DI1.6: Evaluation of PROMISE Standards

Ajith Parlikad, Cambridge
David Potter, INDYON
Dimitris Kiritsis, EPFL
Jacopo Cassina, POLIMI
Kary Framling, HUT
Maurizio Tomsella, POLIMI

DELIVERABLE NO	DI1.6: Evaluation of PROMISE standards
DISSEMINATION LEVEL	**PUBLIC**
DATE	02. December 2006
WORK PACKAGE NO	WP I1: Standardisation
VERSION NO.	0.3
ELECTRONIC FILE CODE	promise di1.6 evaluation of promise standards.doc
CONTRACT NO	507100 PROMISE A Project of the 6th Framework Programme Information Society Technologies (IST)
ABSTRACT	This report is the first of a three-part series of deliverables aimed to document the activities on evaluation and refinement of PROMISE standards and architecture specifications. In this report, we provide an evaluation of the PROMISE Product Data Knowledge Management (PDKM) object model against the requirements of Product Lifecycle Management (PLM) and compare it with Product Lifecycle Support (PLCS) and PLM Services. The report also evaluates the suitability of PMI for standardisation, and describes the action plan for further evaluation and refinement of PROMISE standards over the next 12 month period.

STATUS OF DELIVERABLE		
ACTION	**BY**	**DATE (dd.mm.yyyy)**
SUBMITTED (author(s))	Ajith Parlikad	13.12.2006
VU (WP Leader)	Ajith Parlikad	13.12.2006
APPROVED (QIM)	Dimitris Kiritsis	14.12.2006

Revision History

Date (dd.mm.yyyy)	Version	Author	Comments
06.02.2007	0.1	A. Parlikad	Draft outline
10.05.2007	0.2	A. Parlikad	Submitted for quality review
14.05.2007	0.3	A. Parlikad	Final version

Author(s)' contact information

Name	Organisation	E-mail	Tel	Fax
Ajith Parlikad	Cambridge University	aknp2@cam.ac.uk	441223765606	
Jacopo Cassini	Polimi	jacopo.cassina@polimi.it	390223993951	390223992700
David Potter	Indyon	david.potter@indyon.de	442392345152	442392592327
Kary Framling	HUT			
Maurizio Tomasella	POLIMI			

Table of Contents

List of figures

Abbreviations

PEID: Product Embedded Information Device
PMI: PROMISE Middleware Interface
PDKM: Product Data Knowledge Management
PLCS: Product Life Cycle Support
UML: Unified Modelling Language
XML: eXtensible Markup Language
PDM: Product Data Management
DBMS: Data Base Management System
PLM: Product Lifecycle Management
DSS: Decision Support System
OMG: Object Management Group
MDA: Model Driven Architecture
ERP: Enterprise Resource Planning

1 Introduction

The overall goal for PROMISE work package I1 is to ensure that there is an open channel for the continuing promotion and acceptance of key elements of the standards (or set of standards) and interfaces defined during the PROMISE project.

This report is the first of a three-part series of deliverables aimed to document the activities on evaluation and refinement of PROMISE standards and architecture specifications. This is documented in the description of work as TI1.8. This deliverable evaluates the key PROMISE components where standardisation efforts will be focussed in the next 12 month period.

The EU Commission reviewers at the review meeting in January 2007 made the following recommendation regarding standardisation:

*"**Recommendation 8: Developing an action plan for standards' promotion***
Next DI1.7 (Promotion of PROMISE standards), scheduled at month 30, should include a detailed action plan for standards' promotion, indicating which organisations, bodies or working groups will be contacted, how, by whom, when and for which scope. The consortium shall in particular consider PLM Services (promoted by the Object Management Group, OMG) and a potential collaboration with the Open Group. The middleware being the core element of the PROMISE project could be proposed as a full standard by itself, not limited to additions to existing standards. Web Services that are used by the middleware could have their interface, described in WSDL, described by this standard."

In accordance to the above recommendation, this deliverable presents the following:

- Evaluation of the PDKM object model against requirements of PLM
- Comparison of the PDKM object model with PLCS and PLM Services
- Examination of suitability of Middleware for standardisation

The remaining part of the recommendation is addressed in DI1.7 [2].

2 Evaluation of PDKM object model

In this section we evaluate the PDKM object model for suitability in terms of specifying product data throughout its lifecycle.

The PROMISE PDKM object model has been developed in the context of WPR9, task TR9.2. It is the conceptual semantic data model behind the PROMISE PDKM system, and in particular it describes the core of the Data Management layer of this system, whose main task is to provide a global semantic view on product and product life cycle data for all analysis applications. Please refer to DR9.1 [3] for a thorough description of the different "layers and towers" that constitute the PROMISE PDKM system architecture.

Two different kinds of evaluation will be presented here, considering data representation issues at the semantic as well as at the implementation level. First, we examine the semantic issues.

2.1 Semantic issues

Figure 1 shows the UML class diagram representing the PDKM object model, as presented in DR9.2 [4], where the semantic model was documented for the first time. The semantic data model consists of entities (classes and objects of these classes) that business users are familiar with and can easily communicate about. These entities are further described by attributes and relationships to other entities and to the business terms used in the application contexts. This semantic data model represents a powerful means for users to navigate in the information space and to quickly find the desired piece of information. More importantly, such a semantic model offers a common basis for data communication and exchange between applications, between users, as well as between users and applications.

At a purely semantic evaluation level, the PDKM object model is able to meet all the major requirements of the closed-loop approach to PLM (Product Lifecycle Management). In particular:

- The model is focused on the representation of information concerning product items, as explicitly required by closed-loop PLM, rather than only information of product types, as in the most typical current PLM systems. The model is able, in particular, to represent information on:
    - the identification and tracking of each physical product entity (and/or assembly/subassembly/component, depending on the level of detail needed by the user);
    - the usage/operation of the product in the field, by the appropriate representation of the field data to be managed in each application.
- The model is able to manage product structures, both at the product-type level, i.e. at the level of the product "as-designed", such as in the most typical PLM/PDM systems [5], and at the physical product level, i.e. at the level of the product "as-produced" and "as-used". The level of detail is chosen by the user and not pre-imposed by the semantic model.
- The model enables the creation of appropriate knowledge, starting from the data collected from the field, in order to better support both decision making activities on currently existing product units, e.g. predictive maintenance decisions, and the development of better new generations of products.
- The model supports the semantic description of the different aspects concerning all the product life cycle phases. This comprises information on the major life cycle 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. Again, the level of detail can be chosen by the user and not be pre-decided by the developers of the particular product data standard employed.

