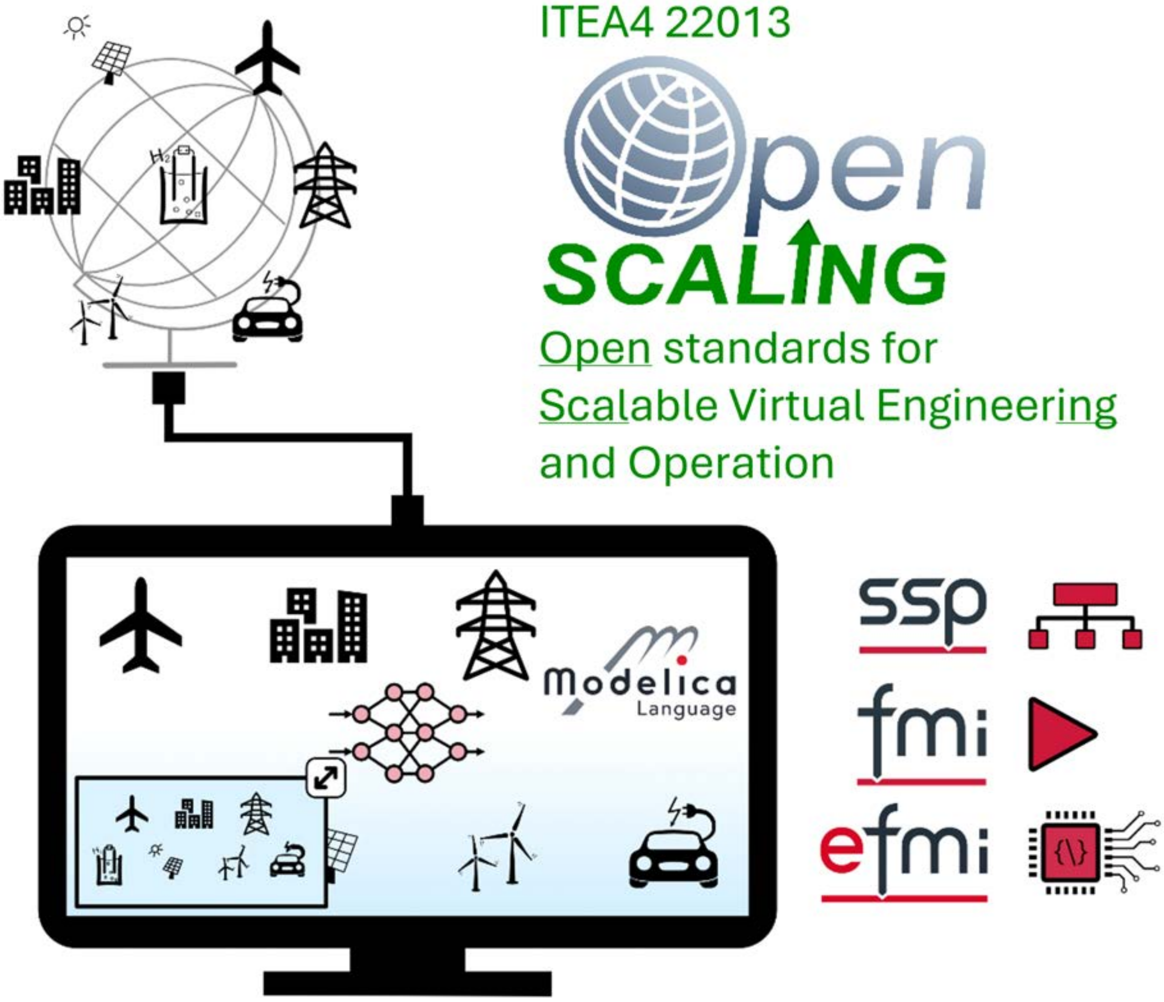


# Deliverable 5.1

## *Requirement specification on Large Scale System (LSS) workflows and their enablers*

Robert Hällqvist

Version 1.0, 2025 09 04

# Table of Contents

# Project Acronyms

| Acronym | Description |
|---|---|
| AI | Artificial Intelligence |
| COP | Coefficient of Performance |
| CPS | Cyber Physical System |
| CSP | Credible Simulation Process |
| DAE | Differential-algebraic equation |
| eFMI | Functional Mock-up Interface for embedded systems |
| FMI | Functional Mock-up Interface (standard for model exchange and co-simulation) |
| FMI component | A model in FMI format (= FMU) |
| FMU | Functional Mock-up Unit (= an FMI component) |
| HVAC | Heating, Ventilation and Air Conditioning |
| LOTAR | Long Term Archiving and Retrieval |
| LSS | Large-scale System |
| M&S | Modelling and Simulation |
| Modelica | Standard for modelling of cyber-physical systems |
| MoSSEC | Modelling and simulation information in a collaborative system engineering context |
| MiL/SiL/HiL | Model/Software/Hardware in the Loop |
| NeuralODE | Ordinary Differential Equation, where a Neural Network (NN) defines its derivative function |
| NN | Neural Network |
| ODE | Ordinary Differential Equation |
| PeN-ODE | Physics enhanced Neural ODE |
| ONNX | Open Neural Network Exchange standard |
| PINN | Physics-informed Neural Network |
| SSP | System Structure and Parameterization (standard for connected FMUs) |
| UQ | Uncertainty Quantification |

# From Full Project Proposal

| ID | Type | Description | Due Month | Access |
|---|---|---|---|---|
| D5.1 | Doc | Requirement Specification | M18 | Public |

# 1. Abstract

This deliverable captures the requirements derived from analyzing the industrial workflows, acting as frame of references, of Work Package 5 in the OpenSCALING project. By aligning with both current practices and anticipated developments in model-based engineering, it ensures that the technological advancements are grounded in practical and industry-relevant needs. The resulting requirements, including chosen data formats, serve as a foundation for guiding tool, and workflow, implementations and standardization efforts across the project. The OpenSCALING project relies on the hypothesis that the best-suited engineering tool is to be used for each engineering task throughout all life-cycle phases. Furthermore, the deliverable can be utilized also after project finalization to support long term communication between end users and tool vendors.

# 2. Introduction and Background

Work Package 5 of OpenSCALING aims to serve as a bridge between the large-scale systems workflows deployed in industry and the technological advancements of the project as a whole. The current and planned future industrial processes for model-based development serve as a frame of reference for the work package. Furthermore, WP5 is conducted in close collaboration with all other project work packages but with a particular focus on WP2-Uncertainty Quantification. WP2 aims to adress packaging of information related to the numerical aspects of M&S credibility assessment whereas WP5 adressess the documentation of processess and activities rendering simulation packages that can be traced throughout their life-cycles. This frame of reference is intended to help ensure the industrial relevance of the research in the dependent work packages. This work package will also provide an interface to standardization bodies not explicitly part of the project, such as OMG and prostep SmartSE. Additionally, three of the primary standards emphasized throughout the project as a whole—Modelica, FMI, and SSP—also serve as cornerstones of WP5.

## 2.1. Contributing stakeholders

The following organizations (PoC) have contriuted to formulating the deliverable in focus. Saab Aeronautics (Robert Hällqvist), Bosch (Hans-Martin Heinkel), Dassault Systèmes (Dag Brück), Linköping University (Lena Buffoni), eXXellent Solutions (Peter Lobner), The Swedish National Road and Transport Research Institute (Maytheewat Aramrattana), Volvo AB (Henrik Lönn), Parker-Hannifin (Viktor Larsson), InQueryLabs (Geza Kulcsar), ABB (Rudiger Franke).

## 2.2. Glossary

This deliverable relies on some central terms. These terms are briefly summarised and explained in the list below.

- A *Large Scale System (LSS)* is a system that incorporates multiple systems or sub-systems, originating in multiple different engineering domains, that together produce emergent behaviour (behaviour that cannot be understood by analysing the LSS individual parts separatly)
- A *system architecture* model captures the architecture of the system to be developed (hardware or software). The system architecture (components, connectors, public parameters and connections between connectors) can be used as a starting point (in terms of topology and interfaces) when developing a corresponding analysis architecture to be used for simulation purposes.
- *Analysis architectures* represent the architecture of the model(s) representing selected aspects of the system architecture of the physical system in focus. Separating between analysis and system architectures sets MBSE activities free of any topological constraints that otherwise could render less efficient modeling and simulation processess. The analysis architecture can represent a subset of the system architecture, several alternative system architectures, and potentially include elements or aspects not part of the system architecture per se.
- An *analysis model* is any model utilized in an analysis architecture in order to design and evaluate the physical system of interest or any of its support systems. In the context of this document, an analysis model is typically an executable simulation model.
- The *analysis model interface* is the interface of modeled components or sub-systems (analysis models) included in an analysis architecture.
- An *Operational Domain (OD)* encompassess the feasible input space of an analysis model or a corresponding component. An Operational Domain can be seen as a requirement on, for example, the analysis model or a consequence of the analysis model implementation. It is meaningful, in many cases, to separate between OD of the analysis model vs. OD of the SoI/system component. E.g. maybe the SoI is designed to operate between -40 to +70C but the analysis model only runs between 0 to +70C.
- A *Domain of Validation (DoV)* encompasses any conducted validation experiments, along with their coordinates, within the, for example, analysis model OD.
- A *Domain of Uncertainty Quantification (DoUQ)* encompasses any conducted uncertainty quantification and propagation experiments, along with their coordinates, within the, for example, analysis model OD.

