DR9.6c: Specification of interfaces (revised)

Written by:
Björn Forss, Jouni Petrow, Trackway
Hong-Hai Do, Anja Klein, SAP
Altuğ Metin, Michael Marquard, InMediasP

DELIVERABLE NO	**DR9.6c: Specification of interfaces (revised)**
DISSEMINATION LEVEL	**CONFIDENTIAL**
DATE	15. May 2007
WORK PACKAGE NO	WP R9: Development of PROMISE Information management system
VERSION NO.	1.0
ELECTRONIC FILE CODE	dr9_6c_1_0.doc
CONTRACT NO	507100 PROMISE A Project of the 6th Framework Programme Information Society Technologies (IST)
ABSTRACT	This document specifies interfaces between the PDKM and other significant components of the PROMISE architecture. Within the PROMISE architecture, these are the Middleware (WP R6), the DSS (WP R8) and PEIDs (WP R4). Thereby, it has strong links to several deliverables and tasks of the work packages R4, R6, and R8. Basically a copy of deliverable DR9.4 "Specification of interfaces" that referred to version 1 of the PDKM, the present deliverable updates the specification with the developments in this area targeted for the PDKM prototype version 2 and developments planned until the end of the PROMISE project.

STATUS OF DELIVERABLE		
ACTION	**BY**	**DATE** (dd.mm.yyyy)
SUBMITTED (authors)	J. Petrow, B. Forss, H.-H. Do, A. Klein[1], M. Marquard, A. Metin	07.06.2007
VU (WP Leader)	A. Edler	07.06.2007
APPROVED (QIM)	D. Kiritsis	07.06.2007

[1] The authors from Trackway, formerly called Stockway, and SAP did not work on DR9.6c itself. However, they were main contributors to DR9.4, which is revised with the present deliverable. Hence, they are listed as authors as well.

Revision History

Date (dd.mm.yyyy)	Version	Author	Comments
06.06.2007	1.0	Michael Marquard	Document initiated (copy of final version of DR9.4), updated, and finalised

Authors' contact information

Name	Organisation	E-mail	Tel	Fax
Jouni Petrow	Trackway	Jouni.petrow@Trackway.eu	+358 40 571 2322	
Björn Forss	Trackway	Bjorn.forss@Trackway.eu	+44 118 9001 064	+44 118 9001 061
Anja Klein	SAP	Anja.klein@sap.com		
Hong-Hai Do	SAP	Hong-hai.do@sap.com		
Michael Marquard	InMediasP	promise@inmediasp.de	+49 3302 559 420	+49 3302 559 124
Altuğ Metin	InMediasP	promise@inmediasp.de	+49 3302 559 420	+49 3302 559 124

Table of Contents

List of figures

Abbreviations:

Abbreviations used in this document:

AP	Application Protocol of STEP
BOL	Beginning of Life
DfX	Design for X
DoW	PROMISE Description of Work
DSS	PROMISE Decision Support System
EOL	End of Life
GUI	Graphical User Interface
IOCI	Inter-Organizational Communication Infrastructure
ISC	Inter-System Communication layer (formerly known as IOCI)
JDBC	Java Database Connectivity
JSP	Java Server Pages
MOL	Middle of Life
ODBC	Open DataBase Connectivity
OLAP	Online Analytical Processing
OMG	Object Management Group
PDKM	Product Data and Knowledge Management
PDM	Product Data Management
PEID	Product Embedded Information Device
PLM	Product Life-cycle Management
PMI	PROMISE Middleware Interface
PROMISE	PROduct life cycle Management and Information tracking using Smart Embedded systems
RHL	Request Handling Layer of the PROMISE Middleware
SCM	Supply-chain Management
SOA	Service Oriented Architecture
STEP	Standard for the Exchange of Product model data
WP	Work package of PROMISE project
XML	eXtensible Mark-up Language

1 Introduction

1.1 Purpose of this document

This document is the outcome of work done in tasks TR9.4 “Specification of interfaces” and TR9.6 “Refinement of DR9.2, DR9.3 and DR9.4”. It specifies interfaces between PDKM and other relevant systems. Within the PROMISE architecture, these are the Middleware (developed by WP R6 “Intra- and Inter-enterprise software infrastructure”), the DSS (developed by WP R8 “Methodologies for decision making for BOL, MOL, EOL”), and PEIDs (topic of WP R4 “PROMISE Core PEID”).

