Linking Design Requirements to FMUs to create LOTAR compliant MBSE models

Clément Coïc[1] Mark Williams[2] Juan Carlos Mendo[3]
José María Alvarez-Rodríguez[4] Marcus Richardson[3]

[1]Modelon Deutschland GmbH, Germany, clement.coic@modelon.com

[2]PDES, Inc. and INCOSE, USA, mark.williams@incose.net

[3]The Boeing Company, USA, juan.c.mendo@boeing.com, marcus.k.richardson@boeing.com

[4]Department of Computer Science and Engineering, Carlos III University of Madrid, Spain, josemaria.alvarez@uc3m.es

Abstract

LOng Time Archiving and Retrieval (LOTAR) of models is key to using the full capabilities of Model-Based System Engineering (MBSE) in the design lifecycle. LOTAR supports model exchange, reuse and the long-term preservation of data. Archiving also supports the preservation of our valuable product designs and knowledge. Intending to be tool agnostic, LOTAR for MBSE uses data standards from the Modelica Association and other consortia. Therefore, it is important that users of the Modelica based tools and standards understand the LOTAR process to ensure that their data is reusable and suitable for exchange and preservation. The PDES-LOTAR MBSE workgroup is developing a series of EN/NAS 9300 process specifications to standardize the data preservation process, across the aerospace industry. These specifications are customized for each specific MBSE modeling domain, including: models illustrating the logical and functional architectures; models to develop and verify design requirements; models depicting behaviors and simulations; and techniques for linking data across multiple models. For the purpose of creating simulation models, the group recommends using the Modelica language standards, FMI (Functional Mockup Interface), eFMI (FMI for embedded systems), and SSP (System Structure and Parameterization) standards. The LOTAR process requires the integration of industrial data, data standards, and compliant vendor software. For the purpose of this example, the archive process was demonstrated by developing a tool-agnostic Modelica model prototype that was exported into an FMU. The unique features of the Modelica and FMI standards were exposed and greatly enhanced using the Modelon Impact software. The recommendations and workflow proposed in this report were derived from this prototype with the intent of significantly improving future LOTAR implementations and promoting the adoption of these standards across multiple industries.

Keywords: LOTAR, MBSE, MODELICA, FMI

1 Introduction

The issue is simple. The software products that we use to build our models are disposable. They jeopardized the retention and reusability of our knowledge. In an Engineering-Manufacturing environment end-of-life is five years for most software versions, and ten years for backwards compatibility. However, after considering the regulatory requirements, in-service support, future product modifications, and the potential for accident investigations, we must consider reliable access for twenty years, or even up to fifty years for aircraft. This is why LOTAR is important, and why it also promotes and accelerates the industry's efforts toward a digital transformation.

The initiative to develop and qualify international data standards rely on developing prototypes that couple the engineering process with industrial data in a repeatable workflow of procedures. In a first publication (Coïc 2020), the authors presented the LOTAR process and first prototype results. This paper extends that effort and has three goals:

1. To present the LOTAR framework for MBSE models, emphasize the reliance on existing data standards (like Modelica, FMI and SSP), and to establish the need for capturing and exposing model metadata.
2. Highlight the process needed to create the prototype in order to support Peer Reviews and future testing of the standard prior to its release.
3. Use the prototype to demonstrate new/enhanced features, provide evidence for change requests, and expose specification improvements to the individual Modelica Association projects (e.g., Modelica language, FMI, eFMI, and SSP).

The first communication was well received and aligned with investigations executed by other consortia and researchers in a parallel timeline. A notable example is the work organized within the European ITEA3 Call6 project (UPSIM 2020) on the topic of Credible models (Gall 2021

and Otter 2022). One key take-away from this project is the diverse and unique use of Modelica records to host and expose metadata parameters and thus compensate for the unstructured support for metadata in the Modelica language specification. A second example is the acknowledgement (Hällqvist 2023) of how differences in the metadata, derived from different versions of the same model, enables the traceability and integration of features demonstrated by multiple models from different authors developed during the product life-cycle. A similar approach was chosen for the work performed in subsequent LOTAR prototypes expressively for traceability purposes and is presented later in section 5.

This paper differs from our previous communication (Coïc 2020) and attempts to expand the context of MBSE implementations (Nallon 2020). We know we can archive models and we also acknowledge that archiving a model alone does not fully enable its reuse. It is also necessary to archive its dependencies, purpose, sensitivity and predictions. In this paper, the authors present the linking capabilities that enable traceability of the model's purpose, objectives, and relationships to other models and artifacts. For the purpose of this communication, we defined explicit links from the behavior model's elements to their respective requirement specification.

Section 2 provides an overview of the industry value when integrating Modelica Standards with standards from other MBSE domains. Section 3 introduces the in-work deliverables from the PDES-LOTAR MBSE team. Section 4 describes an example linking a single Modelica model with its associated requirements model. Section 5 describes how to capture traceability links holistically in the MBSE Archive Information Package (AIP). Section 0 proposes a potential list of future investigations. Section 7 summarizes the results and recommendations.

2 The Value of MBSE Data Standards

Multiple surveys recently conducted by the ProSTEP IVIP Association across the Automotive and Aerospace Industries indicate, that in the next 5 years the efforts focused on Modeling and Simulation and its use for Product certification, will increase at least 50-100% (ProSTEP IVIP 2022).

