications were developed, e.g. for predictive maintenance purposes.

User and control interface layer. This layer implements the interface of the PDKM system to users and to other applications. In particular, it comprises the various interface functionalities offered by single architecture components for different types of users. According to the types of users and to the focus of their work, e.g. design engineer, service man, dismantler, but also power users being in charge of maintaining the entire PDKM system, customised interfaces should be provided with different views on data and allowed analysis functions.

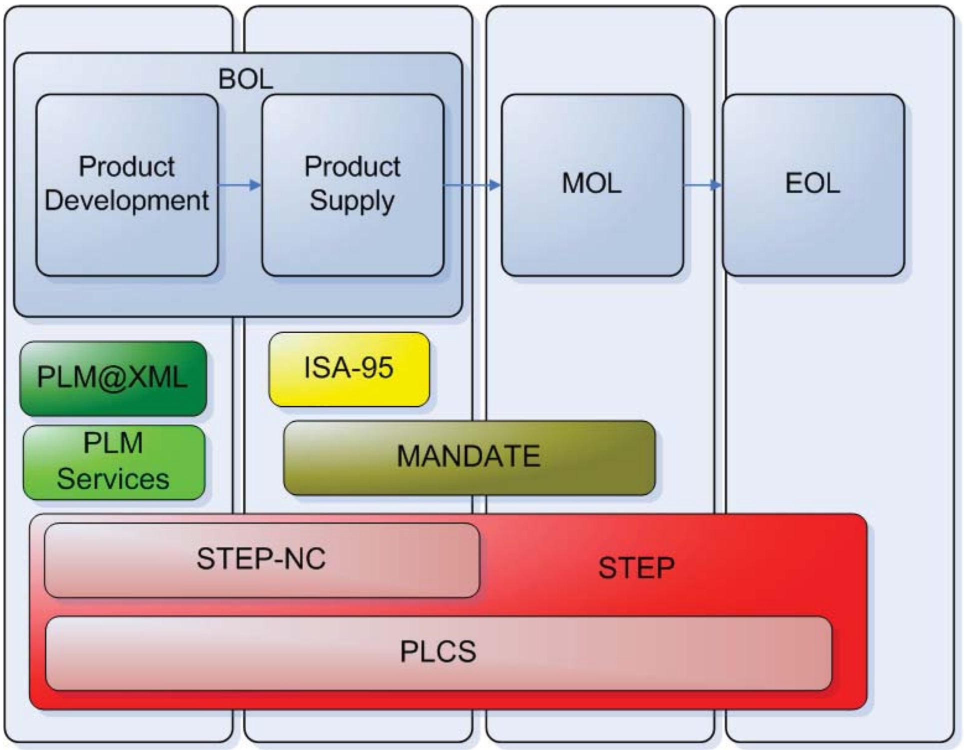
Metadata management tower. This tower aims at collecting and managing metadata from the entire PDKM architecture for documentation purposes. Metadata is used to describe data, users and processes in the construction, maintenance and use of the PDKM system.

System management tower. This component addresses the ongoing maintenance and use of the PDKM system. Relevant tasks of system management include scheduling of data import and update programs, management of users and user groups, authorisation/authentication, performing backup and load balancing activities, etc.

4 Analysis of enterprise standards

Before starting the modelling activities, an analysis of existing PLM standards was carried out, which revealed that, though many proposals exist, each of them is focused on a specific area of the product lifecycle and almost none of them (the only exception being Product Lifecycle Support (PLCS) – see further on) includes all the pieces of information needed to be managed during the whole lifecycle chain, as shown in Figure 2.

Figure 2 Standards through lifecycle phases (see online version for colours)



Six standards were studied in details, since they revealed to be complementary one to each other in the PLM approach, namely ANSI/ISA-95 (or the new ISO/IEC 62264 standard which is derived from it), Mandate (ISO 15531), PLM Services, PLM@XML, and two application protocols of the STEP (ISO 10303) standard: PLCS (ISO 10303-239) and STEP-NC (ISO 10303-238/ISO 14649). These standards share in common some properties and features, but are also distinguished by a lot of remarkable differences. First of all, they were designed by different organisations, with different aims and for different targets. STEP-NC, PLCS and Mandate can be grouped together, because each of them is an International Organization for Standardization (ISO) standard. STEP-NC (2003) and PLCS (2005) belong to the group of standards derived from the STEP standard, which is an industry standard for product data representation and it is composed of several parts (application protocols) whose aim is to focus on a specific industrial context. There are application protocols for product design, for mechanical and electrical engineering, for sheet-metal manufacturing, for product assembly, for automotive industry and so on. PLM@XML (2003) is an open standard developed mainly by Electronic Data Systems Corporation (EDS) and dealing with the product design phase. ISA-95 is an American National Standard Institution (ANSI) standard, but its first part, ANSI/ISA-95.00.01, is also an ISO standard (ISO 62264-1). ANSI/ISA-95 Parts I, II and III describe the interfaces and activities between an enterprise's business systems and its manufacturing control systems: it focuses, thus, mainly on the area corresponding to our product production phase. Finally, Object Management Group (OMG) PLM Services 1.0 is a ProSTEP iViP working result and is standardised by the OMG. It is the first standard comprising current XML and web services technologies with a STEP data model, thus, providing both syntax and semantics. It supports cross-company, cross-domain, cross-system and cross-technology collaboration and provides a solid foundation for collaborative engineering (Feltes, 2005).

Another initiative in the same area, not shown in the figure, is represented by the Physical Mark-up Language (PML), proposed by the Auto-ID laboratories (McFarlane et al., 2003). PML is intended to provide a simple, general language for describing physical objects for use in remote monitoring and control of the physical environment. PML was thought as a part of a wider structure, built around four major components: electronic tags, Electronic Product Code (EPC), PML and Object Naming Service (ONS).