Some existing standards for product data representation are to some extent able to meet all or part of this set of requirements (refer to the next section for two of such standards). However, the PROMISE PDKM object model is able to solve two of the major problems concerning these standards, i.e the *completeness* of the model, in terms of life cycle aspects considered, and the *ease of use and customization*, depending of the specific needs of the implementation to be carried out. Moreover, some of the most interesting modelling criteria which are present in existing standards were taken into account in the modelling phase of the PROMISE PDKM object model. In addition, since the portions of the semantic model regarding products "as-designed" and traceability issues are very general, the same model is in principle compliant with the existing PLM and traceability systems.

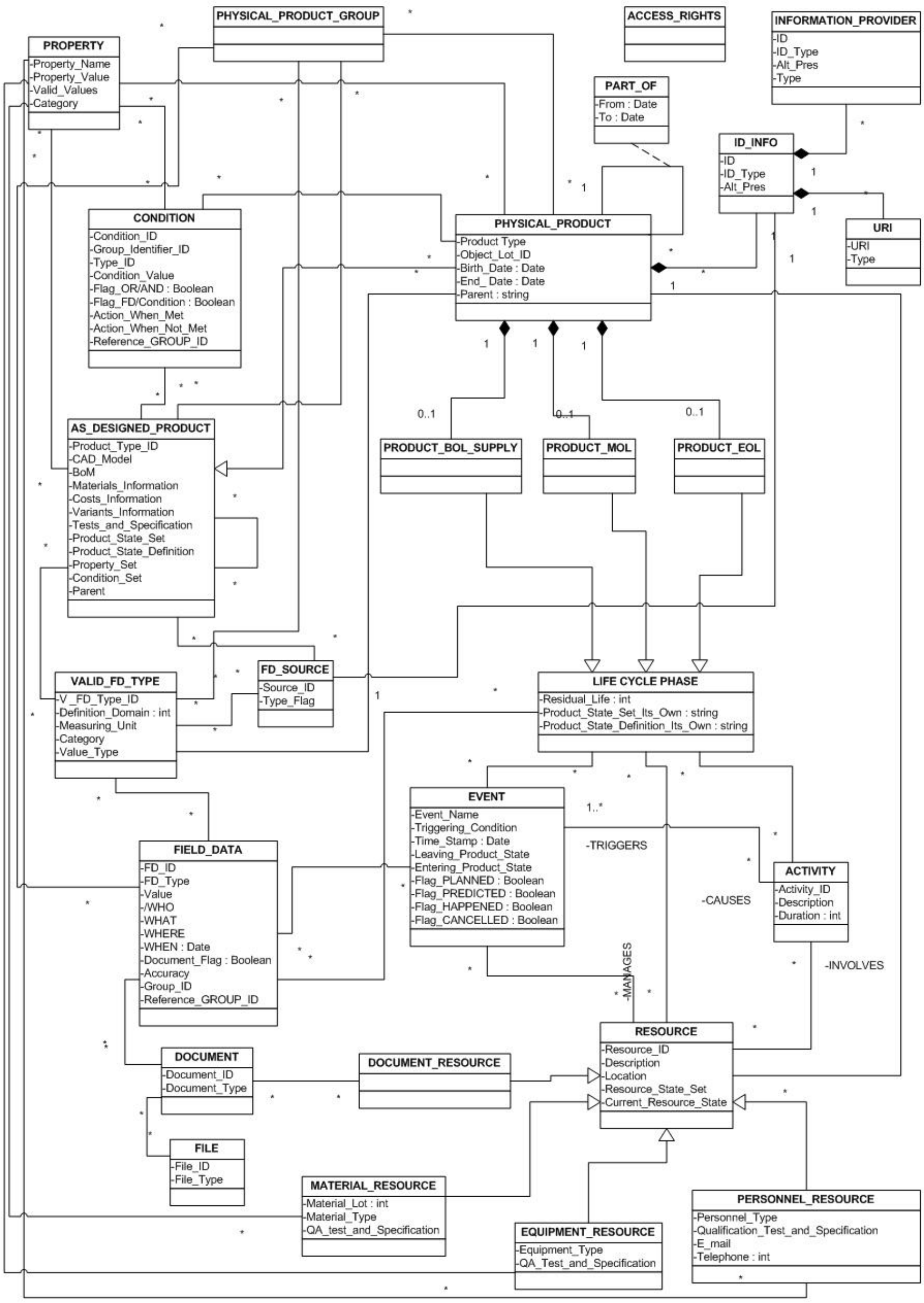

Figure 1: PROMISE PDKM object model

Finally, the semantic model was developed based on the product lifecycle requirements of the ten (formerly eleven) PROMISE application scenarios, but also taking into account the requirements coming from potential future PROMISE products. The natural-target products for the PDKM object model are products of low-to medium complexity, where the representation and management of field data plays a crucial role in the management of the whole product lifecycle, but can also be extended to other domains.

However, a final and exhaustive evaluation of the semantic capabilities of the PDKM object model can only be carried out after the semantic model has been fully implemented in real-world closed-loop PLM systems.

The semantic model is currently undergoing the first steps of the standardization process, taking as a starting point the UML class diagram. A review of the elements of the model is expected to take place in the following months, arising from the implementation and first use of the ten PROMISE demonstrators. The feeling of the developers of the semantic model is however that no big modification of the model will take place, since a first informal assessment of the semantic potential of the model, carried out in the first months after its first release (M18), confirmed its compliance with the requirements of the different demonstrators.

However, for the semantic model to become a proposal for a standard, the current version of the model is not sufficient. The current version only addresses conceptual issues, whereas the version to be proposed as a standard should also provide the specification of the interface of the different modelling elements. This would represent a major improvement of the model and at the same time would specify more clearly what a data model must have in order to be compliant with the proposed standard, and would also serve as a first guide for specific implementations of the model. This improvement of the semantic model is taking place in the context of WPI1, and is expected to result in the final specification of the standard proposal. Moreover, documentation of this new version of the model will become part of the PROMISE Architecture Series.