With an increase in digitalized development and its application to the certification processes through Model Based Engineering (MBE), the industry will discover new opportunities, while also facing new challenges.

Partially due to the short life-cycle of application software, managing diverse digital designs while enabling future model reuse is a challenge across all industries. Facing this challenge is especially pertinent to the Aerospace Industry with products that have a lifespan longer than 40 years.

Digitalization encourages collaborative model sharing and cross-company co-development with the added challenges posed by Intellectual Property (IP) Protection, Change Management, and Configuration Management.

Standards such as FMI, SSP and ISO STEP AP242 provide good solutions for certain domains in isolation, such as Behavior Modeling and Simulation or 3D modeling. However, when archiving or sharing a system design, that is part of a holistic multidisciplinary product, the industry needs standard solutions to integrate the data across domains. The results must provide traceability from top to bottom and bottom to top. From stakeholder requirements, to the design requirements. From the functional architecture to the installed software products. From behavior analysis and simulations to the hardware implementations and vice versa. This is the priority of the PDES-LOTAR MBSE team (Williams 2023).

3 Archiving Domains and Workflow

The PDES-LOTAR MBSE team focuses on the applicable data standards, the process, design sustainment and design reuse. Under the category of LOTAR for MBSE (LOTAR 2023) the following sub-domain documentation has been proposed:

- **Part 500**: **General MBSE**. The fundamental concepts for long term archiving and retrieval of Model-Based Systems Engineering information.
- **Part 510**: **Requirements Management**. The text, graphics, tables, models, parameters, reference and specification information that defines a product.
- **Part 515**: **Validation and Verification.** The "text based" and "parameter based" information (expands on Part 510), including Use Cases and Model Reports.
- **Part 520**: **Analytical Behaviour Models**. The system or component simulations as described by mathematical specification or executable code, containing differential, algebraic and discrete equations (not FEM).
- **Part 530**: **Architecture Models**. The system functional, logical and structural representations, defined using architecture description languages (ADLs).
- **Part 540:** The **Logical Bill of Materials** (LBOM). The system's physical implementation requirements and dependencies that feed the 3D spatial and manufacturing designs.
- **Part 550**: **Digital or Relational Links**. The traceability and relationship specifications between interrelated models and elements across numerous tools.

Figure 1 below depicts the Archive and Retrieval Process Workflow for MBSE as defined in the LOTAR Part 500 document listed above. The goal is to reuse archived data, if applicable, and create the Authoritative Source of Truth (ASoT).