1.2 Context of this document

WP R9 “Development of PROMISE Information management system” is part of Research cluster RC-4, Knowledge Treatment and Decision Making, and is dedicated to the design, specification, and implementation of the PDKM system. As specified in the DoW of the PROMISE project, the roadmap to this goal consists of several tasks and is also closely integrated with the tasks of WP R10 “Design and implementation of an integrated and personalised user interface for the PDKM system”.

The first version of the present document, deliverable DR9.4 “Specification of interfaces”, was outcome of task TR9.4 of WP R9. It was revised in task TR9.6, leading to the present deliverable DR9.6c.

1.3 Document overview, boundary conditions and limitations

Chapter 2 recaps how the PDKM is embedded in the PROMISE architecture. By this, it identifies the relevant interfaces to be discussed in the present deliverable. They are topic of chapter 3. Chapter 4, finally, provides some concluding remarks.

Sections 3.1 and 3.2 describe interfaces to “PROMISE-internal” systems communicating directly with the PDKM, i.e. the PROMISE Middleware and the DSS. On the contrary, section 3.3 discusses how 3rd-party systems not developed within the PROMISE project interface to the PDKM. Regarding the communication between PDKM and PEIDs, section 3.4 gives an outline of the topic of data semantics. Technically, PDKM always accesses PEIDs through the PROMISE Middleware. However, implementation and specific design issues are outside of the scope of this deliverable.

Main inputs for this deliverable have been other deliverables from work packages R4 (PEID), R6 (Middleware), and R8 (DSS), and direct discussions with these work packages. Deliverable DR6.5 “Interface definition and design of enterprise communication infrastructure” defines in a first version the interface to access PEID data through the Middleware, the so-called PMI (PROMISE Middleware Interface). This interface is accompanied by the Inter-System Communication layer (ISC[2]), which is used to reach PEIDs outside one’s own organisation. The ISC is one element of the PROMISE Middleware. Independent of if a communication via the PROMISE Middleware is intra- or inter-organisational, the PMI is used as interface. The PMI is main topic of section 3.2 but plays also a role in the other sections of chapter 3. This underlines the significance of the PMI.

[2] The ISC is a component of the PROMISE Middleware offering a platform for sharing PEID data across multiple organisations.

Both the PMI and the interface between PDKM and DSS will still be developed further to cover requirements recently identified during the realisation of the PROMISE demonstrators. Nevertheless, the present document tries to anticipate these developments. Thou, changes to the specification presented here might occur. They will be documented in deliverable DR9.13 “PDKM Prototype Version 3”, due M40.

1.4 Delta to deliverable DR9.4 “Specification of interfaces”

The present document is based on the former deliverable DR9.4 and aims at completing it with the developments in this area targeted for the PDKM prototype version 2 and developments planned until the end of the PROMISE project. Hence, it basically is a copy of DR9.4 with punctual changes. In order to help the reader to identify the differences between DR9.4 and this deliverable, the most significant changes are listed here:

- References to future versions of PDKM or DR9.4, namely DR9.6c, and future tasks of mentioned work packages have been removed or updated. There are not scheduled any further revisions of this deliverable. Hence, this deliverable tries to cover the most important aspects of the PDKM’s interfaces as intended until the end of the PROMISE project. Nonetheless, minor changes not anticipated until now might occur. They will be described in deliverable DR9.13, due M40.
- The term *IOCI* was replaced by *ISC*. Correspondingly, the terms *RHL interface* and *IOCI interface* were replaced by *PMI*. Besides these “syntactic” changes, it was reflected in the text that these formerly two interfaces have been replaced by one interface. Furthermore, it was incorporated that the PMI, though still under development, is, in contrary to the situation at delivery of DR9.4, now defined in a first version. The main changes with respect to this topic have been in sections 1.3, 3.3, and 3.2. The latter has been changed substantially. Furthermore, the two appendices have been replaced by references to deliverable DR6.5.
- Section 3.4 has been changed substantially.
- In section 3.1, the topic of invoking DSS functionality from the PDKM or vice versa was changed substantially.

Besides that, there have been minor editorial changes not changing the content all over the document.

2 PDKM in the whole PROMISE system

Being a central component of the overall PROMISE approach, the PDKM system aims at systematically integrating and managing data from all life cycle phases of products. The ultimate goal is to integrate product data of the entire life cycle from different sources and furthermore to support comprehensive analysis on the integrated data and to enable the enhancement of operational businesses with the obtained insights on products.

Before discussing the interfaces between the PDKM and other PROMISE components in detail, it is instructive to review the objectives and tasks of the PDKM and how it is functionally related to the other components of the PROMISE architecture. In the following an overview of the overall PROMISE architecture is given and the relevant components of the PDKM architecture are described.

