Narrative summary/Highlights of the course

The course provides a cartography of standards supporting the area of product data exchange, sharing, and retention that support interoperability of enterprise applications in the context of Product Life Cycle process deployment, providing a categorization based on the interoperability framework: Enterprise, Knowledge, Information System, Ontology.
Objectives

- **Detailed description of different standards**
  - For Product data exchange, sharing and retention
  - within PLM strategy

- **Interoperability** of involved enterprise applications:
  - How they contribute

=> **Understanding of underlying ICT standards**:
- Comparison
- Complementarities

Objectives

The goal of the course is to provide a detailed description of the different standards that can support product data exchange, sharing and retention in a PLM strategy, and how they contribute to the interoperability of involved enterprise applications.

After completing the course, the participants will have a clear understanding of these standards and of the way to evaluate, compare and integrate them in an appropriate way to support a PLM strategy and the interoperability of enterprise application implied in such a strategy.

Who should attend?

The course applies to target-groups in IT, Domain knowledge/Industrial or Academic, such as IT expert, PLM expert.

Student requirements

Basic knowledge of industrial projects and tools. Knowledge of some standards used for data exchange and sharing, such as XML. Knowledge concerning Product Data Management.

Recommended precedence

None
Content

1. Introduction: Interoperability needs for PLM strategy for Networked Collaborative Product Development
   Standards
   - Current trends for Industrial Enterprises and organizations and Emerging Challenges
   - Standards
     • STEP application protocols
     • XML vocabulary
     • Given domain ontology
     • UML profiles
     • Standardized process
     • Standardized process or workflow modeling language

2. Associated technologies
   - Express, XML schema, OWL, MDA-MOF, CORBA and WEB services, BPMN-XPDL-BPEL

3. Associated actors and stakeholders
   - NCPD, PLM, Technologies

4. Comparative analysis
   - Advantages, drawbacks, differences, complementarities to support effective interoperability

5. Mapping and transformation
   - The different types of mappings and the different ways to implement them (e.g. XSLT, Express-X, MOF, EAIS)

6. Integration and federation
   - The different ways to integrate them

7. What brings ATHENA results: usage for a NCPD framework and platform
Current trends for the current trends and organization are reinforcing needs for extensive collaboration and interoperability between enterprise applications supporting the business processes.

1- Virtualization of the Product

Competition today is leading enterprises to *short the time-to-market*. It is particularly important to gain parts of a continuously changing and evolving market, but also for shareholders for which shorter industrial projects should allow better visibility and faster benefits. As production is optimized since several years, the phase where important reduction can be expected is the design phase. The idea was to develop new ways of working, based on parallelization of engineering task, also called **Concurrent Engineering**, and to provide computer aided design tools to the engineers for the multiple implied disciplines. Design offices are consequently organizing their work through production of **multidisciplinary e-models**, that will be the inputs of next phases (production planning, test elaboration, production, tests, support, exploitation…).

During the last ten years, Paper based models (2D) have been replaced by **electronic dynamic models**, that will be used for certification of future aerospace products. Due to the Information Technologies nature of these models, numerous challenges exist for product data exchange, sharing, retention and trust. In addition, the tooling for such e-Models is highly complex, and it is no more possible for enterprises to create their own in house software applications as software engineering is out of their core business activities. Such trends consequently imply usage of **software products**, creating a software editors dependency.
PDE2: Standards to support Interoperability for Product Life cycle Management

<table>
<thead>
<tr>
<th>Current trends for Industrial Enterprises and organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Virtualization of the Product</strong></td>
</tr>
<tr>
<td>- Competition leading to short the time-to-market</td>
</tr>
<tr>
<td>- Concurrent Engineering with multidisciplinary e-models</td>
</tr>
<tr>
<td>- Paper based models (2D) being replaced by electronic dynamic models</td>
</tr>
<tr>
<td>- Implied usage of software products</td>
</tr>
<tr>
<td>2. <strong>Virtualization of the enterprise</strong></td>
</tr>
<tr>
<td>- Competition leading to be focused on core business and high value activities</td>
</tr>
<tr>
<td>- Pushing usage of Commercial of the Shelves (COTS) (us In house software)</td>
</tr>
<tr>
<td>- Evolution of Partnership/Subcontracting network</td>
</tr>
</tbody>
</table>
|     - \# number of sub contractors of level 1 \\
|     - Process, methods and Software products heterogeneity |
|   - Complex interdependency between actors and information systems |
| 3. **Product Lifecycle Management from Requirements to recycling** |
|   - Early involvement of downstream activities            |
|   - For a more competitive product (easier to exploit and maintain, better support) |
|   - Product data and metadata to be manage in configuration |
| 4. **Virtual Aircraft for early involvement of downstream activities** |
|   - Through usage of Advanced Systems Simulation (VIVACE) |


2- **Virtualization of the enterprise**

Competition today is also leading enterprises to be focused on core business and high value activities, and to find sub-contractors, risk sharing partners or finally equipment suppliers (e.g. engine, landing gear...) that are the best in class for complementary activities or required components. For such group of enterprises, only one enterprise is the interface with the clients, who can have the impression that they are working with only one enterprise: a virtual enterprise. The underlying model is also leading to outsourcing in order to reduce final cost of the product. When applied on software development, it pushes usage of Commercial of the Shelves (COTS) that is preferred compare to house software development. It is important to point out that several drawbacks exist for virtualization of the enterprise. Pushed to the extreme, it leads the enterprises to be only empty shells as important part of the knowledge and know-how is out of the enterprise. Knowledge management and high level of expertise for collaborative methods is consequently more and more important. It also create Complex interdependency between actors and information systems. Today, current situation (globalization, price of the dollar) consequence is the Evolution of Partnership/Subcontracting network: sub-contracted activities will increase while the number of sub-contractor of range 1 will decrease. Within the virtual enterprise, there is a looser coupling than within the integrated enterprise that implies Process, methods and Software products heterogeneity. Well managed Collaboration and federation of enterprise applications are of a particular importance in such a context.

3- **Product Lifecycle Management from Requirements to recycling**

In order to have more competitive products, other approaches are focusing on reducing number of change and to have them at the earlier stage of product development: the latter a change is required, the more expensive it will be. The other idea is to have Early involvement of downstream activities, in order design right at the first time, and to consequently reduce number of potential future change requests in manufacturing, exploitation or support context. These approaches are also developed for a more competitive product, that will be easier to exploit, maintain and support than those of the competitors. The price of the product is no more considered alone, the price of associated services for exploitation and support are also considered. It is why what is called PLM or Product Life Cycle Management is today strongly being developed within industrial enterprises. As PLM implies a holistic view of the product and concerned actors/stakeholders all along the lifecycle of the product (that can have a duration of 50 years), management of Product data and metadata in configuration is of prime importance.
Emerging challenges

1. How to ensure efficient global Configuration Management and Product Information/data Coherency between different PDM Systems?
2. How to ensure efficient Product Data Exchange, Sharing and Long Term Retention supporting Business Processes?
   - Concerned business process to consider in PLM strategy are
     - Exploitation, Maintenance and Support
     - Change and Configuration Management for Aircrafts in operation
     - Traceability for Legal information
3. How to ensure seamless collaboration of designers and technicians despite heterogeneous environment (process, methods, applications and software products)?

4- Virtual Aircraft for early involvement of downstream activities

   Early involvement of downstream activities is the downstream activities can’t visualize what the future product will be. It was initially done by means of schemas or physical mockups, but these means were too weak to allow future users of product definition in the loop. It can be done today Through usage of Advanced Systems Simulation, that combines a coherent way static and dynamic digital mockups managed in configuration. It is today a domain of investigation that is covered by research projects such as the VIVACE project. Reinforcing the virtualization of the product, and within the context of virtual enterprise having a PLM strategy, usage of Advanced Systems Simulation will imply more complex needs for Technical Enterprise Applications interoperability.

In such a context, some challenges are emerging for next generation of applications in term of interoperability:

1. How to ensure efficient global Configuration Management and Product Information/data Coherency between different PDM Systems?
2. How to ensure efficient Product Data Exchange, Sharing and Long Term Retention supporting Business Processes?
   - Concerned business process to consider in PLM strategy are
     - Exploitation, Maintenance and Support
     - Change and Configuration Management for Aircrafts in operation
     - Traceability for Legal information
3. How to ensure seamless collaboration of designers and technicians despite heterogeneous environment (process, methods, applications and software products)?

Let’s detail these different challenges.
Data exchange and sharing

Today data exchange is ensured with standards dedicated to different domain:

- **STEP Application Protocols** for Computer Aided CAD and PDM tools
- **XML and schemas** for eBusiness
- **UML, XMI and profiles** for software/system design, eventually ISO STEP Application Protocol 233 for system engineering

**Product Data Sharing**

- Product Data Management Systems
- Important for:
  - geographical distribution
  - enterprise applications (numerous users)
  - Controlled product data managed in configuration

**Product Data Sharing consists in organizing all the data describing a product (documents, models, drawings, digital mockup elements) in order to support design, manufacturing and support processes.**

**Product Data Management Systems** are software applications that are enhancing Product Data Sharing between all the actors involved in the design, manufacturing and support process. They are Important:

- For geographical distribution
- For organizing the work of the different involved actors (It is why PDMS are considered as enterprise applications _ numerous users and support of the enterprise business process)
- For Control of product data managed in configuration

But important needs exist today for:

- Integration within enterprises (Enterprise Application Integration)
- Business to Business collaboration for extended enterprises, virtual enterprises, Globalization
- Federation of PDM Systems

Several Standards are being developed for Product Data Sharing and Product Data management systems:

- Schemas (PDM Module) and API (Standard Data Access Interface)
- Interfaces for distributed systems (PDM Enablers)
- Services (PLM Services)
- Change and Configuration Management
- Workflow

Definition of remote data access, remote interfaces and remote services are important for integration of the PDM systems with the other applications of the enterprise, in particular Computer Aided Design and Manufacturing tool. Needs for Business to Business collaboration are growing and are required by extended enterprises, virtual enterprises and globalization.
But some gaps exist when willing to support PDM systems federation:

1. The Business Domain standards for definition of business data, interfaces and services are highly modeling paradigms and technologies dependant:
   - Object paradigm and CORBA for PDM Enablers – messages ensuring access to data are defined in IDL at definition time. At operation time, the ORB will ensure access to data through marshaling/un-marshalling of internal data structures of interconnected systems using an internal hidden binary format defined by the Internet Inter Orb Protocol plus different specified binding of IDL to existing programming languages for data/object structures. The services are available as remote procedures. A proxy is defined at client size, object adapters are ensuring the link with implementation objects and proxy at respectively server and client side. Links with business semantic is ensure at interface level from IDL definitions using stub and skeleton respectively at client and server sides.
   - Service and WEB technologies from W3C and OASIS for PLM services - services are set of operations with inputs and outputs, defined in Web Service Definition Language, itself defined in XML. The input and output types are usually defined by means of XML schemas and the messages are structured using Simple Object Access Protocol.

2. Insufficiency of current workflow systems and workflow standards
   PDM/PLM important business processes are change and configuration management process, that are defining context for design activities and workers. Activities and process are defined at different level:
   - Product data information level by means of PDM application protocols (PDM Module). They are important to store for traceability and affection of agreed activities i.e. attached to a change order
   - Change Management Workflow model, as workflow engines usage ensures
     - efficient logistic of work items within extended enterprise with remote worker (in order to short the design time)
     - parallelization of tasks for concurrent engineering
     - Contextualization and control of activities leading to consult, create, modify or erase product data. It is of particular importance for traceability and quality

But how to proceed for federation of PDM Systems? It is today about impossible to interconnect workflow systems for Cross Organization Change Management Process as different PDM and Workflow systems are used. For enactment of Business process, numerous standards or standalone solutions exist, based on different paradigms:
   - workflow, service choreography, service composition
   - Activity, flow or state oriented
## Configuration Management Process and workflow

**PDM/PLM importance of change and configuration management process:**
- Context for design activities and workers

**Activities and process defined at different levels:**
- Product data information level (PDM Module)
  - Traceability
- Change Management Workflow for controlled data
  - Team work
  - Configuration Management

As execution process engine is communicating with numerous heterogeneous systems, it is important for them to support different underlying technological framework for accessing data, services and interfaces. It is also important to exchange or share relevant data for the processes. Numerous standards are proposed that are to be studied and analyzed in order to validate their relevance for designing and executing cross organization business process for federated PDM.

