	Produktdatenaustausch Product Data Exchange Teil 1: Austausch von Baugruppen Part 1: Assembly Data Exchange	VDA 4956

Product Data Exchange

Part 1: Assembly Data Exchange

Version 1.1

Die Empfehlung wurde entwickelt in Zusammenarbeit mit dem ProSTEP – iViP eV durch Vertreter folgender Firmen: Behr Group, BMW AG, DaimlerChrysler AG, Delphi Automotive Systems GmbH, Eigner+Partner GmbH, PDTec GmbH, Porsche AG, PROSTEP AG, PSA Peugeot Citroën, Renault SAS, Scania AB, T-Systems ITS GmbH, Volkswagen AG, Volvo AB, ZF Friedrichshafen.

VDA Verband der Automobilindustrie

	Product Data Exchange Part 1: Assembly Data Exchange	VDA 4956

Contents

	Product Data Exchange Part 1: Assembly Data Exchange	VDA 4956

Figures

This recommendation is also published as an official Odette recommendation with the same English title and version and by the ProSTEP – iViP Association.

This recommendation has been developed and is supported by the ProSTEP - iViP Association, VDA and Odette International.

	Product Data Exchange Part 1: Assembly Data Exchange	VDA 4956

Introduction

The need

With respect to a closer cooperation between engineering partners, the increasing complexity of supply chain relationships and global partnerships, OEMs, suppliers and engineering service providers need to communicate with a lot of different partners. Not only single parts, but also assembly structures and even BoM structures have to be exchanged. Also for advanced virtual development tools like DMU application (Digital Mock-up) a closer integration of BoM structures and CAD data, as part of the overall product data communication is required.

Goals of the recommendation

The major goal of Part 1 of the recommendation for Product Data Exchange - Assembly Data Exchange - is a standard based solution for the assembly data exchange specified on the application (ARM) level. There may be additional parts dealing with advanced data scope and/or advanced technologies in the future. The goal is to provide clear and unambiguous recommendations by reducing degrees of freedom in the data exchange process.

The recommendation provides an approach to the integration and complementary usage of relevant standards. Referenced standards are

ISO 10303-214 for the data content

ENGDAT Version 2, March 2001 for data delivery and routing.

The benefits of the recommendation are intended as follow:

- Common understanding of product data exchange processes and an incremental (net change) approach to data exchange,
- Systematic harmonization of terms and data constructs (like entities, attributes and relationships) for defining the relevant data subsets for the various product data exchange cases,
- Condensed recommendation to achieve compatible PDM implementations by vendors,
- Support of systematic harmonization between sender and receiver of product data and minimizing partner specific customization requirements.

The recommendation is addressed to :

- PDM data exchange coordinators on the supplier's side who have to specify the requirements and configurations for product data exchange processes with their customers (in particular clauses 2, 3, 3.1, 4.2, and 7),
- Coordinators for supplier integration on the OEM's side, who have to define OEM-specific assembly data exchange requirements, documented in company-specific product data exchange guidelines (in particular clauses 2, 3, 4, 6, and 8)
- Supplier coordinators who must specify requirements to customers as well as to the lower-tier (tier2 – tier "n") suppliers to whom some of those customers' work has been sub-contracted.
- PDM system vendors and/or integration server vendors to support pre-customization of STEP processors according to the recommendation (in particular the clauses 4, 5, 6.3, 6.4, and 6.5).

Structure of the recommendation

Clause 2 introduces the distinction between the first exchange of a relevant scope of product data in a collaboration project from ongoing incremental updates of product data – also known as “net change”.

Clause 3 describes an approach for the exchange of incremental subsets of (net changes to) product data.

Clause 4 describes process modules as components of an engineering business process and defines the required and optional entities and attributes for assembly data exchange as a subset of the STEP AP214 ARM CC06 to support a process module.

Clause 5 introduces the data dictionary provided as Annex A.

Clause 6 provides introductions to set up an engineering business process including several rules that shall be considered.

Clause 7 gives a survey of relevant systems (i.e., PDM/TDM, CAx, data exchange tool), which may be involved in the product data exchange for sender and receiver and specifies recommended applications for data transfer technologies, i.e., using ENGDAT for packaging and definition of values for processing and routing.

Clause 8 provides approaches to recommended strategies for implementation. The companies themselves may extend the requirements of this recommendation by their own specific requirements for product data exchange.

Annex A provides the complete data dictionary with the definition of mandatory and optional entities and attributes relevant for the defined process modules.

Annex B provides a template for company specific product data exchange guidelines, which may contain extensions to the requirements of this recommendation.

Annex C provides two examples, which show how this recommendation can be applied for a concrete use case.

Annex D provides a detailed introduction to a reference mechanism.

1. Abbreviations, definitions and normative references

1.1. Abbreviations

AP214	**- A**pplication **P**rotocol **214** – ISO 10303-214
AIM	**- A**pplication **I**nterpreted **M**odel
ARM	**- A**pplication **R**eference **M**odel
BoM	**- B**ill **o**f **M**aterial
CAD	**- C**omputer **A**ided **D**esign
CAx	- CAD, CAM (Computer Aided Manufacturing), and CAE (Computer Aided Engineering), CAT (Computer Aided Testing)
CC06	**- C**onformance **C**lass **6**
CORBA	**- C**ommon **O**bject **R**equest **B**roker **A**rchitecture
DMU	**- D**igital **M**ock-**u**p
EDI	**- E**lectronic **D**ata **I**nterchange

ENGDAT	**- ENG**ineering **DAT**a message
ERP	**- E**nterprise **R**esource **P**lanning
OAGI	- **O**pen **A**pplications **G**roup, **I**nc.
OEM	**- O**riginal **E**quipment **M**anufacturer
OMG	**- O**bject **M**anagement **G**roup
PDM	**- P**roduct **D**ata **M**anagement
STEP	**- ST**andard for the **E**xchange of **P**roduct Model Data
TDM	**- T**eam **D**ata **M**anagement (also known as Workgroup Data Management)
W3C	**- W**orld **W**ide **W**eb **C**onsortium
XML	**-** E**X**tensible **M**arkup **L**anguage
XMTD	**-** E**X**change and **M**anagement of **T**echnical **D**ata

1.2. Definitions

Assembly: A node in the part structure that is further decomposed into subassemblies or parts. The assembly structure is a non-configurable explicit part structure (i.e., without variations). The assembly structure is a hierarchical structure of arbitrary depth.

Asynchronous Data Exchange: The process of offline data transfer from one site to another. The data is duplicated on the receiving site with no maintenance of object references to the original data on the sending site.

Bill of Material (BoM): A term for all kinds of part lists that may be structured in different contexts, may include means of product variation control, and may include such objects as software or lubricants in addition to parts.