2.1 Global objective and tasks of PDKM

As stated in DR9.1 “Design of PROMISE Information Management System (PDKM)”, the prime objective of the PDKM system is *to perform comprehensive data integration for decision* *making/problem solving to improve BOL, MOL, and EOL operations on products*. This leads to the two major tasks, data integration and data analysis for decision support, which are briefly introduced in the following:

- *Integration of product and field data*: Product data is commonly managed in PDM systems, which are generally intended for BOL, but also required in MOL and EOL businesses. On the other side, field data is specific for MOL and EOL operations. It is to be found in corresponding field databases or can be directly captured by PEIDs attached to products. The Middleware component developed in WP R6 provides a uniform interface to access heterogeneous PEIDs, thereby essentially simplifying the data integration task for the PDKM system.
- *Support for BOL, MOL, and EOL decision making/problem solving*: The second task in developing the PDKM concerns the analysis performed on the integrated data. In order to support a large variety of application scenarios from BOL to EOL with different focus and requirements, a wide range of analysis methods need to be offered, ranging from simple searching and browsing techniques to find the data of interest to advanced methods like querying, reporting, Online Analytical Processing (OLAP), and data mining. Although this task can benefit from the large body of previous approaches developed in the field of data warehousing and business intelligence, the major challenge remains integrating existing analysis algorithms with the PDKM system and evaluating them for life cycle data.

Figure 1 visualises these two tasks in the PROMISE architecture.

2.2 The PROMISE architecture

The identification of the two main tasks of the PDKM has a profound impact on the overall PROMISE architecture, which is visualised in Figure 1. In particular, the PKDM is situated at the top of the PROMISE architecture and represents the most visible component providing a graphical user interface for the users to access and use the entire PROMISE system. Behind the scene, the PDKM integrates data from a variety of data sources supporting the BOL, MOL, and EOL operations, respectively, as specified by the first task of the PDKM.

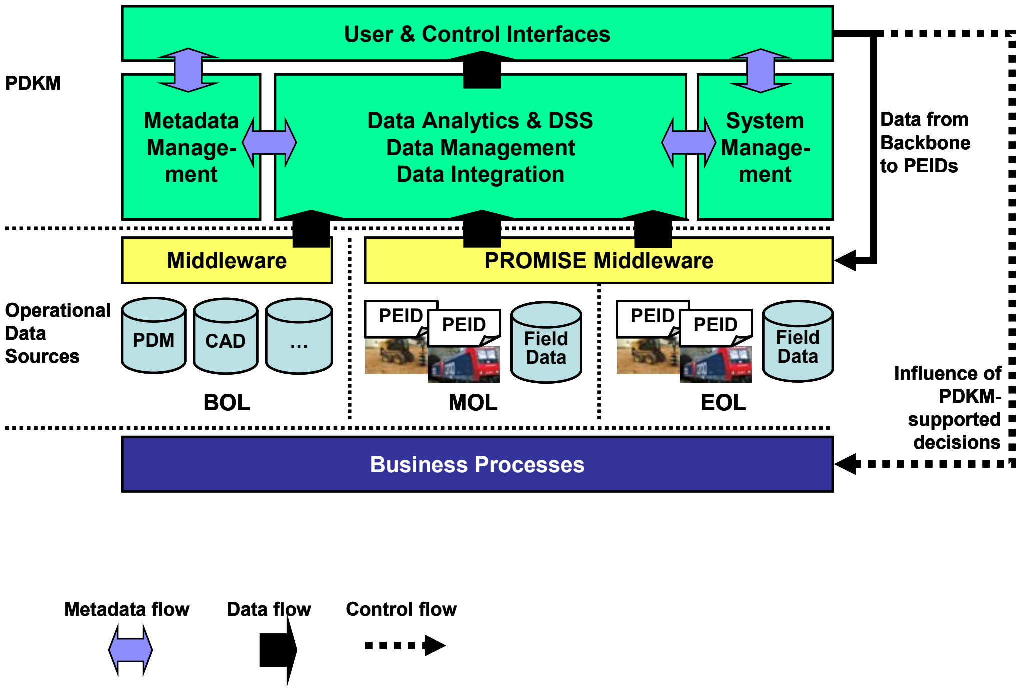

Figure 1: Tasks of PDKM in the PROMISE architecture

Figure 1 also shows how the PDKM closes the information loop. The business processes produce the data that is captured and fed into the PDKM system, while conclusions drawn on this data in turn are fed back to the business processes as the so-called business intelligence decisions. The decisions eventually lead to changes in operational businesses and are then reflected in the PDM systems, field databases etc., maintained by the corresponding businesses, which then feed the PDKM system with new data.

