standart

Session 5: Digital Thread & CHADA

FINAL DISSEMINATION EVENT
NLR Amsterdam (NL), December 17th

NLR: **Julian De Marchi** | Elise Scheers | Florian Hensel | Nikita Noskov | **Carlos Schyns**
TUD: John-Alan Pascoe | Francisco Monticeli | Davide Biagini NCC: Jon Taylor

Funded by
the European Union

Index

01 Concept

Digital Thread

Section 1

purpose

Concept

- **who**: partners in academia, industry, and technology
  - academic research: TUD, UB, UT
  - industry & application: HEX, ICO, NCC, SUZ
  - research & technology transfer: NLR
- **what**: up-to-date materials characterization models & test data
  - new models developed for composite structures with and without defects
  - over 900 tests performed
- **when**: across the span of three years
  - 2023/2024/2025
- **where**: NL, UK
  - data provided by multiple partners from different locations and in varying formats
- **why**: retain accessible data
  - to EU researchers and test labs (in both academia and industry)
- **how**: digital thread
  - raw data stored at Zenodo
  - digital thread hosted by NLR in STS

Section 1

overview

Concept

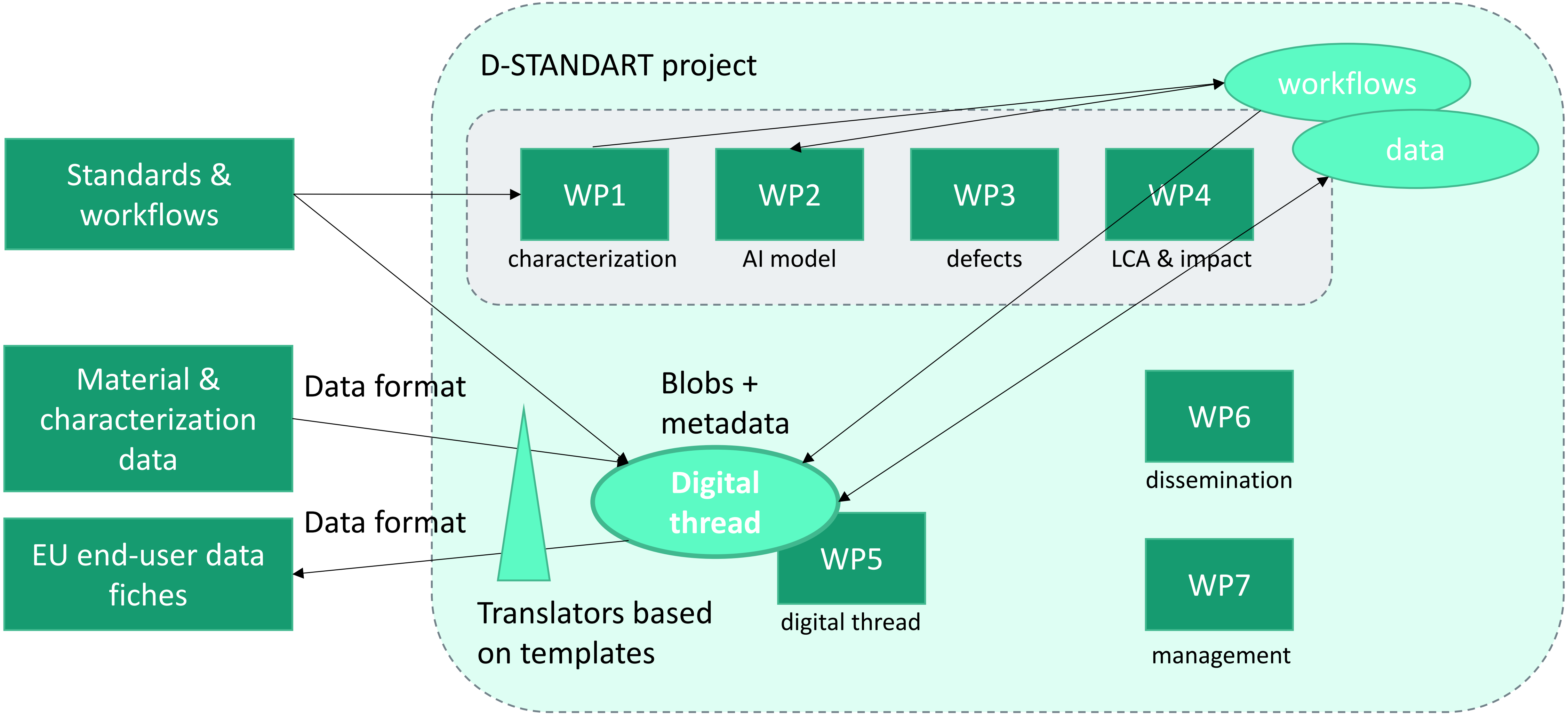
- **Objective 1**: Develop a framework for managing digital threads.
  - D5.1 (M08): Digital Thread architecture (functional specification)
  - D5.2 (M20): Digital Thread platform (implementation of the functional specification)