Engineering Business Process: An engineering business process is a sequence of steps related to the execution of one or more product data exports and driven by an individual engineering activity (e.g., communication of changes, establishing a collaboration).

Data Exchange: A step within an engineering business process that includes the export, the transmission (one-way) and the import of product data from a sending application to a receiving application. Then a round-trip would consist of two exchange steps.

Digital Mock-up (DMU): Digital Mockup is a virtual assembly of the complete product or components of the product in a computer system. The purpose of a Digital Mock-up is all kind of simulations concerning the geometric shape and kinematics or design studies.

Data Module: A data module is a set of data types (entities and their attributes).

Delta exchange: Optional, but out of scope of this document (only accomplished by specific sender-receiver rules set by partners)

Document: A document is a logical container for externally defined product data. A document may be associated to an item or another object and is either represented in physical or digital form.

ENGDAT: An Odette standard that defines a mechanism for the exchange of data packages including information about the sender, the receiver and a description of each physical file in the data package.

The ENGDAT message enables the routing of the data package to the final recipient (technical work at SASIG level: **XMDT** E**X**change and **M**anagement of **T**echnical **D**ata).

File: A file is an atomic container of data on a computer storage medium.

Geometric Model: A geometric model is a representation of a shape. The geometric data are typically generated by CAx systems in digital form.

Item: An item is an element of a product relevant for a bill of material. An item may be a single item, an assembly, raw material or non-geometrical components, e.g., lubricant.

Part 21 File: See STEP Physical File.

PDM Schema: Internationally harmonized intersectional subset of the application protocols AP 203, AP212, AP 214 and AP 232.

Shared area: A subset of PDM, CAx and related data that are shared or exchanged between the partners (e.g. OEM and suppliers) in a project.

STEP Physical File: A file that contains actual product model data encoded as specified in ISO 10303-21. This is the preferred format for the file based asynchronous exchange of product model data.

Process Module: A process module is a set of functionality (e.g. communication and authorization of changes of product data within the scope of this document) which is applied to the data defined within associated data module(s).

Version: A version represents a specific stage of design or maturity of a part, document, or geometric model.

XML: A platform and application independent markup language for representing structured information. XML is suited as a neutral data exchange format especially with means of World Wide Web/ Internet capabilities. **(XML 1.0 :** E**X**tensible **M**arkup **L**anguage; W3C-Recommendation 10-February-1998)

Incremental Data Exchange: An Approach that provides, that only a necessary subset of available product data is exchanged at a certain time.

1.3. Normative references

ISO 10303-214: Core data for automotive mechanical design processes

ISO 10303-21: Clear text encoding of the exchange structure

ENGDAT : Version 2 - March 2001

2. Fundamental Concepts

2.1. First data exchange

An ongoing engineering project is accompanied by a sequence of several engineering processes, including product data exchanges, each of them representing an incremental update of product data. Nevertheless, the first exchange of product data is not intended as an update but rather as an initial load of a relevant scope of product data for a project.

> **Example :** An OEM sends (push principle) or makes available (pull principle) all relevant data for the first time.
>
> **Example :** A supplier sends new data; this is typical as a design proposal during a bidding period often before a manufacturer has detailed product data for the relevant product area or, in another case, a supplier sends back a complete set of data as received by the OEM with certain modifications or additions.

The first product data exchange implies that all created data that is exchanged relevant for a project at a certain time at the sender's side is to be considered and communicated as new data (not a new version of existing data).

2.2. Incremental data exchange

The key feature of incremental data exchange is that only a subset of available product data is exchanged at a certain time. Incremental data exchange reduces the amount of data to be exchanged. This approach supports actual practice in product data exchange where commonly after the first data exchange only subsets or additional portions of product data are exchanged.

The following examples give an idea of incremental data exchange.

> **Example :** An OEM sends or makes available changes of a particular subset of product data to be communicated, e.g., new versions of items, documents, geometry, positioning information or item properties. Usually the data set has links to product data, which has been already sent before or which will be sent later.
>
> **Example :** A supplier sends an update of data. Usually the supplier only has a specific subset of product data from the OEM's viewpoint that he is allowed to modify.

In these cases it is feasible to exchange only the modified portion of product data. Another reason to exchange only a particular portion of product data is that a supplier should only modify and send back the shaft of a gearbox, but he has received the complete gearbox with all connected components in order to provide the environmental context.

Incremental data exchange also includes the exchange of administrative product data without CAx files and the exchange of single items and their administrative product data without item structure.

> **Note :** An "incremental data exchange" is different from a "delta exchange". A delta exchange implies an exchange of differences only usually with respect to a set of data previously sent. That means it is the exchange of results of actions, which have been performed at the sender's side and are to be communicated, e.g., "add component 3 and 4 to an assembly" or "remove document xyz from the set of describing documents of an item". A delta exchange has to be harmonized between two communication partners because change processes usually are company specific.

Figure 1 shows an example where the item marked with an asterisk is an item sent in one package only as a reference (see clause 6.5 for details of the reference mechanism) and details of this item could possibly be sent in a second package.

The symbols used have the following semantics:

A node within an item structure, i.e., a single item or an assembly, representing item definition data.

Represents document definition data, possibly as a component of a document structure for all kinds of managed or unmanaged documents (definitions given in clause 4.3), i.e., for CAD models as well as for non CAD models.

Represents an external file containing geometric definition data, i.e., a CAD model in neutral or native format.

Represents an external file containing non-CAD application data in neutral or native format, e.g., office documents like specifications.

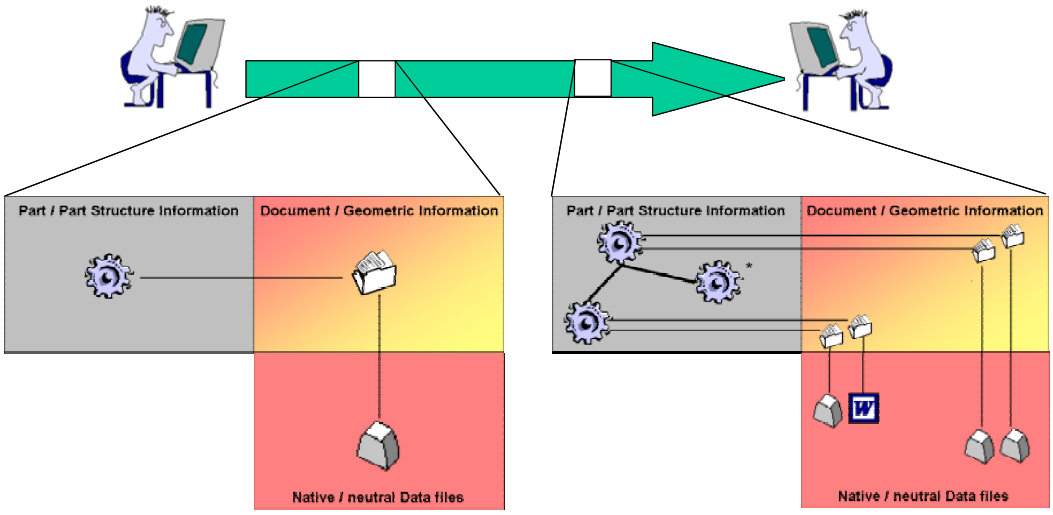

Figure 1 – Example of a data set in an incremental data exchange scenario

3. Process modules and Data modules

Product data exchanges are affected by their context within engineering business processes. The incremental data exchange assumes that an engineering business process can be composed of one or more process module(s). This clause provides an approach for several process modules to compose engineering business processes.

