SESE TOUR 2022

Systems Engineering for a Sustainable World

Making engineering work products available in the long term through context-based interoperability

May 9th, 2022

Biography

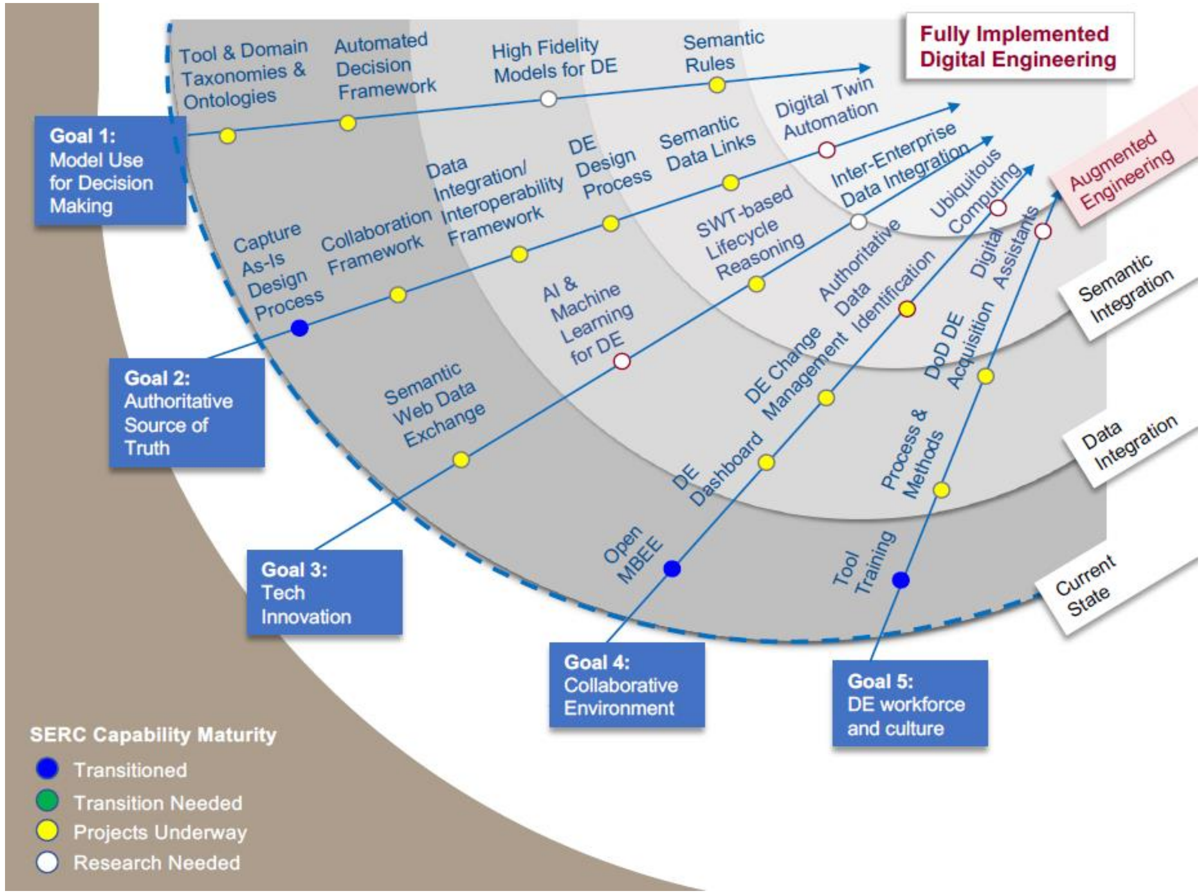
- Juan Carlos Mendo
  - He is Systems Engineer Model-Based Engineering, Boeing Research & Technology. As part of the Model-Based Engineering (MBE) team in Boeing R&D, Juan Carlos has been leading as a Product Owner several projects with focus on Data Interoperability and Digital Thread, digital collaboration with suppliers using Technical Data Packages (TDP) and Data Interoperability Standard Implementation. He has led multiple initiatives for commercial and defense products and customers with the end goals of transitioning to Model Based Systems Engineering (MBSE) and Model Based Development (MBD) at Boeing.
  - Learn more: https://www.linkedin.com/in/jcmendo/

- Assoc. Prof. Jose María Alvarez Rodríguez
  - He is Associate Professor within the Department of Computer Science and Engineering of Carlos III University of Madrid (UC3M). He is Master of Computer Science (2007) and Bachelor of Computer Science (2005) by the University of Oviedo. He has also participated in more than 30 research projects in different competitive programmes and he is the author of more than 80 publications and other research works. He is member of some standardization bodies such as ISO (Artificial Intelligence working group), OMG, ProSTEP, LOTAR, INCOSE (ontologies working group) and INCOSE.
  - (Spain).Learn more: http://www.josemalvarez.es/web/

Agenda / Structure of the presentation

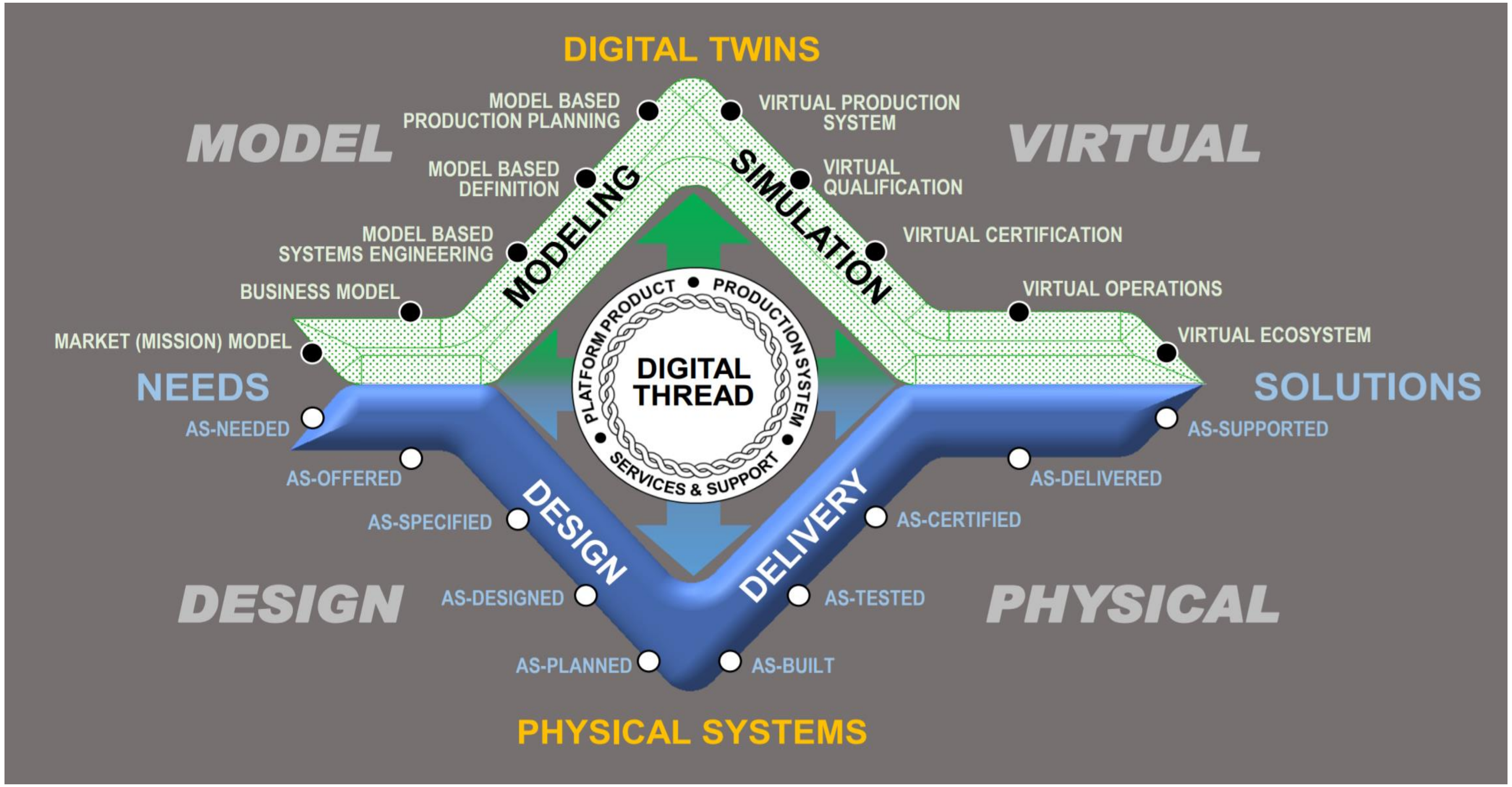
1. Digital thread: digitalization of the engineering lifecycle
2. Capturing the collaborative systems engineering context
3. Overview of LOTAR
4. Case Study: linking requirements and physical models
5. Conclusions and Future steps

Digital thread: digitalization of the engineering lifecycle


Fully Implemented Digital Engineering
Tool & Domain Taxonomies & Ontologies
Automated Decision Framework
High Fidelity Models for DE
Semantic Rules
Goal 1: Model Use for Decision Making
Data Integration/ Interoperability Framework
DE Design Process
Semantic Data Links
Digital Twin Automation
Inter-Enterprise Data Integration
Ubiquitous Computing
Augmented Engineering
Capture As-Is Design Process
Collaboration Framework
SWT-based Lifecycle Reasoning
AI & Machine Learning for DE
Authoritative Data Identification
Digital Assistants
Semantic Integration
Goal 2: Authoritative Source of Truth
Semantic Web Data Exchange
DE Change Management
DoD DE Acquisition
DE Dashboard
Process & Methods
Data Integration
Open MBEE
Tool Training
Current State
Goal 3: Tech Innovation
Goal 4: Collaborative Environment
Goal 5: DE workforce and culture
SERC Capability Maturity
● Transitioned
● Transition Needed
● Projects Underway
○ Research Needed