Due to the wide range of data sources to be supported, the PROMISE architecture includes the so-called middleware components, which aim at shielding the heterogeneity of the data sources and allow the implementation of interfaces between the PDKM and the data sources to be kept at a manageable complexity. It is important to differentiate between the PROMISE Middleware being developed in WP R6 and common middleware, which may be built on existing 3rd-party middleware technologies, such as the Open Database Connectivity (ODBC) standard interface. While the PROMISE Middleware specifically focuses on mediating the communication between the PDKM with the PEIDs attached to products, the common middleware supports data import from persistent storages, such as PDMs and field databases, into the PKDM.

2.3 The PDKM components

Figure 2 illustrates the PDKM architecture with its components and the data sources that are themselves not part of the PDKM but feed it with data. In particular, the architecture shown is a model of abstract *layers* and *towers*, which represent logical building blocks of the architecture and are coupled with each other. At the lowest level of the architecture, the *Data Source* layer comprises all operational data sources feeding the PDKM system with relevant data. The PDKM system itself consists of six components: four layers for *data integration*, *data management*, *data analytics*, and *user and control interfaces*, respectively, and two towers for *metadata management* and *system management*, respectively. The towers provide functions that concern several layers at the same time. In the following, the data sources as well as the components of the architecture are introduced and their interaction is discussed.

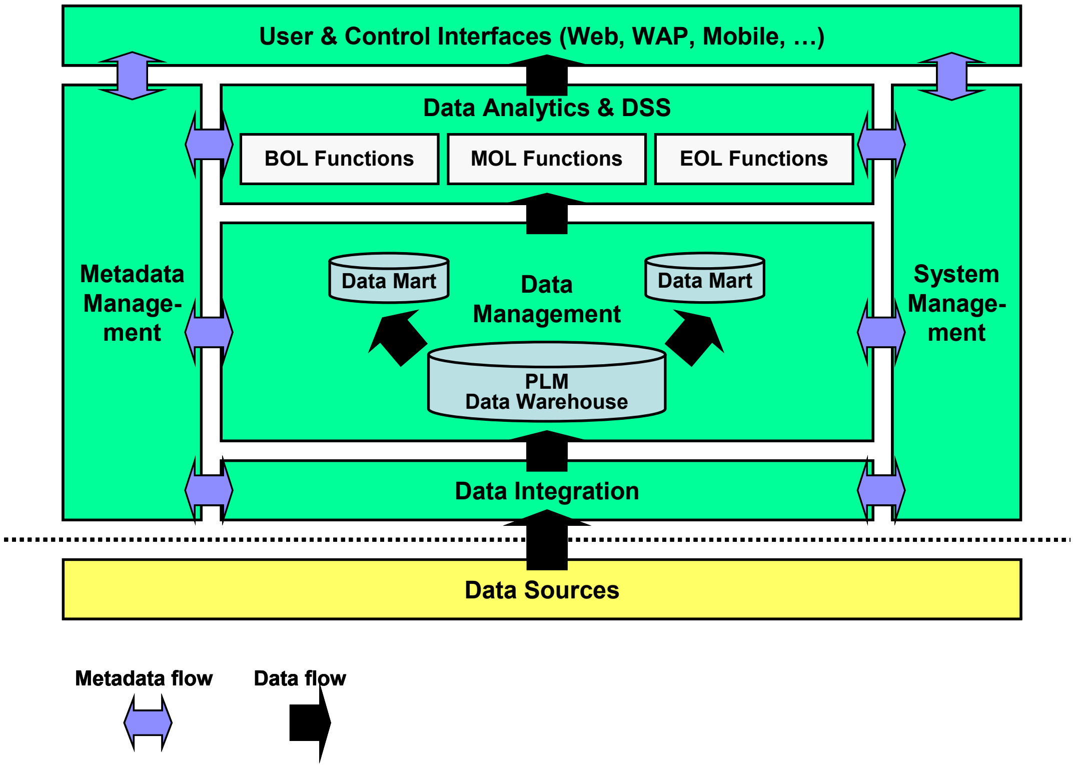

Figure 2: Architecture overview of the PDKM system

2.3.1 Data sources and architecture components

The main purpose and functions of the Data Source layer and the components of the PDKM system are:

- *Data Source layer*: Relevant data is imported from BOL, MOL, and EOL operational systems. BOL-specific data sources include databases, like PDM and SCM, which are largely static. On the other side, MOL and EOL-specific data is captured in a real-time manner from products using the attached PEIDs or in PDM and field databases as required for the MOL and EOL operation. The data sources for the PDKM system, e.g. PEIDs, PDM, and field databases, may be directly accessed or indirectly via a middleware component. Examples of middleware components are the PROMISE Middleware developed in WP R6 for communicating with PEIDs, drivers based on the Open Database Connectivity (ODBC) specification to access relational databases, or standards like OMG's Product Life Cycle Management Services (PLM Services) to exchange data with PLM systems.
- *Data Integration layer*: This layer consists of all tools, scripts, and programs utilised to import source data into the PDKM system. Before passing into the PDKM system, the data might be transformed, cleaned, and unified under a consistent global representation, also called the global schema.
- *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 so-called *PLM Data Warehouse* or *data warehouse* for short, which implements the global schema, thereby providing a uniform data representation for queries and analysis.
- *Data Analytics layer*: This component comprises all algorithms and tools employed to perform data analysis and support decision making/problem solving. On top of the collection of such generic methods are customised analysis applications, e.g. for DfX, Predictive Maintenance, or Effective Recycling. WP R8 is dedicated to the design and development of the Decision Support System (DSS), which is an essential part of this layer.
- *User & Control Interface layer*: This component implements the interface of the PDKM system to users. According to the types of users and 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, personalised interfaces are provided with different views on data and allowed analysis functions. The user interface is provided by WP R10. Apart from data representation and analysis functions, this component provides means to communicate with other systems, e.g. sending updated configuration data to a PEID.
- *Metadata Management tower*: This component 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.

2.3.2 Interaction between components

For the specification of the interfaces, it is necessary to understand thoroughly Figure 2, which also indicates the flows for data and metadata between the data sources and components of the

PDKM architecture. *Data exchange* represents the major interaction activity between components in the PDKM and in the PROMISE system in general. In particular, data is obtained from the operational sources, transformed and unified in the data warehouse in the Data Management layer, and finally, passed to the analysis methods in the Data Analytics layer. On the other hand, *metadata exchange* does not only ensure correct processing and exchange of data, but supports also documentation purposes. It is typically done together with data exchange. Consequently, most interfaces to be specified in this deliverable address both issues, namely data and metadata exchange, by providing corresponding methods.

The complex environment of the PROMISE system presupposes a large number of interfaces to be specified and implemented between the architecture components. However, in this deliverable the focus is on the most important technical interfaces in the PROMISE architecture, namely between the PDKM and the DSS, between the PDKM and the PROMISE Middleware, and finally between the different PDKM instances maintained and controlled by different organisations. The orientation towards the work package structure, especially of WP R6 (PROMISE Middleware), WP R8 (DSS), and work packages R9/R10 (PDKM), requires a closer collaboration between partners of the corresponding work packages and an easier integration of the work done in the different work packages:

- *Interface between the PDKM and the DSS*: This interface mainly describes the information exchange between the Data Management layer and the DSS as central component of the Data Analytics layer in the PDKM architecture. The basis of this interface is the global schema describing all entities that are relevant for analysis tasks. The interface allows the DSS to query and fetch instance data of entities of interest.
- *Interface between the PDKM and the PROMISE Middleware*: The first focus of this interface is the import of real-time field data from PEIDs attached to products. It supports both pull and push mechanisms. Furthermore, the critical issue of availability of the PEIDs is addressed, as they may not always be online or accessible for the PROMISE Middleware. Secondly, this interface aims at supporting the data exchange across the organisational border e.g. the communication between different implementations of the PDKM architecture in different organisations. In this context, this interface deals with security aspects (authorisation, secure transmission, restricted view on data, etc.). However, this document does not address security issues in more detail. They are considered in a dedicated work package, WP R13 “PROMISE End-to-End Security”.
- *Interfaces to 3rd-party systems*: The main interface to 3rd-party systems, like VIPSCARSIS, is via the PMI. Here, it does not matter for the PDKM if the PMI is used via the PROMISE Middleware or if 3rd-party systems directly use the PMI-support of the PDKM. Nevertheless, other interfaces might be provided that are supported by the PLM system that underlies the PDKM. Since it is expected that these interfaces adhere to common standards like STEP, all 3rd-party systems also supporting these standards are enabled to exchange data with the PDKM.
- *Metadata flows*: Metadata is used to ensure correct processing and exchange of data. It is stored and exchanged together with data. For documentation purpose, the Metadata Management tower captures relevant metadata from other components, in particular, the Data Integration, Data Management, and Data Analytics layers. For example, the user can navigate in the captured metadata to get information about the data available in the data warehouse. On the other side, metadata may also actively be utilised for management purposes. For example, by using metadata about user profiles, personalised user interfaces can be automatically generated.