The intention is to reduce the amount of exchanged data.

> **Example :** There is no need to exchange information about the document management, if only changes within the part structure are to be communicated. It is sufficient to communicate the new structure of an assembly.

A process module considers more aspects than only the PDM data scope, for example user interface, and other IT technical aspects. However, these aspects are out of scope of the recommendation. Only the PDM data scope is considered here.

Each process module is closely related to a subset of entities of ISO 10303-214 ARM CC06 that is necessary to support the specified scope of product data to be exchanged. One or more data modules define this subset. The following sub-clauses describe the approach of each process module with its scope. This scope is specified by the list of entities of the associated data modules.

The specified process modules are:

- Communication of changes in Item Identification, Classification and Properties (PM-ICP): This process module is associated with the data module Item Identification, Classification and Properties (DM-ICP),
- Communication of changes in Assembly Structure Relationships (PM-ASR): This process module is associated with the data module Assembly Structure Relationships (DM-ASR),
- Communication of changes in Document and File Management Data (PM-DFM): This process module is associated with the data module Document and File Management Data (PM-DFM),
- Communication of changes in Shape Definition and Transformations (PM-SDT): This process module is associated with the data module Shape Definition and Transformations (DM-SDT),
- Communication of Authorization of Product Data (PM-APD): This process module is associated with the data module Authorization of Product Data (DM-APD).

  Note : The process module PM-APD comprises static authorizations, i.e. who has created / approved what and on which day. Changes of Authorization, dynamic authorization information and changes in work management are postponed to the future and are out of scope of the 'Part 1: Assembly Data Exchange'.

An individual engineering business process has to be mapped to a single process module or composed by a combination of process modules described in the sub clauses below. See clause VI for the engineering business process setup.

More sophisticated examples are given in Annex C.

The data dictionary in Annex A specifies the data scope per data module. The data dictionary is introduced in Clause 5.

When needed, companies may extend the recommendation with company specific data exchange guidelines according the template described in Annex B (see clause 8).

3.1. Process module PM-ICP : Communication of changes in item Identification, Classification and Properties

Description

This process module is involved if an engineering business process causes changes of identification data, classification data or properties for items where the relevant changes are restricted to individual items.

> **Note :** If the assembly structure is concerned, the process module PM-ASR described in 4.2 is used in addition to this module.

The associated data module Item Identification, Classification and Properties (DM-ICP) contains AP214 CC06 constructs necessary to describe the specified data scope :

Item identification

- *item*
- *item_version*
- *item_version_relationship with .relation_type = 'supplied item'*

- *alias_identification*
- *design_discipline_item_definition*
- *person_organization_assignment with role = 'id owner'*
- *organization*
- *application_context*

Item classification

- *specific_item_classification*

Item property

- *item_property_association*
- *property_value_association*
- *property_value_representation*
- *property*
- *property_value*
- *string_value*
- *numerical_value*
- *value_with_unit*
- *unit*
- *value_limit*
- *value_range*
- *material*
- *material_property*
- *recyclability_property*
- *cost_property*
- *mass_property*
- *quality_property*
- *recyclability_property*
- *general_property*

Note : This data module uses a subset of Units of Functionality (UoF) S1 and PR1 of the AP214 ARM.

3.2. Process module PM-ASR : Communication of changes in Assembly Structure Relationships

Description

This process module is involved if an engineering business process causes changes of the logical structure of an assembly.

Note : This assembly structure corresponds to the traditional engineering and manufacturing bill-of-material (BoM). The structure is built up by relationships between instances of *design_descipline_item_definition* interpreted as a specific (single or quantified) usage occurrence of the constituent item within a parent assembly.

The associated data module Assembly Structure Relationships (DM-ASR) contains AP214 CC06 constructs necessary to describe the specified data scope :

Assembly Structures

- *assembly_definition*
- *assembly_component_relationship*
- *next_higher_assembly*
- *item_instance*
- *single instance*
- *quantified_instance*
- *selected_instance*
- *item_definition_instance_relationship*
- *property_value*
- *numerical_value*
- *value_with_unit*
- *unit*
- *value_limit*
- *value_range*

Note : This data module uses a subset of Units of Functionality (UoF) S3 of the AP214 ARM.

3.3. Process module PM-DFM : Communication of changes in Document and File Management data

Description

This process module is involved if an engineering business process causes changes of document management data. If the changes are communicated by means of the exchange of external files, e.g., CAx files or office files, this process module is concerned with exchanging file management data.

Note : A managed document is under PDM revision control, and may distinguish various representations of a document version. The document version represents the minimum identification of a managed document under revision control. A representation of a document means an actual instance of the document and may optionally be associated with one or more constituent external files.

Note : An external file is not managed with PDM revision control or document representation definitions. An external file is an external reference that may be associated with other product data. While managed revision control representing multiple versions and version history is not available for external files, external files may have an optional version identification providing a string labeling the version of the file.

The associated data module Document and File Management Data (DM-DFM) contains AP214 CC06 constructs necessary to describe the specified data scope :

Document management

- *document_assignment*
- *document*
- *alias_identification*
- *specific_document_classification*
- *document_version*

- *document_version_relationship*
- *document_representation*
- *digital_document*
- *document_structure*
- *person_organization_assignment with role = 'id owner'*
- *organization*
- *document_file*

File management

- *digital_file*
- *external_file_id_and_location*

Document properties

- *document_content_property*
- *document_creation_property*
- *document_type_property*
- *document_location_property*
- *document_format_property*
- *language*
- *value_with_unit*
- *numerical_value*
- *unit*

Note : This data module uses a subset of Units of Functionality (UoF) E1 and S1 of the AP214 ARM.

3.4. Process module PM-SDT : Communication of changes in Shape Definition and Transformations

Description

This process module is involved if an engineering business process causes changes of the geometry of an item, i.e., modifications of CAD models or changes of the geometric structure represented as transformations.