- A *Domain of Sensitivity Analysis (DoSA)* encompasses any conducted sensitivity analysis experiments, along with their coordinates, within the, for example, analysis model OD.

## 2.3. Solution approach, analysis and challenges

Todays' SotA LSS development processes are snap shot oriented and lack standardized means to ensure traceability between different development phases, artifacts, engineering domains. This leads to e.g. i) communication difficulties within and between development organizations, ii) design errors, iii) difficulties relying on model-based desision support in the design process and in type certification of products, iv) low efficiency and high development cost. WP5 proposes means to mitigate traceability aspects of credibility in model-based development of LSS.

Figure 1 shows the main activities of OpenSCALING, as well as the workflows and transfer packages that connect these workflows. The activities are distributed throughout three highlighted categories: virtual protyping, virtual testing, and digital twin utilization. These three catogories exemplify Model-Based Engineering (MBE) activities throughout life cycle phases ranging from conceptual design to operation and maintenance [10]. These structures and workflows are also typical of development processes in industry. The results of WP5 are therefore also directly suitable for industrial implementation. In the individual work packages or subtasks of the development processess , specific methods, tools, and data formats are used. In order to achieve the consistency and traceability required for collaboration, the data formats and metadata must be coordinated and standardized, to a certain extent, for the exchange of information between the subtasks. The extent of which standardization is needed is to be adressed in WP5 with input from WP2 and WP1. Such metadata can be semantically integrated in the simulation packages via the introduced term *GlueParticle*. A GlueParticle, as described in the SSP Traceability specification, is "is a concept for bundling and file-based transfer of process-relevant information and resources". Furthermore, a GlueParticle file is a "is a GlueParticle that is in a file-based representation and can therefore be transferred between tools or organizational units such as people, departments, or companies" [6]. There are currently two different GlueParticle files specified in the SSP Traceability Specification: the Decision Task Meta Data (DTMD) and the Simulation Task Meta Data (STMD) files. Both these files are intended to document the result and status of relevant process activites. In addition also a pre-version of the Modeling Task Meta Data (MTMD) is used. The GlueParticle approach developed in prostep SmartSE was used and further developed for this purpose in OpenSCALING (see Figure 2).

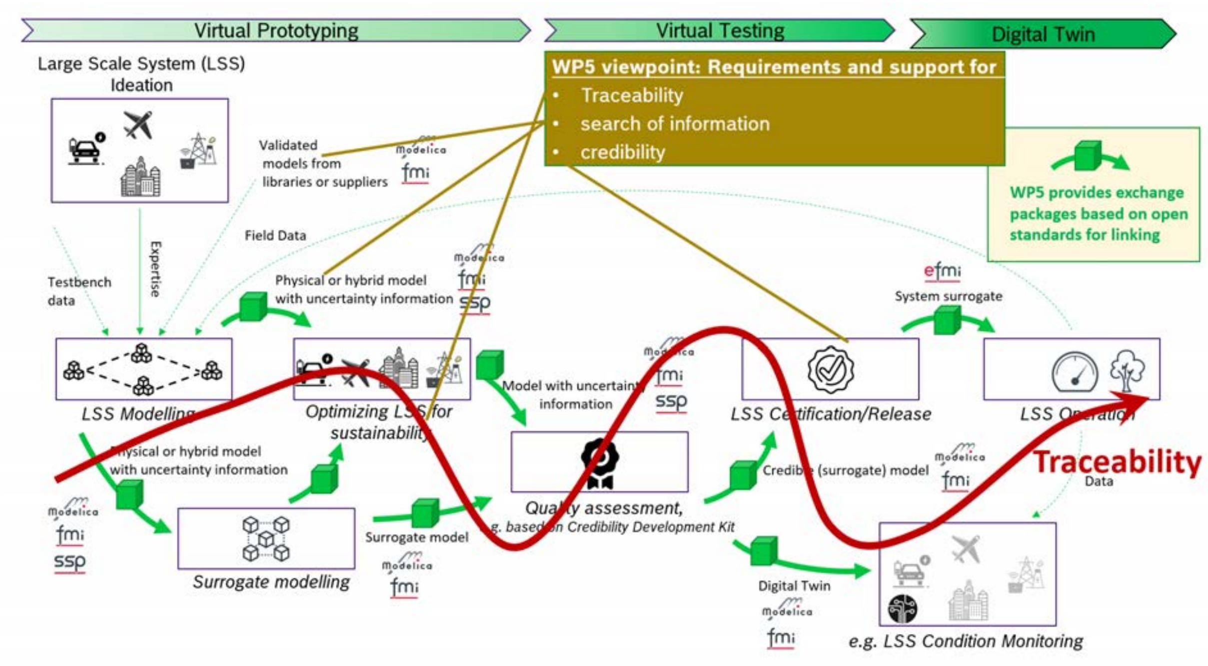


*Figure 1. Viewpoint WP 5 for main activities of OpenSCALING. Main activities of OpenSCALING which are connected by the GlueParticle Exchange Packages for traceability*

**The GlueParticle approach consists of four building blocks that must be aligned with each other**

- Processes
  - Structuring assignment of responsibilities as a basis for the content of the exchange packages. The Credible Simulation Process framework, further detailed and applied through the REDO methodology (see Figure 12), serve as foundation to the GlueParticle approach in focus within the presented work. The Credible Simulation Process Framework specifies a set of high level processess that are deduced as relevant and needed, to a rigour specified by the purpose of the Model-Based Engineering activity, to render credible model-based decision making.
- Methods
  - The methods used specifically in the work packages of the development processess. This includes, for example, methods to gather, aggregate, and present meta data capturing analysis model numerical credibility.
- Information
  - Harmonization of metadata, semantics, interfaces. This document is focused on meta-data content aligned with the MIC-core specifiaction [11], along with information concerning numerical aspects of model credibility assessment as described by Otter et al. [13] and Rosenlund et al. [12]

- Data formats
  - Formats to fascilitate working with heterogeneous IT infrastructures, in a intra and cross orgazational cooparative setting, need to be adopted.

The standards for metadata, data formats and processes listed below are aligned. They also correspond to the preferred formats in the Data Formats chapter. The building blocks are visualized, with exemplified process and standards, in Figure 2.

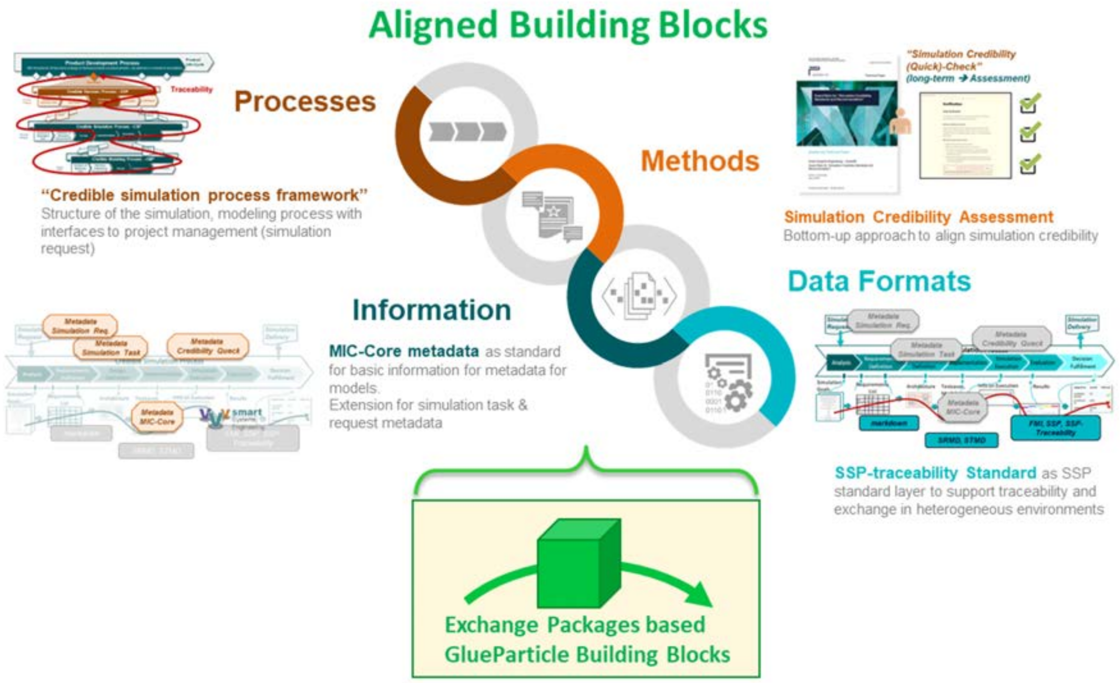


*Figure 2. GlueParticle building blocks which are instanciated in the GlueParticle Exchange Packages*

### 2.3.1. Procedure in engineering or simulation process using the Credible Simulation Process (CSP)

The typical procedure in a development or simulation process is shown in Figure 3. Many recursions take place in such a process. The Credible Simulation Process (CSP) [5] therefore does not represent the chronological sequence but the logical order of the sequence. This is required for traceability. The CSP is therefore a documentation schema and not a chronological process description.

