

## CONCEPT FOR FUNCTION-TO-SIMULATION CONTINUITY

DOCUMENT TYPE: **DELIVERABLE**  
DELIVERABLE N°: **D2.1A**  
DISTRIBUTION LEVEL: **CONFIDENTIAL**  
DATE: **15/03/2022**  
VERSION: **1**



AUTHOR(S): **MARTIN VALDIVIA (TU BERLIN)**  
**PETER TIMMERMANN (PHILIPPS RESEARCH)**  
**ALEA SCOVILL (AGRO INTELLI)**  
**BASTIAAN BEIJER (KE WORKS)**

REVIEWED: **OLAF V.D. SLUIS (TU EINDHOVEN)**  
**CARINA FRESEMANN (TU BERLIN)**

FORMAL REVIEWED: **MARTIN KRAMMER (VIRTUAL VEHICLE)**  
**MICHAEL HOLZNER (ICONDU)**

APPROVED: **MARTIN BENEDIKT (VIRTUAL VEHICLE)**

PROJECT ACRONYM: **UPSIM**  
PROJECT TITLE: **UNLEASH POTENTIALS IN SIMULATION**  
ITEA PROJECT N°: **19006**  
CHALLENGE: **SMART ENGINEERING**  
PROJECT DURATION: **01/10/2020 - 30/09/2023**  
PROJECT WEBSITE: **[WWW.UPSIM-PROJECT.EU](http://WWW.UPSIM-PROJECT.EU)**  
COORDINATION: **VIRTUAL VEHICEL RESEARCH GMBH**  
PROJECT LEADER: **DR. MARTIN BENEDIKT**

**DOCUMENT HISTORY**

<b>REVISION</b>	<b>DATE</b>	<b>AUTHOR</b>	<b>DESCRIPTION</b>
0.1	28.02.2022	<b>MARTIN VALDIVIA</b>	1 <sup>st</sup> version
0.2	01.03.2022	<b>BASTIAAN BEIJER</b>	complements on tooling
0.3	04.03.2022	<b>PETER TIMMERMANS</b>	complements on modelling
0.4	10.03.2022	<b>ALEA SCOVILL</b>	complements on modelling
1.0	18.03.2022	<b>CARINA FRESEMANN</b>	Internal Review
1.4	DD/MM/YYYY	<b>PETER TIMMERMANS</b>	Device-tissue interaction added

## Executive summary

This document presents the function-to-simulation traceability concept to be applied within the UPSIM Project covering traceability, tooling and modelling quality aspects. Initially the credible simulation process<sup>1</sup> (CSP) and its extensions has been chosen as baseline. The proposed SSP traceability standard, also known as glue particle, is represented in the in an existing commercial tool (KE-Chain) serving as user interface as well as docking point for extensions. This document describes three conceptual extensions:

1. A pattern recognition algorithm to propose new glue particles based on identified patterns from existing glue particles and information stored in system architecture and simulation data,
2. a rule-based algorithm for maintenance of the existing glue particles, to identify potential redundant or obsolete glue particles, and
3. a classification algorithm to calculate the verification degree of the full system.

Machine learning algorithms are applied for these extensions, reducing the manual workload and providing a higher level of control and credibility.

The overall goal of credibility and useability of simulation refers to the modelling process itself. Here, several approaches are followed within UPSIM: the continuous improvement of physical facts as well as the validation and verification of these models within the framework of the CSP. In parallel, the need to establish digital twins requires real-time simulation capable models output with respect to the given use case and machine control mechanisms. Two use cases are considered in UPSIM: a catheter for arterial tissue and an agricultural robot, in which the device-tissue interaction and the soil-machine interaction is simulated respectively. Both use cases, as well as the current progress, are reported in this document.

---

<sup>1</sup> Set Level 4to5: <https://setlevel.de>

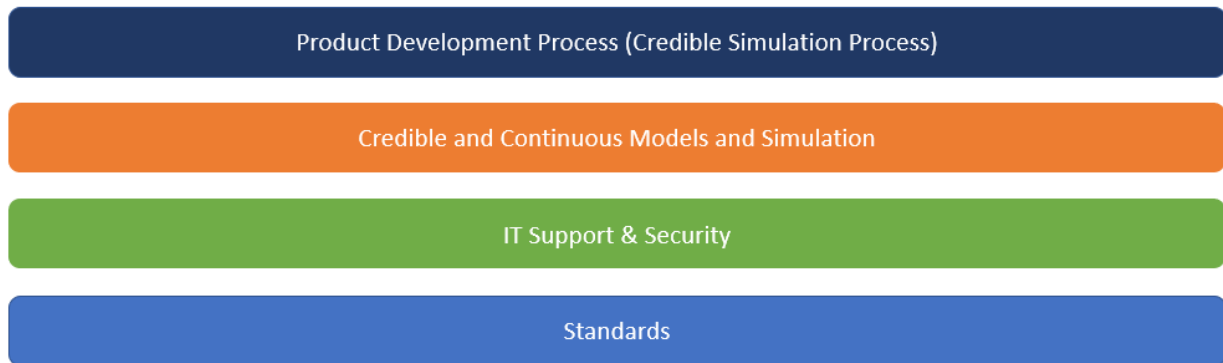
## Contents

1	Introduction .....	5
2	Function-to-Simulation Traceability Concept .....	7
2.1	Relevant definitions .....	7
2.2	State of the Art of Concepts and Tools Support .....	9
2.2.1	Web-based collaborative environment for managing the simulation governance process ..	10
2.2.2	Categorization of traceability solutions.....	12
2.3	UPSIM Traceability Concept.....	13
2.3.1	Traceability architecture .....	14
2.3.2	AI based trace link Services .....	14
2.3.3	Proposition of potentially missing trace links.....	15
2.3.4	Full system verification.....	15
2.3.5	Maintenance of existing glue particles .....	16
3	Use Case: Device-tissue interaction .....	17
3.1	Introduction.....	17
3.2	Arterial tissue models.....	17
3.2.1	Structure of the arterial wall .....	17
3.2.2	Mechanical behavior of the arterial wall .....	18
3.2.3	Damage in blood vessels .....	19
3.3	Constitutive model of the arterial wall.....	20
3.3.1	General formulation .....	20
3.3.2	Formulation including damage.....	21
3.3.3	Implementation into a Finite Element Code .....	21
3.4	Credible simulation process.....	21
3.4.1	Verification .....	22
3.4.2	Validation.....	22
3.5	Discussion .....	22
4	Use Case: Soil – Machine interaction .....	24
4.1	Introduction.....	24
4.2	Soil Model .....	24
4.3	Conclusion and next steps .....	25
5	Conclusions and Outlook .....	26
6	References.....	27
7	Acknowledgment .....	29
8	Appendix .....	31
8.1	Morphological Schema.....	31

# 1 Introduction

Modelling and simulation credibility is an overall goal of the UPSIM project. In line with this goal, required sub-goals of WP 2,3 and 4 are

- 1) to improve the visibility of trace-links throughout the product development process,
- 2) to improve the IT-tool basis for connectivity and traceability,
- 3) to improve the modelling precision and simulation execution time,
- 4) define credibility measurements,
- 5) and to define secure standards for collaboration aspects as well as data standards as enabling basis, see Figure 1.

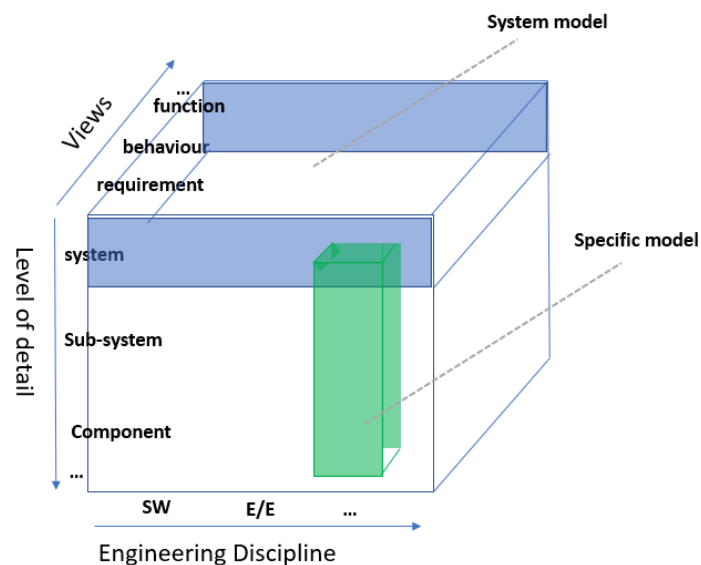


**Figure 1: PMTI Model adopted to UPSIM Project**

This document D2.1a mainly relates to the orange bar, taking the perspective of concepts and methods on how to establish reliable relations, that support the credible simulation process (chosen as UPSIM project reference process). IT tools are considered as needed to describe and manage data, provide a modelling environment, or provide a user interface for interaction.

Furthermore, it investigates how to establish reliable models for the medical example for tissue and potential damage modelling as well as modelling the soil of a field and the hydraulics behaviour of a field-robot to increase steering precision as second example.

Model interaction and traceability concepts are often seen from the perspective of Model-Based-Systems-Engineering. E.g. Hick et.al.<sup>2</sup> describe the required continuity and relation in their model cube, see Figure 2.



**Figure 2: Model Cube based on Hick et al. [1]**

<sup>2</sup> Hick, H., Bajzek, M. & Faustmann, C. Definition of a system model for model-based development. *SN Appl. Sci.* **1**, 1074 (2019). <https://doi.org/10.1007/s42452-019-1069-0>

Relevant research must investigate how to bridge the gap between the system and specific models. This document describes the traceability between the function to system simulation. This is done through an extension of the glue particle (see chapter 2.1). The document also includes methods of presenting reduced order models on a detailed simulation level.

This document 2.1a is established in close collaboration with document 2.1b, where the interchangeability of specific models is discussed on both tool and methods approach until a documentation level, to assure credibility and enable cross-company collaboration.

## 2 Function-to-Simulation Traceability Concept

To understand the traceability concept, first some core definitions and a brief state of the art analysis are introduced. Then, an analysis using the morphological schema from ISYPROM, a research project to accelerate the innovation through model based process and system design, is performed to define the classification and assessment of the approach. Finally, the traceability architecture is presented.

### 2.1 Relevant definitions

To get a better understanding of traceability, some terms need to be defined briefly.