3 PDKM interfaces

In the following section, interfaces between the PDKM and related components of the PROMISE architecture are described. First, the interface between the PDKM and the DSS is discussed. Subsequently, the structure and functionalities of the PDKM – Middleware interface are given and interfaces to 3rd-party systems are characterised. The section ends with a short outline regarding the PDKM – PEID data semantics.

3.1 PDKM – DSS interface

The PDKM and DSS are developed within different work packages (R9 and R8) although DSS is – according to DR9.1 – considered as a part of the PDKM. The results of the work in WP R8 are then integrated into the PDKM. Due to this organisational structure of work distribution, the interface between these systems is described in this deliverable while other interfaces between different PDKM-components (compare Figure 2) are as intra-WP R9/R10 issues not dealt with in detail in this document. Nevertheless, the connection between PDKM and DSS as an integrated part of the PDKM is "tighter" than the connection between PDKM and extra-PDKM systems. However, in order to allow a quick and independent development of the PDKM and DSS prototypes and to support the use of the DSS as a standalone application with an associated database, as well, this connection is also "looser" than the connection between other PDKM components. Of course, this "transfers" to the respective interfaces.

The DSS itself is a Java based web application running in a web application server independent from the PDKM. The decision to use a web application server independent from the PDKM was based on the fact that the portal framework chosen for the PDKM, the SAP NetWeaver Portal (see DR9.1), does not support the most current Java version while this is required by the DSS in order to use several 3rd-party libraries.

It is noted that besides a browser based user interface, the DSS might offer a native GUI with advanced facilities or better usability targeting a small group of users, which might be seen as "DSS power users". However, the scope of the browser based DSS user interface developed by WP R8 together with WP R10 is agreed between work packages R8/R10 and the application scenarios in a way such that the browser based user interface is sufficient for the needs of the PDKM/DSS users.

Concerning the browser based user interface of the DSS, it is underlined that the DSS architecture allows based on web services several user interfaces connecting to the DSS backend. Deliverable DR10.4 "Integration of DSS user interface" is exclusively dedicated to the interface between PDKM and DSS on GUI level and gives also a good overview of the respective DSS architecture. Thus, there can be developed a stand-alone browser based GUI for the DSS e.g. basing on JSPs. However, for the sake of integration, WP R10 developed in collaboration with WP R8 DSS GUIs on the same technical basis as used by the PDKM GUI, i.e. SAP's iViews[3], allowing a seamlessly integrated PDKM/DSS GUI.

With respect to *data integration*, the DSS uses the PLM Data Warehouse, implementing the global schema (see Figure 2), in different ways:

- Data that comes from the Data Integration layer (see Figure 2) or via user input and that needs to be analysed by the DSS is stored here.
- The DSS generates data/information/knowledge to be stored in the PLM Data Warehouse.
- The DSS stores data for DSS internal use like parameters for the required algorithms.

[3] For more details about iViews, refer to the help portal of SAP on http://help.sap.com/.

This leads to different parts of the PLM Data Warehouse:

1. Tables that are readily provided by the PLM system that is used as basis for the PDKM
2. Tables that are required for DSS internal use
3. Tables that are required by the PDKM and are not covered by points 1 and 2

For parts 2 and 3, there is defined a naming standard to avoid conflicts with the PLM system underlying the PDKM.

Based on these different categories of parts of the PLM Data Warehouse, different ways are defined for the access of the DSS to the PLM Data Warehouse:

a) All read access of the DSS uses JDBC. If appropriate, there might be defined additional database views to reduce the complexity of the algorithms used in the DSS.
b) For write access to the above listed parts 2 and 3, the DSS uses JDBC.
c) For write access to part 1, the DSS invokes via web services PDKM functionality that maps the request to API functions provided by the underlying PLM system. This solution was chosen due to the complexity of the data model of the underlying PLM system in order to avoid future problems that are very difficult to analyse and to solve. For calling the PDKM functionality, the DSS uses the PMI format defined in deliverable DR6.5. In detail, the write function defined there is applied.

With respect to *invoking* DSS *functionality* from the PDKM or vice versa, there are several scenarios:

- Primarily, the complete DSS web GUI and single DSS functionality are integrated directly into the PDKM web GUI. This is already discussed above here in this section; more details can be found in deliverable DR10.4, exclusively dedicated to this subject. Thus, DSS functionality is directly accessible by the PDKM user. Moreover, user actions in the PDKM part of the GUI might trigger responses in the DSS part of the GUI using exactly the same mechanisms as described in DR10.4 combined with inter-iView communication supported by the employed SAP NetWeaver Portal.
- Nevertheless, there might also be situations where functionality of the respective other system has to be invoked without direct user interaction. This communication uses web services, which is the already chosen mean for communication between PDKM and Middleware. How to call DSS functionality from the PDKM is described in deliverable DR10.4. For calling PDKM functionality, the DSS uses again the PMI. However, here, the enhancements/extensions of the PMI to be defined (see section 3.2.3) will probably play an important role in order to offer here a powerful interface.