2.2 Implementation issues

The technical management of data in the PROMISE PDKM system, and in particular in the Data Management Layer, requires a data schema that can be implemented in a DBMS (Data Base Management System) and/or a file system. The major requirement is that the elements of the semantic data model, i.e. classes, attributes and relationships, should be consistently mapped to the database schema, so that a drill down from semantic information objects to the real instance data stored in databases or files is possible.

The Data Management layer consists in particular of a central database, the data warehouse, which allows constructing specific views as data subsets of the data warehouse, the so-called data marts. The data warehouse solution offers a significant advantage (see the discussion at pages 14 and 15 of DR9.1 [3]) by keeping a manageable number of connections between the different data sources involved in each application/implementation case and the data marts. In particular, a data mart does not have to integrate multiple operational data sources but only be defined on or extracted from the data warehouse.

Moving to considerations on how the Data Management layer is implemented in the context of the PROMISE project, it was not intended that the solution for demonstration purposes should be built from scratch, but on the opposite it was decided to base the PDKM system implementation on an already existing PDM (Product data Management) system, and for this purpose the mySAP PLM suite was chosen. A proper mapping from the semantic data model onto the specific SAP technical data schema was then necessary.

This mapping was first documented in Section 5 of PROMISE deliverable DR9.2 [4], and has become fundamental for the PROMISE team responsible of developing the PROMISE DSS (Decision Support System), due to the need of this team to constantly map the specific requirements of the different decision support algorithms (the elements of the semantic model involved in each decision support use case) to the same consolidated base (the SAP technical data schema).

Besides these mapping issues concerning the semantic data model and the SAP technical data schema, some other technical issues (which represent by no means a complete list) related in general to the implementation of the semantic data model are reported below, as a first reference for new implementations of the PROMISE PDKM system.

Each association represented in the semantic model has to be implemented by referencing from one object the other via that object's unique key. When such a key is not specified in the semantic model, a technical ID has to be introduced. Uniqueness in this context is meant with respect to the object-type: the tuple *(object-type, key)* uniquely identifies the object in the technical data model. Such a technical ID might also be introduced if it turns out that in the specific case this represents the easiest solution with respect to managing e.g. a key composed of several attributes.

There have to be technical entities that ensure the uniqueness of newly generated object-keys. It might also be necessary/convenient to introduce own tables; e.g. for n:m relations, it might be necessary/convenient to implement new tables to store the tuples *[ID of object 1, ID of object 2]*.

What is treated in the semantic model as an object-attribute-relation might be mapped in the technical model to an object-object-relation; e.g. an object of the AS_DESIGNED_PRODUCT class has as an attribute named Drawing, which from a technical point of view could in many cases be conveniently dealt with a set of objects, maybe of the DRAWING class, that are referenced from the AS_DESIGNED_PRODUCT object.

There might be added objects for e.g. storing all allowed values for the status of a specific product/product type. The introduction of these objects might can be intended to avoid duplicating contents (think to the state definition in the LIFE_CYCLE_PHASE objects).

Concluding, as it often happens when passing from a conceptual model to its implementation, it could turn out that a *strict* 1:1 mapping from the semantic object model to the technical data schema is even not advisable, due to performance issues. This is not strange in the sense that a conceptual data model generally implies a little or even nothing concerning its implementation.

As a last point, it must be highlighted here that some of the PROMISE partners raised the need of developing an implementation of the PROMISE PDKM semantic model using the MySQL open-source DBMS. This was felt to be a good opportunity for exploiting one of the many project results (the semantic data model) in a more easily reusable implementation solution, covering the needs of both academic/research and industrial partners. The work in this direction is expected to start in the following project months.

3 Comparison of PDKM object model with existing standards

In this section, we compare and contrast between the PDKM object model with PLCS as well as PLM Services (as per reviewers' recommendation 8).

3.1 Comparison of PDKM object model with PLCS

We compare the PROMISE data management "standard" with the PLCS against the following points: (a) aim of the standard, (b) vision of PLM, (c) scope, (d) enabled business functionalities, (e) applicable sectors, and (f) complexity of the standard.

3.1.1 Aim of the standard

PLCS ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving. As represented within its logo (Figure 2), PLCS has four main aims:

Figure 2: PLCS aims

- **Support Engineering**. The design of the in-service support system, providing and sustaining the support infrastructure
- **Resource Management**. Covers buying, storing, packing, movement, issue and disposal of products and support elements
- **Configuration Management**. Foundation of the lifecycle support. Includes managing changes of a configured item including tracking of serial-numbered items
- **Maintenance & Feedback**. Maintain, test, diagnose, calibrate, repair and modify products and support elements including the roles of schedules, resources and feedback

The PDKM (Product Data and Knowledge Management) "standard" is a proposal for a way to store and manage lifecycle data for products with a complexity varying from households to locomotives.

The system aims at integrating and managing data from all the lifecycle phases of products and 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 model was developed to provide the system with a basis for representing product data throughout the whole product life cycle, as in the PROMISE viewpoint, thus finally enabling a real closed-loop approach to PLM.

To enable this approach, product lifecycle data management must go well beyond its commonly known frontiers. In particular:

- The focus must be shifted from information on product types to information on product items, virtually each product item. This is 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 decision support systems commonly adopted in each scenario to support decision makers in the value creation process.
- Product items at the different levels of the product structure must be identified, and the related information must be properly collected and managed. This must be 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 must be possible to manage information on product structures related to both products "as-designed" (those typically managed by currently available PLM systems) and to physical products. This last type of information must carry 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 must thus be properly tackled. The PROMISE PDKM System is in this sense compliant with the most widely adopted approaches to product identification and life cycle traceability.
- All of the data which must be collected from the field on physical products, assemblies/subassemblies and components, must also be 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, if necessary, must be available.
- This way of enabling and exploiting the seamless flow, tracing and updating of information on product items, after their delivery to the customers, up to their final destiny (e.g. remanufacturing, recycling, disposal), and then back again to the designer and manufacturer, must also allow to derive useful knowledge from the collected data. The management and updating of this knowledge must also be supported.
- Finally, a semantic description of the different aspects of product life cycle phases in which each PLM scenario is interested must be available. This should comprise information on the major life cycle 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.