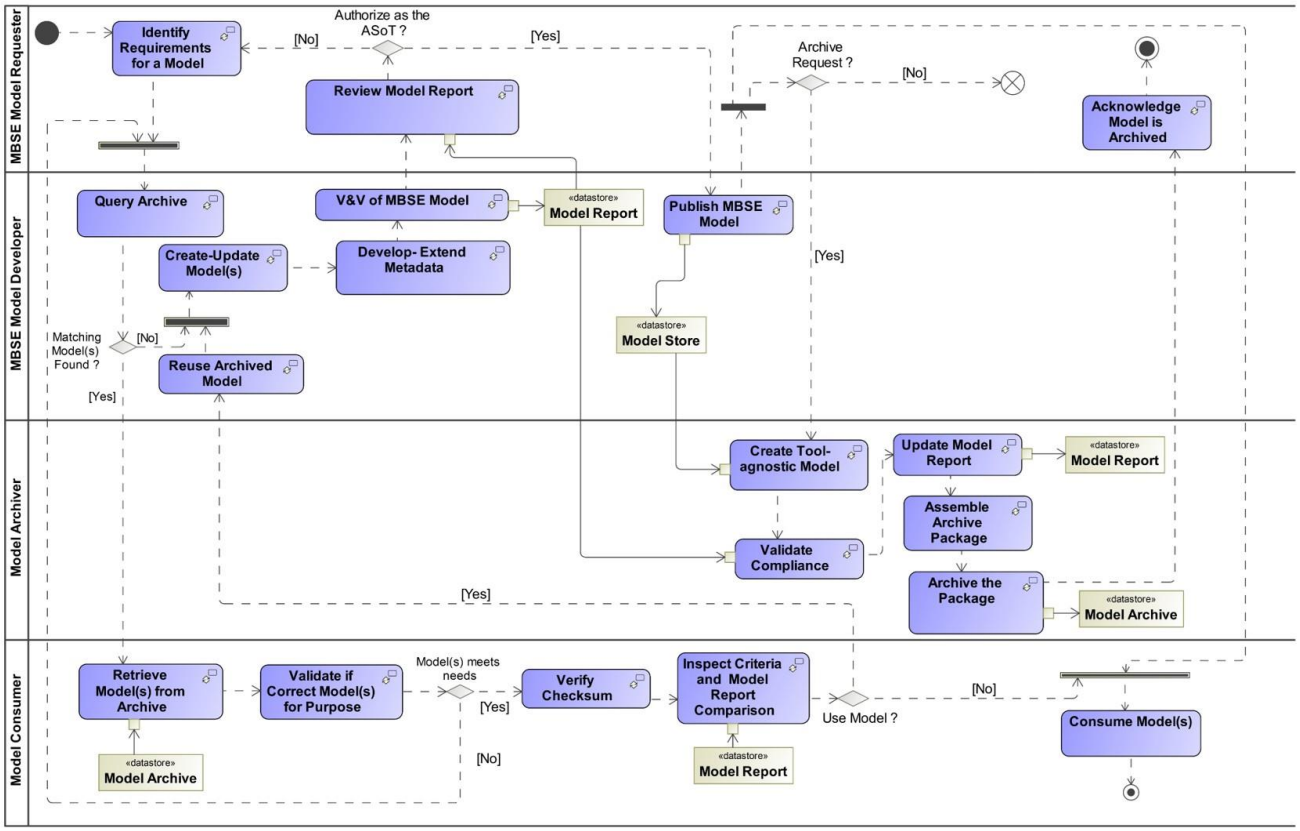

Figure 1. Archive and Retrieval Process Workflow

A pilot implementation of a preliminary workflow for behavior models was described in the first publication (Coïc 2020) and refined in subsequent prototypes producing a generic process for all MBSE models and data.

The "MBSE model" is a generic term used in the workflow and can refer to a model from any of the previously mentioned sub-domains in isolation or, more frequently, to a linked set of cross-domain MBSE models. For the purpose of archiving, models can be assembled in a package, or segregated into different repository packages. Based on guidelines from the OAIS standard (ISO 14721), the model set is referred to as the "MBSE Archive Information Package", AIP, and contains all the artifacts necessary for preserving, retrieving, restoring and disseminating the digital information. This includes the Verification & Validation artifacts of a given system or subsystem, the fixity information relating to security of the archive, and the linking information preserving the relationships and interdependencies of artifacts both internal and external to the package.

The following two sections describe in more detail the prototypes developed. The first example demonstrates how to link a single Modelica model with its associated requirements model, and the second example describes how to manage traceability links holistically at the MBSE AIP package level

4 Prototype: Archiving a Modelica Model with MBSE Links

The previous section introduced the archiving workflow and the architecture of an Archive Information Package. This section will illustrate how each step of the process can be implemented. This presents a solution that is based on open-source packages and, as the wording "prototype" indicates, this is not a unique solution nor a standard for future implementations. However, this proves that the suggested concepts can be implemented by tool vendors and allows us to explore process alternatives and procedural improvements before standards are released.

4.1 Why Start with a Modelica Model?

It seems relevant to remind new users that the LOTAR standard is not tool-dependent when it comes to the model editor. However, the prototypes developed in the scope of this working group should utilize, as much as possible, open standards or open-source packages. Therefore, the starting point of the prototype is a Modelica model. The same common capabilities could be implemented by different tool vendors, including those who rely on proprietary languages or encrypted models. Modelica is not a requirement, it is an open enabler to demonstrate the concept.

4.2 Linking Capabilities in a Modelica Model

Models are developed for a given goal, objective or purpose. Often, models are created to verify that a system design meets its requirements. There might be several models needed to support the verification process. The model might be a testable requirement, describe a functional system architecture, used to keep track of decisions, or validate the selection of a given design technology. The goal of the archived model is to provide a key to information that the model retriever would benefit from accessing in the future. However, it is not enough just to mention or document the model's goal. It is also necessary to link the target model to the actual documents or supporting models that host the information that describes its purpose, boundaries, and the methods used to create it. The model is incomplete without traceability to its underlying requirements. This is the purpose of the Part 550 that is listed above, and also presented in this paper.

The Part 550 does not limit its scope to model goals. It covers any type of link needed to point to the applicable reference information. Indeed, it might be relevant to add traceability links to some of the parameter value details, as recommended in other model authoring references (Gall 2021, Otter 2022).