Among the standards cited so far, the only one including the whole set of lifecycle phases is PLCS. This shares with the PROMISE PDKM system initiative the topic on which it is centred, that is the single-item lifecycle, but from very different prospective and needs. The majority of the standards cited above are on the opposite built around the 'product type' concept, rather than around the 'product item' concept. PLCS aims at very complex products, and also standardises all the procedures and the activities that follow it. The PROMISE PDKM system instead focuses more on simple products and describes only technical details. It is not technically unfeasible to use PLCS to describe simple and mass products, as well as to use the model developed for the PROMISE PDKM to describe complex products (e.g. military aircrafts). However, outside their sector of applicability both models will not perform at best, and their usage will not be sound from a business point of view.

Since there was no readily available standard to cover the purposes of the PROMISE PDKM system, a new data management model had to be developed. After that, since the model was successfully implemented in ten different application scenarios, it is now under proposal as a possible way to fill the lacks of a PLM standard for mass and consumer products.

5 The conceptual model of the PDKM system

The model proposed in the following pages provides a conceptual view on the PROMISE PDKM system, and in particular on its data management layer.

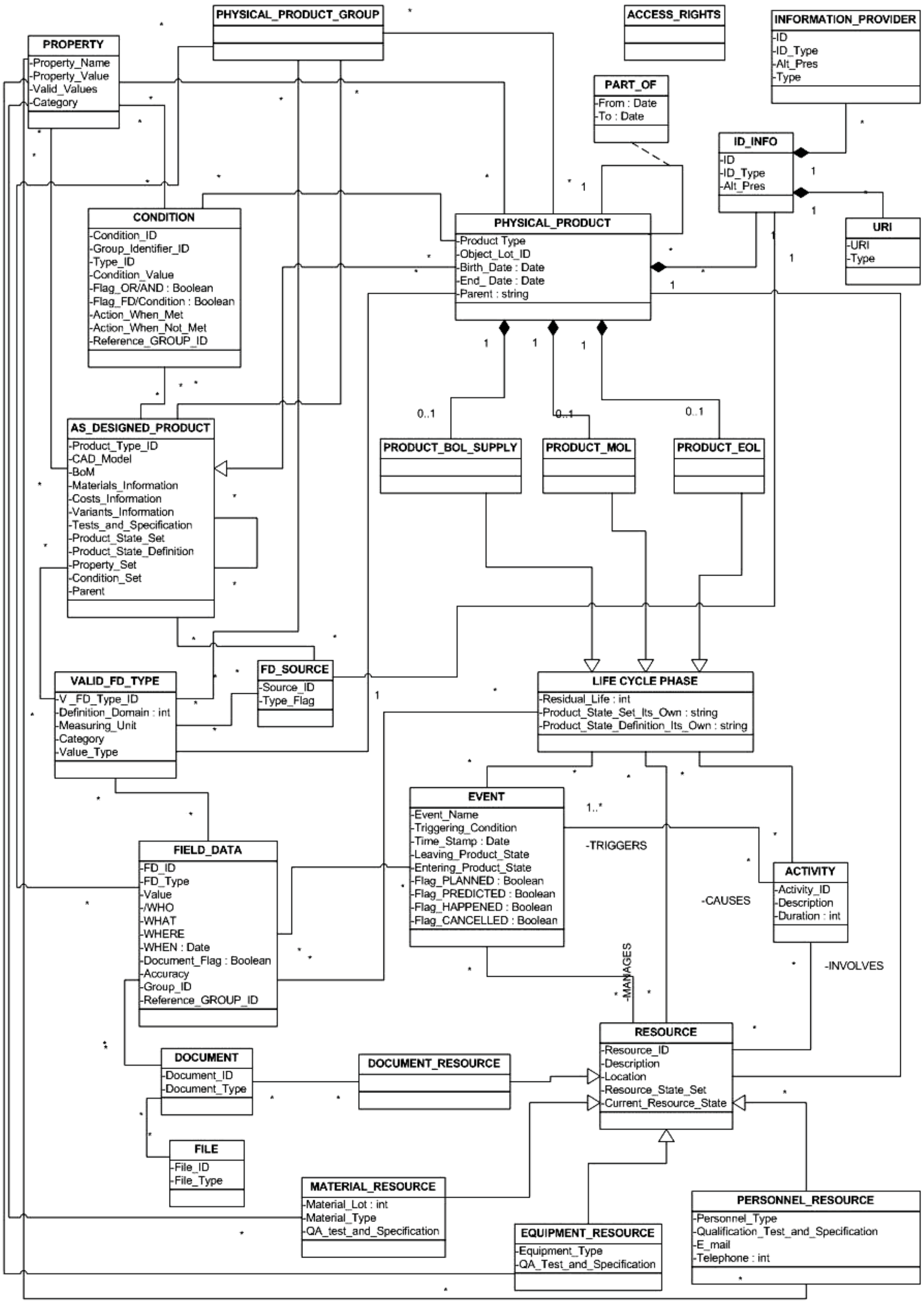
The model is presented here by means of the UML 2.0 class diagram notation. This is sufficient to provide a complete static view on the PROMISE PDKM System, on its objects, on the related attributes, as well as on the existing associations among them (Fowler, 2003).

To enable the PROMISE closed-loop approach to PLM, product lifecycle data management must go beyond its commonly known frontiers. To do this, the model was developed with the following features:

- The focus was shifted from information on product types to information on product items, virtually each product item. This represents a new approach to tackle PLM issues, which requires the identification and tracing of each physical product entity, the access to all of the data available on it, in particular data collected from the field while the product is being operated/used, and finally on the use of this data by the DSSs commonly adopted in each scenario to support decision makers in the value creation process.
- Product items at the different levels of the product structure can be identified, and the related information can be properly collected and managed. This is possible for products with structures ranging from a very low degree of complexity, eventually one-piece products, up to a very high degree, such as cars or trucks.
- Moreover, it is possible to manage information on product structures related to both products 'as-designed' (those typically managed by currently available PLM systems) and physical products. This last type of information carries within itself an always-up-to-date description of the identities of each component/subassembly presently part of the product.
- The problem of correctly identifying and tracing each item during its life has also been tackled. The PROMISE PDKM system is in this sense compliant with the most widely adopted approaches to product identification and lifecycle traceability.
- All of the data on physical products to be collected from the field, assemblies/ subassemblies and components, can be also managed. Pieces of information such as *who*/*what* collected each data record, *what* is the meaning of each record, *when* it was collected and *where* each record can be retrieved, are available.
- The derivation of useful knowledge from field data, as well as its management and updating, is also supported.
- Finally, a semantic description of the different aspects of the product lifecycle phases in which a PLM scenario is interested are also available. This comprises information on the major lifecycle events that are expected to happen, on the different PLM activities related to each particular scenario, as well as on the equipment, personnel and other resources involved in the closed-loop approach to PLM.