3.1.2 PLM Vision

The vision of the PLM behind PLCS (see Figure 3) is that the standard has to be able to follow the life cycle of each single product item, storing all the data in all the life cycle phases, and is expressed in the following figure.

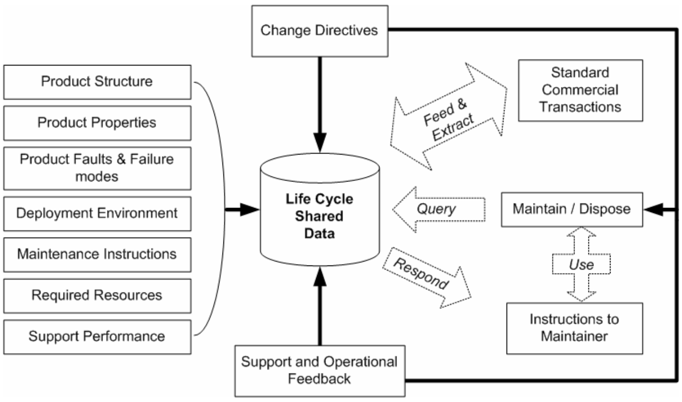

Figure 3: The PLCS vision for PLM

The vision shows how relevant information required for life cycle support can be managed in an integrated form. This life cycle data has two components:

- the APSI (Assured Product and Support Information), which defines the product and its support, and is subject to configuration change management;
- and related information, including feedback on the history of the product, activities and resources. Related information may be updated or corrected with time, but is not subject to configuration change management.

On the other hand, the PLM vision that characterized the PDKM is the same shared with the whole PROMISE project, where the focus is on the single item of the product. The standard so has to follow the life not of a product type, but of each single physical product item, from the design, where the “product type” PLM is considered, to the production, the usage and finally the end of life with the recycling and dismantling.

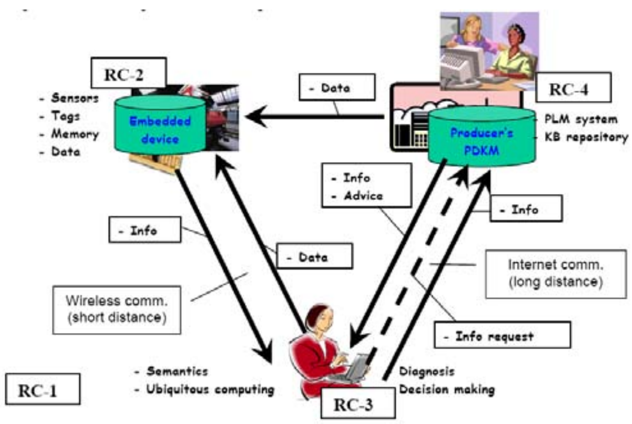

Figure 4: PROMISE PLM vision

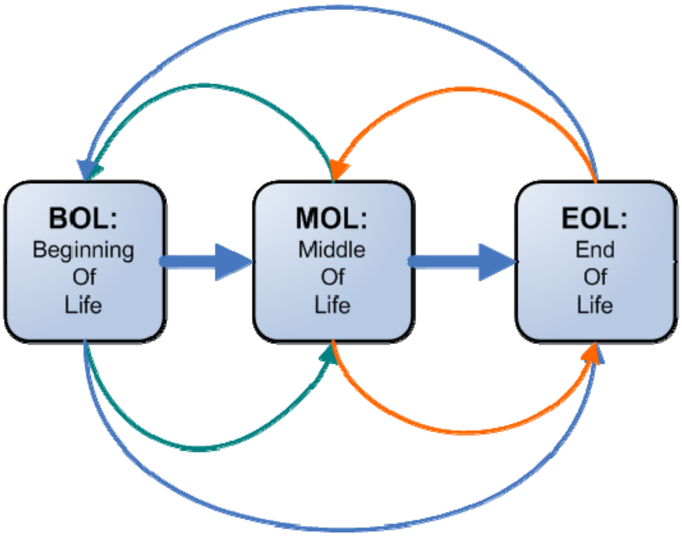

Figure 5: The product lifecycle for PROMISE PDKM

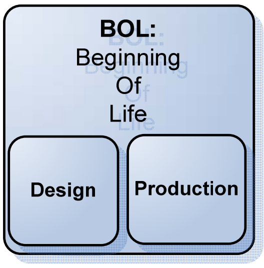

Figure 6: The Beginning-of-Life phase

The first part of the BOL (refer to Figure 6), the design phase, can be decomposed into the design and the production. The product type PLM focuses only on this phase, detailing all what happened to the type, during all the lifecycle of the "project".

The PDKM does not proposes itself as a new standard for the BOL phase, but has an open class to link other already established standards like PLM@XML and PLM Services.

3.1.3 Scope of the standard

The following are within the scope of this part of ISO 10303:

- information for defining a complex product and its support solution;
- information required to maintain a complex product;
- information required for through life configuration change management of a product and its support solution;
- the representation of product assemblies
- the representation of a product through life
- the specification and planning of activities for a product.
- the representation of the activity history of a product.

- the representation of the product history.

The following are within the scope of the PDKM part of the standard:

- information for defining a complex product;
- information required to maintain a complex product (mainly a maintenance manual);
- the representation of product assemblies
- the representation of a product through life
- the management of field data
- the representation of the events that occurred to the product
- the management of the knowledge derived from field data
- the representation of working limits and conditions of the product
- the representation of the product owner
- the creation of a link from the physical product to the centralized database
- the specification and planning of activities for a product.
- the representation of the activity history of a product.
- the representation of the product history.