The training session is aiming to share feedback of such study and analysis coming from implementation of ATHENA pilots for Networked Collaborative Product Development.

Which standards are related to Configuration Management Process and workflow?

First change and configuration management are within the scope of PDM/PLM approaches, as an importance artifact, as PDM and PLM approach are formalizing the context for design activities and workers: organization of what they are providing, linked to a decision process.

Within PDM systems, Activities and process are defined at different levels:
- **Product data information level (PDM Module)**
  - In order to insure traceability
  - In order to defined affectation of agreed activities i.e. attached to a change order
- **Change Management Workflow in order to ensure:**
  - Efficient logistic of work items within extended enterprise with remote worker (in order to short the design time)
  - Parallelization of tasks for concurrent engineering
  - Contextualization and control of activities leading to consult, create, modify or erase product data. It is of particular importance for traceability and quality at team (collaborative work) and enterprise level (configuration management).
But how to proceed for federation of PDM Systems within a PLM strategy?

1. Workflow Interconnection impossible for Cross Organization Process
2. Different PDM and Workflow systems
3. Enactment of Business process
   - numerous standards or standalone solutions
   - different paradigms
4. communication with numerous heterogeneous objects
5. consumption of services
6. Relevant data exchange or sharing

1. Workflow Interconnection impossible for Cross Organization Process transversal process
2. Different PDM and Workflow systems are used and they are not designed for any collaboration.
3. For enactment of Business process, numerous standards or standalone solutions exist, based on different paradigms:
   1. workflow systems, choreography of services or composition of services.
   2. Process modeling can be based on Activities, flows or be state oriented.
4. Federation of PDM implies communication with numerous heterogeneous objects and consumption of heterogeneous services
5. Federation of PDM systems also implies Relevant data exchange or sharing strategy.
Within the ATHENA program, the elaborated to-be business scenario was targeting Product Data sharing between numerous organizations that are member of the virtual enterprise, with as proposed approach establishment of a collaboration space based on manufacturing and ICT (Information and Communication) standards.

CMII provides a set of rules for efficient collaboration management and is used to procure labels for organization and software products.

For executable process modeling, the target was to use ATHENA Cross-Organizational Business Process

The services should be based on standardized specification coming from OMG Mantis (PLM services) or OASIS PLCS group (PLCS PLM services)

Finally, the application protocols for Product data should be the common subpart of all STEP application protocols related to PDM and configuration management: PLM Module that is shared by PLCS and other application protocols. In addition, the PLCS project is using Reference Data Libraries for integration of PLCS with other models of reference that are more specific. This approach was initially promoted by the EPISTLE project (Oil and Gaz domain).
ATHENA PDE2 underlying scoping

1. Industrial partners of ATHENA addresses some of these challenges

2. Standards are very important but
   - numerous
   - different focus – communities - purpose
   - Overlapping
   - no always compatible

3. Important for NCPD and ATHENA to
   1. Understand
   2. Position and compare
   3. Leverage through composition
   4. Disseminate and share

=> motivation and scope of the current training cession

Or Industrial partners of ATHENA addresses some of these challenges
• through Collaborative Product Development business scenarios proposed by EADS CCR in ATHENA A4 project
• Through state of the art and state of the practice
• Through Networked Collaborative Product Development pilot in ATHENA A5 project

Within a networked organization, standards are very important but
• They are numerous
• They have different focus and are provided by different communities for different purpose
• They are sometimes overlapping and sometimes non interoperable

It is consequently important to, within the scope of Collaborative Product Development, to:
• Understand them
• Position and compare them
• Leverage their usage by adequate composition
• Disseminate and share this knowledge, in particular through training activities, for the different involved communities and domains

It is the motivation and scope of the current training cession.
Standards

- STEP application protocols
- XML Vocabularies
- Given Domain Ontology
- UML Profiles
- Standardized Process or Workflow language
- Standardized Processes
The official title of **ISO 10303** is *Industrial automation systems and integration - Product data representation and exchange*.

ISO 10303 is known as **STEP** or the **Standard for the Exchange of Product model data**. It is an *International Standard* for the computer-interpretable representation and exchange of industrial product data. The objective is to provide a mechanism that is capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

Typically STEP can be used to exchange data between **CAD**, **CAM**, **CAE**, **PDM/EDM** and other **CAx** systems. STEP is addressing product data from mechanical and electrical design, analysis and manufacturing, with additional information specific to various industries such as automotive, aerospace, building construction, ship, oil & gas, process plants and others.

STEP is developed and maintained by the ISO technical committee TC 184, **Technical Industrial automation systems and integration**, sub-committee SC4 **Industrial data**. Like other ISO and IEC standards STEP is copyright by ISO and is not freely available. Other standards developed and maintained by ISO TC184/SC4 are:

- **ISO 13584** PLIB - Parts Library
- **ISO 15531** MANDATE - Industrial manufacturing management data
- **ISO 15926** Process Plants including Oil and Gas facilities life-cycle data
- **ISO 18629** PSL - Process specification language
- **ISO 18876** IIDEAS - Integration of industrial data for exchange, access, and sharing
- **ISO 22745** Open Technical Dictionary
- **ISO 8000** Catalogue management systems: Requirements

Much of the work on STEP is currently addressing different functional needs for Product Data Management (PDM) all along the lifecycle of the product (PLM) within the Virtual Enterprise. STEP aims product data exchange and sharing between different actors and stakeholders concerned by a product all along the value chain.
Current activities aims to cover all the phases of the life-cycle and different implied disciplines:

- From requirement engineering based on system engineering and engineering analysis (AP233)
- To Product Support (AP239)
- Through Product Design (AP214, AP203)
- And Manufacturing Planning and Support

Different concerned implied disciplines used different modeling techniques that are covered by dedicated Application Protocols:

- STEP NC
- Electric Harness
- Mechanical Drawing

Design, planning and manufacturing data concerning sub-components or equipment can be reuse for different product, as for example those concerning an engine that can be used by different kind of product as tank, cars, aircraft or trunks.

It is particularly important with the current trend concerning virtualization of the enterprise and component of the shelves for manufactured products.
STEP is divided into many parts, grouped into

Environment

Parts 1x: Description methods: EXPRESS, EXPRESS-X
Parts 2x: Implementation methods: STEP-File, STEP-XML, SDAI
Parts 3x: Conformance testing methodology and framework

Integrated data models

The Integrated Resources (IR), consisting of

Parts 4x and 5x: Integrated generic resources
Parts 1xx: Integrated application resources

PLIB ISO 13584-20 Parts library: Logical model of expressions

Parts 5xx: Application Integrated Constructs (AIC)
Parts 1xxx: Application Modules (AM)

Top parts

Parts 2xx: Application Protocols (AP)
Parts 3xx: Abstract Test Suites (ATS) for APs
Parts 4xx: Implementation modules for APs
Numerous STEP application protocols are available as international standard, but some are still at Preliminary stages (Proposal, Preparatory, Committee, Enquiry, Approval or Publication stage). Some of them are disciplines related (Explicit draughting, Associative draughting, Mechanical design using boundaries, sheet metal die planning and design, composite and metal structural analysis and related design, Electronic assembly, interconnection and packaging design, electronical design and installation, System engineering data representation, Computation Fluid Dynamics…)

Some are Industrial domain related (Ship structures, Building elements using explicit share representation, furniture product and project data), even if most of the time the AP are generic enough to be reused by other domain as for example AP214 (Core data for automotive mechanical design processes). Several functional parts are in fact shared by most of the domain as the one related to product structure and configuration management.

Finally there is a trend to cover all the phase of product lifecycle from AP233 (System Engineering) to AP239 (Product LifeCycle Support) through AP203 (configuration controlled 3D design).

### STEP application protocols

<table>
<thead>
<tr>
<th>Number</th>
<th>Protocol Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.201</td>
<td>Explicit draughting [ATIS 301 = X]</td>
</tr>
<tr>
<td>1.202</td>
<td>Associative draughting [C]</td>
</tr>
<tr>
<td>1.203</td>
<td>Configuration controlled 3D design (214, 215 = X)</td>
</tr>
<tr>
<td>1.204</td>
<td>Mechanical design using boundary rep [Y]</td>
</tr>
<tr>
<td>2.200</td>
<td>Mechanical design using surface rep [X]</td>
</tr>
<tr>
<td>3.200</td>
<td>Composite &amp; metal structural anal &amp; related design[X]</td>
</tr>
<tr>
<td>4.200</td>
<td>Electronic assy, interconnection &amp; packaging design [X]</td>
</tr>
<tr>
<td>5.210</td>
<td>Electronic P-C essay, test, diag, &amp; remanu[X]</td>
</tr>
<tr>
<td>6.212</td>
<td>Electrochemical design and installation [X]</td>
</tr>
<tr>
<td>7.213</td>
<td>Num control (NC) process plans for mach'd parts [X]</td>
</tr>
<tr>
<td>7.214</td>
<td>Core data for automotive mach design processes [X]</td>
</tr>
<tr>
<td>8.215</td>
<td>Mach arrangement [X]</td>
</tr>
<tr>
<td>8.216</td>
<td>Ship mechanical forms [X]</td>
</tr>
<tr>
<td>8.217</td>
<td>Ship fitting [X]</td>
</tr>
<tr>
<td>8.218</td>
<td>Ship structures [W]</td>
</tr>
<tr>
<td>8.219</td>
<td>Dimension inspection [X]</td>
</tr>
<tr>
<td>8.220</td>
<td>Proc. (pct), mfg, assay of layered electrical products [X]</td>
</tr>
</tbody>
</table>

**Legend:** Part Status (C, P, I, L, E refer to Implement):
- C = Proposed Stage (Proposal — for AP initiation)
- P = Proposal Stage (Preliminary plan — for AP publication)
- I = Industrial Stage (Industry support — for AP approval)
- L = Final Stage (Draft issued — for AP registration)
- E = Final Stage (Finalised, ready for publication)

AP = AP214 Core Data for Automotive Mechanical Design Processes
STEP application protocols

STEP Application Module

- **Motivation**
  - APs too big
  - Too much overlap with each other
  - APs documents not sufficiently harmonized
  
  ⇒ development of the **STEP modular architecture** (400 and 1000 series)
  
  - Primarily driven by new AP
    - covering additional life-cycle phases
      - early requirement analysis (AP233), maintenance and repair (AP239), new industrial areas (AP221, 236)
    - In addition older APs prepare for a new edition on a modular basis (AP203, 209, 210)
    - This is an ongoing process.

- **STEP Application modules** define
  - common building blocks to create modular Application Protocols (AP) within ISO 10303
  - Higher-level modules built up from lower-level modules
  - Modules on the lowest level are *wrappers* of concepts, defined in the Integrated Resources (IR) or Application Integrated Constructs (AIC).
  - Modules on a medium level link lower level modules with each other and specialize them.
  - Only modules on the highest levels completely cover a particular area so that they can be implemented.

STEP application modules

Motivations for modules are the following:

- Application Protocols are too big (thousand and thousand of definitions);
- Too much overlap with each other;
- APs documents are not sufficiently harmonized leading to important work for production and validation.

It led to the development of the STEP modular architecture (400 and 1000 series), in order to have APs that are composite e-documents, each document component (module) being reusable by the other application protocols. One benefit is to short the production of an Application Protocol.

Primarily driven by new AP

- covering additional life-cycle phases
  - early requirement analysis (AP233), maintenance and repair (AP239), new industrial areas (AP221, 236)
  - In addition older APs prepare for a new edition on a modular basis (AP203, 209, 210)
  - This is an ongoing process.