#### Traceability definition

Product related decisions are taken increasingly based on simulation results. The product design process uses virtualization to validate simulations that predict behaviors of the product under the modeled conditions. The chain from simulation order to simulation result, as well as all relevant simulation related artefacts need to be traceable and comprehensible. This highlights the importance of traceability, to have an understandability of the complete chain. To define the traceability concept within the UPSIM project, first the term traceability must be defined. Literature indicates many definitions [2] [3] [4] for traceability that are viewpoint-dependent. A workshop took place with multiple UPSIM partners to elaborate the definition of traceability and function-to-continuity traceability. The elaborated definitions are as follows:

*"Traceability is a method to support the development process, where the bidirectional relations between artefacts within the same model or among different models are explicitly documented."*

The definition used in the project for function-to-simulation traceability is given as following:

*"Function-to-simulation-traceability is a method with a focus on simulation, where based on the simulation tasks, the simulation-related elements are linked from the function level through the system development process to the simulation model and the simulation results."*

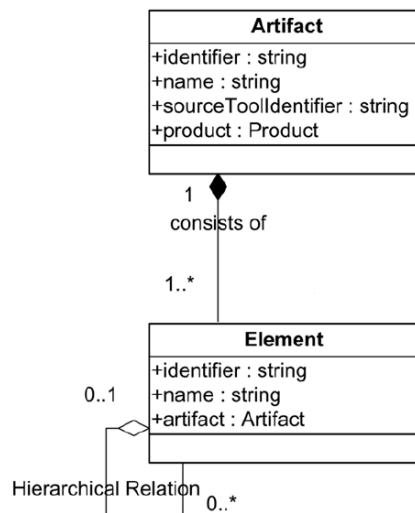
The derivation of this definitions are described in the  [UPSIM SotA Deliverable v0.5.docx](#)

#### Artifact

An artifact describes any kind of product-describing information generated during product development (specifications, functional structures, product structures, etc.), regardless of their informational executability. Each artifact consists of multiple elements that are hierarchical structured. This means it is a conglomerate of hierarchical structured elements. Each artifact also has a unique identifier, an identifier, an indication of which tool it was originally modeled in, and an indication to the product it describes. Artifacts chosen for demonstration purpose are MATLAB Simulink and different SysML diagrams, such as activity diagrams, internal block diagrams and block definition diagrams. These fulfil the MBSE critical bridge between function and simulation on a high and medium level of abstraction of a functional description. There is also the possibility to use of other artifacts related to project management tasks.

#### Element

An element is a component of an artifact. Elements have a hierarchical relationship between them and also have a unique identifier, an identifier and an indication to which artifacts it belongs. In Figure 3 an abstract artifact metamodel is shown, to get a graphic representation of the relations between artifacts and elements. Relevant elements for the UPSIM Projects are different blocks within the different artifacts, such as simulation blocks within MATLAB Simulink, or elements within the SysML Model, such as requirements, activities and blocks.



**Figure 3: Abstract Artifact Metamodel**

### Trace link

A trace link is used to explicitly document the relations between two different elements. They are represented as software objects that allow the unique identification of the two linked elements. Trace links can contain multiple features and metadata to support detailed analysis or the documentation of their creation. Trace links can also have the type, that gives a better understanding to the type of relations the different elements have, however in the scope of the UPSIM project, it was decided to use untyped trace links, since it is considered that the importance is to show the relation.

### Credible Simulation Process (CSP)

Traceability and comprehensibility contribute to simulation credibility. The traceability can be achieved by linking elements from different artefacts with each other. Comprehensibility can be achieved by presenting information generated during the task in a structured documented format. In addition to quality assurance criteria presented in standards as the ISO 26262-2 and the ASPICE, there are different credibility levels that are also defined within the UPSIM project but are not published yet. The needed credibility level depends on the specific application and the potential risk of the simulation task, so the effort for validation must be proportional to the risk associated to the consequences of an incorrect result.

During the product development process, engineers often use simulation as a method to predict behavior of the system or parts of it. These simulation-based engineering tasks (SBET) have simulation as an essential component for the execution and reliability of subtasks or as input for other phases in the product design process. In the SBET the solution approaches for the task itself in addition to the tasks objectives and requirements are defined. The information defined and the task itself become an input for the credible simulation process (CSP).

The CSP [5] is a generic seven step process that provides a framework for the usage of methods, but does not indicate which methods to use.

The phases are adapted from the basic structure of an engineering process and are mentioned in the bullet points below:

- Analyze Simulation Task and Objectives (Analysis Phase)
- Define Requirements for Simulation Setup (Requirement Phase)
- Define Design Specification for Simulation Setup (Design Phase)
- Implement and Assure Quality for Simulation Setup (Simulation Models, Parameters, Tests, Simulation Environment) (Implementation Phase)
- Execute Simulation (Execution Phase)
- Evaluate Simulation Results & Assure Quality (Evaluation Phase)
- Decide about Fulfillment of Simulation Objectives (Fulfillment Phase)

Each phase of the CSP can have sub-elements called steps, that are used to sub-structure the individual phases and can be processed in parallel or sequentially. In Figure 4 the CSP is represented with its phases and its steps.



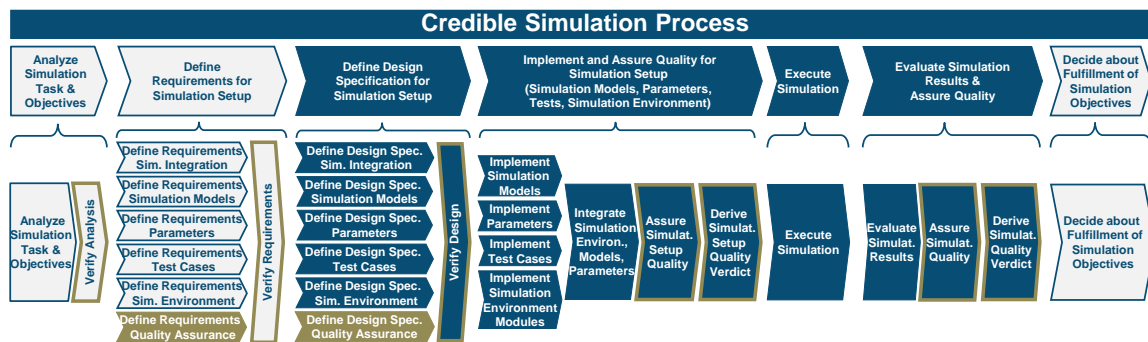


Figure 4: Credible Simulation Process [5]

## Glue Particle

The glue particle approach is a generic consistent data schema created to follow a process. The application of the glue particle in the UPSIM project is to create a consistent data schema that follows the credible simulation process. A glue particle is defined as a tool-independent XML file that records or references all information necessary to enhance the quality assurance, traceability and reusability of simulation processes and results. This XML files specific for simulation tasks are called Simulation Task Meta Data (STMD) files, that can be used to inspect and trace simulation tasks, or to support the reuse of simulation tasks or parts of it.

The information referenced or recorded by the glue particle includes the information handed over as part of the project along with a simulation order and the information used and generated during the planning, preparation, execution, validation and documentation of the simulation task, including externally defined and documented sources and information on responsibilities, quality checks and approvals. This is the basis of the traceability in the UPSIM project. It is intended to use to record or reference the elements within different artefacts related to one simulation task. The glue particle considers that formats for most artefacts already exist. It defines a format that allows the user to tie the relevant information together by referencing the existing artefacts, acting as a glue between the file formats. By including the relevant information through the system development process, a function-to-simulation traceability is achieved.

The glue particle stores a snapshot of the relevant artefacts at a specific date and time. It does not allow to check the version history of an element but indicates the version of the linked element.

The role of a single glue particle is to provide the necessary context and relationships tying the relevant information together in transit between two parties or systems. So, a system cannot be verified by a single glue particle, but by a set of glue particles.

The glue particle concept and the credible simulation process have been established in the SetLevel and the Project SmartSE project.

## 2.2 State of the Art of Concepts and Tools Support

Traceability tools have been developed to support engineers during the product development process, in order to provide proof that all requirements are considered and met, potential change impact may be analyzed quickly, and design rationales may be retrieved later on. There are existing commercial and academic tools; some of them with traceability as part of other features and others specific for the achievement of traceability. There are three approaches identified:

- The first concept are the traceability matrixes, which put all the elements within the artifacts as rows and columns of the matrix. Then the matrix is marked, where a dependency is defined. [6] (e.g. Looemo, Cambridge Advnced Modeller, Enterprise Architect, Cameo Systems Modeller)
- The second concept is an external add-on traceability tool, in which the tool retrieves information from external author tools and connects them within the same tool. This approach uses APIs to retrieve the information and assign and unique identifier to each element. Then the elements are linked to each other manually in the same tool. [2] (e.g. Intercax Syndeia, Reqtify)
- A third concept is a complete integrative approach, in which the whole design process is done within the same tool. Since all elements are comprehended within the same tool, the trace links are also stored in the tool, with no need for third party software. The trace links are created manually by the engineers. [2] (e.g. 3DEXperience)

### 2.2.1 Web-based collaborative environment for managing the simulation governance process

In the UPSIM project, the tool KE-chain<sup>3</sup> provides a web-based collaborative environment to manage the setup and execution of simulation processes. An impression of the KE-chain environment can be seen in Figure 5. Here the user can see on the left the login screen, where users can login using their personal account. Next, a user is able to navigate to any assigned project. In a project, forms can be accessed which allow a user to manage both manual and automated tasks.



**Figure 5: Impression of the KE-chain environment for UPSIM. From left to right the interfaces are shown for the login page, a projects overview, the project dashboard and an examples of a project specific form.**

KE-chain has a flexible data modelling environment capable of defining parts in a tree-like structure. For each part, properties can be specified. Example of property types are: text, integer, decimal numbers, attachments, selection lists, part reference properties. For each element of the model the multiplicity can be specified (exactly one and zero to many). Depending on the part model multiplicity, the instances of the data can be added either through a user interface or through scripts. On top of the data model, KE-chain forms can be used to create workflow driven user interfaces through which access to the data can be given to one or more assigned users. In a form, configurable widgets (tables, forms, attachment viewers, text, signatures) are used to create a visualization for the data. For each widget, access rights to the data can be specified with view and edit rights. Besides the user interface, either the REST API or Python API called Pykechain<sup>4</sup> can be used to automate the reading, processing and writing of data.