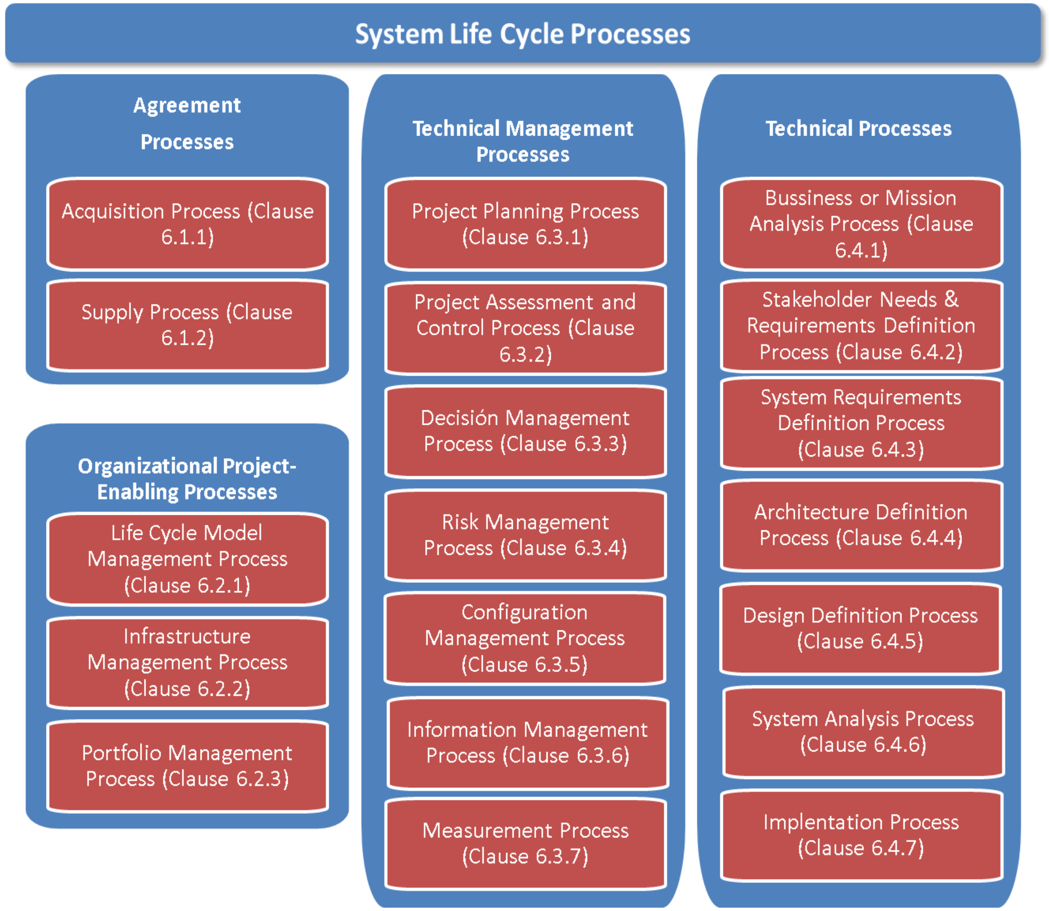
- Source: https://www.researchgate.net/publication/340649785_AI4SE_and_SE4AI_A_Research_Roadmap

Digital thread: digitalization of the engineering lifecycle



Source: Boeing

Lifecycle processes



System Life Cycle Processes

Agreement Processes

- Acquisition Process (Clause 6.1.1)
- Supply Process (Clause 6.1.2)

Organizational Project-Enabling Processes

- Life Cycle Model Management Process (Clause 6.2.1)
- Infrastructure Management Process (Clause 6.2.2)
- Portfolio Management Process (Clause 6.2.3)

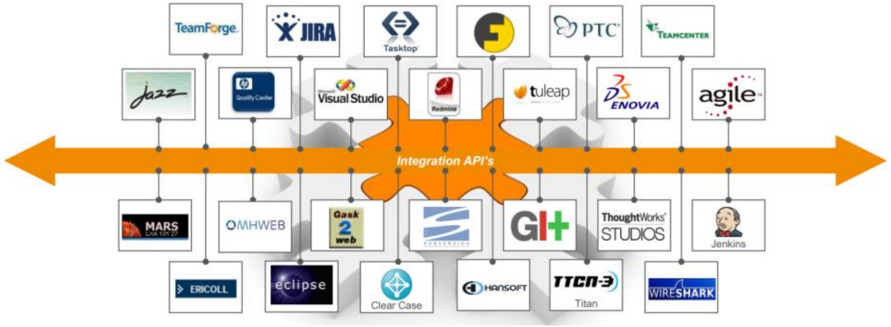
Technical Management Processes

- Project Planning Process (Clause 6.3.1)
- Project Assessment and Control Process (Clause 6.3.2)
- Decisión Management Process (Clause 6.3.3)
- Risk Management Process (Clause 6.3.4)
- Configuration Management Process (Clause 6.3.5)
- Information Management Process (Clause 6.3.6)
- Measurement Process (Clause 6.3.7)

Technical Processes

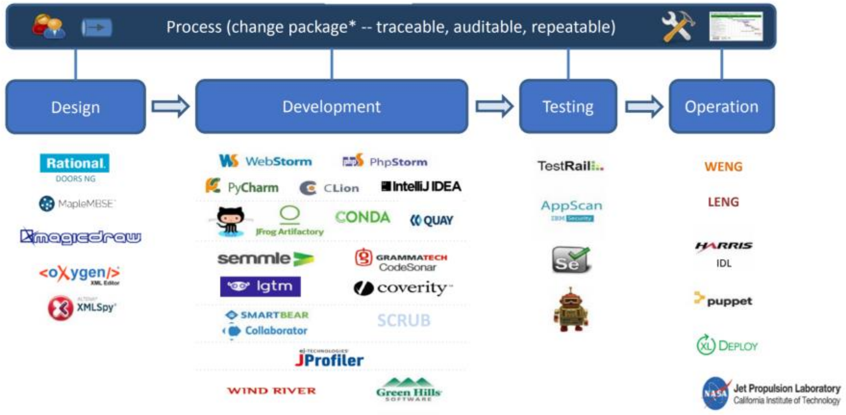
- Bussiness or Mission Analysis Process (Clause 6.4.1)
- Stakeholder Needs & Requirements Definition Process (Clause 6.4.2)
- System Requirements Definition Process (Clause 6.4.3)
- Architecture Definition Process (Clause 6.4.4)
- Design Definition Process (Clause 6.4.5)
- System Analysis Process (Clause 6.4.6)
- Implentation Process (Clause 6.4.7)

Engineering (and corporate) environment



Mats Berglund (Ericsson)

http://www.ices.kth.se/upload/events/13/84404189f85d41a6a7d1cafd0db4ee80.pdf



Safety-Critical Software Environment

Source: https://www.nist.gov/system/files/documents/2019/04/05/14_delp.pdf

Digital thread: digitalization of the engineering lifecycle

Type of Exchange	Description	Examples
Full Collaboration	Multiple parties use a common configuration management system	Access to single hosted corporate network Cloud environment
Model Interoperability	Multiple parties augment the same digital model	Supplier/Buyer software synchronization Data Standards
Digital Data Exchange	Using data translation tools for consumption	**Supplier Requirements Exchange (SRX) System** Cameo Inter-op
Data Hand-Off	Standard practice for documents	Email

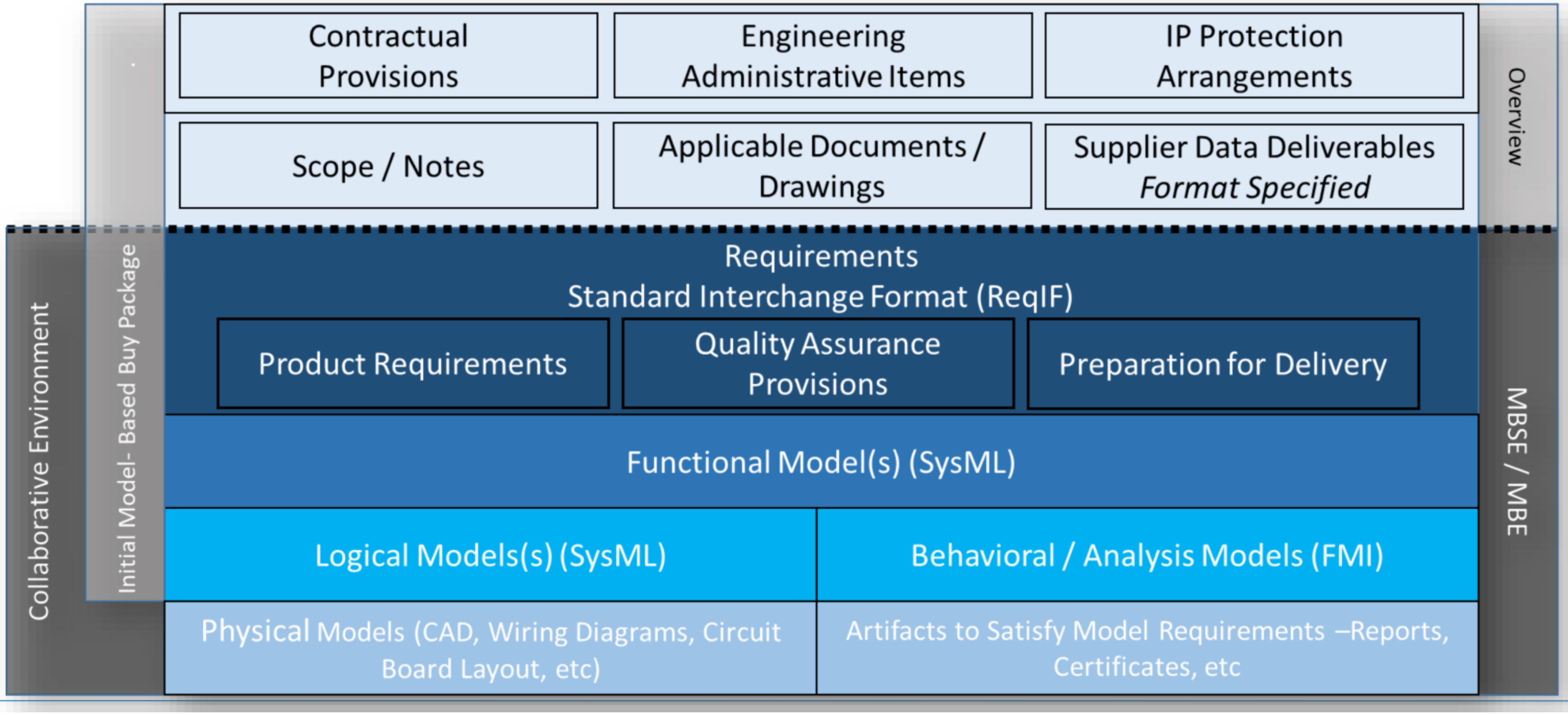
Increased Collaboration

Improved Digital Thread

Source: Boeing-McGowan-FrameworkForDevelopingMod-MBSE-Open (GDPIS-SUMMIT 2019)

