The NASA STEP Testbed

Pan Galactic Engineering Framework (PGEF)

Implementing a STEP-based Intelligent Master Model for Systems Engineering and the “Model-Based Enterprise”

Stephen C. Waterbury

NASA / Goddard Space Flight Center

April 8, 2003
Aerospace PDE 2003 Workshop
at NIST

PGEF Acronyms and Definitions

- **API** ..... Application Programming Interface
- **CAD** ..... Computer-Aided Design
- **CAE** ..... Computer-Aided Engineering (typically, analysis)
- **Express** ..... an information modeling language
- **Express-X** ... maps between & creates views of Express models
- **PDM** ..... Product Data Management
- **PGEF™** ..... Pan Galactic Engineering Framework
- **PGPDM** ..... Pan Galactic PDM (a PGEF *application*)
- **SE** ..... Systems Engineering
- **STEP** ..... Standard for the Exchange of Product Model Data (ISO 10303)

The NASA STEP Testbed

- Purpose: Leverage STEP technology to support collaborative engineering in NASA projects
- Approach: Implement STEP-based software for Systems Engineering (SE) and Model-Based Enterprise (MBE)
  - Pan Galactic Engineering Framework (PGEF) software
  - NASA federated STEP-based Systems Engineering Repository
  - Collaborate with and support NASA centers in implementation and deployment

Goals of PGEF (Hub or "Server" Functions)

- Integration of model information from all engineering disciplines (electrical, mechanical, analysis, systems engineering, etc.)
- Standards-based Dictionary service for import and management of data dictionaries
- Unified view of all system modeling parameters (which are referenced in the data dictionary!) throughout the life cycle
- Management of Structured Product Documents* linked to CAX/SE models
- Standard model library and archive, enabling design re-use
- Support for "model-driven" acquisition (model-based specs, etc.)
- Full spectrum of model repository services, including CM and long-term data retention (archiving)

* Structured Product Documents: documents of which some portion is linked to (and should be validated against) product data. Examples: Requirements, ICD, Drawings, ECR, ECO, Specifications

Goals of PGEF (UI or "Client" Functions)

- Product Data Browser (analogous to Web browser)
- Searching and browsing model information from all engineering disciplines (electrical, mechanical, analysis, systems engineering, etc.)
- Dictionary terms search, browsing, creation, and editing
- Model parameter reporting, inspection, notification (by several methods) of changes or impacts, and display of applicable dictionary definitions
- Creation and editing of Structured Product Documents*, validated against applicable models and dictionaries in the repository
- User and tool interfaces to model libraries and archives
- Creation of various types of conceptual, functional, behavioral, and physical assembly trees for trade studies, digital mock-ups, and simulations