As part of a form model and the associated workflow, some tasks are automated using scripts, other tasks are manual. For automated tasks scripts are integrated in KE-chain as services. There are two type of services:

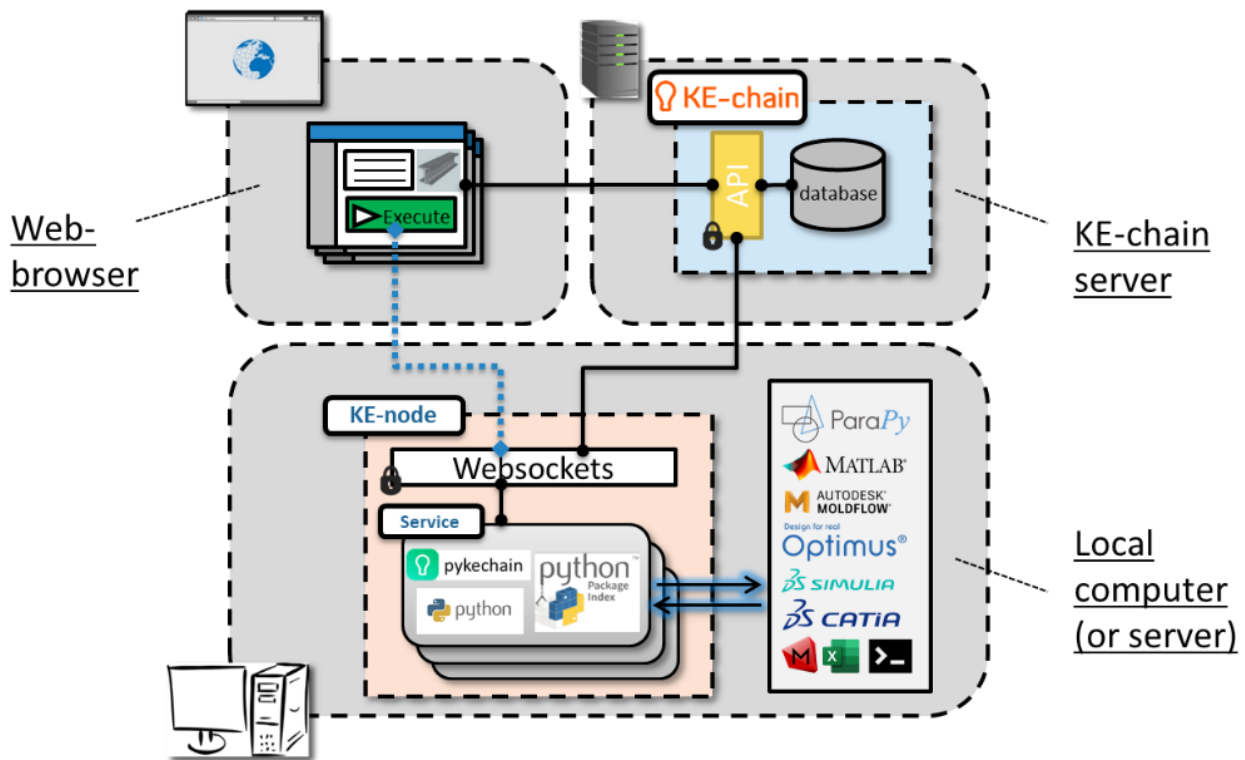
- On-platform script: Is executed on platform using Docker technology. Scripts use only Python scripts and can make use of any Python library which is pip installable. The python script can be a single Python file, or a zipped directory of python scripts which can be generated using kecpkg-tools<sup>5</sup>.
- Remote scripts: Integrated on local pc or on a server using KE-node technology. This enables a Python script to call other local (CAD) software

<sup>3</sup> “KE-chain engineering platform,” KE-works, [Online]. Available: <http://www.ke-chain.com/>. [Accessed 3 April 2022].

<sup>4</sup> KE-works, “KE-chain Python SDK Pykechain,” KE-works, [Online]. Available: <https://pykechain.readthedocs.io/en/main/>. [Accessed 3 April 2022].

<sup>5</sup> KE-works, “KECpkg-tools,” KE-works, [Online]. Available: <https://github.com/KE-works/kecpkg-tools>. [Accessed 04 3 2022].

For the importing and exporting of glue particle XML files for example, on-platform scripts should be developed. The way KE-node will be integrated with KE-chain is illustrated in Figure 6. KE-node will listen to remote service executions which can be invoked by users through the web-browser in the KE-chain platform. When receiving the trigger to execute, a Python script is triggered. This local Python can be integrated with any local software s.a. MATLAB, Excel, CAD tools either through a direct (Python) API or as command-line executable.



**Figure 6: Remote service integration using a KE-chain client called KE-node**

In the KE-chain application a form template has been developed which contains the data model, workflow and visualization for the simulation governance process. An example of the first task of this form in KE-chain can be seen in Figure 7. Here the user can see the first table of the workflow task "1. Analysis" through which the user can add/edit/delete/import/export and search on the inputs of a specific simulation case. In this example the Mild Hybrid Variant AAA-55 example use-case has been selected.

The screenshot displays the KE-chain software interface. The main content area shows a task titled "Examples - Mild Hybrid Variant AAA-55 - Credible simulation process - Analysis". The task is 45% complete and has 4 assignees. The process is divided into six phases: 1. ANALYSIS, 2. REQUIREMENTS, 3. DESIGN, 4. IMPLEMENTATION, 5. EXECUTION, and 6. EVA. The current phase is "Analyze the Simulation Task & Objectives (Analysis Phase)".

The "Inputs" section contains a table with the following data:

Inputs	Classification	Annotations	Link	Document
General project information	General project information	Engineering task, development objectives and specifications of higher-level engineering process A variant of a mild hybrid drive based on the DDC platform is to be developed. This application is a low cost variant	<a href="http://www.google.nl">http://www.google.nl</a>	
		Description of product, the requirements and environment (meant is the environment of the real product, e.g. for car -> road, climate zone, user group, frequency of use, ...)		

Figure 7: Example of the credible simulation process form in KE-chain

### 2.2.2 Categorization of traceability solutions

In the scope of the research project ISYPROM, a morphological analysis was proposed to analyze the classification and assessment of traceability approaches [7]. It is also considered a valuable tool to define new solutions for traceability approaches. In this section, the proposed traceability concept is going to be analyzed with that tool. The description of the criteria is shown in the Appendix. An overview of the categorization in the morphological schema is shown in Figure 8, where the red boxes indicate the selected property for each criterion within each subsection.

subsection: data management						
acquisition of artefacts	integrative	non-integrative				
artefact manipulation	yes	no				
duplication of artefacts	original data	copied data				
subsection: planning of traceability						
traceability schema	yes	no				
granularity	parameter tracing	no parameter tracing				
supported types of data structures	list	hierarchy	polyhierarchy	net		
subsection: recording of trace links						
modelling support	none	rule based	wizard	reuse	template	
subsection: use of trace links						
change handling	none	notification	propagation			
type of traceability	qualitative	quantitative				
traceability purpose	verification	analysis	synthesis	documentation	progress monitoring	synchronization
representation	matrix	graph	cross-reference			
subsection: properties of trace links						
process phase context	phase-internal	phase-spanning				
artefact context	artefact-internal	artefact-external				
semantic	typed	untyped				
storage location	central	distributed				

criteria
property

**Figure 8: Morphological Schema**

The classification within the morphological schema, gives a better understanding of the traceability approach and its purpose. This methodology supports the architecture development of the traceability concept of the UPSIM Project.

The indicated properties indicate the approach that is intended to be taken in the traceability concept. Knowing the considered properties for the traceability concept, methods for improving the concept can be suggested. So instead of using conventional algorithms to support the modeling, machine learning algorithms can be used for maintenance of existing trace links and proposing of new trace links. The use of machine learning in comparison to conventional algorithms is advantageous, since it can consider multiple features for the predictions. Additionally, these algorithms improve themselves continuously. There are also open options like the creation, visualization and management of trace links, that are going to be analyzed deeper in the chapter 2.3.

## 2.3 UPSIM Traceability Concept

The UPSIM Traceability concept uses the glue particle as a basis. The glue particle approach is on one hand the data structure along the simulation task or workflow in a standardized structure where the artefacts or elements are linked. The traceability is achieved along the information chain. On the other hand it supports a data format (XML-STMD) to enable the exchange of these information packages between tools or IT environments (see chapter 2.1). The elements from different artefacts are linked by adding metadata that is user defined. The metadata used in the UPSIM project is related to the CSP (see section 2.1). The data structure remains the same, but the data format changes. The glue particles are then processed with ML algorithms to support the engineers in their maintenance, proposing new glue particles and verifying the system. In the next paragraph the architecture of the traceability concept is explained in further detail.

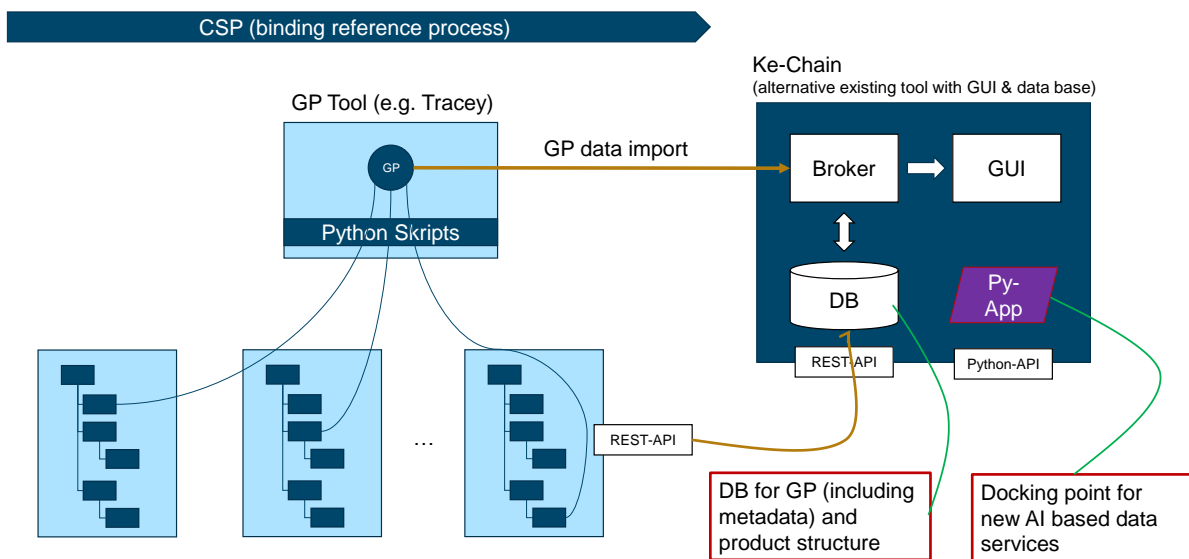
### 2.3.1 Traceability architecture

This section introduces an overview of the architecture of the traceability concept. The architecture consists of two parts, the glue particle creation part and the glue particle management part. Both parts are connected with each other through a REST-API.

The glue particle creation part consists of a tool, that references or records elements from different artifacts in different authoring tools to trace them to each other based on a simulation task. The glue particles are XML files that record or link the simulation task relevant information and links the elements in different artefacts that need to be traced along the system development process to the simulation including its execution and its results following the credible simulation process. The original artefact format is converted through scripts to the glue particle specific format. A tool (e.g., Tracey) is used to create glue particles and gather all the relevant information. These glue particles are then passed to the second part of the architecture, the glue particle management.

The glue particle management part consists of the storage, analysis and processing of glue particles. The stored data comprehend metadata and product data. This is performed by a web-platform that enables real time collaboration like the KE-chain. The broker included in this tool retrieves glue particles through an API and stores them in the internal data base. Product data from other databases can also be coupled to the glue particle management part, to get more information for their analysis. The analysis or the content of the glue particles can be displayed to the user through a graphical user interface. The resimulation capabilities of the glue particle can also be performed on this platform. This enables an additional credibility feature, as the simulation as performed by one institution can then be repeated by another institution within the supply chain to confirm the simulation results indicated in the report. KE-chain runs Python-applications on the platform, that analyze and process the glue particles. The Python applications contain ML algorithms that support the engineers in determining the verification degree of the functions, the maintenance of the glue particles, and propose new glue particles based on existing ones.

Figure 9 shows the schematic architecture of the traceability architecture. A detailed description of the Ke-Chain tool and its way to operate is described in 2.2.1.



