

Development of Promise Data Structure

Jacopo Cassina¹, Maurizio Tomasella¹, Marco Taisch¹, Micheal Marquard²,
Altug Metin², Andrea Matta¹

¹ Politecnico di Milano

Piazza Leonardo da Vinci, 32, 20133 Milano, Italy

² InMediasP, Henningsdorf, Germany

Abstract. This work presents the product object model developed inside the eu-funded FP6-IST project PROMISE, which aims at closing the whole set of information loops concerning a product's life. The ultimate goals of the project are: to integrate product data from the entire life cycle via different sources, to support comprehensive analysis on this data and finally to enhance the operational businesses with the obtained insights on products. To achieve these goals, a set of hardware and software tools are being developed. This paper presents the conceptual model behind one of the components of this infrastructure, called the PDKM (Product Data and Knowledge Management) System, which is responsible for the integration and management of both product data and knowledge from all lifecycle phases, on a logically consistent basis.

1 Introduction

Within the globally scaled scenario, the “product” and its related management is becoming unavoidably a key-aspect, creating a “product centric” or “product-driven” problem. This kind of approach is represented by three main layers: PLM (Product Lifecycle Management), Product Extensions and Product Traceability.

PLM in particular has emerged as an enterprise solution. It implies that all software tools/systems/databases, such as CAD, PDM, CRM, etc., used by the various departments and suppliers throughout the product lifecycle have to be integrated such that the information managed by these systems can be shared promptly and correctly between people and applications. Nevertheless, PLM is not primarily an IT problem, but at first, it represents a strategic business orientation of the whole enterprise [Garetti 2004]. From a strategic organization point of view, the adoption of a “product centric” approach means a remodelling of all of the relations existing among the resources (people and equipment) involved into the relevant business processes specifically oriented in a “product” lifecycle direction. From an ICT point of view, this product-centric approach to product and production management, which

- until now - has been executed no more than in “isolated islands” (e.g. PDM and ERP), is being integrated into a larger system, in order to provide a wider and more effective use of product and production information.

The PROMISE (PROduct lifecycle Management and Information tracking using Smart Embedded systems) project’s approach to PLM aims at developing a new-generation Product Information Tracking and Flow Management system. This system will allow all actors that 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, EOL), at any time and any place in the world.

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

PROMISE PLM System is composed of many software and hardware systems and related infrastructures, the main are:

- The PROMISE PDKM (Product Data and Knowledge Management) system, for the management of both product data collected from the field via smart product-embedded devices, and knowledge created and updated from this data, in order to enhance e.g. the design of new products in the future.
- The PROMISE DSS (Decision Support System), which is part of the PDKM system and is devoted to support lifecycle decision making activities, thus providing the analytical basis to the whole project. This is done by defining decision strategies to be applied in the different application scenarios, as well as the related algorithms implementing these strategies.
- A set of PEIDs (Product Embedded Information Devices), i.e. RFID (Radio Frequency IDentification) active and passive tags, sensors and on-board computers, with the related embedded and backend software systems.

In the following, particular attention will be paid to the first of these systems. First, a description of it will be provided, and then the conceptual model behind the development of the same system will be presented.

2 The PROMISE PDKM

The PROMISE PDKM (Product Data and Knowledge Management) 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. The Promise Project and the PDKM are extensively explained in a previous paper of the authors. [Cassina 2006].

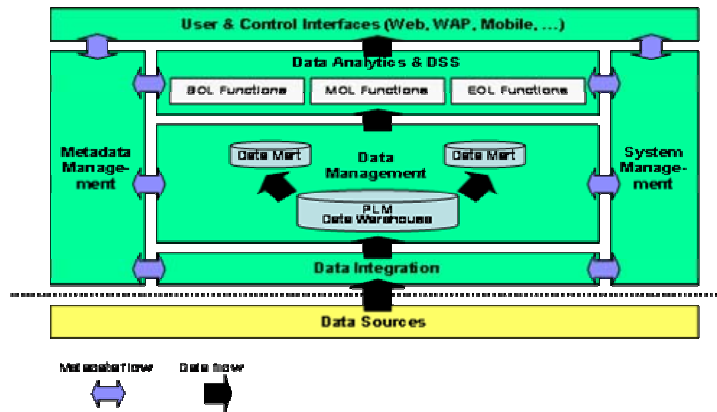


Fig. 1. Architecture overview of the PDKM system

3 Analysis of Enterprise Standards

To develop a flexible and easily compatible data model, an enterprise standards analysis has been done [Cassina 2006]. Many standards exist, each one focused on a specific area of the product lifecycle, but none including all the pieces of information needed to be managed during the whole lifecycle chain, as shown in the next figure, that represent the analyzed standards and their collocation within the product life.