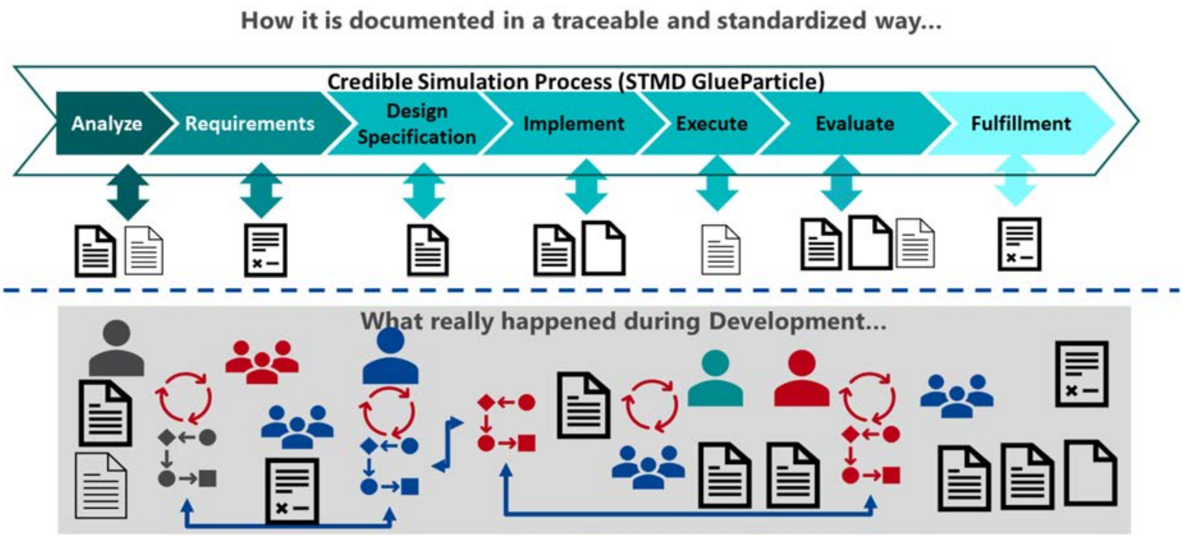


*Figure 3. The Credible Simulation Process supports the logical sequence*

Figure 4 shows how the information, the CSP and the coordinated SSP traceability [6] data format Simulation Task Meta Data (STMD) interact. The information and data are linked to the STMD XML schema file according to their type. This STMD XML file with the links can then be saved or exchanged and is the basic building block for traceability. The GlueParticle SSP traceability standard requires that the importing and exporting tools and data management systems support this standard and structure. However, no specifications are made as to how the internal data management tool must be structured.

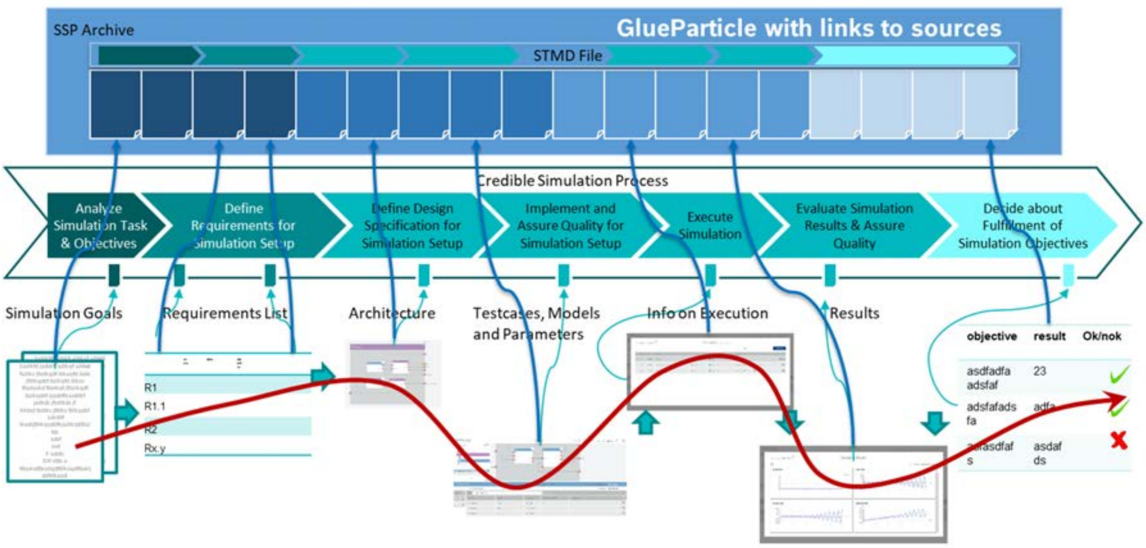


*Figure 4. Interaction of information, CSP and SSP-traceability data format STMD*

### 2.3.2. Interaction process, data and workflow using the Credible Modeling Process (CMP)

There is a large overlap between the process steps for a simulation and modeling process (see figure 5)

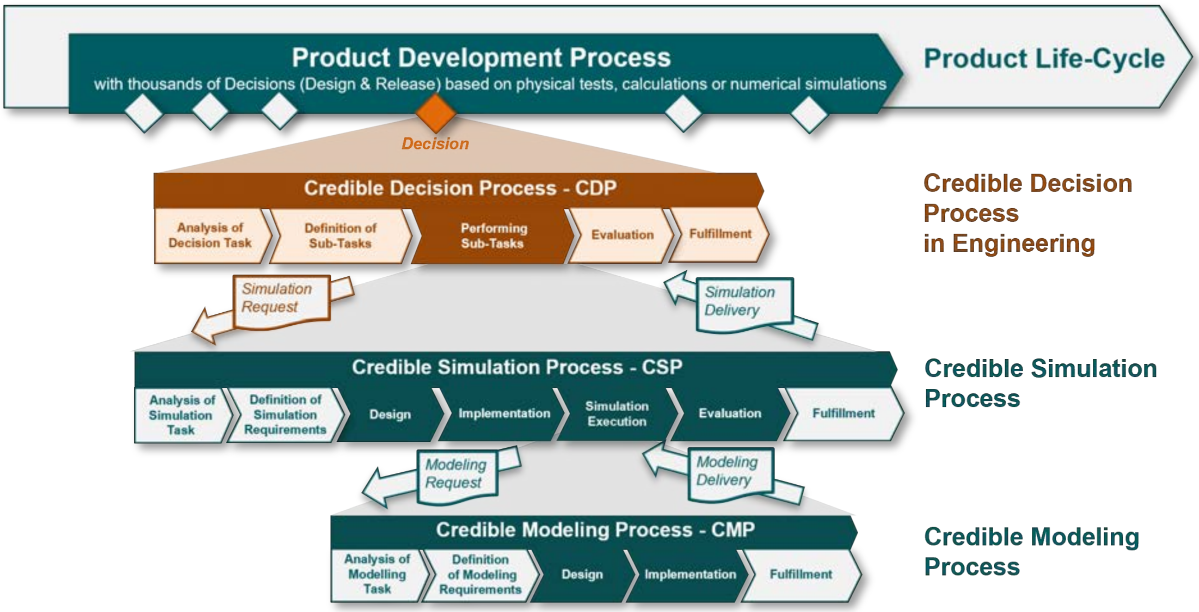


*Figure 5. Credible Simulation Process Framework*

The relevant process steps are

- Analyze the problem at hand
- Deduce and breakdown requirements related to Modeling & Simulation (M&S) activities
- Formulate a (M&S) design specification
- Implementation (resulting in a tested model or simulation infrastructure)
- Execution (only for the simulation task, not the modeling task)
- Evaluation of the simulation results (only for the simulation task, not the modeling task)
- Fulfillment

We analyzed the typical modeling processes, there are typical challenges in simulation model development (Figure 6), which are also challenges in the simulation process.

- Mismatch of interface, parameter specification to implementation in simulation tool
- Copy / Paste errors

- Inconsistent versions, content of requirement & design specification and documentation
- High review efforts
- Often no single source of truth

Establishing functionality to ensure traceeability between the evolution of artifacts, and the decision that drive the evolution, is seen as means to mitigate a number of these challanges. If not adress, models, simulations, and in the end model based decision making can not be done in a credible way.