Figure 3 shows the PDKM semantic object model. In the following, a detailed description of its elements is proposed, in order to outline how the requirements stated above are met by the proposed set of classes, attributes and associations.

Figure 3 The PDKM semantic object model



5.1 Identification of product items

The PROMISE approach to PLM is a ‘product instance centric’ one. Identification at the item level is indeed a key factor in order to achieve closed-loop PLM. The PHYSICAL PRODUCT class conceptually represents each single product item, but also, if needed, each instance of product component or subassembly, depending on the level of detail. This class is further discussed in Section 5.2 below.

A correct identification process of these items of possibly different nature is tackled by the PROMISE approach moving from the PEID concept, which can be defined here as the joint between a product/article/machine/component and its electronic representation within an information system. The PEID is responsible for uniquely identifying a component and is a means for collecting information on the product from the environment, either directly or via reference to a backend system. As the PEID may not be connected to the backend system all the time, it should provide means of local information storage, until the information can be relayed to the backend system at a later point in time. The communication with the backend system can either be carried out directly from the PEID using GPRS, internet connection, etc. or via a PEID reader (device controller).

These capabilities of the PEID concept and related physical devices must be exploited by the PDKM system via a proper information model. Focusing here more on the identification problem, the present model is compliant, at least in a conceptual viewpoint, with the most common traceability and identification systems, in view of a potential wide acceptance and implementation of the PROMISE approach. The Dialog System, developed by the Helsinki University of Technology (Kärkkäinen et al., 2003a; Kärkkäinen et al., 2003b; Främling et al., 2006), the WWAI-World Wide Article Information concept (http://www.wwai.org) and the AUTO-ID EPCGlobal proposal (Cole and Engels, 2005; Soylemezoglu et al., 2006), are just some of these existing approaches.

The PDKM classes defined for these purposes are the ID_INFO, the INFORMATION_PROVIDER and the Uniform Resource Identifier (URI) classes, which together enable the identification of product instances and the retrieval of the related information.

A key role is played here by the ID_INFO class, where one can find the item identifier (ID attribute), the coding schema used (ID_Type attribute) and eventually the representation of the identifier in other additional/alternative formats (Alt_Pres attribute). There are then two kinds of links to additional information, represented by the URI and the INFORMATION_PROVIDER classes. The former identifies the external data sources which are linked to the ID, when relevant for some scope (as in the Dialog System, where URI stands for Uniform Resource Identifier). The URI attribute represents for instance the IP address on the World Wide Web where the information can be retrieved. The INFORMATION_PROVIDER class contains heterogeneous information that can be used to control requests for information by a traceability system, such as the definition of the adopted inter-enterprise communication systems (which, in traceability systems, take care of identifying the information providers).

Finally, the ACCESS_RIGHTS class plays a more abstract role and is representative of the PDKM infrastructure aiming at the control of user access, by verifying rights and capabilities associated to user profiles and user roles. These access rights should be managed at different levels and concern, directly or indirectly, most of the classes contained in this semantic object model. The presence of this class here without connections to any other class is only to underline at a semantic level the need for managing these pieces of information in the PDKM.

5.2 Description of product structures

In order for the model to represent both atomic products (i.e. one-piece products) and complex ones, some classes had to be devoted to the representation of different kinds of product structure.

A first example is provided by the PHYSICAL_PRODUCT class, which states the product type, the lot to which it belongs, the 'birth date' of the product, and the 'end date' in case the product has reached the end of its life. The Parent attribute acts instead as a link to the father node in the tree representing the product physical structure, in case of complex products, where it is also possible to track the history concerning the physical components/subassemblies belonging to a single physical product object. The PART_OF association class was developed for this purpose, and the From and To attributes are there to represent the time instants when the component/subassembly is, respectively, attached to and detached from the physical product. It is also possible to keep record of past components/subassemblies, based on previous records of objects of this association class.

Another important class is the AS_DESIGNED_PRODUCT class, which describes on the other hand the product 'as-designed' structure, with all the needed information, such as CAD data (CAD_Model attribute), Bill Of Materials (BoM attribute), information on product costs and variants, on product materials, and all the other pieces of information which are typically stored and managed by PDM as well as PLM systems.

The self-association between AS_DESIGNED_PRODUCT and itself, together with the Parent attribute, plays a similar role to those of the PHYSICAL_PRODUCT class, in representing complex structures.

Product_State_Set and Product_State_Definition assert the list of states an object of a certain product type can pass through during its lifetime, and in particular the definition of the set of parameters identifying each state and a definition of how it can be concluded that the product has entered or left a given state.

Finally, the set of properties and conditions applicable to the given product type are also referred by the Property_Set and Condition_Set attributes, respectively. Section 5.3 is devoted to a description of these particular attributes.

5.3 Properties and conditions

The proposed model is also capable of representing specific properties and conditions which must be verified, or must hold, for some specific product types and/or product items.