**Figure 9: Traceability Concept Architecture**

### 2.3.2 AI based trace link Services

Within the UPSIM project, a support to the design process with artificial intelligence (AI) is suggested. The goal of AI in the project is to reduce the workload of the engineers. These are listed as follows:

- AI Application 1: Proposition of potentially missing trace links
- AI Application 2: Full system verification
- AI Application 3: Maintenance of existing glue particles

These applications are described in chapters 2.3.3, 2.3.4 and 2.3.5. As a general principle, the control of the applicability of glue particles or trace links remains with the engineers, therefore a confirmation step is always required. It is considered relevant to apply AI in the manner described above because of the amount of data that needs to be managed in development processes. Automotive OEMs and Tier-1

suppliers have to deal with more than 40.000 requirements for one single development project. For each requirement, several functions and even more simulations may be linked. For human capacities this amount of information is impossible to manage, without violation of requirement correctness and consistency.

UPSIM Project's Work Package 1 will define when on the process checks and confirmations are relevant and possible and the kind of engineers that perform them. This can also be validated with the implementation of the work package.

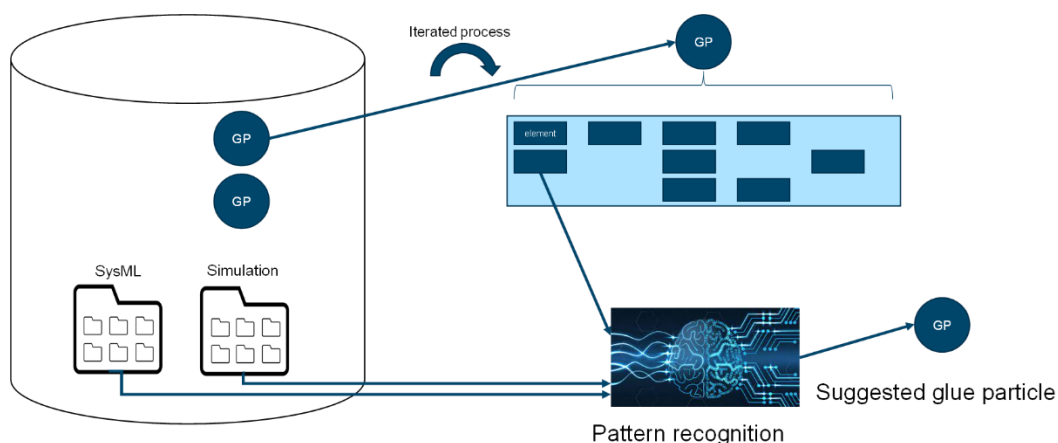
The used ML-algorithms use a supervised learning approach, that uses data generated by the engineers to learn from it, and then predicts new outcomes based on this data. All the predictions are treated as suggestions that need to be validated by the engineers.

### 2.3.3 Proposition of potentially missing trace links

The purpose of simulation is to help the engineers predict the behavior of a system or subsystems. So, there are a number of simulation tasks that need to be performed to validate the system. With a credible simulation and in the scope of the UPSIM project, readiness levels are defined to determine how credible a simulation is. The use of AI for proposing new glue particles consists in the analysis by a pattern recognition algorithm (e.g., structural algorithms) of the existing valid glue particles to detect patterns in the linked elements within the glue particle. The pattern is then applied to elements in the product data that have not been gathered in a glue particle and are similar to the already used elements. The similarity is determined either by the product structure or by the content of the elements. The algorithm collects the elements and includes them in a new incomplete glue particle that is suggested to the engineers. The engineers accept or reject this glue particle.

In case it is accepted, a new simulation can be performed to validate the contained elements in the glue particle. By determining the validation degree of the functions, this approach could also be used to complete the validation process with new simulations to existing and partially simulated functions.

A schematic overview is shown in Figure 10.

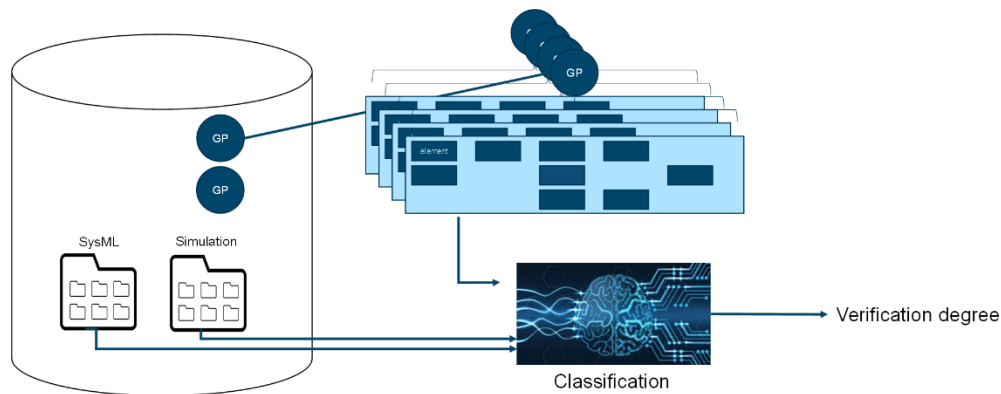


**Figure 10: Overview of Proposition of Potentially Missing Trace Links**

### 2.3.4 Full system verification

Glue particles are independent to each other. They collect the data through the process related to a simulation task. This traceability approach is simulation-oriented, so the entire system functionality in the context of the development process is not fully considered. To get a better overview of the status of the development process, an algorithm analyses the existing glue particles and determines which requirements have already been evaluated with simulation. The algorithm also analyses the elements in the Simulation and SysML models to determine the totality of existing functions created in the development process. Then a rule-based completeness check is performed, to identify the verification degree of the functions described in a SysML model. That means, there is a check to see if all functions describing a system are linked to simulations. The ahead proposal functionality may be reused in this context in order to suggest potentially missing links in the above-described manner.

This approach may be extended by using validation metrics from the credible simulation process stored in the glue particle, to determine a validation degree of the whole system. A schematic overview is shown in Figure 11.

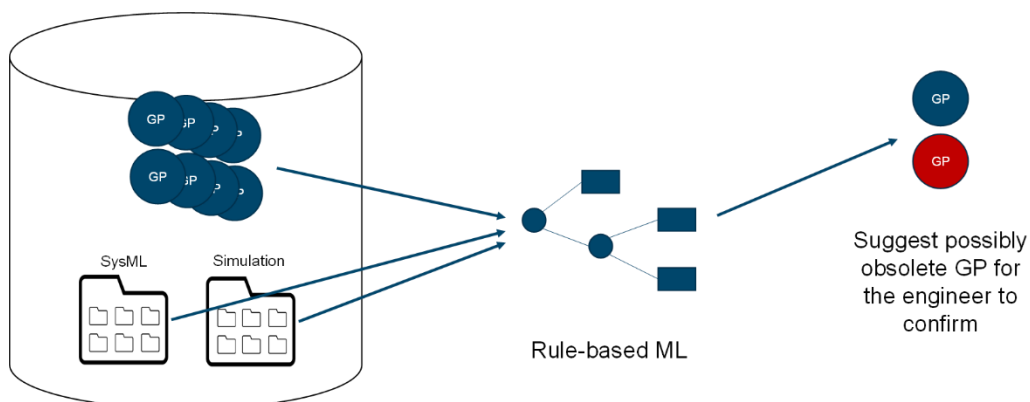


**Figure 11: Overview of Full System Verification**

### 2.3.5 Maintenance of existing glue particles

During the dynamic product development process, changes are unavoidable. This implies that work that had already been completed is no longer up to date. This leads to glue particles that include deleted or out-versioned elements and out of date or low credibility simulations. The glue particles are obsolete or not valid anymore. In addition, a simulation may not have gotten the expected result or credibility, so a new simulation with different boundary conditions, parameters, or a complete different mathematical model has been performed. This suggests that the glue particles containing these simulations and non-valid elements are obsolete and should be indicated correspondingly. It is important to remark that these obsolete glue particles can still be used as reference for simulations that have been performed previously or for out-of-date requirements, so their validity is still granted. However, these are not used for the system verification or validation, which gives them only a referential use.

A rule-based ML algorithm suggests possibly obsolete glue particles to the engineers by analyzing the structure of the glue particles and the current valid elements from the product data. This can be recognized through the metadata of these elements, including the date of creation and the version. It can also be considered to add metadata related to the valid version of the element. By following a set of rules defined by the algorithm designer, that gets optimized through the development process by the learning capability of the algorithm, taking the valid elements and comparing them with the content of glue particles, the algorithm suggests which glue particles may be out of date. The suggestion needs to be confirmed or rejected by the engineers. This loop enables the learning of the algorithm, that can adapt the rules based on the feedback given by the engineers. A schematic overview of this approach is shown in Figure 12.



**Figure 12: Overview of Maintenance of Existing Glue Particles**



## 3 Use Case: Device-tissue interaction

### 3.1 Introduction

The particular interest on device-tissue interaction is part of the medical demonstrator. This demonstrator focusses on decreasing the time to market of next generation medical devices, such as minimally invasive imaging catheters. To this end, product development should be strongly supported by virtual design and testing by means of validated high-fidelity digital twins of these devices and human tissue (e.g., arteries). Our improved understanding of the device-human tissue interaction from these interacting digital models will lead to quantified loading conditions within the device and the surrounding tissue. Consequently, relevant failure mechanisms at device and tissue level are predicted by the computational models resulting in improved performance, reliability, and patient safety.

The function-to-simulation traceability covers the simulation-related elements, that are linked from the function level through the system development process to the simulation model and the simulation results. For the specific case of the computational simulation of the interactions between a medical device and a human tissue (e.g., arteries) the simulation-related elements comprise:

1. Device and tissue models, with a focus on the tissue models,
2. The selection of a tissue model suited for implementation into a simulation code,
3. The verification and validation of the simulation model, where during verification it is checked if the theory is correctly implemented, and during validation it is determined if the simulation model correctly describes the experiments.

In the present contribution, these elements will be further specified for the carrier for device-tissue interaction selected in the UPSIM project. This carrier is the development of a physics-based simulator of guidewire and catheter insertion into a human artery system. The aim of the simulator is to prevent tissue damage induced by the guidewire/catheter by changing the design or the procedure to find an optimum between deliverability and minimal tissue damage.

### 3.2 Arterial tissue models

To simulate device-tissue interaction for the selected carrier, models for the medical device and models for the arterial tissue are required. The *global* models for the medical device (e.g., guidewire) are assumed to be reasonably straightforward from an engineering point of view, although the geometry, contact conditions and description of blood vessel material behavior will be challenging. Moreover, in case *local* device failure aspects are important the device behavior may still complicate the simulations. Yet, modeling techniques to describe the medical device itself will not be considered in the present document.

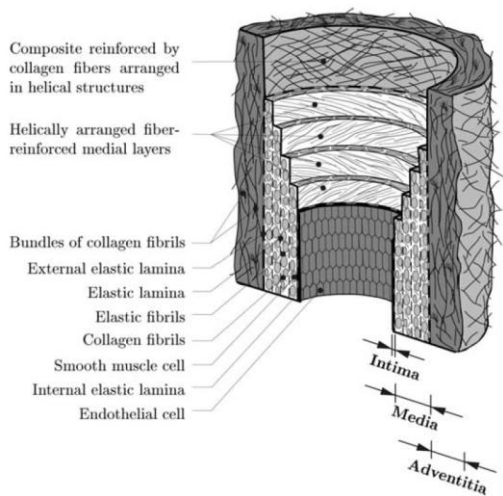
In contrast to the global device models, models describing arterial tissue behavior are much more complicated and, in most cases, not available in commercial simulation codes. To investigate device-induced tissue damage, therefore the focus is on the description of the constitutive behavior of arterial tissue. The constitutive description is based on the histology and micromechanics of arterial walls.

