Product Data Technology in a network:

White Paper for PDM-Integration of OEM and Supplier in the Automotive Industry

Technical Aspects of Integration Approaches
## History

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Authors</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Draft</td>
<td>D. Plischke, ProSTEP</td>
<td>Creation</td>
</tr>
<tr>
<td>1.1</td>
<td>Draft</td>
<td>M. Sachers, ProSTEP</td>
<td>Extension / Change</td>
</tr>
<tr>
<td>1.1a</td>
<td>Draft</td>
<td>M. Sachers, ProSTEP</td>
<td>Extension / Change</td>
</tr>
<tr>
<td>1.1b</td>
<td>Draft</td>
<td>M. Sachers, ProSTEP</td>
<td>Change, Removal of incomplete chapters</td>
</tr>
<tr>
<td>1.2</td>
<td>Draft</td>
<td>M. Sachers, ProSTEP</td>
<td>Extension of Use Cases in Scenario Web integration</td>
</tr>
<tr>
<td>1.3 / 1.3.1</td>
<td>Draft</td>
<td>M. Sachers, M Leber; ProSTEP</td>
<td>Extension of chapters 3.1, 5.4 and 5.5</td>
</tr>
<tr>
<td>1.3.2</td>
<td>Draft</td>
<td>M. Sachers, ProSTEP</td>
<td>Extension of use cases</td>
</tr>
<tr>
<td>1.3.3</td>
<td>Draft</td>
<td>Th. Mechlinski, ProSTEP</td>
<td>Extension of use cases</td>
</tr>
<tr>
<td>1.3.4</td>
<td>Review version</td>
<td>M. Sachers, ProSTEP</td>
<td>Extension of use cases, re-editing</td>
</tr>
<tr>
<td>1.3.4b</td>
<td>Review version</td>
<td>M. Sachers, ProSTEP</td>
<td>Re-sampling of section 5</td>
</tr>
<tr>
<td>1.4</td>
<td>Public</td>
<td>M. Sachers, ProSTEP</td>
<td>Several re-definitions of use cases</td>
</tr>
<tr>
<td>1.4.1</td>
<td>in work</td>
<td>M. Sachers, ProSTEP</td>
<td>Several re-definitions of use cases</td>
</tr>
<tr>
<td>1.5</td>
<td>Public</td>
<td>M. Sachers, ProSTEP</td>
<td>Integration of sub-scenarios</td>
</tr>
<tr>
<td>1.5.1</td>
<td>Public</td>
<td>M. Sachers, ProSTEP</td>
<td>Integration of sub-scenarios</td>
</tr>
<tr>
<td>1.6</td>
<td>Draft</td>
<td>P. Habel, ProSTEP</td>
<td>Extension of use cases</td>
</tr>
</tbody>
</table>
# Contents

**PREAMBLE** .................................................................................................................. 4

1 **BACKGROUND AND MOTIVATION** ........................................................................ 5

2 **GENERAL SCOPE** .................................................................................................... 6

3 **USE CASES** .............................................................................................................. 7

3.1 **PDM data exchange** ............................................................................................ 7

3.1.1 Export of assembly data .......................................................................................... 7

3.1.2 Import of assembly data .......................................................................................... 9

3.2 **PDM Web integration** .......................................................................................... 11

3.2.1 Scenario: Neutral PDM Web Client ........................................................................ 11

3.2.2 Scenario: PDM system synchronisation .................................................................. 54

4 **CONCEPTS OF PDM INTEGRATION** ..................................................................... 59

4.1 **File-based data exchange (asynchronous)** .......................................................... 59

4.1.1 Characteristics ....................................................................................................... 60

4.2 **PDM Web Clients** ................................................................................................. 60

4.2.1 Native Web Clients ............................................................................................... 61

4.2.2 Neutral Web Clients .............................................................................................. 61

4.2.3 Characteristics ....................................................................................................... 62

4.3 **Data sharing / Online integration** ......................................................................... 63

4.3.1 Characteristics ....................................................................................................... 63

5 **STRATEGIC RECOMMENDATIONS FOR PDM INTEGRATION** ..................... 64

5.1 **General OEM Strategy** ....................................................................................... 64

5.2 **Recommendations for suppliers** .......................................................................... 65

5.2.1 Processes and collaboration models ....................................................................... 65

5.2.2 Neutral PDM web integration ............................................................................... 70

5.2.3 Experiences from projects .................................................................................... 70

5.3 **Tools for PDM Data Exchange** ........................................................................... 72

6 **SUMMARY** .............................................................................................................. 73

7 **REFERENCES** ......................................................................................................... 74

8 **ANNEXES** ............................................................................................................... 75

8.1 **Technical selection criteria for PDM integration solutions** ................................. 75

8.2 **Cost drivers in PDM projects** ............................................................................... 75
PREAMBLE

This document is a versioned document containing requirements and results of the PDTnet project. Due to its character of a “living document” it will be extended and completed gradually according to the progress of the PDTnet project. This is why not all chapters will be completely finished in this document version.

Third parties are kindly invited to provide their comments and additional requirements to the PDTnet project (pdtnet-contact@prostep.de).
1 BACKGROUND AND MOTIVATION

Product Data Management (PDM) with the parts Geometry Management, Document Management, Product Management and Configuration is the platform for research & development (R&D) in the manufacturing industry.

In the world-wide automobile industry one of the goals is to reduce development time and costs. The complete development and simulation of a new product on a computer is the first step towards achieving this goal. The computer model of the "digital car" can be used to perform trials and calculations for which expensive prototypes are still needed today. This computer model is created by the combined effort of a number of developers using different Data Creation systems.

Due to the fact that each system defines its own data format the communication process along the computer-aided process chain today is tackled by duplicate data creation, redundant data management and expensive data conversion processes.

In order for the vision of “Digital Car” to become reality, a neutral representation of the digital car is needed, which can be processed by all the partners involved irrespective of data creation system used. This is why the car industry has been investing a great deal of time and money in the development and application of the neutral product data description in accordance with ISO 10303 (STEP) for several years. The parts of the STEP standard that are especially relevant for the automobile industry are the application protocols AP214 and AP212.

However, the standardized data model is only one precondition to allow data sharing, exchange and transfer in a heterogeneous data creation systems environment. A common understanding about processes and operations creating and processing product data as well as standardized implementation methods are required in addition.

In addition, an increasing portion of engineering development work is given out by OEMs to suppliers. This requires a stronger integration of processes, information and systems between OEM and supplier. In order to cope with this, various IT solutions were developed during the last years. They can be generally classified into four groups:

- Asynchronous file-based data exchange
- PDM integration based on Web-clients
- Synchronous data sharing
- Small-sized concepts and solutions, that are based on the techniques listed above.

The automotive companies gained experiences in the development, configuration and use of these technologies. This paper summarizes these experiences and tries to give recommendations and hints specifically for SME suppliers regarding the selection and implementation of the different technologies.
2 GENERAL SCOPE

This White Paper covers three main topics regarding the integration of PDM processes, information and systems between OEM and supplier:

- Cooperative product development of OEM and suppliers in the automotive industry
- Integration of PDM approaches in heterogeneous and homogeneous system environments
- Requirements and results out of application projects for PDM integration

The White Paper focuses on the results and experiences gained in the PDTnet project and, additionally, summarizes scenarios, concepts, solutions and general results out of the PDMI1 projects.

The exchange and integration scenarios and processes which are subject to this paper are described in section 3 “Use cases”. The use cases cover main processes identified in the PDTnet project necessary for file-based PDM data exchange and web-based PDM integration. For web-based PDM integration recommendations for the implementation of the neutral product data protocol “PDTnet XML Schema” are given in the additional document “PDTnet Implementation Guide”.

In section 4 general concepts for PDM integration of OEM and supplier are described. This includes technical aspects, advantages and disadvantages of different integration and exchange technologies.

In section 5 strategic recommendations for suppliers regarding the selection and implementation of PDM integration solutions are given. This includes recommendations or guidelines of currently running and already finished projects and experiences out of the PDTnet project.

The annex includes general technical selection criteria and cost drivers for PDM projects.

---

1 PDMI: Product Data Management based on International Standards: Name of 2 predecessor projects of PDTnet
3 USE CASES

This chapter describes the uses cases that are subject to this paper. They are categorized in the main scenarios “PDM data exchange” and “PDM Web integration”.

3.1 PDM DATA EXCHANGE

The PDM data exchange scenario is characterized by a file-based, asynchronous data exchange between two partners. The difference between the use cases is mainly the type and extend of data being exchanged. The currently documented use cases are the main use cases for file-based PDM data exchange. They will be specified and extended continuously. Particularly, detailed use cases from the currently working VDA/ProSTEP Assoc. Group “PDM Data Exchange” [VDA-DE-01] will be integrated in this scenario.

3.1.1 EXPORT OF ASSEMBLY DATA

3.1.1.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.1.1.2 Process purpose

Export of product data (meta data and geometry) of assemblies and parts by one partner to another via exchange of ENGDAT packages (STEP AP214 CC6 files, CAD files).

3.1.1.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that selects and processes product data to be exported.</td>
<td>Person</td>
</tr>
<tr>
<td>PDM System</td>
<td>Party, that provides the relevant product data and functionality for product data management. This is usually a company’s PDM system, which also can be extended by a tool, that provides extended STEP processor functionality.</td>
<td>System</td>
</tr>
<tr>
<td>Data Exchange (DE) Tool</td>
<td>System, that provides communication with a network and functionality to automatically process and pack/unpack file packages (usually ENGDAT-based).</td>
<td>System</td>
</tr>
</tbody>
</table>

3.1.1.4 Process definition

The process steps are:

1. User selects parts/documents/CAD models (using the functionality of the PDM system):
   - Selection of root/top level assembly by assembly (version) number
   - Selection of affected sub-assemblies or parts (could be controlled by a context or specific algorithm)
   - Exclusion of elements from selected set is possible

2. PDM system generates STEP PDM file (AP214 CC6):
• Passing assembly structure tree and collecting transformation matrices (if appropriate)
• Generating STEP PDM file

3. User selects addressee of data (using the DE tool or PDM tool)

4. Download of digital files from PDM system

5. DE Tool generates ENGDAT package including message abstract, STEP PDM file(s) and digital files (CAD/CC2 files, ...)

6. DE Tool initiates sending of ENGDAT message

The order of the process steps could differ depending on specific user requirements and system scenario. **Examples** for possible alternative process step orders are:

- a): 1. → 3. → 4. → 2. → 5. → 6.
- b): 3 → 1. → 2. → 4. → 5. → 6.

**3.1.1.5 Process flow diagram**

At the moment no flow diagram exists.

**3.1.1.6 Process start and end states**

Start state / precondition:
The user knows the assembly/part identifiers and digital file (CAD model) identifiers which are supposed to be exported. At least, the identifier of an assembly, which serves as an entry node, is provided. Additionally, a specific „context handle“ is known (project, change status, work order, ...) is known.

*Alternative a)*: Depending on the user environment also a top-level document ID can be the entry node to a structure.

*Alternative b)*: A top-level part and a specific configuration, which controls the way of the expansion of the tree (sub-parts, kind of documents,...), is known.

End state / post condition E1 (Success):
An ENGDAT package including the STEP PDM file and all selected digital files were successfully sent to the addressee.

End state / post condition E2 (Failure):
DE Tool delivers failure notification/report to user. The reasons can be:

• The STEP processor failed.
• The download of files from the PDM system failed.
• The DE Tool failed.

**3.1.1.7 Constraints and assertions**

Currently the number of STEP files included in one ENGDAT package is recommended to be restricted to 1 (VDA DFÜ Group). Nevertheless, the intention of BMW is to allow more than 1 STEP file per ENGDAT message. **See: Topics under discussion**

**3.1.1.8 Relevant data**

• Documents/digital files (CAD files)
• Document meta data
• Assembly/part master data
• Assembly structure data (incl. transformation data)

3.1.1.9 **Topics under discussion / Remarks**

• Currently no engineering change information is included in the STEP PDM file.
• Should more than 1 STEP file be allowed in an ENGDAT message?

3.1.1.10 **Diagrams**

At the moment no further diagrams exist.

### 3.1.2 **IMPORT OF ASSEMBLY DATA**

#### 3.1.2.1 **Owner of the use case**

This use case was defined by the Work Group 1 of the PDTnet project.

#### 3.1.2.2 **Process purpose**

Import of product data (meta data and geometry) of assemblies and parts by one partner to another via exchange of ENGDAT packages (STEP AP214 CC6 files, CAD files).

#### 3.1.2.3 **Partner role descriptions**

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that processes product data that has been imported.</td>
<td>Person</td>
</tr>
<tr>
<td>PDM System</td>
<td>Party, that provides the relevant product data and functionality for product data management. This is usually a company’s PDM system, which also can be extended by a tool, that provides extended STEP processor functionality.</td>
<td>System</td>
</tr>
<tr>
<td>Data Exchange (DE) Tool</td>
<td>System, that provides communication with a network and functionality to automatically process and pack/unpack file packages (usually ENGDAT-based).</td>
<td>System</td>
</tr>
</tbody>
</table>

#### 3.1.2.4 **Process definition**

The process steps are:

1. The DE tool receives an ENGDAT package.
2. The DE tool unpacks the ENGDAT package and stores STEP PDM and CAD files in defined directories (routing).
3. The PDM system evaluates the received STEP PDM file and displays the included data (assembly data, part data, CAD file meta data) and, optionally, generates an analysis report (comparison of existing data and data to be imported). This step can be initiated by the user or by the DE tool (if it is appropriately integrated). → see **Topics under discussion**
4. The user manually processes the data and integrates it into the database of the PDM system or, alternatively, no manual interaction is done. → see **Topics under discussion**
The DE tool can notify the user of the import process in different ways, e.g. via e-Mail, via PDM system message, a.s.o.

