Creep Testing Ontology (CTO) for Constructing LLM-driven Creep Literature Knowledge Graph

To obtain the academic degree of

Master of Material Science

The Institute for Applied Materials (IAM), Karlsruhe Institute of Technology (KIT)
Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT)
FIZ Karlsruhe — Leibniz Institute for Information Infrastructure

Dissertation

by

Soheil Molaei

Date of oral examination: 24.08.2026

Supervisors:
Prof. Dr. Harald Sack
Prof. Dr. Peter Gumbsch
Dr. Hossein Beygi Nasrabadi

Abstract

Aerospace and power-generation materials operate under extreme loads and temperatures, so creep testing is vital for understanding how they will perform over time. A huge amount of creep data is available through unstructured text, tables, and figures of scientific publications, however, the lack of a standardized structure, machine-readable vocabulary, and semantic representation make exploration, compression and reuse of such creep data very challenging.

This thesis addresses this gap by developing the Creep Testing Ontology (CTO), a BFO-compliant (Basic Formal Ontology) domain ontology for uniaxial creep testing of metallic materials, aligned with ISO 204:2023. CTO covers the full processing–structure–property chain: material identity and chemical composition, manufacturing and heat-treatment history, microstructural features (grain size, precipitate characteristics), creep test conditions (temperature, stress, standard), and measured creep properties (rupture time, minimum creep rate, elongation, stress exponent, activation energy), together with full provenance metadata linked to the source publication.

The ontology is developed following OBO Foundry best practices using the Ontology Development Kit (ODK). It reuses the Platform MaterialDigital Core Ontology (PMDco). Here the key methodological aim is to establish an Ontology Design Pattern (ODP)-based development workflow: reusing existing PMDco patterns for process chains, chemical composition, and mechanical properties wherever applicable, while contributing novel ODPs (like for microstructural characterization and distinguishing planned and measured test parameters) to close gaps in the existing pattern library. This ODP-driven approach is intended to yield a modular, extensible ontology structure that can serve as a template for future mechanical-testing ontologies beyond creep. CTO is implemented in Protégé, validated through HermiT reasoning and SPARQL-based competency question testing, documented with Widoco, and publicly released under a CC BY 4.0 license with a persistent w3id.org identifier (https://w3id.org/pmd/cto/).

CTO was used as the semantic backbone for evaluating whether combining formal ontology engineering with large language model (LLM)-based information extraction and knowledge graph (KG) construction can retrieves creep metadata from scientific publications and populates a FAIR-compliant creep literature knowledge graph (CreepLitKG). This thesis reused the already developed LLM-text extraction and KG construction pipelines, and respectively evaluated them against the manually annotated benchmark publications dataset and data retrieval via SPARQL queries.

Table of contents

List of Figures

Chapter 1: Introduction

1.1. Research Data Management in Materials Science and Engineering

The digital transformation of industrial activities, driven by the fourth industrial revolution (Industry 4.0), has fundamentally changed how materials data are generated, stored, and utilized. Digitalizing production processes is expected to facilitate increased productivity, reduced manufacturing costs, energy sustainability, and carbon emission reduction [1], [2]. Within this transformation, the field of Materials Science and Engineering (MSE) is undergoing a particularly dynamic shift: automation, high-throughput methods, and data-based algorithms are revolutionizing production and characterization facilities, and national initiatives worldwide aim to achieve an integral understanding of the entire materials life cycle, from raw materials to operating components and beyond [3].

1.1.1. Key Challenges in Materials Data Management

Materials data originate from a wide range of sources including experimental measurements, computational simulations, and multiscale characterization techniques. The inherent diversity of MSE (encompassing perspectives from natural sciences such as physics, chemistry, and crystallography, along with methods from various engineering fields) presents a significant challenge to digital transformation. Each discipline contributes its unique viewpoint, specialized terminology, and distinct data culture, which naturally leads to the presence of materials data in heterogeneous formats and structures [3]. While these diverse data formats are a direct consequence of the interdisciplinary nature of MSE, they complicate communication within the MSE landscape: the variety in formats and structures, often incompatible with one another, hinders the seamless fusion of data, thereby impacting the exchange of information and knowledge [3].

The scale of the problem is substantial. Plenty of mechanical testing data are generated daily in thousands of materials test labs around the world, and most of this data cannot be utilized by external parties for technical purposes, data-driven research, or product development [4]. The reasons are multifaceted: data may not be findable because they exist only as printed papers or files in unstructured, non-publicly available repositories; even when findable, access may be restricted; and even when accessible, unreadable or image-like data formats, distinct data languages, or a lack of appropriate data structure and test metadata may make them difficult to interoperate. In the case of interoperable datasets, there are further concerns about reusability, including inadequate documentation of processing and testing parameters, inaccurate or non-standardized testing procedures, and a lack of data quality control or long-term preservation strategies [4], [5].

As a result, a high amount of data generated daily in worldwide materials testing labs does not meet the criteria of being Findable, Accessible, Interoperable, and Reusable (FAIR) [6] and therefore cannot be reused for data-driven product development purposes. It has been estimated that the non-existence of FAIR data invokes costs to the European economy of at least €10.2 billion per year [7]. In this context, the FAIR principles (which outline the characteristics that modern data sources, infrastructures, tools, and vocabularies should possess) have become the guiding framework for research data management (RDM) in MSE [3], [8].