The PROPERTY class, originally inspired by the ISA-95 standard, defines the name of the property (Property_Name attribute), the possible values (Valid_Values) the property can take, the actual value (Property_Value) and some other kind of specification (Category), if needed. This class is associated to both the PHYSICAL_PRODUCT class and the AS_DESIGNED_PRODUCT class, because it may be the case that for some specific instances of a given product type, i.e. for one or more product items of that type, some specific condition must be verified or property must hold, in addition to those which are common to all of the product items of the same type.

The CONDITION class was developed to define some either atomic or complex kind of condition which must be checked in some product lifecycle scenario. For instance, it can be important to check if the current reading of some sensor attached to the product is over a pre-defined threshold, and eventually to start the needed activities in order to perform the needed maintenance before the product breaks down. The Condition_ID attribute univocally identifies the condition, while the Group_Identifier_ID and Reference_Group_ID attributes are used to define complex conditions, by grouping atomic conditions together. The Type_ID attribute states if a condition relates to a property of a product type/instance or to some kind of data collected on the field (in this case, the field data type must be specified, as well as the interested data source). Finally, the actions to be taken in case the condition is met/not met must be specified (Action_When_Met and Action_When_Not_Met attributes, respectively).

Finally, the PHYSICAL_PRODUCT_GROUP class is intended to model the possibility for the PDKM system and its users to group together a set of PHYSICAL_PRODUCT objects on the basis of some commonalities, e.g. concerning their BOL as-designed structure or the field data collected. This is stated by the associations linking the PHYSICAL_PRODUCT_GROUP class with the AS_DESIGNED_PRODUCT class, the PHYSICAL_PRODUCT class and the FIELD_DATA class. A brief description of the latter is provided in Section 5.5.

5.4 *Lifecycle phases and related information*

Different PLM scenarios have different requirements not only in terms of how many components must be identified and are subject to field data collection, but also in terms of the number of lifecycle phases during which the product items are monitored, identified, tracked and traced.

The proposed model must cover these different needs. For this reason, the three classes named PRODUCT_BOL_SUPPLY, PRODUCT_MOL and PRODUCT_EOL were created. The first class refers to the pieces of information related to the BOL phase of a product instance, from the production phase to the final delivery of the product to the customer. Recall that the information concerning the design of the product is excluded from this class, since objects of all of these three classes can only be instantiated with reference to existing physical product items. The PRODUCT_MOL class refers to the pieces of information related to the MOL of a product instance, i.e. the usage phase and the maintenance/service phase. Finally, the PRODUCT_EOL class virtually refers to the pieces of information related to the whole set of possible EOL scenarios of a product instance (e.g. the remanufacturing phase, the recycling phase, etc.), depending on the specific case. Cardinalities of the composition associations linking PRODUCT_BOL_SUPPLY, PRODUCT_MOL and PRODUCT_EOL to PHYSICAL_PRODUCT are there to indicate that only one object of each of these classes can be instantiated for each physical product item. Moreover, these objects must be created in the ‘natural’ sequence following the lifecycle phases sequence, and above all, it is not mandatory in principle to instantiate the object of the PRODUCT_EOL class, since this depends again on the specific case. For instance, a PLM scenario only focused on predictive maintenance of boilers or machine tools may instantiate only the PRODUCT_BOL_SUPPLY and the PRODUCT_MOL objects, but, on the opposite, a PLM scenario managing the whole life of a car, e.g. because of the existing laws (see the EU directives on End-of-Life-Vehicles – ELV) must also comprise the detailed description of the EOL information.

The LIFE_CYCLE_PHASE class then describes some important issues such as the residual life of a product component (Residual_Life attribute), or the definition of the set of states in which a product item can be at a given moment in time (Product-State_ Set_Its_Own and Product_State_Definition_Its_Own attributes). Note that in this last case the attributes on states definition are not part of the physical object, and the reason is that for different lifecycle phases a different state list could have to be defined.

5.5 Field data

Field data can be of different types (VALID_FD_TYPE class), and is collected by means of sources such as sensors (FD_SOURCE class). It might be organised in documents (DOCUMENT class) with physical files (FILE class) attached.

The FD_ID attribute univocally identifies each field data record, while the FD_Type attribute states the field data type (e.g. that a given field datum represents a temperature of a certain sensor). The Document_Flag attribute says if the field data has an attached document related to it, while the Value and Accuracy attributes should be self explaining.

The WHO attribute says ‘who’ is responsible for the field data measurement, i.e. which is the source of the field data. This information can be also derived from the corresponding object of the FD_SOURCE class linked to the same FIELD_DATA object. The WHAT attribute explains in details what the field data stands for, i.e. the meaning of the data itself, while the WHERE attribute states, if needed, the location where the measurement was carried out. The WHEN attribute then represents the time stamp indicating the moment in time when the measurement was carried out.

Finally, the Reference_GROUP_ID and the Group_ID attributes are used when there arises the need of grouping some records of the same field data type together, e.g. because of the need of clustering in some way the data before analysing it.

5.6 Events, resources and activities

In order to describe with a deep level of details each lifecycle phase, and everything happening to the product during each phase, the model makes use of the EVENT, RESOURCE and ACTIVITY classes, which represent a second and most detailed method to manage information about the physical product within the model, being focused on the management of more complex situations than those well-described only by simple field data.

These classes were partially inspired by terminology in the field of discrete event simulation on the one hand, and by some similar classes within the STEP-PLCS standard on the other hand.

The motivation for such classes follows naturally from the need of the user of PROMISE technologies to model the information concerning the lifecycle phases of interest. Examples of possible uses of these classes are, within a predictive maintenance scenario, events such as ‘product breakdown’ or ‘maintenance mission finished’. These must be properly managed, using resources such as maintenance teams, or activities such as the maintenance itself, and must have a corresponding set of objects into the semantic information model.

The final aim is to rely on this information in order to make decisions and definitely create value. As an example, the availability of resources such as free hours of the garage crew to be possibly allocated, together with the availability of the needed materials and equipments should be checked, and the related maintenance activities should be economically planned and then managed.