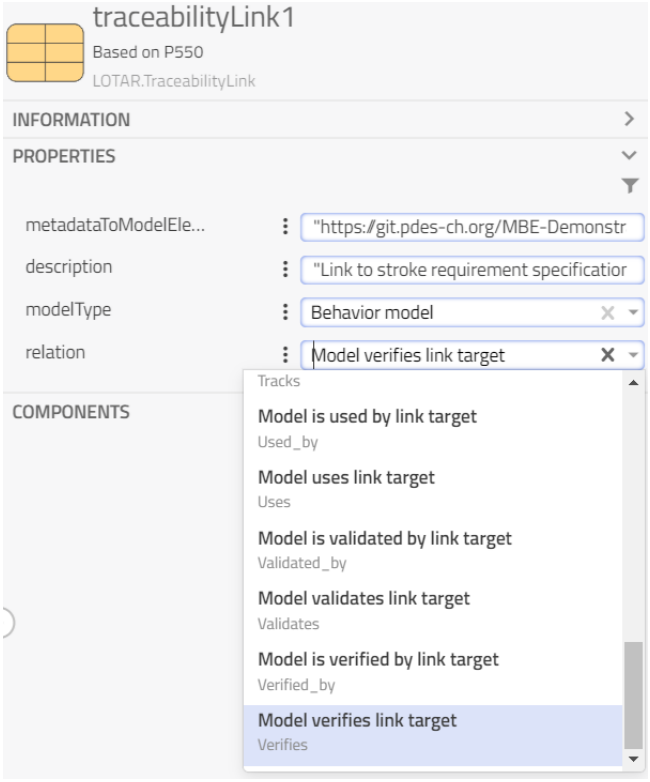

Figure 2 Modelica Linking Record Example

When should the different links be created? The simple answer is that we should not constrain the method of creating the model and we should think from the perspective of the model developer. It might be convenient to create the link at the moment the relationship appears, or when it is created, or when the model construction is characterized as mature. When defining the stroke-length parameter of an actuator, based on a requirement or part specification, establish the link when the parameter value definition is entered. On the other hand, the value may be a variable with a host of possible solutions. Even if there is not a definitive or obvious solution, add a preliminary link in the Modelica model as a place holder, as a reminder to add the parameters later. This is accomplished by defining a Modelica record, as illustrated in Figure 2.

A record is a Modelica class. It can be instantiated, modified or subclassed as many times as desired. Another advantage of this implementation is that LOTAR relies on a Functional Mock-up Unit (FMU) to archive analytical data. The content of the Modelica record is readable in the FMU. This feature could then be used to bring the linking to the top level of the package and expose it to the archiving platform, independent of the hierarchical level in which the record has been instantiated in the Modelica model. **Generate the FMU and Model Manifest**

As previously stated, the LOTAR MBSE model archive relies on FMI and SSP standards. In most analytical tools it is easy to generate an FMU. Modelon Impact enables the functionality by a simple right click on the model and selecting "export FMU".

Once the FMU is available, it is easy to access its "modelDescription.xml" and populate the Part 520 manifest. The optimum approach is for the tool to do this automatically when the FMU is generated. The P520 manifest was presented in detail in our previous paper (Coïc 2020) and will not be repeated here.

At this stage, a convenient additional step is to test the validity of the generated FMU with the FMU Checker (FMU 2023). This ensures that the FMU to be archived satisfies the FMI specification.

4.3 Collecting Links from the FMU

The links are embedded in the Modelica model at different hierarchical levels. This does not represent a convenient way to store the information and enable access when committed to an archive. Instead, it would be preferable to make the dependencies available at the archive level so that the archive platform supports link queries. One example: "find all the models that verify a given requirement". One way to achieve this is to name each Modelica P550 record similarly (e.g. containing the term "traceabilityLink") and perform a regular expression search later on to find these. The common naming can easily be achieved with the "defaultComponentName" annotation for a Modelica class. Provided that the links are defined as Modelica records, and instantiated in a non-protected way (within a restricted class), the parameters of the linked instances can be allocated to the FMU as variables. This is easily accomplished via a simple glob or regular expression, and automated using an external editor or tool specific package (PyFMI 2018). To expand and multiply the integration and traceability options, the Part 550 model links can be easily appended to the Part 520 manifest prior to archiving.

4.4 Generate Model Report

When searching a model archive of FMUs, it is extremely convenient and desirable to include the model's reference results in a specified format. This is part of the retrieval verification process to ensure that the FMU behaves as intended. This is one of the benefits for creating a Model Report. When reusing models after retrieval, it is likely that the model consumer will not have the exact same set-up to run the model than before archiving (e.g. different tool or tool version). The model consumer needs confidence that the model usage is consistent and does not deliver different results. Comparing the simulation output to the Model Report builds confidence to the model's validity. Therefore, the recommended practice is to add a reference comprising the results of the model's execution to the archive package. A simple approach is to run the model with specified tolerances representing the prescribed FMU inputs, and save these in a csv file together with the results from a selection of variable trajectories. The csv file is then linked to the model manifest, together with fixity information (usually a checksum), to validate its state of preservation during the retrieval process. The eFMI approach for generating a Behavioral Model representation of csv reference results (eFMI 2021), is a suitable alternative and the LOTAR MBSE working group started investigating it further.

4.5 Self-criticism of the Archival Prototype

This prototype proved that the archiving workflow is stable when adding links, and that the links can be created at any level of the model hierarchy. It also proved that they can then be exposed at the package level to significantly improve searching the archive.

The prototype results infer additional benefits. Without careful manual preparation, valuable metadata is often missing in the modelManifest.xml of an FMU. The same Modelica record approach could be used for adding this information to the FMU archive and simplify the automation creation of the Part 520 manifest.

In summary, a key next step is to clearly specify what should be included in the Model Report. Only having reference results might not be enough. It could be relevant to point at specific behavior(s) in the time trajectories, or post-processing procedures to compute dependent variables, that could then be linked to other documents or models. This could be the focus of future investigations and prototypes.

5 Prototype: Enable Traceability across a MBSE Archive Information Package

The LOTAR Part 550 process defines the concept of defining MBSE relationships between cross-domain models and model elements. The technology is based on the Open Services for Lifecycle Collaboration (OSLC 2021) data standard and reuses some of the properties defined in the different OSLC vocabularies. Examples include OSLC Requirements Management (RM), and the new Systems Modeling Language (SysML® V2 2022) metamodel. The Part 550 guidelines support the other Part 5XX processes by defining alternative methods to establish and maintain links across MBSE models and then expose those links in the AIP.