**STEP Application modules** define

- common building blocks to create modular Application Protocols (AP) within ISO 10303
- Higher-level modules built up from lower-level modules
- Modules on the lowest level are *wrappers* of concepts, defined in the Integrated Resources (IR) or Application Integrated Constructs (AIC).
- Modules on a medium level link lower level modules with each other and specialize them.
- Only modules on the highest levels completely cover a particular area so that they can be implemented.
It is possible to obtain STEP modules repository from Sourceforge (STEPMOD). It contains set of XML documents and style sheets allowing to navigate within modules, per modules or per Application Protocols. It is used for the new APs and Modules developers. It contains also EXPRESS-G diagrams and EXPRESS definitions.
It also contains generic model of the Business Processes for which the AP is defined.
STEP technical framework provides languages and structures (or constructs) allowing to define business concepts within Application Protocols that will be used for product data exchange, sharing and Retention between organizations and Computer Aided Applications.
STEP application protocols
STEP AP and other technologies and initiatives

- **Links with ICT technologies** coverage through STEP parts
  - XML (part28), UML (part25), Java (part 22)…
- **Joint effort with other technology standardization bodies**
  - Object Management Group
    - Manufacturing working group (Mantis)
      - PDM Enablers, PLM services
    - Ontology working group (Profile for STEP through the Mexico project)
    - SysML (Developed jointly with AP233)
  - OASIS: PLCS project
    - PLCS PLM services, Reference Data Libraries (links with semantic WEB)

⇒ Important recognition within the manufacturing community and from other initiatives
⇒ Openness ensure through numerous joint initiatives
⇒ Links with emerging important IT technologies through the bindings

[OASIS Technology Report]
March 19, 2001] The STEPml “library of XML specifications” has been designed as a publication forum and education center for XML-based STEP product data schemas governing process integration, supply chain management, collaborative engineering, analysis, manufacturing, and customer support. STEPml is sponsored by PDES, Inc., an international industry/government consortium. In February 2001, STEPml published the first three in a series of planned resources which combine the "semantically rich, international standard data models from STEP (ISO 10303) with the widespread infrastructure of XML and the Web. STEP is an international standard for the representation of product data; STEP models are documented using EXPRESS, a formal object-flavored language that has a robust constraint definition capability. STEPml takes the data models from STEP and publishes them as XML specifications; the STEPml XML specifications are automatically generated from STEP schemas. Resources published to date include a STEPml XML DTD for the STEP PDM Schema, a STEPml Product Identification and Classification Specification, and a technical overview of the STEP Object Serialization Early Binding (OSEB). STEPml is one of several STEP-XML initiatives now gathering momentum. Several bindings have been designed for the mapping of semantically-rich EXPRESS data models to XML, and are documented in ISO's 306-page Proposed Draft Technical Specification Product Data Representation and Exchange. Implementation Methods: XML Representation of EXPRESS Schemas and Data [ISO TC184/SC4/WG11 N140. ISO/PDTS 10303-28:2000(E)]. The ISO/SC4 10303-25 project is also developing an EXPRESS to OMG XMI binding.
STEP Application Protocols can be considered as Collaborative Product Development ontological models in the manufacturing community! They implied a lot of effort and investment from Industry that are to be reused (Cost of ontology definition is very high)

Product Data Management (PDM) and Engineering Data Management (EDM) systems are enterprise applications!

Important APs to consider within the scope of ATHENA and collaborative Product Development are AP233, AP203/AP214 and PLCS, in particular subparts linked to STEP Modules, and that provide description of Shared Product Breakdown structure, Configuration and Change Management and Person and organization.

STEP technologies and bindings to technologies that enable interoperability (XML, UML, Services) are also important and will be detailed latter.
XML vocabularies
XML specifications

- XML SGML adaptation for Internet by W3C
  - HTML capabilities extension
  - based on extensible tags
    - allowing to create users’ vocabularies
    - Often called XML languages
    - XML called a meta language i.e. language to describe languages
- About 10 main specifications for core XML.
- About hundreds of specifications – that will probably no more exist in 2 years. Some examples:
  - SMIL (for multimedia WEB composite documents)
  - SVG for vectors based drawing
  - MathML for mathematical formulas
  - XUL for user interface
  - BPEL for executable Business Processes
  - XPDL for XML Process Definition
  - WSDL for description of WEB services
  - SOAP for message description
  - RDF for description of resources
  - OWL for definition of ontological models
  - XMI for XML Model Interchange

XML is an adaptation of SGML for Internet by W3C.
It’s also an opportunity to extend HTML capabilities, as HTML is to much static and as HTML tags are not very useful for documents meta-description.
XML is based on extensible tags, that allow the user to create its own set of tags, called a vocabulary. Such vocabularies are often called XML languages. It’s why XML is described as a meta language: a language to describe languages.

About 10 main specifications for core XML.
About 200 specifications – that will probably no more exist in 2 years

Some examples: SMIL (for multimedia WEB composite documents, SVG for vectors based drawing, MathML for mathematical formulas, XUL for user interface, BPEL for executable Business Processes, XPDL for XML Process Definition, WSDL for description of WEB services, SOAP for message description, RDF for description of resources, OWL for definition of ontological models, XMI for XML Model Interchange...
The 10 XML rules
1. You will be useful on Internet
2. You will support a big variety of applications
3. You will be SGML compatible
4. You will have to be easy to write programs which manipulate you
5. You will have the minimum of optional functions
6. You will be human readable
7. You will be available quickly
8. The specification which will describe you will have to be simple and concise
9. A document respecting you will have to be easy to build
10. You can not be concise

XML Document structure is the following:
- Header
- Schema (DTD, XML Schema, Schematron,...)
- XML Document

Evolution of usage of XML from eDocument (DTD) to
- Data interchange (XML schema) and rule based validation (Schematron)
- Distributed Resources on the WEB (RDF)
- Semantic WEB (OWL)

⇒ Today to be XML compliant does not mean anything
⇒ What is important is the used XML vocabulary and how it is supported by concerned applications.

The usage of XML evolved XML from eDocument (DTD) to Data interchange (XML schema) and rule based validation (Schematron) and is today targeting distributed Resources on the WEB (RDF) and Semantic WEB (OWL).

Today to be XML compliant does not mean anything.
What is important is the used XML vocabulary and how it is supported by concerned applications.
So DTDs provide a vocabulary (or constructs) allowing to define the business vocabulary for electronic Documents.
So the XML-Schema language provides a vocabulary (or constructs) allowing to define the business vocabulary for electronic Documents with structured data. It means that the eDocuments is also to be used by software components managing information, and not only people.
The XML vocabularies related to PDM and PLM are the following:

1. **Product Data Markup Language (PDML)**
   - Extensible Markup Language (XML) vocabulary for interchange of product information between PDM or government systems (JEDMICS)
   - part of Product Data Interoperability (PDI) project (Joint Electronic Commerce Program Office and several other Federal Government agencies and commercial entities)
   - Does not seem active anymore since 2004

2. **STEP ISO10303 part 28 XML Binding**
   - Edition 1: 3 different bindings for electronic documents (DTDs)
   - Edition 2: binding to XML schema (DIS in 2006)

3. **MANTIS PLM Services and PLCS PLM Services**
   - Both initiatives defines PLM services defined as WEB services
   - XML schemas are defined for service operations inputs/outputs typing
   - based on PDM Modules/PDM Schema (ARM) with some adaptation
<table>
<thead>
<tr>
<th>Ontology</th>
<th>What Ontology is</th>
</tr>
</thead>
<tbody>
<tr>
<td>• In both computer science and information science, an ontology is a consensual explicit formal model that represents a domain of interest.</td>
<td></td>
</tr>
<tr>
<td>• For artificial intelligence and semantic WEB, it should in addition support reasoning about the objects in that domain and the relations between them.</td>
<td></td>
</tr>
<tr>
<td>• For artificial intelligence, the semantic web, software engineering and information architecture, ontology is considered as a form of knowledge representation about the world or some part of it.</td>
<td></td>
</tr>
<tr>
<td>• Ontology generally describes:</td>
<td></td>
</tr>
<tr>
<td>• Individuals: the basic or &quot;ground level&quot; objects</td>
<td></td>
</tr>
<tr>
<td>• Classes: sets, collections, or types of objects[1]</td>
<td></td>
</tr>
<tr>
<td>• Attributes: properties, features, characteristics, or parameters that objects can have and share</td>
<td></td>
</tr>
<tr>
<td>• Relations: ways that objects can be related to one another</td>
<td></td>
</tr>
</tbody>
</table>

Ontology
What ontology is
In both computer science and information science, an ontology is a consensual explicit formal model that represents a domain of interest.
For artificial intelligence and semantic WEB, it should in addition support reasoning about the objects in that domain and the relations between them.
For artificial intelligence, the semantic web, software engineering and information architecture, ontology is considered as a form of knowledge representation about the world or some part of it.
Ontology generally describes:
• Individuals: the basic or "ground level" objects
• Classes: sets, collections, or types of objects[1]
• Attributes: properties, features, characteristics, or parameters that objects can have and share
• Relations: ways that objects can be related to one another
Ontology

Ontology languages

- **Ontology language**: formal language used to encode the ontology
- Numbers of such languages exist, both proprietary and standards-based:
  - KIF is a syntax for first-order logic based on S-expressions
  - CycL based on first-order predicate calculus with some higher-order extensions (from Cyc project)
  - OWL…
- **Ontology Web Language**
  - follow-on from
    - RDF and RDFS
    - earlier ontology language projects OIL, DAML and DAML+OIL.
  - Intended to be used over the World Wide Web
  - All its elements (classes, properties and individuals) are defined as RDF resources, and identified by URIs
- **In ATHENA and NCPD context, OWL choice**
  - OPAL definition proposed in OWL
  - As Semantic WEB component integrated in WEB technologies set, adapted to Business to Business

Ontology languages

An ontology language is a formal language used to encode the ontology

Numbers of such languages exist, both proprietary and standards-based:

- KIF is a syntax for first-order logic based on S-expressions
- CycL based on first-order predicate calculus with some higher-order extensions (from Cyc project)
- OWL…

Ontology Web Language is a follow-on from RDF and RDFS, and earlier ontology language projects OIL, DAML and DAML+OIL. It Intends to be used over the World Wide Web. All its elements (classes, properties and individuals) are defined as RDF resources, and identified by URIs.

In ATHENA and NCPD context, OWL choice was made for:

- OPAL definition (proposed in OWL)
- As Semantic WEB component integrated in WEB technologies set, adapted to Business to Business
So the RDFS language provides a vocabulary (or constructs) allowing to define the business RDF vocabulary for distributed resources (semantic graph) on the WEB. It provides the basis to develop ontology based intelligent agents on the WEB to work on disparate resources of the WEB.
So the OWL language provides a vocabulary (or constructs) allowing to define a business ontology for distributed knowledge (semantic graph) on the WEB. It complements RDF to provide the basis to develop the semantic WEB.
As STEP APs are consensual explicit formal descriptions of domain of interest, they can be considered as ontology. But they don't support reasoning the way defined by Artificial Intelligence or Semantic WEB, even if rules based inference is possible through usage of EXPRESS-X.

If STEP technologies are not ontology oriented, some modules are supporting exchange of categorization and properties.

Reference Data Libraries, used jointly with STEP APs for data integration, were created in OWL in EPISTLE project for model federation. Since EPISTLE, RDLs are used and defined in the PLCS Project.

A binding proposed by ExpressForFree initiative and within EuroSTEP ShareASpace tool. Binding and transformation tool proposed in ATHENA pilots by EADS CCR as part of Networked Collaborative Product Development infrastructure (STEP Mapper).
Unified Modeling Language™ - UML

- OMG's most-used specification
- Generic Graphical general purpose Modeling Languages for
  - application structure, behavior, and architecture
  - but also business process and data structure.

- key foundation with Meta Object Facility (MOF™) for OMG's Model-Driven Architecture®

- Unifies every step of development and integration
  - from business modeling
  - through architectural and application modeling
  - to development, deployment, maintenance, and evolution

- Extensible through profiling

Unified Modeling Language™ - UML

UML is OMG's most-used specification. It is a generic Graphical general purpose Modeling Languages for application structure, behavior, and architecture, but also business process and data structure.

UML is a key foundation with Meta Object Facility (MOF™) for OMG's Model-Driven Architecture®. It unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution

Extensible through profiling.
UML, Profiles and DSM

- **UML Profiles**
  - constrained and customized the language for specific domains and platform
    - business modeling
    - Services, Business Process, Data
    - EXPRESS (San Francisco Project), OWL-Topic Maps (OMG Ontology PSIG)
  - Collection of additional Stereotypes and Tagged values applied to UML features together with constraints
  - Examples
    - SysML, a DSM language for systems engineering, linked to ISO STEP AP233
    - CORBA profile, J2EE Profile…
    - PIM4SOA Profile from ATHENA

UML Profiles is a tool allowing to constrained and customized the language for specific domains and platform. It is used in particular for:

- business modeling
- Services, Business Process, Data
- EXPRESS (San Francisco Project), OWL-Topic Maps (OMG Ontology PSIG)

UML profiles consist in collection of additional Stereotypes and Tagged values applied to UML features together with constraints.