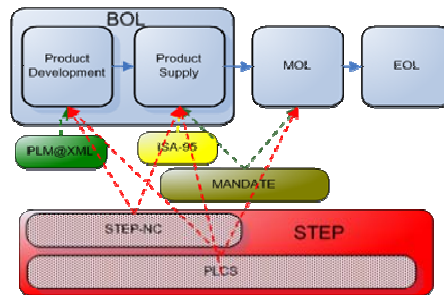


Fig. 2. Standards through Life Cycle Phases

4 Semantic model of the PDKM

The model provides a conceptual view on the PROMISE PDKM System, representing the main concepts belonging to the domain of interest, i.e. to the field of product data modelling throughout the whole product life cycle.

The model is represented in the UML 2.0 modelling language. In particular, since the focus is on the static view on the PROMISE PDKM System, only the UML Class Diagram will be reported in the following.

Figure 3 represents the PDKM Semantic Object Model. The model is conceptually divided into two main areas of interest, though these two areas are well linked together by a proper set of associations.

- A first area, covering the upper portion of the diagram, comprises basic pieces of information such as the serial number of the product instance, the product type to which it belongs, the product structure of the product if needed, the main properties valid for the product instances, the conditions to be checked on them, etc. In addition, this area also describes the product as a product type. This latter however does not represent the main focus of the present model, and for this reason will not be treated deeply in the following.
- A second area models the pieces of information connected to the different life cycle phases in which the PROMISE end-user is interested. This enables the description of the main events out of which a certain life cycle phase is composed (i.e. product failures or breakdowns, replacements of components of a complex product, etc.), of the PROMISE end-user's resources involved in the scenario concerning that life cycle phase (i.e. the garage crew, the designer, the production manager, etc.), and finally the activities performed by these resources in that life cycle phase (e.g. dismantling of a car's components, maintenance of a truck, etc.). Besides this, an important portion of this area is dedicated to the representation of field data, one of the crucial elements in the PROMISE approach. In the following, some major portions of the diagram will be discussed in details, in order to outline its most important and original features.

4.1 Identification of product items

The PROMISE approach to PLM is a "product instance-centric" one. Each instance of a certain product type should be followed all along its life cycle in order to close the desired information loops, thereby creating value. The concept of PEID (Product Embedded Information Device) is capable of enabling the link to all these product items and their related information. A central portion of the semantic model should thus reflect this approach and properly represent the information on each product at the item level. The classes involved in these traceability issues are in particular the ID_INFO, the INFORMATION_PROVIDER and the URI classes; these classes together enable the identification of product instances and the retrieval of the information on where it is possible to find other information on the same product instances.

Many traceability systems have been developed up to now (e.g. the Dialog System developed by the Helsinki University of Technology, the WWAI-World Wide Article Information concept, the AUTO-ID proposal, etc.), and the set of classes cited above should be compatible with all of them, at least from a conceptual viewpoint. There are two types of links to additional information, URI and INFORMATION_PROVIDER objects. The central class here is the ID_INFO class, where one can find the identifier of the product item (ID attribute), the coding schema used (ID_Type attribute) and eventually other formats in which the id can be presented (Alt_Pres attribute) for some reason of clarity, use, etc. The URI class 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 identifies e.g. the IP address on the web where the information can be retrieved, while the INFORMATION_PROVIDER class contains information that can be used to control the request for information from a traceability system. This includes e.g. the inter-enterprise communication systems which, as usual in traceability systems, take care of identifying the information providers (also possible with the URI information sources).