5.1 Background

Traceability management becomes a cornerstone activity during the development, archival and reuse of the system models generated during the early product lifecycle. Different modeling and process standards infer the need for traceability, however, the following definition from the ISO/IEC/IEEE 24765:2017 "Systems and Software Engineering Vocabulary" defines the traceability features that will be addressed by the Part 550 process.

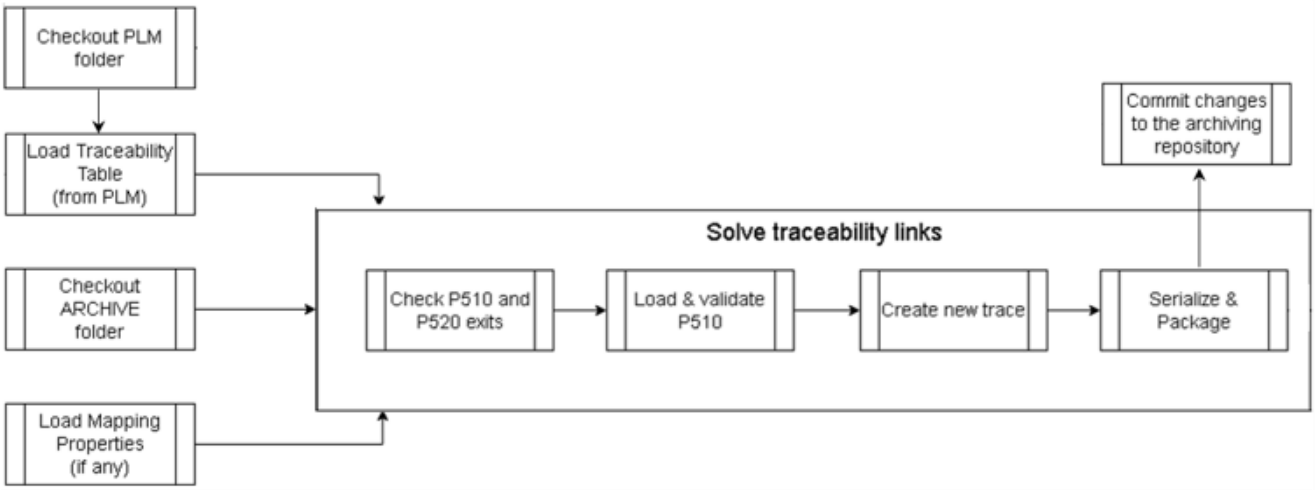

Figure 3 Workflow to enable consistent traceability for a full MBSE AIP

"*Traceability - the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another*"

By applying the following activities to Figure 3, traceability can be utilized throughout the archiving process:

- Creation: In this activity, the engineer identifies the need to create a trace between two system artefacts. A trace, as an entity, shall contain different metadata (unique identifier, timestamp, name, author, source, target, validated, active, etc.). Usually, the creation process shall also consider the approach to define each trace (top-down, bottom-up) and its owner.
- Use/application: Once a trace is created and validated, it is possible to use it for different purposes, e.g., change impact analysis, completeness-maturity analysis (system verification coverage), etc. To do so, it is necessary to provide tools that allow engineers to query or browse the traces and identify both the source and target system artefacts.
- Plan and management: Recognized as a system engineering activity, the process architects must establish how traces are created, defined, identified, validated, recovered, etc. The process shall include technological support and measures assessing the trace quality.
- Maintenance: Projects, products and services are dynamic and change over time. Some parts are often reused across multiple systems, but aligned to different functions and interfaces. Suppliers also evolve their subsystems and components in response to evolving requirements. To properly exploit the information about traces, it is necessary to establish mechanisms to evaluate whether a trace is still active and up-to-date. Traces shall be versioned to properly utilize the linked information between system artefacts. It should be noted that trace maintenance may rely on human validation.

On the other hand, the infrastructure for traceability management (Cleland-Huang et al. 2014) requires different capabilities such as: artifact and trace access to model layers or different repositories, a traceability information model, a trace query layer, a trace generation engine, and some user interaction or reporting mechanism to ensure that the trace was properly created.

From a theoretical point of view, a system traceability function (Alvarez-Rodríguez et al. 2020) can be defined as a function T that for a given resource r_k^i , a target set of resources R_j and a context C will generate a set of mappings $\{(r_k^i, r_k^j, c)\}$ where the input resource and other resource r_k^j will be linked together under a certain value of confidence c as the next equation shows:

$$T: r_k^i \times R_j \times C \rightarrow \{(r_k^i, r_k^j, c)\} / r_k^i \in R_i \wedge r_k^j \in R_j \wedge c \in \mathbb{R} \quad (1)$$

This definition can be generalized and applied to an entity reconciliation process between two different sets of resources, R_i *and* R_j as represented in the next equation:

$$T: R_i \times R_j \times C \rightarrow \{(r_k^i, r_k^j, c)\} / r_k^i \in R_i \wedge r_k^j \in R_j \wedge c \in \mathbb{R} \quad (2)$$

The output of this function will be a set of values $\{(r_k^i, r_k^j, c)\}$ mapped to a confidence level. Where r_k^i represents an element in the source system artifact, and r_k^j represents an element in the target system artifact. Human and tool-based validation techniques shall be used to measure correctness and confidence. In the context of this pilot, we assume that the traceability function already exists and we designed and implemented a process to check the consistency of each link.