3.1.4 Enabled business functionalities

The following functionalities are enabled by PLCS:

- **Activity Management** - Request, define, justify, approve, schedule and capture feedback on activities (work) and related resources
- **Product Definition** - Define product requirements and their configuration, including relationships between parts and assemblies in multiple product structures (as-designed, as-built, as-maintained)
- **Operational Feedback** - Describe and capture feedback on product properties, operating states, behavior and usage
- **Support Solution and Environment** - Define and maintain the necessary support solution for a product in a specified environment including the opportunity to provide support (scheduled downtime), tasks, facilities, special tools and equipment, and personnel knowledge and skills required. PLCS will also relate organizations, personnel and facilities with the product needing support.

In comparison, the following functionalities are enabled by the PDKM:

- Product definition
- Field data management
- Management of the data about the events that occurred to the product
- Product knowledge management
- Centralized product fleet management
- Centralized data analysis
- Simple data analysis (based on conditions within the model itself)
- Availability of data for DSS

3.1.5 Applicable industry sectors

ISO 10303 was developed to represent the data associated with major engineering assets, such as aircraft, ships or weapon systems, which may exist as a product design or as realized products. It can also be applied to much simpler products such a pump or an amplifier. The industry focus of this part of ISO 10303 is shown by the shaded area in Figure 7 which is characterized by complex products, with long lives and demanding support requirements.

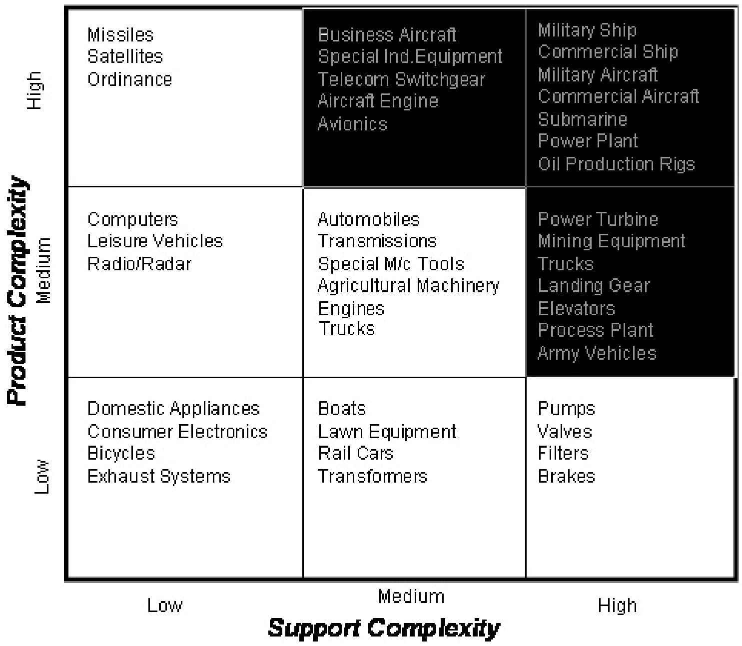

Figure 7: Applicable sectors for PLCS

The benefits from using this part of ISO 10303 to organise product support information rise as the product or the support arrangements are subject to more frequent change.

The PDKM data model was designed at the very beginning as an internal research of POLIMI about the needs for a standard for mass goods, then has been improved and deeply reworked during the PROMISE project allowing a real implementation within the MySAP-PLM suite, being able to manage the data from the ten applications of the promise projects, which are, for example cars, trucks, diggers, refrigerators, boilers, plastic materials and milling machines. Summing up, the PDKM is focused on the shaded area of Figure 8.

3.1.6 Complexity of the standard

PLCS is a very complex standard. In fact it provides not only a data model to maintain and manage the PLM data information, but a mechanism to support complex products and systems during all their life.

This goes far beyond standardizing the way of administrating data, but standardizes also the processes and the activities that are carried out during the lifecycle. So to implement this standard a reorganization of the activities carried out on the product is also desired.

A good hint of the complexity can be seen in the existence of more then 200 data classes and more then 350 activity diagrams. All these reasons suggest that the effort for the implementation of PLCS is high, and the process requires time and efforts.

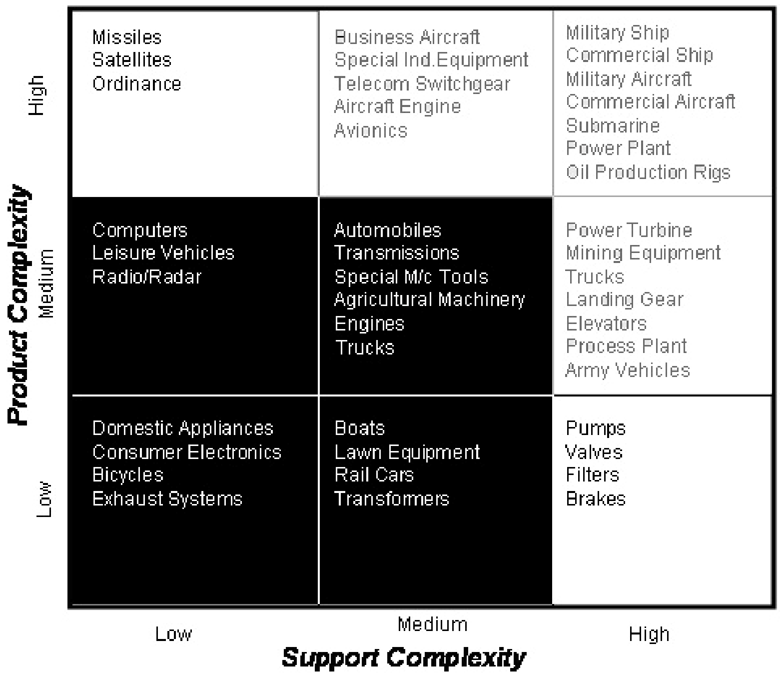

Figure 8: Application sectors for PROMISE PDKM

In comparison, the PDKM object model is relatively simple. The basis of its simplicity is its fractal and recursive structure, that allows it to cover all the complexity needed by the different kind products with only 26 classes, as shown in Figure 1, that represent the whole data structure.

The PDKM does not aim to standardize nor propose business procedures and activities, but focuses only on data managing and technical issues, being enablers of business and management procedures suggested and developed within the project, but also of others that have not being considered till now.