4.2 Description of product structures

In order for the model to be capable of modelling both “atomic” products (i.e. “one-piece” products) and complex products, some classes should be devoted to representing different kinds of product structure, following the specific needs of each application case. A first example of these classes is given by the PHYSICAL_PRODUCT class, which states the product type, the lot to which it belongs, the “birth date” of the product, the “end date” in case the product has reached the end of its life, and finally the product structure “as produced”, with the specification of the identifier of all the instances of the components/subassemblies belonging to this structure. Another important class is the AS_DESIGNED_PRODUCT class, which describes on the contrary the product “as designed” structure, with all the needed information, such as CAD data, BoMs (Bill of Materials), cost information, and all the other pieces of information which are typically stored and managed by PDM (Product Data Management) systems.

4.3 Properties and Conditions

Another important feature of the proposed model is the capability of modelling properties which must be valid for some specific product type and/or product item. This is made possible by the PROPERTY class, which can also be used to describe the properties related to some important resources involved in the PLM application case of interest. This class was originally inspired by ISA-95.

The CONDITION class aims at expressing some either atomic or complex kind of condition which must be checked in some product life cycle scenario. E.g. 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 up. 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 and concerning a specific product instance (in this case, the kind of field data 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).

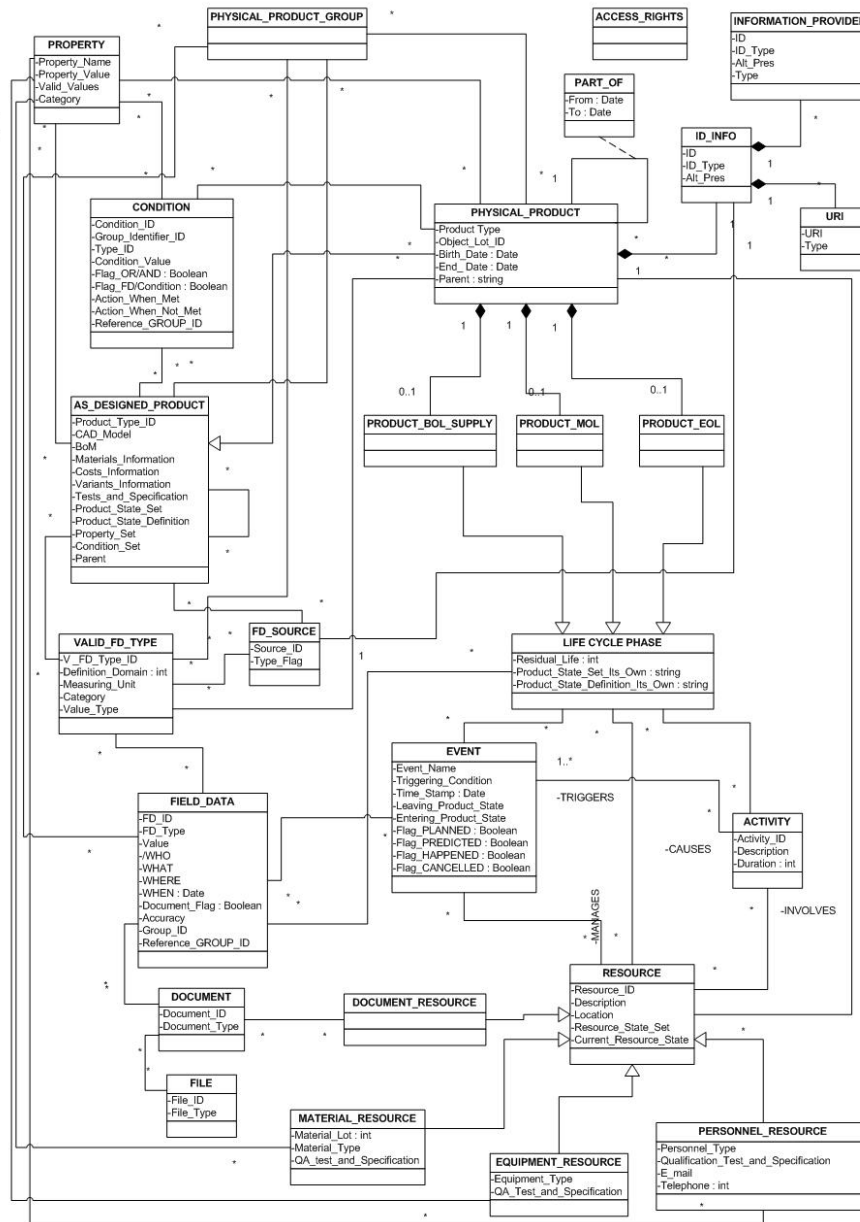


Fig. 3. PROMISE PDKM semantic object model

4.4 Life Cycle Phases

A PLM data modelling framework such as the proposed model must be also capable of modelling the whole set of lifecycle information on the product

considered in each application case. Different applications are related to a different set of life cycle phases. 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 names of these classes reflect the viewpoint of PROMISE on the product life cycle. The first class refers to the pieces of information related to the BOL (Beginning Of Life) phase of a product instance, from the production phase to the final delivery of the product to the customer (thus only the information concerning the design of the product is excluded from this class). The second class refers to the pieces of information related to the MOL (Middle Of Life) of a product instance, i.e. the usage phase and the maintenance phase. Finally, the third refers to the pieces of information related to the whole set of possible EOL (End Of Life) phases of a product instance (e.g. the remanufacturing phase, the recycling phase, etc.). All the pieces of information which are common to these three phases are provided by the class named LIFE_CYCLE_PHASE, which e.g. describes some important issues such as the residual life of a product component, or the set of states in which a product instance can be.

4.5 Event, Resources and Activities