- **Objective 2**: Develop placeholders (templates) for digital threads to support coupling AI technologies and fatigue model.
  - D5.3 (M30): Digital Thread database (fully populated and accessible)

- **Objective 3**: Support design (CAE) software making use of digital threads.
  - D5.4 (M32): Working plugin for lifetime fatigue estimation

architecture

Concept



02

Characterization data (CHADA)

Digital Thread

CHADA

Standardization Landscape

- **EMMC**: European Materials Modelling Council
  - emmc.eu

- **CHADA**: materials characterization data
  - EU Materials Characterisation Council (EMCC: CWA17815:2025)
- **MODA**: materials modelling data
  - EU Materials Modelling Council (EMMC: CWA17284:2018)

- **EMMO**: Elementary Multi-perspective Material Ontology
  - github.com/emmo-repo/EMMO

- **CHAMEO**: characterization methodology domain ontology
  - github.com/emmo-repo/domain-characterisation-methodology
- **CHOCO**: characterization of composites ontology
  - developed by NLR, but not yet published

03 Structuring data

Digital Thread

knowledge graphs

Structuring Data

Knowledge Graphs are structured networks of entities and their relationships, designed to capture meaning and context rather than just raw data.

They transform disconnected information into actionable knowledge, enabling both humans and machines to reason, discover, and connect ideas more intelligently

Knowledge Graphs are used in real world applications such as:

- *Search Engines*
- *Virtual Assistants*
- *Enterprise Data Integration*
- *AI & Machine Learning*
- *Scientific Research*

Knowledge Graphs compared to traditional data storing techniques offer:

- *Flexibility: Schema-less design adapts to evolving data*
- *Interoperability: Connects heterogeneous datasets seamlessly*
- *Discovery: Reveals hidden connections and supports inference*
- *Human + machine understanding: Bridges natural language and structured data for richer insights.*

why knowledge graphs are more interpretable by machines

Structuring Data

Knowledge Graphs are more easily interpretable for machines because they make relationships first-class citizens.

They are directly interpretable, whereas traditional relational databases hide them behind table structures and joins.

- **Relationships are explicit:** stored as edges between nodes, not hidden in unknown foreign keys.
- **Semantics are built-in:** ontologies and vocabularies attach meaning to data, not just structure.
- **Uniform representation**: everything is expressed as triples (subject–predicate–object), easy for machines to parse.
- **Reasoning & inference:** machines can derive new facts (e.g., infer similarity, classify entities) beyond what's explicitly stored.
- **Flexible linking**: IRIs and namespaces allow global, unambiguous references across datasets.
- **Graph traversal:** machines can follow paths naturally (*Scientist → authored → Paper → studied → Material*), enabling pattern discovery.