Note : The detailed geometry of an item shape is typically in scope of CAD systems. Rather, the externally defined geometry (CAD model) is identified by a PDM system as an external representation of the item shape that is related to the item.

Note : A transformation in this context means the positioning of items or assemblies in the coordinate system of the next higher assembly. ISO 10303 AP214 CC06 provides several methods to define a transformation. This recommendation is independent of the preferred method.

The associated data module Shape Definition and Transformations (DM-SDT) contains AP214 CC06 constructs necessary to describe the specified data scope :

Item shape

- *external_model*

- *external_geometric_model*
- *external_picture*
- *person_organization_assignment*
- *organization*
- *item_shape*
- *shape_element*
- *model_property_association*
- *model_property_value*
- *property*
- *general_property*

Transformations

- *geometric_model_relationship*
- *geometric_model_relationship_with_transformation*
- *transformation*
- *transformation_2d*
- *transformation_3d*
- *cartesian_coordinate_space*
- *cartesian_coordinate_space_3d*
- *cartesian_coordinate_space_2d*
- *unit*
- *shape_description_association*
- *geometric_model*

Note : This data module is defined as a subset of Units of Functionality (UoF) E1, PR1 and S1 of the AP214 ARM.

3.5. Process module PM-APD : Communication of Authorization of Product Data

Description

This process module is involved if an engineering business process requires the static authorization (usually of changes) of product data, i.e., without work management information and approval relationships.

Example : If any of the other process modules is involved and the engineering business process requires, e.g., the identification of the creator or approval information in addition to the identification of an organization as id-owner, this module is involved.

Note : Changes of Authorization and changes in work management are intended to be treated in future and are out of scope of the 'Part 1: Assembly Data Exchange'.

The associated data module Authorization of Product Data (DM-APD) contains AP214 CC06 constructs necessary to describe the specified data scope :

Person and organization

- *organization* (with other role than 'id owner')
- *date_and_person_organization*
- *person*
- *person_in_organization*

Date and time

- *date_time*
- *date_time_assignment*
- *date_and_person_assignment*

Approval

- *approval*
- *approval_status*

and

- *work_order*
- *activitiy*
- *activity_element*
- *project*

Note : This data module is defined as a subset of Units of Functionality (UoF) S1 and S5 of the AP214 ARM.

4. Data dictionary

The data dictionary describes the data modules. The relevant data are identified as a subset from ISO 10303-214 ARM CC06 as being mandatory or optional according to the definitions in Rule 5 and Rule 6 (see clause 6.4).

The data dictionary (see Annex A) describes the selected objects, their attributes and, if necessary, predefined or recommended attribute values.

The structure of the data dictionary table is defined as follows (see) :

- Column 1: Object Id, sequential numbering of selected entities and their attributes within a data module,
- Column 2: Name of the entity or of an associated attribute (begins with leading dot),
- Column 3: Short definition of the entity or attribute,
- Column 4: References to other entities in the data dictionary,
- Column 5: Examples of recommended values or complete list of allowed values (for attributes only),
- Column 6: Remarks and/or rules on entity and attribute level. The specification of a default value means that the related AIM entity does not allow an empty value', i.e. in case no user data is available for this attribute the default value is used,
- Column 7: Status of entity / attribute according to AP214 CC06 subset,
- Column 8: Recommended status of entity / attribute with respect to defined use cases. The element receives the status mandatory (m) or optional (o). That means, if an entity has the status 'm' then the instantiation of the entity and at least its mandatory attributes is absolutely required for a corresponding data module, while entities with status 'o' do not have to be instantiated. If an

optional object is chosen to be relevant for a certain data module, e.g., because of a user requirement, this entity and at least its mandatory attributes have to be instantiated, too,

- Column 9: Exceptions differing to the recommended status for the entity,
- Column 10: Description for the exception,
- Column 11: Reference to the corresponding PDM Usage Guide [3] section to specify the relevant recommendations for AIM based implementations.

Object ID	Name	Description	reference to	Examples or complete list of recommended values	Remarks/Rules	in AP214IS COs	recommended	Exceptions: recommended	Exceptions: Description	Reference to PDM Usage Guide
APD1	Approval	is a judgment concerning the quality of those product data that are subject of the Approval. An Approval represents a statement made by technical personnel or management personnel whether certain requirements are met. The absence of approval information does not imply any approval status by default.					m			Chapter 13.2 Approval
APD2	.is_applied_to S[1:?]	specifies the objects to which the Approval is assigned. There shall be at least one object that the Approval is_applied_to.	ICP9 OR ICP18 OR DFM13 OR DFM22			m	m			
APD3	.level	represents the aspect for which the object subject to approval, by reference as 'is_applied_to', is endorsed.		e.g. 'design', 'disposition', 'equipment order', 'planning'		o	m			
APD4	.status	indicates the maturity of acceptance of the object, the Approval is applied to, for the specified Approval.level	APD9		The approval levels are defined within a given life cycle stage view and are typically dependent upon the approval cycles within an individual business process. For each defined .level, generally there is a defined sequence of values for the status, usually leading towards the status 'approved' indicating achievement of the particular level.	m	m			
APD5	.is_approved_by S[0:?]	specifies personnel responsible for the Approval and the dates of the Approval.	APD52		In cases, where more than one person or organization has to sign off on the Approval, usually the last sign-off date corresponds to the date specified as the 'actual date'.	o	o			
APD6	.actual_date	specifies the date when the Approval actually became valid. If this attribute is absent, the approval has not yet occurred, i.e., it is pending.	APD76			o	o			
APD7	.planned_date	specifies the date when the Approval is or was supposed to be performed.	APD76			o	o			
APD8	.scope S[0:?]	specifies the set of Organization objects for which the Approval is valid.	APD55			o	o			
APD9	Approval_status	is the state of acceptance of some product data.					o	m[1]	if APD1 is selected	Chapter 12 Alias Identification

Figure 2 – Fragment of Data Dictionary as defined in Annex A

5. Engineering Business process setup

This clause describes the principal way to establish a product data exchange according to this recommendation independent of whether it is done manually, by the user interface, or (partially) automated.

Engineering processes are typically set up by PDM data exchange coordinators based on the context of the project and participating partners; thereafter the end user would normally have a choice of processes predefined for use in his project.

5.1. Engineering Business processes

The common product data exchange practice can cover a wide range of engineering business processes. The implementation details of these engineering business processes will depend on the system architecture and the internal engineering processes at the collaborating partners' sites. This sub clause provides an overview on several frequently executed engineering business processes. Nevertheless, this overview is not claimed to be complete or definitive. Two engineering business processes are illustrated more in detail in *Figure 3* and *Figure 4*.

Note : These following two examples assume that the supplier is only responsible for the components P5 and S-A4 including its subcomponents.