Up to this point, none of the mentioned modelling elements has been intended to describe each single life cycle phase in the way the same phase is intended to be managed, i.e. none of the shown modelling elements represented the main events happening during a certain life cycle phase, the people and other kinds of resources of the company which are involved in the life cycle phase, and the activities performed during this phase. These issues are addressed by the EVENT, RESOURCE, and ACTIVITY classes. These classes of the proposed model were inspired by the production simulations, that usually uses the same concepts. Moreover there are some similar classes within STEP-PLCS and resources are structured using the same approach of ISA-95.

For instance, in a typical predictive maintenance scenario, such as those in the PROMISE project, one would like to model the event of breakdown of a component/subassembly or even of the entire product as a whole, as well as the maintenance activities and the resources, human and not human, involved in such activities. The aim could be for example to predict the point in time when the product will probably break down, to plan the related maintenance activities, and finally to record the actual time instant when the breakdown really occurs, or even to delete the breakdown event from the "list of predicted events", just because the component causing the possibility of product breakdown has been replaced by a new one, which eliminates the main cause of breakdown. The repair activities at the garage of the maintenance provider should also be first described and then properly managed in accordance with the corporate strategy of the company and in a PLM vision. So the availability of resources like free hours of the garage crew to be possibly allocated, or even the availability of the needed materials and equipments to perform the maintenance activities should be checked, and eventually, such as in some PROMISE application scenarios, the maintenance activities should be economically planned and managed.

For these purposes, the RESOURCE, ACTIVITY and EVENT classes with a certain set of attributes have been added to the model. In addition, three associations, one for each pair of these classes, have been added to state that an event triggers an activity, which involves some resources, which in turn manage the event as an important part of a specific life cycle phase. 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 duration in terms of time. An activity has at least two events associated with it: the event "ACTIVITY STARTS" and the event "ACTIVITY ENDS". The event is triggered by some kind of condition and causes in general the shift of the product state from some "STATE A" to another "STATE B". Again, one should have the possibility, as written above, to 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). In addition, an activity can cause an event, such as the maintenance activity can cause the "REPLACEMENT OF COMPONENT XYZ PERFORMED", with a consequent update of the product's residual life that, if no more under the minimum threshold, causes the "PRODUCT BREAKDOWN" event to be cancelled, such as in the example above. Finally, the resources can be human beings (PERSONNEL_RESOURCE class), equipments (EQUIPMENT_RESOURCE class), materials (MATERIAL_RESOURCE class) and documents (DOCUMENT_RESOURCE class). Some of the information related to these resources is given as attributes, and some other kind of information is specified as objects of the PROPERTY class. Some important examples 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, the maintenance manual or the CAD model of the product layout as objects of the DOCUMENT_RESOURCE class.

