



DR9.6b: Specification of System Functions (revised)

Written by:
Altuğ Metin, Michael Marquard, InMediasP

DELIVERABLE NO	DR9.6b: Specification of System Functions (revised)
DISSEMINATION LEVEL	PUBLIC
DATE	15.05.2007
WORK PACKAGE NO	WP R9: Development of PROMISE Information management system
VERSION NO.	1.0
ELECTRONIC FILE CODE	DR9_6b_V1_0.Doc
CONTRACT NO	507100 PROMISE A Project of the 6th Framework Programme Information Society Technologies (IST)
ABSTRACT:	The system functions of the PDKM are described. Basically a copy of deliverable DR9.3 "Specification of System Functions" that referred to version 1 of the PDKM, the present deliverable updates the scope of different areas of functionality that are provided to manage product-related data throughout the complete life cycle of the products.

STATUS OF DELIVERABLE		
ACTION	BY	DATE (dd.mm.yyyy)
SUBMITTED (authors)	A. Metin, Michael Marquard	24.05.2007
VU (WP Leader)	A. Edler	24.05.2007
APPROVED (QIM)	D. Kiritsis	24.05.2007

Revision History

Date (dd.mm.yyyy)	Version	Author	Comments
07.05.2007	0.1	Altuğ Metin	DR9.3 used as basis; identification of differences between PDKM prototype version 1 and version 2
14.05.2007	0.2	Altuğ Metin	Further input provided.
22.05.2007	0.3	Altuğ Metin	Final version
23.05.2007	1.0	Michael Marquard	Final revision and delivery

Authors' contact information

Name	Organisation	E-mail	Tel	Fax
Altug Metin	InMediasP	promise@inmediasp.de	+49 3302 559 420	+49 3302 559 124
Michael Marquard	InMediasP	promise@inmediasp.de	+49 3302 559 420	+49 3302 559 124



Table of Contents

1	INTRODUCTION.....	2
1.1	PURPOSE OF THIS DELIVERABLE	2
1.2	DELTA TO DELIVERABLE DR9.3 “SPECIFICATION OF SYSTEM FUNCTIONS”	3
2	PDKM SYSTEM FUNCTIONS.....	3
2.1	WEB PORTAL.....	4
2.2	PRODUCT AND PRODUCT STRUCTURE MANAGEMENT	4
2.3	CONFIGURATION MANAGEMENT	5
2.4	FIELD DATA FUNCTIONS	5
2.5	FIELD DATA INPUT	6
2.6	INCIDENTS MANAGEMENT.....	7
2.7	DOCUMENT MANAGEMENT	7
2.8	KNOWLEDGE MANAGEMENT.....	8
2.9	DATA ANALYSIS AND KNOWLEDGE GENERATION FUNCTIONS	8
2.10	ACCESS CONTROL	8
2.11	WORKFLOW MANAGEMENT.....	8
2.12	ADDITIONAL FUNCTIONS	8
2.13	TRANSLATION OF TERMS	9
3	CONCLUDING REMARKS.....	10
	REFERENCES	11

Abbreviations:

Abbreviations used in this document:

BoM	Bill of Material
DSS	Decision Support System
ECC	SAP’s ERP Central Component
ERP	Enterprise Resource Planning
PDKM	Product Data and Knowledge Management
PEID	Product-Embedded Information Devices
PLM	Product Lifecycle Management
PMI	PROMISE Middleware Interface
PROMISE	PROduct life cycle Management and Information tracking using Smart Embedded systems
SSL	Secure Sockets Layer
WP	PROMISE work package
XLS	MS Excel file type

1 Introduction

This Specification of System Functions of the PROMISE Product Data and Knowledge Management System (PDKM) is part of the complete PDKM definition and prototype specified in WP R9: Development of PROMISE Information Management System. An overview over this work package and the PDKM is given in deliverable DR9.1, which also provides a conceptual architecture for the system.

The functions of the PDKM specified in this document are derived from the actual and perceived needs of future system users as described in the PROMISE application scenarios with their respective deliverables to enable them to gain technical, social and economic benefit from the application of a PDKM system. These needs arise from a number of sources and include the growing awareness of environmental protection and energy conservation, the obligation to adhere to the resulting, by necessity, more complex legislative environment of international business and the increasing pressure of expected economic success.

The described system functions represent the experience of various engineering fields ranging from electronic component manufacture to white goods, automotive applications and including also potential use in guided transportation.