It is noted that there will not be any integration with respect to the native, not web based DSS GUI mentioned above. However, with respect to data integration the type of user interface is not important.

3.2 PDKM – Middleware interface

3.2.1 General use of the PMI

The interface between the PDKM and the PROMISE Middleware is the so-called PMI (PROMISE Middleware Interface). The PDKM system interacts with the Middleware to be able to retrieve data about or from a PEID and data about different kinds of events like the so-called PLM events (see the future deliverable DR6.7 “PLM event creation and communication”, due M36). Via the Middleware, the PDKM is also able to write data to PEIDs. The data can be read in real time from the PEID itself or from buffers in the Middleware. Historical data can be retrieved from other PDKM systems or field databases. Data can be written in real time to the PEID or write commands can be buffered in the Middleware.

As indicated in the previous paragraph, the Middleware is able to buffer requests and responses. This is an important feature, especially, since PEIDs usually are not continuously connected to/accessible by the Middleware.

The communication between Middleware and PDKM is based on a service-oriented architecture (SOA) where the Middleware provides web services that the PDKM system can utilise to retrieve and transmit the desired data. Different components in the Middleware provide different types of services. Using the Middleware, the PDKM system can make requests to PEIDs in a single organisation environment. Thou, in the same way it can interact with PEIDs that reside outside the requesting organisation in a multi organisational environment. Regardless of, in which kind of environment it currently finds itself, the PDKM system has not to take care of, to which component of the Middleware to send a service request, but simply uses the PMI and the Middleware takes care of the rest.

Worth noticing is that for all PMI-requests, there is the possibility for requesting systems to specify certain recipients of a request. This enables the PDKM on the one hand to request PEID data not only directly from the PEID itself but also from for example field databases, which is of specific relevance with respect to historic data. On the other hand, the PDKM can serve in an analogue way as an information provider for other systems.

This PMI-option to specify certain recipients of a request can also be used if the backend system e.g. the PDKM wants to write to a PEID or read the latest sensor value only when the PEID is at a certain RHL (Request Handling Layer of the PROMISE Middleware). It should be noted, that in this context an RHL is not necessarily equivalent to a specific location. Nevertheless, this scenario could e.g. be used to address a PEID-writer or –reader that primes PEIDs during the production process of a product, which might a brick to the solution of how the life of a product/PEID in a PROMISE scenario starts. This issue is still under consideration among several work packages (see also section 3.2.3).

3.2.2 Backend to backend communication via the PROMISE Middleware

As already indicated in section 3.2.1, the PMI cannot only be used for PDKM-PEID-communication but is also suitable for communication between different backend systems. With respect to this, the extensions of the PMI to be defined (see section 3.2.3) will offer even more options as already described in section 3.2.1.

To use this option, a PDKM system or other backend systems like field databases have to support the PMI or parts of it and act as an information source in the ISC network (see section 1.3). Instead of implementing PMI-support directly into the respective backend systems, a respective adapter/middleware[4] can alternatively add required PMI-features to the backend systems. Then, for example, historic data for a PEID can be retrieved via the PROMISE Middleware from these sources.

3.2.3 Further remarks on PMI

Interesting aspects about PDKM-internal aspects with respect to PMI are described in deliverables DR9.6b "Specification of System Functions (revised)" and DR9.7 "Specification for the integration". For example, in DR9.7 can be found that the PMI-support of the PDKM is not only

[4] not in the sense mentioned elsewhere in this document

suitable to import or to request data but can also be applied to trigger any other functionality like requests to the DSS.

The PMI is defined in a first version in deliverable DR6.5. However, further changes like support of PLM events and other types of events are currently under consideration involving several PROMISE work packages in order to realise also non-standard cases in PROMISE demonstrators and to make the PMI more powerful. These changes will most probably be documented in deliverable DR12.2 “Architecture guide”, which will be updated every six months starting from M30. This means that a first view on significant PMI changes might be found in DR12.2 of M36.

If PMI updates will not be documented in DR12.2, deliverable DR9.13, due M40, will provide a reference to the respective documentation.