Establishing a collaboration between OEM and Supplier

The first example (*Figure 3*) illustrates the establishment of collaboration between OEM and Supplier where the OEM sends two product data 'placeholders' (S-A4 V1, P5 V1) for a new subassembly and a new detail part, which have to be designed by the supplier as components of an existing assembly. The placeholders are accompanied by informative product data, i.e., product data that are not allowed for modification by the supplier. The Supplier sends back the product data of the detail part and the subassembly including the product data for all components of the subassembly.

Supplier delivers a change in assembly design (addition of completely new components)

The second example (*Figure 4*) illustrates the integration of a single item into a subassembly (S-A4 V1), representing a completely new component (P10 V1) resulting in a new version of that subassembly (S-A4 V2).

Furthermore, engineering business processes may be :

- Supplier delivers a change in part or assembly design (update of part or assembly data),
- Supplier delivers a change in part or assembly design (removal of components),
- Supplier suggests location changes outside his responsibility,
- OEM notifies supplier about changes within the shared area (notification only),
- OEM notifies supplier about changes within the shared area (notification plus work request).

Note : Part 2 of the recommendation for product data exchange will handle the last three processes.

The following sub clauses generally describe how engineering business processes can be set up. Annex C describes in detail how the engineering business processes can be set up for the examples in *Figure 3* and *Figure 4*.

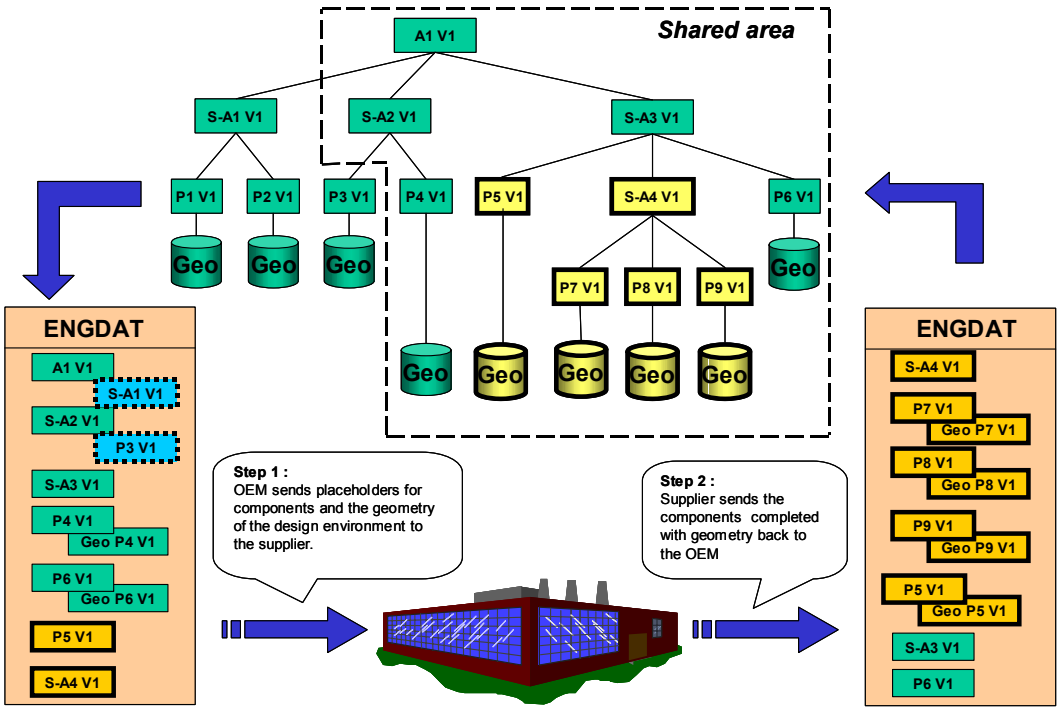

Figure 3 – Establishing a collaboration between OEM and Supplier

Figure 1 and *Figure 4* use the several symbols with the following meaning :

- A1 V1 exchanged object, data unchanged,
- Geo exchanged geometry file, data unchanged,
- *P3 V1* object just referenced (reference mechanism as defined in Annex D),
- S-A4 V1 exchanged object, data changed,
- Geo exchanged geometry file, data changed.

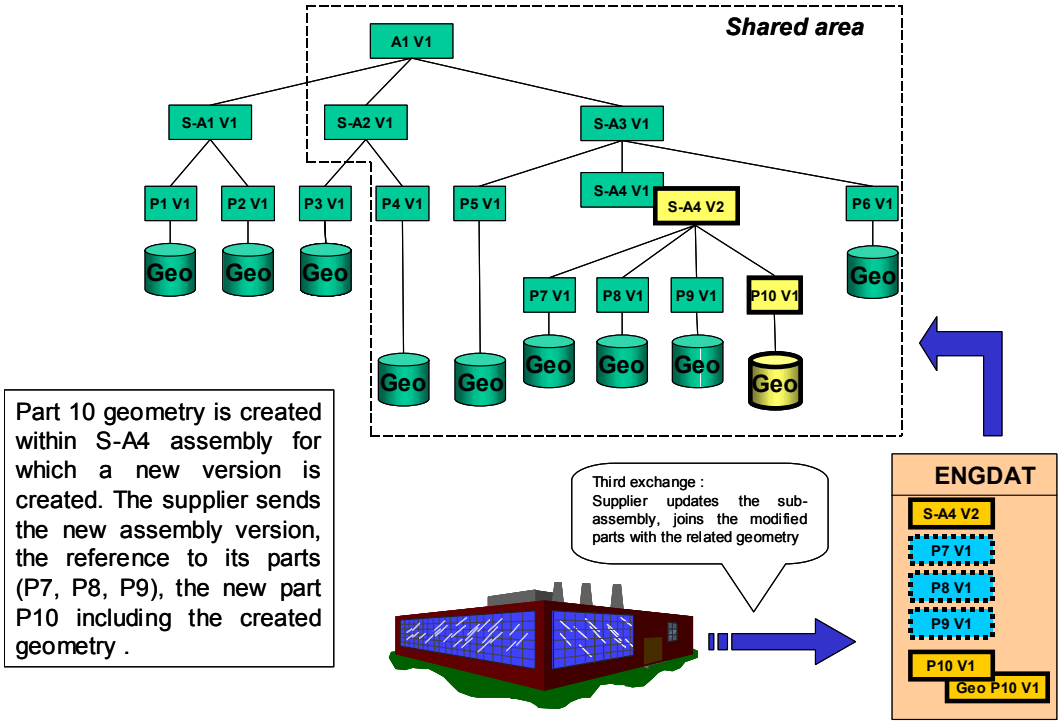

Figure 4 - Supplier delivers a change in assembly design (addition of completely new components)