Digital thread: digitalization of the engineering lifecycle

- **Utilization of electronically readable industry standard formats for consumption**
- **Selection of models for collaborative development based on business case**



Contractual Provisions

Engineering Administrative Items

IP Protection Arrangements

Scope / Notes

Applicable Documents / Drawings

Supplier Data Deliverables *Format Specified*

Overview

Requirements
Standard Interchange Format (ReqIF)

Product Requirements

Quality Assurance Provisions

Preparation for Delivery

Functional Model(s) (SysML)

Logical Models(s) (SysML)

Behavioral / Analysis Models (FMI)

Physical Models (CAD, Wiring Diagrams, Circuit Board Layout, etc)

Artifacts to Satisfy Model Requirements –Reports, Certificates, etc

Collaborative Environment

Initial Model- Based Buy Package

MBSE / MBE

Source: Boeing-McGowan-FrameworkForDevelopingMod-MBSE-Open (GDPIS-SUMMIT 2019)

Digital thread: digitalization of the engineering lifecycle

Automation

Requirement identification and generation

Model population

Documentation and compliance

Traceability

Recovery traces

Consistency checking

Management

Models

Integration and exchange

Link logical (descriptive) ←→physical (analytical)

Reuse

Simulation

Configuration

Orchestration

Link

V&V

Quality (CCC)

Information sharing with providers

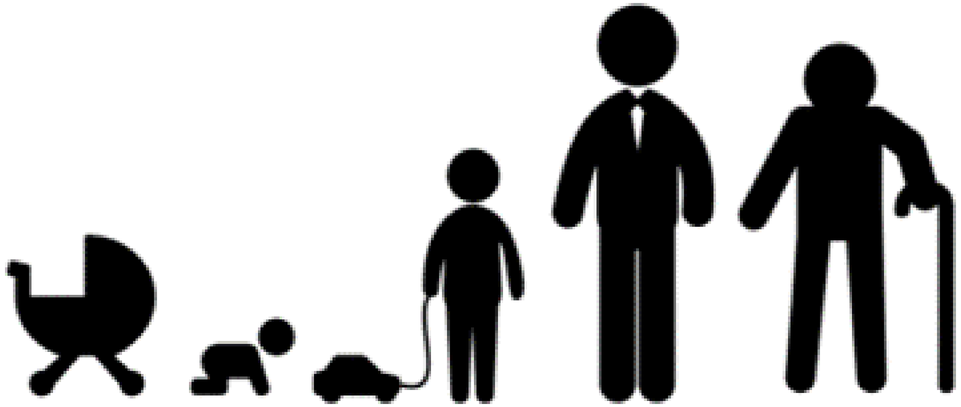
Configuration Management

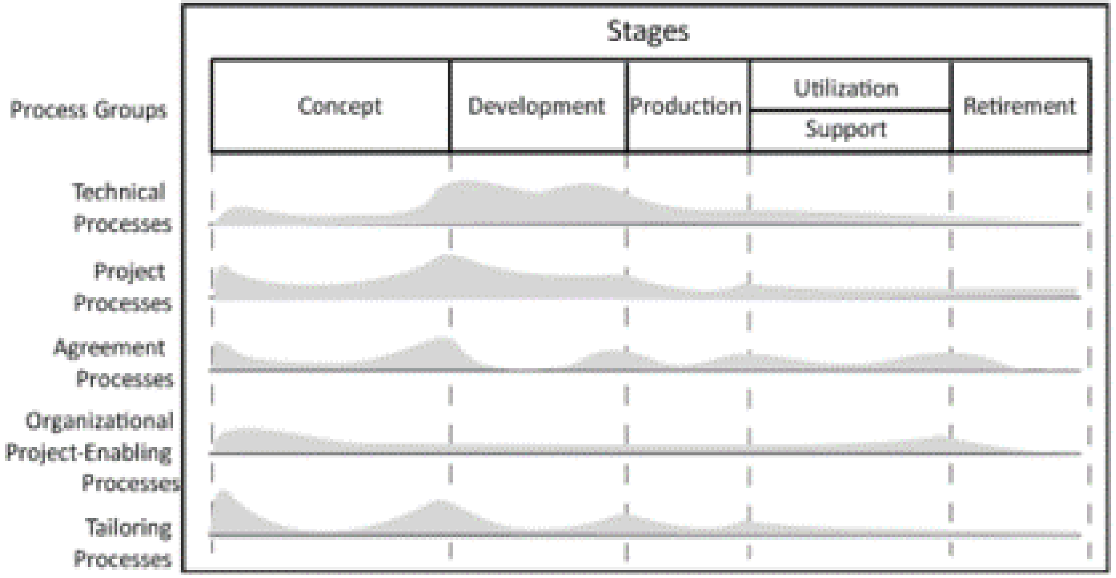
Evolution and information sharing

Digital thread: digitalization of the engineering lifecycle

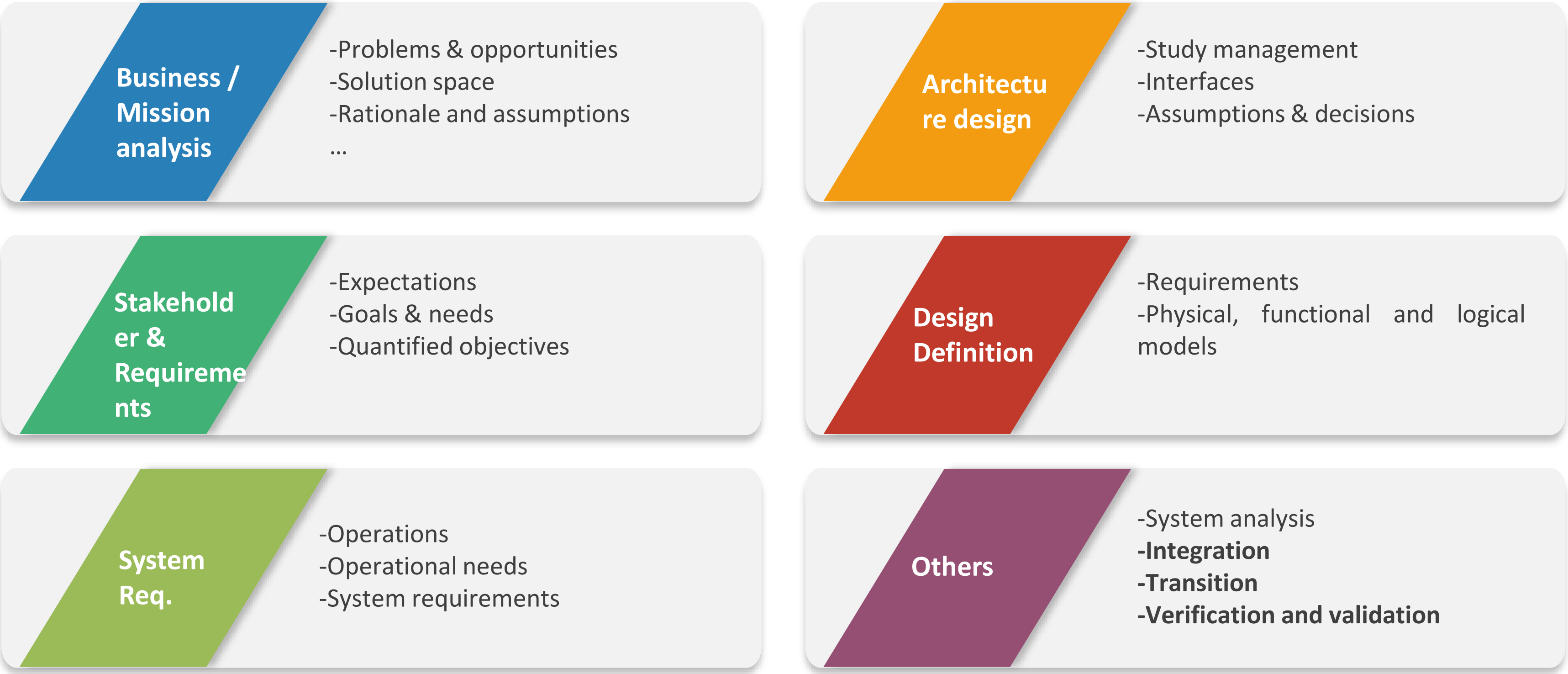
Life cycle

System life cycle





Capturing the collaborative systems engineering context



Business / Mission analysis

-Problems & opportunities
-Solution space
-Rationale and assumptions
…

Architecture design

-Study management
-Interfaces
-Assumptions & decisions

Stakeholder & Requirements

-Expectations
-Goals & needs
-Quantified objectives

Design Definition

-Requirements
-Physical, functional and logical models

System Req.