3.1.2.5 Process flow diagram

At the moment no flow diagram exists.

3.1.2.6 Process start and end states

Start state / precondition:

- An ENGDAT package including a STEP PDM file and one or more digital files (CAD files,...) has been received successfully. This means:
  - The ENGDAT message contains the expected correct data.
  - No inconsistencies between STEP file and references to digital files exist. → see Topics under discussion
  - User selected the mode for import (update, create, ...)

End state / post condition E1 (Success):

- The received PDM data has been successfully integrated in the PDM systems' database.
- The received CAD files have been successfully stored in the defined storage areas.
- Partial incorporation of data in the PDM system, if the user allowed it.

End state / post condition E2 (Failure):

The process results in a failure message. A failure can occur due to the following reasons:

- The ENGDAT message contains errors and can not be processed correctly.
- The STEP PDM file contains errors and can not be processed correctly (syntactically, semantically AP214 CC6/PDM Schema, ...).
- The loading process into the PDM system caused errors.

3.1.2.7 Constraints and assertions

At the moment none are defined.

3.1.2.8 Relevant data

- Documents/digital files (CAD files)
- Document meta data
- Assembly/part master data
- Assembly structure data (incl. transformation data)

3.1.2.9 Topics under discussion

- Who or which system checks, whether the STEP file and the references to digital files included in an ENGDAT message are consistent? Definition of a separate use case?
- On supplier’s side: How to handle product/document meta data, that is not managed by the own PDM system (or no PDM system exists) but that has to be re-exported to the OEM?
  - Export of version/status information for re-exported assemblies/parts could be discussed. At the moment no version/status information is used.
- BMW: The CATIA model name must not be changed by the supplier.
- On supplier’s side: How to associate product data identified by OEM identifiers to product data in the own PDM system?
• On supplier’s side: How to manage different assembly structures?

3.1.2.10 Diagrams

At the moment no further diagrams exist.

3.2 PDM Web Integration

The PDM Web integration scenario is characterized by a data access on remote systems using internet functionality and technology. The PDM Web integration does not provide a real online integration, but due to the usage of data streaming techniques and due to the possibility of an immediate reply by a system it comes near to it.

One major characteristic of product structure data is, that structures can be build from different views. These views usually are defined to fulfill the specific requirements of different application fields and life cycle stages during the product creation process. Examples for different views on product structure data are:

- Configured product structure, which includes all components of all variants of a product and the appropriate configuration and effectivity information.

- Bill of Material structure, which represents the assembly sequence of a single product and includes the quantity of each part in the product. This structure usually comprises solely parts which are already released for production.

- Engineering design structure, which includes complete versioning and release data of assemblies and parts, as well as associated document and document versioning information.

- Digital mock-up structure, which includes each instance of a part/assembly including its relative or absolute positioning information.

- “Pure” document structure, which comprises only documents and document structures.

To enable different systems to communicate this kind of data it is necessary to define and harmonize a way to uniquely represent and request different views on structures. This is one main topic of the PDTnet project.

Regarding the way of communication two sub-scenarios are defined: Using a neutral PDM Web client to access different PDM data providers (these are usually different PDM systems in different companies), and synchronizing PDM systems by a neutral communication protocol.

3.2.1 Scenario: Neutral PDM Web Client

This scenario is characterized by a Web-browser based “Neutral PDM Web Client”, which is used to access different PDM servers. All data requests are manually initiated by a user who interacts with the Neutral PDM Web Client.
3.2.1.1 Authentication / Start-up of session

3.2.1.1.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

3.2.1.1.2 Process purpose

This process allows a user to be authenticated via the PDM web client by one or more PDM server(s).

3.2.1.1.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to log in a remote PDM server. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.1.4 Process definition

This use case includes the initiation of the connection between PDM Web client and PDM server, the authentication and personalization of the user. This use case usually initiates all following communication and data transfer between a user, using the PDM web client, and a PDM server (also called “site”).

Two alternative authentication processes are possible, which can also be combined:

1. The first attempt to access a remote PDM server will automatically start the authentication process.

2. The user explicitly starts a login procedure to authenticate in one or more PDM server(s) in the beginning of a session.

The following accesses to specific PDM data will be validated within the use case “Authorization”.

3.2.1.1.5 Process flow diagram

At the moment no flow diagram exists.
3.2.1.1.6 Process start and end states

Start state S1:
- The user owns a **user name** and a **password** valid for a certain PDM server (site).
- The Web Client provides the necessary **site information** for the network connection.
- The user knows a valid **development project** to be authorized to access product data on the PDM server.
- The PDM server provides an authentication service based on user, password and session.

End state E1 (Success):
- The user is successfully logged in and, optionally, the PDM server returns a session id.

End state E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not allowed to access the PDM server (return message: “Permission denied”).
  - The PDM server itself is not available.

3.2.1.1.7 Constraints and assertions

A **development project** defines a project in which persons work together on a certain set of product data.

A development project can be a car/vehicle project, a module development project, ...

3.2.1.1.8 Relevant data

- User name, password, development project, site information (PDM server system), optional: session id

3.2.1.1.9 Diagrams

At the moment no further diagrams exist.

**3.2.1.2 Authorization**

3.2.1.2.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.2.1.2.2 Process purpose

This process validates the access rights of a specific user (designer, group, department, company) to access specific product data on a PDM server.

3.2.1.2.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to access PDM data on a remote PDM server. This could be a person, who interacts with the PDM Web Client, or a system, that triggers</td>
<td>Person / System</td>
</tr>
</tbody>
</table>
3.2.1.2.4 Process definition

This use case describes the authorization process of a user who attempts to request specific product data on a PDM server. It is used by all other use cases (e.g. when extracting product structure trees).

The exact process description is dependent on the authorization mechanisms provided by the PDM server.

3.2.1.2.5 Process flow diagram

At the moment no flow diagram exists.

3.2.1.2.6 Process start and end states

Start state S1:
- A previous authentication process was successful (e.g. by given session id).
- The PDM server provides an authorization service based on user, password and session related to specific product data elements. Additionally, the association of product data elements to a development project has to be supported.
- Specific product data that is requested by a user.

End state E1 (Success):
- The user is identified to have the appropriate rights to access the requested product data. The calling process is enabled to provide the product data to the user.

End state E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reason:
  - The user is not allowed to access the requested product data. Since it could be intended to keep the existence of the requested data completely secret, the user should not get the information “Access denied”. Instead, he should get a failure message like “Data not found”.

3.2.1.2.7 Constraints and assertions

The PDM server provides an authorization service based on user, password and session related to specific product data elements. Additionally, the association of product data elements to a development project has to be supported. The detailed mechanisms of authorizing specific users to access specific product data elements depend on the PDM server’s internal authorization features and company-specific customizing.

Specific assertions:
- The PDM server manages the association of user/development project to a specific server-internal role concept.
- The general role „owner“ is provided having all rights for the owned data objects.
• Defined access rights to all other (not owned) data objects are: View, Download, Write, Create

3.2.1.2.8 Relevant data

• User name, password, development project, optional: session id
• Requested product data

3.2.1.2.9 Topics under discussion

• The topic “Authorization and Network Security” is under discussion and is being documented in a separate PDTnet working paper.

3.2.1.2.10 Diagrams

At the moment no further diagrams exist.

3.2.1.3 Start node identification

3.2.1.3.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

3.2.1.3.2 Process purpose

Identify the start node of a product structure to enable browsing in the product structure.

3.2.1.3.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM System can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.3.4 Process definition

This use case defines the process of identifying the start node of a product structure in a PDM server. The end state / post condition of the use case is the precondition for the start of the use cases “Browsing down/up product structure data”.

The process steps are:

1. User enters ID (part number and optionally part version number) or WildCard (“*” for “all”)
2. PDM Web Client submits search request → Exception: The PDM Server does not respond
3. PDM Server receives ID or Wildcard and triggers search in PDM System → Exception: The connection between PDM Web Client and PDM Server is down
4. PDM System executes query in its database → Exceptions: Database is not available, no data found, user is not authorized to access the data, etc.
5. PDM Server returns start node and list of views
6. PDM Web Client displays list of start nodes

3.2.1.3.5 Process flow diagram

At the moment no flow diagram exists.

3.2.1.3.6 Process start and end states

Start state / precondition S1:
The user is correctly logged in, connected to the server, positively identified and authorized.
• The service is available.
• The user enters an ID („Sachnummer” etc.) or wildcard for the structure start node.

End state / post condition E1 (Success):
• List of product structure nodes including their possible views / configurations

End state / post condition E2 (Failure):
• In case of missing authorization: Exception, message: “No items found or access denied“.

3.2.1.3.7 Constraints and assertions

At the moment none are defined.

3.2.1.3.8 Relevant data

• Product structure data

3.2.1.3.9 Topics under discussion

• The user should be able to enter either internal or external part master ids (“Alias-Query”)

3.2.1.3.10 Diagrams

At the moment no further diagrams exist.

3.2.1.4 Browsing down product structure data

3.2.1.4.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

3.2.1.4.2 Process purpose

This process allows a user starting with the product structure to get a view on all product structure relevant data including document (structure) data that is relevant for this specific user or a specific project, independently of the provider of the data.

3.2.1.4.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
</tbody>
</table>
### 3.2.1.4.4 Process definition

This use case includes the browsing of product structure data **down** a product structure, basic part classification data and associated document meta data. For browsing **up** a product structure ("where used" query), a separate use case is defined.

The following requirements are defined:

- Multiple views on the product structure have to be supported, e.g. lead view, supplier’s assembly structure, spare part structure, second tier supplier’s view, etc.
- The relationship between different base classification data has to be handled (customer’s and supplier’s data).
- The assignment of structure and classification data to documents has to be consistent and browsing documents must result always in displaying identical information.
- The user defines a set of parameters (filter information), that specifies characteristics of the desired structure nodes in detail. Filtering the data will be defined as a separate use case "PDM filter".
- Browsing in different PDM server systems has to be supported. This means, the change of a server site has to be possible ("**Multi-site support**") when the user selects a structure node, which links to a supplied item provided by another PDM server. This enables the user to browse into a sub-structure of the development partner (e.g. OEM user browses into sub-structure of supplier or vice versa) and to see the information consistently in one single structure tree. The concept for this mechanism is the following:
  - Reference tables connecting the OEM part identifiers to the supplier part identifiers ("alias identifiers") are managed by the PDM servers, containing for each exchange node:
    - Own part id (item_version to be supported)
    - Corresponding alias id on PDM server of partner
    - Unique identifier for partner PDM server site: harmonized organization ID, e.g. "bmw.de"
  - An additional reference table for the association of organisation id and URL (server site connection) is provided on the PDM Web Client site

The process steps are:

1. **PDM Web Client sends a query for substructure specified by the user to the PDM Server**
   a. In case of the structure node being a "supplied item", i.e., the selected structure node represents an alias identifier:
      - Client retrieves alias site connection information (URL) from reference table
      - Client asks user for password for alias site (only in case of first request to this site)
      - Client performs Login, Start node query on alias server site using current development project

   **Steps repeated by PDM Server for each product structure node in the scope of the query:**
   2. Check authorization regarding requested data → Exception: Access denied (PDM Server)
   3. Collect requested data within PDM Server

   **End of repeated steps.**

4. **PDM Server sends data to PDM Web Client**
5. Display structure and items in PDM Web Client

3.2.1.4.5 Process flow diagram

3.2.1.4.6 Process start and end states

Start state / precondition S1:
- A specific development project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project's life time. These items are identified by identifiers.
- The end state / post condition of use case „Start node identification“ or one of the children of the start node.
- The user is correctly logged in and authorized to access the requested information.
- The level of depth down the start node / current node is defined (default: 1 level down the current node).
- The necessary filter information is defined, i.e., the result of the use case “PDM filter” is provided.

End state / post condition E1 (Success):
- The process results in a **filtered** list or a structure tree containing at least the identifiers of product data items, and additional information about the items (e.g. URLs to documents or additional item information to be downloaded).

End state / post condition E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the data.
  - The requested data is not available on the PDM server.

3.2.1.4.7 Constraints and assertions

- If process step 2 leads to an exception regarding a specific structure node, the whole process must continue. The structure node affected by the exception is not included in the collected data set.

3.2.1.4.8 Relevant data

- Product structure data
- Basic part classification data
- Document meta data
3.2.1.4.9 Diagrams

At the moment no further diagrams exist.

3.2.1.5 **Browsing up product structure data**

3.2.1.5.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.2.1.5.2 Process purpose

This process allows a user starting with a specific product structure node to get a view on all relevant product structure nodes in which this specific node is included (“Where used” query). For browsing down a product structure, a separate use case is defined.

3.2.1.5.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.5.4 Process definition

This use case includes the browsing of product structure data up a product structure („where used“ query).

The following requirements are defined:

- Multiple views on the product structure have to be supported, e.g. lead view, supplier’s assembly structure, spare part structure, second tier supplier’s view, etc.
- The user defines a set of parameters (filter information), that specifies characteristics of the desired structure nodes in detail.

The process steps are:

1. PDM Web Client sends a query for „where used“ nodes specified by the user to the PDM Server
   *Steps repeated by PDM Server for each product structure node in the scope of the query:*
   2. Check authorization regarding requested data → Exception: Access denied (PDM Server)
   3. Collect requested data within PDM Server

*End of repeated steps.*

4. PDM Server sends data to PDM Web Client
5. Display structure and items in PDM Web Client. The way of presentation and needed interaction have to be defined by the application projects.

