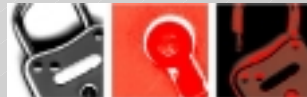


Developing a Customized STEP Implementation

in the context of PDM application domain



Dr.-Ing. Anna Wasmer
PD Tec GmbH
D-76131 Karlsruhe, Germany
Fon: +49-721-9658-675
Fax: +49-721-9658-263
E-Mail: wasmmer@pdtec.de
Web: www.pdtec.de

P|D|Tec.
Enabling Product
Data Technology

NASA STEP Workshop, Pasadena, CA, January 16/19, 2001

P|D|Tec.

Challenges of eEngineering communication

■ eEngineering requires high degree of communication

- Concurrent/ simultaneous engineering
- OEM - Supplier collaboration
- Use of catalogue/standard parts

■ Need for expanded life cycle management through

- Strategic partnerships
- Development cooperations
- Inclusion of key suppliers into business workflow
- Global environment

■ Exchange of 'just' geometry is not sufficient

- Exchangeability of product definitional information (e.g. product identification) and organizational information (e.g. approval) required

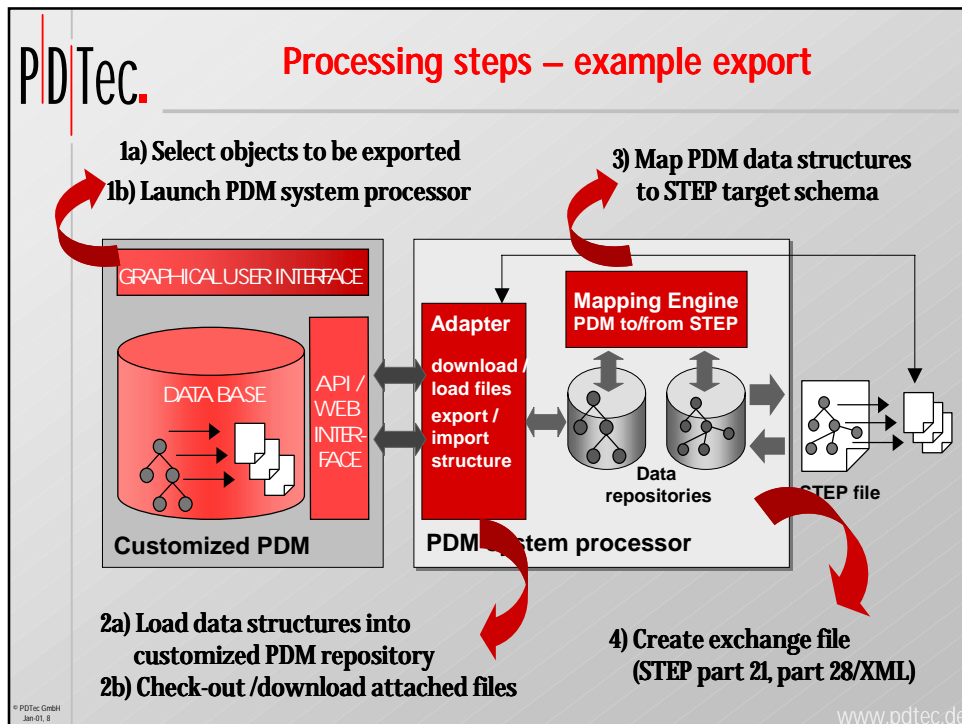
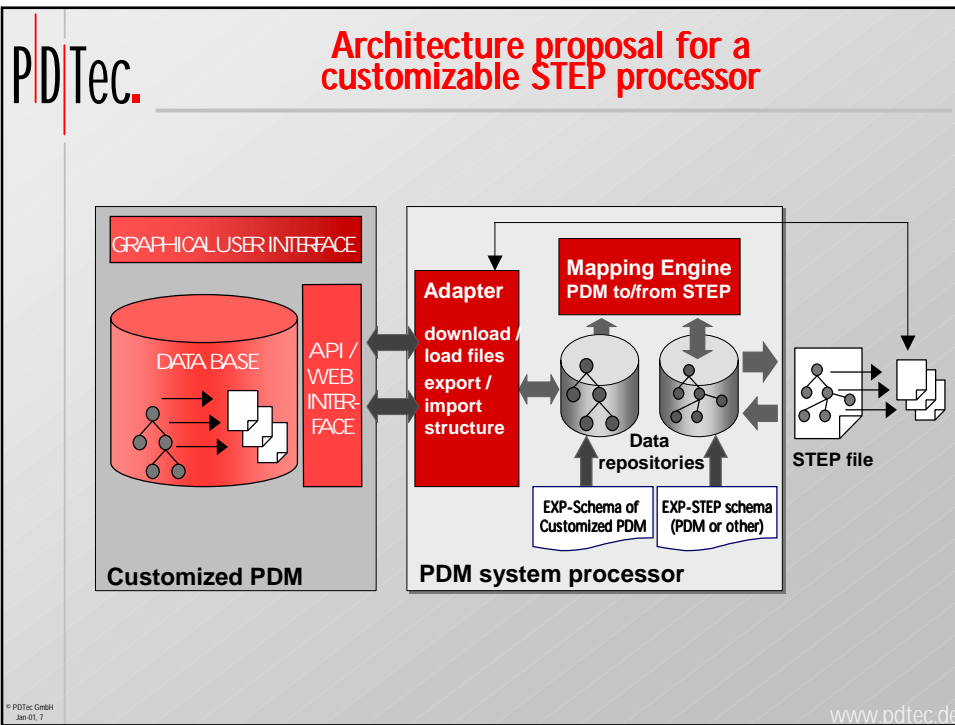
■ Data managed by PDM systems is a basis for eEngineering communication