3.3 Interfacing 3rd-party systems

As described in section 3.2.2, one possibility for the PDKM to interface with 3rd-party systems is by using the PMI. This might be by direct implementation of the interface into the system but also an adapter between the system and the ISC could realise this. Notice that, for the PDKM, it is irrelevant if the PMI is used via the PROMISE Middleware or if 3rd-party systems directly access the PDKM using the PMI-support implemented there.

The PLM system underlying the PDKM might provide a further option of interfacing to 3rd-party systems. One of many reasons not to build the PDKM from scratch but to extend an existing PLM system (compare the results of WP R7 “Information and Knowledge Management Methodologies”) was that those systems often support import and export based on established standards like specific STEP APs. All 3rd-party systems also supporting those standards are thus enabled to exchange data with this PLM system and thereby with the PDKM.

Last but not least, the underlying PLM system is likely to support import and export on some proprietary formats. These formats can be used as well for interfacing with 3rd-party systems, possibly supported by suitable adapters.

The variety of possible 3rd-party systems, which may usefully interface with the PDKM and DSS, is potentially large and mostly of unforeseeable provenance. However, the choice of overall architecture for PROMISE (see Figure 1), and in particular the application of a flexible Middleware to connect the Operational Data Sources to the relevant part of the PDKM architecture enables a simple realisation of interfaces to both existing and future 3rd-party systems. Such Systems can take advantage of the ISC (see section 3.2.2), or, in the case of 3rd-party systems with a continuing and significant contribution to the application of PROMISE within a specific field, may connect directly to the PDKM or the Middleware with a bespoke interface solution.

An example of a 3rd-party system interfacing with the PDKM is the Field Data Base for the A10 Bombardier BOL Demonstrator, which is based on a usual database management system. In this case, the PROMISE Middleware is omitted and the Field Data Base directly accesses the PDKM using the PMI. On this way, MOL and EOL data from rail vehicle onboard PEIDs are passed to the PDKM and DSS systems in a format suitable for further data processing.

3.4 PDKM – PEID data semantics

The communication between the PDKM and PEIDs is only carried out via the PROMISE Middleware. Hence, the PDKM – PEID data semantics have to be defined using the means of the PDKM – Middleware interface.

With respect to this, the PMI defined in a first version in deliverable DR6.5 (see also section 3.2) and basing on the eXtensible Mark-up Language (XML) provides a concept of defining data semantics that is generic enough to support the requirements of all possible applications. In this approach, the diverse data types and structures of all possible data producers (PEIDs) are integrated in one semantic model. There, PEIDs are represented by PROMISE-wide uniquely identifiable targetDevices that comprise for each elementary information source that they support an infoItem that is uniquely identifiable within a targetDevice. For details, refer to DR6.5.

The only step that then has to be performed once for each PEID in the PDKM (or other backend systems using the PMI) is to define a mapping from the tuples, each consisting of a targetDevice's ID and an infoItem's ID, to the corresponding PDKM-internal identifiers and data objects. Similarly, tuples of targetDevice-IDs and infoItem-IDs are agreed between the PDKM and backend systems to communicate for example PLM event data (see section 3.2.3). Here, however, a targetDevice does not necessarily correspond to one PEID but usually identifies a physically existing product (that may have attached several PEIDs).

Besides that pre-data processing approach, the PMI offers also functionality that allows a receiving system to request metadata for a received data record from the sending system. This can e.g. be used during run time when a data record with unclear semantics is received. This option, however, will probably not be used by the PDKM.

4 Concluding remarks

This document specifies interfaces between the PDKM and other significant components of the PROMISE architecture. Thereby, it represents a link to several deliverables and tasks of other work packages (WP R4, WP R6, WP R7, and WP R8).

The communication between the PDKM and PEIDs is carried out via the PROMISE Middleware employing the PMI. This interface, presented in DR6.5 in a first version, can be used for both intra-organisational communication and data exchange across organisational borders. The latter is enabled by the ISC architecture, a part of the PROMISE Middleware. The PMI is the main interface to 3rd-party systems, too. However, common standards like STEP AP's supported both by the PLM system underlying the PDKM and respective 3rd-party systems enable further data exchange.

Moreover, the PDKM – DSS interface enabling the communication between the Data Management layer and the DSS as central component of the Data Analytics layer in the PDKM architecture was presented in detail. A short prospect of the PDKM – PEID data semantic topic was given.

As stated above, the PDKM interfaces and especially the PMI will be developed further in collaboration with other work packages. Nevertheless, it was tried to anticipate expected changes. Thou, modifications might occur that are not covered by the present deliverable. They will be documented in deliverable DR9.13, due M40.