The functions include those already available from less efficient or uneconomic sources to those, which are currently unattainable without the use of future technology but perceived to be beneficial for the future administration of the complete product life cycle like the PDKM.

1.1 Purpose of this deliverable

This document describes data processing functions to enable the creation, recording, addendum, and modification of product data throughout the complete life cycle of both real and virtual products. Furthermore and of prime significance is the ability to derive, in an efficient and accurate manner, from the accumulated product data improved knowledge about the product's performance, application and service history as well as the product's suitability for its intended and actual usage. While deriving such knowledge is mainly the task of the integrated Decision Support System (DSS) developed by WP R8 "Methodologies for decision making for BOL, MOL, EOL" and described in respective deliverables of that work package the PDKM provides the necessary data basis for the DSS.

The resulting knowledge accumulation must be administered by the PDKM. On the one hand, this has to be done in a way such that it is easily accessible in a controlled way to authorized users and "knowledge owners". On the other hand, the needs of the integrated DSS, which is besides others intended to enable the improved knowledge to be put to targeted use for the improvement of existing products as well as significantly contributing to new product design, have to be supported.

The PDKM system – whilst initially intended for use with consumer, industrial and transportation products – will be capable in the same manner of managing the life cycle of virtual products such as simulation tools and control software for engineering applications, and data processing tools for financial and administrative applications.

Interface functions are not described in the present deliverable but discussed in detail in a separate deliverable dedicated to this subject, DR9.6c "Specification of interfaces (revised)".

While the present deliverable rather gives an overview, a more detailed description of functions directly accessible for users can be found in deliverable DR10.5 “Handbook for usage of the PDKM prototype version 2”.

1.2 Delta to deliverable DR9.3 “Specification of system functions”

The present document is based on the former deliverable DR9.3 and aims at completing it with new functions targeted for the PDKM prototype version 2 and functions planned until the end of the PROMISE project. Hence, it basically is a copy of DR9.3 with punctual changes. In order to help the reader to identify the differences between DR9.3 and this deliverable, the most significant changes are listed here:

- References to future versions of PDKM or DR9.3, namely DR9.6b, have been removed. There are not scheduled any further revisions of this deliverable. Hence, this deliverable tries to cover the most important functions of the PDKM as intended until the end of the PROMISE project. Nonetheless, minor functions not anticipated until now might be added to the PDKM. They will be described in the revision of the PDKM handbook DR10.5, deliverable DR10.6d due M40.
- References to deliverables DR9.2 and DR9.4 have been updated to their revised counterparts DR9.6a and DR9.6c. Deliverable DR10.5, available together with the present deliverable, was referenced where applicable.
- The term *Installed Base* was replaced by *Equipment Structure*, while *Installed Base Template* was removed completely.
- Sections 1.2 and 2.11 have been added (the latter causing a shift in the following section numbers).
- Sections 2.3 and 2.5 have been changed substantially.
- Section 2.2: “User-defined metadata” was added.
- Section 2.4: “Enhanced visualisation features” was added.
- Section 2.10: “Role depending appearance” was added.

Besides that there have been minor editorial changes not changing the content.

2 PDKM System Functions

In the following sub-sections, the system functionality of the PDKM is described. Part of the PDKM concept is not to implement the PDKM from scratch but to extend an existing PLM system. This approach allows focusing on application functions that are important for the application scenarios in the PROMISE context while getting additionally a whole bundle of useful "standard" functionality that might be helpful as well in the context of the application of PROMISE solutions even if they are not in the primary focus of the project like “standard” PLM functionality. This includes also that an architecture backbone like for security aspects is readily available. These additional and architecture functions are not described in detail in this deliverable.

As underlying PLM system SAP ECC was chosen. To avoid ambiguities that are often immanent to semantic terms like “product” in the following will be introduced and used the corresponding SAP-ECC-terms respectively the PDKM-specific terms if an object is not SAP-ECC- but PDKM-specific. To draw the line to the object model defined in deliverable DR9.6a “Specification of system object model (revised)”, the corresponding class-names from the object model are mentioned as well. In order to ease the reading of the document this terms are not introduced by accompanying text but simply in brackets behind the corresponding semantic term that is set in italics like “... *as-designed product* (SAP: Material; DR9.6a: AS_DESIGNED_PRODUCT) ...”.

If there is only a corresponding PDKM-specific term “SAP” is replaced by “PDKM”. Alternatively the SAP/PDKM-term is introduced directly in the respective context followed by the keyword “SAP-term” respectively “PDKM-term” in brackets. The succeeding use of the SAP/PDKM-term in the text is indicated by setting this term in italics like “*Material*” in order to distinguish it easily from semantic terms.