3.1.7 Summary

The two analyzed standards cover a similar topic, that is the single item lifecycle, but from very different prospective and needs. PLCS aims at very complex products, and also standardizes all the procedures and the activities that follow it. The PDKM instead focuses on simpler 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 PDKM to depict complex products such as military aircrafts lifecycle. However, outside their sector of applicability the standards will not perform at best, and their usage will not be sound from a business point of view.

3.2 Comparison of PDKM object model and PLM Services

In this section, we compare the PROMISE data management “standard” with the PLM Services standard developed by the Object Management Group (OMG). PLM Services and PDKM, as will be explained in this section, are not in competition and there is no overlap between them. For this reason the overview of the two standards has been shortened and only the main details have been provided here.

3.2.1 Aim of the standard

The current standard PLM Services 1.0 is a ProSTEP iViP working result and is standardized by the Object Management Group (OMG). It is the first standard comprising current XML and Web Services technologies with a STEP data model – thus providing both syntax and semantics. OMG PLM Services 1.0 is ready to use and has proven strong functional capabilities which support cross-company, cross-domain, cross-system and cross-technology collaboration. As a comprehensive framework based on the OMG Model Driven Architecture (MDA) OMG PLM Services provide a solid foundation for collaborative engineering scenarios like browsing in distributed product data structures, design in context, product data visualization, and others. Unique about OMG PLM Services 1.0 is that it supports industry-related use cases by defining both the informational and operational process-related aspects.

PLM Services aims to be a Standard for Collaborative Engineering considering PDTnet und PDM Enablers within the Automotive Industry. The standard is suitable to implement cross-company scenarios with shared access to distributed PLM data sources. The major focus of the standard is to provide a common, easily available access mechanism to PLM information and to support collaborative engineering work in a heterogeneous system world. Standardized interfaces to the majority of the required PDM functionality allow short set-up times for establishing working partnerships. Organizational data are within the scope of the OMG PLM Services specification. So, all PLM information may be extended by creator and owner information as well as alias names suitable for cross-company referencing of parts, documents and others.

The scenario is realized by providing a definition for a common neutral data model and access methods based on common web technology, easily accessible for a large number of automotive companies. Using a web-based transport mechanism delegates aspects of authentication and authorization as well as encryption and other security requirements to already existing technologies and infrastructures. OMG PLM Services just adds a mechanism to access valid data by session-related identifiers. This allows efficient interactive scenarios.

The standard is suitable to implement cross-domain scenarios interconnecting PLM, CAD, ERP and other planning system data sources. The structural information provided by product or document information given in the PLM system provides a comprehensive access methodology to complete product definition data including support measurements for manufacturing, process data, logistics, change management and others.

The standard is suitable to implement aggregated simultaneous views on multiple PLM systems. Sharing a common data model as defined by the specification and merging such a common view is an obvious application. But OMG PLM Services provides not only a shared and system-independent view on PLM data but defines the semantics of access functions and the necessary answers of the PLM systems to form valid PLM data. The aggregated view on multiple PLM systems is a one system image on distributed data. Users are no longer forced to use multiple interfaces to assess data from different sources. OMG PLM Services furthermore allows easy exchange of viewers to meet different requirements of different user groups. The need for erroneous data replication is minimized.

The standard is suitable to implement integration scenarios with existing STEP file-based infrastructures. Due to the normative mappings between AP214 AIM – the data model exchanged in STEP files – and the XML representation – exchanged by the Web Services as defined in the

OMG PLM Services specification – a complete round-trip and unambiguous data transfer scenario becomes feasible. All the necessary transformation steps are defined in the standard. Therefore, a seamless transformation between an XML message comprising a PLM data set of information and a corresponding complete STEP Part21 representation suitable for processing by an industry strength STEP processor is defined by the OMG PLM Services standard specification. Furthermore, the MDA nature of the specification would allow to extend the data exchange capabilities to other implementations or representations and support the interoperability and compliance of the different data representations.

3.2.2 PLM Vision

PLM Services aims to the standardization of the communication of the product development information among different partners of the automotive sector. Figure 9 and Figure 10 explain how PLM services aims to improves the business of enterprises.

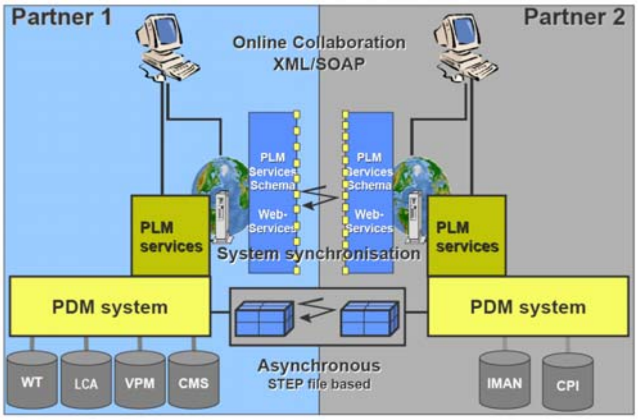

Figure 9: PLM Services network schema

PLM Services aims to improve greatly the development phase of producs, but cares only about this. The focus is on the product type, so the design and the project of the product. The physical item is outside the PLM Services aim.

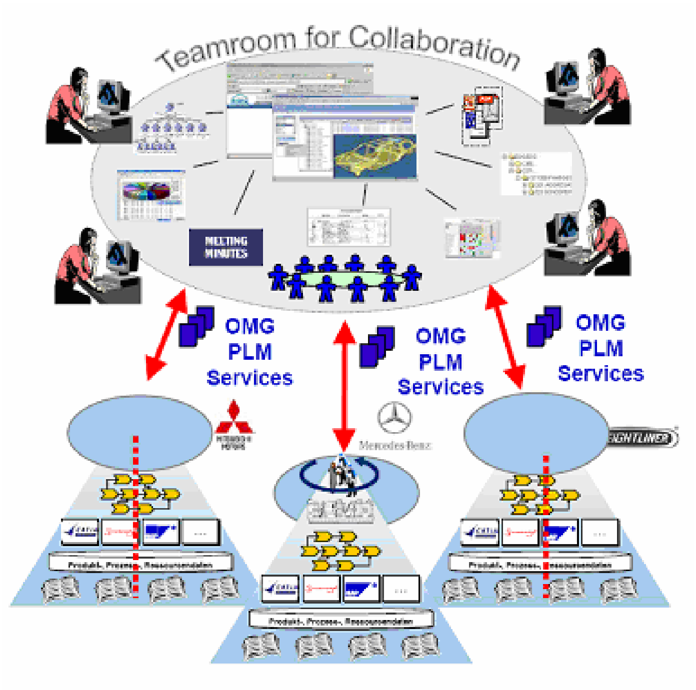

Figure 10: PLM Services usage

3.2.3 Summary

PLM Services aims to standardize the Product Type lifecycle data management, so to manage the development phase where the product is designed with strong collaboration of different enterprises, with different revisions of the same document.