- **Commercial PDM systems often are toolkits that need to be customized**
 - Customer specific data model
 - » adapt/extend the PDM standard data model
 - » build / implement own data model
 - Customer specific scope and constraints
 - » Terminology
 - » Attribute value domains (e.g. names of approval or lifecycle states)
 - » Part numbering systems
 - Customer specific business practices and processes , e.g.
 - » implement processes to support their specific business practices (object lifecycles, roles, signatures, ..)
 - » use of assembly structure or document structure to describe geometric relationships and transformation matrices between parts
 - Customized user interface

- **Initial processor customization according to system customization**
- **Continuous evolution and change of implementations**
 - E.g. new data types, attributes and attribute value sets
- **Different mapping „variations“ may be necessary to communicate with other applications (e.g. PDM, ERP, CAD,..) and organizations**
 - Different instantiation practices
 - Bilaterally agreed attribute value mappings
 - Support different views of the product, e.g. as designed, as built

- **Evolution and change of supported standards**
 - e.g. new revisions; migration to modules approach
- **Ability to use and support selection of different standards and target data models**
 - Different set of target modules or schemas (e.g. STEP APs)
 - Different representation format of target data set (e.g. STEP part 21 or part 28/XML)
- **Specific system environments**
 - e.g. network distribution, client-server, web based access, integration with exchange tools

- **System vendors / processor implementors**
 - Standard system processors have to be easily customizable
 - » customer specific data model, scope and constraints
 - » customer specific business practices and processes
 - » evolution and change of supported standards
 - » support of different standards/ target data models
- **User companies**
 - Use implementation technology that provide
 - » good support for initial processor development/customization
 - » good support for continuous evolution of processor according to PDM implementation phases
 - » flexibility to support different exchange scenarios and business relationships
 - » possibility to include different mapping „variations“ to optimize exchange with selected partners
 - » easy integration in changing system environments
 - » migration path to new standards and target data models



- (1) **Create data repositories**
 - Identify STEP target data models (PDM schema, APxxx, set of modules)
 - Generate EXPRESS representation of customized PDM data model
 - » e.g. tool for automatic extraction
 - (2) **Develop system adapter**
 - Define business practices and rules depending on specific user settings
 - » e.g. when to perform certain actions (e.g. check-in/-out, replace, add, etc.)
 - Business Logic implementation
 - Import /export data of data structures and attached files via API
 - (3) **Develop Mapping engine**
 - Mapping specification (EXPRESS-X)
 - Definition of extensible configuration tables e.g. attribute value mappings, bilateral agreements, etc.
 - Automatic generation of mapping engine
 - (4) **Additional functionality (e.g. PDM GUI extensions for data selection, browsing and editing, checking modules)**
- ➔ Steps (2), (3), and (4) can be parallelized