#### 3.2.1 Structure of the arterial wall

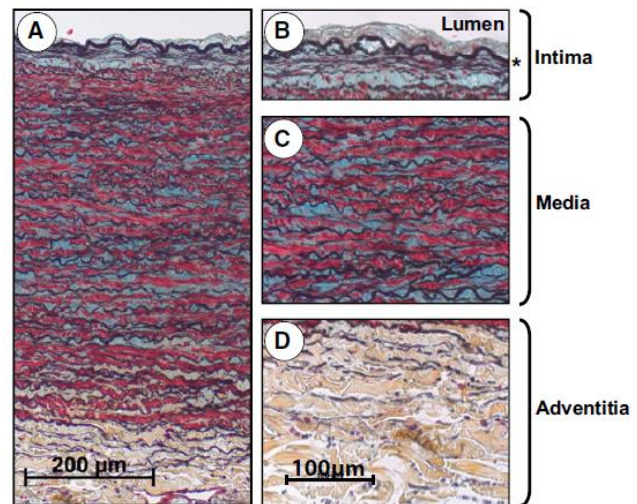
Arteries can be roughly subdivided into two types: elastic and muscular [8]. Elastic arteries have relatively large diameters and are in general located close to the heart. Muscular arteries are located at the periphery, like the femoral artery in the upper leg. For now, the focus is on the aortic wall (elastic type), but the simulation framework will also be applicable to the muscular artery type. Arteries are comprised of multiple layers with different builds. Each layer consists of a ground substance, elastin and collagen fibers, and smooth muscle cells (SMCs) with different properties and orientations. From the inside to the outside of the wall, 3 layers are distinguished: the intima, the media and the adventitia. A schematic of an arterial wall is shown in Figure 13.

A microscopic overview of an abdominal aortic wall is shown in Figure 14. The intima consists of a single layer of endothelial cells, that for healthy young individuals does not contribute to the mechanical properties of the arterial wall. This may change with age as the intima tends to stiffen. The media mainly consists of elastin (dark blue/black) and SMCs (red), while the adventitia mainly consists of collagen (brown). Also present are proteoglycans, that interlink collagen fibrils [9], and are particularly visible as light blue areas in the media. According to [9], data indicates minimal contribution of

proteoglycans to the tensile properties of arteries. As can be observed in the left section of Figure 14, the layer constituents described above gradually vary near the interfaces, so therefore each layer is a mixture of all.



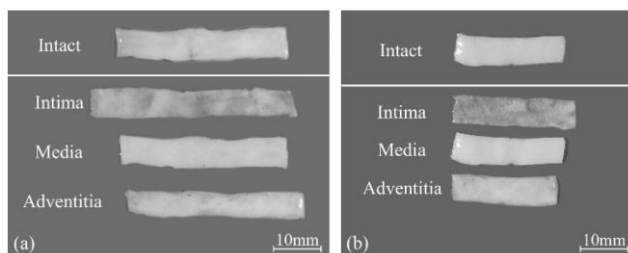
**Figure 13: Schematic of the arterial wall (from [8])**



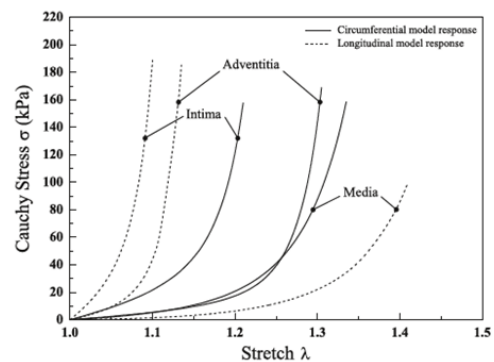
**Figure 14: Microscopic overview of the abdominal aortic wall (from [10])**

### 3.2.2 Mechanical behavior of the arterial wall

Proper mechanical characterization of the arterial wall requires separation into intima, media and adventitia [11]. An example of this is shown in Figure 15. Note that the samples shown were cut in axial and circumferential direction. The length difference from the intact specimen indicates that in each layer residual stress may be present. The mean mechanical model response (13 human samples) in tension clearly illustrates the anisotropic behavior of each individual layer and the difference between the layers (Figure 16).



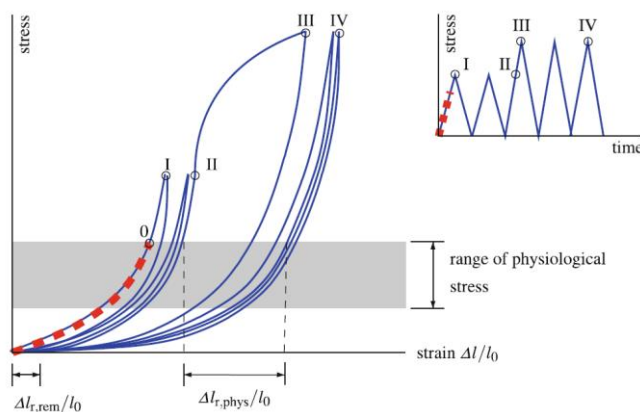
**Figure 15: Separation of fresh human aortic specimen (from [11])**



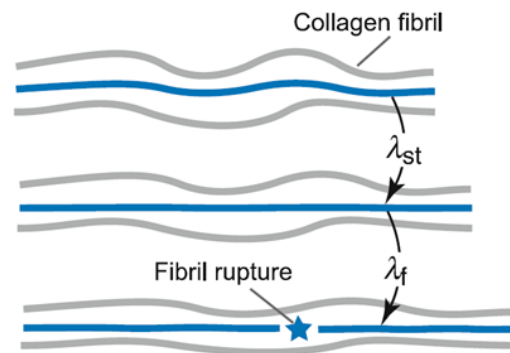
**Figure 16: Stress-stretch model response based on mean mechanical data from 13 non-stenotic human left anterior descending coronary arteries (from [11]).**

Tensile testing of arterial walls using loading/unloading cycles reveals different response regimes and hysteresis (Figure 17). Cyclic loading and unloading up to point I gives slight hysteresis and a nearly elastic pre-conditioned sample (without hysteresis) in point II. Loading beyond this domain up to point III (supraphysiological) gives stress softening and a lower unloading curve (hysteresis). Inelastic deformations (a remaining strain  $\Delta l_{r,rem}/l_0$ ) occur at full unloading after point III or may be present in the physiological loading regime ( $\Delta l_{r,phys}/l_0$ ) where they are dominated by the stress softening response of the tissue. Also indicated in Figure 17 is the range of normal physiological loading, where the response

will be nearly elastic, and a dashed red curve that indicates the initial response up to the maximum physiological stress.



**Figure 17: Schematic of mechanical response of soft biological tissues under cyclic supraphysiological loading (from [10])**



**Figure 18: Schematic of undulated collagen fibrils that gradually straighten and rupture during deformation (from [9])**

The observed hysteresis in arterial mechanical behavior is characterized by the fact that loading and unloading stress-strain paths will remain unchanged until a previous maximum strain is exceeded. This is very similar to the Mullins effect in filled rubbers [12], which is attributed to internal damage mechanisms occurring during loading. For the case of an arterial wall such a damage mechanism may be collagen fiber rupture, shown in Figure 18.

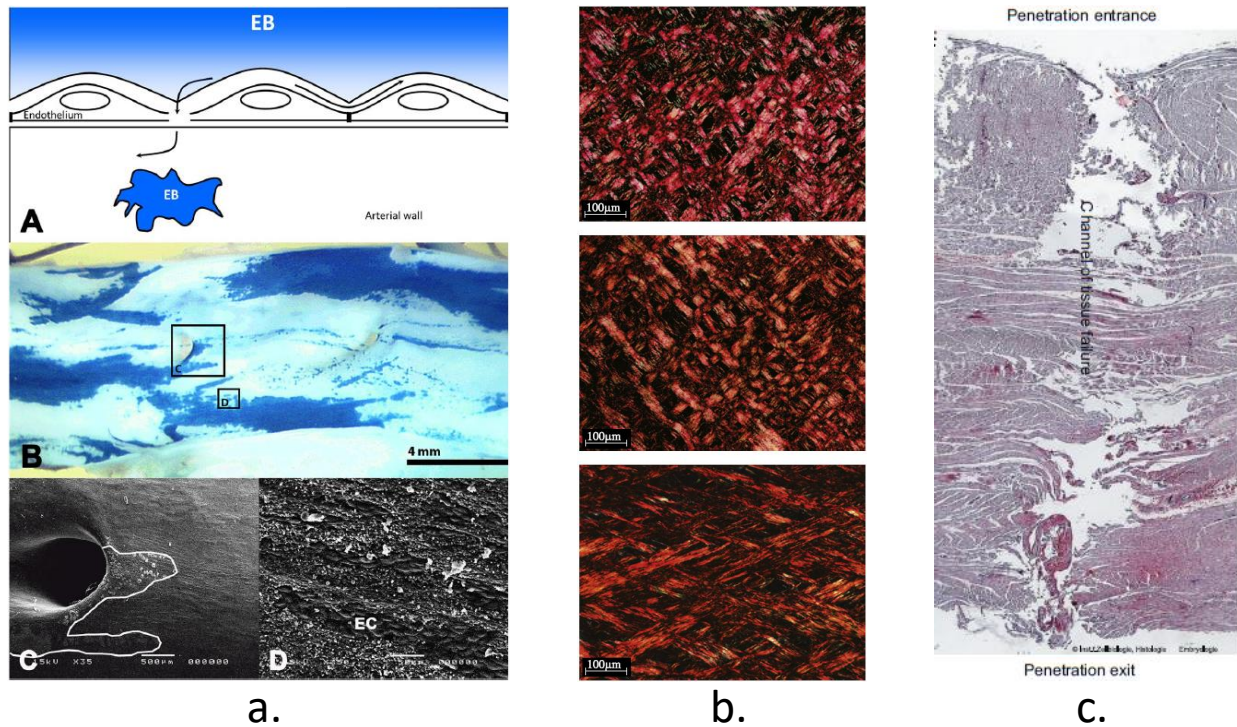
### 3.2.3 Damage in blood vessels

If damage is defined as “injury or harm that reduces value or usefulness” [13], then damage can be quantified by assessing this reduction in value or usefulness. A first step in damage quantification is the visualization of damage. In this document, three types of arterial damage that are of interest in the context of device induced arterial damage are distinguished:

1. Dead cells
2. Ruptured fibrils
3. Puncture or dissection

Dead cells can be visualized under a microscope using staining techniques, e.g., using Evans Blue (EB) to detect dead endothelial cells at the inner wall of a blood vessel [14]. It is noted that in living organisms, dead cells are usually replaced by living cells after some time. Ruptured collagen fibrils can be visualized under a microscope using Picrosirius Red (PR) stains and the collagen birefringent properties under polarized light [15]. Puncture or dissection can be visualized by preparing ultrathin slices of tissue using a microtome and staining or preparing these samples for light microscopy or Scanning Electron Microscopy (SEM) [16]. From this study it is also concluded that the detection of an object puncturing a vessel wall is quite straightforward if applied force and displacement are measured during the test. Examples of these types of damage are shown in Figure 19.