2.1 Web Portal

Different participants in different places all over the world can work simultaneously with the PDKM system over the internet via the web interface. Any web browser can act as a client and there is no installation on client computers needed. All user functionality described in the subsequent sections is offered in the web portal. Only the configuration of the system and administrative tasks are carried out on the backend system.

For all the objects described below the users have the possibility to create, change and view them within the limitations of their role (compare section 2.10).

To protect the communication between clients and the server technically (see also section 2.10) the communication is using the SSL (Secure Sockets Layer) security protocol i.e. the portal will be called by a URL starting with “https” like “<https://www.promise-portal.com>”.

2.2 Product and Product Structure Management

The PDKM system enables the creation and management of products and product structures. Product structures establish hierarchical relationships between components and parts of a product.

Product and product structure management in PDKM starts with the creation of an as-designed product, which contains design information about a *specific product type* (SAP: Material; DR9.6a: AS_DESIGNED_PRODUCT) and an *as-designed product structure* (SAP: Bill of Material, BoM) which covers the components and parts structure of that *Material*. There is maximally one BoM for each *Material* but a *Material* does not necessarily have an associated *BoM* if it is not required to track its structure. A *Material* is identified by its unique ID. In PDKM objects that represent parts of a *Material* are itself represented by a *Material* and are merged together to a *BoM*.

After the *Material* and/or the *BoM* is created, an *individual product* (SAP: Equipment with Serial Number, shortly Equipment; DR9.6a: PHYSICAL_PRODUCT) and its respective *as-build/as-used* (in the following shortly “as-used”) *product structure* (SAP: Equipment Structure) can be generated on this basis. An *Equipment Structure* reflects the actual bill of material (in the common sense, not the SAP-term) of the *Equipment*. There is exactly one *Equipment Structure* for each *Equipment* in the field. Parts of an *Equipment* are itself represented by *Equipments* or a *Material* with associated amount. The latter might be the preferred solution if e.g. 100 screws of a certain type shall be modelled as parts of an *Equipment*. An *Equipment* can be identified by its unique ID or its serial number (in the common sense, not the SAP-term) and the ID of the respective *Material*.

In SAP ECC, *Equipment* and *Serial Number* are two independent objects. Especially *Serial Number* is not only a number but a complex object representing a physical product with a serial number (in the common sense). For a *Serial Number* there can be activated the *Equipment-view* which allows additional operations on that *Serial Number*. By activating this view, the *Serial Number* is uniquely associated with an *Equipment*. In the PDKM-context *Equipment* and *Serial Number* are usually used together therefore in this text the term “*Equipment*” will be used instead of “*Equipment with Serial Number*”. If “*Equipment*” is meant without *Serial Number* this will be indicated explicitly or is clear from the context. When using the PDKM there might be

nevertheless the need to create *Equipments* without *Serial Numbers* e.g. when modelling wheels of a truck. This is possible and does not limit the functionality of the PDKM.

For each *Material* and *Equipment* a set of metadata is stored in the form of attributes to these objects. A respective *subset of this metadata* (SAP: Classification; DR9.6a: PROPERTY) is customisable. Besides this option, metadata can also be attached to *Materials* and *Equipments* in the form of *documents* (SAP: Document Info Record; DR9.6a: DOCUMENT_RESOURCE with associated DOCUMENT; for Document Management functions see section 2.7). PDKM enables users browsing through product structures or search for products using metadata and thereby generating different kinds of reports like where-used-reports¹. The PDKM also supports searching for products based on user-defined metadata. Metadata can be customised using the PDKM backend system by key users. The search functionality of the PROMISE Portal will display a list of additional metadata, which can be used as search criteria.

Besides the *Materials* and *Equipments*, there are *product templates* (PDKM: Equipment Template). From a system point of view, an *Equipment Template* is an *Equipment* that can be semantically recognised as template. This means especially that all operations that are possible for *Equipments* are also possible for *Equipment Templates*. This fact is not mentioned explicitly for all functions described in this document but only for those where it has special importance. *Equipments* can be instantiated either directly from a *Material* or from an *Equipment Template* while it is possible to trace back from the *Equipment* to its corresponding *Material* and eventually the respective *Equipment Template*. This intermediate layer *Equipment Template* was introduced since it is assumed that it might be more convenient and clearly structured for the user not to have to instantiate for every variant of a product a new *Material* but to instantiate an associated *Equipment Template*. An example of its application is given in section 2.4.