Section 3

core building blocks underlying knowledge graphs

Structuring Data

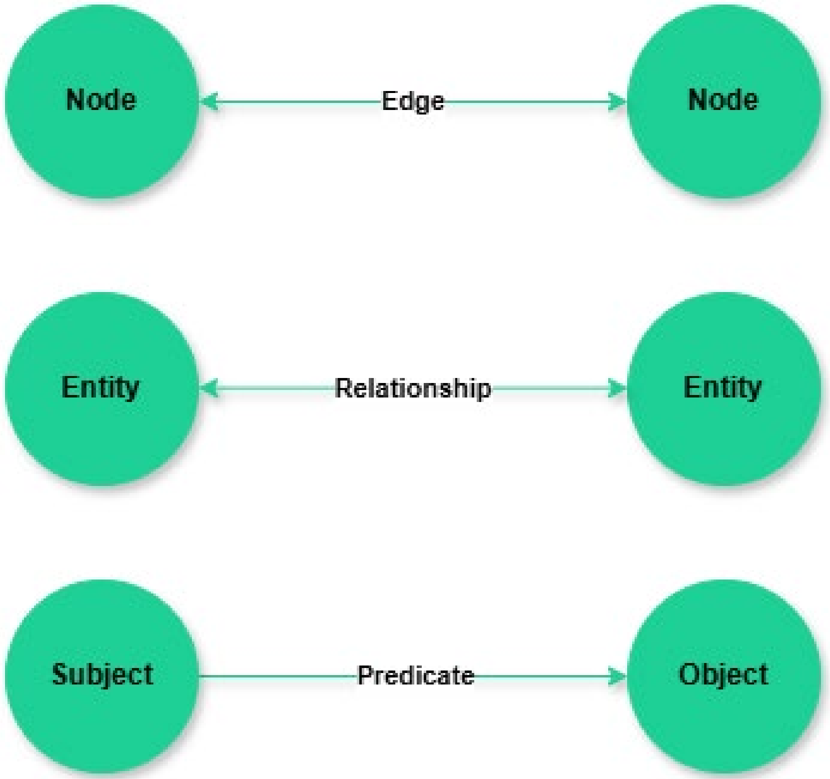
- **Ontology:** Formal model that defines classes, properties, and rules
  - Provides the semantic backbone of a Knowledge Graph
  - Enables reasoning and inference beyond raw data
- **Vocabulary:** A collection of terms (classes + properties) used to describe data
  - Examples: FOAF, schema.org
- **IRI (Internationalized Resource Identifier):** Unique identifier for entities, classes, and properties
  - Ensures global unambiguous references (e.g., http://example.org/Customer123)
- **Namespace:** A prefix mechanism to group IRIs under a common base
  - Prevents naming conflicts across vocabularies
  - Example: *ex:Customer* where ex maps to *http://example.org/*

- **RDF (Resource Description Framework):** The W3C data model for representing data as triples (subject–predicate–object). Commonly used to form the basis of a Knowledge Graph
- **SPARQL:** A query language for RDF data. Enables searching, filtering, and reasoning over Knowledge Graphs

triples

Structuring Data

- **Graph structure**
  - Data is modeled as a graph with nodes (entities) and edges (relationships)
  - E.g. *Material A → hasProperty → Hardness, John Doe→ authored → Book B*
- **Entities & Relationships**
  - Entities: things like people, materials, papers, concepts
  - Relationships: semantic links (e.g., *authored,* but also *hasProperty*)
  - Enables machines to reason over meaning, not just match data
- **Subject–Predicate–Object (SPO) triples (RDF Data Model)**
  - Subject: the entity (*Material A*)
  - Predicate: the relationship (*hasProperty*)
  - Object: the value or linked entity (*Hardness*)