PDKM instead aims to standardize the product item lifecycle data, which is the physical instantiation of the project. Moreover the design phase in the PDKM has not been developed, but has been kept as an empty class to be filled with other standards.

So PLM Services and the PDKM can be considered perfectly complementary. Moreover a collaboration is possible and such plans with the Object Management Group (OMG) is presented in DI1.7 [2].

4 Evaluation of PROMISE Middleware Interface

PROMISE Middleware is being developed in WP R6. The main role of Middleware is to allow PDKM systems to communicate with PEIDs even though the PEIDs may be mobile and only have intermittent network connectivity. In reality, the Middleware should provide a generic means for any "nodes" in a network to query and update information of other nodes if/when needed. This means that it should also be possible to implement e.g. PDKM-to-PDKM and PEID-to-PEID communication using the Middleware.

The structure of a PROMISE Middleware node is shown in Figure 11. The UPnP CorePAC interface has been defined for direct communication between the Middleware node and PEIDs with limited computing resources that do not allow them to implement an embedded PMI interface. Promise deliverables DR4.2 [6] and DR6.2 [7] specify this UPnP CorePAC to a level that makes it possible for third-party developers to implement it. However, no standardisation effort has been planned in PROMISE for this UPnP interface.

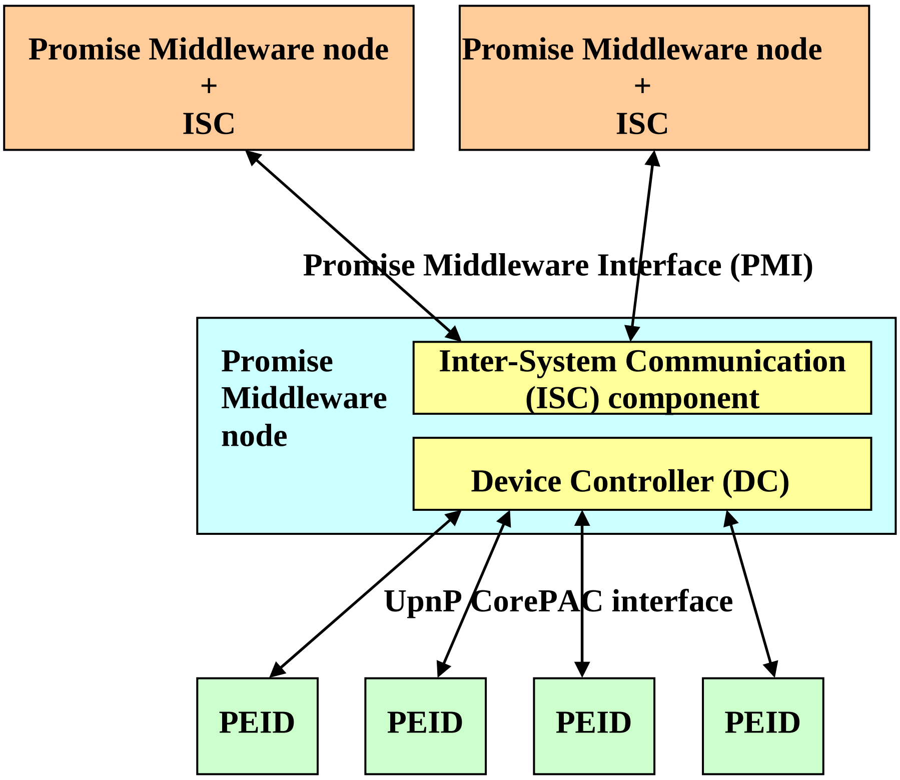

Figure 11: Structure of a PROMISE middleware node and the interfaces it can support.

The Promise Middleware Interface (PMI) has been specified as a Web Service interface in PROMISE deliverable DR6.5 [8]. The public interface has been defined using few methods with only one parameter, thus reducing the semantics of the method interface as much as possible. Instead, most of the semantics has been included into the XML messages exchanged over PMI. The message semantics are defined by XML Schemas. Including most of the semantics in the messages themselves signifies that it is easier to extend them in the future than if the semantics would be defined in the method interface. It also means that even though PMI is currently implemented using WSDL/SOAP technology, it would also be easy to support PMI using other communication protocols.

A major driver behind PMI is that three PROMISE partners already have their own Middleware implementations that have been developed at least partially for different application domains and using different design principles. In order to profit from these different Middleware implementations (and possibly other Middleware implementations in the future), PMI has been defined in a way that such different implementations would be able to communicate successfully with each other, thereby accomplishing the functionality needed by PLM together while continuing to fulfil their original purpose as well as possible.

The inter-operability of the three different PMI implementations is currently (April 2007) being tested, which has already been useful for refining the PMI version presented in DR6.5 [8]. Potential needs for extensions have also been identified. Improvements and extensions to PMI will be documented in the PROMISE architecture series documents. The final description of PMI should be detailed and tested enough to be directly usable as a standard's document.

The current PMI mainly uses "pull-based" functionality, where information can be requested (or written) either directly (synchronous operation) or using a call-back interface (asynchronous operation). However, it is not decided for the moment whether all so-called PLM events would be sent using this pull-based functionality or if a more "push-based" functionality would also be needed. For the moment, such push-based functionality is obtained using a subscription mechanism but it remains to be identified whether that is sufficient in all cases.

Many of the design principles of PMI seem to have elements in common with the EPC Information System (EPCIS) proposal of EPCglobal, published on April 16th, 2007. It is still too early to make any conclusions about whether EPCIS could be used in the PROMISE Middleware and to what extent that could influence the evolution of PMI. This activity will be performed in the coming months and its outcomes will be reported in DI1.6 and DI1.7 due M36.