Moreover, for each resource a set of possible states is defined, and the current state is recorded. This is required for example in cases where the information on the availability of the garage all along a certain time period can be very important to plan the maintenance activities at the garage, such as in the PROMISE application scenario concerning the predictive maintenance performed over an entire fleet of trucks. Thus, two states such as AVAILABLE and NOT AVAILABLE can be defined as exhaustive and mutually exclusive states, and the setting of the product state of the garage to one or the other of these values can be used to understand if a given time interval can be assigned to the maintenance of a specific product item or not.

In addition, 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. Such a scenario is up to now not so realistic, but anyway it is interesting to notice that the boundary between an object as

a resource and an object as a product itself, as implied by the PROMISE vision, is not so well defined and even not so thick as one would think.

4.6 Field Data

Another central class of the diagram is the FIELD_DATA class, which enables the overall PROMISE approach to PLM by collecting data from the field, thus also enabling the improvement of product performance and in general the creation of economic value from PLM activities.

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

In the following, a brief overview of the most important attributes of the FIELD_DATA class is reported. The FD_ID attribute univocally identifies each field data record, while the FD_Type attribute states the type of field data (e.g. that it is a temperature of a certain sensor). The Document_Flag attribute says if the field data as 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 the location where the measurement was carried out (if needed). The WHEN attribute then represents the timestamp 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 is 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 Conclusions

This work led to the development of a semantic model for a PDKM, which will be able to interoperate within different systems and store all kinds of product life cycle data. This model was also tested using different application scenarios within the PROMISE consortium.

At the present time the development of the technical model starting from the semantic model is ongoing and a first prototype has been developed in collaboration with InMediaSp and SAP, partners of the PROMISE Project. The model will be also improved with the results from the use of this prototype.

Finally, the semantic model will also be used for standardization efforts; at first it will be proposed to other standardization institutions, like for example the PLCS community, to be merged inside these standards.

Acknowledgement

This work has been partly funded by the European Commission through the FP6-IST Project entitled PROMISE: PROduct lifecycle Management and Information tracking using Smart Embedded systems (No. IST-2004-507100). The authors wish to acknowledge the Commission for their support. We also wish to acknowledge our gratitude and appreciation to all the PROMISE project partners for their contribution during the development of various ideas and concepts presented in this paper.

References

1. Cassina J., Taisch M., Terzi S.; "Towards an Intelligent Extended Product"; IFIP APMS 2005.
2. Cassina J., Tomasella M., Marquard M., Metin A., Matta A., Taisch M.; "Development of the semantic object model for a PDKM System"; ICE 2006: pg.383-390.
3. Garetti M. Terzi S., Product Lifecycle Management: definition, trends and open issues, Proceedings at III International Conference On Advances In Production Engineering, 17 - 19 June 2004, Warsaw, Poland.
4. ISO9000, International Standard Organisation (ISO), 2000, International Standard ISO 9000:2000 Quality Management Systems - Fundamentals and vocabulary.
5. Jansen-Vullers J., A. van Dorp, B. Beulens, 2003, Managing traceability information in manufacture, 2003, International Journal Of Information Management 23 : 395-413.
6. Kärkkäinen J., G. Holmström, J. Främling, G. Artto, 2003, Intelligent products – A step towards a more effective project delivery chain, Computers in Industry 50 : 141-151.
7. Morel G., H. Panetto, A. Zaremba, G. Mayer, 2004, Manufacturing enterprise control and management system engineering rationales and open issues, IFAC Annual Reviews in Control, 27 : 199-209.
8. McFarlane D., J. Sarma, G. Chirn, J. Wong, A. Ashton, 2003, Auto-ID systems and intelligent manufacturing control, Journal of Engineering Applications of Artificial Intelligence, 16 : 365 – 376.
9. Promise Project Deliverable 9.1 and 9.2 www.promise.no
10. Van Moll J.H., 2002, The importance of Life Cycle Modelling to the development and testing of complex products, TestNet Najaarsevenement, Nieuwegein.