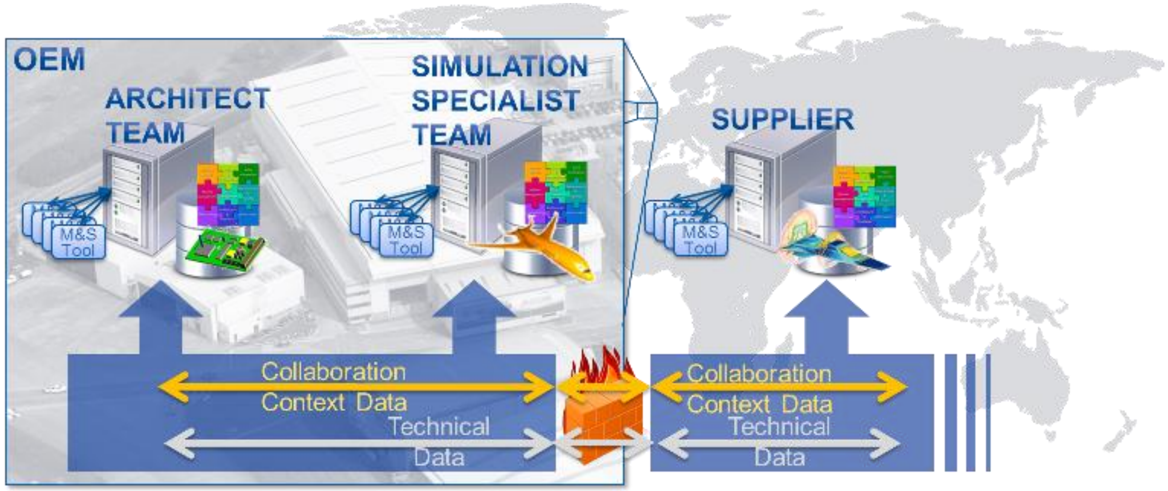
-Operations
-Operational needs
-System requirements

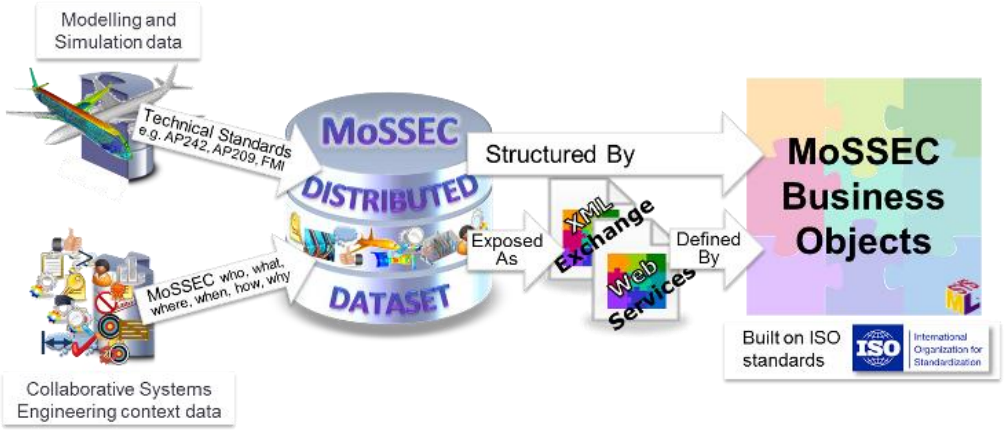
Others

-System analysis
-Integration
-Transition
-Verification and validation

Source: "Traceability across domains, platforms and organizations", White Paper MoSSEC ed1 – v1.0, 2016.

“Modelling and Simulation information in a collaborative Systems Engineering Context” (ISO 10303-243 – MoSSEC)


OEM
ARCHITECT TEAM
SIMULATION SPECIALIST TEAM
SUPPLIER
M&S Tool
M&S Tool
M&S Tool
Collaboration Context Data
Technical Data
Collaboration Context Data
Technical Data


Modelling and Simulation data
Technical Standards e.g. AP242, AP209, FMI
MoSSEC who, what, where, when, how, why
Collaborative Systems Engineering context data
MoSSEC
DISTRIBUTED
DATASET
Structured By
Exposed As
XML Exchange
Web Services
Defined By
MoSSEC Business Objects
Built on ISO standards
ISO
International Organization for Standardization

Similar works

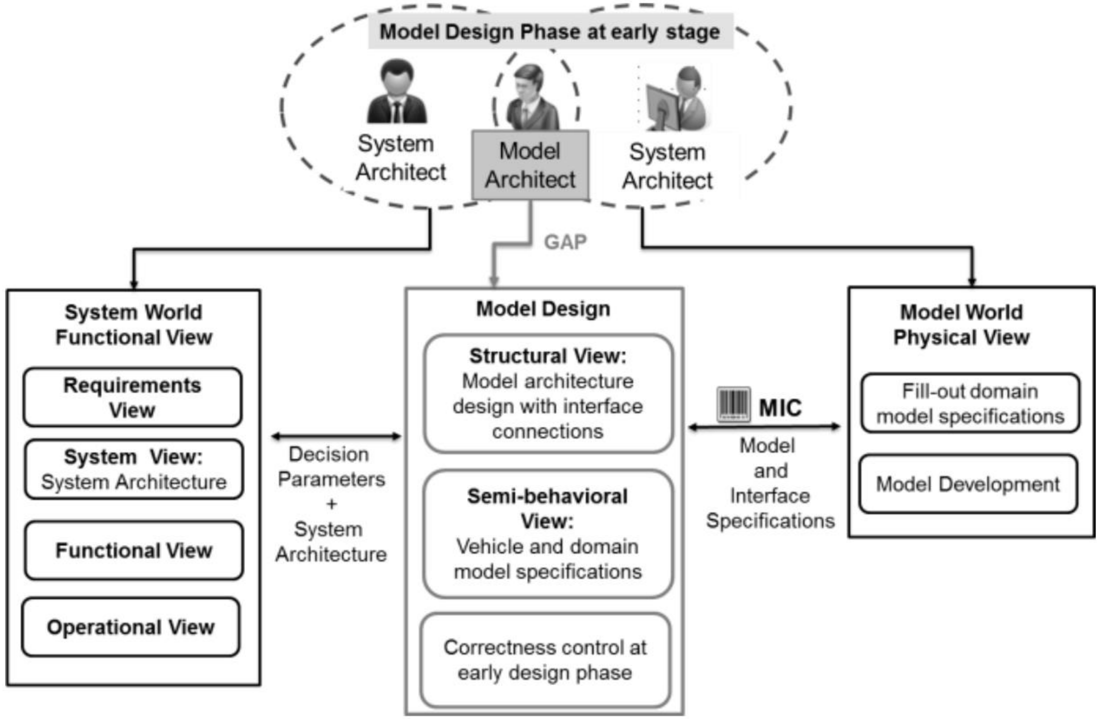

Model Design Phase at early stage
System Architect
Model Architect
System Architect
GAP
System World Functional View
Requirements View
System View: System Architecture
Functional View
Operational View
Decision Parameters + System Architecture
Model Design
Structural View: Model architecture design with interface connections
Semi-behavioral View: Vehicle and domain model specifications
Correctness control at early design phase
MIC
Model and Interface Specifications
Model World Physical View
Fill-out domain model specifications
Model Development

Fig. 1. Research Gap in Collaborative Model Development Process

https://download.afnet.fr/ASD2019/ASD2019-07-MouadhYagoubi.pdf

https://hal.archives-ouvertes.fr/hal-01184938/document

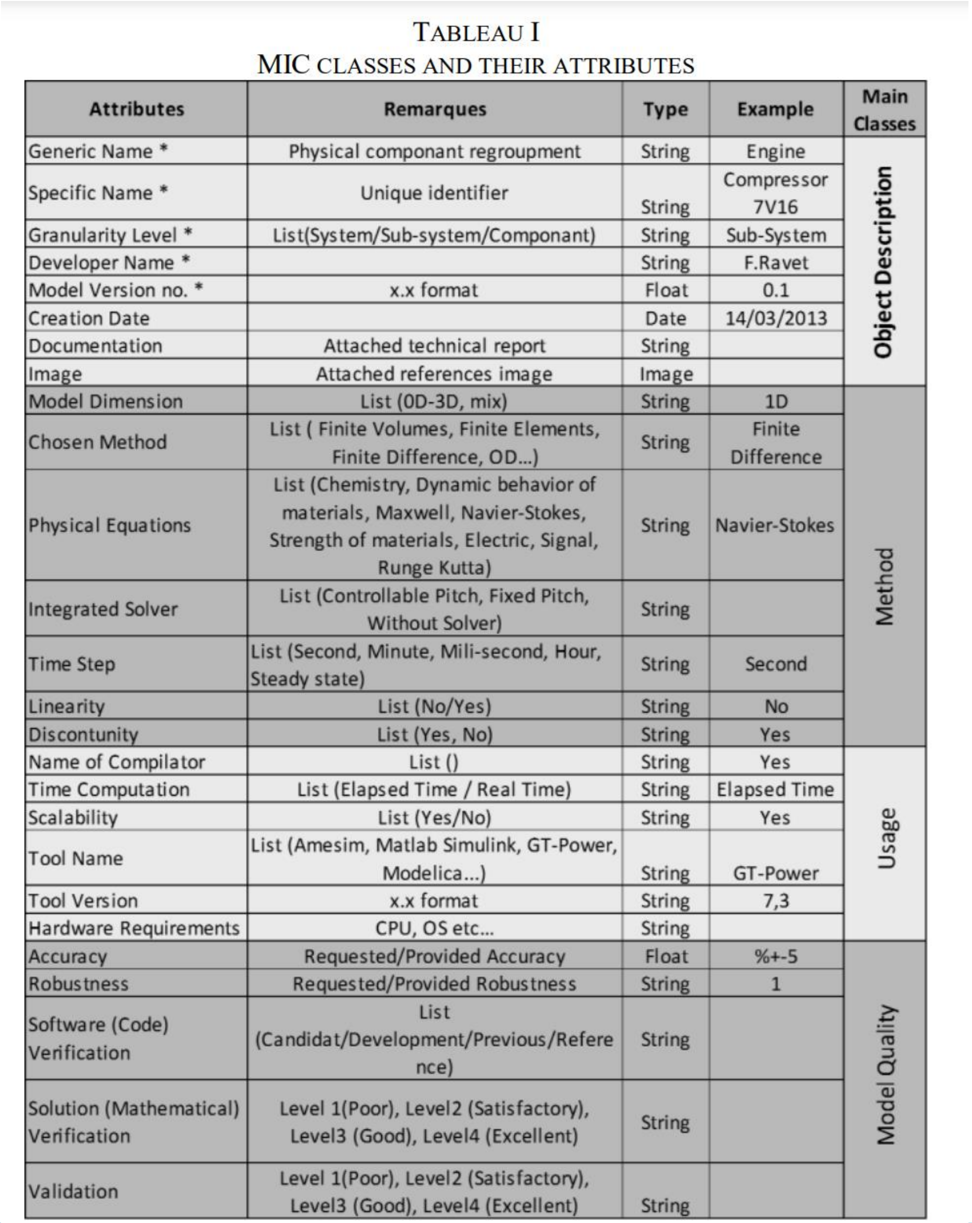
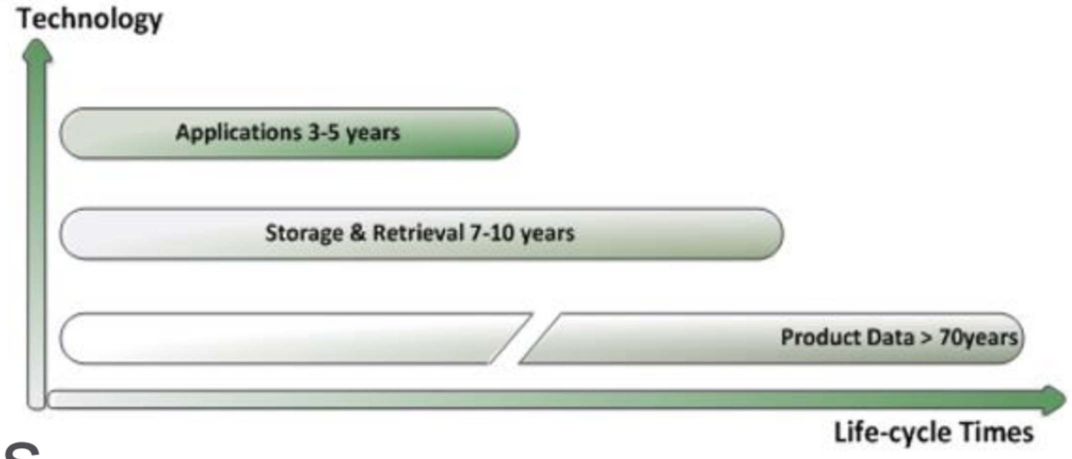