- **Structural data mapping language to**
 - allow an unambiguous specification of the relationship between models defined in EXPRESS (ISO 10303-11)
 - support mapping of data defined by one EXPRESS model to data defined by another EXPRESS model
- **Defined as extension of the EXPRESS language**
- **Available as part 14 of the SC4/STEP series standards**
- **Committee draft (CD) successfully balloted (Dec. 2000)**
- **10303-14 will go forward for registration as a DIS (Draft International Standard), after ballot comment resolution**
- **Generation of EXPRESS-X Map statements from textual mapping specifications (e.g. mapping tables)**

- **Explicit specification of one-way mappings between n source and m target EXPRESS schemas**
 - EXPRESS-X does not support the explicit definition of inverse mappings in the same mapping specification
- **Modular structure of mappings**
 - Mappings can be REFERENCED like EXPRESS SCHEMAS
- **Declarative approach of the specification**
 - Independence of execution order
 - Readability
 - Same level of abstraction as mapped data models
- **Procedural extensions**
 - Inclusion of predefined algorithms on source and target data
- ➔ **Clear separation between description and execution model**

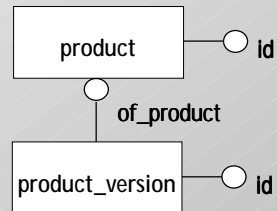
- Identification of source and target schemas
- Use of externally defined mappings and functions
- (Data-) Type mapping
- Declaration of constants (EXPRESS syntax) and target/view instances which are not directly related to source data
- Declaration of procedures and functions (EXPRESS syntax)
- **MAP and VIEW declarations**
 - **MAP**
specification of the mapping between pre-existing source and target schemas (entities)
 - **VIEW**
definition of a view schema (view entities) derived from one or more source schemas (source entities)

EXPRESS-X example

SOURCE SCHEMA:



TARGET SCHEMA:



EXPRESS-X SCHEMA:

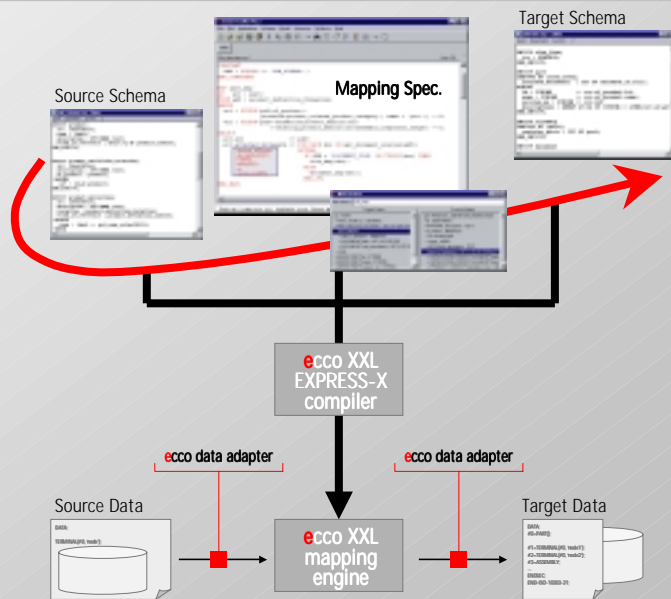
```

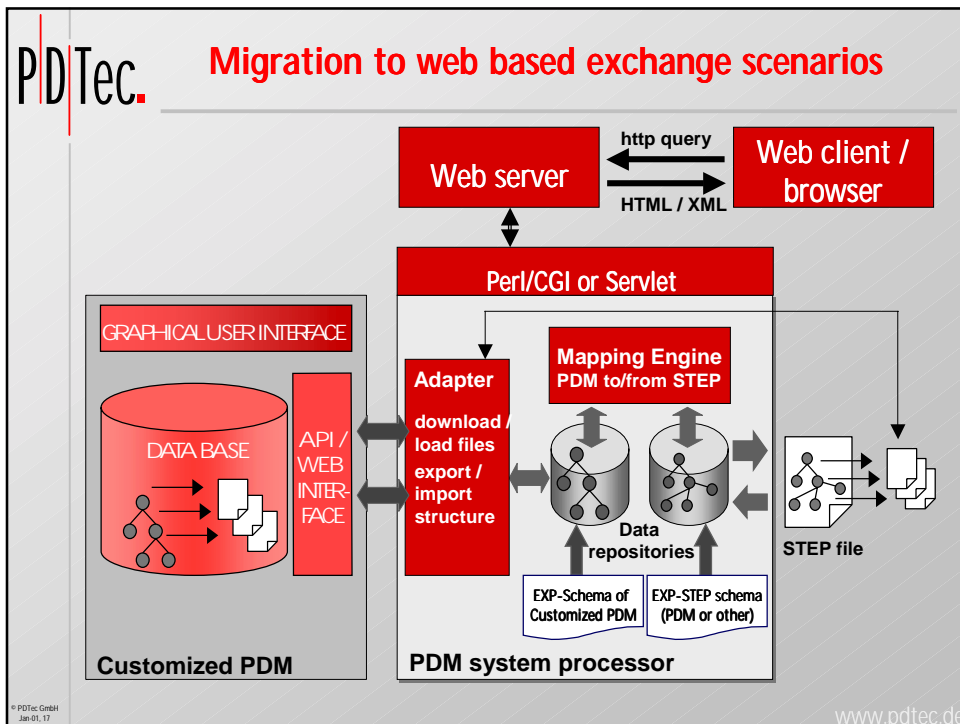
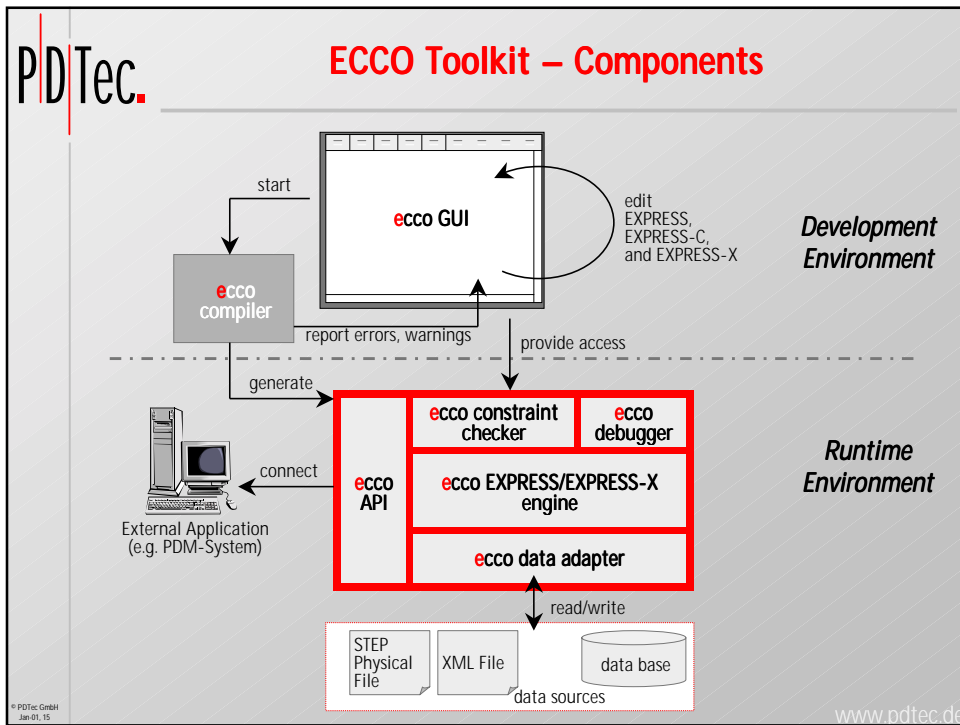
MAP product_map AS
  p : product;
FROM
  prt : part;
SELECT
  p.id := prt.partno;
END_MAP;
  
```

```

MAP product_version_map AS
  pv : product_version;
FROM
  prt : part;
SELECT
  pv.id := prt.version;
  of_product := product_map(prt);
END_MAP;
  
```

Automatic generation of the mapping engine using ECCO XXL





- **Advantages of system adapter and mapping engine as separate processor components**
 - Changes in system API's or technology replacement (e.g. web based interface) do not affect mapping engine
 - Reuse of sys. adapter for mappings to different target schemas
 - » Support of additional standards (STEP data models, or others)
 - Use of EXPRESS-X reduces development time for data mappings and allows easy adaptation of mapping engine
 - Independence of data representation format on target side
 - » Changes in target schema/ mapping do not affect system adapter
 - Parallelization of development and customization work
 - » use of different experts
 - Distribution in the network
 - Easy migration to new technology components (e.g. web)
 - Easy integration of additional development tools (schema extractor, schema comparison, graphical mapping tool)