3.2.1.5.5 Process flow diagram
At the moment no flow diagram exists.

3.2.1.5.6 Process start and end states

Start state / precondition S1:

- A specific engineering development project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project’s life time. These items are identified by identifiers.
- The end state / post condition of use case „Start node identification“ or one of the children of the start node, that means item or item_version. Item_version is optionally in order to enable the access of versioning information starting from the part number. Additionally, the single_instance can be identified (maybe by user interaction). This is “nice to have“ in general, but required as precondition for a “Search in design space“ functionality.
- The user is correctly logged in and authorized to access the requested information.
- The level of depth up the start node / current node is defined and restricted to direct parent or root node (default: direct parent node).
- The necessary filter information is defined, i.e., the result of the use case “PDM filter“ is provided.

End state / post condition E1 (Success):
- The process results in a filtered list or a structure tree containing only identifiers of product data items (root nodes or direct parent nodes). Only structure nodes which the user is authorized to see are included.

End state / post condition E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the data.
  - The requested data is not available on the PDM server.

3.2.1.5.7 Constraints and assertions

- Whenever the PDM System is providing a single_instance concept, the start node used may be the single_instance. If the single_instance is used, there is no necessity for repeating process steps 2 and 3. This statement needs to be evaluated!
- The level of depth up the start node / current node is defined and restricted to direct parent or root node (default: direct parent node).
- Exactly one root node exists for one development project.
- Need of unique filter, that displays the root node only once.
- Within the Web Client GUI the change to one of the resulting development project (in case of a result list containing root nodes) should be possible.

3.2.1.5.8 Relevant data

- Product structure data

3.2.1.5.9 Diagrams

At the moment no further diagrams exist.

3.2.1.6 Download of product data – General remarks

The following uses cases dealing with the download of product data has to be described under consideration of two main criteria:
1. **What** product data is to be downloaded?
   a. Download of a single digital file (simple user interaction)
   b. Download of a set of digital files (or a single digital file)
   c. Download of meta data including structures

2. **How** is the product data to be downloaded?
   a. Using online download: via HTTP, only for available documents – no conversion functionality provided
   b. Using offline download (e.g. via OFTP)

Due to these distinctions the general use case “Download of product data” is divided into several use cases, which are described in the following sections.

### 3.2.1.7 Initiation of an offline download

3.2.1.7.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.2.1.7.2 Process purpose

This process initiates an asynchronous transfer of documents as digital files. The parameters for the transfer process are predefined.

3.2.1.7.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that initiates the download of product data. This could be a person, who</td>
<td>Person / System</td>
</tr>
<tr>
<td></td>
<td>interacts with the PDM Web Client (Design engineer at OEM or at the supplier</td>
<td></td>
</tr>
<tr>
<td></td>
<td>site), or a system, that triggers the PDM Web Client.</td>
<td></td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>system that acts as a server. The PDM system can be extended by a Web Server to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>build the complete PDM Server.</td>
<td></td>
</tr>
<tr>
<td>EDI-Tool</td>
<td>System, that provides functionality for asynchronous file transfer. Usually, the</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>transfer via OFTP/ENGDAT is supported.</td>
<td></td>
</tr>
</tbody>
</table>

3.2.1.7.4 Process definition

The process steps are as follows:

1. User sends request for download.
2. PDM server processes user data (user ID, password, development project, list of files).
3. PDM server initiates download process (to EDI tool).
4. Start of the transfer process.
5. WebClient presents the "process is started/initiated" information.
Errors are handled externally of the Web Client (EDI Tool, PDM system).

3.2.1.7.5 Process flow diagram

The general context is shown in the following diagram.

3.2.1.7.6 Process start and end states

Start state S1:
- Successful authentication / authorization
- The development project is available
- Download scenario is already chosen (Download of Single Documents, Set of Documents, ...) and the files are selected.
- The transfer process is already defined. Only users for whom predefined transfer processes exist, are allowed to use function.
- The PDM server interface provides pull- and/or push-mechanism

End state E1 (Success):
The process “ends” with a message, which is displayed by the Web Client to the user:
  a) Process started successfully (online: Client interface)
  b) See Remarks

End state E2 (Failure):
The process results in a failure message. A failure can occur due to the following reasons:
- No development project defined.
- The user is not authorized to access the data.
- Functionality is not supported for this object type or by the PDM server interface.

Other errors that can occur during the offline transfer process are handled externally of the Web Client, usually by the EDI Tool and/or the PDM system (e.g. via e-mail to the user).
3.2.1.7.7 Constraints and assertions

- Only one PDM interface is accessed.
- Only one transfer process is supported for the current user.
- Parameters for the transfer process (file conversion, formats, ...) have to be predefined within the EDI Tool.
- The file names are generated by server/system specific rules.

3.2.1.7.8 Relevant data

- Document meta data
- Document data (digital file)

3.2.1.7.9 Remarks

- Depending on the functionality of EDI tool and PDM system a message, that the exchange process finished successfully, could be sent to the user (e.g. via e-mail by the sending EDI Tool).
- A check-in/check-out mechanism for downloaded documents / digital files may be provided by the PDM server, but this is not considered or required by this use case.

3.2.1.8 Download of a set of digital files (or a single digital file)

3.2.1.8.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.2.1.8.2 Process purpose

This use case describes the download functionality of a set of digital files or a single digital file to the storage of a local computer.

There exist two ways of defining the files that have to be downloaded by user interactions with the PDM Web Client:

1. Clicking on the root object (item, item_version, ddid, document – any node in the structure) identifies all documents that belong to this structure for download.

2. Realizing „Shopping Functionality”: The user browses through the structure and identifies one or more single files for download.

A combination of both use cases needs also to be supported.

3.2.1.8.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between</td>
<td>System</td>
</tr>
</tbody>
</table>
3.2.1.8.4 Non-functional requirements

The following requirements with respect to the design of the PDM Web Client GUI are defined:

- On the left side of the PDM Web Client GUI the normal structure tree is used. The result of the use case is a list of documents on the right side.
- Documents without authorization for this user should be marked with a special color on the right side.
- Documents that are already used by others have to be marked with a special color. Despite this situation the user is able to mark the document for download, because an offline download could support this functionality.
- To any documents belongs a radio button for excluding the file from the download list.
- The download button (online and / or offline) is placed below the document list on the right side.
- Below the list a calculation field is placed, that provides the sum of the file size (file_size – properties) of all selected files in the list (file_size has to be provided by the server interface).

3.2.1.8.5 Process definition

The standard process consists of the following steps (the steps directly refer to elements of the user interface of the PDM Web Client):

1. The context menu (“right mouse click”) is used to initiate the „Document Selection for Download“ or “Add to shopping list” functionality:
   a. Clicking on a structure node object (item_version or document) identifies the root object of which the digital files of the underlying document structure shall be downloaded. The following steps are defined:
      - If the user selects an item_version: User specifies filter parameters for the document type (e.g. “2D geometry”, “3D geometry”, “DMU”, “FEM”, ...) to be downloaded. By this, the appropriate document structure is determined.
      - User specifies filter parameters for the digital file type (e.g. “STEP”, “CATIA native”, “TIFF”, “VRML”, ...) to be downloaded.
      - Resulting digital files are delivered by PDM server and added to the shopping list of the Client.
   b. Using „Shopping Functionality“: The user browses through the structure and identifies one or more single files for download.

2. The result list is shown in the right frame of the PDM Web Client GUI (“Shopping list”).

3. The status of selected files (i.e. “no authorization for download”, “file is used”) will be marked with special colors.

4. Radio buttons, that belong to digital files, can be used to exclude the file from download (the default setting is “file included”).
5. The PDM Web Client calculates the total amount of file size for all selected files.

6. The User starts the download by Clicking the Online or Offline Download button.

**Online Download:**

7. The PDM Web Client sends a query to the PDM Server.

8. The PDM Server sends the requested digital file data to the PDM Web Client.

9. The PDM Web Client receives the digital files and stores them in the local file system.

10. A notification is sent to the User (in case of success and in case of failure).

**Offline Download (see also “Initiation of an Offline Download”):**

7. The PDM Web Client sends a query to PDM server interface or to the involved EDI-Tool → Input to use case “Initiation of an Offline Download”.

8. For the file export from the PDM Vault a copy of the document should be created, no file locking mechanism (for parallel use by other users) should be implemented. The export could be triggered by the PDM Server or by the EDI-Tool.

9. A Client notification is created by the EDI–Tool.

3.2.1.8.6 Process flow diagram

At the moment no flow diagram exists.

3.2.1.8.7 Process start and end states

Start state S1:
- The user has been successfully authenticated.
- The user is authorized to know that the digital files exists.
- The user has got structure information (in the left frame of the PDM Web Client GUI).
- The final trigger is the selection in the context sensitive menue („Document Selection for Download”) that belongs to items or documents.

End state E1 (Success):
- A list of documents that were selected shown in the right frame of the PDM Web Client GUI).
- A trigger for the PDM Server interface for checkout – functionality.
- A trigger for data exchange tools (offline e.g. DXM, RVS, SWAN).
- Offline Download: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- Online Download: A notification for the User, if the download is finished (with success or not).
- The digital files, that have been specified by the user for download, are stored on the local storage.

End state E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the PDM server.
  - The user is not authorized to download the digital file.
• The PDM server itself is not available.
• **Offline** Download: Triggering the EDI-Tool failed.
• Export (checkout) functionality failed (digital file doesn’t exist, the file is already used by an other user).
• **Online** Download: Not sufficient disc space for storing the files.
• **Online** Download (maybe **Offline** Download, depending on the used EDI tool): The document list shows those files that caused the problem with a special color (if this information is available).

3.2.1.8.8 Constraints and assertions

• The downloaded files are always compressed as a package, even if there is only one single file. This may not be needed for some offline transfer processes.

• The file names as well as the package names are generated by server/system specific rules.

3.2.1.8.9 Relevant data

• CAD native files (e.g. CATIA models)

• Other digital data formats (TIFF, PDF, Word, PPT ...)

• Metadata contained in a Part21-file (CC6, CC8, PDM Schema, ...), that is **already existing and available for download** (this is a difference to the use case „Download of Metadata including structures“)

3.2.1.8.10 Topics under discussion / Remarks

• A User could have the authorization for reading document master data, but isn’t allowed to download the document.

• Duplicate names of digital files are not allowed (if such duplication occurs in an application project it needs to be solved).

• If the file is used by another user two possibilities of system reaction are possible:
  1.) **Offline** Download: The EDI-Tool is waiting for the export until the file becomes unlocked.
  2.) **Online** Download: The process is creating a failure notification.

3.2.1.9 Download of a single digital file (**simple user interaction**)  

3.2.1.9.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

3.2.1.9.2 Process purpose

This process allows a user to download a single specific digital file (geometry file, TIFF, ...) from a remote PDM server to a local storage.

The download also includes the viewing of digital files, as far as a viewing tool is automatically started on the user side after the download process has finished. This process is called "simple viewing".
3.2.1.9.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.9.4 Process definition

This use case includes the identification of a single digital file to be downloaded, the start, the monitoring of the progress and the check of the success of the data transport from a PDM server to a local storage.

The process steps are as follows:

1. The user identifies the digital file to be downloaded from the PDM server.
2. The User starts the download by choosing the **Online** or **Offline** Download entry in the right click menu.

**Online Download:**

3. The PDM Web Client sends a query to the PDM Server.
4. The PDM Server sends the requested digital file data to the PDM Web Client.
5. The PDM Web Client receives the digital file and displays it directly, opens an external application to display it or let the user store it in the local file system.
6. A notification is sent to the User (in case of success and in case of failure).

**Offline Download (see also “Initiation of an Offline Download”):**

3. The PDM Web Client sends a query to PDM server interface or to the involved EDI-Tool → Input to use case “Initiation of an Offline Download”.
4. For the file export from the PDM Vault a copy of the document should be created, no file locking mechanism (for parallel use by other users) should be implemented. The export could be triggered by the PDM Server or by the EDI-Tool.
5. A Client notification is created by the EDI–Tool.

3.2.1.9.5 Process flow diagram

At the moment no flow diagram exists.

3.2.1.9.6 Process start and end states

Start state S1:
• The user has been successfully authenticated.
• The user is authorized to know that the digital file exists.
• The user has got a list or a structure tree containing at least the identifier of the digital file and an appropriate URL.
• The kind of the access (viewing, changing) is specified. Currently only viewing functionality is considered.
• The final trigger is the selection in the context sensitive menu (“Download selected file online/offline”) that belongs to a selected single digital file.

End state E1 (Success):
• Offline Download: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
• Online Download: A notification for the User, if the download is finished (with success or not).
• The digital file, that has been specified by the user for download, is opened and displayed or stored on the local storage.

End state E2 (Failure):
• The process results in a failure message. A failure can occur due to the following reasons:
  • The user is not authorized to access the PDM server.
  • The user is not authorized to download the digital file.
  • The requested digital file is not available on the PDM server.
  • The PDM server itself is not available.
  • Offline Download: Triggering the EDI-Tool failed.
  • Export (checkout) functionality failed (digital file doesn’t exist, the file is already used by another user).
  • Online Download: Not sufficient disc space for storing the files.

3.2.1.9.7 Constraints and assertions
• The downloaded file is always uncompressed if it is sent online. Then the file can be opened directly and maybe viewed using a client plug in or an external application. Compression is only allowed if an offline transfer process implies a package mechanism.
• The file name is generated by server/system specific rules.

3.2.1.9.8 Relevant data
• Document meta data
• Document data (digital file)

3.2.1.10 Download of meta data including structures

3.2.1.10.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

3.2.1.10.2 Process purpose

This use case allows the user to identify meta data including structures that he wants to store in a local file system, or that he wants to import into an own PDM system.

The format of the transferred data differs:
Online download: The data is transmitted as an XML stream (SOAP message response). File representations are not supported in this case.

Offline download: The data is sent as an file within the download package. It can be a STEP AP214 Part21 or a PDTnet XML file, which is specified in the server configuration and considers requirements at target side.