The associations among the RESOURCE, ACTIVITY and EVENT classes basically state that an event triggers an activity, which involves some resources, which in turn manage the event. It is also possible that an activity causes an event, such as ‘replacement of component X finished’.

The attributes state that an event is something related to a specific time instant, while an activity generally concerns a time interval and is thus associated to a time duration. An activity has at least two events associated with it: the event ‘activity starts’ and the event ‘activity ends’, but it can surely have more than two. The event is triggered by some condition and may cause the shift of the product state from some ‘state A’ to another ‘state B’. Again, one can mark with a proper flag if the event is a planned event, or if it is a predicted event, or again if the event has already happened, or has been cancelled because it cannot happen anymore (refer to the Flag attributes in the EVENT class).

Finally, the resources can be human beings (PERSONNEL_RESOURCE class), equipments (EQUIPMENT_RESOURCE class), materials (MATERIAL_RESOURCE class) and documents (DOCUMENT_RESOURCE class), according to the ISA-95 standard.

Some of the pieces of information related to these resources, the most simple ones, are provided as attributes, and some other, more complex, are specified as objects of the PROPERTY class. Examples of objects of these classes can be the maintenance crew as objects (e.g. one for each person) of the PERSONNEL_RESOURCE class, the tools for performing the maintenance activities as objects of the EQUIPMENT_RESOURCE class, the spare parts needed as objects of the MATERIAL_RESOURCE class and finally the product user manual or the maintenance manual as objects of the DOCUMENT_RESOURCE class. Moreover, for each resource a set of possible states is defined, and the current state is recorded.

In conclusion, there also exists an association between the RESOURCE class and the PHYSICAL_PRODUCT class, to state that it sometimes can be possible that the object of the PLM system which is a resource for one company, e.g. a truck used for the delivery of the products produced by the company, may be a product item for another company, e.g. it can be part of a fleet of trucks on which the truck builder/dealer performs predictive maintenance.

6 An example from the PROMISE application scenarios: closed-loop PLM in the white goods sector

One case among the ten PROMISE application scenarios will be described in the following, showing in particular how the proposed model can provide the needed semantic means to tackle closed-loop PLM in the white goods sector. The Indesit scenario, dealing with the analysis and management of the lifecycle of refrigerators, was chosen to informally show here the applicability of the model. More formal verifications are out of scope for the present paper.

6.1 The Indesit case

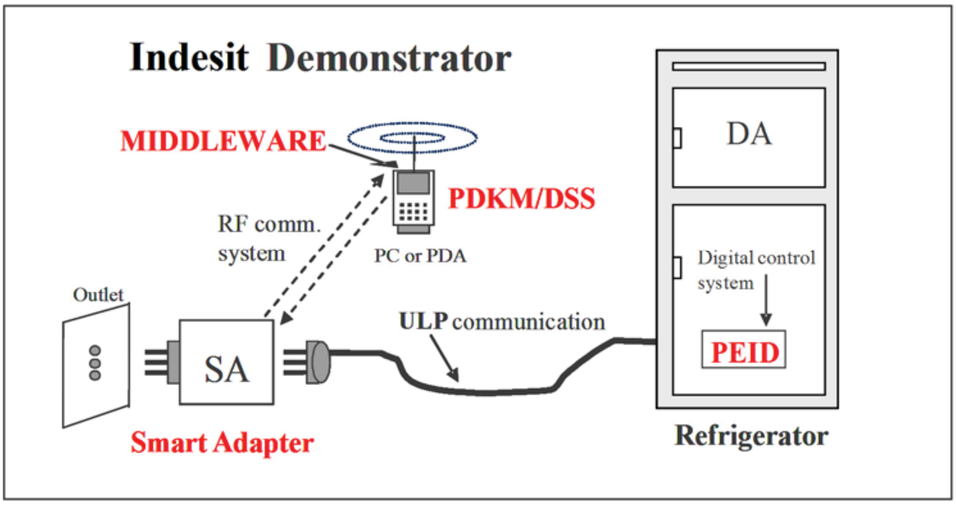
The physical elements of the Indesit application scenario in PROMISE are a refrigerator, indicated (Figure 4) as Digital Appliance (DA), with its digital control board acting as the PEID in this case, an interface device, indicated as Smart Adapter (SA), which is placed between the power cable of the household appliance and its electric plug (Outlet) and finally a wireless communication link (RF communication system) between the SA device and the remote monitoring centre, where the PROMISE DSS runs predictive maintenance algorithms, allowing both long term diagnostics of the product and the management of maintenance missions and of spare parts. The main goals of Indesit in PROMISE activities are as follows:

To demonstrate the improvements brought by the project activities into the in-line testing process (BOL phase) of the refrigerator, where the product is tested in order to check its proper functioning. During this phase, some of the measured parameters (electric load characteristic parameters) are stored into the memory device of the PEID, where they will be later read by the predictive maintenance algorithms as one of their fundamental inputs.

To demonstrate the improvements brought by the project activities into the product installation process, where the DA and the SA are installed by a technician in a domestic environment and automatically recognised by a home network controller (emulated by a local PC in the demonstrator) using Universal Plug-and-Play (UPnP) technology.

To demonstrate the breakthrough innovation represented by the predictive maintenance operations, where field data, sent by means of the SA, are received through the so-called PROMISE MIDDLEWARE (installed in a local PC in the demonstrator), stored in the PROMISE PDKM database and processed by the PROMISE DSS (also both installed in a local PC in the demonstrator).

Figure 4 Structure of the Indesit demonstrator in PROMISE (see online version for colours)



Field data (statistical and diagnostic data) flow can be schematically described as follows. Field data, coming from sensors and actuators, are stored by the control system of the refrigerator into its non-volatile memory. These data are first sent to the SA (which basically provides data related to the energy consumption of the household appliance) and later to the PDKM system, and then analysed by the PROMISE DSS, in order to eventually find out malfunctioning problems on one or more of the refrigerator components. If an incipient failure is detected or predicted, an email is sent to the service company, which is thus enabled to perform predictive maintenance actions on the refrigerator.