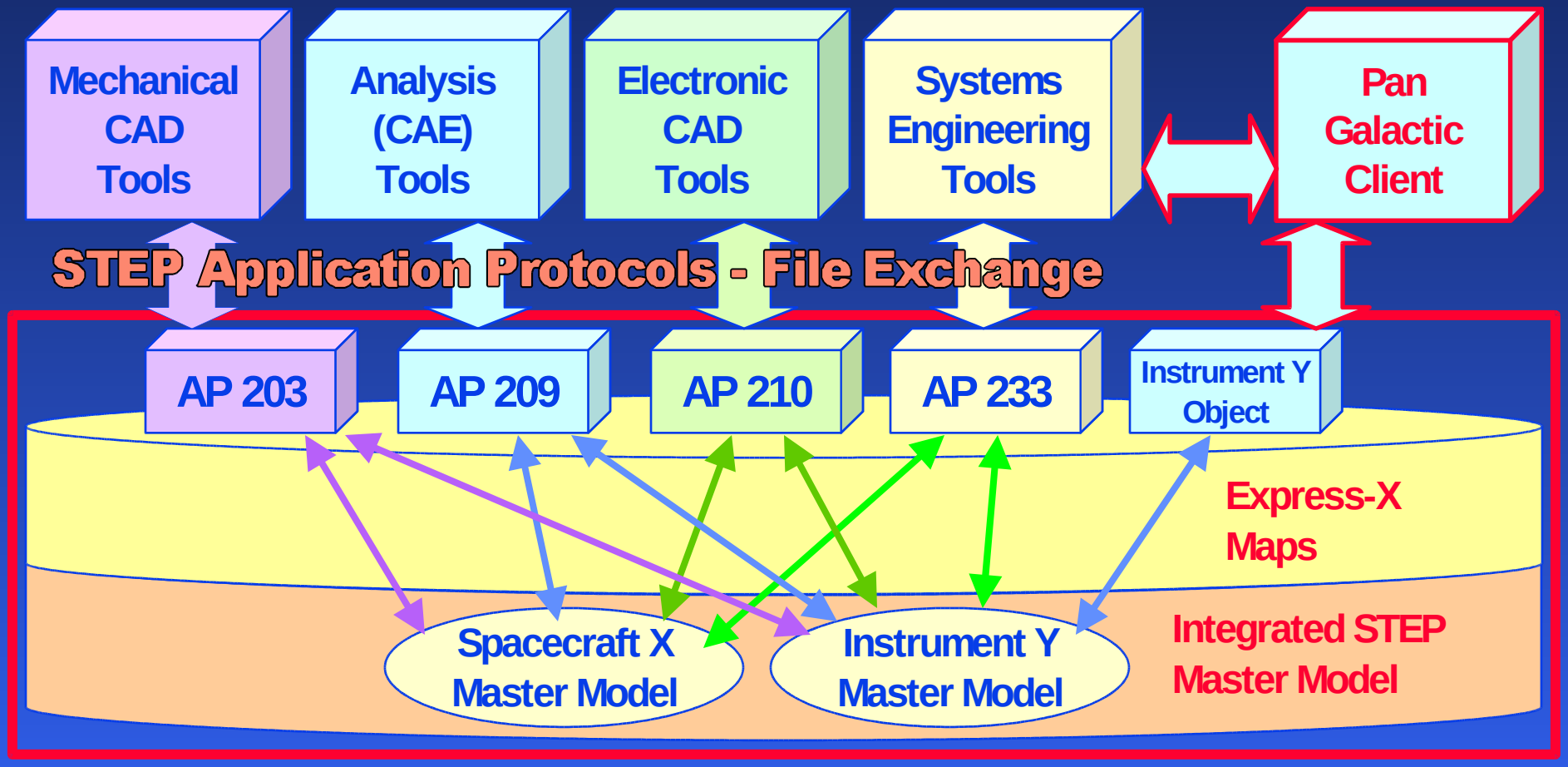
* Structured Product Documents: documents of which some portion is linked to (and should be validated against) product data. Examples: Requirements, ICD, Drawings, ECR, ECO, Specifications

Okay, fine, but why Python?!

- Very coherent language and library structure
  - **Deeply Object Oriented (built-in types are objects, too – strings, etc.)**
- Clear syntax – makes for code that's easy to understand and maintain
  - **Almost "pseudo-code"**
  - **Excellent automatic documentation generators**
- Platform-independent
  - **Possibly even more so than certain other languages that claim to be … ;^)**
  - **Only one implementation of Python interpreter on each platform**
- Plays well with other languages and on various platforms, so is an excellent integration language
  - **Extensible with C or C++**
  - **Has COM, CORBA, various other libraries**
  - **Is used by Google, Pixar, Verity, and others for this purpose**
- **Other cool stuff …**