5.2 Overview of the Part 550 information model

The Part 550 information model is defined following the principles of the Part 5xx series: 1) a set of types inherited from the Part 5xx series, 2) a set of specific built-in types and relationships for traceability purposes, and 3) extensibility mechanisms to adapt the definitions to specific use cases. The data model definition is represented by an XML schema, as shown in Figure 4, to express both structural constraints and data consistency checks in a parsable format. The data model defines: 1) general PLM info to capture metadata about the tools generating the information; 2) link management metadata depicting configuration management, authorship, ownership, link behaviors, etc.; 3) model link identifiers, and the information contained within each link; and 4) link types and model types that expose information about the source and target artifacts defined by the trace. As a basic example, the next figure partially shows a Part 550 manifest including a trace between two packages.

5.3 Specify a Workflow to enable Consistent Traceability for a full MBSE AIP

Building on the previous definitions, the main goal of the workflow is to enable consistent traces in the MBSE AIP. This relies on addressing the following requirements and utilizing a process that implements the Figure 3 workflow:

```
<?xml version='1.0' encoding='UTF-8'?>
<P550_Manifest>
  <GeneralPLMInfo>
    <Unique_object_id>addcf204-e49c-4a6a-adf5-b2e17fd8a96f</Unique_object_id>
    <Created_on>2022-11-30</Created_on>
  </GeneralPLMInfo>
  <Model_Links>
    <traceabilityLink>
      <metadataToModelElement>tmp\ARCHIVE\1-Specification\Component_2.zip</metadataToModelElement>
      <modelElementToElementLink>tmp\ARCHIVE\2-Design\Component_2.fmu</modelElementToElementLink>
      <linkType>requirementsModel</linkType>
    </traceabilityLink>
  </Model_Links>
</P550_Manifest>
```

Figure 4 A Part 550 Schema example (extract from Notepad++)

1. The system shall process a set of traces designated as an Authoritative Source of Truth (ASoT). The traces can be generated using two methods: 1) inferring the traces from the actual packages that have been archived (as show in the previous section), or 2) taking as reference a set of traces managed and generated by an external tool such as a Product Lifecycle Management-PLM). For both methods, the system shall cross-check for consistency and completeness, and ensure that no trace is missing or out-of-date.
2. The system shall analyze the trace information according to the Part 550 information model and ensure that for any reference, all previously defined traces between two system artifacts actually exists. A trace exists *if* both artifacts are in their designated repositories, are labeled using a unique identifier and contain the property information designating the trace behavior. As shown in Figure 2, the P510 requirement for actuator stroke length, is linked to the P520 actuator model element.
3. The system shall solve logical identifiers/links in alignment with the actual physical resources. The set of traces may contain URLs (Uniform Resource Locators) as identifiers that are then translated into physical paths in a file system.
4. The system shall populate a Part 550 manifest for each verified trace, by gathering information from each artifact's originating model manifest (e.g., Part 510 for requirements models and Part 520 for analytical models).
5. The system shall establish an inventory of all of the consistent traceable system artifacts by generating a new Part 550 AIP containing the up-to-date and consistent trace information.
6. The system shall solve inconsistencies such as duplicate or different trace names for the same type of trace, by assigning name extensions or deleting duplicates.
7. The system shall be capable of gathering trace information from a GIT-based repository.

5.4 Implementation of the workflow

For the prototype, the workflow defined in Figure 3 was implemented using Python. More specifically, the following steps and technologies have been used.

1. Prototype Set-up: Two repositories were created to simulate both a PLM and Archive repositories (GitHub PLM and GitHub Archive). Created as Part 510 and Part 520 compliant packages, the repositories contained the models, the trace information and system artifacts.
2. Step 1: Check-out the repositories in a local repository using the Python library *gitpython*.
3. Step 2: A static trace file is defined as a reference, to ensure the consistency and completeness of the archival process. A table of extended trace names is defined to avoid any identification disambiguation. Both are loaded into the Python environment as the ASoT.
4. The python application identifies the model specific files in the local file system. The AIPs are unpackaged, and the program loads and validates the specific manifests in the form of Part 510 and Part 520 instances. Then it begins analyzing each trace for consistency. If everything is correct, the trace report is saved and added to the original MBSE AIP package.
5. As depicted in Figure 5, all of the verified traces are regenerated in the form of a Part 550 instance and then used to create a new MBSE AIP package. This package is then committed to the archival repository.

5.5 Self-criticism of the Traceability Process

The design and implementation of the prototype to ensure the consistency of traces has demonstrated the feasibility of the Part 510, Part 520 and Part 550 schemas in a MBSE environment. From a conceptual point of view, the Part 550 defines a set of extensible elements and properties to define links with metadata. However, the strategy to select the artifacts and packages that will be part of the MBSE AIP may be externally configured to only include model packages that are relevant for archival purposes. This capability is especially applicable when

sharing information with a third-party. Especially if each traceability link is defined to unify the package contents. Currently, it is possible to define traces at any level in the hierarchy: system, subsystem, component or even a single system element. This would also apply to requirements, requirement modules or requirements at the specification level. From a technical perspective, the set of utilities needed to validate the Part 5xx schemas and build the AIP packages should be delivered as a formalized Python application suitable for any practitioner to use and test against their environments.