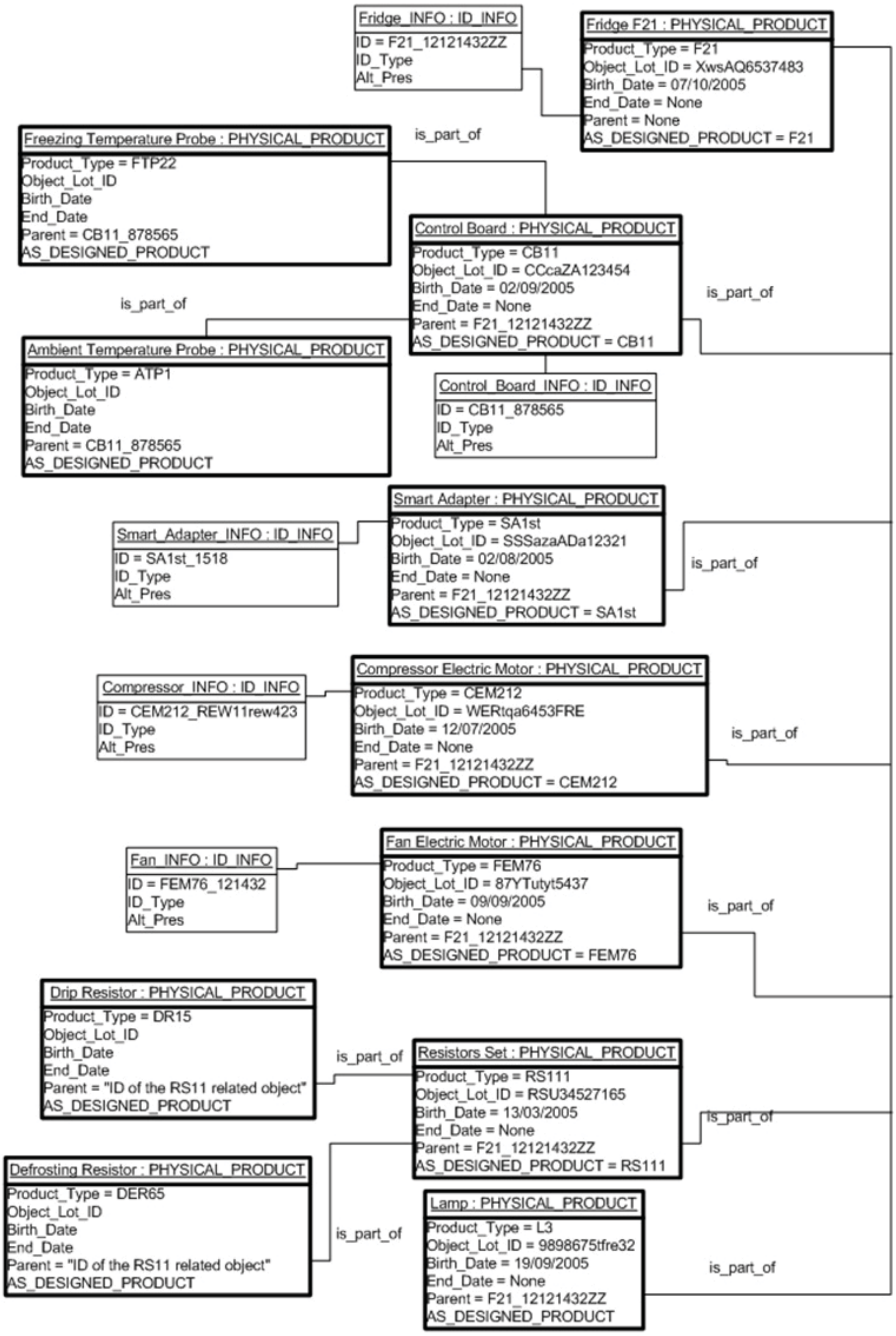
6.2 MOL structure of a typical refrigerator

Figure 5 presents an UML 2.0 object diagram showing an exemplificative instantiation of the classes of the PDKM conceptual data model in the Indesit case. The bold objects of the PHYSICAL_PRODUCT class represent the different components of a typical refrigerator which are involved in the implementation of a real-world predictive maintenance application, namely the refrigerator itself (as a whole), and its different components as follows:

- the electronic control board, with the related sensors, i.e. the freezing temperature probe and the ambient temperature probe
- the smart adapter, fundamental enabler for communication with the predictive maintenance central platform
- the electric motor of the compressor
- the electric motor of the fan
- the set of resistors, whose status is fundamental for running the predictive maintenance algorithms, itself divided into the drip resistor and the defrosting resistor
- the lamp.

The objects of the ID_INFO class are there to keep record of the identifiers of the product and/or of the components, where necessary. Moreover, to keep record of the design data related to the product and its different components, permanent links to proper objects of the AS_DESIGNED class (not represented explicitly in the diagram) are present by means of attributes of the objects of the PHYSICAL_PRODUCT class.