Model-Based Enterprise

Master Model Integration Using STEP



Pan Galactic Server

The Product Master Model Evolution

Population of Master Model Contents Over the Project Life Cycle

Pre-Phase A

Integrated Mission Proposal

- Mission Parameters
- Requirements
- Functional Model
- Preliminary Components

Phase A/B

Prototyping and Analysis

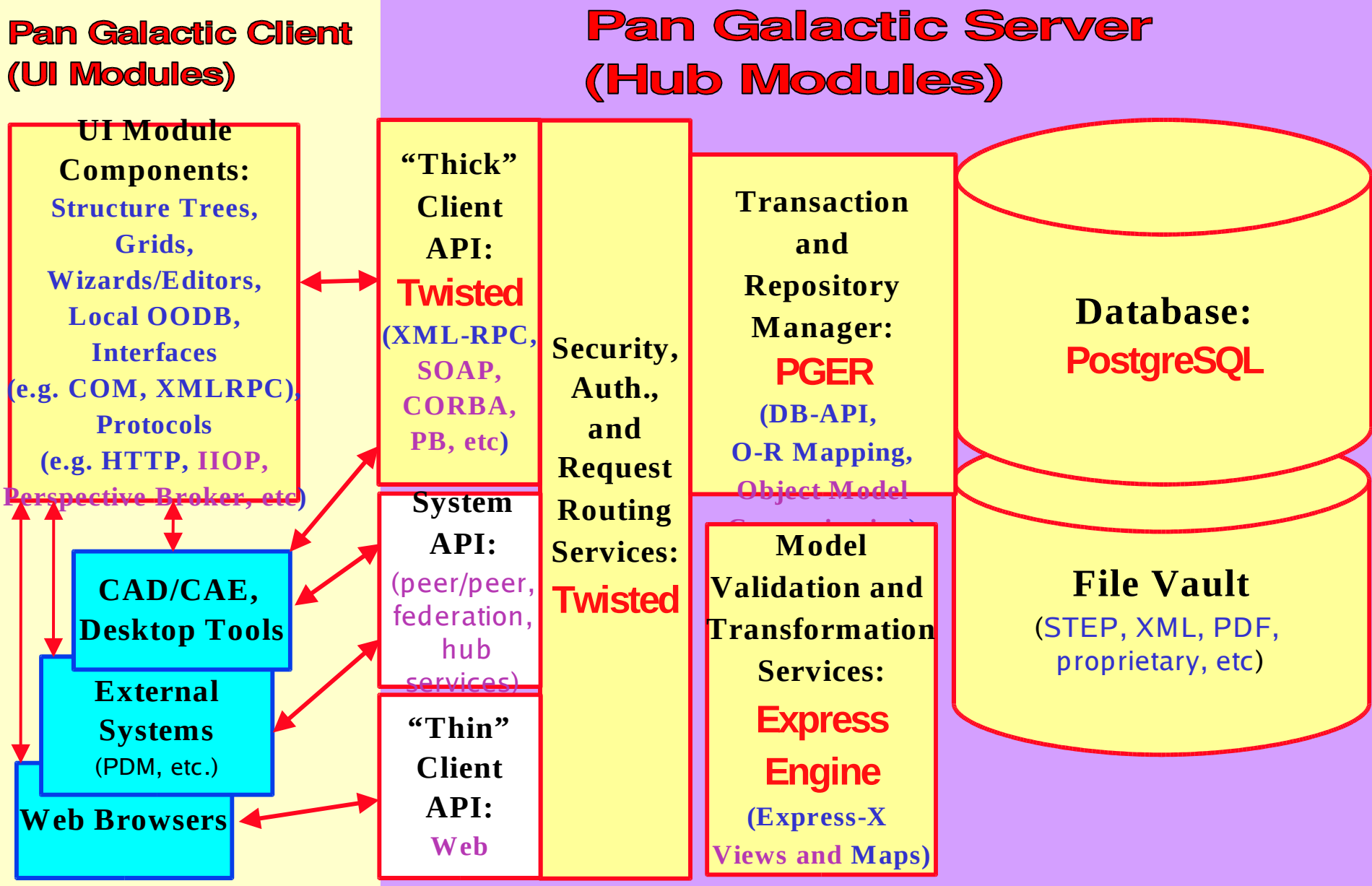
- Mission Parameters
- Requirements
- Functional Design
- Behavioral Model
- Physical Model