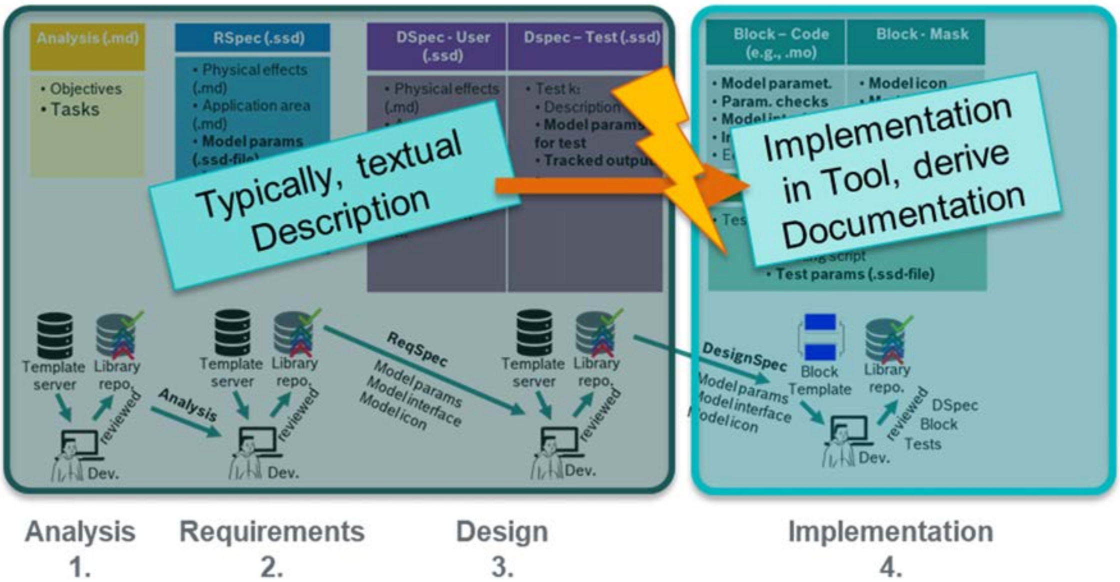


*Figure 6. Current state of practice in typical modeling processes*

To overcome the challenges, we have therefore added the workflow layer to the Credible Modeling Process (CMP) in the OpenScaling project (Figure 7). Figure 7 visualizes three different layers: a schematic process representation at the top, relevant example dataformats in the middle, and screenshots of different tools parcing the data artifacts at the bottom.

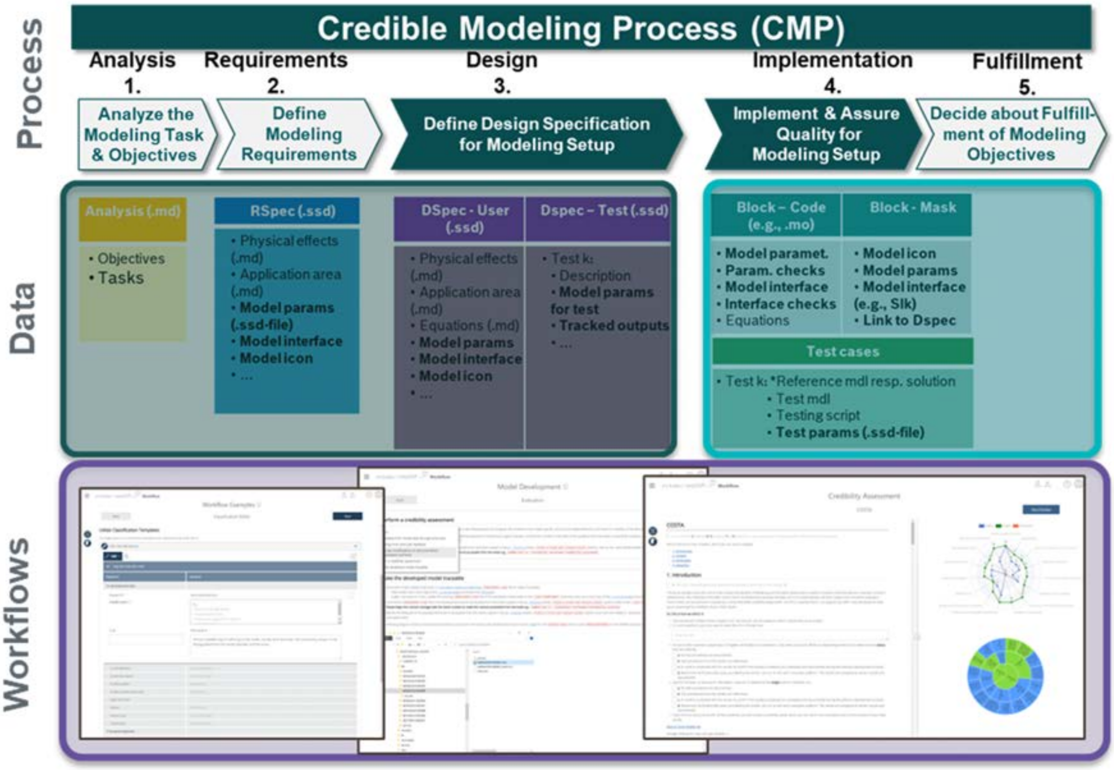


*Figure 7. Three layer approach for the Credible Modeling Process (CMP)*

We need the CMP data schema, which is analogous to the CSP, to structure the information. The data layer is separate from this and is connected to the process layer via links. The data is stored and versioned in a database The user should not interact directly with these two layers but should do so via the workflow layer. The workflows are created specifically for the users and their way of working

**The use of the GlueParticle building blocks, especially the SSP and SSP traceability standards, enables the following advantages**

- Well-defined process to convey information in a structured manner
- Use of SSP for efficient (tool agnostic and minimalistic) model exchange
- Use of SSP Traceability to store and exchange process information in a standardized form
- Easy to use via workflows based on common formats (Markdown, YAML)
- Automatic process & model documentation
- Data exchange via SSP package Tool support for this is incrementally implemented as part of OpenSCALING project and it is tested using sucessively expanded examples for final verification through selected demonstrators as defined in WP1.

### 2.3.3. Use of metadata with SRMD and MIC Core

A subset of the SSP Tracability Standard Layer is the Simulation Resource Meta Data (SRMD) format (Figure 9). The SRMD format is described as “SRMD files are used to define essential metadata for resources that can help users quickly understand the content” in the SSP Traceability specification. The SRMD format allows key-value pairs to be packaged in a portable XML file that can be parsed and presented by any tool supporting the SSP Traceability standard. It is used in the OpenScaling project for the standardized exchange of metadata. For the exchange of the core attributes of model metadata, the MIC-Core standard is to be used to the extent required by each application, which provides a harmonized semantic of the attributes, see Figure 7. Figure 8 illustrates a set of MIC-Core specified attributes instatiated in an SRMD file; for example, the keyword *adminitrative-data.model.name* has value *permanent Magnet Syncronous Machine in abc*.

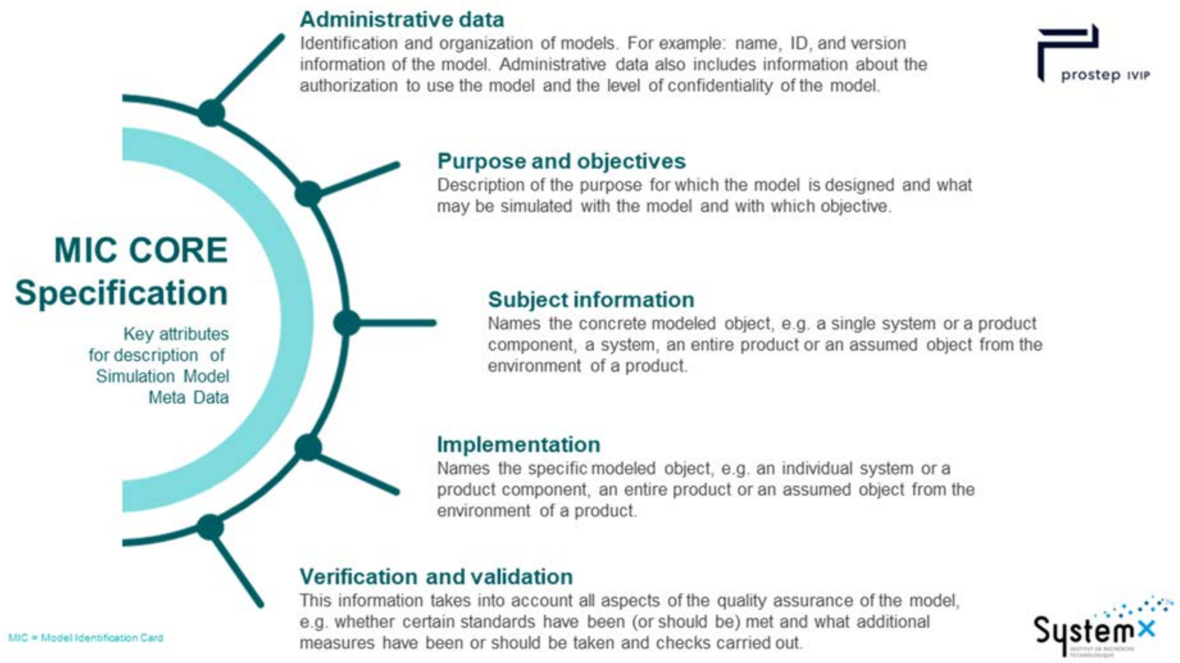


*Figure 8. Overview of attributes which semantics is aligned with several meta data standards*