6 Future Investigations

The archiving process described in this paper refers to the MBSE Archive Information Package (AIP), which is primarily used to archive and retrieve data. A very similar and complementary MBSE package architecture is used for the purposes of model sharing and synchronous model co-development between different companies. For example, between customer and contractor, or between the OEM and its suppliers.

In this second case, the assembly and sharing of the package happens earlier in the lifecycle of the product and potentially mixes content at different levels of maturity. However, the model content, metadata and relationships between cross-domain are very aligned with the general archiving use case. The package used for collaboration purposes is defined as the Digital-Technical Data Package and multiple association bodies (ProSTEP IViP DDP 2021) (INCOSE DEIX 2020) are investigating how to formalize and standardize the packaging process. The PDES-LOTAR MBSE team intends to leverage these activities to further define a metamodel extending the content and normalizing the package architecture to resemble a LOTAR AIP. Furthermore, it is also expected that the implementer prototypes presented in Section 4 and Section 5 will evolve with the following capabilities:

- Integrating both prototypes into a single Use Case leveraging a Stratoliner Control System Use Case example and following the archiving workflow described in Figure 1.
- Defining and instantiating a Model Report for each example as indicated in Section 4.6.
- Developing a compliance validation capability against the model report.
- Building a prototype MBSE AIP for each example including a holistic view of dictionary links as defined by ProSTEP's Digital Data Package (DDP).
- Extending the use case to use sysMLv2 to define a neutral and archivable version of the functional-logical architecture and the V&V cases.

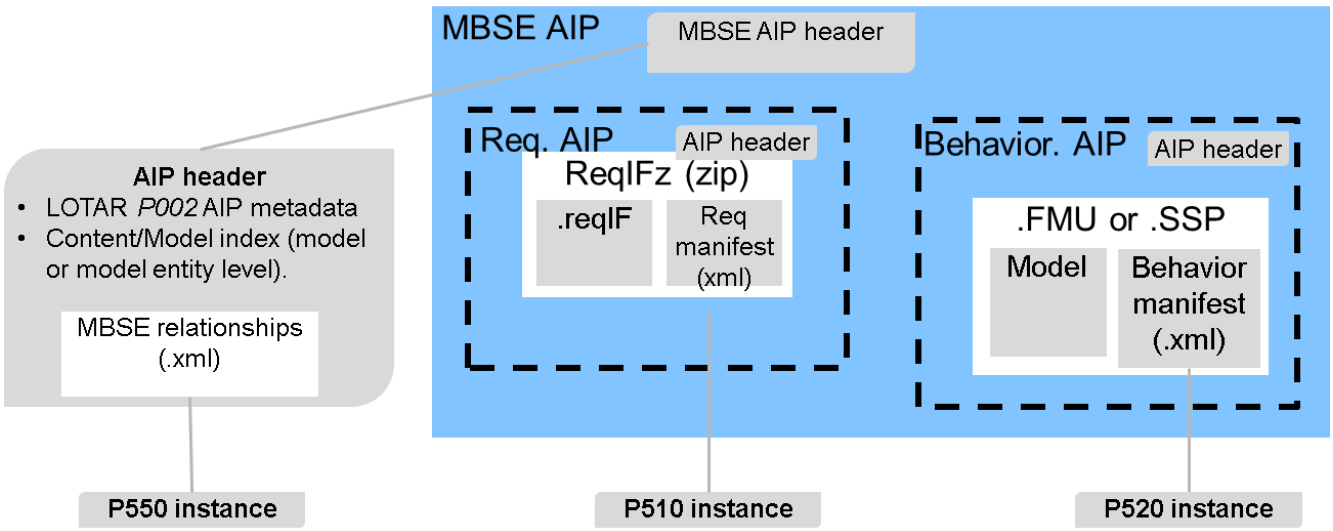

Figure 5 A simplified MBSE AIP package containing a requirement (Part 510) and an analytical (Part 520) model.

7 Conclusion

Standards such as FMI, SSP, ISO STEP AP242 and AP243 provide good solutions for certain engineering domains in isolation, such as Behavior Modeling and Simulation or 3D modeling. However, when archiving or sharing a system design that is part of a holistic multidisciplinary product, the industry needs standard solutions that integrate the data across domains and provide traceability. Part of this responsibility belongs to the consortia that develop the STEP and Modelica standards, absent of the MBSE methods of modeling. LOTAR provides consistent guidelines that define: how to archive the relevant product data for preservation, and how to retrieve, verify and reassemble the archived information. LOTAR's value becomes evident when reusing/migrating design information beyond the life of the software tools. But long-term preservation is needed for accident investigations, when exploring future product support/modifications, and to resolve part/design obsolescence

The Modelica standards needs to better integrate and link with other domains in the MBSE space in order to become a useful asset for the digital certification process. The current standard does not adequately expose or manage the metadata needed to link the model, nor capture the underlying purpose, objectives, methods and risks.

An example was shown where a Modelica model is linked to its associated requirements and how that becomes part of a bigger MBSE package to convey and store the system definition and design. Enabling these connections is a breakthrough that ultimately provides a seamless and bidirectional traceability path or "Digital Thread" to the Verification and Validation Activities and artifacts produced downstream, thus streamlining the certification process. This capability needs to be both part of the standard specification and incorporated into the vendor tools and software products. When combined with web services, the digital thread capability is nearly at hand.