Phase C/D

Detailed Design, Build, and Test

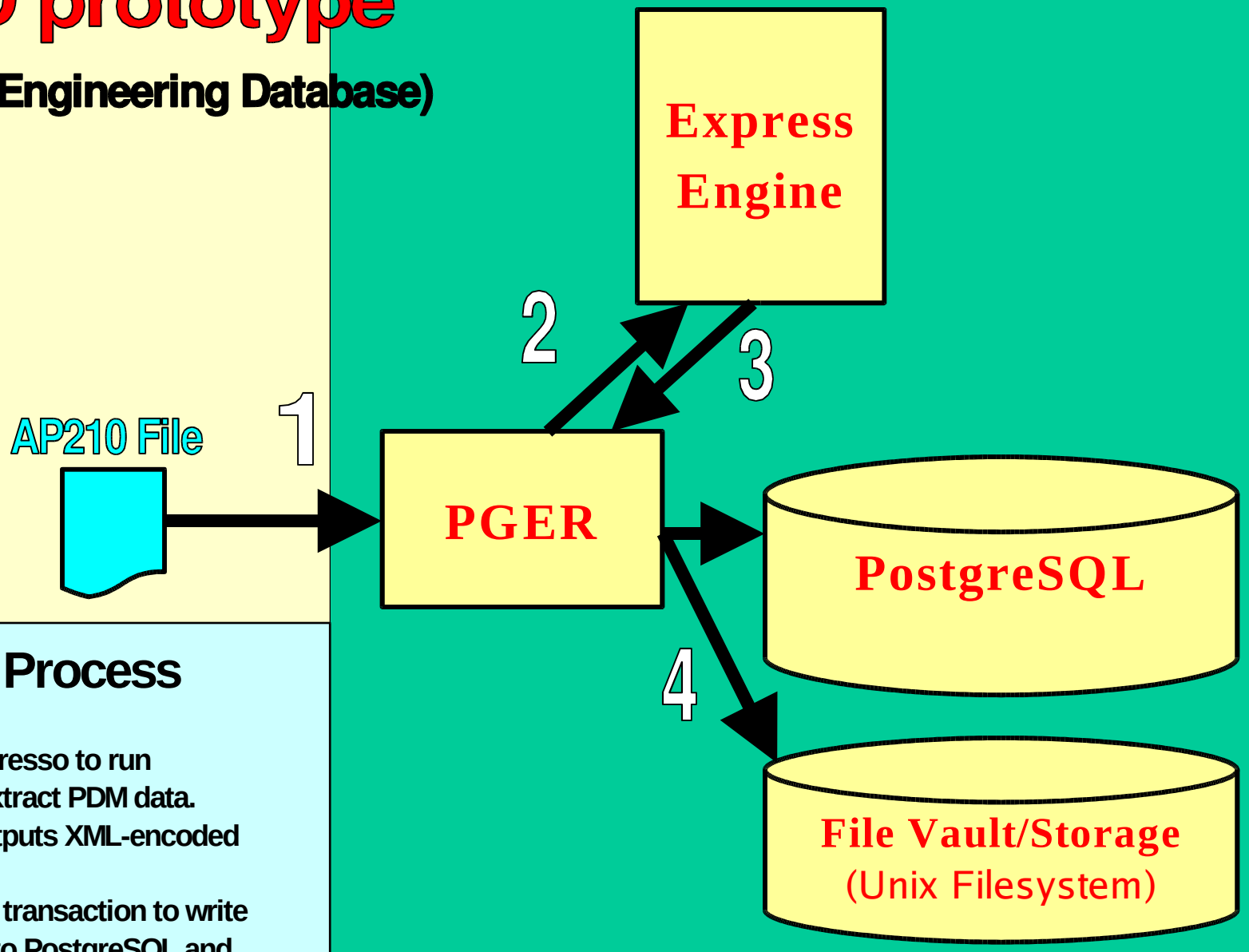
- Mission Parameters
- Requirements
- Functional Design
- Behavioral Design
- Interface Control Docs
- Physical Design/Arch.