These SRMD XML files can be transferred as part of the SSP packages. The SRMD format and MIC-Core standard are already supported by several tools (Figure 8)


*Figure 9. The SRMD format and MIC-Core standard is already supported by several tools*

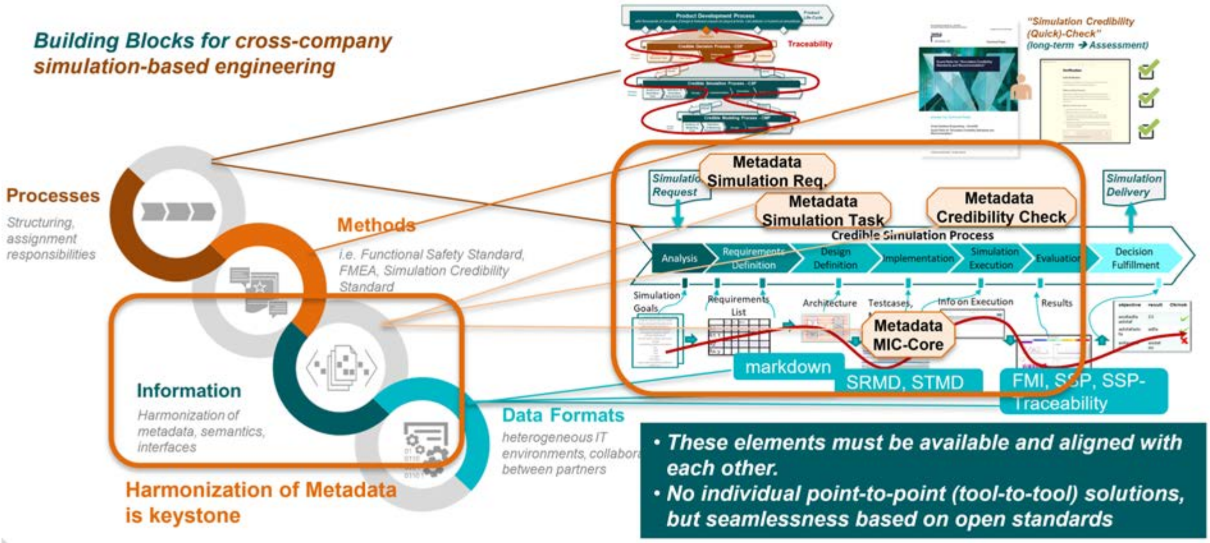


*Figure 10. Usage of the building blocks processes, methods, information (metadata) and data formats*

## 2.3.4. Metadata for Simulation Task and Request

The harmonization of the semantics of metadata is a keystone in collaboration. Therefore we defined and aligned also a set of metadata for simulation requests and simulation tasks. In Figure 10 is shown how the building blocks processes, methods, information (metadata) and data formats fits together. * We need a clear process structure with defined exchange packages between the processes. * We need an alignment of information and metadata for traceability and collaboration * We need standardized formats for exchange in hetergeneous simulation and IT environments * Methods should use and support this.

We defined approximatly 150 potential metadata attributes which are typically used and needed for realizing traceability. There are three different kinds of attributes: Mandatory, Recommended, Optional. An overview about the detailing of the metadata explanation is shown in Figure 11.

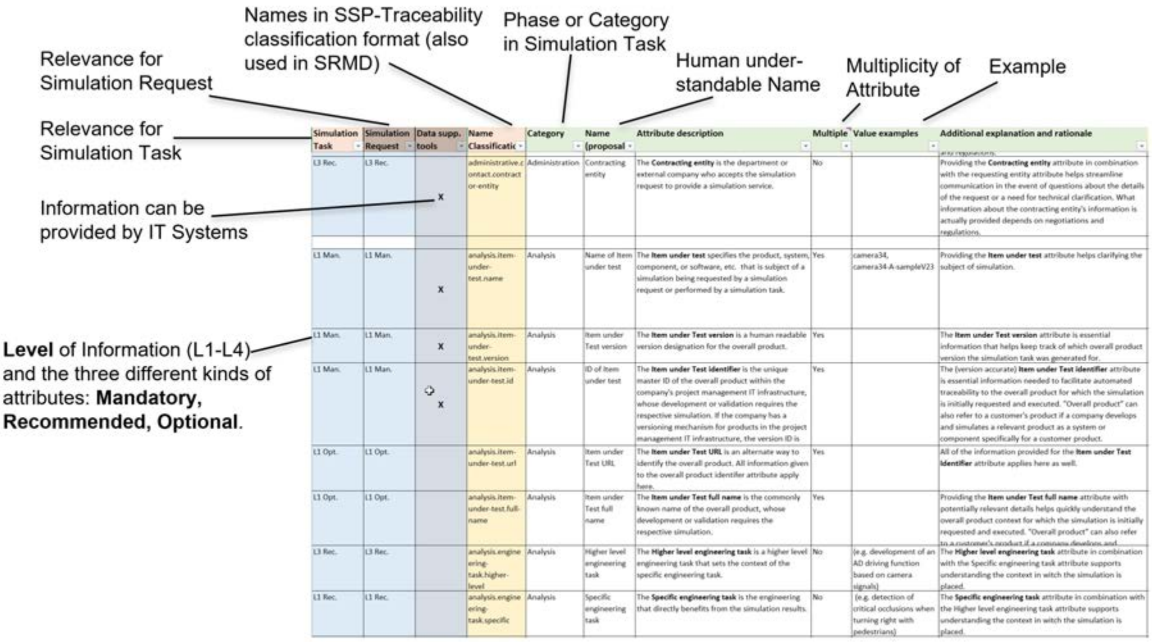


*Figure 11. Overview of the detailing of the metadata explanation*

It depends on the use case, how many metadata are needed. An overview of typical usecases and the metadata which are needed is given in Figur 11. If there is for example only a traceability for the description and fulfillment of the task necessary, then only about 8 Metadata are necessary (UC1)

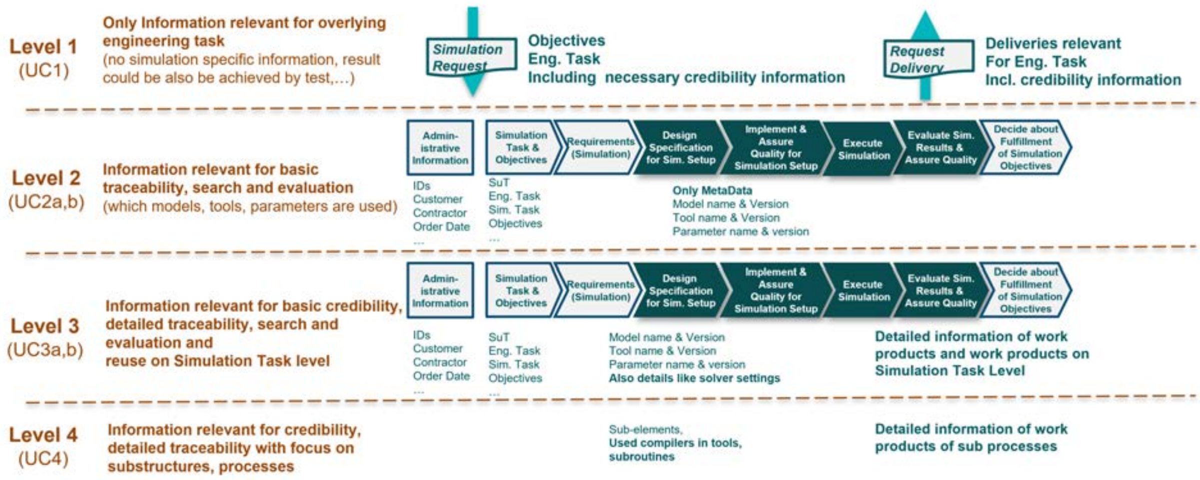


*Figure 12. Overview of typical usecases and the metadata which are needed*

# 3. Example application processess and their relations.

## 3.1. Round-trip Enginnering for the Development of cOmplex systems (REDO)

In this section, a round-trip engineering solution for the development of complex systems (REDO), such as aircraft, is presented. The interested reader can find a more detailed description in, for example, [1,2]. Round-trip engineering is a development methodology sprung out of the software development community [8, 9]. It highlights paralellization in development which is enabled by frequent points of syncronization. Tool support enabling engineers to move freely between disciplines are a pre-requisite to Round-trip engineering and, interoperability standards are seen as a prerequisite to derive efficient tool support. This particular approach has been incrementally formulated by Saab Aeronautics throughout numerous collaborative and internal research projects including the ITEA projects OpenCPS [3] and EMBrACE [4]. The presented process does not include a life-cycle view of development. However, this view can be incorporated through the mechanisms of the SSP Tracability standard and the Credible Simulation Process (CSP) as described in the previous section. Please note that most of the process steps of the CMP can be identified in the REDO methodology as presented in Figure 12.