If the detail level covers digital documents the download of these files will be initiated. The download of existing Part 21 files is not covered by this use case either. For this, see use case “Download of a set of digital files (or a single digital file)”. If the data is sent offline, the files may be added to the download package, which is specified in the server configuration and considers requirements at target side.

This functionality covers the access of multiple PDM Server Interfaces. For this, two possibilities exist:

1. The user has access to the PDM data of his direct (!) partners. This is covered by the use cases.
2. All other alternate possibilities are managed by the PDM server interface (e.g. data in a 2nd-tier supplier’s PDM system).

3.2.1.10.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server.</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.10.4 Non-functional requirements

The following requirements with respect to the design of the PDM Web Client GUI are defined:

- The level of detail ("configuration") can be defined depending on the application project. The technology for defining this configuration is not defined yet.
- The approval status of the relevant data has to be managed by the PDM server interface via authentication and authorization use cases.
- The user is not able to exclude single objects that belong to the tree defined by the start node.
- An additional use case is needed: „PDM Filter“. This use case enables the user to define some special properties that restrict the following amount of managed data.

3.2.1.10.5 Process definition

The standard process consists of the following steps (the steps directly refer to elements of the user interface of the PDM Web Client):

1. Using the context menu ("right mouse click") for starting the use case. The User can use this menu only for items and documents in order to be AP214 compliant in any cases.
2. By identifying the menu button “download of metadata” a sub-menu appears that provides all available levels of detail (called “configurations”): download of part master data, download of part and document master data, ...

3. The user identifies the wished level of detail using the sub-menu.

4. If the user defined to download structure information the next sub-menu appears: “Level of structure depth”.

5. In the right frame a list of items appears that were defined for the download process. The user is able to use a scrollbar for browsing through the list.

Optionally: If the download information was not already received by the client the following steps will be performed:

5a. The client is calling the PDM server using a specified query.

5b. The server generates the product data and sends the XML stream to the client interface.

Mandatory:

6. The User starts the download by choosing the Online or Offline Download entry in the right click menu.

Online Download:

7. The PDM Web Client sends a query to the PDM Server.

8. The PDM Server sends the requested data as a XML stream to the PDM Web Client.

9. The client takes the XML stream and:

9a. calls the “Upload Query” to the second PDM system or

9b. writes an XML file.

Offline Download (see also “Initiation of an Offline Download”):

7. The PDM Web Client sends a query to PDM server interface or to the involved EDI-Tool → Input to use case “Initiation of an Offline Download”.

8. A Client notification is created by the EDI–Tool.

3.2.1.10.6 Process flow diagram

At the moment no flow diagram exists.

3.2.1.10.7 Process start and end states

Start state S1:
- Successful results of Authorization and Browsing use cases.

End state E1 (Success):
- Offline Download: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
• **Online Download**: A notification for the User, if the download is finished (with success or not).
• The selected meta data including structures is stored in a XML file on a local computer (file system), or generated as XML stream as input for the Upload use case.

**End state E2 (Failure):**
• The process results in a failure message. A failure can occur due to the following reasons:
  • The user is not authorized to access the PDM server.
  • The PDM server interface detected a problem.
  • The user is not authorized to download the requested data.
  • The PDM server itself is not available.
  • **Offline Download**: Triggering the EDI-Tool failed.
  • **Online Download**: Not sufficient disc space for storing the file.

3.2.1.10.8 Relevant data
• All product data (part master, document master etc.)

3.2.1.10.9 Topics under discussion / Remarks
• This download use case ends by creating a XML file or a XML stream. This data can be re-used by Upload Use Cases.
• Definition of „configurations“: Should they be based on XSLT?

**3.2.1.11 Generic object query**

3.2.1.11.1 Owner of the use case
This use case was defined by the Work Group 2 of the PDTnet project.

3.2.1.11.2 Process purpose
This use case allows a user to generically access objects (items, documents) as result of a specified filter condition. Feasible filter parameters and the functionality for the collection and provision of these objects have to be provided by the PDM server. Therefore, this generic use case can be specialized to further detailed use cases. Examples for detailed use cases are:

• Find all parts contained in a design space by providing bounding box parameters.
• Find heat sensitive parts by providing temperature parameters.

3.2.1.11.3 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to request information This could be a person, who interacts</td>
<td>Person / System</td>
</tr>
<tr>
<td></td>
<td>with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td></td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is</td>
<td>System</td>
</tr>
</tbody>
</table>
usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.

3.2.11.4 Process definition

The process steps are:

1. User chooses the intended (and provided) functionality (specialized query).
2. User defines a development project or uses the existing one.
3. Web Client displays the parameter names, that have to be provided to filter out the correct data within the PDM server, according to the chosen functionality (see 1.).
4. User provides required parameter values (objects properties, bounding box information, ...) and initiates query to PDM server interface (single PDM Interface).
5. PDM System is processing the query that results in an object list.
6. Object list is displayed within the Web Client.

3.2.11.5 Process flow diagram

The main mechanism for “Generic object query” is shown in the following diagram. For more details see the specialized use cases.

3.2.11.6 Process start and end states

Start state S1:
- The authentication and authorization of the user was successful.
- A valid development project is existing.
• The available specialized types of object queries related to specific objects have been previously submitted by the PDM server (see use case “Start-up of session”).

End state E1 (Success):
• List of objects that were requested according to the specialized query and filter parameters. Example for specialized query “Search in design space”: All parts contained in the defined design space as a list of items.

End state E2 (Failure):
• The process results in a failure message. A failure can occur due to the following reasons:
  • No development project defined
  • The user is not authorized to access the data (see also use case “Authorization”).
  • The requested data is not available on the PDM server.
  • Functionality is not supported for this object type.

3.2.1.11.7 Constraints and assertions

Only one single PDM server is accessed. A generic object query that is sent simultaneously to more than one PDM server is not supported.

3.2.1.11.8 Relevant data

• Product structure data
• Basic part classification data
• Document meta data
• Document data

3.2.1.11.9 Diagrams

UML diagrams are provided for the specialized use cases.

3.2.1.12 Search in design space

This use case is a specialization of the use case “Generic object query”.

3.2.1.12.1 Process purpose

Purpose of the “Search in design space” process is to query all parts which are located in the neighborhood of a given part. This use case allows a designer at the supplier site to search for parts which are positioned in a certain area around a specified part. The calculation of the neighborhood relation of parts will be done by using the “bounding boxes” of the parts. The user should be able to “blow up” the bounding box around a part in order to get all parts in a certain distance of the given part.

3.2.1.12.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server.</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is</td>
<td>System</td>
</tr>
</tbody>
</table>
3.2.1.12.3 Process definition

The process could be seen as a query in which the query parameters do not exist as discrete PDM data in the PDM system. Actually, the criteria for the evaluation of the result set is the geometrical relation between the given part and all other parts in a given assembly. For example, the designer has to modify the design of the oil pump of a car. He needs to know which parts are located near to the pump to be able to check whether the modified pump fits into the space left for this device. With the search described here, he can find those parts easily. This use case would probably only be relevant for the OEM side of the PDTnet project.

The following requirements are defined:

- The parts found during the search are displayed in form of a “virtual container” which contains all parts meeting the design space criteria. The virtual container is an assembly which is only created temporarily and which does not represents any form of a real assembly. It is only meant as a set of objects and therefore can be displayed as an assembly with one and only one level.
- It should be possible to combine different search criteria (search in design space, search by defining PDM data filters). For example, all temperature sensitive parts in a certain distance of a hot part have to be found by the query.
- In order to ensure the clearness of visualization, the formerly displayed structures should be made available by means of a “Pull down list” or by “Tabs” which allow to go directly to the assigned structure display.
- The resulting set of items should allow to perform a download (online or offline) on certain items selectable by the user
- The user should optionally be able to define an assembly (“Start node”) in which the parts to find are contained. For example, all parts in an combustion engine should be found.
- Another option is to enter the depth of search, the levels of deepness in an assembly.

3.2.1.12.4 Process flow diagram
3.2.1.12.5 Process start and end states

Start state / precondition S1:

- A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project’s life time. These items are identified by identifiers.
- The end state / post condition of use case „Start node identification“ or one of the children of the start node, that means an item
- The user is correctly logged in and authorized to access the requested information.
- The necessary filter information is defined (see use case “Generic object query”).

End state / post condition E1 (Success):

- The process results in a virtual container (see 3.2.1.12) containing all the accessible parts found during the query. The number of parts found is displayed.
- The virtual container contains the transformation matrices of the parts in relation to the car origin.
• If no parts resp. no accessible parts were found, an empty virtual container is presented. The number of parts found is displayed, in this case it is 0.

End state / post condition E2 (Failure):
• The process results in a failure message. A failure can occur due to the following reasons:
  • The selected part contains no geometry. Therefore, there is no possibility to find any parts in the neighborhood of the part. This should be reported by the message “Part contains no geometry”.

3.2.1.12.6 Constraints and assertions
• The selected part has to contain any geometry as a base for the query.

3.2.1.12.7 Relevant data
• Product structure data

3.2.1.12.8 Diagrams

3.2.1.13 Upload of product data – General remarks

According to the “Download” use cases the following uses cases dealing with the upload of product data have to be described under consideration of two main criteria:

1. **What** product data is to be uploaded?
   a. Upload of a single digital file (simple user interaction)
   b. Upload of a set of digital files (or a single digital file)
   c. Upload of meta data including structures

2. **How** is the product data to be uploaded?
   a. Using online upload: via HTTP, only for available documents – no conversion functionality provided
   b. Using offline upload (e.g. via OFTP)

Due to these distinctions the general use case “Upload of product data” is divided into several use cases, which are described in the following sections.

3.2.1.14 Upload of a set of digital files (or a single digital file)

3.2.1.14.1 Process purpose

This process allows a user to upload files which were created or changed on a local storage to a remote PDM server.

3.2.1.14.2 Process definition

This use case corresponds mainly to use case “Download of a set of digital files (or a single digital file)”. Additionally, it requires two functionalities:

• Identification of correct structure nodes for the integration of uploaded data.
• Creation/change of structures and/or structure nodes, if appropriate. This functionality is closely related to the underlying access authorization concept. Due to the variety of PDM
system-specific access authorization architectures this topic is closely depending on the PDM system functionality.

3.2.1.14.3 Process flow diagram

At the moment no flow diagram exists.

3.2.1.14.4 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to store PDM data on a remote PDM server. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.14.5 Process start and end states

Start state S1:
- The user has got files (compressed as a package) stored on his local file system to be uploaded.
- The user knows the correct structure nodes in the database of the PDM server for the integration of the data.

End state E1 (Success):
- **Offline Upload**: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- **Online Upload**: A notification for the User, if the upload is finished (with success or not). The displayed target structure is refreshed on the screen.
- The files, that had been specified by the user for upload, are stored on the remote PDM server and attached to the target structure. Maybe some new structure nodes were created to attach the files to.

End state E2 (Failure):
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the PDM server.
  - The user is not authorized to upload the digital files.
  - The user is not authorized to create needed structure nodes.
  - The server can’t create needed structure nodes with default values.
  - The specified data could not be integrated in the database of the PDM server (e.g. the correct structure nodes for data integration could not be identified).
  - The PDM server itself is not available.
  - **Offline Upload**: Triggering the EDI-Tool failed.

3.2.1.14.6 Constraints and assertions

- The uploaded files are always compressed as a package, even if there is only one single file. This may not be needed for some offline transfer processes.
The files can be assigned to one or more target elements. If all uploaded files would be assigned to one single element, this will be selected within a SOAP message parameter. If there are more complex relations between files and elements, a PDTnet schema is sent to the server containing all needed data. In case of an offline transfer, it can be replaced by a STEP AP214 Part21 file, which is specified in the server configuration and considers requirements at target side.

The target element to assign an uploaded file to can be of type “Item_version” or “Document_version”. In case of a “Document_version” the file can be assigned directly. If an “Item_version” is selected, the server has to create a document with default values to assign the file to. If any creation is not possible, the action fails and the user is notified.

Any directives/parameters for the upload process are stored at server side.

3.2.1.14.7 Relevant data

- Product structure data
- Basic part classification data
- Document meta data
- Document data

3.2.1.14.8 Topics under discussion / Remarks

- In case of an online transfer, it is not decided yet, if the PDTnet schema (containing complex relationships between files and structure elements) is sent as a XML stream within the SOAP message or as a XML file within the file package.

3.2.1.15 Upload of a single digital file (simple user interaction)

3.2.1.15.1 Process purpose

This process allows a user to upload a single file which were created or changed on a local storage to a remote PDM server.

3.2.1.15.2 Process definition

This use case corresponds mainly to use case “Download of a single digital file”. Additionally, it requires two functionalities:

- Identification of the correct structure node for the integration of uploaded data.
- Creation/change of structures and/or structure nodes, if appropriate. This functionality is closely related to the underlying access authorization concept. Due to the variety of PDM system-specific access authorization architectures this topic is closely depending on the PDM system functionality.

3.2.1.15.3 Process flow diagram

At the moment no flow diagram exists.

3.2.1.15.4 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to store PDM data on a remote PDM server. This could be a person, who interacts</td>
<td>Person / System</td>
</tr>
</tbody>
</table>
with the PDM Web Client, or a system, that triggers the PDM Web Client.

<table>
<thead>
<tr>
<th>PDM Web Client</th>
<th>System, that provides the communication between user and PDM server</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

### 3.2.1.15.5 Process start and end states

**Start state S1:**
- The user has got a single file stored on his local file system to be uploaded.
- The user knows the correct structure node in the database of the PDM server for the integration of the data.

**End state E1 (Success):**
- **Offline Upload:** A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- **Online Upload:** A notification for the User, if the upload is finished (with success or not). The displayed target structure is refreshed on the screen.
- The file, that had been specified by the user for upload, is stored on the remote PDM server and attached to the target structure. Maybe some new structure node were created to attach the file to.