5.2. Instructions for engineering business process setup

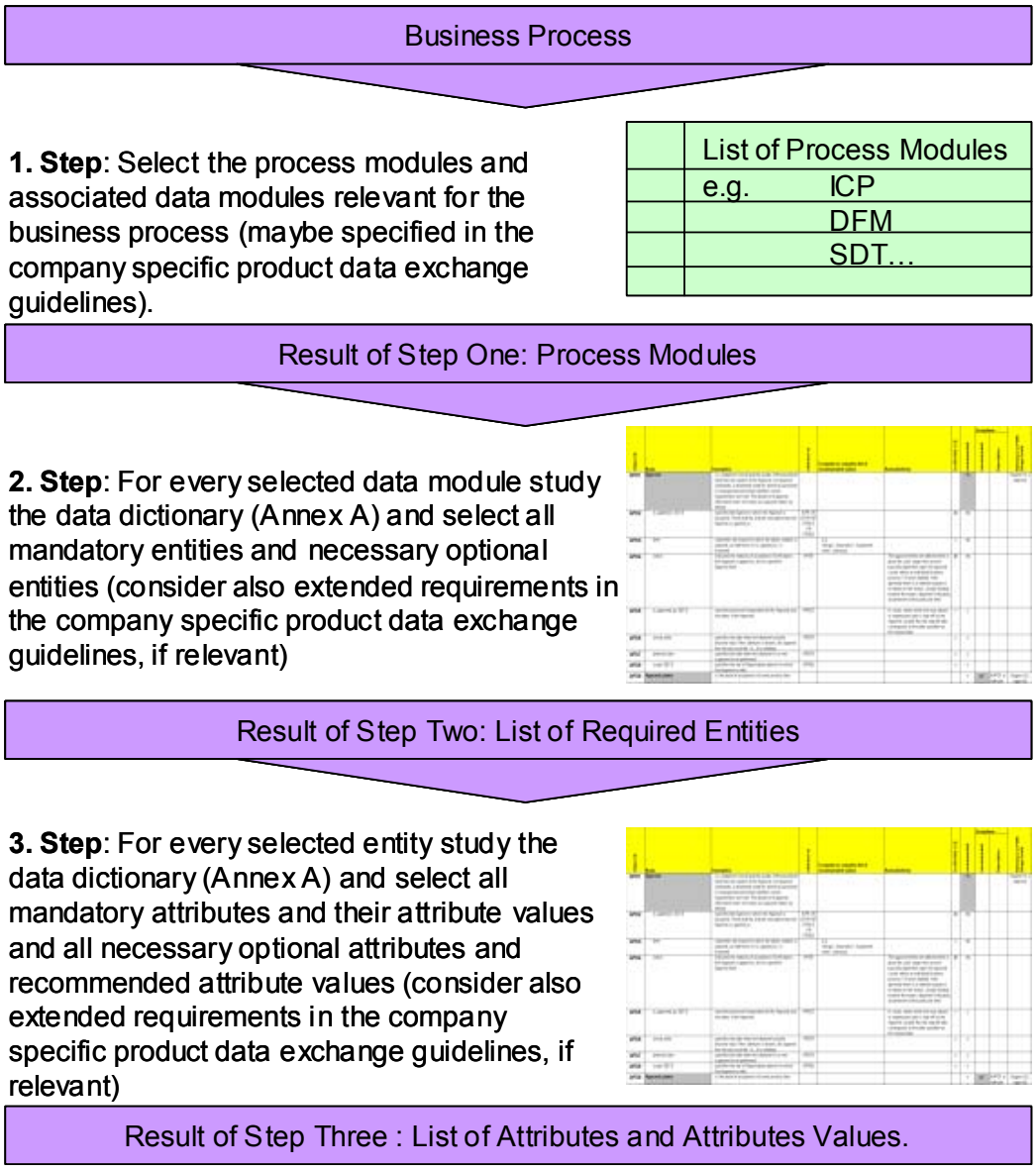

Figure 5 – Instruction diagram for the specification of an engineering business process setup

Figure 5 illustrates the steps to set up a new engineering business process for product data exchange, i.e., the steps to find out which entities and attributes have to be instantiated depending on the engineering business process.

Note : Since step Two and Three are based on the same data dictionary in Annex A they can be combined in one step

5.3. Step 1 : Analyze engineering Business Process and identify relevant Process Modules

The sender needs to decide which process modules (may be one or more) are necessary to compose an individual engineering business process. The result is the combination of necessary process modules with their associated data modules to support the engineering business process. Nevertheless, it is possible that an individual company specified additional process modules, which are not in scope of the current version of recommendation, e.g., dynamic work management information.

Rule 1 : A process module is involved if an engineering business process causes the communication of changes of data covered by that associated data module.

Rule 2 : If a process module is involved it's associated data module has to be completely instantiated regarding the related scope of product data.

> **Example :** If an assembly with four components is communicated with document and file management data for one component, the process module PM-ICP has to be completely instantiated for the assembly, i.e. all components have to be exchanged, and the process module PM-DFM has to be completely instantiated for the concerned component. (not for the entire assembly).

Rule 3 : If concerned data is only needed to give some more sophisticated information but has not been changed, the appropriated process module is not necessarily involved and the concerned entity instances could be referenced (see clause 6.5).

> **Example :** If the engineering business process causes modifications of the geometry of a single item resulting in a new document version and a new item version without any influence on the assembly structure, the data modules DM-ICP, DM-DFM and DM-SDT would be aggregated. See Annex C for more sophisticated examples.
>
> **Example :** If item identification is only used to identify the item to which exchanged document management information is associated, but the item identification itself has not been changed anyway, the item could be referenced. The process module PM-ICP is not necessarily involved.

5.4. Steps 2 and 3: Specifying Entity Instances, Attributes and Attribute Values

The sender needs to decide which entities and attributes (with recommended attribute values) are necessary to support the engineering business process. The result is a list of required entities and their required attributes as a subset of all objects, which are in scope of the selected data module. It is recommended to use the data dictionary according the following rules :

Rule 4 : If a process module is involved in the product data exchange, all mandatory entities and all their mandatory attributes of the associated data module have to be specified.

Rule 5 : STEP preprocessors must populate a **mandatory entity**. A mandatory attribute must be instantiated by STEP preprocessors if the appropriate entity is instantiated. A STEP postprocessor shall expect and interpret all mandatory entities and attributes within a STEP file as valid user data.

Rule 6 : An **optional entity** may be instantiated by a STEP preprocessor but if it is instantiated at least all mandatory attributes have to be populated. An optional attribute may be populated by a STEP preprocessor. A STEP postprocessor shall interpret optional entities and optional attributes as valid user data if they appear in a STEP file.

> **Note :** The data dictionary provides a committed specification of mandatory and optional entities and attributes for each process module (see Annex A).