The swimlanes in Figure 12 describe the processes in which the actors, working within each swimlane, develop the resulting artifacts deduced as relevant during the application of Model Based Engineering (MBE): system architecture, analysis architecture, analysis models, and simulation results. Communication cross swimlane borders is shown to be conducted by means of the standards in focus: requirements models can be communicated using the SysML or the Common Requirements Modeling Language (CRML), architecture models using SysML or as System Structure Description (SSD) files, analysis architectures as SSD files, and analysis models as Functional Mock-up Units (FMUs) or System Structure and Parameterization (SSP) files depending on the level of abstraction the analysis model is to capture. The intended use of the model or simulator that is to be developed is communicated as CRML and/or Modelica code, and/or as natural language, and/or through the SRMD format of the SSP Traceability specification. Any information needed to communicate Verification, Validation, Sensitivity Analysis (SA), or Uncertainty Quantification (UQ) data is packaged according to the schema specified within the OpenSCALING project.

The architecture is separated into two different swimlanes in Figure 12. This separation is done to highlight the need for different views of the architecture. The system architecture artifacts express the design space, with a focus on the systems, their constituent components, and their interfaces to be evaluated, analyzed, and developed. The analysis architecture artifacts, on the other hand, express the design space and the interfaces of the artifacts needed for analyzing and evaluating the system architecture and the SoI requirements. The constituent components can, for example, be differently grouped in the two different representations; for example, as a result of analysis model availability and sought reuse, or available modeling competence and tooling. The artifacts are likely developed in the same tools, by the same engineers, but they are founded on different needs and requirements. The system architecture is a result of the requirements on SoI to be developed. The analysis architecture is not only founded on the SoI requirements, but also the intended-use of the models or simulators needed for the evaluation.

Analysis models are typically executable representations of domain-specific models or simulators needed for the analysis. However, the analysis model swimlane also includes the development of support models in the Create design parts activity. Support models are here seen as the models providing the information needed to evolve the executable analysis models from representing many different configurations of the SoI to representing the configurations to be evaluated. Finally, analysis models are advantageously communicated to its stakeholders via the FMI and SSP standards as both these standards intend to provide a least common denominator for that particular type of model-based communication. The Implement Analysis Model process of the REDO methodology is mapped against the CMP in Figure 13. The REDO methodology thus implements, and details, selected aspects of the CMP. Please note that the process activities shown in Figure 12 and Figure 13 are slightly different grouped in the two different views. Figure 12 aims to highlight the need of design information, such as geometry representations, and thus separates the development activity into two activities.

The Execute simulation swimlane includes an activity in which the intended- use of the simulation is formulated. This should be seen as an end-user view of the simulation purpose. This activity needs to be executed in close dialog with the Modelling and Simulation engineers developing the analysis models such that the analysis model's purpose can be agreed upon with clearly stated acceptance criteria that allow for objective evaluation. Furthermore, the Assess credibility activity includes the aggregation of gathered evidence from, for example, Verification & Validation of the constituent analysis model(s) as well as any traceability information as specified in, for example, MIC-Core.

The feedback characteristic of round-trip engineering is highlighted in Figure 12 by the control flow of each swimlane (dashed arrows) as well as the activities marked as green. The activities marked as green provide the feedback across the different processes; the Execute Simulation process initiates propagation of the simulation results via a central storage of digital information. This central storage is polled in all the dependent processes in which the simulation results are being evaluated.

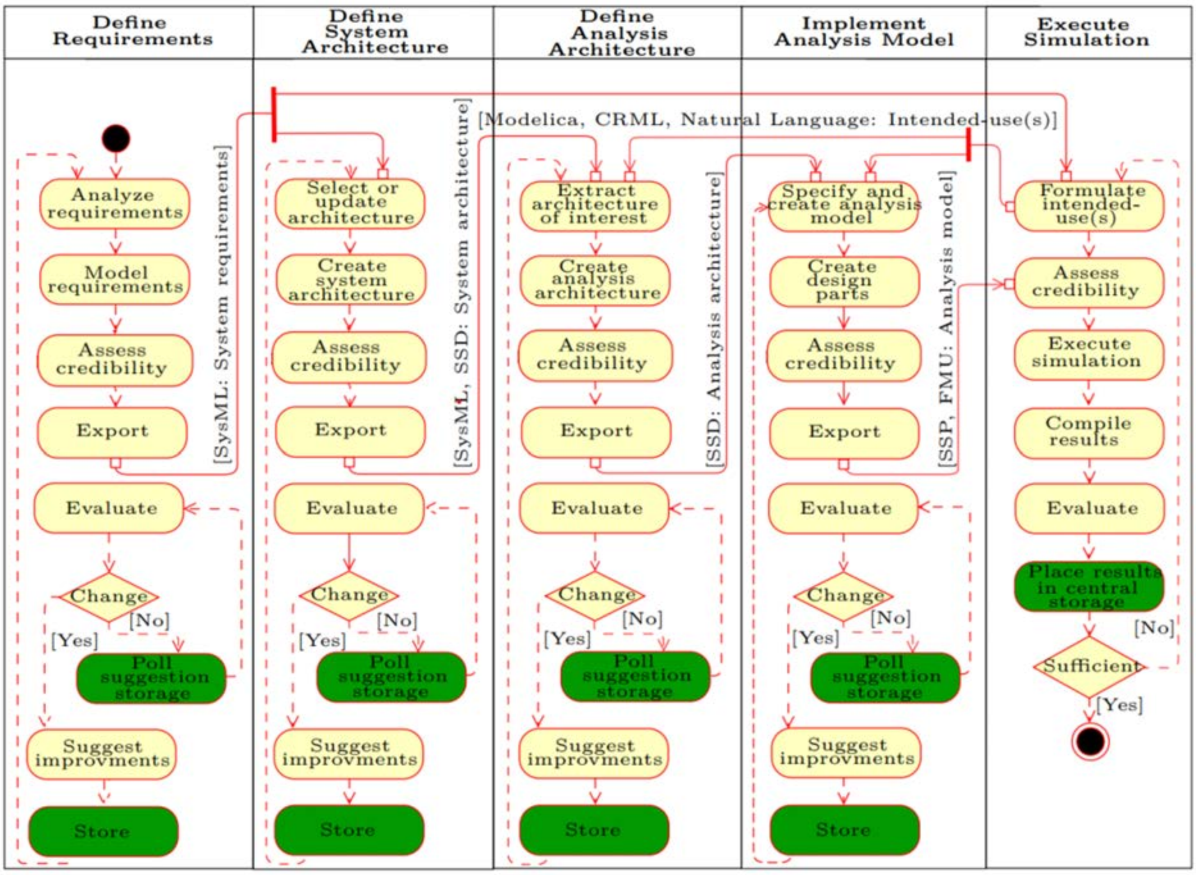


*Figure 13. Round-trip Enginnering for the Development of cOmplex systems (REDO)*

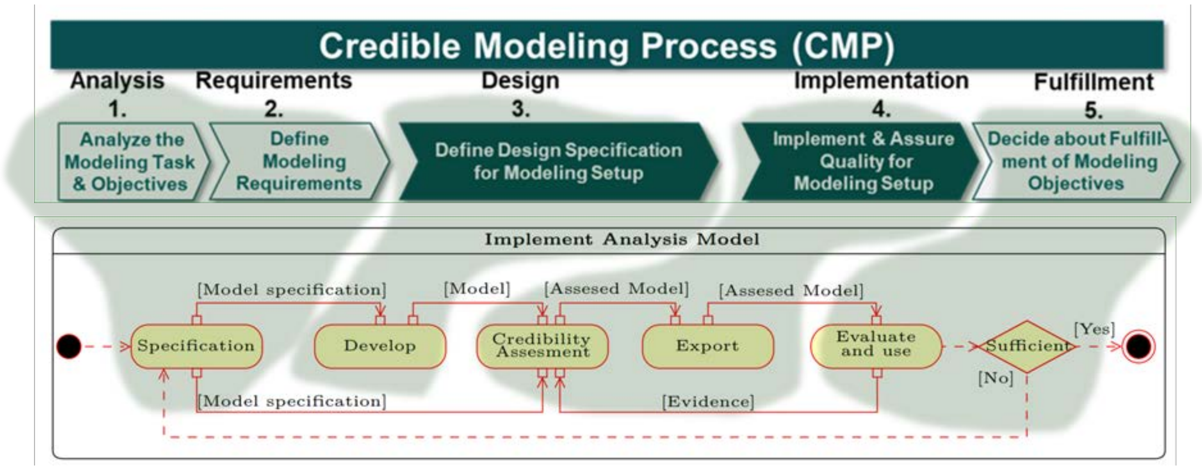


*Figure 14. Mapping between the Credible Modeling Process (CMP) and the Implement Analysis Model Process of the REDO methodology*

## 3.2. Software and Embedded Systems

AUTOSAR is a software platform for automotive software, that also entails an application software modelling approach. Software components, compositions, ports, datatypes and connectors can be modelled at a level of detail that is enough for building the target code for ECUs. But it also brings sufficient (and necessary) detail for building and integrating artifacts for a simulation target.