**End state E2 (Failure):**
- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the PDM server.
  - The user is not authorized to upload the digital file.
  - The user is not authorized to create needed structure nodes.
  - The server can’t create needed structure nodes with default values.
  - The specified data could not be integrated in the database of the PDM server (e.g. the correct structure node for data integration could not be identified).
  - The PDM server itself is not available.
  - **Offline Upload:** Triggering the EDI-Tool failed.

### 3.2.1.15.6 Constraints and assertions

- The uploaded file is always uncompressed. Compression is only allowed if an offline transfer process implies a packaging mechanism.

- The target element to assign an uploaded file to can be of type “Item_version” or “Document_version”. In case of a “Document_version” the file can be assigned directly. If an “Item_version” is selected, the server has to create a document with default values to assign the file to. If any creation is not possible, the action fails and the user is notified.

- Any directives/parameters for the upload process are stored at server side.

### 3.2.1.15.7 Relevant data

- Product structure data
- Document meta data
- Document data (digital file)
3.2.1.16 Upload of meta data including structures

3.2.1.16.1 Process purpose

This process allows a user to upload meta data including structures to a remote PDM server. This data was created or changed on a local storage or is the result of a download process.

3.2.1.16.2 Process definition

This use case corresponds mainly to use case “Download of meta data including structures”. Additionally, it requires two functionalities:

- Identification of correct structure nodes for the integration of uploaded data.
- Creation/change of structures and/or structure nodes, if appropriate. This functionality is closely related to the underlying access authorization concept. Due to the variety of PDM system-specific access authorization architectures this topic is closely depending on the PDM system functionality.

3.2.1.16.3 Process flow diagram

At the moment no flow diagram exists.

3.2.1.16.4 Partner role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that wishes to store PDM data on a remote PDM server. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server. The PDM system can be extended by a Web Server to build the complete PDM Server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.16.5 Process start and end states

Start state S1:
- The user has got data stored on his local file system or stored temporarily as a result of a download process.
- The user knows the correct structure nodes in the database of the PDM server for the integration of the data.

End state E1 (Success):
- Offline Upload: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- Online Upload: A notification for the User, if the upload is finished (with success or not). The displayed target structure is refreshed on the screen.
- The data, that had been specified by the user for upload, is stored on the remote PDM server and integrated into the target structure.

End state E2 (Failure):
The process results in a failure message. A failure can occur due to the following reasons:

- The user is not authorized to access the PDM server.
- The user is not authorized to upload the data.
- The specified data could not be integrated in the database of the PDM server (e.g. the correct structure nodes for data integration could not be identified).
- The PDM server itself is not available.
- Offline Upload: Triggering the EDI-Tool failed.

### 3.2.1.16.6 Constraints and assertions

The new structure is sent as PDTnet XML to the server. The data can be assigned to one or more target elements. If the whole uploaded structure should be assigned to one single element, this will be selected within a SOAP message parameter. If there are more complex relations between the new and target elements, the PDTnet Schema also contains the target elements and the relationships to them. In case of an offline transfer, the PDTnet XML can be replaced by a STEP AP214 Part 21 file, which is specified in the server configuration and considers requirements at target side. In case of an online transfer, STEP AP214 Part 21 is not supported and the PDTnet XML is always sent as content of the SOAP message.

- Referenced files has to be uploaded separately using the use cases “Upload a single digital file” or “Upload a set of digital files”. If the data is sent offline, the files may be added to the upload package, which is specified in the server configuration and considers requirements at target side.
- Any directives/parameters for the upload process are stored at server side.

### 3.2.1.16.7 Relevant data

- Product structure data
- Basic part classification data
- Document meta data
- Document data

### 3.2.1.17 Change notification

#### 3.2.1.17.1 Process purpose

The designer of a part needs notification when a change to a part happens which affects one of the parts he is responsible for. This could take place when a part in the neighborhood of a given part is changed in its dimensions or properties or when a part in an assembly is moved to another place than before. The user specifies the parts on which he wants to be notified by using the functionality of subscribing specified in use case “Change content of subscription list”.

#### 3.2.1.17.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>E-Mail Client</td>
<td>System, that is able to maintain the user’s e-mail</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is</td>
<td>System</td>
</tr>
</tbody>
</table>
3.2.1.17.3 Process definition

Target of the process is the evaluation of objects being changed since the last visit of the user to this object. When a modification of those objects is being detected, an appropriate message has to be delivered to the user. Objects could be parts (and part versions), documents (and document versions) or models.

Changes to report could be:
- Creation of a new version of an object
- Change of the release status of an object
- Objects are deleted
- Geometry has changed
- Properties have changed

The following requirements are defined:
- Two possibilities of detecting changes on the server side are conceivable. Which of them is used is depending on the PDM server implementation:
  - Whenever an object linked to anybody’s subscription list is changed, an e-mail is sent to the user(s)
  - In certain periods of time, the subscription lists of all users are checked against the objects they include. When a modification of a certain object is detected, an e-mail is sent to the user.
- The frequency and content of e-mail notifications (confidential data must not be included!) are defined server-specifically.

3.2.1.17.4 Process flow diagram

3.2.1.17.5 Process start and end states

Start states / preconditions S1 and S2:
- User has access to his e-mail client

End state / post condition E1 and E2 (Success):
- An e-mail notification about changes to one of his objects collected in the clipboard is sent to the user
3.2.1.17.6 Constraints and assertions

Currently none are defined.

3.2.1.17.7 Relevant data

- Product meta data

### 3.2.1.18 Display content of subscription list and confirm changes

#### 3.2.1.18.1 Process purpose

To get an overview about objects being changed on the PDM server, the user should be able to display the contents of his subscription list in which he collects all the objects to track. The changed objects should be displayed in an emphasized style to show the status of being changed.

The current content of the subscription list including notifications of changes can be requested by the Web Client:

- when logging in at the server
- when interactively initiated by the Web Client user.

#### 3.2.1.18.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>

#### 3.2.1.18.3 Process definition

Target of the process is the evaluation of objects being changed on the PDM server since the last visit of the user and the notification of the user by displaying the content of the subscription list. When a modification of those objects is being detected, the objects are marked as changed in the subscription list and the reasons of the changes are displayed.

The following requirements are defined:

- The user controls the start of the evaluation process via the client. The results of the evaluation process are displayed directly in the client.
- The change notification data is transferred by the PDM server using the data constructs provided by AP214 (work management information). An additional transfer of change management/notification documents (like PDF files) is currently not needed.
- The user must be able to define and to modify the content of his subscription list (see use case “Change content of subscription list”)
- The subscription list should be represented as a separate folder within the Web Client GUI.

#### 3.2.1.18.4 Process flow diagram
3.2.1.18.5 Process start and end states

Start state / precondition S1:

- A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project’s life time. These items are identified by identifiers.
- The user is correctly logged in and authorized to access the requested information.

End state / post condition E1 (Success):

- The process results in a virtual container (see use case “Search in design space”) containing all the objects in the subscription list.
- Objects modified since the last look on the subscription list are displayed emphasized. Deleted objects are displayed in a different style.
- After confirmation, the modification status of the objects is reset and in the case of deleted objects in the PDM system, they are also deleted from the subscription list.

3.2.1.18.6 Constraints and assertions

Currently none are defined.

3.2.1.18.7 Relevant data

- Product meta data
- Work management data

3.2.1.19 Change content of subscription list

3.2.1.19.1 Process purpose

The idea of the PDTnet subscription list is, that the user needs a sort of folder in which he can collect objects. The purpose of the Subscription lists is to collect objects for which the change notification should be provided. The modification of the objects in this subscription lists is tracked and the user will be notified if such a modification takes place.
The user should be able to change the content of his subscription list. The subscription list contains all objects the user wants to be notified when changes are applied to them.

3.2.1.19.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Server</td>
<td>System, that provides the relevant PDM data. This is usually a company's PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.19.3 Process definition

a) The user selects objects in the subscription list and wants the PDM system to delete the objects from the subscription list.

b) The user selects objects in the PDTnet client and wants the PDM system to link those objects into the subscription list.

The following requirements are defined:
- The user has got a subscription list in the PDM system
- For use case a), the content of the subscription list with the objects to delete have to be displayed.
- For use case b), the objects to add have to be displayed in the client.

3.2.1.19.4 Process flow diagram
3.2.1.19.5 Process start and end states

Start state / precondition S1 (use case a):

- A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project’s life time. These items are identified by identifiers.
- The user is correctly logged in and authorized to access the requested information.
- The content of the subscription list is being displayed in the client.

Start state / precondition E2 (use case b):

- A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project’s life time. These items are identified by identifiers.
- The user is correctly logged in and authorized to access the requested information.
- Product data is displayed.

End state / post condition E1 and E2 (Success):

- The process results in an updated view to the subscription list.

3.2.1.19.6 Constraints and assertions

- The user must own a subscription list.
3.2.1.19.7 Relevant data

- Product meta data

3.2.1.20 Filter product structure and meta data (“PDM filter”)

This use case had been identified and will be described soon.

3.2.1.20.1 Topics under discussion

- The result of this use case will be the input to the use cases “Browsing of product structures”.
- At least, the following fixed filters (on attribute level) should be provided for items, item_versions, documents, doc_versions, digital_files:
  - Latest version
  - Last modified
  - File type = ...  
  - Approval/Release status = ...

3.2.1.21 Product Class Identification

3.2.1.21.1 Process purpose

Identification of a top level product_class to enable browsing of an abstract product structure.

3.2.1.21.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.21.3 Process definition

This use case defines the process of identifying the start node of an abstract product structure in a PDM server. The end state / post condition of the use case is the precondition of the use case “Browsing of an abstract product structure”.

The process steps are::

- The user enters an ID or WildCard.
- PDM Server receives ID or WildCard and triggers search in PDM System -> Exception: The PDM Server does not respond.
- PDM System executes query in its database -> Exception: Database is not available, no data found, user is not authorized to access the data, etc.
- PDM Server returns a list of product_class and product_component nodes.
- PDM Web Client displays the resulting product_class nodes. If the list has only one member it shall be displayed as the root node of a tree. If the list contains more than one
node than the result should be displayed as a list from which the user may select one node that is than displayed as the root node of a tree.

Remark: according to the CC8 Recommended Practices, each product_class is associated to one instance of product_component (with relation_type='realization’) having the same attribute values. From this instance of product_component (not displayed within the client), the abstract product structure may be traversed (ProductStructureQuery).

Remark: product_functions are not supported at this time.

3.2.1.21.4 Process flow diagram

*At the moment no process flow diagram is provided.*

3.2.1.21.5 Process start and end states

Start state / precondition S1:
- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id or WildCard.

End state / post condition E1 (Success):
- The list of resulting nodes is displayed as described above.

End state / post condition E2 (Failure):
- The process results in a failure message.

3.2.1.21.6 Constraints and assertions

At the moment none are defined.

3.2.1.21.7 Process data

- Product_class information

3.2.1.21.8 Open points

At the moment none are defined.

3.2.1.21.9 Diagrams

*At the moment no diagram is provided.*

3.2.1.22 Browsing of Abstract Product Structures

3.2.1.22.1 Process purpose

This process allows a user starting with an identified product_class, product_component or alternative_solution to get information on the subcomponents of an abstract product structure (product_component or item_instance).

3.2.1.22.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a</td>
<td>Person / System</td>
</tr>
</tbody>
</table>
3.2.1.22.3 Process definition

The process steps are:

- The PDM Web Client evaluates if the product structure information is already obtained then it is directly displayed in a table.
- The PDM Web Client sends a query for a substructure of product_class, product_component or alternative_solution specified by the user to the PDM Server.
- For each product structure node in the scope of the query the PDM Server
  - Checks the authorization regarding the requested data
    -> Exception: Access denied
  - Collects requested data within the PDM Server
- PDM Server sends data to the PDM Web Client.
- PDM Web Client displays the resulting nodes within the structure. The kind of relationship (e.g. product_structure_relationship of kind “decomposition” or “realization”) and child node (product_component or item_instance) should be displayed within the Neutral Web Client.

Remark: only one level of the product structure is retrieved at a time.

Remark: product_functions are not supported at this time, and only product_structure_relationships from product_component to product_component from alternative_solution to item_instance and from alternative_solution to product_component are supported.

Remark: all the subtypes of item_instance are supported (single, quantified and selected). selected_instance is used in the case of a quantity ‘as needed’: selected_instance.selection_quantity refers to an instance of value_limit with limit=0 and limit_qualifier=’minimum’.

Remark: this functionality is also available on item_version nodes if they are handled both as part (for their usage) as well as product_component (having an own abstract product structure). In this case, the function handles the item_version just as if it was a product_component.

3.2.1.22.4 Process flow diagram

At the moment no flow diagram is provided.

3.2.1.22.5 Process start and end states

Start state / precondition S1:
- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):
- The list of resulting of the resulting nodes is displayed as described above.
3.2.1.23 Browsing of Alternative Solutions within an Abstract Product Structures

3.2.1.23.1 Process purpose

This process allows a user starting with an identified product_component [or alternative_solution] to get information on the [sub-]alternative solutions of an abstract product structure.

3.2.1.23.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is usually a company's PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.23.3 Process definition

The process steps are:
- The PDM Web Client evaluates if the alternative_solutions are already obtained then it is directly displayed in a table.
- The PDM Web Client sends a query for the alternative solutions of a product_component [or alternative_solution] specified by the user to the PDM Server.
- For each alternative solution node in the scope of the query the PDM Server
  - Checks the authorization regarding the requested data
    -> Exception: Access denied
  - Collects requested data within the PDM Server
- PDM Server sends data to the PDM Web Client.
PDM Web Client displays the resulting nodes within the structure. The kind of child node (alternative_solution, technical_solution, final_solution, supplier_solution) should be displayed within the Neutral Web Client.