From this brief overview it is concluded that to assess device-induced tissue damage, puncture or penetration may be the easiest. This is because it does not require an advanced visualization technique like histology and staining, that at present is not available at one of the consortium partners. A test set-up to measure force and displacement during testing of arterial tissue is sufficient to detect puncture [16], and this has been developed in the project. The Evans Blue staining technique to visualize endothelial damage seems feasible to achieve and is currently under investigation, but it is doubtful if this is a type of device-induced tissue damage that can be avoided during surgery. Furthermore, it seems to only have a minor influence on the patient’s health, since the artery recovers quickly from the damaged endothelial cells [14]. The PR staining method to assess collagen fiber damage in the ventricular wall seems the most interesting, also to study the micromechanics of arterial tissue fibril deformation. However, it is also the most cumbersome, and damaged fibrils have not been observed [15], even at stretch levels in the supraphysiological range.



**Figure 19: Examples of tissue damage: a. Damaged endothelial cells (turned blue in B) caused by an injury inflicted by a medical device (like a guidewire, stent or IVUS catheter). SEM shows a damaged area with endothelial cell loss in C and an undamaged area in D (from [14]); b. Collagen fibers with PR staining at 1.05 stretch (top), 1.35 stretch (middle) and 1.50 stretch (bottom). Damaged fibrils cannot be distinguished (from [15]); c. Histological section of a porcine ventricular wall after it has been punctured by a flat-bottomed cylindrical punch (from [16]).**

### 3.3 Constitutive model of the arterial wall

#### 3.3.1 General formulation

From a constitutive modelling point of view, arterial tissues belong to the class of hyperelastic materials [8]. This means that the constitutive relation between a stress and a strain measure can be derived from a strain energy density function  $W$ . Here, the stress measure is defined as  $\mathbf{S}$  (invariant second Piola Kirchhoff stress tensor) and the strain measure is defined as  $\mathbf{E}$  (invariant Green-Lagrange strain tensor). The hyperelastic constitutive relation then takes the general form

$$\mathbf{S} = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}}$$

The total strain energy density is split into a hydrostatic  $W^{\text{vol}}$  and a deviatoric part  $W^{\text{dev}}$ . The deviatoric part is composed of the contributions of all the constituents of the arterial wall. Then, the expression for the total strain energy density becomes ([17], [13])

$$W = W^{\text{vol}} + W^{\text{dev}} = W^{\text{vol}} + W^{\text{gr}} + W^{\text{el}} + W^{\text{SMC}} + W^{\text{fib1}} + W^{\text{fib2}}$$

$W^{\text{gr}}$  is the contribution of a ground substance, which includes the proteoglycans,  $W^{\text{el}}$  is the elastin contribution,  $W^{\text{SMC}}$  is a contribution of the Smooth Muscle Cells, and  $W^{\text{fib1}}$  and  $W^{\text{fib2}}$  represent two families of collagen fibers that can be distinguished by their orientation direction with respect to the axial direction of the artery. Each of the strain energy densities are formulated based on assumptions of the deformation of underlying fibril and layer microstructure [8], which may vary per paper and research group. The references in this section only give a limited overview. It should be realized that the more strain energy density contributions are included, the more characterization experiments are required. Since the material concerned is biological tissue with a large spread in properties, one can imagine that this is not straightforward, even more when it is realized that each of the 3 arterial wall layers need to be characterized individually. Examples of characterization efforts can be found in [18], [19], [20].

### 3.3.2 Formulation including damage

To capture the experimentally observed hysteresis during loading / unloading cycles of a blood vessel in tension, internal damage mechanisms must be included in the constitutive description. A survey of damage models for soft tissues is given in [12]. For biological tissues, damage can be included in each of the deviatoric contributions to the strain energy density. For each contribution  $i$ , a continuous damage parameter  $D^i$  is defined with a value between 0 (no damage) and 1 (complete damage). The evolution of the damage parameter should be such that energy consumed by the damage process is irrecoverable. Furthermore, it should also keep track of the entire deformation history so that damage proceeds from a previously reached level, as indicated by the loading / unloading experiments. If damage evolution is accounted for in each of the contributions to the deviatoric strain energy density function this complicates the constitutive description and material parameter quantification. Often, damage evolution in certain arterial constituents can be ignored, and for instance the two collagen fiber families can be assumed to develop damage according to the same equation [15] [17]. This may simplify the description of the constitutive behavior of arterial tissue significantly.

A constitutive model of arterial tissue including damage is developed in [15], with material properties determined for each layer of human thoracic and abdominal aortas. This model combines the contribution the ground substance, the elastin component and SMCs as a single matrix contribution to the deviatoric strain energy density,  $W^m$ . The deviatoric part of the strain energy density using a classical damage concept may be expressed as [13]

$$W^{\text{dev}} = (1 - D^m)W^m + (1 - D^{\text{fib1}})W^{\text{fib1}} + (1 - D^{\text{fib2}})W^{\text{fib2}}$$

The damage evolution for each constituent is defined according to the same relation: it depends on the difference between the maximum obtained strain energy density during the deformation history and the current undamaged strain energy density. In addition, two damage related material parameters are included per contribution. From the material parameter fits of each layer in 14 thoracic and 9 abdominal human aortas it was found that damage evolution in the matrix can be ignored, and that the two families of collagen fibers develop damage with nearly identical damage related material parameters. Furthermore, satisfactory model fits were also obtained by testing samples of intact vessel walls, so not considering three individual layers. This was confirmed in [21] for thoracic and abdominal aortic aneurysmal tissues. This simplifies the constitutive description and modeling effort significantly. In addition, different sets of material parameters are available for this model formulation. This motivates its use (as described in [15] and [21]) to describe the arterial wall for the simulation of device – tissue interaction.

### 3.3.3 Implementation into a Finite Element Code

Most commercial Finite Element Codes now have an implementation available for biological tissue without damage, similar to the model description in [8] and [18] (so including the behavior of fibrous networks). The inclusion of a damage parameter in the model formulation, however, should be done using user supplied subroutines. The user must provide the stress increment and incremental stiffness matrix to the finite element code at a given strain increment.

## 3.4 Credible simulation process

Verification, validation, and uncertainty quantification (VVUQ) are seen as crucial in the computational modeling of medical devices including their interaction with parts of the human body by the American Society of Mechanical Engineering (ASME)<sup>6</sup>. Importantly, from a *regulatory perspective*, the U.S. Food and Drug Administration (FDA) has recognized the relevance of this topic. Consequently, the FDA published a guidance document on the credibility of computational simulations<sup>7</sup>.

The verification of the user implementation is an important stage of model development and contributes to a credible simulation process. If the model implementation is verified, the model should be validated by comparing simulation results to dedicated experiments. In this section, verification and validation activities are outlined for the UPSIM project.

<sup>6</sup> <https://www.asme.org/codes-standards/find-codes-standards/v-v-40-assessing-credibility-computational-modeling-verification-validation-application-medical-devices>

<sup>7</sup> <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/assessing-credibility-computational-modeling-and-simulation-medical-device-submissions>

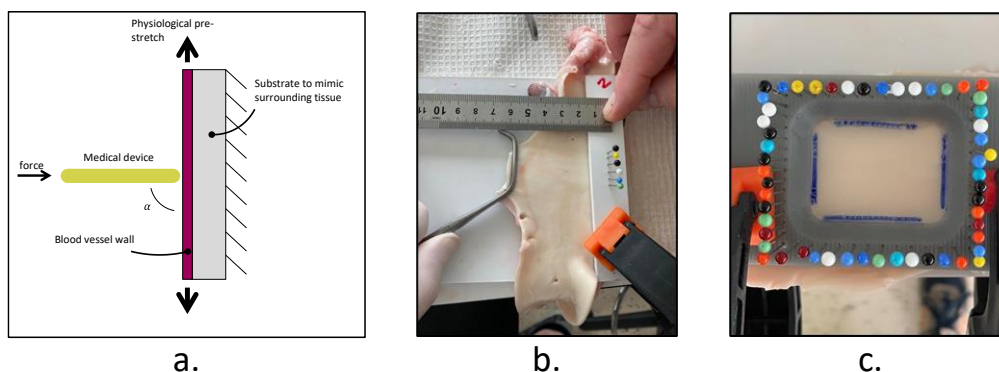
### 3.4.1 Verification

To verify the implementation of the constitutive model including damage in the FE code, the following activities are foreseen:

- Compare uniform deformation states in the FE code with (semi-analytical) solutions, e.g., using Matlab scripts.
- Check for quadratic convergence of the solution process to see if the consistent stiffness matrix is correctly implemented.
- Check on convergence of the calculated degrees of freedom upon simulation time step refinement
- Check on convergence of the calculated state variables upon mesh refinement and element technology (e.g., linear – quadratic elements, full – reduced integration)
- convergence to see if the incremental stiffness matrix is correctly implemented.
- Compare inhomogeneous simulation results to inhomogeneous simulation results reported in the literature like biaxial stretching of a perforated tissue [22], balloon angioplasty [22], or arterial clamping [23]. For this verification care should be taken that the models used in the literature are based on the same constitutive relations. Alternatively, a literature survey can be performed to find papers with inhomogeneous simulation results that are based on the model and parameters derived in [15] or [21].

### 3.4.2 Validation

To validate the selected models with respect to the ability to describe experimentally observed biological tissue behavior, a set-up is being developed (Figure 20a). The idea is to deliberately induce damage with a medical device on a flat section of animal aortic tissue (Figure 20b and c) while measuring force and displacement. An attempt will be made to measure the amount of damage, e.g., using staining techniques and microscopy, and relate this to the value of the calculated damage parameter in the simulation.



**Figure 20: a. Schematic of the experimental set-up to damage animal arterial tissue; b. Sample preparation including pre-stretch; c. Clamping device of arterial tissue**

Using the same set-up, also puncture of animal arterial tissue will be studied. The force at which penetration occurs is a threshold that should not be exceeded during the simulation of catheter insertion at any stage at any point of the arterial wall.

## 3.5 Discussion

In the present document the model selection, implementation and verification / validation for a credible simulation process of a medical device- aortic tissue interaction have been outlined. An important aspect is which type of tissue damage should be studied: some minor tissue damage may lead to cell death, but these cells are replaced by biological processes and may therefore not be fatal. Another point of concern is that the damage parameter included in the biological tissue models may not bear a physical relation to microstructural damage processes, like collagen fibril rupture. This has been pointed out by several research groups ( [15], [12], [9], [24]). So there is no clear definition of what vascular tissue damage is, and conventional indicators of mechanical injury (like visible failure or loss of stiffness) may not adequately identify the tissue's tolerance to failure [9]). For soft tissues, the link between damage

variables and crack propagation occurring in puncture or dissection has not been established [12], and is a topic for investigation.