To complement the AUTOSAR application software description, EAST-ADL as an architecture description language, defines non-software-relevant aspects of the embedded system. As such it also includes plant model components, with a focus on functional and software aspects. Variability, requirements, functional safety and timing annotations are further aspects of the EAST-ADL model.

EAST-ADL complements FMI as it manages complementary aspects, in particular the product line perspective. FMI defines the behavior of plant components while FMI-SSP defines structure of a specific system instance, or set of instances. In the REDO context, the EAST-ADL model supports the analysis architecture by its purpose-driven representation of non-software. The software and embedded representation would on the other hand support both system architecture and analysis architecture, as the software, ECUs, electrics and electronics can be represented according to the real-world system.

The REDO implement and execute simulation swimlanes can largely be performed automatically, based on a variant rich REDO analysis architecture. Using variability mechanisms of EAST-ADL, multiple simulation purposes and platforms can be captured in the same model. Prior to simulation, the integrated architecture model is resolved to configure structural and behavioural variability. An integrated simulation model is then integrated for the chosen platform, for example as an SSP or Simulink model.

Depending on configuration, the simulation will be synthesized and executed. For example, one may choose between energy or performance, high or low fidelity, Simulink or OMSimulator [7] in the configuration, and the corresponding components of the model would be kept and parameterized accordingly.

A large scale system workflow requires a semantically well-defined representation of the system under development and its related artefacts. A combination of EAST-ADL and AUTOSAR can represent development artefacts for automotive embedded systems. Variability and traceability is covered to secure management of the inherent complexity.

An overview is found in AUTOSAR_EAST-ADL_SimulationDescription. ERTS_2018_Simulation-BasedVerificationforCI focuses on SIL simulation.

# 4. Data Formats

This section lists the dataformats relevant for the artifacts in focus of the OpenSCALING project in generall and WP5 specifically. Each listed format is tagged with “exchange” or “internal”. These tags highlight whether the format is suitable for capturing information used during artifact development (“internal”) or if the format is suitable for communicating the artifact itself with external stakeholders (“exchange”). So, the “exchange” formats are, in the context of the REDO methodology, suitable for communication across the swimlanes and “internal” are suitable for work within each swimlane. The formats selected to be in project focus are marked as **bold**. Please note that, at the time of writing, no format for the exchange of requirements has been selected. Requirements breakdown, modeling, and exchange is not the primary focus of the OpenSCALING project. Formats in cursive are deduced as relevant for partner specific project work .

The presented and selected dataformats rely on well established core formats. These core formats are: * CSV: https://en.wikipedia.org/wiki/Comma-separated_values CSV files are textual and organize content according to columns or rows. * JSON: https://www.json.org/ The Json format is textual and human readable because a key-value pair approach is used. The file will thus contain the field names of values and not only the values causing some overhead. * YAML: https://yaml.org/ The YAMLformat is textual and human readable because a key-value pair approach is used. The file will thus contain the field names of values and not only the values, thus causing some overhead * XML: https://www.w3.org/XML/ XML is textual, a bit verbose and not so readable, but has the benefit that an XML schema can be used to prescribe structure. * Text: Free text that follows syntax of a specific format, such as scripts and languages

**Documentation**

- **Markdown** (with extension supported by gitlab and github)
  - **MathML**
  - **MathJax**
- **SSP Traceability** (exchange)
- *Html* (exchange and internal)
- *Word* (internal)
- *LaTeX* (internal)
- *ASCIIdoc* (exchange and internal)

**Analysis Models**

- **FMI**
  - **2.X** (exchange)
  - **3.X** (exchange)
- *Modelica* (internal and exchange)
- *Simulink* (internal)
- STEP (exchange)
- **SSP** (exchange)
- *ONNX* (exchange)
- *Autosar* (internal and exchange)

**ECU Networks**

- *Autosar* (internal and exchange)
- *LS-BUS* (FMI3.0) (internal and exchange)

**Analysis and System Architectures**

- **SSP**: SSD, SSM, SSV
- *SysML*
  - *v1* (internal)
  - *v2* (internal and exchange)

**Parameters**

- **SSV** (exchange)

- **SSD** (exchange)
- *General XML* (only the MA standards)
- *FMI-LS-Ref* (exchange)
- Binary formats

**Requirements**

- *ReqIF* (internal and exchange)
- *Textual* (internal)
- *CRML* (internal and exchange)
- *SysML*
  - *v1* (internal)
  - *v2* (internal and exchange)

**Testcases and time series data formats**

- *SysML Flow charts* (SysML v2 has verification cases) (internal and exchange)
- *General Scripts* (in any language) (internal)
- *Matlab v4*
- *CSV*
- *hdf5*
- *JSON*
- *Markdown*
- *FMI-LS-Ref* (exchange)
- *eFMI* (behavior model containers) (exchange)
- ***FMI* (Relevant if testcases are expressed as executable analysis models) (exchange)**
- ***MA-CSV* (Modelica Association CSV Specification) (exchange)**

**Traceability of Simulation Task**

- **SSP-Traceability** (exchange)
- *OSLC* (internal)

**Simulation results related to V&V and UQ of analysis models (static information), coupling to WP2**

- **SSP-Traceability**
  - *SRMD* (for expressing V&V and UQ results and model Operational Domains, Domains of Validity/UQ)(exchange)
- **LS-UQ** (for expressing V&V and UQ results and model Operational Domains, Domains of Validity/UQ)(exchange)
- *Modelica* (for expressing V&V and UQ results)(exchange)(internal)
- *SSV* (for expressing V&V and UQ results) (exchange)

**Meta Data/Information (tool versions, git tags, etc..)**

- **SSP-Traceability**
  - **SRMD[XML]** (exchange)
- *JSON* (internal)
- *MIC-Core* (linked to SSP Traceability, there is an implementation in SRMD of MIC-Core)

# 5. LSS workflow tool requirements

This section summarizes the requirements deduced from analysing the workflows and data formats in focus of WP5. At the time of writing, tool support for all requirements is available in, or accounted for in the road map of, all the relevant tools developed by the tool vendors participating in the project. Each requirement is tagged with information connecting it to one or more project demonstrators. The tagged demonstrators indicate the requirement relevancy and suggested platform for verification. Furthermore, the adopted requirement numbering scheme is thought to indicate requirements break down (for example, Req 1.1 describes a specialised sub-set of Req 1).

**Req 1 (Demonstrator 1.6)**

- Export of system architectures shall be done according to System Structure Description (SSD) xml format of the System Structure and Parameterization (SSP). Focus on SSP 2.0.

**Req 1.1 (Demonstrator 1.6, Demonstrator 1.3)**

- Export of analysis architectures shall be done according to System Structure Description (SSD) xml format of the System Structure and Parameterization (SSP). Focus on SSP 2.0.

**Req 2 (Demonstrator 1.6)**

- Import of system architectures shall be done according to System Structure

Description (SSD) xml format of the System Structure and Parameterization (SSP). Focus on SSP 2.0.

**Req 2.1 (Demonstrator 1.6, Demonstrator 1.3)**

- Import of analysis architectures shall be done according to System Structure Description (SSD) xml format of the System Structure and Parameterization (SSP). Focus on SSP 2.0.

**Req 2.2 (Demonstrator 1.6, Demonstrator 1.3)**

- An Architecure authoring tool (system or analysis) should provide tool support to manage updated (imported) architectures. This includes diff/merge functionality concerning both static code and behaviour within the authoring tool.

**Req 3 (Demonstrator 1.6, Demonstrator 1.3, Demonstrator 1.5)**

- Export of Analysis model (component or a system on the ssp-file level) interfaces shall be done according to the ModelDescription.xml format of the Functional Mock-up Interface (FMI) (FMI 2 and FMI 3), or according to the System Structure Description (SSD) xml format of the System Structure and Parameterization (SSP), depending on the granualarity required by the intended-use and he tool support of relevant tools.

**Req 4 (Demonstrator 1.6, Demonstrator 1.3, Demonstrator 1.5)**

- Import of Analysis model (component or a system on the ssp-file level) interfaces shall be done according to the ModelDescription.xml format of the Functional Mock-up Interface (FMI), or according to the System Structure Description (SSD) xml format of the System Structure and Parameterization (SSP), depending on the granualarity required by the intended-use.