TABLEAU I
MIC CLASSES AND THEIR ATTRIBUTES

Attributes	Remarques	Type	Example	Main Classes
Generic Name *	Physical componant regroupment	String	Engine	Object Description
Specific Name *	Unique identifier	String	Compressor 7V16	
Granularity Level *	List(System/Sub-system/Componant)	String	Sub-System	
Developer Name *		String	F.Ravet	
Model Version no. *	x.x format	Float	0.1	
Creation Date		Date	14/03/2013	
Documentation	Attached technical report	String		
Image	Attached references image	Image		
Model Dimension	List (0D-3D, mix)	String	1D	Method
Chosen Method	List (Finite Volumes, Finite Elements, Finite Difference, OD…)	String	Finite Difference	
Physical Equations	List (Chemistry, Dynamic behavior of materials, Maxwell, Navier-Stokes, Strength of materials, Electric, Signal, Runge Kutta)	String	Navier-Stokes	
Integrated Solver	List (Controllable Pitch, Fixed Pitch, Without Solver)	String		
Time Step	List (Second, Minute, Mili-second, Hour, Steady state)	String	Second	
Linearity	List (No/Yes)	String	No	
Discontunity	List (Yes, No)	String	Yes	
Name of Compilator	List ()	String	Yes	Usage
Time Computation	List (Elapsed Time / Real Time)	String	Elapsed Time	
Scalability	List (Yes/No)	String	Yes	
Tool Name	List (Amesim, Matlab Simulink, GT-Power, Modelica…)	String	GT-Power	
Tool Version	x.x format	String	7,3	
Hardware Requirements	CPU, OS etc…	String		
Accuracy	Requested/Provided Accuracy	Float	%+-5	Model Quality
Robustness	Requested/Provided Robustness	String	1	
Software (Code) Verification	List (Candidat/Development/Previous/Reference)	String		
Solution (Mathematical) Verification	Level 1(Poor), Level2 (Satisfactory), Level3 (Good), Level4 (Excellent)	String		
Validation	Level 1(Poor), Level2 (Satisfactory), Level3 (Good), Level4 (Excellent)	String		

Why LOTAR?

LOng Term Archiving and Retrieval



- LOTAR is Enabled by Standards
- MBSE Justification: Safety, Accident Investigations, Maintenance, Regulations, Obsolescence
- Assume Application versions 3yr; storage/access 10yrs; translate to stable formats for 50yr product cycles.

Source: Mark Williams (Boeing)

Developing LOTAR 5xx standards

The structure of the family, organized in a series of parts, NAS 9300:

- **Part 500**: Fundamentals and Concepts for long term archiving and retrieval of Model-Based Systems Engineering information
- **Part 510**: Long term archiving and retrieval of Requirement management “text, graphics, table based” and “parameter based” information
- **Part 515**: Long term archiving and retrieval of Validation and Verification “text based” and “parameter based” information (expanding Part 515)
- **Part 520**: Long term archiving and retrieval of analytical models described by specification or executable code, containing differential, algebraic and discrete equations
- **Part 530**: Long term archiving and retrieval of models defined using architecture description languages (ADLs), ISO 42010, e.g. industry standards: AADL, SysML, UML etc.

(proposed)

Source: Mark Williams (Boeing)

LOTAR for MBSE View

GOAL: Retrieval using a standard data representation and available tools

WHAT: Basis for Certification/Qualification, After-market Support Customer Services, Field and Accident Investigations, Part Obsolescence, Design Reuse, New Technology Development

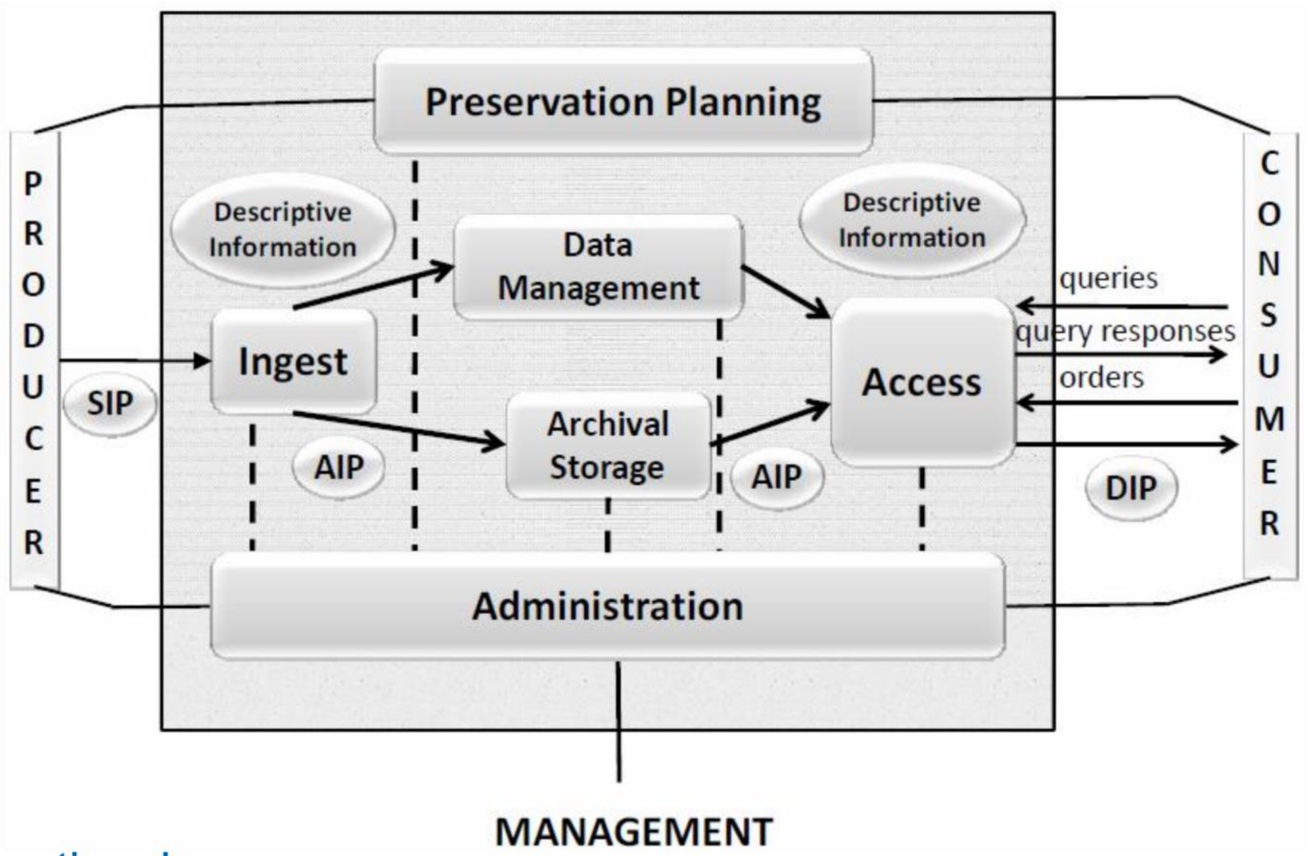
DATA: Systems Architecture not captured on schematics, the LBOM, Requirements traceability, software function

HOW: Capture the Object model, relationships, and metadata

Source: Mark Williams (Boeing)