3.2.1.23.4 Process flow diagram

At the moment no flow diagram is provided.

3.2.1.23.5 Process start and end states

Start state / precondition S1:
- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):
- The list of resulting of the resulting nodes is displayed as described above.

End state / post condition E2 (Failure):
- The process results in a failure message.

3.2.1.23.6 Constraints and assertions

At the moment none are defined.

3.2.1.23.7 Process data

- Product_structure_relationships, Product_components

3.2.1.23.8 Open points

At the moment none are defined.

3.2.1.23.9 Diagrams

At the moment no diagram is provided.

3.2.1.24 Retrieve Configuration Data within an Abstract Product Structures

3.2.1.24.1 Process purpose

This process allows a user starting with an identified alternative_solution or item_instance to get information on the configuration of an abstract product structure.

3.2.1.24.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is usually a company's PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>
3.2.1.24.3 Process definition

The process steps are:

- The PDM Web Client evaluates if configuration information is already obtained then it is directly displayed in a table.
- The PDM Web Client sends a query for the configuration[s] of an alternative_solution or item_instance specified by the user to the PDM Server.
- For [each] configuration node in the scope of the query the PDM Server
  - Checks the authorization regarding the requested data
  - Exception: Access denied
  - Collects requested data within the PDM Server
- PDM Server sends data to the PDM Web Client.
- PDM Web Client displays the resulting nodes within the structure. The associated Specification referenced through Configuration and Class_specification_association should be displayed within the Neutral Web Client as a property of the configuration.

Remark: currently, configuration may be only displayed on alternative_solution and item_instance, but not on product_component and product_function.

Remark: according to the recommendations of the ‘Product Configuration and Specification’ workgroup, for complexity reason the specification_expression corresponding to the logical rule stored within the legacy system is mapped to a single string and mapped to a pseudo-Specification.id. This specification is directly referenced by the Class_specification_association. The category of this specification has id=/DUMMY.

Remark: the product_class referenced by the class_specification_association will not be displayed to the Web-Client, since it is either derived from the root node of the abstract product structure, or is project independent (for example in the case on configured assembly structures) and would have to be instantiated with a product_class of kind ‘enterprise’.

Remark: if the usage of a part or product_component is not configured (i.e. the associated logical rule is empty), this function will give no results.

3.2.1.24.4 Process flow diagram

At the moment no flow diagram is provided.

3.2.1.24.5 Process start and end states

Start state / precondition S1:
- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):
- The list of resulting of the resulting nodes is displayed as described above.

End state / post condition E2 (Failure):
- The process results in a failure message.

3.2.1.24.6 Constraints and assertions

At the moment none are defined.
3.2.1.24.7 Process data

- Alternative_solution, Item_instance, configuration, product_class,
  class_specification_association, specification, specification_category

3.2.1.24.8 Open points

At the moment none are defined.

3.2.1.24.9 Diagrams

At the moment no diagram is provided.

3.2.1.25 Viewing of Change Management Information

3.2.1.25.1 Process purpose

Browsing through a product structure the user is able to see the assigned change management information.

3.2.1.25.2 Partner/actor role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Party, that requests PDM data. This could be a person, who interacts with the PDM Web Client, or a system, that triggers the PDM Web Client.</td>
<td>Person / System</td>
</tr>
<tr>
<td>PDM Web Client</td>
<td>System, that provides the communication between user and PDM server</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server</td>
<td>System, that provides the relevant PDM data. This is usually a company's PDM system that acts as a server.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.1.25.3 Process definition

The process steps are:

- The user selects a node (product_class, product_component, item_version) within the PDM Web Client.
- The PDM Web Client evaluates if work management information is already obtained then it is directly displayed in a table.
- If work management information is not obtained the PDM Web Client sends a query for this node to the PDM Server.
- PDM System executes query in its database
  -> Exception: Database is not available, no data found, user is not authorized to access the data, etc.
- PDM Server sends obtained work management data to the PDM Web Client.
- PDM Web Client displays the resulting data in a table.

Remark: according to the CC8 Recommended Practices, the effectivity references an event_reference, which references again an activity. Effectivity_assignment.effective_element and Activity_Element.element both reference the product_class, product_component or item_version node.

Remark: other object nodes are not supported at this time.

3.2.1.25.4 Process flow diagram
At the moment no flow diagram is provided.

3.2.1.25.5 Process start and end states

Start state / precondition S1:
- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user selects a node of kind product_class, product_component or item_version in the tree view.

End state / post condition E1 (Success):
- The resulting information is displayed as described above.

End state / post condition E2 (Failure):
- The process results in a failure message.

3.2.1.25.6 Constraints and assertions

At the moment none are defined.

3.2.1.25.7 Process data

- Activity, Activity_element, Effectivity, Effectivity_assignment, Event_reference

3.2.1.25.8 Open points

At the moment non are defined.

3.2.1.25.9 Diagrams

At the moment no diagram is provided.

3.2.1.26 Viewing of Digital Documents

This use case is not defined yet.

3.2.2 Scenario: PDM system synchronisation

This scenario is characterized by two PDM systems, which are being synchronized by the use of a neutral product data communication protocol.

The use cases for this scenario are currently being defined and discussed.

3.2.2.1 Creation/exchange of identifiers between OEM and supplier

This use case had been identified and will be described soon.
3.2.2.2 **Project setup**

3.2.2.2.1 Process purpose

The project setup processes describes the technical initiation of a synchronization project between an OEM and a supplier.

3.2.2.2.2 Role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OEM-Administrator</td>
<td>This person has the access rights to create and modify basis information and setups in the server system at OEM side.</td>
<td>Person</td>
</tr>
<tr>
<td>Supplier-Administrator</td>
<td>This person has the access rights to create and modify basis information and setups in the server system at supplier side.</td>
<td>Person</td>
</tr>
</tbody>
</table>

3.2.2.2.3 Process definition

The process steps are:

- Define a new or choose an existing project environment
- Create/select an exchange node
- Create/select the access rights concerning this node
- Create/select a mapping table which does the translation of the description of entities between the partners e.g. part name, date information, milestone description, version description
- Optionally define the detail level of documents for the partner (e.g. catia model with history, catia model without history, hull geometry, 3D maps, bounding box)
- Connect the exchange node with the appropriate exchange node of the partner
3.2.2.3 Automatic synchronization of documents

3.2.2.3.1 Process purpose

With the help of this process documents shall be synchronized between the PDM systems of the partners. The process is controlled by a batch process.

Definition of synchronization:

In this use case synchronization means that a synchronization subsystem shall take care that the same state of documents (and document versions) exists in both PDM systems. Usually this will mean a two way exchange of documents.

It must be possible to define which PDM system starts with a synchronization interval.

Every PDM systems first detects if there were changes locally compared to the state of the last synchronization. If there where changes the system detects the changes and stores the information which documents have been changed and should be transferred to the PDM system of the partner.

The predefined PDM system sends its information about changes, it would like to transfer to the system, of the partner. This is information is just a request not the actual files. The partner system compares this request with its own requested changes and detects if there are conflicts. It then sends back a modified request for documents it likes to receive from the partner. The partner system processes this request and sends the documents. Then the process repeats for the other system.

In the case of conflicts or disruption the there must be some kind of notice to the user who has created this synchronization task and the VPM administrator.

The system must detect that if a synchronized document has not been modified since the last synchronization so that it does not transfer this document again.

The documents that have to be synchronized must be identified by unique attributes (e. g. last change date, new document). These attributes should be selectable from a list of predefined attributes.

For each document and its versions there must be the possibility to define in which way it shall be synchronized. There must a way to define which is the leading (master) system for a document. If there is a change of the same document that shall be synchronized in both PDM systems at the same time, there must be a warning because of this conflict that informs the both users who have done the modifications.

An exception of the usual exchange request are deleted documents. It has to be discussed if delete request are processed in the partners system. Usually a deletion of documents must be prevented.

During the synchronization the system should be in a quiet state as possible. An optimal state would be no user interaction e.g. midnight. If this can not be done there must be a way of defining "snapshots", means a status of a system at a specified point of time that shall be synchronized.

It still has to be discussed in which way exchanged documents shall be presented/organized in the remote system if there is no exchange of product structure or no mapping for
structures/configurations. For example these documents could be organized under a node below the exchange node in each system.

Every transaction must be logged.

3.2.2.3.2 Role descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDM Web Server (OEM)</td>
<td>System, that provides the relevant PDM data at OEM side. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server (Supplier)</td>
<td>System, that provides the relevant PDM data at supplier side. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
<tr>
<td>Synchronization Subsystem</td>
<td>Software system, that synchronizes data between two PDM systems automatically.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.2.3.3 Process Definition

The automatic start of the synchronization of documents use case describes the way synchronization tasks should run in batch mode without user interaction. The synchronization task will be stored by the synchronization subsystem and be called by the defined frequency (e.g. every workday, midnight)
3.2.2.4 Automatic replication of product structure and configurations

3.2.2.4.1 Process purpose

This process defines the way product structures and configurations are exchanged between an OEM and a supplier.

The product structures/configurations do not really have to be the same at both partners sites. In such a case there must be a mapping between the structures/configurations.

3.2.2.4.2 Role Descriptions

<table>
<thead>
<tr>
<th>Role name</th>
<th>Role description</th>
<th>Role type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDM Web Server (OEM)</td>
<td>System, that provides the relevant PDM data at OEM side. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
<tr>
<td>PDM Web Server (Supplier)</td>
<td>System, that provides the relevant PDM data at supplier side. This is usually a company’s PDM system that acts as a server.</td>
<td>System</td>
</tr>
<tr>
<td>Synchronization Subsystem</td>
<td>Software system, that synchronizes data between two PDM systems automatically.</td>
<td>System</td>
</tr>
</tbody>
</table>

3.2.2.4.3 Constraints and assertions

There will be no synchronization of product structures/configurations. During the project setup phase it must be defined if structures will be pushed to or pulled by the system of the partner.

A replication shall be possible in both ways.
4 CONCEPTS OF PDM INTEGRATION

There are three different basic techniques for realizing data exchange and communication between development partners (Fig. 4.1).

For the file based PDM data exchange the only standard available is the STEP standard.

Web clients for several PDM systems are already available. They allow the viewing, selecting and input of data into the customer system. They cannot offer full PDM functionality or replace a PDM system at a supplier’s site.

Data sharing implementations for product data have already been realized in some projects. However, harmonization of semantics and data structures is essential for utilizing these techniques. In all data sharing approaches concerning product data, STEP has been chosen as the semantic and structure baseline.

**Figure 4.1: Approaches for Product Data Communication**

Each concept has specific characteristics regarding technology, process integration, types of information and costs.

4.1 FILE-BASED DATA EXCHANGE (ASYNCRONOUS)

File-based data exchange is based on an asynchronous replication and transmission of data.

Because of the different PDM systems at each manufacturer – often there are individual in-house systems – a native data exchange of PDM data is not possible. The native data exchange is even not possible if sender and receiver are using the same PDM system because if the customization of PDM systems in each company.

The only possibilities are to develop a specific interface for each communication partner or to use the only standard STEP (ISO 10303). With this data format all relevant PDM data can be stored in an exchange file and transmitted to the receiver. For this purpose the application
Pilot projects based on this technology have been performed and are ongoing. They have proven that PDM data exchange based on STEP can be used in practice. Several companies want to start STEP based PDM data exchange in 2000.

The general scenario for a file-based data exchange is shown in Fig. 4.2. On the sender's side, the data of a PDM system is pre-processed to a file in STEP format. It contains organizational and structural data, if appropriate. CAx data is exported to native or STEP geometry files. A data exchange tool (DA tool) generates a file package out of these multiple files. The ENGDAT (VDA 4955) definition is used as a basis for this.

On the receiver's side the ENGDAT package is being unpacked. The STEP file is imported into the PDM system (or a small-sized PDM solution, respectively). The CAx files are routed directly to the CAx or TDM systems, if appropriate.

![Diagram of PDM data exchange](image)

**Figure 4.2: General scenario for file-based data exchange**

**4.1.1 CHARACTERISTICS**

- The processes of the sender and the receiver can be independent.
- The communication process is transparent and can be logged.
- The received data can be processed to fit into the own data management environment.
- The data exchange can be time consuming.
- The data exchange leads to redundancy on both sides.
- Data consistency can not be assured.
- The responsibility of the sending partner for the correctness of the transmitted data ends with the correct completion of the transfer process. In case of defective data the receiver is forced to perform appropriate corrections.

**4.2 PDM WEB CLIENTS**

PDM Web clients are applications, that allow a user to access a remote PDM database via internet technology (internet, extranet or other industrial network). This type of integration does not provide a real online connection, but due to the usage of data streaming techniques and due to the possibility of an immediate response by a remote system it comes near to it.
Web clients generally transfer only metadata. To transfer bigger amounts of data, like CAD models, the Web client initiates file transfer processes.

The outstanding data language for the transmission of data via internet is XML (eXtensible Markup Language). XML is a subset of SGML and will be the internet standard of the near future. There are tools for XML available. XML interfaces are easy to develop. But until now there are no clear definitions how to use XML for PDM data exchange. There is no defined structure or semantic for PDM for XML.

PDM Web clients can be categorized into native and neutral Web clients.

### 4.2.1 Native Web Clients

Native Web clients are front-ends of a particular PDM system. They allow a user to interact with a specific PDM system via internet. They generally support the functionality of the according PDM server system.

Web clients usually use the HTTP protocol for data communication with a PDM server. Based on this, they can use different techniques for data requests and data representation (HTML, forms, XML, ...). For native web clients, the semantics and the structure of the exchanged data and requests is based on the data model of the PDM system. This means, that it works only in connection with one proprietary PDM system.