**Req 5 (Demonstrator 1.6)**

- Modelling and Simulation tools shall support incremental addition/removal of resources (existing or new) in analysis architectures (FMUs, ModelDescription files, SSPs, SSDs, SSMs, SSVs, SRMDs, DTMDs, and STMDs) to an SSP file:

**Req 5.1 (Demonstrator 1.6)**

- Modelling and Simulation tools shall support the creation and removal of links (for example SSM files) between added/removed supplemental resources (SSVs, SRMDs, DTMDs, STMDs) and dependent artifacts in analysis architectures (FMUs, Modelica models, SSPs)

**Req 6 (Demonstrator 1.6, Demonstrator 1.3, Demonstrator 1.5)**

- Tools shall support the incremental addition/removal of meta-data capturing life-cycle information (Meta-data for simulation task) and uncertainty quantification information (WP2 format). The meta-data is to be expressed on a standardized format (Modelica, STMD, DTMD,SRMD of the SSP Traceability specification) to fascilitate interoperability towards other exploited standards and for simplified implementation in all tools relevant during development. Examples of meta-data to capture are: artifact generation/modification date, purpose, static metrics of credibility, dynamic metrics of credibility, etc.

**Req 6.1 (Demonstrator 1.6, Demonstrator 1.3, Demonstrator 1.5)**

- Tools shall support the incremental addition/removal of meta-data artifacts, and their constituent entries, capturing credibility information expressed in the XML format specified in WP2.

**Req 6.2 (Demonstrator 1,6, Demonstrator 1.5, Demonstrator 1.3)**

- Tools shall provide functionality to navigate in and interpret meta-data as specified in WP2, in order to identify and separate between Domains of Validation (DoVs) and Domains of Uncertainty Quantification (DoUQs) encapsulated in the same meta-data set.

**Req 6.3 (Demonstrator 1.6, Demonstrator 1.3, Demonstrator 1.5)**

- Importing tools shall be enable to parse and present information related to the numerical aspects of a conducted credibility assessment experiment packaged in the xml format specified in WP2.

**Req 6.4 (Demonstrator 1.6, Demonstrator 1.3)**

- Modelica modeling tools should be enable to export information related to the numerical aspects of a credibility assessment to the xml format specified in WP2.

**Req 6.5 (Demonstrator 1.3, Demonstrator 1.5)**

- It shall be possible to trace from a system, subsystem or component in an SSP container to the model revision, or iteration/version, in a version control system used to create this SSP container or model.

**Req 6.6 (Demonstrator 1.5, Demonstrator 1.3, Demonstrator 1.6)**

- It shall be possible to repeat the surrogate model creation with stored meta-data and re-insert updated surrogate.

**Req 6.7 (Demonstrator 1.3, Demonstrator 1.5)**

- It shall be possible to trace from a simulation result back to the model revision, or iteration/version, in a version control system used to generate these results.

**Req 6.8 (Demonstrator 1.5, Demonstrator 1.3)**

- Store information for postprocessing/ visualization tools and files for (sub-)models. It shall be possible to visualize results of a model or model part with given template files for visualization software

**Req 6.9 (Demonstrator 1.5, Demonstrator 1.3)**

- Store information for required result accuracy
- It shall be possible to store result variable bounds for checking if a result is in certain bounds, for transient simulations for given time interval.

**Req 7 (Demonstrator 1.6, Demonstrator 1.5)**

- Tools shall support an agreed upon standardized format (proposal: MA-CSV) for capturing time series data, for example for exchanging simulation boundary conditions and model validation and verification data. The time-series format should be aligned with that of FMI and SSP.

**Req 8 (Demonstrator 1.6)**

- Executable analysis architecture (composition) aggregated credibility information should be traceable to its source (component)

**Req 9 (Demonstrator 1.6)**

- Relevant tools shall provide support for authoring, parsing, and executing nested SSP-files (for maximising simulation artifact reuse and configuration management).

**Req 9.1 (Demonstrator 1.6)**

- Relevant tools shall provide support for authoring and parsing nested SSD-files (for maximising simulation artifact reuse and configuration management).

**Req 10 (Demonstrator 1.6)**

- Any relevant textual information, that need to be communicated between engineering tools and disciplines, should be expressed using markdown, only with the extensions supported by github, packaged and linked in a simulation package (SSP-file)

**Req 11 (Demonstrator 1.6)**

- Any information in a simulation package that is unused or functionally unchanged by an importing tool shall remain part of the simulation package upon re-export.

**Req 12 (Demonstrator 1.3, Demonstrator 1.5)**

- Modeling tools shall support connectors describing a physical coupling including flow and stream variables using FMI and SSP, when required by the modeling method.

**Req 13 (Demonstrator 1.3, Demonstrator 1.5, Demonstrator 1.6)**

- Support of Modelica Bus connectors describing a signal coupling based on a dictionary of signal names in FMI and SSP based on the FMI3 terminals of type Bus.

**Req 14 (Demonstrator 1.5, Demonstrator 1.3)**

- Store information regarding pre-compiled model parts if not clear from model parameters.

# 6. Summary and conclusions

As part of WP5, existing approaches such as the prostep SmartSE GlueParticle approach—with its building blocks—and the SSP traceability layered standard were used and further developed. Furthermore, industrial MBSE processes, such as the REDO methodology, could benefit greatly from selected aspects of, for example, SSP Traceability, and a mapping to the Credible Simulation Process framework has therefore been initiated in this document. This mapping, along with the needs of the demonstrators developed within the OpenSCALING project, has been utilized to derive a set of tool requirements needed to continuously evolve, parse, and exploit the exchange packages in focus. From the perspective of WP5, the workflows and standardized exchange packages that connect the sub-workflows are essential for any tool-agnostic methodology that allows the best-suited engineering tool to be used for each engineering task. To achieve the desired consistency and traceability, it is considered necessary to harmonize the processes and the workflows derived from them. Additionally, to consider the methods used—for example, for the model-based development of Large Scale Systems (LSS) and for credibility assessment—the relevant lifecycle information and credibility metadata (related to numerical predictions) need to be captured in the exchange packages. The GlueParticle approach developed in prostep SmartSE is being used and further developed for this purpose within OpenSCALING. A cooperation agreement between the prostep SmartSE and OpenSCALING projects regarding the mutual exchange of information for the use and improvement of standards has been established to address these identified needs.

# 7. References

1. Hällqvist, Naeser, Wallén Axehill, and Herzog. Heterogenous System Modelling in Support of Incremental Development, Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences, 2022, Stockholm, Sweden
2. Hällqvist. On the Realization of Credible Simulations in Aircraft Development: Efficient and Independent Validation Enabled by Automation, 2023, Linköping University Electronic Press
3. ITEA 3, OpenCPS project partners, Project 14018:Open Cyber-Physical System Model-Driven Certified Development}}, Retrieved from [https://www.opencps.eu/], Accessed: 2018-06-21
4. ITEA 4, Environment for model-based rigorous adaptive co-design and operation of CPS, Retrieved from [https://itea3.org/project/embrace.html], Accessed: 2019-11-12
5. SmartSE Project Partners, SmartSE Recommendation V4, Part H – Credible Simulation Process, Tech Report, Retrieved from https://www.prostep.org/fileadmin/prod-pay-download-8c1d/PSI_11_V4_SmartSE_Part_H_CSP.pdf Accessed: 2025-06-24, 2024.
6. Modelica Association. SSP Traceability Specification. Version 1.0.0-Beta2. Retrieved from [https://github.com/modelica/ssp-ls-traceability], Acessed: 2025-06-26
7. Lennart Ochel et al. OMSimulator – Integrated FMI and TLM-based Co-simulation with Composite Model Editing and SSP, Proceeding of the 13th International Modelica Conference, 2019.
8. D. Rosca and L. Domingues, “A systematic Comparison of Roundtrip Software Engineering Approaches applied to UML Class Diagrams,” Procedia Computer Science, pp. 861-868, 2021.
9. S. Sendall and J. Kuster, “Taming Model Round-Trip Engineering,” in Proceedings of Workshop on Best Practices for Model Driven Software Development, Vancover, 2004.
10. International Council on Systems Engineering. Systems Engineering Handbook. 4th ed. John Wiley and Sons, Inc., 2015. isbn: 9781118999400.
11. ProStep IVP, MIC Core Specification, Retrieved from [https://mic-core.github.io/MIC-Core/main/], Accessed: 2025-08-15
12. Rosenlund, Erik et al. (2025). “Objectively Defined Intended Uses, a Prerequisite to Efficient MBSE”. In: Proceedings of the American Modelica Conference 2024, pp. 29–42. DOI: 10.3384/ecp20729.
13. Otter, Martin et al. (2022). “Towards Modelica Models with Credibility Information”. In:

Electronics 11.17. DOI: 10 .3390/electronics11172728.