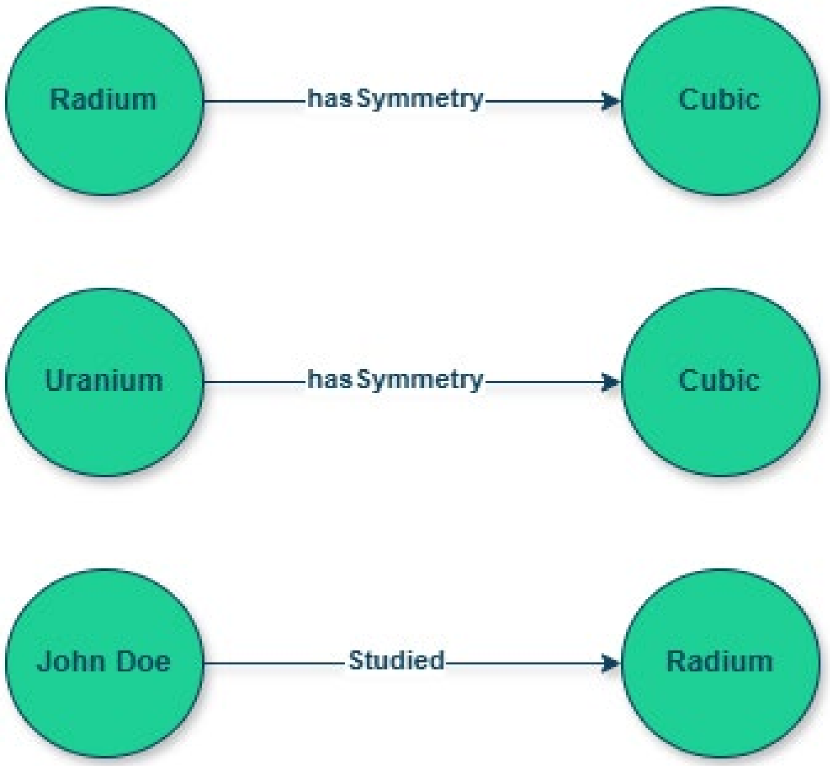


inference example

Structuring Data

Suppose: if two materials share the same crystal symmetry, they belong to the same class of materials.

- **Ontology encodes this as:**
  - Material → hasSymmetry → SymmetryType
  - **Rule:** Materials with equal SymmetryType are similar
- **Instance data**
  - *Radium → hasSymmetry → Cubic*
  - *Uranium → hasSymmetry → Cubic*
  - *John Doe → studied → Radium*
- **Machine inference**
  - From ontology: Materials with the same symmetry are similar
  - From data: Radium and Uranium both have Cubic symmetry
  - Derived fact: Radium → isSimilarTo → Uranium



Since John Doe studied Radium, the machine can infer that John Doe might also be interested in Uranium.

This fact wasn't explicitly stored — it was derived by reasoning over ontology + instance data.

Section 3

four types of triples

Structuring Data

On **Ontology Level:**

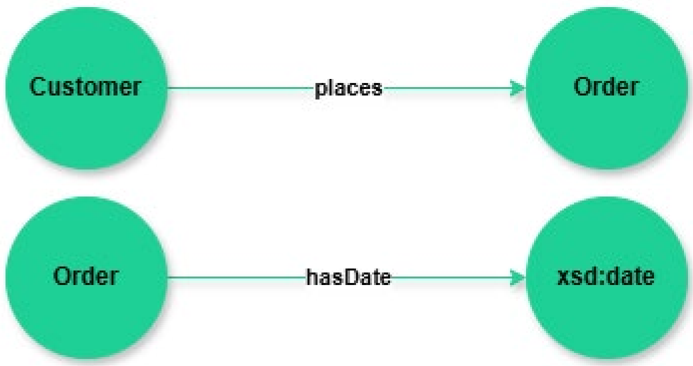
1. **Class → Class** (via object property)
2. **Class → Datatype** (via data property)

On **Instance Data Level:**

3. **Entity → Entity** (via object property)
4. **Entity → Literal** (via data property)