Rule 7 : If a mandatory entity or attribute according to the definitions in Rule 5 and Rule 6 is missing for an involved data module, it is an error.

Rule 8 : If a mandatory entity or attribute is not available, an explicit 'dummy' element is required, i.e., an instance of an object with its mandatory attributes having 'dummy' values, according to the following specification :

- '' (empty string) if the sender has currently no value for an actually mandatory attribute, considered to be **valid user data**,
- /NULL if the sending system does not manage or the processor does not export a specific mandatory attribute,
- In case a concerned entity is classified as a reference (see clause 6.5), '' (for mandatory attributes) and $ (for unused optional attributes in a STEP physical file) are **not** considered to be **valid user data**.

Rule 9 : In case of a combination of data modules a mandatory flag overwrites an optional flag.

Rule 10 : Every time a data module is reused, every single data, i.e., instance of an entity, attribute or an actual attribute value, needs to be resent or at least be referenced (see clause 6.5). This is valid even if the concerned entity or attribute is declared as optional within the data dictionary. Otherwise the data that has been available before is interpreted as being removed.

Example : If an item property that has been sent earlier has changed and the data module DM-ICP is involved in a data exchange, all properties of that item have to be instantiated and not only the changed property.

Example : If an item has been sent earlier and only one of three associated documents has been changed resulting in a new item_version, at least the process modules PM-ICP and PM-DFM with their associated data modules have to be involved considering :

1. For the changed document, all data within the scope of the data module DM-DFM have to be completely instantiated. The other two documents must either be referenced or instantiated completely as well (see clause 6.5).
2. All other unchanged user data related to the item, that has been instantiated once before, e.g., item properties have to be completely instantiated again.

5.5. Reference Mechanism

A reference is a means to identify a single instance or a group of related instances at object level.

Rule 11 : A reference is explicitly classified by a general classification.

Rule 12 : A sender has to provide a minimum of entities, attributes and attribute values, which are necessary to uniquely identify the referenced object. Otherwise a receiver will interpret the same object as a new one because of different identifiers.

Rule 13 : Other data associated to the referenced instance or group of instances shall not be instantiated.

Rule 14 : A reference is applicable to an existing object at sender's and/or receiver's site or represents an object as a 'placeholder', i.e., an instance may be referenced at a certain time of data exchange if it was exchanged before or if it will be exchanged later on.

Annex D provides a detailed description of the reference mechanism by means of instantiation diagrams.

The application of the reference mechanism is recommended for items, documents, external files.

Example : An assembly structure is exchanged, but (some) components are only referenced and not exchanged including their complete definition at the same time. Nevertheless, the number of relationships between the assembly and components has to represent the complete list of components of the assembly, independent whether an individual component is referenced or its complete definition is exchanged.

Example : A managed document may be exchanged, but the representing file is only referenced (including source property).

Example : A managed document may reference an item, which it is describing, without sending the item data at the same time.

6. Systems and technologies for Product Data Exchange

6.1. Application system categories

This clause deals with the application system categories, which may play a role within the product data exchange process. Every company or project has to identify the concrete applications being used and has to check the necessary functionality.

The following application system categories may be relevant for product data exchange :

- Data generating systems like CAx systems or office applications. These applications provide files in neutral or native formats, which are referenced as external files associated to PDM data.
- Team data management (TDM) systems for the management of specific application data. In the CAx environment these applications are usually provided by the vendor of CAx systems for the management of the CAx data. These applications may provide additional structure, positioning, configuration and property information, mainly on the geometric level.
- PDM systems, which manage item, document and geometry information**, and associated workflow processes**. A PDM system is characterized by a distinction between items and documents and may be used for several data generating systems (e.g., several CAx systems). These applications may provide all kinds of product data.
- Smart PDM Tools, which are able to read, write, edit and visualize the content of a PDM STEP file and may be used as a communication front-end. A Smart PDM Tool has no persistent database. These applications may provide the same data as a PDM system.
- Applications for Digital Mock-up (DMU) processes, which build up virtual models of products or portions of products to provide visualization, analysis and simulation. These applications are mainly receivers of the geometric assembly information. They may provide geometric data and geometric structures with positioning information.
- Applications for data transfer (EDI), routing, packaging with ENGDAT, translation, checking, notification, archiving, etc. These applications enable, optimize and document the physical data transfer as part of the product data exchange process.
- Bill-of-material (BoM) and product configuration applications (e.g., ERP systems). These applications may deliver part structure, effectivities and configuration data.
- Project management and work flow applications. These applications may deliver project, work, and change management data.
- Design releasing / configuration management systems, which indicate what activities (purchasing, prototype, soft/hard tooling, etc.) are authorized.

- Issues tracking / Engineering Work Order tracking systems, which identify problems after authorized activity has begun and track resolution.

Figure 6 provides typical system architecture for assembly data exchange.

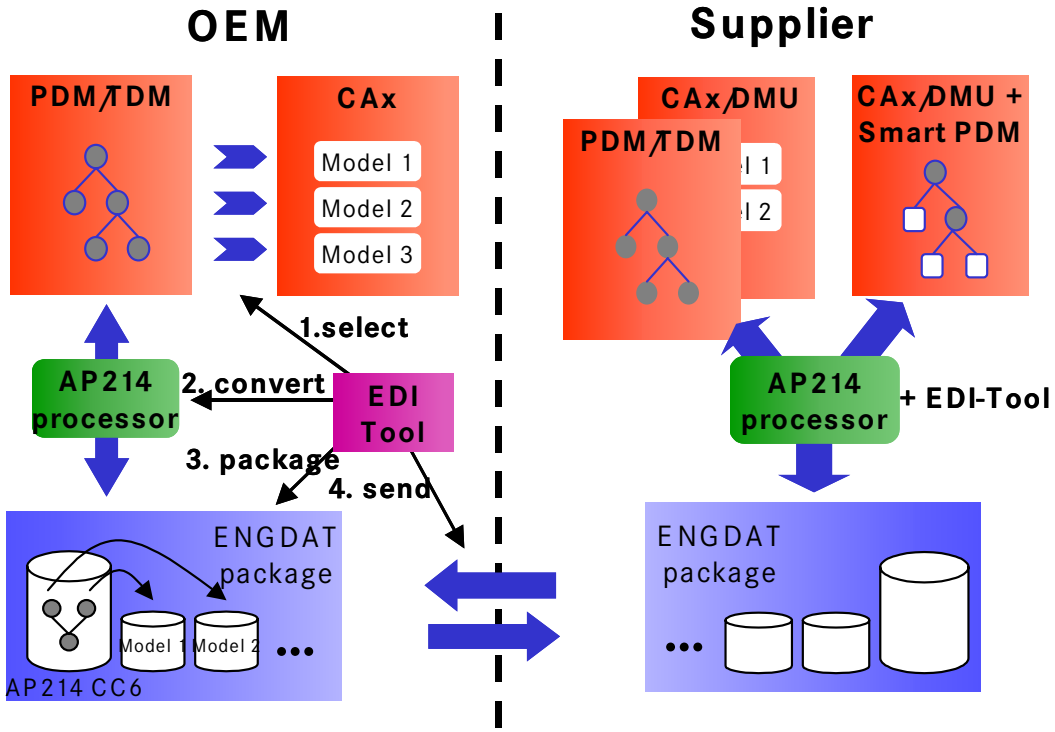

Figure 6 – Typical system architecture for assembly data exchange

6.2. Data transfer technology

This clause provides information for the data transfer, focused on asynchronous data transfer methods used in Europe. Other relevant communication technologies are mentioned shortly.