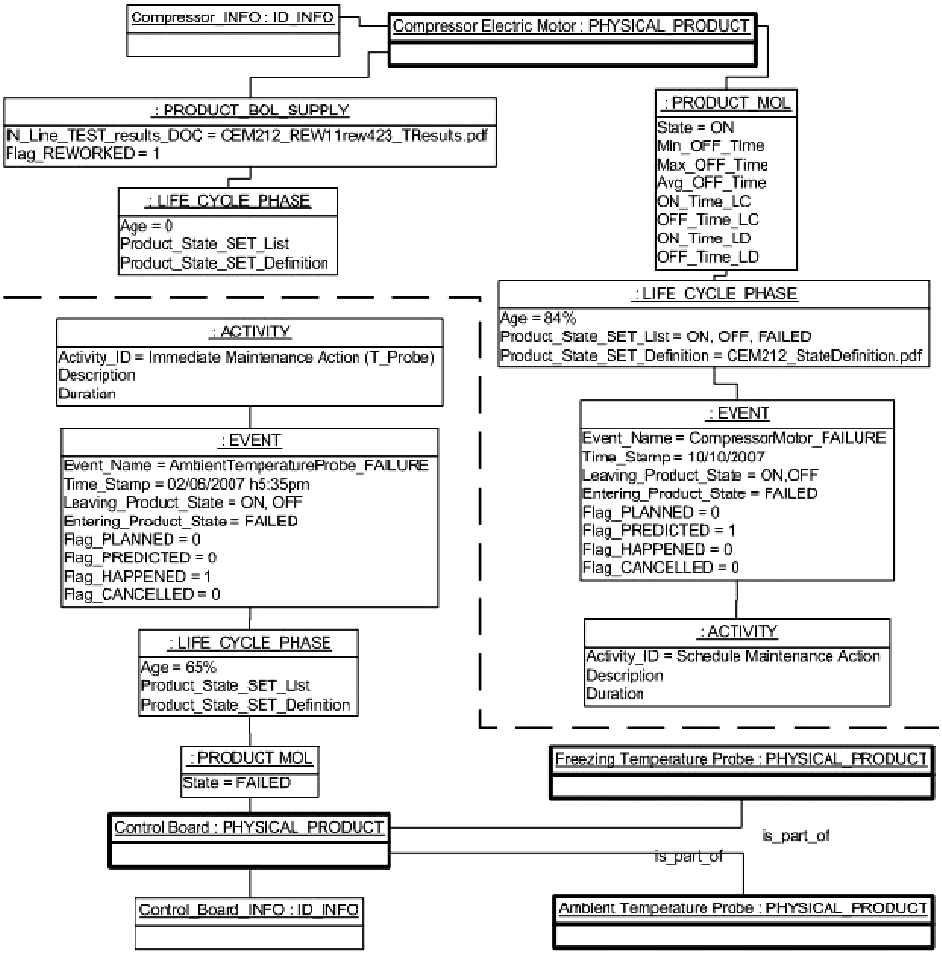
Figure 5 MOL structure of a typical refrigerator



6.3 Predictive maintenance of refrigerators

Figure 6 represents an instantiation of objects related to a realistic case of maintenance of a typical refrigerator. Two components of the refrigerator are indicated, namely its compressor and its control board. The upper portion of the figure indicates that the considered compressor (see the related PRODUCT_BOL_SUPPLY object) was reworked while being produced, and that the results of the final in-line tests are available in a specific file of the file system. More precisely, after the rework the compressor was certified to be correctly functioning and (see the LIFE_CYCLE_PHASE object attached to this PRODUCT_BOL_SUPPLY object) then its 'Age' attribute was appropriately set to '0'. Though not explicitly shown in the figure, the same type of setting was done for all of the rest of the refrigerator's components, after passing the final in-line inspection, by changing the attributes of the related objects.

Figure 6 Example of predictive maintenance of the refrigerator


Compressor_INFO : ID_INFO
Compressor Electric Motor : PHYSICAL_PRODUCT
: PRODUCT_BOL_SUPPLY
IN_Line_TEST_results_DOC = CEM212_REW11rew423_TResults.pdf
Flag_REWORKED = 1
: LIFE_CYCLE_PHASE
Age = 0
Product_State_SET_List
Product_State_SET_Definition
: PRODUCT_MOL
State = ON
Min_OFF_Time
Max_OFF_Time
Avg_OFF_Time
ON_Time_LC
OFF_Time_LC
ON_Time_LD
OFF_Time_LD
: LIFE_CYCLE_PHASE
Age = 84%
Product_State_SET_List = ON, OFF, FAILED
Product_State_SET_Definition = CEM212_StateDefinition.pdf
: ACTIVITY
Activity_ID = Immediate Maintenance Action (T_Probe)
Description
Duration
: EVENT
Event_Name = AmbientTemperatureProbe_FAILURE
Time_Stamp = 02/06/2007 h5:35pm
Leaving_Product_State = ON, OFF
Entering_Product_State = FAILED
Flag_PLANNED = 0
Flag_PREDICTED = 0
Flag_HAPPENED = 1
Flag_CANCELLED = 0
: EVENT
Event_Name = CompressorMotor_FAILURE
Time_Stamp = 10/10/2007
Leaving_Product_State = ON,OFF
Entering_Product_State = FAILED
Flag_PLANNED = 0
Flag_PREDICTED = 1
Flag_HAPPENED = 0
Flag_CANCELLED = 0
: LIFE_CYCLE_PHASE
Age = 65%
Product_State_SET_List
Product_State_SET_Definition
: ACTIVITY
Activity_ID = Schedule Maintenance Action
Description
Duration
: PRODUCT_MOL
State = FAILED
Freezing Temperature Probe : PHYSICAL_PRODUCT
Control Board : PHYSICAL_PRODUCT
is_part_of
is_part_of
Control_Board_INFO : ID_INFO
Ambient Temperature Probe : PHYSICAL_PRODUCT

The rest of the objects regarding the compressor say that a failure was predicted to happen (EVENT object) around late October 2007, and that for this purpose a proper maintenance action must be scheduled (ACTIVITY object). The current state of the compressor is ‘ON’ (PRODUCT_MOL object), i.e. it is correctly functioning, and its age is about 84% (LIFE_CYCLE_PHASE object). The failure prediction as well as the estimation of the aging of the compressor were performed in the past by the DSS, which is part of the PDKM, and the related objects were created/modified as a consequence.

The lower portion of the figure shows that a failure of one of the temperature probes attached to the control board happened on 2 June 2007. In particular, the failure is related to the Ambient Temperature Probe. Thus, the control board of the refrigerator results to be (PRODUCT_MOL object) ‘FAILED’. The LIFE_CYLCE_PHASE object also shows the additional information that, at the moment of the failure, the age of the probe was about 65%. This failure requires an immediate action.