5 Summary of standards evaluation

The review of PROMISE architectural components results in a revision of PROMISE's standardisation plans (refer to DI1.5 [1] to the standardisation plans as of M24). The key variation in the standardisation plan is that PROMISE will expand its focus on the standardisation efforts on Product Lifecycle Event notifications to the whole PROMISE Middleware Interface (PMI) as recommended by the reviewers. Summarising, PROMISE standardisation activities will focus on the following:

1. PROMISE Architecture: The standardisation work package partners will continue to support WP R12's activities on the development of the architecture specifications, and will evaluate these specifications through discussions with the wider standards community. The Open Group will be used as the primary mechanism to achieve this.

2. Standard for product lifecycle data representation: This standard will be based on the PDKM object model.

3. Standard for product lifecycle data exchange. This standard will be based on the PROMISE Middleware Interface.

6 Action plan for evaluation and refinement of PROMISE standards

PROMISE architecture specifications and proposals for standards arising from them have to be evaluated and refined continuously in order to increase popularity and therefore the possibility of widespread adoption. In order to achieve this, the Open Group (described in detail in DI1.7 [2]) will be used as the primary mechanism. Furthermore, a number of tasks have been planned in order to complete the standardisation process of the two standards. These are outlined in the following sub-sections.

6.1 Standard for product lifecycle data representation

- **Publication of the PDKM object model in the Open Group Architectures Forum**.

  The PDKM object model defined in DR9.2 [4] will be modified to provide a complete description of the specification. This work will be carried out with the highest priority and will be published in the Open Group Architectures Forum to generate discussions and lead to further refinement of the model.

- **Development of MySQL reference implementation**.

  As explained before in section 2, there is a need to develop an implementation of the PROMISE PDKM semantic model using the MySQL open-source DBMS. This is a good opportunity for exploiting the semantic data model in a more easily reusable implementation solution, covering the needs of both academic/research and industrial partners. This need was expressed both within the PROMISE consortium as well as the wider industrial community as this allows adoption of the standard without getting locked into proprietary software and technologies. This work will be carried out in the next six months and will be presented in DI1.6 due M36.

- **Development of a definitive version of the UML 2.0 class diagram representing the PROMISE PDKM semantic object model**.

  This involves development of a definitive version of the UML 2.0 class diagram representing the PROMISE PDKM semantic object model, based on a final analysis of the eventual new requirements coming from the ten PROMISE demonstrators, which could have arisen in the last year of project activity (during which the model was not substantially modified). This will result in a UML 2.0 class diagram and the related detailed explanation. This will be presented in DI1.6 due M42.

- **Development of a UML 2.0 class diagram representing the PDKM object model at the interface level.**

  This involves development of a new UML 2.0 class diagram representing the specifications at the interface level, in order for the PROMISE PDKM standard proposal to be more specific concerning its implementation, and thus to make it more easy to be implemented. This will result in a UML 2.0 class diagram with the specification of the semantic object model at the interface level, and the related explanation. This will also be presented in DI1.6 due M42.

6.2 Standard for product lifecycle data exchange

The PROMISE product lifecycle data exchange standard will be based on the PMI. The specifications for the PMI, in particular the product lifecycle event notification is currently under development under workpackages R6 and R12. The first step in this domain would be to define a series of tasks similar to those defined in section 6.1 required to evaluate and refine the product lifecycle data exchange standard.

In the meanwhile, Cambridge and Indyon will work closely with the development of PMI specifications to understand similarities between the interfaces defined by PROMISE and those defined in EPCIS by EPCglobal. On these lines, a detailed comparative study between the PROMISE architecture (PMI in particular) and the EPC Network Architecture (EPCIS in particular) will be performed and reported in DI1.6 due M36.

7 Conclusions

This report is the first of a three-part series of deliverables aimed to document the activities on evaluation and refinement of PROMISE standards and architecture specifications. In this report, we have provided an evaluation of the PDKM object model against the requirements of PLM and compared it with PLCS and PLM Services. In doing so, we have shown that the PDKM object model offers definite advantages over the two standards, and the industry would benefit from it.

We also evaluated the suitability of PMI for standardisation and concluded that efforts will be focussed on standardising the PMI as a whole as opposed to product lifecycle events as planned earlier. Finally, we described the action plan for further evaluation and refinement of PROMISE standards over the next 12 month period.

The next deliverable due M36 will report on evaluation and refinement activities carried out in the six month period. It will also describe action plans for evaluation and refinement of the PMI specifications, which is expected to evolve through the work done in WP R6.

Bibliography

[1] A. Parlikad, J. Brusey, K. Framling, D. Barisic, L. Rabe, H.-H. Do, C. Dikici, M. Neugebauer, and A. Metin, "DI1.5: Recommendations for standards,"DI1.5, 2006.

[2] A. Parlikad, J. Cassina, D. Potter, D. Kiritsis, L. Rabe, P. Folan, M. Tomasella, and K. Framling, "DI1.7: Promotion of PROMISE standards,"DI1.7, 2007.

[3] J. Anke, H.-H. Do, M. Marquard, A. Metin, C. C. Rostad, and B. Smith, "DR9.1: Design of PROMISE Information Management System (PDKM),"DR9.1, 2005.

[4] M. Tomasella, J. Cassina, A. Metin, and M. Marquard, "DR9.2: Specification of the system object model,"DR9.2, 2006.

[5] A. Saaksvuori and A. Immonen, *Product Lifecycle Management*, 2Rev Ed ed Springer-Verlag Berlin and Heidelberg GmbH & Co., 2005.

[6] G. Stromberg, D. Barisic, and M. Verbeek, "DR4.2: PEID core prototype,"DR4.2, 2005.

[7] J. Anke, G. Hackenbroich, S. Kuhn, M. Neugebauer, and Z. Nochta, "DR6.2: Reader Controller Specification,"DR6.2, 2005.

[8] B. Forss, J. Petrow, K. Framling, C. Dikici, and J. Anke, "DR6.5: Interface definition and design of enterprise communication infrastructure,"DR6.5, Nov.2006.