File based asynchronous data transfer is used for the exchange of product data. In principle, the sender transmits files via an offline medium, e.g., a tape, or via online file transfer. This results in a replication of the sender's data at the receiver's site. Sub-clause 7.3 provides a detailed description of the used formats.

Synchronous data access and data sharing enable distribution of data to different databases and locations. There is a minimum of redundancy to have a single set of consistent data. Used formats and programming technologies are CORBA and especially the PDM Enablers for administrative product data standardized by OMG (www.omg.org) or XML standardized by W3C (www.w3.org) with an appropriate data schema.

Web clients have - similar to the synchronous data access and data sharing - access to the data of a server system, e.g., via the Internet. Web clients have no persistent data. The user is logging into an application running on a server and may update the data on the server. Technologies used for this kind of applications are usual Internet technologies like JAVA, JAVASCRIPT, HTML and/or XML. Web clients may be server application system specific or neutral. An Approach for the access of product data by means of neutral Web clients is tackled by the PDTnet project (www.pdtnet.org). Further information of the access by means of neutral Web clients is given in the PDTnet Whitepaper [2].

6.3. Data types within an exchange package

In general, there are two types of data sent in a file based asynchronous data exchange scenario within a data package :

- Application data of specific user applications,
- Transfer data for the delivery note.

Application data comprise administrative product data mainly coming from PDM systems and product data generated by CAx systems or other applications.

The transfer data include information about data routing and processing, e.g., sender, receiver, data formats, files contained in the data package, or encryption.

Administrative product data (PDM data)

To exchange administrative product data there is only a single standardized data format available: ISO 10303 (STEP). PDM data shall use the format defined by ISO 10303-214 'Core data for automotive mechanical design processes', also referred to as STEP AP214, Conformance Class 6 (CC06).

> **Note :** The required data as defined in clause V are mainly within the limits of the PDM Schema that was created by the major STEP centers worldwide, i.e., PDES, Inc. in the U.S., ProSTEP Association in Germany, and JSTEP in Japan. The PDM Schema is completely harmonized with STEP AP214 CC06.

Within a data package, the PDM file has to reference all data files of the user applications, which are included in the data package. The content and use of the relevant STEP data schema is described in Annex A.

Data of user applications

The format of the data from the generating user applications like CAx systems or office applications is not restricted. They may be in a native or a neutral format, e.g., STEP for CAx data. All data files of user applications have to be referenced by the PDM file within a data package. Only documents referenced by PDM may be attached to a data exchange.

Transfer data

Typical information related to the data transfer is :

- Sender and receiver, especially routing information on the receiver's site,
- Used systems and processors to create the data,
- Contents of the package, i.e., a list of files,
- Formats of the files,
- Purpose of sending, i.e., project name or work order,
- Encryption method,
- Dependencies on other data packages.

The recommended standard for transfer data in the European Automotive Industry is ENGDAT by ODETTE (www.odette.org). The recommendations by the German VDA are divided into five parts (VDA 4951). The fifth part of the VDA 4951 describes the aspects of using ENGDAT in combination with STEP PDM data.

Furthermore there are other formats to represent subsets of the above listed transfer data. Examples are :

- ISO10303-232 (STEP AP 232) by ISO (www.nist.gov, www.pdesinc.aticorp.org),
- XMTD by SASIG.

Note : There are activities by UN/CEFACT and OASIS to migrate several EDIFACT messages to XML called ebXML.

In addition to these standards there are several company or consortium specific formats used in practice to represent the transferred data.

7. Company specific extensions of recommendation

This recommendation is harmonized by specifying process modules, which decompose engineering business processes. Each process module is closely related with a minimal recommended subset of STEP AP214 CC06 data objects with recommended attributes and attribute values.

Nevertheless, several OEM (and maybe some 1st tier suppliers) have specific engineering business processes (business cases) and requirements for the instantiation of data and for values of the attributes to support these engineering business processes. These requirements are often depending on the engineering business processes and the system architecture and can hardly be harmonized.

In order to support a sophisticated product data exchange practice, company specific product data exchange guidelines are necessary to provide a detailed documentation of company specific requirements for the product data exchange processes between a given company and its suppliers.

The company specific product data exchange guideline has to specify :

- Necessary functionality and technical constraints for product data exchange,
- Engineering business processes, which affect the product data exchange,
- Composition of engineering business processes by means of process modules,
- Extension of the number of mandatory entities and attributes, further restrictions concerning, e.g., document structures, assembly level depth,
- Predefined attribute values and the specific semantics of those values.

A company specific product data exchange guideline shall not suspend or relax any recommendation. Optionally, a company specific product data exchange guideline may reference accompanying internal and external guidelines, standards and recommendations.

7.1. Template for company specific product data exchange guidelines

This recommendation provides a template (see Annex B) for company specific product data exchange guidelines. The template specifies a recommended structure and an approach for the content with some examples. The goal is to achieve a harmonized look and feel, i.e., style and layout for the guidelines and an unambiguous common understanding of the usage in practice for implementers and system vendors. Company specific product data exchange requirements shall be documented as a clear, compact and practical guideline to support the system vendors, supplier's operations management and implementation project managers.

This recommendation contains only the template for the company specific product data exchange guidelines, not the guidelines themselves. Each company is responsible for the release of its own guideline, but it is recommended that all companies comply with the template.

A company specific data dictionary as an annex to the company product data exchange guideline collects the company specific terminology and provides a semantic description and the reference to the corresponding term or data object defined in the data dictionary in this recommendation.

To keep a company specific data dictionary compact only extended requirements and amendments to this recommendation shall be documented.

8. Bibliography

[1] PDM Schema Usage Guide (www.pdm-if.org)

[2] PDTnet Whitepaper (www.pdtnet.org)

Annexes

A – Data dictionary

B – Template for company specific Product Data Exchange guideline

C – Example for an engineering business process setup

D – Reference mechanism