### 4.2.2 Neutral Web Clients

Neutral web clients are independent of a certain PDM system. They shall allow a user to access different PDM systems using a unique interface. The scenario for a neutral Web client is shown in fig. 4.3. A main challenge in this scenario is the integration of the neutral web client and the PDM system on the supplier’s side. If no PDM system exists, the neutral web client provides at least viewing/browsing, download and upload functionality for PDM data.

![Diagram](image)

**Figure 4.3:** General scenario for a PDM integration based on a neutral web client

To implement this technology, the PDTnet project provides the “PDTnet STEP AP214 / XML Schema”. This XML-based data schema provides best use of proved STEP technology and actual web technology. It provides the answers to important
questions for setting up feasible solutions for the system-independent integration of partners by combining STEP AP214 and XML:

- Which data has to be exchanged?
- Which data model shall be used?
- What is the meaning of data?

The answers are provided by **STEP AP214**, which defines a standardised and proven data model for the needs of the automotive industry and other production industries.

- How to transport / access data?
- Which data format to use?
- Which systems are feasible?

The answers are provided by **XML Schema**, which defines a standardised syntactic data representation that can be by processed by commercial XML-capable software systems (e.g. web-browsers). Additionally, a protocol for online system access (SOAP) is used.

The neutral web integration concept of the PDTnet project allows a user by interacting with a Neutral PDM Web Client to access the PDM system of the customer and the own PDM system (if existing).

![Diagram](image)

**Figure 4.4**: Neutral PDM Web Integration in a development network

The “PDTnet STEP AP214 / XML Schema” contains product structure data, administrative product data (approval data, effectivity data, ...) and document data (e.g. CAD model data). The detailed description of the content is described in the “PDTnet Implementation Guide” [PDTnet-IG-11].

### 4.2.3 Characteristics

- Standardized and common software modules can be used (particularly for neutral clients).
- A neutral Web client provides a unique access for different external PDM servers.
- The responsibility of the sending partner for the correctness of the transmitted data can be ensured because of the direct communication possibilities via network. In case of defective data the receiver is forced to perform appropriate corrections.
- The PDM Web client technology supports an automated communication with external PDM systems.
• Performance and availability depend on the internet and are not guaranteed. This disadvantage is not valid, if a reliable network like ENX (European automotive network) is used.
• A PDM Web client usually does not provide real PDM functionality to exactly support the supplier's internal processes. Generally, an own PDM system is necessary. Nevertheless, the “Browsing/Viewing” functionality for product structure data and additional administrative and document information can be a major advantage.

4.3 DATA SHARING / ONLINE INTEGRATION

Data Sharing is a synchronous data access using a network. Data will be not replicated. There can be links between data objects across the network. For the user it should be transparent where the data are physically stored. Data sharing is used e.g. for the integration of CAx and TDM applications to support a concurrent product development in a distributed team of users.

With the reliable network based on ENX some preconditions will be available soon. Technically there are two reasonable alternatives for data sharing implementations:

• PDM Enablers based on CORBA standardized by OMG
• XML with PDM specific definitions

In the area of PDM Enablers some PDM vendors are active. Prototype solutions are existing for Metaphase (SDRC), ENOVIA (Dassault Systemes / IBM), Windchill (PTC). Pilot projects to use this technique are in the beginning. For semantical aspects STEP PDM definitions are used.

4.3.1 CHARACTERISTICS

• There is no redundancy or inconsistency of the shared data and the data is up-to-date by principle.
• The validity of data that shall be stored is guaranteed due to the possibility to check it immediately. Invalid data will not be accepted.
• Security mechanisms can be difficult to implement.
• The reliability of the network is a precondition.
• Data accesses can not be logged easily.
5 STRATEGIC RECOMMENDATIONS FOR PDM INTEGRATION

5.1 GENERAL OEM STRATEGY

The character of the manufacturer-supplier relationship has changed dramatically during the last decade.

The OEM has become an integrator and project manager, and the supplier has assumed more and more responsibility for product and process development. Complete modules are developed and manufactured by system suppliers who are performing tasks today that OEMs performed in the past.

The supplier is integrated in the core development process, and the IT infrastructure has to support this process requirement.

Of course, the degree of integration depends on the role the supplier plays in the process.

One consequence of the close integration of the supplier in the development process of the customer is an increasing segmentation on the part of the supplier. The supplier has to set up different structures to reflect the various process requirements of his customers. This includes

- customer-specific process rules and standards,
- customer-specific data formats
- customer-specific systems -- to an increasing extent in all areas of the development process: CAD/CAM, PDM and BoM, and documentation systems

This means that once a project - e.g. for VW - has been completed, development team from that project cannot be used for a new project for a different customer - e.g. DaimlerChrysler - without a lot of time and money spent on familiarizing the members of the team with the new project. Not to forget the cost of purchasing and maintaining the required hardware and software.

The process sovereignty of the supplier is lost since the supplier's processes are intertwined to an increasing degree with those of their customer.

This apparent contradiction can only be resolved by co-operation between business partners in the search for the best solution for the parties involved.

What is needed are integration approaches that

- are tailored to actual process requirements and which take the overall process costs (including subsequent costs incurred by the supplier) into consideration,
- are based on checklists and process agreements which also make possible the integration of heterogeneous system landscapes, and
- are fast and inexpensive to implement

As is the case with optimizing processes within a company, integration across company boundaries must not be reduced to simply finding an appropriate system. What is needed is a change in the way in which those involved think:

- We have to move away from the sub-optimization of individual process steps towards viewing the process chain as a whole.
• We have to *move away* from focusing on THE system towards viewing the processes and making a decision regarding the system based on this view.

As mentioned earlier, what is required is a change of paradigm towards co-operation between partners [LaNiMa-99].

To support collaborative development processes integrated solutions and systems for product data communication are necessary. The requirement of supporting different processes and systems of different customers suggests the use of neutral communication and data protocols on all levels: network, security, data formats and information content (semantics). In addition, various technical criteria for concrete implementations have to be considered.

### 5.2 Recommendations for Suppliers

#### 5.2.1 Processes and Collaboration Models

Usually, the development processes on supplier’s and OEM’s side are different. This, of course, is the main reason for data integration problems in the field of PDM. Therefore, an intensive need for agreements on a synchronization of the processes exists. At the beginning of a new development project the partners should exactly define the development and, if appropriate, the manufacturing processes. An example for involved processes between an OEM and a general contractor / system supplier is shown in figure 5.1.

![Diagram](image)

Figure 5.1: Development processes between DC and Steyr-Daimler-Puch (Source: Steyr-Daimler-Puch Fahrzeugtechnik)

The integration scenarios and processes have to be differentiated in dependency on the relationship between the manufacturer and the supplier in a particular project and the corresponding processes. It is recommended to the supplier to first check the role he takes within the cooperation with the manufacturer. This will be a major help for the further detailed planning of the cooperation processes and information flow.

To provide a guideline for suppliers how to cooperate with manufacturers, the VDA established the work group “Kooperationsmodelle” (“Cooperation models”) [VDA-KM-01].
The goal of this work group is the classification of cooperation models between development partners. This classification serves as a first entry-point for a supplier to set up an efficient collaboration with a manufacturer. It extends the VDA-Recommendation “SE-Checklist” [VDA-4961], which provides more detailed information about concrete characteristics of collaborative work between partners.

The cooperation models are classified using three main criteria:

- The general characteristic of the cooperation model describes the way of process integration of the sub-contractor in the processes of the contractor and the responsibility of the sub-contractor. It is extended by a criteria matrix for cooperation models.

- The depth of integration in the complete product creation process. The following types of integration are defined:
  - Production integration (e.g. assembly inspection)
  - Process integration into the processes of the contractor (adoption of instructions of contractor’s processes, work within the contractor’s processes)
  - Functional integration (e.g. functional integration of audio and navigation system)
  - Geometric (three-dimensional) integration (e.g. DMU)

- Necessary information extend and exchange

The VDA group defines 6 classes of cooperation models, named from the viewpoint of the sub-contractor:

- General contractor, who offers the whole car development and/or production of vehicles to a manufacturer; examples are Steyr-Daimler-Puch or Karmann. Of course these kind of suppliers have numerous sub-suppliers of all other kind again.
- System supplier, who is responsible for development and/or production of function-related systems. Often these kind of suppliers are so-called 1st-tier suppliers and have several sub-suppliers.
- Module supplier, who is responsible for development, integration and/or production of complex modules. The main emphasis of its work is the geometric integration of modules for all product variants.
- Component supplier, who is responsible for development and/or production of one component (this is generally a position within the contractor's bill of material). Component suppliers are not (or only few) integrated in the business process of the contractor.
- Part supplier, who is manufacturing single (standard) parts or simple assemblies.
- Design service provider and residential engineers, which are offering design support for manufacturing companies. They provide services (data), not products.

Some characteristics and recommendations for these cooperation models are explained in the following sections.

5.2.1.1 General contractor

5.2.1.1.1 Process characteristics

The general contractor has similar processes as the OEM itself. The only difference is, that he has no own brand and he is ordered to produce a particular vehicle. The general contractor offers the whole vehicle development and production to a manufacturer. Examples are Steyr-Daimler-Puch or Karmann.

5.2.1.1.2 Need for information
If the car is a variant of cars produced by the OEM, the general contractor needs the complete information about the product including geometry, structure, bill-of-materials, variants, suppliers etc. If a “new” model without too much relationships to other models of the manufacturer is developed, the primary data is managed by the general contractor.

5.2.1.3 Recommendations

A general recommendation for the systems and the data exchange technologies for a general contractor is not possible. There are contradictory requirements which only can be solved with different solutions.

There is a big need for the supplier to work on the same data at the same time as the manufacturer itself. It is very difficult to share the tasks and the necessary data between the supplier and the manufacturer and sequential working is not efficient. This scenario would be best supported by using the same tools with direct access and working on the same data base.

The contradictory requirement is that the general contractor of course needs an own company wide product data management solution to support his own processes. It can not be expected, that the supplier can share the PDM functionality with the customer. It is much more probable, that the supplier has more than one customer with different PDM solutions. This requires an independent PDM solution for the supplier as well.

Two techniques may be adequate to support this second scenario:

- online automatic data synchronization e.g. via PDM Enabler or XML technology
- offline automatic synchronization by trigger or time schedule in addition to sequenced processes (responsibility for data scopes are distributed)

The second technique is available and already used in first projects (e.g. DaimlerChrysler – Steyr-Daimler-Puch). The minimum preconditions for this scenario are:

- availability of STEP processors
- agreed upon data mapping for both directions
- agreed upon responsibilities for changes and change processes

In both cases it is very difficult to support different but consistent views on the product data at the manufacturer and the supplier (product structure, configuration, properties, approval stati etc.).

5.2.1.2 System supplier

5.2.1.2.1 Process characteristics

The system supplier is responsible for the development and/or production of function-related systems. He ensures the functional integration of a system, which components can be located at different places of the vehicle.

The system supplier owns the primary data of the system and delivers only a “customer view” on the data with reduced version, compressed structure, customers identification etc. to the contractor.

5.2.1.2.2 Need for information

The system supplier particularly manages geometry data for each system component (for DMU purposes). The DMU analysis is performed within the system environment of the contractor.
The system supplier needs actual product data which includes geometry, structure, configuration (variant), design space, connecting parts, material, identification, approval status etc. The view on these data may be different (e.g. a complex assembly on the supplier’s side is represented as a single part on the OEM’s side). Properties and attributes of parts/assemblies may be different, too.

5.2.1.2.3 Recommendations

The supplier has to decide whether he needs an own company wide PDM solution and if yes, which type. This decision is dependent on many different aspects (see also annexes). But both cases have to be considered:

If the supplier has an own company wide PDM system, he may have the same requirements for tools as the complete car manufacturer. He also has the intention to connect his system with the customers system. If the PDM system of the supplier does not have a STEP interface for PDM data, he may work with an additional tool which can be linked to the PDM systems. These tools may be proprietary web clients, neutral web clients or offline tools (see chapter 5.3.)

If the supplier does not have a company wide PDM system he usually is not able to process the PDM data from the customer without an additional tool. CAx systems do not have the capability to process PDM data. In this case, the above mentioned tools can provide a suitable solution.

5.2.1.3 Module supplier

5.2.1.3.1 Process characteristics

The module supplier is responsible for the development, integration and/or production of complex modules. The main emphasis of its work is the geometric integration of modules for all product variants.

The module supplier owns the primary data of the module and delivers only a “customer view” on the data with reduced version, compressed structure, customers identification etc. to the contractor.

5.2.1.3.2 Need for information

The need for information is similar to that of the system supplier. The module supplier particularly manages geometry data for each system/module component (for DMU purposes). The DMU analysis is performed within the system environment of the contractor.

The module supplier needs actual product data which includes geometry, structure, configuration (variant), design space, connecting parts, material, identification, approval status etc. The view on these data may be different (e.g. a complex assembly on the supplier’s side is represented as a single part on the OEM’s side). Properties and attributes of parts/assemblies may be different, too.

5.2.1.3.3 Recommendations

The recommendations are similar to those of the system supplier. The supplier has to decide whether he needs an own company wide PDM solution and if yes, which type. This decision is dependent on many different aspects (see also annexes). But both cases have to be considered:
If the supplier has an own company wide PDM system, he may have the same requirements for tools as the complete car manufacturer. He also has the intention to connect his system with the customers system. If the PDM system of the supplier does not have a STEP interface for PDM data, he may work with an additional tool which can be linked to the PDM systems. These tools may be proprietary web clients, neutral web clients or offline tools (see chapter 5.3.)