ONTOLOGY LEVEL

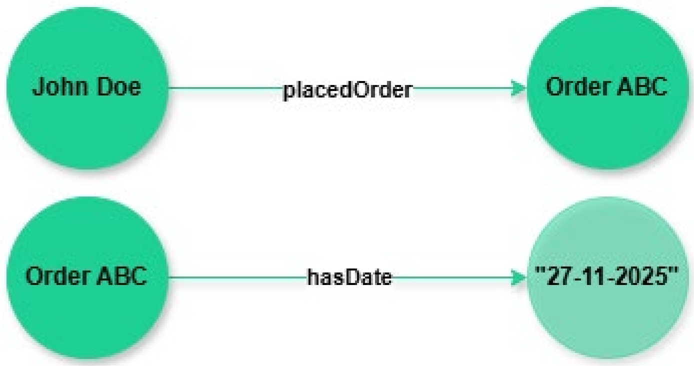
1
2



OBJECT PROPERTY

DATA PROPERTY

INSTANCE DATA



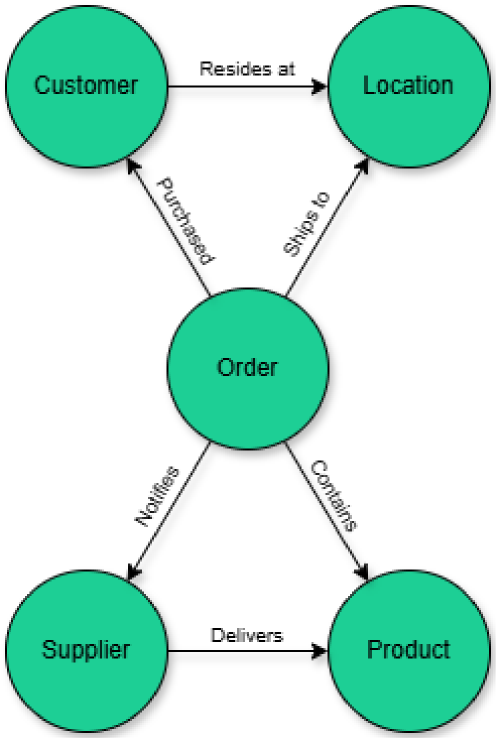
3
4

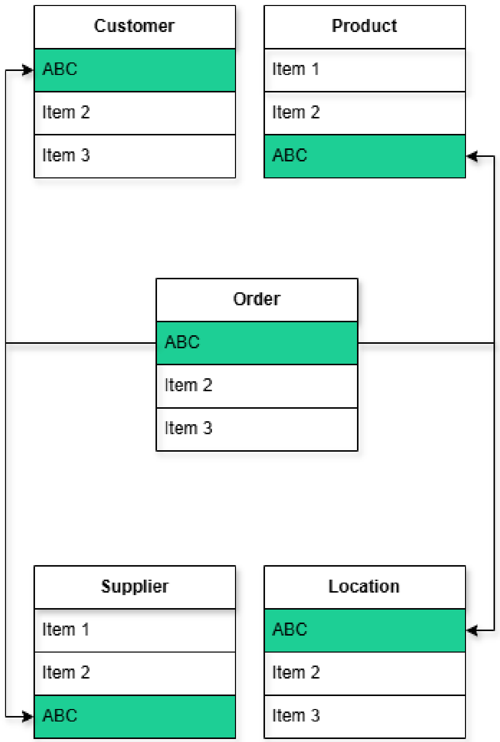
knowledge graphs vs relational DBs - architecture

Structuring Data

Knowledge Graph

- an **Ontology** defines classes, properties, and rules
- **Classes:** types of entities (e.g. *Customer*)
- **Properties:**
  - **Object properties:** relationships (e.g. *Resides at*)
  - **Data properties:** attributes (e.g. *has name*)
- **Individuals:** actual instances (e.g., "*John Doe*")





Relational DB (value-data pairs)

- A **DB Schema** defines tables, columns and constraints
- **Tables:** Represent entity types (e.g. *Customer*)
- **Columns:**
  - **Foreign Key:** relationships (e.g. *customer_id in foreign table*)
  - **Attributes:** store values (e.g. *name, price*)
- **Rows**: actual instances (e.g., "John Doe" in customer table)

knowledge graphs vs relational DBs - queries

Structuring Data

Knowledge Graph – SPARQL Query

```
PREFIX ex: <http://example.org/>
SELECT ?customer
WHERE {
        ?customer a ex:Customer .
        ?customer ex:residesAt ?location .
        ?location ex:hasName "Amsterdam" .}
```

Relational DB – SQL Query

```
SELECT c.*
FROM Customer c
JOIN Location l ON c.location_id = l.id
WHERE l.city = 'Amsterdam';
```

- **SPARQL queries are semantic, SQL queries are structural**
  - The terms in the SPARQL query are **domain-level vocabulary**, not arbitrary schema labels
  - In SQL the tables could be called differently dependent on the designers choice (Customer could be called 'cust_table' f.e.)
- **SPARQL queries traverse relationships directly, SQL requires explicit joins**
  - In SQL you must know which columns link tables and how to join them correctly
  - SPARQL you can move from Customer → Location or Location → Customer without worrying about joining tables
- **NOTE:** Convenient to do 1st line (PREFIX) in SPARQL – otherwise you'd have to write 'http://example.org/Customer' in full

summary

Structuring Data

- **Knowledge Graphs are structured networks of entities and their relationships**

- Knowledge Graphs are **more easily interpretable for machines** because they make relationships first-class citizens

- **Data are modeled as a graph (RDF)** with nodes (entities) and edges (relationships): **SPARQL the query language**
  - Subject–Predicate–Object triples