Some profiles examples are:

- SysML, a DSM language for systems engineering, linked to ISO STEP AP233
- CORBA profile, J2EE Profile…
- PIM4SOA Profile from ATHENA
Domain Specific Modeling

Domain specific modeling is a way of designing and developing systems:
- IT systems as computer software but also
- Manufactured product (CAD, CAM, CAx)
- Organizational systems (Enterprise modeling)
- Knowledge systems

- Systematic use of a Domain Specific Language (DSL) to represent the various facets of a system, textual and/or graphical
- Support of higher-level abstractions than General-Purpose Modeling languages
  \[\Rightarrow\] less effort and fewer low-level details to specify a given system
- UML profile mechanism allows it to be constrained and customized for specific domains and platforms

Some communities are recommending systematic use of a Domain Specific Language (DSL) to represent the various facets of a system, textual and/or graphical, as they support of higher-level abstractions than General-Purpose Modeling languages. It implies less effort and fewer low-level details to specify a given system.

If UML is a general purpose language, UML profile mechanism allows UML to be constrained and customized for specific domains and platforms.
XML Metadata Interchange (XMI)

| OMG standard for exchanging metadata information via XML |
| for any metadata whose metamodel can be expressed in Meta-Object Facility (MOF) |
| XML usage |
| UML models interchange format |
| serialization of other language models |
| In OMG’s vision |
| abstract models represent the semantic information |
| instances of arbitrary MOF-based modeling languages such as UML or SysML |
| concrete models represent visual diagrams |
| Visual language not accurate for exchange and sharing => XMI |
| XML Metadata Interchange (XMI) |
| UML-based modeling tools |
| MOF-based metadata repositories in distributed heterogeneous environments |
| software generation tools (model-driven engineering) |
| Several versions: 1.0, 1.1, 1.2, 2.0 and 2.1 (2.x radically different from the 1.x) |
| XMI implementations incompatible => exchange rarely possible |
| Other XML standards for representing metadata |
| As WEB Ontology Language (OWL), built upon RDF |

XML Metadata Interchange

XML Metadata Interchange is OMG standard for exchanging metadata information via XML. It can be used for any metadata whose metamodel can be expressed in Meta-Object Facility (MOF). XML usages are:

- UML models interchange format
- Serialization of other language models

In OMG’s vision, abstract models represent the semantic information and instances of arbitrary MOF-based modeling languages such as UML or SysML. Concrete models represent visual diagrams.

As visual language are not accurate for exchange and sharing, it implies the creation of a textual language: XMI.

XML Metadata Interchange (XMI) is used for exchange of models between UML-based modeling tools, but also for MOF-based metadata repositories in distributed heterogeneous environments, and software generation tools (model-driven engineering).

Several versions: 1.0, 1.1, 1.2, 2.0 and 2.1 (2.x radically different from the 1.x). From experience, XML implementations are most of the time incompatible. Consequently, exchanges are rarely possible.

Finally, some other XML standards for representing metadata exist as WEB Ontology Language (OWL), built upon RDF.
So the XMI language provides a vocabulary (or constructs) allowing to define business concepts (without instances) for application or system models interchange (while UML provides constructs for application or system modeling).
Some PDM or PLM Domain Specific languages exist:

- ISO STEP part 25: Express to UML Binding; AP can be transform as abstract UML model. 
  ATHENA pilots component of furniture and aerospace scenarios provided by UNINOVA (UML 1.5);
- San Francisco Project: EXPRESS as DSL through UML profiling;
- ATHENA Aerospace pilot (EXPRESS to UML profiles for Service Oriented Enterprise Application);
- SysML and AP233; AP233 is the underlying information model for SySML, that is a DSL.
Process Modelling: several definitions

- “A coordinated set of activities that are connected in order to achieve a common goal” (WFMC)

- “All the real-world elements involved in the development and maintenance of a product, i.e. Artifacts, production support, activities, agents and process support” (Jean-Claude Derriame et al.)

- “A series of activities that are linked to perform a specific objective” (CAM-I, Consortium for Advanced Manufacturing International)

- “Set of interrelated or interacting activities which transforms inputs into outputs” (ISO)

- “A process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs” (BPMI.org)

Several definitions are available for process:
- “A coordinated set of activities that are connected in order to achieve a common goal” (WFMC)
- “All the real-world elements involved in the development and maintenance of a product, i.e. Artifacts, production support, activities, agents and process support” (Jean-Claude Derriame et al.)
- “A series of activities that are linked to perform a specific objective” (CAM-I, Consortium for Advanced Manufacturing International)
- “Set of interrelated or interacting activities which transforms inputs into outputs” (ISO)
- “A process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs” (BPMI.org)
In fact, the definition depends on the usage of the process model. And numerous usages exist:

• Total quality
• Management control
• Activity based costing (ABC) and Activity Based Management (ABM)
• Process reconfiguration
• ISO 9000-2000
• Process and Project management
• System Engineering (Software, Manufactured Product, Organisation)

• Extension with Information and Communication Technologies
  – Supply Chain Management
  – Customer Relationship Management
  – Product Lifecycle Management
  – PortFolio Management
  – Enterprise Resource planning

⇒ Numerous Domain Specific languages for the different usage as different goals and purpose, targeting or not automation using engines or interpreter.

Process modelling and process execution are also often provided as extension of Information and Communication Technologies in several domains:

• Supply Chain Management
• Customer Relationship Management
• Product Lifecycle Management
• PortFolio Management
• Enterprise Resource planning

Finally, numerous Domain Specific languages exist for the different usage as different goals and purpose, targeting or not automation, and using engines or interpreter.
Process Modelling: several generic representations

- **Activity oriented**
  - activity (transformation of a system)
  - links
  - connections
  - resources
  - inputs
  - Outputs

- **Flow Oriented**
  - elements oriented circulation
  - activities are not considered
  - connections
  - actors
  - Sequences

- **State Oriented**
  - State
  - Transition
  - Condition ("declenchement")
  - Event
  - Synchronisation

### Composition of service
- Scope = one participant

### Choreography of service
- Scope = several participants

For all these domains, several generic representations are available:

1. **Activity oriented**
   - The concepts used then are activity (transformation of a system), links, connections, resources, inputs and Outputs

2. **Flow Oriented**
   - The concepts used then are elements oriented circulation, connections, actors and sequences. Activities are not considered.

3. **State Oriented**
   - The concepts used then are State, Transition, Condition ("declenchement"), Event and Synchronisation.

With development of Service Oriented Architecture, emerging technologies are distinguishing:

- Composition of service (Scope = one participant)
- Choreography of service (Scope = several participants)
An another important classification is to be considered. Process models can be modelled in order to be:

1. Fully automated
   - It is the case with Programming languages, with distinction between Programming language for standalone applications and Programming language for WEB and Distributed application.

2. Semi automated, that includes:
   - Workflow systems: distribution of activities between actors and systems
   - Applications implying interaction with users based on a define process

3. Non automated process:
   - The execution system of the process is not an automate, but an organisation (e.g.: enterprise, technico_operational system for modern battlefield...)
   - The process is not schedule by an automate. Such a process can nevertheless be modelled with a software application. It should then enable efficient Process Design by supporting Representation, Analysis, Management and Exchange - with possible model automatic checking, analysis and validation through parsing, rule checking and simulation.
Process Modelling: several standards

- **Fully automated**
  - Programming languages
    - Programming language for standalone applications: procedural languages
    - Programming language for WEB and Distributed application: BPEL
- **Semi automated**
  - Workflow systems: Wfmc Standards including XPDL, Wf-Ml…
- **Non automated process**
  - Representation
    - BPMN (from OMG), ARIS, numerous FlowChart, IDEF0, some UML and SysML diagrams
  - Analysis
    - ISO 9001/2/3 with PCDA approach, Activity based costing (ABC), Unified Process…
  - Management
    - SPEM (development project), Pert/Gantt (project), ISO 15288 (Organisation for Engineering of systems)
  - Exchange: PIF, PSL, BPEL, XPDL, XMI with accurate profile, PDM module subpart (with STEP or XML technologies)

For the different kind of processes, several standards exist:

For Fully automated process:
- Programming language for standalone applications: procedural languages
- Programming language for WEB and Distributed application: BPEL

For Semi automated process:
- Workflow systems: Wfmc Standards including XPDL, Wf-Ml…

For Non automated process:
- Representation: BPMN (from OMG), ARIS, numerous FlowChart, IDEF0, some UML and SysML diagrams
- Analysis: ISO 9001/2/3 with PCDA approach, Activity based costing (ABC), Unified Process…
- Management: SPEM (development project), Pert/Gantt (project), ISO 15288 (Organisation for Engineering of systems)
- Exchange: PIF, PSL, BPEL, XPDL, XMI with accurate profile, PDM module subpart (with STEP or XML technologies)

Due to the very various process, it is very difficult to impose one single standard. PSL language defined by ISO defined a core set of concepts and then several extension for the various usage. But PSL is based on KIF that is not very used, especially on the WEB.
The Workflow Management Coalition, founded in August 1993, is a non-profit, international organization of workflow vendors, users, analysts and university/research groups. The Coalition's mission is to promote and develop the use of workflow through the establishment of standards for software terminology, interoperability and connectivity between workflow products. Consisting of over 285 members, spread throughout the world, the Coalition has quickly become established as the primary standards body for this rapidly expanding software market... The Coalition is divided into three major committees, the Technical Committee, the External Relations Committee, and the Steering Committee. Small working groups exist within each committee for the purpose of defining workflow terminology, interoperability and connectivity standards, conformance requirements, and for assisting in the communication of this information to the workflow user community... The Coalition has established a number of working groups, each working on a particular area of specification. The working groups are loosely structured around the "Workflow Reference Model" which provides the framework for the Coalition's standards program. The Reference Model identifies the common characteristics of workflow systems and defines 5 discrete functional interfaces through which a workflow management system interacts with its environment - users, computer tools and applications, other software services, etc." [website description 2002-09]
Here is a mindmap of all the Wfmc specification in 2007, that complete the previous slide and indicates documents that are still in construction on those that are difficult to find.
The architecture of reference provides 6 main components of a workflow system, and identifies the 5 types of interfaces (process definition interface, administration and monitoring interface, interoperability interface, client application interface and invoked application interface).
Key concepts for a workflow system are:

• Workflow Process Model definition

• Workflow Process Model Instances (created through enactment services on the basis of available Workflow Process Models)

• Work list: the list of work items published or notified by the enactment service that each workflow participants have to perform in order to complete the workflow model instance.
Here is an example of implementation of the architecture and the components defined by Wfmc (Shark and Jawe from the Enhydra project).
It allows to see how a definition of workflow model looks like in XPDL and the corresponding visual representation proposed by Jawe (that is not based on any standard).
Business Process Modeling Notation

- BPMN 1.0: May, 2004, «BPMN 1.0 specification» published
- February, 2006, BPMN 1.0 adopted as OMG specification
- Currently, 39 companies have implemented BPMN
- Process Modeling: ordered business activities sequence capture + supporting information.
- Business Processes describes how Business reach its objectives.
- Several level:
  - Process Maps
  - Process Descriptions
  - Process Models
- BPMN supports all
- Consensual flow-chart type notation
- BPMN aims to support mechanism for BPEL generation
  - A process developed by analysts should then be directly reused by an execution engine in place of being interpreted and translated
  - Still not sufficiently specified
  - BPEL to BPMN?