OPEN ARCHIVAL INFORMATION SYSTEM

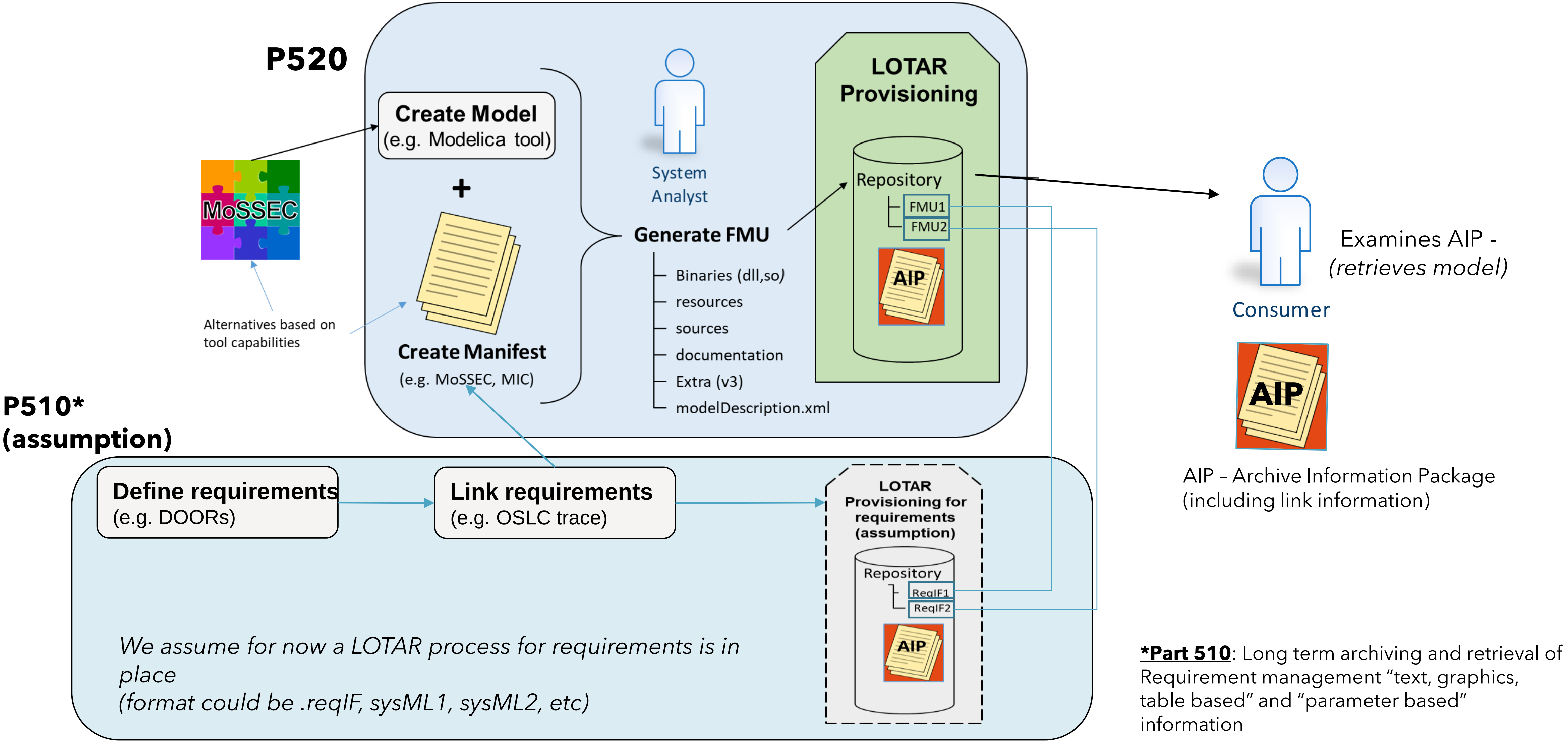
OASIS reference model - ISO 14721



http://www.lotar-international.org

Source: Mark Williams (Boeing)

Case Study: linking requirements and physical models



Examines AIP - *(retrieves model)*

We assume for now a LOTAR process for requirements is in place
(format could be .reqIF, sysML1, sysML2, etc)

AIP – Archive Information Package (including link information)

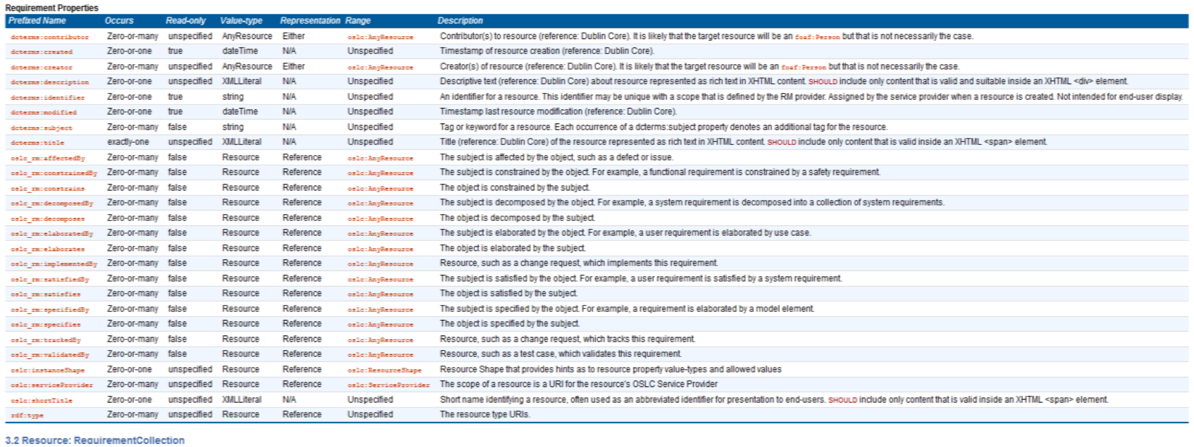
<u>*Part 510</u>: Long term archiving and retrieval of Requirement management "text, graphics, table based" and "parameter based" information

Case Study: linking requirements and physical models

Key Aspect	ReqIF	OSLC RM	SysML Requirement Diagram	FMI/FMU
Communication protocol	File (distribution as a zip file)	HTTP Service	File / HTTP Service in SysMLV2	File (distribution as a zip file)
Syntax	XML	RDF/XML (and other RDF serialization formats)	XML (serialization may change depending on the tool) JSON (SysML V2)	XML and other formats depending on the content of the package
Semantics	XML Schema (depending on the end user)	OSLC RM Shape (common attributes)	SysML metamodel	XML Schema
Identification	Unique global identifiers Managed by the tool	HTTP URIs Managed by the service provider	Unique identifiers Managed by the tool	Unique global identifiers Managed by the tool
Linking	Local/Global	Local/Global	Local/Global	Local/Global
Content	Defined by the tool (e.g. enriched text)	OSLC RM Vocabulary + Defined by the tool (e.g. enriched text)	Defined by the tool (e.g. enriched text)	Defined by the tool (parameters, variables, etc. required for the simulation in the different configuration files: model description, etc.)
Non-functional: maturity, evolution, community, etc.	Mature, Standard, Community			

Case Study: linking requirements and physical models

- Define and use a **core P5XX metadata set**, extracted from P520
- Add OSLC RM as initial subset of extended requirements properties



Requirement Properties

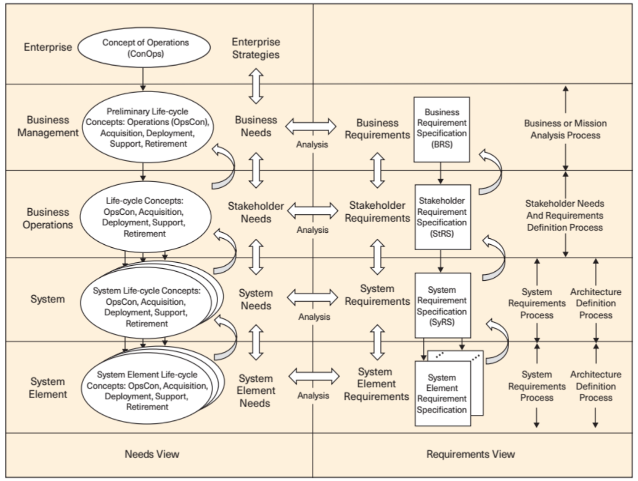
Prefixed Name	*Occurs*	*Read-only*	*Value-type*	*Representation*	*Range*	*Description*
dcterms:contributor	Zero-or-many	unspecified	AnyResource	Either	oslc:AnyResource	Contributor(s) to resource (reference: Dublin Core). It is likely that the target resource will be an foaf:Person but that is not necessarily the case.
dcterms:created	Zero-or-one	true	dateTime	N/A	Unspecified	Timestamp of resource creation (reference: Dublin Core).
dcterms:creator	Zero-or-many	unspecified	AnyResource	Either	oslc:AnyResource	Creator(s) of resource (reference: Dublin Core). It is likely that the target resource will be an foaf:Person but that is not necessarily the case.
dcterms:description	Zero-or-one	unspecified	XMLLiteral	N/A	Unspecified	Descriptive text (reference: Dublin Core) about resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML <div> element.
dcterms:identifier	Zero-or-one	true	string	N/A	Unspecified	An identifier for a resource. This identifier may be unique with a scope that is defined by the RM provider. Assigned by the service provider when a resource is created. Not intended for end-user display.
dcterms:modified	Zero-or-one	true	dateTime	N/A	Unspecified	Timestamp last resource modification (reference: Dublin Core).
dcterms:subject	Zero-or-many	false	string	N/A	Unspecified	Tag or keyword for a resource. Each occurrence of a dcterms:subject property denotes an additional tag for the resource.
dcterms:title	exactly-one	unspecified	XMLLiteral	N/A	Unspecified	Title (reference: Dublin Core) of the resource represented as rich text in XHTML content. SHOULD include only content that is valid inside an XHTML <span> element.
oslc_rm:affectedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is affected by the object, such as a defect or issue.
oslc_rm:constrainedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is constrained by the object. For example, a functional requirement is constrained by a safety requirement.
oslc_rm:constrains	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The object is constrained by the subject.
oslc_rm:decomposedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is decomposed by the object. For example, a system requirement is decomposed into a collection of system requirements.
oslc_rm:decomposes	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The object is decomposed by the subject.
oslc_rm:elaboratedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is elaborated by the object. For example, a user requirement is elaborated by use case.
oslc_rm:elaborates	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The object is elaborated by the subject.
oslc_rm:implementedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	Resource, such as a change request, which implements this requirement.
oslc_rm:satisfiedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is satisfied by the object. For example, a user requirement is satisfied by a system requirement.
oslc_rm:satisfies	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The object is satisfied by the subject.
oslc_rm:specifiedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The subject is specified by the object. For example, a requirement is elaborated by a model element.
oslc_rm:specifies	Zero-or-many	false	Resource	Reference	oslc:AnyResource	The object is specified by the subject.
oslc_rm:trackedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	Resource, such as a change request, which tracks this requirement.
oslc_rm:validatedBy	Zero-or-many	false	Resource	Reference	oslc:AnyResource	Resource, such as a test case, which validates this requirement.
oslc:instanceShape	Zero-or-one	unspecified	Resource	Reference	oslc:ResourceShape	Resource Shape that provides hints as to resource property value-types and allowed values
oslc:serviceProvider	Zero-or-many	unspecified	Resource	Reference	oslc:ServiceProvider	The scope of a resource is a URI for the resource's OSLC Service Provider
oslc:shortTitle	Zero-or-one	unspecified	XMLLiteral	N/A	Unspecified	Short name identifying a resource, often used as an abbreviated identifier for presentation to end-users. SHOULD include only content that is valid inside an XHTML <span> element.
rdf:type	Zero-or-many	unspecified	Resource	Reference	Unspecified	The resource type URIs.