Anatomy of the Pan Galactic Engineering Framework (PGEF)



NED prototype

(NASA Engineering Database)



Check-in Process

1. File upload
2. RM calls Expresso to run Express-X to extract PDM data.
3. Expresso outputs XML-encoded PDM data.
4. RM executes transaction to write mapped data into PostgreSQL and AP210 file to “vault” (mapped data contains a reference to the file).

Use Cases

- **Collaborative High-Level Mission Analysis**
  - Participants: several users, at possibly distributed sites
  - Roles: systems engineer, engineers in several different disciplines, principal investigator (scientist), [others?]
  - Processes:
    - Capture of initial mission parameters / requirements (via spreadsheet entry and/or other means) [--> further define what "capture" means, plus what are CM requirements for this process …]
    - Proposal and analysis of alternate mission scenarios (including technologies, specific vehicles/components/etc.) and record each with comments, parameters, calculations, analyses, etc. [--> further define types of data to go into these mission scenario analysis "packages"]

Use Cases

- **Distributed Multi-Disciplinary Simulations**
  - Participants: several users, at possibly distributed sites
  - Roles: engineers in several different disciplines, systems engineers, engineering management, project management
  - Processes:
    - Level 1: Creation and CM of assembly trees of possibly multi-disciplinary simulation models, with model parameters coordinated among discipline models
    - Level 2: HLA implementation; repository-driven “hub” services for distributed simulation with CM and histories
    - Levels 3-n TBD (further integration with design and analysis models and modeling processes, etc.)

Use Cases

- **Collaborative Multi-Disciplinary Design and Analysis**
  - Participants: several users, at possibly distributed sites
  - Roles: engineers (designers/analysts) in several different disciplines, [systems engineer,] [project management,] [others?]
  - Processes:
    - Level 1: Capture and CM of design and analysis models, documents, comments, etc. (essentially, basic PDM)
    - Level 2: Sharing of geometries and semantics between discipline models (ex.: Electromechanical scenario: “round-trip” model exchanges between disciplines)
    - Level 3: Derivation of analysis models from design models (ex.: Russell Peak’s “PBAM” concept) [--> this scenario has many levels and possible modes of implementation]
    - Levels 4-n TBD (simulations, optimizations, etc., etc.)

Use Cases

- **Structured Product Document (SPD) Management**
  - Participants: several users, at possibly distributed sites
  - Roles: engineers in several different disciplines, engineering management, project management, acquisition management
  - Processes:
    - Level 1: Creation, validation, CM, approval, and release/publishing of SPD's, including full workflow, event notification, digital sign-off, and repository/Web access
    - Level 2: Extensive use of Document Type Definitions (DTD's), templates, and reusable doc components
    - Level 3: Interactive interfaces with model CM process; SPD updating automated and user-monitored
    - Levels 4-n TBD (full integration with modeling process, etc.)

Part 2: Implementation

- **Structured Product Document (SPD) Management**
  - Participants: several users, at possibly distributed sites
  - Roles: engineers in several different disciplines, engineering management, project management, acquisition management
  - Processes:
    - Level 1: Creation, validation, CM, approval, and release/publishing of SPD's, including full workflow, event notification, digital sign-off, and repository/Web access
    - Level 2: Extensive use of Document Type Definitions (DTD's), templates, and reusable doc components
    - Level 3: Interactive interfaces with model CM process; SPD updating automated and user-monitored
    - Levels 4-n TBD (full integration with modeling process, etc.)