BPML.org developed BPML (XML Process Execution language) and realized the need for ungraphical representation language.
BPML later replaced by BPEL as targeted execution language. In August, 2001, Notation Working Group was created with 35 members, that published BPMN 1.0 in May, 2004, «BPMN 1.0 specification».
In February 2006, BPMN 1.0 was adopted as OMG specification.
Currently, 39 companies have implemented BPMN.
Process Modeling is considered as capture of a sequence of ordered business activities and supporting information. Business Processes describe how Business reach its objectives. There are several levels of process modeling:
- Process Maps – simple activity flow charts
- Process Descriptions: – flow charts extended with additional information, but not sufficiently to really define performance
- Process Models – flow charts extended with information enough to be able to analyze, simulate or execute the process
BPMN supports each of these levels. It is a notation of type flow-chart for definition of business processes, resulting on agreement between multiple software tools vendors who have their own notation but would like to have a single one for end-user benefit in term of understanding and training.
BPMN aims to support mechanism for BPEL generation: a process developed by analysts should then be directly reused by an execution engine in place of being interpreted and translated.
Nevertheless, the two standards relationships are still not sufficiently specified, in particular for BPEL to BPMN.
BPEL4WS 1.1 by OASIS specified in August 2001

- **Serialized process in XML**
  - programming in the large (long-running asynchronous processes)
- **BPEL is an orchestration language, not a choreography language** (i.e. scope of one participant against several)
- **Initial Goals**
  - Goal 1: external entities abstract interaction with WSDL 1.1
  - Goal 2: XML based, not graphical
  - Goal 3: Web service orchestration abstract and executable concepts
  - Goal 4: hierarchical and graph-like control regimes (less fragmentation of process modeling space)
  - Goal 5: Simple Data manipulation and control flow functions
  - Goal 6: Identification mechanism for process instances at application message level
  - Goal 7: Implicit creation and termination of process instances as basic lifecycle mechanism + planned future advanced operations.
  - Goal 8: Long-running transaction model (compensation, failure recovery for long-running business processes)
  - Goal 9: Use Web Services as the model for process decomposition and assembly.
  - Goal 10: Web services standards based (modular composition)

- **Preparation of a new version called WS-BPEL2**
  - WS-BPEL 2.0 with significant differences and incompatibility
  - Numerous extensions
  - Last available draft: August 2006 at [http://docs.oasis-open.org/wsbtel/2.0/wsbtel-v20-rtd4.html](http://docs.oasis-open.org/wsbtel/2.0/wsbtel-v20-rtd4.html)
- **BPEL4People**

BPEL4WS 1.1 was specified in August 2001 by OASIS, which is being specifying **WS-BPEL 2.0** with significant differences and incomaptibility.

BPEL describes process serialized in XML and aims to enable programming in the large (long-running asynchronous processes).

BPEL is an orchestration language, not a choreography language (i.e. scope of one participant against several).

Initial Goals of BPEL are the following:

**Goal 1:** interaction with external entities through Web Service using WSDL 1.1, with "abstract interaction" i.e. dependence on portType definitions, not on port definitions.

**Goal 2:** XML based language definition, not a graphical one.

**Goal 3:** set of Web service orchestration concepts to be used by external (abstract) and internal (executable) views of a business process. **Goal 4:** Provide both hierarchical and graph-like control regimes, and allow their use to be blended as seamlessly as possible. This should reduce the fragmentation of the process modeling space.

**Goal 5:** Data manipulation functions for the simple manipulation of data needed to define process data and control flow.

**Goal 6:** Identification mechanism for process instances that allows the definition of instance identifiers at the application message level.

**Goal 7:** Implicit creation and termination of process instances as the basic lifecycle mechanism. Advanced lifecycle operations such as "suspend" and "resume" may be added in future releases for enhanced lifecycle management.

**Goal 8:** Long-running transaction model based on compensation actions and scoping to support failure recovery for parts of long-running business processes.

**Goal 9:** Use Web transaction model as the model for process decomposition and assembly.

**Goal 10:** Build on Web services standards (approved and proposed) as much as possible in a composable and modular manner.
Preparation of a new version called WS-BPEL2 aims to provide new activity types, Completion condition in forEach activity, Variable initialization, XSLT for variable transformations, XPath access to variable data, XML schema variables in Web service activities, locally declared messageExchange, Clarification of Abstract Processes (syntax and semantics). However, the WSBPEL Technical Committee is expected to publish validated WSBPEL 2.0 late in 2006 (August 2006 Draft version available at http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v20-rddl.html).

Some activities are targeting to produce BPEL4People, in order to adapt it to process modeling taking into account actors.

### Design tools
- ActiveBPEL Designer
- WebSphere Studio Application Developer Integration Edition v5.1
- NetBeans 5.5

### Execution tooling
- Active BPEL Engine
- ActiveBpel Enterprise Server
- Oracle and IBM move BPEL to the BPI forefront
- Process eXecution Engine
- Microsoft BizTalk Server 2004
- WebSphere Business Integration Server Foundation

As illustrated by the table of content of BPMN, one of the main target is today BPEL (and not any more BPML).

Iterative development is made in order to align BPEL and BPMN conceptual models.
Here is an example of BPEL usage for modeling a process. BPEL is a standard for choreography. Only one lane is concerned, and the other lanes are not known, as only received messages and sent messages are known.

It makes the Process applications and interfaces independent, as only messages are concerned. But it is consequently not possible to model a process concerning several lanes and participants. It is also impossible to model a workflow model.

For such reason, several are considering BPEL only as a programming language.

One advantage of BPEL is that it provides a way to establish communication between several lanes under the control of several engines.
Here are some examples of the BPMN visual language.

BPMN provides a very rich set of concepts, that is more rich than the one associated to BPEL. It can be use as well to describe process trunk than composition of processes or workflow process models.

Several standardized process exist, in particular for what is concerning quality. Important standardized process for PLM are:

1. ISO 9001 (Wikipedia)

   “ISO 9000 is a family of ISO (the International Organization for Standardization) standards for quality management systems. ISO 9000 was developed from the British Standards Institution’s BS 5750. The ISO 9000 standards are maintained by ISO and administered by accreditation and certification bodies.

   Although the standards originated in manufacturing, they are now employed across a wide range of other types of organizations. In fact, according to ISO in 2004, "service sectors now account by far for the highest number of ISO 9001:2000 certificates - about 31% of the total" - source: the ISO Survey 2004

   ISO 9000 is quite similar to ISO 14000. Both pertain to how a product is produced, rather than how it is designed. For example, ISO 216 very precisely specifies sizes of paper. ISO 9000 and ISO 14000 are more general, referring to processes, rather than any single product.

   ISO 9000 is intended to make sure that the product—any product—has been produced in the most efficient and effective manner possible. ISO 9000 does not guarantee the compliance (and therefore the quality) of end products and services; rather, it certifies that consistent business processes are being applied.

   ISO 14000 exists to ensure the product (again, any product) has the lowest possible environmental impacts.”
Standardized process

• Quality processes
  – Defined within the scope of an organization
    • Defined a consensual way (all actors are implied in PCDA approach)
  – Proposed by an organization and leading to certification
    • ISO 9001 and PCDA
    • Capacity Maturity Model
    • CMII Configuration Management model II
      – Leading to certification of enterprise
      – Leading to certification of software product

• Modeling languages independent
  – But modeling them can leverage their validation, usage and control

2- CMM (Wikipedia)

“Capability Maturity Model (CMM) broadly refers to a process improvement approach that is based on a process model. CMM also refers specifically to the first such model, developed by the Software Engineering Institute (SEI) in the mid-1980s, as well as the family of process models that followed. A process model is a structured collection of practices that describe the characteristics of effective processes; the practices included are those proven by experience to be effective. [1]

The Capability Maturity Model can be used to assess an organization against a scale of five process maturity levels. Each level ranks the organization according to its standardization of processes in the subject area being assessed. The subject areas can be as diverse as software engineering, systems engineering, project management, risk management, system acquisition, information technology (IT) services and personnel management.

CMM was developed by the SEI at Carnegie Mellon University in Pittsburgh. It has been used extensively for avionics software and government projects, in North America, Europe, Asia, Australia, South America, and Africa. [2] Currently, some government departments require software development contract organization to achieve and operate at a level 3 standard.”

3- CM II (Configuration Management Institute)

“Configuration Management is the process of managing products, facilities and processes by managing the information about them, including changes, and ensuring they are what they are supposed to be in every case.

CMII expands the scope of Configuration Management (beyond design definition) to include any information that could impact safety, quality, schedule, cost, profit or the environment.

CMII shifts the emphasis of Configuration Management to (1) accommodate change, (2) accommodate the reuse of standards and best practices, (3) ensure that all requirements (all released information) remain clear, concise and valid, (4) communicate (1), (2) and (3) to each user promptly and precisely and (5) ensure that results conform to the requirements in each case.”
Special focus on CM II

Configuration Management
- process of managing products, facilities and processes by managing the information about them, including changes, and ensuring they are what they are supposed to be in every case.

CMII expands CM scope:
- include any information that could impact safety, quality, schedule, cost, profit or the environment
- Emphasis on
  1. accommodate change
  2. accommodate the reuse of standards and best practices
  3. ensure that all requirements (all released information) remain clear, concise and valid
  4. communicate (1), (2) and (3) to each user promptly and precisely and
  5. ensure that results conform to the requirements in each case.

Configuration Management is a key function of Product Data Management System and is a key issue for PLM strategy and collaborative product development within networked organization.

CM II label is possible for organization but also for Software Product (in particular PDM systems like Windchill 6.8).

It is the reason why it is used in ATHENA Aerospace pilot component (c.f. ATHENA B4 and B5 deliverables related to Aerospace business scenarios and pilots).

It should illustrate usage of similar quality process in other business case to enable efficient interoperability and collaboration.

Configuration Management

Configuration management is a process for managing products, facilities and processes by managing the information about them, including changes, and ensuring they are what they are supposed to be in every case.

CMII expands CM scope. It includes any information that could impact safety, quality, schedule, cost, profit or the environment. It provides emphasis on

1. accommodate change
2. accommodate the reuse of standards and best practices
3. ensure that all requirements (all released information) remain clear, concise and valid
4. communicate (1), (2) and (3) to each user promptly and precisely and
5. ensure that results conform to the requirements in each case.

Configuration Management is a key function of Product Data Management System and is a key issue for PLM strategy and collaborative product development within networked organization.

CM II label is possible for organization but also for Software Product (in particular PDM systems like Windchill 6.8).

It is the reason why it is used in ATHENA Aerospace pilot component (c.f. ATHENA B4 and B5 deliverables related to Aerospace business scenarios and pilots).

It should illustrate usage of similar quality process in other business case to enable efficient interoperability and collaboration.
Here is the set of actors and stakeholders having a role on interoperability of the NCPD. They are defined in the Business scenarios of ATHENA (DB4.3).

- Engineers designing the product
- Legacy systems that support processes of organizations to achieve their goals – plus opera
- Organizations that are member of the network
- Organization that govern and manage the network and its members (organizations)
- Organization that design the network
- Organization that support operational usage and evolution of the platform
- Providers of ICT solution components and services
- Operational support of the low layers of the infrastructure
Here is the set of actors and stakeholders having a role for PLM, and considering that a NCPDO is set:

- NCPD actors
- Technicians manufacturing the product
- Technicians supporting the product
- Operators/users of the product
- Implied Enterprises and organizations for manufacturing, supporting, operation and support that can be members of the NCPDO
Here is the set of actors and stakeholders that are applicative solution providers, and have some responsibility on the architecture of the technical information system, in particular choice of standards, implementation of standards and usage of implementation of standards (using or not specific feature or having a policy based on conformance to standards):

1. Software component products or commodities - Application Service Providers for
   - Technical components (Domain independent) as RDBS, OS, Web servers, Web navigators, Workflow systems, BPM modeler and execution, office tools…
   - Applicative components (Domain Specific – e.g. PDM, ERP, Portfolio Management, e-Procurement or SCM systems)

2. Information system architects

3. Information system support, evolution and integration
   - Urbanism teams
   - EAI teams
   - B2B teams
Here is the list of identified important standards for NCPD, that should allow to derived other set of actors and stakeholders:

- Standardization organization
- Commercial software providers
- Designer and developers...

The identified Information and Communication Technologies are:

1. ICT and Services Standards
   - WEB W3C (HTTP, FTP, XML, RDF, OWL, BPEL)
   - Interoperability of application OMG (IIOP, CORBA, UML, MOF, MDA)
   - Support services (ITIL)
   - Design (CMMI, UP...)