3.2 Resource: RequirementCollection

- Add Requirements quality and requirements V&V metadata based on relevant standards such as ISO/IEC/IEEE 29148

Case Study: linking requirements and physical models

- **Inherit** elements and properties of the previous P5xx (e.g. existing **P520**)
- **Inherit** elements and properties from existing **standards/specifications like OSLC Requirements Management for Requirements Collection**
- Consider **the key elements of the Requirements Engineering Process** (types and properties)



Enterprise
Concept of Operations (ConOps)
Enterprise Strategies
Business Management
Preliminary Life-cycle Concepts: Operations (OpsCon), Acquisition, Deployment, Support, Retirement
Business Needs
Analysis
Business Requirements
Business Requirement Specification (BRS)
Business or Mission Analysis Process
Business Operations
Life-cycle Concepts: OpsCon, Acquisition, Deployment, Support, Retirement
Stakeholder Needs
Analysis
Stakeholder Requirements
Stakeholder Requirement Specification (StRS)
Stakeholder Needs And Requirements Definition Process
System
System Life-cycle Concepts: OpsCon, Acquisition, Deployment, Support, Retirement
System Needs
Analysis
System Requirements
System Requirement Specification (SyRS)
System Requirements Process
Architecture Definition Process
System Element
System Element Life-cycle Concepts: OpsCon, Acquisition, Deployment, Support, Retirement
System Element Needs
Analysis
System Element Requirements
System Element Requirement Specification
System Requirements Process
Architecture Definition Process
Needs View
Requirements View

Figure 1. Transformation of needs into requirements (Ryan, 2013).

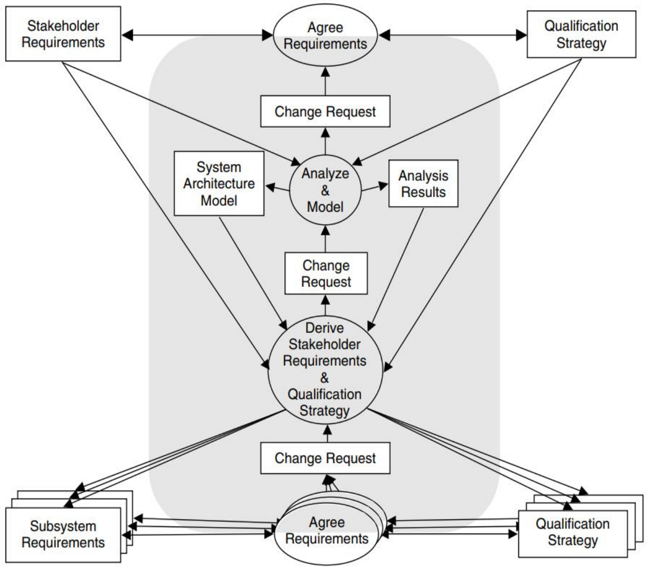
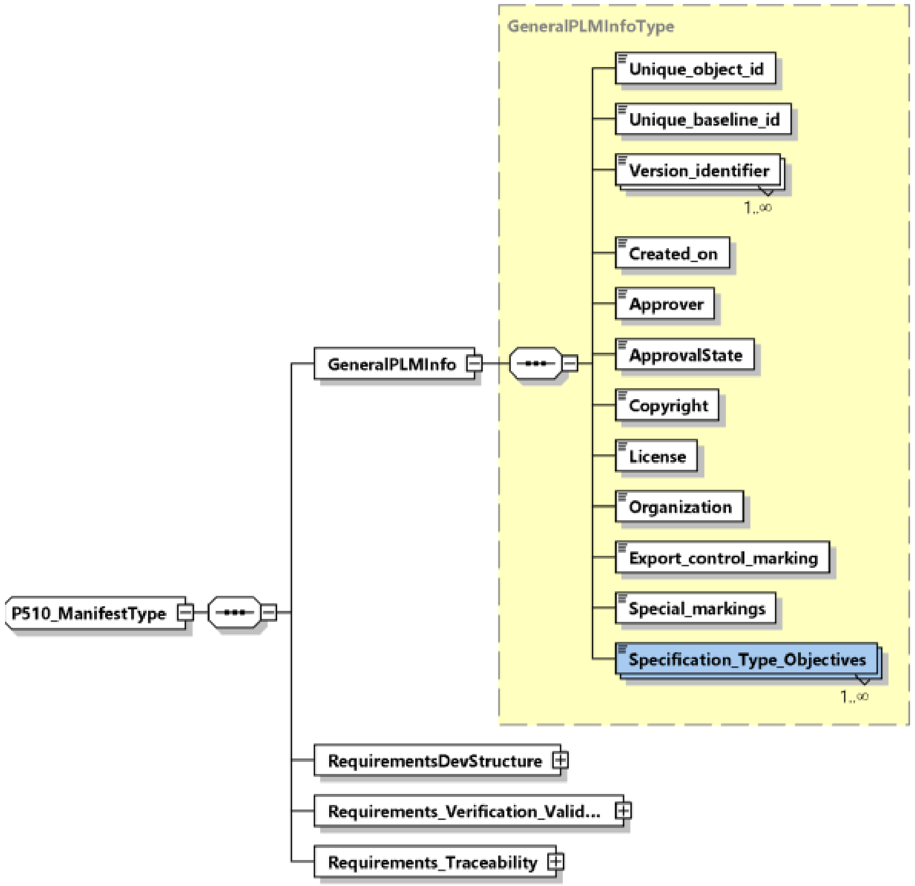


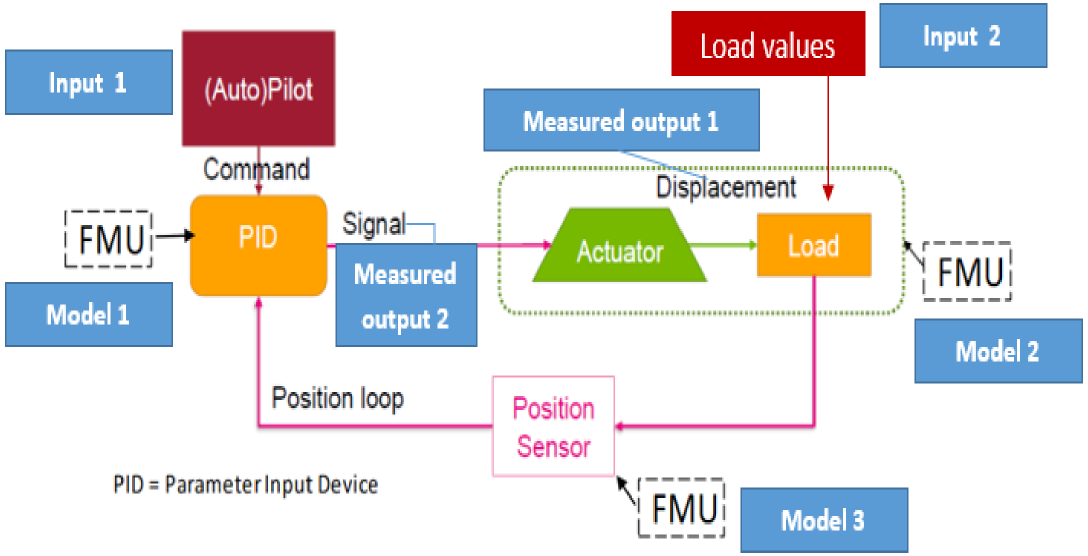
Figure 6.9 Transforming stakeholder requirements directly to subsystems.

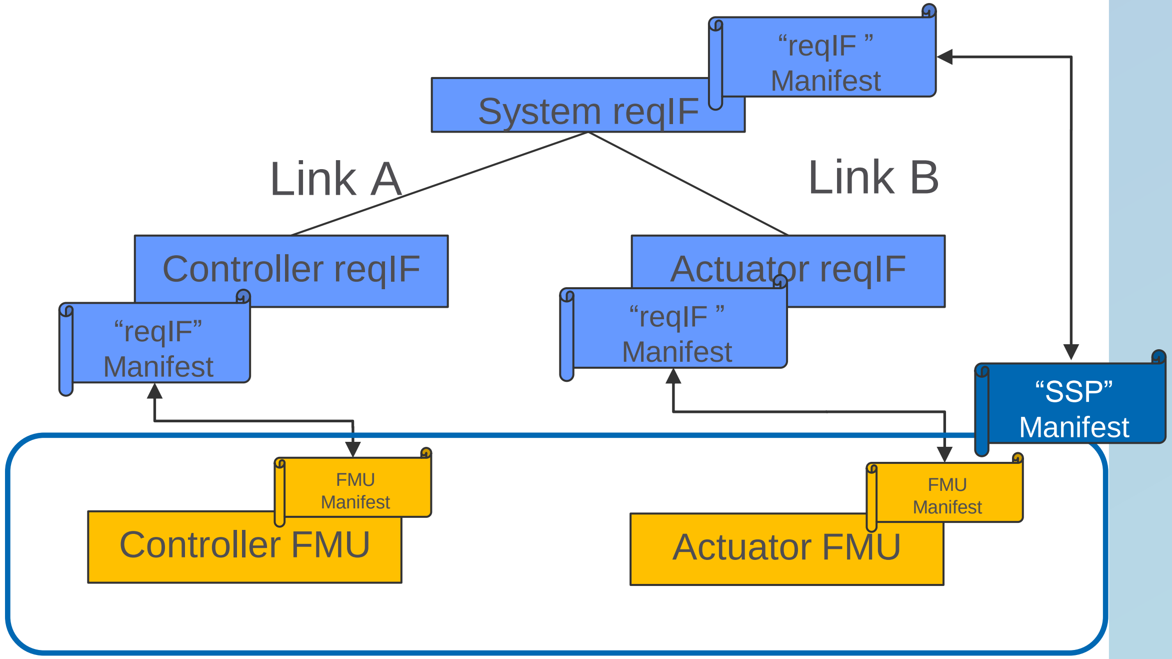
Case Study: linking requirements and physical models (P510 Manifest high-level structure)