Acknowledgements

The work leading to these results has received funding from the ProSTEP Ivip Association and from PDES, Inc. Data Management resources were provided by the Tools Integration and Model Lifecyle Management Working Group (TIMLM) of INCOSE. The prototype execution was performed by Modelon and the Juan Carlos III University of Madrid. Special acknowledgements to the LOTAR MBSE Working group that has played a critical role defining and validating the prototypes in the weekly meetings and quarterly internal workshops – particularly to Roger Bolton from The Boeing Company, to Gregory Pollari from INCOSE and to Kurt Woodham from NASA Langley Research Center, for their careful review and constructive comments on this paper.

References

Alvarez-Rodríguez, José María, Roy Mendieta, Valentin Moreno, Miguel Sánchez-Puebla, and Juan Lloréns. 2020. "Semantic Recovery of Traceability Links between System Artifacts." Int. J. Softw. Eng. Knowl. Eng. 30 (10): 1415–42. https://doi.org/10.1142/S0218194020400197.

Bouskela, Daniel, Alberto Falcone, et al. (2021). "Formal Requirements Modeling for Cyber-Physical Systems Engineering: an integrated solution based on FORM-L and Modelica". In: Requirements Engineering - accepted for publication.

Bouskela, Daniel and Audrey Jardin (2018). "ETL: A New Temporal Language for the Verification of Cyberphysical Systems". In: *2018 Annual IEEE International Systems Conference (SysCon)*. URL: https://ieeexplore.ieee.org/document/ 8369502

Cleland-Huang, Jane, Orlena C. Z. Gotel, Jane Huffman Hayes, Patrick Mäder, and Andrea Zisman. 2014. "Software Traceability: Trends and Future Directions." *In Future of Software Engineering Proceedings*, 55–69. Hyderabad India: ACM. https://doi.org/10.1145/2593882.2593891.

Clément Coïc, Adrian Murton et al (2020), Modelica, FMI and SSP for LOTAR of Analytical mBSE models: First Implementation and Feedback, *Modelica Conference 2020*. https://doi.org/10.3384/ecp2118149

eFMI Standard, January 2021, v1.0-alpha.4, https://www.efmi-standard.org/media/home/eFMI-Standard-1.0.0-Alpha-4.html

FMU Compliance Checker, https://github.com/modelica-tools/FMUComplianceChecker

Gall, Leo, Martin Otter, Matthias Reiner, Matthias Schäfer, Jakub Tobolár (2021) "View of Continuous Development and Management of Credible Modelica Models", *Proceedings of the 14th International Modelica Conference*, September 20-24, 2021, Linköping, Sweden https://doi.org/10.3384/ecp21181359. European ITEA3 Call6 project UPSIM

GitHub Archive, https://github.com/chemaar/lotar_archive_example

GitHub PLM, https://github.com/chemaar/lotar_plm_example

Hällqvist, Robert (2023) On the Realization of Credible Simulations in Aircraft Development: Efficient and Independent Validation Enabled by Automation, https://doi.org/10.3384/9789179295981

ISO 14721, (2012) Space data and information transfer systems - Open archival information system (OAIS), https://www.iso.org/obp/ui#iso:std:iso:14721:ed-2:v1:en

ISO/IEC/IEEE 24765, (2017), "Systems and Software Engineering – Vocabulary",

https://www.iso.org/obp/ui#iso:std:iso-iec-ieee:24765:ed-2:v1:en
Nallon J., Williams M., (2020). “MBSE Tools Database Update, and Integrate Models with Tools”. Presentation: INCOSE International Workshop, TIMLM Working Group, Torrance, CA, USA. URL: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2020:iw2020_timlm_mbseworkshop.pdf
Otter, Martin, Matthias Reiner, Jakub Tobolár, Leo Gall, Matthias Schäfer, (2022), “Towards Modelica Models with Credibility Information”. Electronics 2022, 11, 2728. https://doi.org/10.3390/electronics11172728
OMG Systems Modeling Language (SysML®) V2 (2022). Standard. Object Management Group-OMG. https://github.com/Systems-Modeling/SysML-v2-Release.
OSLC, Open Services for Lifecycle Collaboration, (2021), an API data linking standard developed by the Organization for the Advancement of Structured Information Standards (OAIS) consortium, https://open-services.net/specifications/
PDES, Inc., Product Data Exchange Standards, https://pdesinc.org/.
ProSTEP ivip Association (2022), Smart Systems Engineering Collaborative Simulation-Based Engineering Version 3.0, https://www.prostep.org/fileadmin/downloads/PSI_11_V3_SmartSE_Rec_and_Part_A-I.zip
PyFMI, https://pypi.org/project/PyFMI/
SysML V2, Systems Modeling Language (2022), data standard developed by the Object Management Group (OMG), http://www.omgsysml.org/.
UPSIM, Unleash Potentials in Simulation, (2020-2023), ITEA4 project, https://itea4.org/project/upsim.html
Williams, M (2023), TIMLM-PDES-LOTAR 101, INCOSE presentation, MBSE, Projects Overview and Team History, INCOSE https://incose2.sharepoint.com/