2. Hardware component products providers – Environment Providers (who host your application)
3. Network infrastructure components providers – Internet Access Providers
4. Specification, Design, Development and support tools/services for software, hardware, networks components
Comparative analysis

Advantages

Process and services
- CMII Process of reference for efficient organization – best practice
- PLM Services Standardized API for PLM
- BPEL WEB oriented, orchestration, transaction support
- BPMN Rich conceptual model, rich expressivity
- Wfmc Workflow oriented, several technologies supported that allows usage of legacy system

Data
- STEP AP: Simple, IT technologies independent but computable, Consensus (ISO) by expert of the domains
- OWL ontology: Support of models federation, reasoning, WEB oriented
- XML vocabularies: WEB adapted, computable, numerous tools, extensibility
- UML models: Unified Graphical Language – not proprietary

Comparing the standards, we can point out for:

1. Process and services
   - CMII Process of reference for efficient organization – best practice
   - PLM Services Standardized API for PLM
   - BPEL WEB oriented, orchestration, transaction support
   - BPMN Rich conceptual model, rich expressivity
   - Wfmc Workflow oriented, several technologies supported that allows usage of legacy system

2. Data
   - STEP AP: Simple, IT technologies independent but computable, Consensus (ISO) by expert of the domains
   - OWL ontology: Support of models federation, reasoning, WEB oriented
   - XML vocabularies: WEB adapted, computable, numerous tools, extensibility
   - UML models: Unified Graphical Language – not proprietary
Comparative analysis

Differences

Process and services

- CMII Process that can’t be computed
- PDM Services is an Application Programming interface based on CORBA and MDA
- PLM services provides Service definition based on WEB services technology using UML (based on MDA)
- BPEL WEB: orchestration and WEB technologies dependency
- BPMN: graphical notation with rich concepts
- Wfmc Workflow oriented: based on Wfmc architecture, more adapted to legacy as technology independent
  ⇒ all based on different conceptual models and technological frameworks

Data

- STEP AP: user standards, entity-relation based, ICT independent
- Consensus (ISO) by expert of the domains
- OWL ontology: Support of models federation, reasoning, WEB oriented
- XML vocabularies: WEB adapted, computable, numerous tools, extensibility
- UML models
  ⇒ All based on different meta models and paradigms

Comparing Process and services, what differentiates the different standards is the following:

- CMII Process that can’t be computed
- PDM Services is an Application Programming interface based on CORBA and MDA
- PLM services provides Service definition based on WEB services technology using UML (based on MDA)
- BPEL WEB: orchestration and WEB technologies dependency
- BPMN: graphical notation with rich concepts
- Wfmc Workflow oriented: based on Wfmc architecture, more adapted to legacy as technology independent
  ⇒ all based on different conceptual models and technological frameworks

Concerning Data, the main differentiators are the following:

- STEP AP: user standards, entity-relation based, ICT independent
- Consensus (ISO) by expert of the domains
- OWL Ontology: Support of models federation, reasoning, WEB oriented
- XML vocabularies: WEB adapted, computable, numerous tools, extensibility
- UML models

It is important to point that all the concerned standards are based on different meta models and paradigms.
But it should be possible to enabled interoperability providing:

• A frame helping to use the characterize and federate the standards together, in order to take the best of each of them and leverage standard usage;

• Appropriate methodology.
It should be possible, as targeted for the NCPD, to use a federated way:

- Process non executable – not formally modeled, as CM II;
- Workflow- that should be technology independent (integration of legacy), as those defined by Wfmc, and eventually CBP defined by ATHENA CBP project;
- Published services that can be published automatically with different forms in order to be composed: CORBA, WEB services, other… Application servers – the target for NCPD is to reuse standardized specification provided by OMG Mantis or OASIS PLCS concerning PLM services;
- Information repositories with different underlying domain ontologies and underlying modeling and execution languages

Using together the different standards, it should be possible,

- Enhancing executable interconnected workflow with distributed enactment
- Enabling Semantic PLM on the WEB,
- Enabling federation of numerous legacy systems and infrastructure.
### Mapping and transformation services required

- Not possible to harmonized
  - Heterogeneity as a fact
- Harmonization us “progress and innovation”
- Harmonization and complex changing environment
  \[\Rightarrow\text{Focus} = \text{fast, efficient and low cost interconnection at business, application and technology level}\]
- Decoupling
- Several specialized and focused languages
- Virtualization \(\rightarrow\) globalization \(\rightarrow\)
  \[\Rightarrow\text{collaboration} \rightarrow\text{exchange}\]
  \[\Rightarrow\text{Mapping and transformation importance} \rightarrow\]
  \[\Rightarrow\text{exchange and sharing quality importance}\]

Starting point is to consider that it is not possible to harmonized: heterogeneity is a fact and will even become more important (same starting point than OMG when developing CORBA).

In addition:
- harmonization (and standardization) is most of the time going against progress and innovation;
- harmonization is not possible in a complex and always reconfigured network.

It is the reason why focus should be on capability to perform fast, efficient and low cost interconnection at business, application and technology level.

Business, application and technology levels should be decoupled. It is also important to admit that specialization on different phases of the lifecycle will imply usage of different specialized and focused languages. Finally virtualization and globalization will imply more numerous collaboration and consequently required more and more exchange. In such a context, mapping and transformation services are an important item to set up a collaboration platform. A particular attention should be done on exchange and sharing quality.
Integration and federation

Integration
– centralized decision and harmonization
– Non viable according the future trends if networks

Federation
– Fast reconfiguration
– Distributed semantic resources
– Mapping and transformation facilities
– Set of standardized and open components (commodities)

Technologies and standards can be classified according they are more adapted to integration and federation

Integration and Federation are two very different approaches, based on different strategies. For integration, it implies centralized decision and harmonization. It is non viable to target integration alone according the future trends if networks.

Federation is more dealing with fast reconfiguration, distributed semantic resources, mapping and transformation facilities and finally set of standardized and open components (commodities).

But in the current context, integration and federation have to be considered together as complementary approaches, in particular when willing to address at the same time engineers needs and enterprise/program needs.

Technologies and standards can be classified according they are more adapted to integration and federation.

For example:
• RDF and OWL are more related to federation while RDBS and UML are more related to integration
• BPEL is more related to federation while Wfmc standards are more related to integration
What brings ATHENA results

Usage for a NCPD framework and platform
ATHENA provides the opportunity to identify how innovative approach for interoperability of enterprise applications can be applied to PLM strategy within the virtual enterprise, solving some identified and important gaps to enable such approach.

1. Issues, problems and needs were issued from establishment of Business as is and to be scenario for NCPD (ATHENA B4 project)
2. Proposals for models generated collaboration platform
   - From enterprise, domains, knowledge and applications models (A1 and A6 ATHENA Projects)
   - On Service oriented execution platforms (A5 ATHENA project)
3. Cross organizations business processes (A2 ATHENA project)
4. Transformation and Communication based on semantic mediation
5. Pilot integrating ATHENA Action line A and based on usage of standard for practical experiment and proof of concept was the opportunity to identify and classify the important standards that are real enablers of the interoperability required in order to establish a NCPD Platform (ATHENA B5 project)
6. Lessons learnt and practical way to address the problem are available on the synthesis of piloting activities and through the NCPD platform made available by EADS CCR.
Impact on standards

- Cross Organizational Business Process:
  - To be reflected in BPEL, BPMN and Wfmc standard
- Model Generated Platforms approach as Collaboration space enabler
- Federative approach
  - Semantic WEB technologies (Ontology- semantic mediation)
  - knowledge based approach (Enterprise Modeling-AKM Task patterns)

- Federative and Integrative
  - Incompatible for tool integration (technical level)
  - links through model of reference (semantic)

- Future work item
  - proposal for Manufacturing Standardization bodies based on industrial pilots (e.g. S.E.I.N.E. project)
  - Dissemination through networks (GALIA, AFNET, ASD STAN)

The ATHENA impact on standards should be:
- For Cross Organizational Business Process, enrichment of BPEL, BPMN and Wfmc standards in order to cover identified needs during NCPD piloting activities
- For Model Generated Platforms approach with well defined Enterprise Application Agile architecture and communication/transformation services are real enablers to set up a collaboration space
- For Semantic WEB technologies and knowledge based approach are accurate for federative approach, as illustrated by approach adopted by B4 to analyze relationships between needs and proposed solutions.
  Outputs of ATHENA to consider are ATHENA Ontology Project semantic mediation tools for business documents and messages, and task patterns for active knowledge visual modeling.

Federative and Integrative approaches are most of the time incompatible when willing to provided integrated tools (technical level), as illustrated when trying to integrate Ontology tools with Cross-organizational Business Process tools. Nevertheless links should be established between communities to avoid creation of redundant conceptual and semantic model of reference (e.g. OWL-S or task patterns against BPEL, XPDL and BPMN) that will make it difficult to share and exchange information between tools based on these different paradigms.

Future work item proposal for Manufacturing Standardization bodies based on industrial pilots (e.g. S.E.I.N.E. project) and dissemination of ATHENA research and piloting results through existing networks concerned by Collaborative Product Design (GALIA, AFNET, ASD STAN, etc)
To go further, it is possible to refer to ATHENA Public WEB site, Enterprise Interoperability Centre WEB site and finally the ATHENA Aerospace piloting WEB Site. In particular, the last one propose complementary material related to Product Data Engineering:

- Online prototypes and demonstrators, based on open source and free tools in order to remove licenses and price barriers for those willing to practice
- Business and Pilot scenarios
- Reference to the described standards and their usage in the NCPD
- Complementary training support on the presented standards developed during the project that are part of ITIN courses.
Complements

Illustrations and technical aspects

More on the virtualization of the product

Concurrent engineering today implies production and usage of multidisciplinary 4D (3 Space Dimension + time dimension for simulations) e-Models (i.e. models produced by means of software and saved as electronic documents).

Such models are very various. In this example coming from Airbus, the considered models are:
- Master geometry
- Space allocation mock-up
- Definition models
- System installation requirement dossier
- Equipment Installation Requirement dossier
- Stress Design Reference bases…

As it is very difficult within virtual enterprise to have only one tool per kind of model, it implies to be able to share and exchange these models between heterogeneous tools, and consequently to have models that are software products independent. Long term retention and archiving also imply such kind of requirement: as software technologies and products are evolving faster and faster, with several cycles during lifecycle of one product, changes of software product versions and software products will occur several time during a program implying product data migration. Impact of such changes can be reduced only making produced eModels software product independent.

E.g.: Some Business Units of EADS were using CADDS software product from Computer Vision. PTC bought Computer Vision some years ago. Or PTC was proposing a new innovative CAD system, called Pro-Engineer, that was a concurrent of CADDS. PTC decided to stop evolution of CADDS, in order to push Pro-Engineer. The minimum support was ensured for CADDS, at an expensive cost. Finally EADS concerned Business Units had to select a new CAD tool and to perform important migration of existing models. But the used Software Product was not Pro-Engineer… but CATIA, the competitor!
More on Virtualization of the enterprise with Networked organizations sharing Product Data

Here is an example extracted from Aerospace industry.

The OEM is on top, and is working with a complex network for the different phases of the lifecycle of the product: engineering, manufacturing and support.

At each stage, several layers exist for established relationships. The OEM have direct contact with some enterprises, that themselves have their own network that is also concerned by the product.

During design phase, the concerned stakeholders are:
  - Service Suppliers
  - Risk Sharing Partners
  - Global Suppliers of level 1, 2, 3...
  - Equipment Suppliers of level 1, 2, 3...
  - Engine Suppliers of level 1, 2, 3...

During Manufacturing phase, the concerned stakeholders are:
  - Suppliers of level 1, 2, 3...

Very often, partners responsible of design of a sub-components are also responsible of the manufacturing.

During support phase, the stakeholders are:
  - Civil airlines and airlines manufacturer suppliers
  - Military agencies with military agencies manufacturer suppliers

All these actors have to exchange and shared configured and controlled product data for design, simulation, manufacturing and support.

In such a context, other important stakeholders are the long term product data archiving and retention, as the International Product Data Standards organizations.
More on Virtual Aircraft for early involvement of downstream activities through usage of Advanced Systems Simulation

As the target is today early involvement of downstream activities, i.e. production and support, and due to advance visual modeling and simulation capabilities enhanced by Information Technologies and Computers, the goal of the VIVACE research program is to develop methods and tools to support effective usage of virtual aircraft (numeric aircraft – static and dynamic mockups - defined by means of Computer Aided design tools) within the industrial environment.

As a consequence, product data that were initially static 2D drawing, are today more complex as they are 3D models (static mockups) and 4D Models (dynamic mockups, with the dimension of time). It can be even more complex as several times can be considered, with different references and purpose (e.g. real time, logical time…) according the considered system (mechanical system for cinematic, production system, simulation system…).

It is why Simulation Lifecycle Management is being considered as one of the important issue within the VIVACE research program.
More on Product Lifecycle Management

Product Lifecycle management considers the product from requirements to recycling phases. Product data and metadata are consequently to be controlled and managed in configuration for a very long duration: duration of an aircraft product that can be up to 50 years.

Or metadata are defined by different actors: software tools providers, design offices and finally product data standards or model of reference. In order to allow long term retention and to be software tools and technology independent, it means that the used meta-models are also to be controlled and managed in configuration.