2.3 Configuration Management

The PDKM system is capable of tracking the history of *Equipments*, i.e. it is traceable, in which *Equipment Structures* an *Equipment* has been installed before. This information also allows the reconstruction of an *Equipment Structure* for a certain point of time in the product lifecycle.

Information regarding configuration management will be displayed in the PDKM by default. In the product details window, the product history and the component list will be presented to the user. The product history can be considered as a where used list, which gives an overview of all the places (superior components) where the respective product was or is installed. The start and end dates of the installations can also be seen in the product history. Furthermore, the component list displays all the sub-components that were/are installed in the considered product. The view on the component list can be filtered by specifying an effective date.

2.4 Field Data Functions

There has to be distinguished between

- field data in the form of single values like the current mileage of a locomotive and
- field data in the form of documents like a specific report.

With respect to the first, the PDKM system offers functionality to manage this kind of data that is gathered during the usage of *Equipments* by using *Measurement Documents*² (SAP-term; DR9.6a: FIELD_DATA). In order to manage different *types of Measurement Documents* (SAP: Characteristic; DR9.2: VALID_FD_TYPE) and thereby both allowing a classification of

¹“Report” in this context means the display of a list and summarising data in the user interface.

² A *Measurement Document* has nothing to do with a document in the common sense.

Measurement Documents and controlling the application of *Measurement Documents* a respective *Characteristic* has to be defined prior to attaching *Measurement Documents* to *Equipments*. *Characteristics* can then be assigned to *Equipments* or *Equipment Templates* by defining for those a respective *Measuring Point* (SAP-term; DR9.6a: FD_SOURCE) using this *Characteristic*. A *Measuring Point* corresponds to a sensor/an info item of a PEID (compare DR6.3 for the term info item). When *Equipments* are instantiated from an *Equipment Template*, *Measuring Points* associated to the *Equipment Template* will be made available for the *Equipment* respectively copies of these *Measuring Points*. Finally, *Measurement Documents* containing a timestamp can be stored for the *Equipment*.

In many cases, *Measuring Points* respectively the corresponding *Measurement Documents* are semantically bundled to a record of measurements. This information can also be stored since PDKM offers functionality to model groups of *Measuring Points* respectively their associated *Measurement Documents*. A group of *Measuring Points* might e.g. correspond to all sensors/info items of 1 PEID or a subset of sensors of 1 *Equipment*. The values of *Measurement Documents* of different *Measuring Points* for a given time period can be shown graphically for an *Equipment*.

The PDKM prototype version 2 includes enhanced field data visualisation functionality. Multiple field data types, for which data is stored for a certain product instance, can be visualised in a graph. In order to ease the comparison of field data values, single graphs can be transformed along both the time and the values axis.

With respect to field data documents, it is possible to associate *Document Info Records* (see section 2.7) to *Equipments* that can semantically be recognized as field data.

Both *Measurement Documents* and field data documents can also be associated to *Equipment Templates* as foreseen in DR9.6a.

2.5 Field Data Input

The PDKM system receives field data records using XML and web services technology. Field data records are usually submitted to the PDKM via the PROMISE Middleware using the PROMISE Middleware Interface (PMI) defined in a first version in deliverable DR6.5 “Interface definition and design of enterprise communication infrastructure”. There are offered user functions that allow users to regulate the PDKM – Middleware interaction like triggering or stopping the data import or sending information to PEIDs via the PMI. Details about the PDKM – Middleware interface can be found in deliverable DR9.6c.

Besides that, the user has the option to upload field data records contained in an XML-file adhering to one of two supported XML schemas. Besides the support of the PMI format, the XML structure already mentioned in deliverable DR9.3 is still supported. By using additional tools, which are not seen as part of the PDKM, and which re-format e.g. XLS-files into one of the required XML-structures, it is possible to upload large amounts of field data into the PDKM. These tools might be re-used/recycled in the PROMISE context e.g. for development of adapters connecting *Equipments* to the PROMISE Middleware.

The PDKM manages mapping information to support the PMI. The mapping information establishes the connection between the PDKM internal IDs of objects, which might not always be known to the user, and the identifiers that are used within an application scenario. After creating *Measuring Points* for a product instance, a user gets presented a list of available *Measuring Points* for this product in the PROMISE Portal. Through respective user interfaces, the mapping information can be maintained. The identifiers that are specified by the user are the ones that are

used in the PMI formatted XML file (for both import via web service and manual import). This enables the PDKM to allocate the data to import to the right objects.