As a consequence, the central management division of the company is notified about this need, and already aware of the forthcoming potential failure of the compressor (and thus of the related thermodynamic circuit), can appropriately schedule a maintenance action with the related visit to the customer’s site, where both problems can be solved together, saving time, cost and avoiding mishaps to the customer.

7 Standardisation efforts

The instantiation of the model shown in the previous pages is just one of the ten that were carried out within the PROMISE project, showing the ability of the proposed model to satisfy the needs of very different products ranging from electronic devices to cars and trucks. Due to these successes, this model has been proposed as a possible solution for the lacking of a standard that can be implemented in a simple, fast and inexpensive way for products with a low and medium complexity, such as the consumer goods. In the context of this standardisation effort, the conceptual semantic reference model is being improved with the contribution of interested people within the PLM community.

8 Conclusions

This work presented a new-generation conceptual reference model for closed-loop PLM, aiming at the interoperation with different software/hardware systems and at the storage of virtually all kinds of product lifecycle data. The model was implemented in the context of the PROMISE project (within the MySAP-PLM suite), and tested in ten different applications, but alternative implementations are on the way, in order to create an open reference implementation, but also a fully object-oriented implementation.

Finally, the conceptual model became the basis for a standardisation effort in the context of closed-loop PLM. Results in this direction are expected in the forthcoming years.

Acknowledgements

This work was partly funded by the European Commission through the FP6-IST-IP-507100-2004 project entitled PROMISE (PROduct lifecycle Management and Information tracking using Smart Embedded systems). The authors wish to acknowledge the European Commission for their support, as well as the whole PROMISE consortium for the contribution in the development of various ideas and concepts presented throughout the paper. The authors want to give a special thanks to InmediasP GmbH and Indesit Group, and particularly to Eng. Giovanni Felicetti, Eng. Altug Metin and Eng. Michael Marquard for the engagement, the good ideas and good time spent together in this work.

References

Cassina, J., Taisch, M. and Terzi, S. (2005) 'Towards an intelligent extended product', *IFIP*.

Cassina, J., Tomasella, M., Marquard, M., Metin, A., Matta, A. and Taisch, M. (2006) 'Development of the semantic object model for a PDKM System', *ICE*, Milan, Italy.

Cole, P.H. and Engels, D.W. (2005) *Auto ID – 21st Century Supply Chain Technology*, AutoID Whitepaper.

Feltes, M.(2005) 'PLM-services – a standard to implement collaborative engineering', *OMG Technical Meeting*, Athens.

Fowler, M. (2003) *UML Distilled. A Brief Guide to the Standard Object Modeling Language*, 3rd ed., Addison-Wesley Professional.

Främling, K., Kärkkäinen, M., Ala-Risku, T. and Holmström, J. (2006) 'Agent-based model for managing composite product information', *Computers in Industry*, Vol. 57, No. 1, pp.72–81.

Garetti, M. and Terzi, S. (2004) 'Product lifecycle management: definition, trends and open issues', *Proceedings at III International Conference On Advances In Production Engineering*, 17–19 June, Warsaw, Poland.

Hirsch, B.E., Thoben, K.D. and Eschenbaecher, J. (2001) Using e-business to provide Extended Products, Automation, Automation days, Helsinki, Finland.

Jansson, K. and Thoben, K-D. (2002) 'The extended products paradigm. An introduction. DIISM2002', *The 5th International Conference on Design of Information Infrastructure Systems for Manufacturing*, Osaka, Japan.

Jun, H.B., Kiritsis, D. and Xirouchakis, P. (2007a) 'Research issues in closed-loop PLM', *Computers in Industry*, (SCI).

Jun, H.B., Shin, J.H., Kim, Y.S., Kiritsis, D. and Xirouchakis, P. (2007b) 'A Framework for RFID applications in Product Lifecycle Management', *International Journal of Computer Integrated Manufacturing: Wireless Manufacturing*, (SCIE).

Kärkkäinen, M., Holmström, J., Främling, K. and Artto, G. (2003a) '*Intelligent products – a step towards a more effective project delivery chain*', Computers in Industry, Vol. 50, pp.141–151.

Kärkkäinen, M., Ala-Risku, T. and Främling, K. (2003b) 'The product centric approach: a solution to supply network information management problems?' *Computers in Industry*, Vol. 52, No. 2, pp.147–159.

Kiritsis, D., Bufardi, A. and Xirouchakis, P. (2003) 'Research isues on product lifecycle management and information tracking using smart embedded systems', *Advanced Engineering Informatics*, Vol. 17, pp.189–202.

Morel, G., Panetto, H., Zaremba, A. and Mayer, G. (2003) 'Manufacturing enterprise control and management system engineering: paradigms and open issues', *IFAC Annual Reviews in Control*, Vol. 27, pp.199–209.

McFarlane, D., Sarma, J., Chirn, G., Wong, J., Ashton, A. (2003) 'Auto-ID systems and intelligent manufacturing control', *Journal of Engineering Applications of Artificial Intelligence*, Vol. 16, pp.365–376.

PLCS (2005) INTERNATIONAL STANDARD ISO 10303-239:2005(E), *Industrial Automation Systems And Integration – Product Data Representation and Exchange* – Part 239, Application protocol: Product lifecycle support.

PLM@XML (2003) SDK V4.0, EDS Plm Solutions. Available online at: www.eds.com

Soylemezoglu, A., Zawodniok, M.J., Cha, K., Hall, D., Birt, J., Saygin, C. and Sarangapani, J. (2006) 'A testbed architecture for Auto-ID technologies', *Assembly Automation*, Vol. 26, No. 2, pp.127–136.

STEP-NC (2003) ISO 14649-Part 1, *Overview and Fundamental Principles*, 1st ed.

Terzi, S., Cassina, J., Chiari, G. and Panetto, H. (2004) 'Traçabilite des produits : une approche holonique', *Modélisation et simulation pour l'analyse et l'optimisation des systèmes industriels MOSIM'04*, 1–3 September, Nantes, France.