Importance of manufacturing standards comes from there are not produced nor controlled by Information and Technology community; consequently, changes rhythm and horizon of usage is more adapted to industrial programs than to the ICT ecosystem, where rhythm of change is higher and horizon shorter (new technology every year, new version of a software product tool every 6 months, support proposed for a version of a commercial tool is about 2 years except or is charged a high cost.

It points out that some strategic governance is required having in mind such a context.
How to ensure efficient global Configuration Management and Product Information/data Coherency between different PDM Systems?

From technical information system management point of view, numerous difficulties exist.

A PDM Application is at the same time:

- An actor and a resource of the business processes of the enterprise (Design, Production, Support…)
  - Defined by the methods of the Design Office
  - Used in operation by programs and enterprises during the different phases of the lifecycle of the product
- A part of the information and communication system under the responsibility of IT department, that should ensure an efficient support of the business processes
- A software based solution, that is more and more based on usage of commercial software product.

Or the software products are themselves integration of pre-existing of the shelves software components, based on implementation of different standards coming from numerous standardization bodies and validated by experiences of different user groups.

Due to the enterprise virtualization logic, all the components have their own independent and uncorrelated lifecycle.

It created complex interdependency between the actors and created risks and difficulties to govern and control evolution of the technical enterprise information system.
How to ensure efficient Product Data Exchange, Sharing and Long Term Retention supporting Business Processes?

Different actors and stakeholders are concerned by the different phases of the lifecycle of an Aerospace product.

All have their own concerned and business processes, depending of Business Information they need to manage. Or today both Business Processes and Information systems are supported by Applications, that more and more are based on software products.

The product data are consequently distributed across different organizational and applicative boundaries. But coherency of controlled product data managed in configuration is particularly important in order to reduce costs and risks for product engineering, exploitation and support.

As a consequence, there is a need to be able to federate legacy PDM, SCM, ERP and support systems in particular to be able to exchange, share and retain all the product data required for efficient collaboration and contractualisation of this collaboration.
How to ensure seamless collaboration of designers and technicians despite heterogeneous environment (process, methods, applications and software products)?

In order to allow engineers to work seamlessly together, a lot of interoperability gaps exists:

- **Heterogeneous IT systems**
- **Heterogeneous Software products**
- **Heterogeneous methods, process and organization of product data defined by the design offices and the program**
- **Heterogeneous interested for the involved enterprises and communities two persons are belonging too**

If it is not the role of these actors to decide how to organize the whole enterprise and the resources of the program, it is important not to forget that if provided infrastructure does not allow them to work, they will decide by their own to use other methods and tools that will not be under control and that will allow them to perform their task, resulting in lost of visibility for programs and enterprises, and difficulties to optimize the performance of the organizations. In addition, it could be source of numerous risks and costs for the enterprises.
Technologies

EXPRESS and other STEP technologies
STEP files (ISO 10303-21)

- STEP files: the most widely used data exchange form of STEP.
- Easy to read
  - ASCII structure
  - one instance per line
- format defined in ISO 10303-21 Clear Text

**Encoding of the Exchange Structure**
- encoding mechanism to represent data according to a given EXPRESS schema
- not the EXPRESS schema itself
- also called p21-File, STEP Physical File, STEP models
- file extensions .stp and .step

(Wikipedia)

STEP-File is the most widely used data exchange form of STEP. Due to its ASCII structure it is easy to read with typically one instance per line. The format of a STEP-File is defined in ISO 10303-21 Clear Text Encoding of the Exchange Structure. ISO 10303-21 defines the encoding mechanism on how to represent data according to a given EXPRESS schema, but not the EXPRESS schema itself. STEP-File are also called p21-File and STEP Physical File. The file extensions .stp and .step indicate that the file contain data conforming to STEP Application Protocols.

A typical example looks like this:
ISO-10303-21; HEADER; FILE_DESCRIPTION( /* description */ ('A minimal AP214 example with a single part'), /* implementation_level */ '2;1'); FILE_NAME( /* name */ 'demo', /* time_stamp */ '2003-12-27T11:57:53', /* author */ ('Lothar Klein'), /* organization */ ('LKSoft'), /* preprocessor_version */ '', /* originating_system */ 'IDA-STEP', /* authorization */ ''); FILE_SCHEMA ('AUTOMOTIVE_DESIGN { 1 0 10303 214 2 1 1}'); ENDSEC; DATA;

#10=ORGANIZATION('O0001','LKSoft','company');
#11=PRODUCT_DEFINITION_CONTEXT('part definition',#12,'manufacturing');
#12=APPLICATION_CONTEXT('mechanical design');
#13=APPLICATION_PROTOCOL_DEFINITION('automotive_design',2003,#12);
#14=PRODUCT_DEFINITION('0',$,#15,#11);
#15=PRODUCT_DEFINITION_FORMATION('1',$,#16); #16=PRODUCT('A0001','Test Part 1','',(#18)); #17=PRODUCT_RELATED_PRODUCT_CATEGORY('part',$,(#16));

**HEADER section**
As you can see from the example given above the file is split into two sections following the initial keyword ISO-10303-21: The HEADER section has a fixed structure consisting of 3 to 6 groups in the given order. Except for the data fields time_stamp and FILE_SCHEMA all fields may contain empty strings.

FILE_DESCRIPTION : description and implementation_level. The version and conformance option of this file. Possible versions are 1 for the original standard back in 1994, 2 for the technical corrigendum in 1995 and 3 for the second edition. The conformance option is either 1 for internal and 2 for external mapping of complex entity instances. Most often you will find here the value "2;1". Throwing external mapping is also possible but only very rarely used. The values '3;1' and '3;2' indicate extended STEP-Files as defined in the 2001 standard with several DATA sections, multiple schemas and FILE_POPULATION support.
STEP files (ISO 10303-21)

- File structure
  - Header section including schema used and other metadata
  - Data section

- Similar to XML documents without the schema (DTD, XML Schema or other)

FILE_NAME: name of this exchange structure. It may correspond to the name of the file in a file system or reflect data in this file. There is no strict rule how to use this field.

time_stamp: indicates the time when this file was created. The time is given in the internal data time format ISO 8601, e.g., 2003-12-27T11:57:53 for 27 of December 2003, 2 minutes to noon time.

author: the name and mailing address of the person creating this exchange structure

organization: the organization to whom the person belongs to

preprocessor_version: the name of the system and its version which produces this STEP-file

originating_system: the name of the system and its version which originally created the information contained in this STEP-file.

authorization: the name and mailing address of the person who authorized this file.

FILE_SCHEMA. For version 2 only one express schema together with the object identifier of the schema can be listed here. From version 3 on this is relaxed for several entries. The last 3 header groups are only valid from version 3 on.

The concept of FILE_POPULATION is very close to schema_instance of SDAI. Unfortunately during the standardization process it was not possible to achieve an agreement to get these concepts fully synchronized. Therefore JSDAI adds further attributes to FILE_POPULATION as intelligent comments to cover all missing information from schema_instance. This is supported for both import and export.

SECTION_LANGUAGE allows to assign a default language for either all or for a specific data section. This is needed for those Express schemas which do not provide the capability to specify in which language string attributes of entities such as name, and description are given.

SECTION_CONTEXT provide the capability to specify additional context information for all or single data sections. This can be used e.g. for STEP-APs to indicate which conformance class is covered by a particular data section.
1. First indicate that the file is a part 21 file, so indicate the used syntax. Here no name space indicated. Implicitly, it’s ISO 10303. But there’s no pass to reach it as STEP is implementation independent, we are not supposed to have some Internet connection.

2. Second provide a HEADER section describing the file (name, description, author, date, …) and more especially the schema used. It will be possible to validate the data using the EXPRESS schema. Idem for the path to reach the schema.

3. Third, data section, where each object is identified by a number preceded by #. It allows to refer an object in another object (equivalent to ID IDREF in XML).

DATA section
The DATA section contains application data according to one specific express schema. The encoding of this data follows some simple principles.

Instance name: Every entity instance in the exchange structure is given a unique name in the form "#1234". The instance name must consist of a positive number (>0) and is typically less than 2^63. The instance name is only valid locally within the STEP-file. If the same content is exported again from a system the instance names may be different for the same instances. The instance name is also used to reference other entity instances through attribute values or aggregate members. The reference instance by be defined before or after the current instance.

Instances of single entity data types are represented by writing the name of the entity in capital letters and then followed by the attribute values in the defined order within parenthesis. See e.g. "#16=PRODUCT(...)" above.

Instances of complex entity data types are represented in the STEP file by using either the internal mapping or the external mapping.

External mapping has always to be used if the complex entity instance consist of more than one leave entity. In this case all the single entity instance values are given independently from each other in alphabetical order as defined above with all entity values grouped together in parentheses.

Internal mapping is used by default for conformance option 1 when the complex entity instance consists of only one leave entity. The encoding is similar to the one of a single entity instance with the additional order given by the subtype definition.

Mapping of attribute values: Only explicit attributes get mapped. Inverse, Derived and re-declared attributes are not listed since their values can be deducted from the other ones. Unset attribute values are given as ".". Explicit attributes which got re-declared as derived in a subtype are encoded as ".TRUE." in the position of the supertype attribute.

Mapping of other data types: Enumeration, Boolean and logical values are given in capital letters with a leading and trailing dot such as ".TRUE.. For characters with a code greater than 126 a special encoding is used. The character sets as defined in ISO 8859 and 10646 are supported. Note that typical 8 (e.g. west European) or 16 (Unicode) bit character sets cannot directly be taken for STEP-file strings. They have to be decoded in a very special way. Integers and real values are used identical to typical programming languages The elements of aggregates (SET, BAG, LIST, ARRARY) are given in parenthesis, separated by ",". Care has to be taken for select data types based on defined data types. Here the name of the defined data type get mapped too.

History
Some details to take care of:
The first edition of ISO 10303-21:1994 had some bugs and got fixed by a Technical Corrigendum. If you want to study this standard you better read...
The second edition ISO 10303-21:2002, including all the fixes and extensions for several data sections.
Part 21 defined two conformance classes. They differ only in how to encode complex entity instances.

Conformance class 1 which is always used enforce the so called internal mapping which is more compact.

Conformance class 2 which is not used in practise enforces the external mapping always. In theory this would allow better AP interoperability since a postprocessor may know how to handle some supertypes, but may not know the specified subtypes.

The 1st edition of part 21 enforces the use of so called SHORT NAMES, which is optional in the 2nd edition. In practise SHORT NAMES are not used today.
The 2nd edition allows for several data sections. In practise only one data section is used (1st edition encoding)
Here is the description on how a EXPRESS can describe a bookstore.

Bookstore is a set of several books, each book being described by a title, an author, a data, a ISBN reference and a publisher.

All these information are formalized through typed entities, that are used for production of the P21 file on the previous slide.
ISO 10303-22 is a part of the implementation methods of STEP with the official title Standard data access interface or simply SDAI. SDAI defines an abstract Application Programming Interface (API) to work on application data according to a given data models defined in EXPRESS. SDAI itself is defined independent of a particular programming language. Language bindings exist for

- Part 23 - C++ language binding of the standard data access interface
- Part 24 - C binding of the standard data access interface
- Part 27 - Java binding to the standard data access interface with Internet/Intranet extensions

The development of language bindings for FORTRAN and the interface definition language (IDL) of CORBA were canceled.

The original intent of SDAI and its bindings to programming languages was to achieve portability of software applications from one implementation to another. This was soon abandoned because there were only a few commercial implementations and they differed significantly in their detailed APIs. Today the term SDAI is sometimes used for all kinds of APIs supporting STEP, even if they only partially follow the strict functionality as defined in ISO 10303-22 and its implementation methods, or not at all. Part 35 of STEP (Abstract test methods for SDAI implementations) provides a formal way how to prove the conformance of an implementation with SDAI.