2.6 Incidents Management

Information about the usage of *Equipments* can also be captured in form of *incidents* (SAP: Notification; DR9.6a: EVENT). The PDKM will offer different PROMISE-specific *Notification Types* (SAP-term) like Breakdown Events, (general) Notifications, Malfunction Reports, and Maintenance Requests. BT-LOC-Incidents and MTS-Incidents are examples of *Notification Types* that can be instantiated to *Notifications* in the PDKM. *Notifications* are attached to *Equipments* or *Equipment Templates* and can contain any *additional (field) data that describes the Notification* (SAP: Item; DR9.6a: FIELD_DATA). In order to control which *Items* can be associated to a *Notification*, *types of possible Items* (SAP: Code; DR9.6a: VALID_FD_TYPE) are grouped in *Code Groups* (SAP-term) that are associated with the respective *Notification Type*. Furthermore, *Notification Types* can be configured to allow the user to associate a *Document Info Record* (see section 2.7) to a *Notification*.

Additionally, there is the possibility to associate *Measurement Documents* via *Measuring Points* to a *Notification*. However, when analysing the test data provided so far by the PROMISE application work packages it turned out that the use of *Items* is more feasible than using *Measurement Documents*.

Notification reports can be generated in PDKM for a certain *Notification Type*, the status of a *Notification*, range of *Equipments*, range of *Materials*, or a time period. “Report” in this context means the display of a list and summarising data in the user interface. *Notifications* can be entered manually into the system but also bulk loaded from XML-formatted files.

2.7 Document Management

In the lifecycle of a product there are many documents that are related to it. PDKM offers functionality to attach to

- elements in a product structure,
- the product structure itself,
- *Notifications* of specific types,

and some other objects *Document Info Records* containing metadata about associated physical files (SAP: Original; DR9.6a: FILE). To a *Document Info Record* there are associated 0 to many *Originals*³.

As an application example *Document Info Records* containing design information might be attached to a *Material* and affect the BOL phase in the product lifecycle. Whereas *Document Info Records* containing usage information (e.g. usage profiles, analysis documents etc.) might be attached to an *Equipment* or an *Equipment Template* and affect the BOL or EOL phase in the lifecycle. Since the *Equipment* is instantiated from a *Material* design information provided via a *Document Info Record* is also accessible via the *Equipment* and information from the MOL and EOL phases stored in *Document Info Records* respectively the *Originals* associated to it is accessible via the *Material*.

For *Document Info Records* and *Originals* there is provided the “standard” document management functionality expected from a “standard” document management system e.g. version tracking, check in/check out of *Originals* etc.

³ Using a *Document Info Record* without associating an *Original* to it offers principally the opportunity to store additional metadata to the mentioned objects. This option is not used so far in the PDKM.

2.8 Knowledge Management

There has to be clearly distinguished between knowledge generation and the management of generated or otherwise provided knowledge. The first one is subject of section 2.9 while the latter is described here.

The knowledge management uses the functions described for field data (section 2.4), incidents (section 2.6), and document management (section 2.7) while the respective objects can be semantically recognised as knowledge by respective indicators that are set during the creation of these objects.

2.9 Data Analysis and Knowledge Generation Functions

Most of the analysis and knowledge generation functions are provided by the PROMISE DSS developed in WP R8. In DR9.6c is described in detail how the DSS is integrated into the PDKM. Besides that there has been realised so far only some basic but already useful functions for browsing and displaying data and knowledge (see previous sections) like displaying *Measurement Documents* in one graphic allowing to compare the data from different *Measuring Points*.

2.10 Access Control

The users of the PDKM system logging on to the web interface are authenticated by a login name and password.

According to the needs of the different application scenarios the system administrator can define complex user roles by grouping elementary user roles that grant access to creating, changing, or viewing the different objects in the system. Thereby these elementary user roles are in a reasonable granularity. E.g. it does not make sense to allow somebody to create a specific object without allowing him at the same time to view it.

The user menu appearance depends on the role of the respective user account. Users with a standard account will be able to access the standard PDKM functions as well as the DSS functions. Furthermore, there are user accounts with the role of a key user who will additionally be able to access the PDKM backend functionality. The access to the backend is also done via the PROMISE Portal. See deliverable DR10.5 for more details.