If the supplier does not have a company wide PDM system he usually is not able to process the PDM data from the customer without an additional tool. CAx systems do not have the capability to process PDM data. In this case, the above mentioned tools can provide a suitable solution.

5.2.1.4 Component supplier

5.2.1.4.1 Process characteristics

Component suppliers usually are not directly involved in the development processes of the contractor. This is particularly valid for suppliers of standard parts.

Component suppliers are responsible for development and/or production of one component (this is generally a position within the contractor’s bill of material).

5.2.1.4.2 Need for information

5.2.1.4.3 For the component supplier it is necessary to exchange simplified geometry data (for DMU purposes) with material information, approval stati etc. The geometry data of connected parts or modules is also of importance.

5.2.1.4.4 Recommendations

If no company wide PDM solution exists, the component or part supplier needs an additional tool to process the PDM data from his customers. The already mentioned three possible tools are suitable: proprietary web clients, neutral web clients or offline tools (see chapter 5.3.)

5.2.1.5 Part supplier

5.2.1.5.1 Process characteristics

Part suppliers usually are not directly involved in the development processes of the contractor. This is particularly valid for suppliers of standard parts.

Part suppliers do not provide order-related product development services. They deliver standard parts, or they produce parts according to the contractor’s instructions.

5.2.1.5.2 Need for information

5.2.1.5.3 The information exchange between a part supplier and his contractor is mainly done under the initiation and responsibility of the contractor. The exchanged information is geometry data, identification numbers of the contractor, material information etc. The supplier only needs data of the contractor if a change occurred. For this purpose, a change notification mechanism is useful.

5.2.1.5.4 Recommendations
If no company wide PDM solution exists, the component or part supplier needs an additional tool to process the PDM data from his customers. The already mentioned three possible tools are suitable: proprietary web clients, neutral web clients or offline tools (see chapter 5.3.)

5.2.1.6 Design Service Provider / Residential Engineer

5.2.1.6.1 Process characteristics

Design service suppliers and residential engineers are working directly in the process chain of the customer. The customer may be the car manufacturer as well as most types of the before mentioned suppliers.

5.2.1.6.2 Need for information

Because of the direct involvement of design service suppliers and residential engineers in the processes of the contractor they have to work on the same data. The data created by this kind of suppliers is usually used by the designers of the customer/contractor directly.

5.2.1.6.3 Recommendations

Design service suppliers and residential engineers usually do not have own PDM systems. They work within the customers environment using the same systems. Residential engineers are located within the customers company. Design service suppliers can have their own locations, but often they have direct access to the customers systems.

For this kind of suppliers a direct access to the customers systems is crucial. Therefore the native / proprietary system environment is necessary, i.e. proprietary web clients. It has to be investigated whether neutral web clients are suitable, too.

5.2.2 Neutral PDM Web Integration

The increasing intensity of collaboration between development partners requires the possibility for the partners to continuously provide and access actual product data. Particularly DMU processes are affected. Suitable solutions for this requirement can be provided using current web technology.

Native web clients are already offered on the market (see also 4.2.1). The use of this technology usually forces suppliers to use different web clients of different customers.

To allow the neutral web communication using only one web client for different customers the PDTnet project provides the “PDTnet STEP AP214 / XML Schema”. The Neutral PDM Web Client of PDTnet provides the functionality to fulfil the use cases described in section 3.2. Particularly the functionality of browsing/viewing product data (structures, administrative and document data) makes the Neutral PDM Web Client also a suitable tool for small suppliers without an own PDM system.

5.2.3 Experiences from Projects

For the last years a number of PDM integration projects between OEMs and suppliers had been conducted. A lot of experiences came out of these projects, of which some major ones are explained in the following chapters. These “lessons learned” are expressed as
recommendations, which should be considered in ongoing and future PDM integration projects.

### 5.2.3.1 Semantical definition of relevant PDM data objects

Many time and effort is spend during PDM integration projects for the semantical mapping between the supplier's and OEM’s data objects. This task is a basic precondition before starting the implementation or customization of data exchange processors. This work has to be done in every single project, but it can be reduced if a clear description of the own relevant data objects already exists. This is particularly valid for OEMs, that will perform a great number of integration projects.

It is recommended to generate a mapping table, that maps the company-specific data objects (application data) to neutral STEP data objects. This ensures a maximum of commonly understandable terms.

<table>
<thead>
<tr>
<th>Application Element</th>
<th>Mapping onto STEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>Description</td>
</tr>
<tr>
<td>Name</td>
<td>Entity</td>
</tr>
<tr>
<td>Description</td>
<td>Attribute</td>
</tr>
<tr>
<td>Examples / Valid Values</td>
<td>Rules</td>
</tr>
</tbody>
</table>

Table 5.1: Example for a mapping table

The table should particularly contain valid example values for attributes. Also rules that describe dependencies between elements and/or values shall be identified. Even if data is relevant, which is not covered by the STEP standard, the table shall be filled without a mapping to STEP.

### 5.2.3.2 Distinction of document and product structure data

One crucial point in PDM integration projects is the exchange of product structure data. During the design phase of products two main “views” on product structure exist: One is the product structure as it is defined by Bill of Material systems, the other is the structure as it is defined by the designer within a CAD or TDM system. The latter is also called document structure, because it contains CAD files, which can be interpreted as documents.

It is important to distinguish these two terms. For a receiver of product or document structure data, it is necessary to import and integrate the data correctly in his own systems. In some cases it could be necessary to split the data and to store it in different systems (e.g. CAD documents in a PDM system, product structure data in a BoM system).

There are various reasons, why document structures (particularly CAD model structures, which are a specific type of document structures) are build. At SFT, an analysis resulted in the following aspects:

a. Model splitting: The size of one CAD model is too large or different modeling techniques are used (e.g. BREP, wire frame, CSG, ...).

b. A (specific) assembly structure is represented by a CAD model structure.

c. Necessity of specific checks: E.g. a model space is defined for installation investigations.

d. Differences between external and internal structure at the supplier’s side: E.g.: The internal CAD model of a product consists of several single models, the external model for the customer consists of one single overall model.

e. Differences between external and internal structure at the customer’s side.

f. Additional structuring of CAD partial models (one part, but several models).
All above mentioned document structures do not represent any type of product structure. Only in case b) the document structure may correspond to a product structure.

5.3 TOOLS FOR PDM DATA EXCHANGE

As mentioned before, the most simple scenario of PDM data exchange is to exchange data packages which are including CAx and PDM data. This scenario was the major topic of the projects PDMI and STEP/PDMI2. For this scenario there are tools available which can be used by suppliers for PDM data processing. These tools are the minimum infrastructure to process STEP PDM data. They are not reasonable for suppliers with our PDM solutions with STEP processors. Therefore these tools are especially useful for small and medium suppliers.

There are two kinds of tools available to process PDM STEP data:

a. universal tool independent from any other software system or OEM
b. tools which are dependent on a software system or OEM

A tool of the first kind is the PDM Editor from ProSTEP GmbH which is especially designed to be used by a designer at suppliers site. With this tool the user is able to create, read, change and write PDM data files according to STEP AP214 CC6 or the STEP PDM Schema. The PDM Editor is deeply configurable to be able to customize it in dependency of the customer/supplier processes (terminology, visibility, changeability etc.). First configurations i.e. for BMW are available. Integrations with CAx systems are possible. For BMW a CATIA V4 integration was developed. The PDM Editor is available for major MS Windows and UNIX operating systems. It was tested with PDM files from BMW, DaimlerChrysler and Volkswagen.

GoSTEP214 from Goset is a tool of the first kind as well. The tool is supporting a subset of AP214. The tool is available for MS Windows operating systems.

The COM/STEP tool from T-Systems is dependent on CATIA (Dassault Systems/IBM) and from DaimlerChrysler. COM/STEP is working in CATIA environment and is limited to those objects, attributes and relationships of AP214 CC6 which are in use at DaimlerChrysler. COM/STEP is available for some UNIX operating systems.

With these kind of tools every company is able to take part in the communication of PDM data with customers and suppliers for less than 3.000 Euro of investment.
6 SUMMARY

To fulfill the today’s requirements on engineering collaboration in an OEM and supplier network supplier’s are recommended to use standardized technologies for product data integration and exchange. Depending on the intensity of collaboration and existing system and process environments the following technologies are adequate means:

- For asynchronous data exchange of PDM and CAD data the use of STEP AP214 and ENGDAT is recommended.

- For online PDM data integration and communication the use of neutral and harmonized protocols should be used. For this, the PDTnet project provides the “PDTnet STEP AP214 / XML Schema”.

- Small suppliers should use software tools for the exchange of PDM data, which are able to view and edit STEP AP214 files.

These technologies will help to optimize product data integration and exchange processes and to make them more efficient.
7 REFERENCES


[VDA-DE-01] Results of VDA/ProSTEP Assoc. Group “PDM Data Exchange”: Preparation of a VDA recommendation; currently in work.


8 ANNEXES

8.1 TECHNICAL SELECTION CRITERIA FOR PDM INTEGRATION SOLUTIONS

A supplier has to decide what technical approach he will use for a PDM solution. This selection has to be done on the basis of various criteria. Some main technical selection criteria, that have to be considered, are as follows:

- Performance
  - Frequency of the transfer
  - Transfer rate (depends on the data amount)

- System architecture
  - Client-server-communication
    - Transfer protocol (TCP, SNA, ...)
    - Internet standards (HTTP)
    - Distributed communication (CORBA)
    - Asynchronous File transfer (FTP, OFTP, ...)
    - Type of communication: Batch, transaction
    - Initialization (automatic / manual)
    - Transfer (push/pull/notification to the sender / addressee)
  - Information exchange between the locations (central DB / Replication)
  - GUI (Front-end) (3270, Windows, Browser, ...)
  - Used standard software components
  - Database type (RDBMS, OODBMS, file-based, hierarchical)
  - Security (encryption, authentication, identification)
  - Backup options

- Administration
  - Central / local
  - Support

- Integration of other systems / interfaces
  - Migration
  - Formats (STEP, XML, HTML, ...)

- Configuration
  - Different views on product data (OEM/Supplier)
  - GUI at the supplier's side
  - Functionality with reference to OEM conditions

The criteria catalogue shown above does not represent a complete set of aspects, but it shows the main technical blocks of selection criteria for PDM solutions.

8.2 COST DRIVERS IN PDM PROJECTS

The estimation of costs occurring out of PDM integration projects is difficult to generalize. A recommendation of the VDI (VDI 2219) provides detailed information regarding the introduction and the economic efficiency of PDM systems. The major cost blocks in a PDM...
project are listed on the following page and described by the major cost elements. They are based on the mentioned VDI guideline 2219 and on experience.

**Specification:**

This cost block summarizes the costs for all specifying work necessary for the new PDM system and its implementation: specification of the
- user functions (call, parameters, procedure, result, dependencies etc.)
- user interface (layout, field types, field lengths, visibility, access, changeable/read only etc.)
- data model (objects, attributes, relationships, logical, physical)
- migration (scope, mapping, export/import format, interfaces)
- introduction (numbering systems, classification systems)
- reengineering of processes; definition of work-flows
- integration (CAD, TDM, PDM, data exchange tools etc., scope, interface)
- interfaces (scope, mapping, technique, format, feedback, consistency etc.)
- implementation (number of seats, locations, infrastructure, type of seats, PDM module combinations, replication, distribution etc.)
- administration (locations, capacity, profiles, tasks etc.)

**Project Management:**

This cost block summarizes all costs for organization and management work
- Time schedules, work plans, capacity plans, budget plans
- suppliers organization (subcontractors, consultants, vendors)
- management reports/presentations
- task coordination, progress control, technical harmonization of tasks, decision support etc.

**Investment for hard- and software:**

This cost block describes all hard- and software costs
- Contract negotiation
- new computers, upgrading of existing computers
- new software components (OS, data base, communication/network etc.)
- PDM system licenses (for seats with different functional needs)

**Implementation:**

This cost block summarizes all costs for development tasks for the PDM system
- customizing of the user interface, data model, functionality, work-flows etc.
- development or customizing of interfaces for system integration (Pro/I, DXM etc.)
- test activities (use cases, test, evaluation, report)
- proof of concept, pilot phase

**Migration:**

This cost block summarizes all costs for moving data from legacy systems into the PDM systems (EWD, partially DAS etc., PNS etc.)
- Definition of scope
- mapping between the systems
- determination of exchange technique/format
- development of interface, test of data exchange
- data exchange
• verification of data
• documentation/education for/of users
• shut down of old system

**Rollout and Introduction:**

This cost block summarizes all costs for rollout and introduction
• training and education of users and administrators
• coaching of users
• introduction of new work-flows
• creation, adaptation of documentation for users
• installation of necessary software at seats

**Maintenance/Hotline/Support:**

This cost block summarizes all costs for the run of the system
• Concept for maintenance, hotline and support
• Contract negotiation with vendor/service provider
• annual maintenance
• Annual hotline / support
• In-house system administration

The mentioned cost blocks and elements occur in every PDM integration project.
Additionally, the amount of costs is triggered by some major cost drivers. These cost drivers for the implementation of a PDM integration are:

1. Frequency of the data-exchange
2. Amount of data transfer
3. Quality of the data
4. Scope of used functionality of the data transfer system
5. Scope of customization
6. Number of systems which have to be integrated
7. Number of legacy systems which have to be migrated
8. Number of locations, integrated OEMs or suppliers
9. Scope of necessary process reengineering
10. Number and complexity of work-flows which should be implemented

Of course there are some not quantifiable success factors which should be taken into account and which can have a significant influence onto the project costs. The main success factors are:

• qualification and capacity of the PDM project managers and administration team
• project support by high management, i.e. for difficult decisions
• qualification and capacity for support by the system vendor and/or service provider
• acceptance by the users; preparation of the users
• completeness and quality of specification