The main components of SDAI are:

- SDAI dictionary schema, a meta level EXPRESS schema to describe EXPRESS schemas
- Managing objects
  - SDAI session to control the whole SDAI environment for a single user/thread including optional transaction control
  - SDAI repository the physical (typically) container to store SDAI models and Schema instances, e.g. a database
  - SDAI model a subdivision of an SDAI repository, containing entity instance according to a particular EXPRESS schema
  - Schema instance a logical grouping of one or several SDAI models, making up a valid population according to a particular EXPRESS schema
- Operations
  - to deal with the managing objects
  - to create, delete and modify application data (entity instance, attribute values, aggregates and their members)
  - to validate application data according to all the constraints and rules specified in EXPRESS
ISO 10303-28 – XML Binding

- **STEP-XML** is the official name of ISO 10303-28
- **XML** to represent EXPRESS schemas and data
- **WITHIN scope**
  - Late Binding (schema independent)
  - Early Binding (schema dependent)
  - Schema-specific and schema-independent XML markup declarations mapping
  - Form of Part 28 XML documents
  - XML markup declarations for EXPRESS schemas
  - EXPRESS primitive data type values as XML element and attribute
- **OUTSIDE scope**
  - XML markup declarations for semantic intent of the schema
  - XML 2 Express mapping
  - Reverse mapping of EXPRESS 2 XML
  - XML schema final use mapping
- Allows to use AP within the WEB/XML technological frame
- Basis for schemas used with PLM services defined in WSDL (MANTIS PLM Services and PLCS PLM Services)

STEP-XML is the official name of ISO 10303-28; it specifies the use of the Extensible Markup Language (XML) to represent EXPRESS schemas (ISO 10303-11) and the data that is governed by those EXPRESS schemas.

The following specifications are WITHIN the scope of ISO 10303-28:

**Late Bound** XML markup declaration set, independent of all EXPRESS schemas, to describe the XML representation of the data governed by each schema

**Early Bound** XML markup declaration sets, for each of the schemas, to describe the XML representation of the data governed by that specific schema

The mapping between the schema-specific and schema-independent XML markup declarations

The form of XML documents containing EXPRESS schemas and/or data governed by EXPRESS schemas

The XML markup declarations that enables XML representation of EXPRESS schemas

The representation of EXPRESS primitive data type values as element content and as XML attribute values

The following specifications are OUTSIDE the scope of ISO 10303-28:

XML markup declarations that depend on the semantic intent of the corresponding EXPRESS schema

The mapping from a XML markup declaration to an EXPRESS schema. Note: Given a XML markup declaration set and its corresponding data set(s), it is possible to create an EXPRESS schema that captures the semantic intent of the data. However, this would requires an understanding of the meaning and use of the data that may not be captured by the XML markup declarations.

The mapping from an XML representation of an EXPRESS schema back to the initial EXPRESS schema

The mapping from the XML markup declarations that have been derived from an EXPRESS schema back to the initial EXPRESS schema

The mapping of the final use of a XML schema.
Inside the scope
- Mapping of EXPRESS constructs into the UML Interchange Metamodel
- For exchange conforming to the XMI standard

outside the scope
- purposes other than exchange using XMI
- EXPRESS constructs not aligned with XMI
  - global and local rules;
  - super type constraints;
  - expressions, functions, procedures, and constants;
  - explicit attributes re-declared as derived attributes;
  - remarks;
- UML to EXPRESS

XML support should make the binding useless
- Part 25 used by most of transformation tool (e.g. Component provided by Uninova within ATHENA pilots or Exff)

Note
Some difficulties to map some Express Construct
Due to difficulties to exchange UML model by means of XMI, such a standardized binding can be useless as soon as UML tool will not implement XMI the accurate way
Part 25 is nevertheless used by most of the transformation tool (e.g. Component provided by Uninova within ATHENA pilots or Exff)
PDE2: Standards to support Interoperability for Product Life cycle Management

Technologies

XML and Schema Language for XML

XML document

<?xml version="1.0"?>
<!DOCTYPE BookStore SYSTEM http://www.books.org/bookstore.dtd>
<BookStore>
  <Book>
    <Title>My Life and Times</Title>
    <Author>Paul McCartney</Author>
    <Date>July, 1998</Date>
    <Publisher>McMillin Publishing</Publisher>
  </Book>
  ...
</BookStore>

1. First, using a DOCTYPE declaration, with the name of the root element of the XML document instance

2. Indicate name and path of the DTD (could be local or on the WEB).

Note: it's also possible to define the DTD within the document, or to mix the both with priority to the DTD inside the XML document.

**Document Type Definition (DTD)**, defined slightly differently within the XML and SGML specifications, is one of several SGML and XML schema languages, and is also the term used to describe a document or portion thereof that is authored in the DTD language. A DTD is primarily used for the expression of a schema via a set of declarations that conform to a particular markup syntax and that describe a class, or type, of SGML or XML documents, in terms of constraints on the structure of those documents. A DTD may also declare constructs that are not always required to establish document structure, but that may affect the interpretation of some documents.

DTD is native to the SGML and XML specifications, and since its introduction other specification languages such as XML Schema and RELAX NG have been released with additional functionality.

As an expression of a schema, a DTD specifies, in effect, the syntax of an "application" of SGML or XML, such as the derivative language HTML or XHTML. This syntax is usually a less general form of the syntax of SGML or XML.

In a DTD, the structure of a class of documents is described via element and attribute-list declarations. Element declarations name the allowable set of elements within the document, and specify whether and how declared elements and runs of character data may be contained within each element. Attribute-list declarations name the allowable set of attributes for each declared element, including the type of each attribute value, if not an explicit set of valid value(s).
Document Type Definition
- Native in XML definition (also in SGML definition with minor deference)
- Simple and very used
- But:
  - no support for newer features of XML — most importantly, namespaces
  - Lack of expressivity
  - non-XML syntax to describe the schema

XML Schema: May 2001 W3C Recommendation
- XML Schema instance
  - XML Schema Definition (XSD), with ".xsd" extension
  - XML document
  - schema
    - set of rules for ‘valid’ XML document
    - Extends DTD capabilities – data types
    - large number of built-in and derived data types
    - Post-Schema-Validation Infoset (PSVI)

Schematron
- XML structure validation language
- presence or absence of patterns in trees.
- based on XPath
- Schematron can be used as an adjunct to DTDs or XML Schema
- Different kind of constraints

Schematron has been standardized to become part of ISO/IEC 19757 - Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation - Schematron.
Here is the description on how a DTD can describe a bookstore.

Bookstore is a set of several books, each book being described by a title, an author, a data, an ISBN reference and a publisher.

All these information are PCDATA, it means any kind of textual character set (by opposition to binary data that are not to be read by person but by automates).
Here is the description on how an XML-Schema schema can describe a bookstore.

Bookstore is a set of several books, each book being described by a title, an author, a data, a ISBN reference and a publisher.

All these information are now strings.

Expressiveness of XML Schema is higher, as numerous datatypes are available to describe the PCDATA. The number of occurrences can be detailed very precisely, and very complex types can be described.

Finally, XML-Schema is itself using XML syntax, and is consequently an XML document.
Schematron

Overview

The Schematron differs in basic concept from other schema languages in that it **not based on grammars** but on finding **tree patterns** in the parsed document. This approach allows many kinds of structures to be represented which are inconvenient and difficult in grammar-based schema languages. If you know XPath or the XSLT expression language, you can start to use The Schematron immediately.

And it has free and open source implementations available. The Schematron is trivially simple to implement on top of XSLT and to customize. (There are also implementations in Python and Perl)

The Schematron allows you to develop and mix two kinds of schemas:

- **Report elements** allow you to diagnose which variant of a language you are dealing with.
- **Assert elements** allow you to confirm that the document conforms to a particular schema.

The Schematron is based on a simple action:

First, **find** a context nodes in the document (typically an element) based on XPath path criteria;

Then, **check** to see if some other XPath expressions are true, for each of those nodes.

The Schematron can be useful in conjunction with many grammar-based structure-validation languages: DTDs, XML Schemas, RELAX, TREX, etc. Indeed, Schematron is part of an ISO standard (DSDL: Document Schema Description Languages) designed to allow multiple, well-focussed XML validation languages to work together. You can even embed a Schematron schema inside an XML Schema `<appinfo>` element or inside a RELAX NG schema.

[From http://www.schematron.com/overview.html]
Technologies

OWL and RDF
Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model using XML but which has come to be used as a general method of modeling knowledge, through a variety of syntax formats (XML and non-XML).

The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion “The sky has the color blue” in RDF is as a triple of specially formatted strings: a subject denoting “the sky”, a predicate denoting “has the color”, and an object denoting “blue”.

This mechanism for describing resources is a major component in what is proposed by the W3C’s Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF’s simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.

A collection of RDF statements intrinsically represents a labeled, directed pseudo-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models traditionally used in computing today. However, in practice, RDF data is often stored in relational database representations sometimes also called triple stores. As RDFS and OWL demonstrate, additional ontology languages can be built upon RDF.
Here is the description on how RDFS allows to describe a bookstore.

Bookstore is a set of several books, each book being described by a title, an author, a data, an ISBN reference and a publisher.

All these information are now literal resources

Expressiveness of RDFS is higher than DTDs, as it allows to create labeled, directed pseudo-graph and to reuse XML-Schema constructs.
An RDF document is different than from an XML-Schema document, as it refers to rdf and rdfs while an XML-Schema based document refers to xsd.
Ontology WEB Language

- Markup language for publishing and sharing data using ontologies on the World Wide Web
- Designed for use by applications instead of just presenting information to humans
- Facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics
- 3 increasingly expressive sublanguages:
  - OWL Lite
  - OWL DL (for reasoning)
  - OWL Full (for semantic networks)
- Several possible syntaxes
  - RDF-XML
  - RDF-XML simplified
  - N3
  - N-Triple
  - Full RDF...
- Focus of research into tools, reasoning techniques, formal foundations and language extensions.
- Major technology for the future implementation of a Semantic Web
- Target: framework for asset management, enterprise integration and the sharing and reuse of data on the Web
- More facilities for expressing meaning and semantics than XML, RDF, and RDF-S
- Ability to represent machine-interpretable content on the web

[Wikipedia]

Sublanguages

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

**OWL Lite** supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. OWL Lite also has a lower formal complexity than OWL DL; see the section on OWL Lite in the OWL Reference for further details.

**OWL DL** supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computed) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL.

**OWL Full** is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. The following set of relations hold. Their inverses do not.

- Every legal OWL Lite ontology is a legal OWL DL ontology.
- Every legal OWL DL ontology is a legal OWL Full ontology.
- Every valid OWL Lite conclusion is a valid OWL DL conclusion.
- Every valid OWL DL conclusion is a valid OWL Full conclusion.
Here is the description on how OWL, with RDF/XML syntax, allows to describe a bookstore. Bookstore is a set of several books, each book being described by a title, an author, a data, an ISBN reference and a publisher. 

All these information are now set of OWL classes, individuals and properties. Expressiveness of OWL is higher than RDFS, as it allows to create constraints and rules to support inference and reasoning engines.

This slide describes only the class and properties structure…
… while this slide describes the individuals.
Technologies

MDA-MOF
This figure summarizes the Model Driven approach supported by OMG’s Model Driven Architecture.

The Application Platform Independent Model with appropriate profile can be use to create the Application Platform Specific Model (through annotation or transformation) and then to generate code for applications. The reverse engineering should be possible for retro-engineering but also in order to support iterative development taking into consideration legacy operational enterprise systems that are continuously evolving.
Here is the description on how UML, with the corresponding XMI representation, allows to describe a bookstore.

Bookstore is a set of several books, each book being described by a title, an author, a data, an ISBN reference and a publisher.

UML main focus is on classes, not instances. The used paradigm is object class.

The model is used for software engineering, not for data exchange, distributed resources or semantic WEB knowledge bases.

XMI is targeting exchange of UML models between different UML modeling tools.
Technologies

WFMC workflow, BPMN, BPEL, OWL-S
From Information Technology point of view, the architecture will also consider the different repositories and roles that will help to define the use cases.
3 different times are considered:

The build type when the workflow process model is being defined

The first level of RunTime, when the workflow process model is deployed and made available through a workflow engine that implement enactment services.

The second level of RunTime, when an instance of the workflow process model is running, implying participants.
Numerous specifications exist, providing description of the concepts (dictionary), the architecture and specification for the different kind of interfaces.

Here is the description provided by Wfmc that is not complete and up-to-date.
Example with Change Management Process

Here is illustration of core concepts of Workflow systems through the user interface of Jawe:

- Process properties
- Participants
Here is the illustration of the running workflow engine enacting workflow models, based on Wfmc Standards. It illustrates:

- The enactment server, made available through CORBA and implementing Workflow facilities of OMG
- The administration component
Here is illustration of some other concepts used during the second level of runtime:

- Monitoring of the workflow process
- Work list handler
- WAPI (application invocation interface that is technology independent and implement by means of tool agents for different technologies to invoke).
- And finally, Wf-XML services that allow a modeling tool to interrogate a workflow engine concerning enacted workflow models and to import them.