2.11 Workflow management

The PDKM covers basic workflow management functionality based on personal *Notifications*. For a certain *Equipment*, a distribution list can be defined, which includes user names of portal users. These users are notified automatically if a certain event occurs. As an example of such an event, the system supports the definition of possible minimum and maximum values for a field data type associated to an *Equipment*. If then a field data value is received, which is outside of the defined interval, the users in the distribution list will be notified with a personal *Notification*.

These *Notifications* can have a status (“outstanding”, “in process” or “complete”) and comments. This functionality demonstrates the possibility to support simple workflows within PDKM. Similarly, it can be defined if respective users shall receive a generated e-mail informing about certain events.

2.12 Additional functions

Since the PDKM is build upon an existing PLM system, the users of the PDKM have access to all functions offered by this system within the limitations of their assigned role (compare section 2.10) and possible restrictions set by the PDKM-specific functions described above.

An overview of the functions of the actually used system, SAP ECC 5.0, is available on [1]. Some of the functions mentioned above are readily provided by the underlying PLM system. They have been explicitly described because of their importance in the PROMISE context.

2.13 Translation of terms

In the previous sections, there have been used terms of the different domains:

- The semantic domain that non-expert users use when talking about the objects
- The SAP-domain that is relevant since the PDKM is implemented based on SAP software
- The object model domain that references to the respective objects defined in the object model of deliverable DR9.6a

Table 1 summarises the translation between the terms of the different domains. Thereby it should be considered that not all translations are indeed a 1:1 mapping. For the semantic domain as a matter of course, not all respective terms that correspond to the terms of the other domains are given but only some examples.

Semantically (examples)	SAP	Object model (DR9.6a)	Comment
specific product type	Material	AS_DESIGNED_PRODUCT	
as-designed product structure	Bill of Material, BoM	self-association of AS_DESIGNED_PRODUCT	
individual product	Equipment with Serial Number, shortly Equipment	PHYSICAL_PRODUCT	
as-build/as-used (shortly “as-used”) product structure	Equipment Structure	self-association of PHYSICAL_PRODUCT	
product templates	Equipment Template (PDKM-term)	AS_DESIGNED_PRODUCT	From SAP-system point of view an Equipment Template is an instance of an Equipment semantically recognizable as Template.
customisable subset of metadata of products	Classification	PROPERTY	
document metadata	Document Info Record	DOCUMENT or (depending on the context) DOCUMENT_RESOURCE with associated DOCUMENT	
document (in contrary to document metadata), (physical) file	Original	FILE	
field data in the form of a single value	Measurement Document	FIELD_DATA	
field data in the form of a document	Document Info Record	FIELD_DATA with associated DOCUMENT	

Semantically (examples)	SAP	Object model (DR9.6a)	Comment
	semantically recognisable as field data with associated Original	with associated FILE	
type of field data	Characteristic	VALID_FD_TYPE	
a sensor/an info item of a PEID	Measuring Point	FD_SOURCE	
incident, event	Notification	EVENT	
type of incident, event	Notification Type	(no correspondence)	
additional (field) data describing an incident, event	Item	FIELD_DATA	
types of possible additional (field) data describing an incident, event	Code	VALID_FD_TYPE	
classes of types of possible additional (field) data describing an incident, event	Code Groups	(no correspondence)	
Knowledge	depending on the context Notification, Measurement Document, or Document Info Record with associated Original(s) semantically recognisable as Knowledge	depending on the context FIELD_DATA or EVENT with eventual associations	Certainly all product-related data stored in the system is accessible for the user. Context-dependent the methodical access to it might result in knowledge.

Table 1: Translation between terms of different domains

3 Concluding remarks

The system functions of the PDKM have been specified. This document was based on deliverable DR9.3 that focussed on PDKM prototype version 1. DR9.3 was copied and completed, integrating the developments since version 1 and anticipating future developments planned until the end of the PROMISE project.

There is provided in different functional areas functionality to manage product-related data throughout the complete life cycle of products. This includes besides others the management of field data, incidents and knowledge thereby governing information about areas like the product's performance, application, and service history. The data is managed in a way such that the DSS developed by WP R8 and described in respective deliverables of that work package can efficiently



access this data in order to derive knowledge and support decisions/solve problems as requested by the users.

The data and knowledge administered by the PDKM is easily accessible in a controlled way to authorized users via a web portal.

References

1. Help portal for SAP ECC 5.0:
http://help.sap.com/saphelp_erp2004/helpdata/en/e1/8e51341a06084de10000009b38f83b/frameset.htm; besides that, a comprehensive, often context sensitive online help is integrated into SAP ECC.