- **Ontologies** together with **Vocabularies** form the formal model that defines classes, properties, and rules
  - Provides the semantic backbone of a Knowledge Graph

During the afternoon Demo Session, links to the D-STANDART project's Specimen Test Suite software will be made

04

Key results & opportunities

Digital Thread

key results

Digital Thread

Community and dissemination:

- M10: sister project registration with EU OntoCommons (ontocommons.eu)
- M11: sister project registration with EU CIRPASS initiative (cirpassproject.eu)
- M17: start collaboration with CENELEC on CWA17815: 2021→ 2025 update for CHADA
- M23: presentation for nanoEMCommons at University of Cambridge
- M24: sister project registration with EU CIRPASS-2 initiative (CoP – community of practice)
- M25: CWA17815:2025 released
- M26: data sharing festival
- M28: presentation at EMMC conference
- M28: matCHMaker workshop
- M29: start collaboration with CENELEC on OntoWF: ontology semantics for materials characterization
- M34: expert member on CAELESTIS discussion panel
- M34: DPP & dataspaces event
- M35: CENELEC OntoWF draft submitted
- M36: (limited) availability of raw and post-processed datasets on STS
- *M37: OntoWF draft for release*
- *M38: EMMC matCHMaker collaboration on data structure standardization for materials characterization*

key results

Digital Thread

Deliverables:

- D5.1 Digital thread functional architecture
- D5.2 Digital thread collaboration platform
- D5.3 Digital thread data exchange format specifications
- D5.4 Fatigue model plugin

Results:

- CHADA for composite materials characterization
- MODA for composite materials modelling
- CENELEC CWA17815:2025 update

Knowledge:

- How to implement knowledge graphs for industry 4.0 applications
- How to implement new materials ontologies for aerospace
- How to structure data for FAIR purposes
- How to leverage D-STANDART results in other HE projects, both ongoing and upcoming

opportunities

Digital Thread

- **Partnership with SINTEF & Fraunhofer IWM (Institute for Mechanics of Materials)** :
  - form2rdf/data2rdf – Python packages that convert various forms / data formats (e.g. CSV, Excel, JSON, Python) into RDF graphs, facilitating semantic data integration and interoperability
  - DSMS: Dataspace Management System – enables the provenance and cataloguing of data through a combination of classical relational databases and semantic technologies

- **FAIR data structuring under partnership with EMMC**:
  - DCAT (W3C): Data Catalog – a vocabulary for publishing searchable data catalogs
  - QUDT (NASA): Semantic specifications for units of measure, quantity kind, dimensions and data types
  - PROV-O (W3C): Ontology to represent and interchange provenance information generated in different systems and under different contexts
  - OO-LD: Object-Oriented JSON-LD
  - MatCHMaker (EMMC): Collaboration on FAIR data standards development for materials characterization
  - CHOCO (NLR): Formal publication of characterization of composites ontology

opportunities

Digital Thread

- **EMMC**:
  - Updated CHADA (characterization data) export
  - Traceable links to CHAMEO (characterization of materials ontology)
  - RDF (W3C): Resource Description Framework – a linked-list data model for metadata to automate deep data query across datastores
  - Tripper: Uniform back end for triple stores + OTEAPI: Open Translation Environment API
  - Contact points: Gerhard Goldbeck | Pierluigi Del Nostro | Anne De Baas | Alexandra Simperler

Thank you!

Contact points for any question:

Presenter

- Julian DE MARCHI (NLR)
- Email: Julian.DeMarchi@nlr.nl
- Phone: +31 88 511 3792

Presenter

- Carlos SCHYNS (NLR)
- Email: Carlos.Schyns@nlr.nl
- Phone: +31 88 511 4486

Project Lead

- Marco NAWIJN (NLR)
- Email: Marco.Nawijn@nlr.nl
- Phone: +31 88 511 4486

Project Management Officer

- Mathieu LIONS (LUP)
- Email: Mathieu.Lions@l-up.com
- Phone: +33 7 83 28 19 82

Communication Officer

- Peggy FAVIER (LUP)
- Email: Peggy.Favier@l-up.com
- Phone: +33 6 75 64 10 78

FOLLOW US ON:

in

zenodo

d-standart.eu

standart

for Durability Modelling of Composites

Funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

UK Research
and Innovation