Puncture of an arterial wall, however, is a severe type of damage, that can be detected both from a discontinuity in the force-displacement curve and by microscopic inspection of the vessel wall [25]. The force at which this occurs is a threshold not to exceed during the simulation. Furthermore, this force will depend on the proper description of the arterial wall constitutive behavior, that may still require a damage formulation. Simulations of the process of a device penetrating the arterial wall may require fracture mechanics-based simulation tools, like cohesive zone formulation [25] or XFEM [26].

## 4 Use Case: Soil – Machine interaction

### 4.1 Introduction

The function-to-simulation traceability covers the simulation-related elements, that are linked from the function level through the system development process to the simulation model and the simulation results. For the specific case of soil-machine interaction the simulation-related elements comprise

1. soil and machine hydraulic models, with a focus on the soil models,
2. The FMU preparation of soil and hydraulics models so that they may be used in INTO-CPS and Maestro framework in conjunction with collected real time sensory data from the machine,
3. The verification and validation of the simulation model, where during verification it is checked if the theory is correctly implemented, and during validation it is determined if the simulation model correctly describes the experiments.

In the present contribution, these first elements will be further specified for the example of a seeding machine robot in miniature selected in the UPSIM project. The aim of the simulation is increasing the steering precision of the field robot. The soil-machine interaction increases the complexity of the steering, as the movement of the soil and the robot differs greatly when compared to the field robot driving on pavement. Farmers require a high level of steering precision when the robot and implement is operating in the field. The use of the credible simulation process enhances the conventional simulation by supporting the validation of the robot. By following this process, data from previous simulations that is structured on a specific way can be analyzed correspondingly and perfected for future simulations. This allows to add new conditions and parameters, such as changes in the simulation model. By comparing the results of these improved simulations to test results, the simulation models get a higher credibility, enabling eventually the validation of the robot by simulations alone.

### 4.2 Soil Model



**Figure 21: Graphic of the dynamics between the soil, machine and tool [27].**

To be able to model soil using physics-based modelling, a surrogate modeling approach was used using soil-tool interactions to be able to define the parameters. Using this method, the DEM (Discrete Element Method)-engineering parameters for the three main soil types, sandy loam, sand and sandy clay were investigated from share and blade tests in soil based on the angle of internal friction, cohesion and unit weight [28]. Four sets of DEM parameters were calculated and used in simulation.

Chrono was used to perform the simulations and simulate the interaction between soil and the machine. Experimental setups were simulated using three test configurations: a 0.10m vertical blade, tested in four depths, a 0.12m angled blade tested in four depths and a 0.05m blade tested in a single depth at four rake angles.



**Table 1: Simulation parameters for sandy loam, sand and sandy clay soils**

Property	Symbol	Range	Sandy loam	Sand	Sandy clay	Sandy loam
Particle density ( $\text{kg m}^{-3}$ )	$\rho_p$	2000-3200	2424.9	2579.3	2208.6	2332.2
Young's modulus (kPa)	$E$	$1 \cdot 10^6 - 1 \cdot 10^7$	$8.0 \cdot 10^6$	$1 \cdot 10^7$	$9.5 \cdot 10^6$	$1 \cdot 10^7$
Poisson's ratio (-)	$\nu$	0-0.5	0.25	0.10	0.10	0.10
Adhesion (N)	$C$	0-5	0.70	0.20	1.40	3.50
Coefficient of restitution, soil (-)	COR	0-1	0.84	0.90	0.58	0.20
Coefficient of static friction (-)	$\mu_s$	0-2	1.3	1.06	1.2	1.20
Coefficient of kinetic friction (-)	$\mu_k$	0-2	0.72	0.64	1.06	0.78
Particle diameter (m)	$d_p$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$
Velocity (m/s)	$v_{\text{blade}}/v_{\text{tool}}$	-	1.00	0.48	0.56	0.84
Number of bodies	$N_p$	1981	193160	317780	193160	317780



**Figure 22: Example of one of the simulated tests in sand with 317780 bodies using a blade at 0,1 m depth.**

Simulations and tests indicated that the physics-based soil model was able to differentiate between the three soil types. A series of soil bin tests with shares and blades, pull bar tests and simulations have been performed to calibrate the physics-based soil model.

### 4.3 Conclusion and next steps

AgroIntelli completed two soil models at two levels of fidelity. Chrono provides an open-source soil model directly available within its simulation program and AgroIntelli has created its own physics-based soil model. AgroIntelli's physics-based soil model requires significant computational resources and cannot be run in real time, but likely gives a better result than Chrono's model.

Combining the agriculture use case with the two soil models, AgroIntelli will be able to test the effect of the robot's steering with the two different soil models. The two models will also give AgroIntelli a chance to explore the machine/tool and soil interaction and how closely we can replicate what is happening in the field in the simulation environment. In addition, AgroIntelli will be able to explore how the soil impacts the robot's steering performance when the soil parameters are changed.

## 5 Conclusions and Outlook

This document presented the function-to-simulation traceability concept and two use cases that are analyzed in the UPSIM Project. The literature research and the state-of-the-art analysis indicate, that traceability is an important topic within system development. The use of the glue particles automates the gathering of information related to the credible simulation process facilitating the system developers work.

The use of AI facilitates further the system developers work by generating new possible glue particles, maintaining existing glue particles and determining the full system verification. The potential use of AI algorithms is justified by allowing the analysis of multiple features to be processed for predicting outcomes and being able to improve the algorithm by itself during the system development process in comparison to heuristic algorithms.

The use of existing tools coupled together like Tracey and KE-chain to give a better functionality of the overall concept is also justified since it enables a combined functionality that improves both tools.

The traceability concept also fulfills the classification and assessments analyzed in morphological schema.

The next steps include the deeper analysis of the ML algorithms to indicate which elements exactly are analyzed by them to predict the expected outcomes. Also, the coupling of the tools Tracey and KE-Chain and the implementation of the ML-algorithms within the tools. Afterwards, a testing of the whole traceability concept with the presented use cases.

## 6 References

- [1] H. Hick, M. Bajzek and C. Faustmann, "Definition of a system model for model-based development," *SN Applied Sciences*, 23 August 2019.
- [2] G. Beier, *Verwendung von Traceability-Modellen zur Unterstützung der Entwicklung technischer Systeme*, Stuttgart: Fraunhofer-Verlag, 2004.
- [3] M. Edwards and S. L. Howell, *A Methodology for Systems Requirements Specification and Traceability for Large Real Time Complex Systems*, Virginia: NAVAL SURFACE WARFARE CENTER SILVER SPRING MD, 1992.
- [4] G. Spanoudakis and A. Zisman, "Software Traceability: A Roadmap," *Handbook of Software Engineering and Knowledge Engineering*, 08 2005.
- [5] H.-M. Heinkel and K. Steinkirchner, "Credible Simulation Process," 02 07 2021. [Online]. Available: <https://setlevel.de/assets/forschungsergebnisse/Credible-Simulation-Process.pdf>. [Accessed 15 02 2022].
- [6] M. Maurer, *Structural Awareness in Complex Product Design*, München: Technische Universität München, 2007.
- [7] S. F. Königs, G. Beier, A. Figge and R. Stark, "Traceability in Systems Engineering – Review of industrial practices, state-of-the-art technologies and new research solutions," *Advanced Engineering Informatics*, pp. 924-940, 2012.
- [8] G. A. Holzapfel, T. C. Gasser and R. W. Ogden, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models," *Journal of Elasticity*, vol. 61, pp. 1-48, 2000.
- [9] T. C. Gasser, "Damage and Failure of the Vascular Wall," in *Encyclopedia of Continuum Mechanics*, H. Altenbach and A. Öchsner, Eds., Springer, Berlin, Heidelberg, 2020, pp. 551-562.
- [10] J. W. Beenakker, B. A. Ashcroft, J. H. Lindeman and T. H. Oosterkamp, "Mechanical properties of the extracellular matrix of the aorta studied by enzymatic treatments," *Biophysical journal*, vol. 102, no. 8, pp. 1731-1737, 2012.
- [11] G. A. Holzapfel, "Collagen in Arterial Walls: Biomechanical Aspects," in *Collagen: Structure and Mechanics*, P. Fratzl, Ed., Boston, MA, Springer, 2008, pp. 285-324.
- [12] W. Li, "Damage Models for Soft Tissues: A Survey," *Journal of Medical and Biological Engineering*, vol. 36, no. 3, pp. 285-307, 2016.
- [13] N. Famaey, E. Kuhl, G. A. Holzapfel and J. Vander Sloten, "Cardiovascular Tissue Damage: An Experimental and Computational Framework," in *Computer Models in Biomechanics*, G. A. Holzapfel and E. Kuhl, Eds., Springer, Dordrecht, 2013, pp. 129-148.
- [14] A. Autar, A. Taha, R. van Duin, I. Krabbendam-Peters, D. J. Duncker, F. Zijlstra and H. M. van Beusekom, "Endovascular procedures cause transient endothelial injury but do not disrupt mature neointima in Drug Eluting Stents," *Scientific Reports*, vol. 10, no. 1, p. 2173, 2020.
- [15] H. Weisbecker, D. M. Pierce, P. Regitnig and G. A. Holzapfel, "Layer-specific damage experiments and modeling of human thoracic and abdominal aortas with non-atherosclerotic intimal thickening," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 12, pp. 93-106, 2012.
- [16] T. C. Gasser, P. Gudmundson and G. Dohr, "Failure mechanisms of ventricular tissue due to deep penetration," *Journal of Biomechanics*, vol. 42, no. 5, pp. 626-633, 2009.
- [17] E. Anttila, D. Balzani, A. Desyatova, P. Deegan, J. MacTaggart and A. Kamenskiy, "Mechanical damage characterization in human femoropopliteal arteries of different ages," *Acta Biomaterialia*, vol. 90, pp. 225-240, 2019.
- [18] T. C. Gasser, R. W. Ogden and G. A. Holzapfel, "Hyperelastic modelling of arterial layers with distributed collagen fibre orientations," *Journal of the Royal Society Interface*, vol. 3, pp. 15-35, 2006.
- [19] A. García, E. Peña, A. Laborda, F. Lostalé, M. A. De Gregorio, M. Doblaré and M. A. Martínez, "Experimental study and constitutive modelling of the passive mechanical properties of the porcine carotid artery and its relation to histological analysis: Implications in animal cardiovascular device trials," *Medical Engineering & Physics*, vol. 33, no. 6, pp. 665-676, 2011.
- [20] N. Famaey, G. Sommer, J. Vander Sloten and G. A. Holzapfel, "Arterial clamping: Finite element simulation and in vivo validation," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 12, pp. 107-118, 2012.