Case Study: linking requirements and physical models

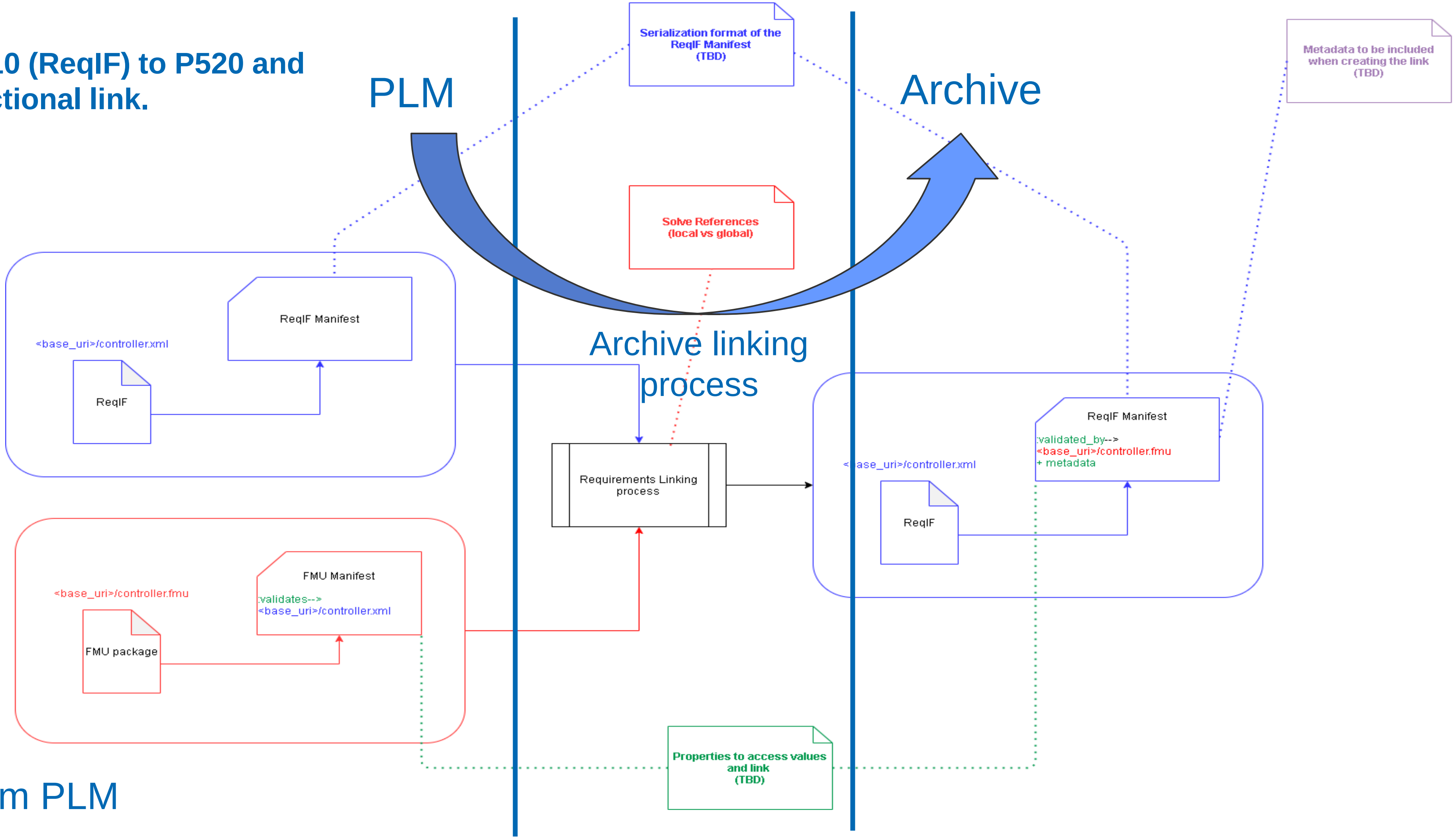
- ✓ Traces created and stored in P510 and P520 manifests, subsystem level.
- ✓ Complete use case with requirements information in reqIF/SysML format.
- Extend to include system level P520 (SSP) and P510 (ReqIF) linking.





Case Study: linking requirements and physical models

Example linking P510 (ReqIF) to P520 and bidirectional link.


PLM
Archive
Serialization format of the ReqIF Manifest (TBD)
Metadata to be included when creating the link (TBD)
Solve References (local vs global)
ReqIF Manifest
<base_uri>/controller.xml
ReqIF
Archive linking process
Requirements Linking process
ReqIF Manifest
:validated_by-->
<base_uri>/controller.fmu
+ metadata
< ase_uri>/controller.xml
ReqIF
FMU Manifest
<base_uri>/controller.fmu
:validates-->
<base_uri>/controller.xml
FMU package
Properties to access values and link (TBD)

Traceability table from PLM

Demo: execution

```
Loading traces file: .\repo_example\PLM\traceability_table.csv
Found: 3 traces.
Loading mappings file: mappings.csv
Found: 1 mappings.
```

Assumption:

Specification and behavioral models exist in the ARCHIVE before linking resolution.

```
PLM Trace: 1-Specification\Component_2.zip, 2-Design\Component_2.fmu, Validated_by
Unpackaging: .\repo_example\ARCHIVE\1-Specification\Component_2.zip
Processing: temp\p510_Component_2.xml
Valid P510 document: True
Trace name:  Validated_by
Existing traces: 0
Saving updated P510 document:  temp\p510_Component_2.xml
Packaging directory:  temp  into file:  Component_2.zip
Copying  temp\Component_2.zip  to  .\repo_example\ARCHIVE\1-Specification
```

Consistency checks

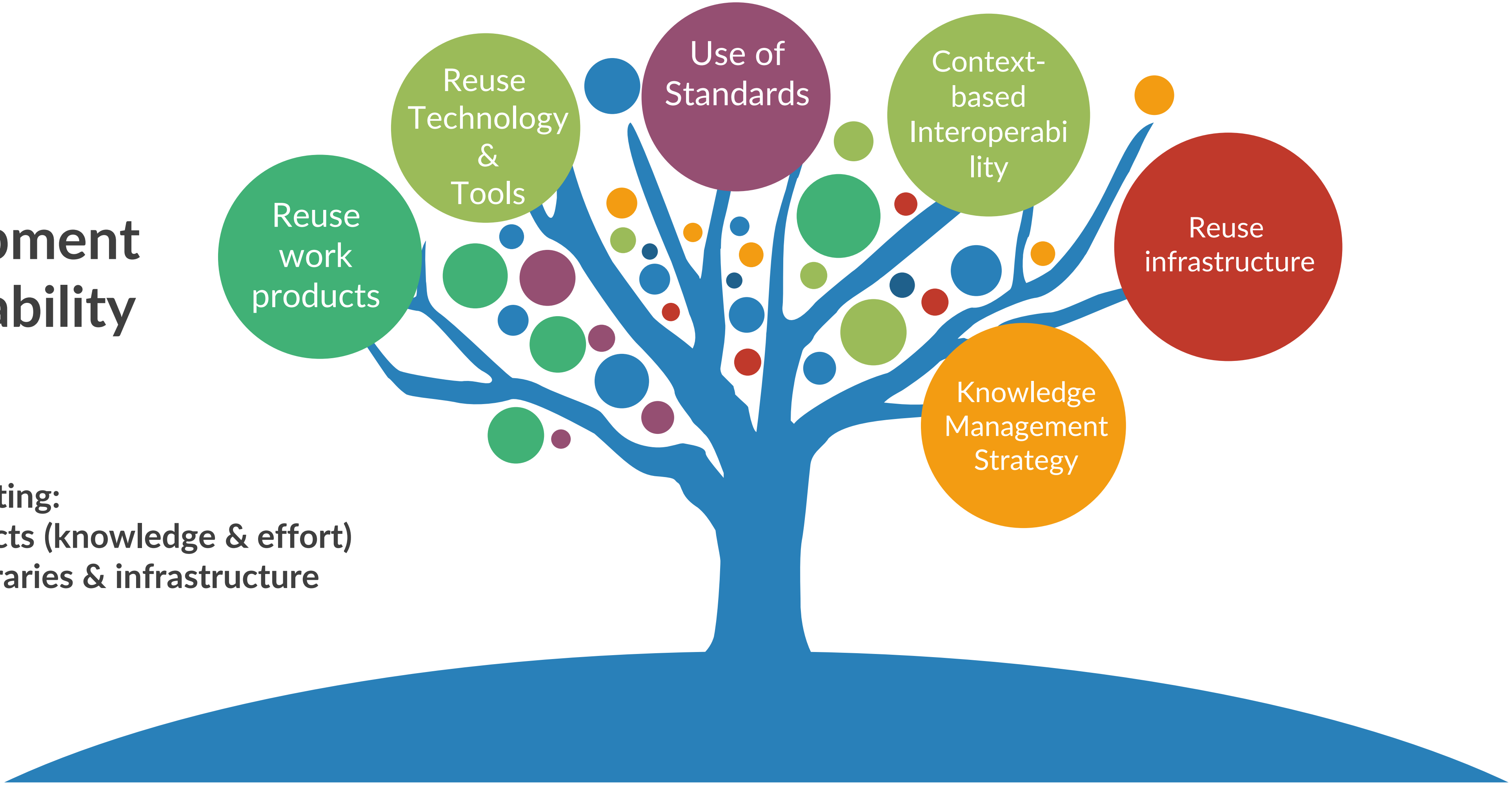
- The specification and model files exist in the archive.
- The specification contains an XML compliant with the P510 Schema.
- The trace name in the traceability table is a valid trace name.

Other potential checks:

- All specifications have, at least, a valid link to a P520.
- All models are linked to some specification? (not necessarily)

Making your engineering lifecycle sustainable

Development Sustainability Tree



Reuse of existing:
-Work products (knowledge & effort)
-Software libraries & infrastructure

Conclusions and further steps

Capture your System lifecycle through interoperable **Manifests**

Use of P5xx **standards (and others)**

Traceability as a first-class member of your lifecycle

Provide traceability mechanisms

Avoid vendor-lock in

Utilities to check P5xx

Future extensions to cover any work product

Family of **standards P5xx under development** (some of them already approved)

Reuse of other existing specifications for OAIS

Make your engineering process and work products properly available in the long term

Take advantage of technologies & tools

Automate tasks and consistency checks

Acknowledgements

LOTAR

LONG TERM ARCHIVING AND RETRIEVAL

The work leading to these results has received funding from the LOTAR standardization group under the statement of work “Prototype Linking Requirements to FMI Package (2021)”.

Learn more: https://lotar-international.org/