- [21] D. M. Pierce, F. Maier, H. Weisbecker, C. Viertler, P. Verbrugghe, N. Famaey, I. Fourneau, P. Herijgers and G. A. Holzapfel, "Human thoracic and abdominal aortic aneurysmal tissues: Damage experiments, statistical analysis and constitutive modeling," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 41, pp. 92-107, 2015.
- [22] V. Alastrué, J. F. Rodríguez, B. Calvo and M. Doblaré, "Structural damage models for fibrous biological soft tissues," *International Journal of Solids and Structures*, vol. 44, no. 18-19, pp. 5894-5911, 2007.
- [23] B. Fereidoonzhad, R. Naghdabadi and G. A. Holzapfel, "Stress softening and permanent deformation in human aortas: Continuum and computational modeling with application to arterial clamping," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 61, pp. 600-616, 2016.
- [24] D. Balzani, "Damage in Soft Biological Tissues," in *Encyclopedia of Continuum Mechanics*, H. Altenbach and A. Öchsner, Eds., Springer, Berlin, Heidelberg, 2020, pp. 562-576.
- [25] C. Forsell and T. C. Gasser, "Numerical simulation of the failure of ventricular tissue due to deep penetration: The impact of constitutive properties," *Journal of Biomechanics*, vol. 44, no. 1, pp. 45-51, 2011.
- [26] S. N. Chittajallu, A. Richhariya, K. M. Tse and V. Chinthapenta, "A Review on Damage and Rupture Modelling for Soft Tissues," *Bioengineering*, vol. 9, no. 1, pp. 1-35, 2022.
- [27] F. Foldager, *Physics-based Modeling and Simulation of Agricultural Machine - Soil Interaction*, Aarhus: Department of Mechanical and Production Engineering, 2021.
- [28] F. Foldager, C. Jorgensen, L. Andersen, L. Munkholm, O. Green and O. Balling, "Mapping the angle of internal friction and cohesion to the input parameters of DEM-based soil using surrogate modeling," *In review at Journal of Terramechanics*, 2022.

## 7 Acknowledgment



### PROJECT PARTICIPANTS:

Virtual Vehicle Research GmbH (AT)  
Virtual Vehicle Research GmbH (DE)

Aarhus University – DK  
Agro Intelligence ApS – DK  
3D Mapping Solutions GmbH – DE  
Audi AG – DE  
Automotive Solution Center for Simulation e.V. – DE  
Infineon Technologies AG – DE  
Deutsches Zentrum für Luft- und Raumfahrt (DLR) – DE  
iCONDU GmbH – DE  
LTX Simulation GmbH – DE  
Robert Bosch GmbH – DE  
Technische Universität Berlin – DE  
University of Augsburg – DE  
Virtual Vehicle Research GmbH – DE  
Volkswagen A.G. – DE  
softwarehelden GmbH & Co. KG  
Eindhoven University of Technology – NL  
In Summa Innovation b.v. – NL  
KE-works BV – NL  
LifeTec Group BV – NL  
NLR - Royal Netherlands Aerospace Centre – NL  
Philips Electronics Nederland BV – NL  
Philips Consumer Lifestyle B.V. – NL  
Reden BV – NL  
Sioux LIME BV – NL  
Unit040 Ontwerp B.V. – NL  
University of Groningen – NL  
BEIA Consult International – RO  
Lucian Blaga University of Sibiu – RO  
The Manufacturing Research Centre (MTC) – UK

**DISCLAIMER**

This ITEA3 Project has been made possible by a financial contribution by the German Federal Ministry of Education and Research (BMBF), by the Austrian Research Promotion Agency (FFG), by the Rijksdienst voor Ondernemend Nederland (Rvo) and by the Innovation Fund Denmark (IFD). The Publication as provided reflects only the authors' view.

Every effort has been made to ensure complete and accurate information concerning this document. However, the author(s) and members of the consortium cannot be held legally responsible for any mistake in printing or faulty instructions. The authors and consortium members retrieve the right not to be responsible for the topicality, correctness, completeness or quality of the information provided. Liability claims regarding damage caused by the use of any information provided, including any kind of information that is incomplete or incorrect, will therefore be rejected. The information contained on this website is based on author's experience and on information received from the project partners.

## 8 Appendix

### 8.1 Morphological Schema

Schematic overview on and classification of traceability terminology for the use in Systems Engineering as proposed in [7].

#### 1. Data management criteria

Criteria in this section describe data management aspects of traceability approaches.

##### 1.1. Acquisition of artefacts

Regarding the ability to acquire artefacts two different philosophies exist: integrative tools which import and synchronise artefacts from existing authoring tools without extra manual work and non-integrative tools which allow for recording trace links between artefacts that are exclusively created in the traceability tool. Integrative approaches have a higher chance to improve consistency with low effort. Example: An integrative tool would allow importing and updating the requirement specification from the RMS (requirements management system) and the product structure from the PDM (product data management) in order to model dependencies between them.

##### 1.2. Artefact manipulation

Not all traceability tools allow for creating or modifying the actual artefacts. Some only read the artefacts intentionally forcing developers to manipulate data exclusively in the specialized authoring tools. A different approach would allow data manipulation within the traceability tool (e.g. change of parameter values or status information), even though this represents a new source for inconsistency. Example: The status of a requirement from the RMS can be updated in the traceability tool when the corresponding CAD (computer-aided design) validation supplied new results.

##### 1.3. Duplication of artefacts

Considered artefacts can be original data, enabling traceability modelling on the same data that is used and created in authoring tools. Alternatively copied data is created in the respective tool for traceability purposes. Applying the traceability approach to original data within other databases avoids new sources for data inconsistency but requires a powerful IT-Integration and represents a security risk since data can be directly accessed and manipulated from different locations. Example: The traceability tool holds and displays a copy of the requirement text from the RMS.

#### 2. Planning of traceability

This section lists basic, conceptual aspects of traceability approaches.

##### 2.1. Traceability schema

This criterion indicates whether the traceability meta-model of the approach determines the types of objects that can be linked with a certain type of relation. Restricting the possible trace links between certain objects will reduce flexibility of modelling but enables greater support for recording and IT-interpretation due to more formal representation. Example: The traceability tool only allows "specify"-links between requirements and components.

##### 2.2. Granularity

Trace links between engineering data can be modelled on different levels of detail. An approach supports parameter tracing if it is possible to model trace links down to the granularity of single parameters. The granularity is one of the important factors of influence in the conflict between economical modelling and detailed, comprehensive traceability. Example: In parameter tracing the X-, Y- and Z-position of the drivers eye (e.g. in mm) defined in a requirement or a conceptual model can be used as a direct input for the rear view validation.

##### 2.3. Supported types of data structures

Engineering data within artefacts is usually structured in form of lists or hierarchies. A rising degree of relations leads to polyhierarchical (one child belongs to several parents) or net structures. The data structure criterion pinpoints the different types of structures that can be imported and processed by the specific tool. Example: The door mirror specification is a list; its product structure is a hierarchy; the function structure is a poly-hierarchy; the whole rear view system is a net.

#### 3. Recording of trace links

This section addresses the question how trace links can be modelled.

##### 3.1. Modelling support

The vast effort, necessary to model traces, represents one of the major drawbacks for traceability application in practice. There are, however, different techniques to reduce this effort by providing specific modelling support. Rule based recording comprises all methods that apply rules to the artefacts in order to automatically derive and record traces between them. This approach can be further improved by the implementation of wizards that suggest traces and delimit the number of possibly linked objects. An easy solution is to simply reuse and adapt existing models which is possible for similar projects,

especially in adaptive development. Reuse can be further improved with template approaches which comprise explicit logic supporting the adaptation of generic models to specific models.

#### 4. Use of trace links

The following categories address the aspired use of the links.

##### 4.1. Change handling

Change notification allows elements (and responsible engineers) to be informed about changes that occurred to a linked object which helps to identify data inconsistency. Change propagation enhances this functionality by triggering a subsequent action on affected objects. Examples: The rear view validation (the responsible engineer) is informed about a new driver position (notification). The status of the rear view requirement is updated when the validation computes new results (propagation).

##### 4.2. Type of traceability

Two major types of traceability can be distinguished based on their information content. Qualitative traceability only allows identification of related objects whereas the more detailed support, termed quantitative traceability, makes a statement about how much a property value of a related object is affected. Quantitative traceability supplies more information to the developer but requires a higher modelling effort.

##### 4.3. Traceability purpose

Depending on the nature of a project, traceability can fulfill various purposes. Trace links can be used for verification purposes: they help to compare the as-is values of a technical system to its to-be properties. On the other hand trace links can also be deployed in order to study a system and to help understanding it. We summarise these kinds of activities where mainly trace links as such are considered (e.g. impact analysis [20] or FMEA [6]) under the term analysis. Further on, modelling trace links can be part of a creative process that actually defines system properties during systems synthesis (e.g. linking a function to a control unit). In some branches traceability is mandated by a number of standards such as DO-254, ISO 9000ff, ISO/EC 15504 (SPICE). Therefore, companies operating in that area are obliged to create a documentation of trace links. A more beneficial purpose for traceability is to exploit the modelled trace links in order to support progress monitoring for example when creating progress reports. A rather technical goal is to utilise the trace link model for synchronisation purposes between different artefacts (consistency). Examples: Traces are used to gather the current driver position for the rear view validation and to compare the results with the specification (verification). Traces are used to identify an impact from a modified mudguard on the size of the outer mirror (analysis). The traceability tool is used to define that the heating of the outer mirror is controlled by the door control module (synthesis). A trace shows that the Japanese surround view requirement is partly fulfilled by a camera solution (documentation). Traces are used to gather the status of all relevant validations for a particular gateway of the process (process monitoring).

##### 4.4. Representation of trace links

In order to use the recorded trace links to improve transparency they need to be presentable to users. For that purpose trace links can be represented in a traceability matrix, a two-dimensional grid where rows and columns contain an artefact each. Marks at the intersections represent the existence of a link between the according elements. The same content can also be represented in form of a node-link graph, where the elements of an artefact are displayed as nodes and trace links as edges. Alternatively, links can be represented as a cross-reference within the artefacts for example in form of a hyperlink.

#### 5. Properties of trace links

This section contains a further classification of trace links. Some properties of trace links are widely discussed in traceability research with no common understanding or clear definition regarding their meaning. Because of the disagreements, we propose a separation of these meanings and a new classification.

##### 5.1. Process phase context

The category which is related to the first meaning is called process phase context. It describes whether a trace link is connecting artefacts from the same (phase-internal) or from different process phases (phase-spanning). If there are several requirements specifications throughout the development process in one or several requirements management tools, trace links between them are considered phase-internal. Example: A requirement of the rear view system is derived from a requirement of the vehicle specification (phase-internal). That requirement specifies a component or function (phase-spanning).

##### 5.2. Artefact context

The second category refers to the second meaning and is called artefact context. It describes whether a trace link is connecting elements from exclusively one (artefact-internal) or more (artefact-spanning) artefacts. Example: A requirement of the rear view system is derived from a requirement of the vehicle specification (artefact-spanning). A rear view requirement refines another rear view requirement (artefact-internal).

##### 5.3. Semantics of trace links



The category “semantics of trace links” describes whether links have richer semantics or not. Since there is no common understanding in research about the necessary number of links we intentionally limit the distinction to untyped and typed links. Examples for typed links are: satisfies, justifies, describes, depends on, and validates.

#### 5.4. Storage location of trace links

Depending on the implementation of the traceability approach, trace links can either be stored separately in one central location (as separate models) or distributed in at least one of the according artefacts. Example: The trace link between the driver’s eye position and the rear view validation is either stored in one of the CAD files (distributed) or in a separate, central database (central).