1.1.2. Need for digital, machine-readable, AI-ready data

The need for digital, machine-readable, AI-ready data has become increasingly urgent because AI systems depend fundamentally on the quality, structure, and usability of the data on which they are trained. The poor-quality or poorly structured datasets can create downstream "data cascades," in which seemingly small problems in formatting, consistency, or bias compound into substantial errors in model performance, reliability, and trustworthiness. For this reason, data should not merely be digitized in a basic sense; they must be prepared in ways that make them technically suitable for machine learning, well documented, standards-compliant, and usable across digital workflows. In this view, AI-ready data are those that are intentionally designed for machine processing from the start, rather than being retrospectively adapted after data collection [9].

In MSE, this need is even more critical because the field generates complex, heterogeneous, and often noisy data from experiments, characterization techniques, simulations, and multiscale analyses. The materials-data integrity researches highlight that AI models in this domain are only as reliable as the data used to train them: incomplete, inconsistent, biased, or manipulated data can misdirect discovery and lead to unreliable predictions. Therefore, machine-readable materials data should include structured raw instrument files, rich metadata, clear provenance, and documentation of processing steps so that datasets are reproducible, verifiable, and suitable for automated analysis. Likewise, the AI-ready data framework stresses the importance of appropriate file formats, machine-readable metadata, attached documentation, APIs, and version control, all of which reduce friction for AI use and allow both humans and machines to interpret, validate, and reuse the data effectively. Thus, the move toward digital, machine-readable, AI-ready data is not only a matter of convenience, but a foundational requirement for trustworthy and scalable AI-driven materials research [9], [10].

1.1.3. Importance of consistent data representation

The importance of consistent data representation is especially high in MSE because the field inherently produces highly heterogeneous data. Materials data originate from experiments, simulations, characterization methods, and manufacturing processes across multiple structural scales, from the atomic to the macroscopic level, and they are generated using many different standards, models, terminologies, and file formats. As a result, data from different sources are often difficult to compare, combine, or interpret in a uniform way. The consistent data representation is therefore essential for making heterogeneous materials data understandable to both humans and machines. In this context, *ontologies* provide a shared, semantically explicit, and understandable vocabulary that can be used to structure raw data, processed data, and contextual metadata in a uniform way, thereby supporting interoperability and reuse [5].

A consistent representation of data is also important because it enables comparability, reproducibility, automated workflows, and data-driven research. When methodological procedures and their metadata are mapped into structured process templates and ontology-based models, they create a digital representation of materials science reality that is human-readable, computer-readable, and machine-processable. This standardization supports the integration of experimental data into repositories, facilitates automated data pipelines, and allows validation of units, ranges, missing metadata, and dataset quality. In materials science, where machine learning and digital workflows increasingly depend on high-quality and well-structured data, consistent data representation is therefore not only helpful for organizing information, but a fundamental prerequisite for reliable knowledge generation and scalable data-driven innovation [5].

1.2. Ontologies for Formal, Explicit Specification of a Shared Conceptualization

1.2.1. Ontology definition

A prerequisite for implementing the FAIR principles and building a comprehensive database infrastructure is formulating formal, standardized knowledge representation through ontologies [4], [5]. The most widely cited definition of an ontology in computer science is that of Studer et al. (1998) [11], who defined an ontology as 'a formal, explicit specification of a shared conceptualization.' In this definition, 'formal' means machine-readable abstraction of a domain; 'explicit' refers to well-defined constraints, relationships, and dependencies between concepts; 'shared' constitutes the building and agreement by a group of different people; and 'conceptualization' is an abstract model describing a particular field of knowledge [4], [11], [12].

In other words, ontologies form a sustainable representation of expert knowledge by using a formal, standardized language to express a vocabulary as an interconnected collection of concepts, attributes, relations, constraints, individuals, and axioms [4], [13]. They are formal collections of concepts and their relationships, systematically and explicitly organizing knowledge in various domains for both humans and machines, often employing a commonly agreed-upon, though not technically required, shared and consistent vocabulary. Ontologies reduce language barriers and ambiguities through standardized terminology, facilitating efficient data exchange and providing a clear mapping to the domain's context [3].

In the context of materials science, ontologies semantically structure and annotate materials data, represent knowledge sustainably, facilitate storing data and information in semantically organized ways, and support data handling [4], [5]. As a result of these functions, ontologies enable: (i) knowledge organization through hierarchical structure and relationships between concepts; (ii) global conceptualization for materials information integration; (iii) linked materials data publishing; (iv) semantic query support through languages such as SPARQL; (v) interoperability between related domains; and (vi) automatic inference support for discovering new materials knowledge [11].

1.2.2. Ontology languages and key components

Ontologies are implemented using standardized knowledge representation languages developed under the Semantic Web technology stack. The Web Ontology Language (OWL) is the primary language for authoring ontologies, recommended by the World Wide Web Consortium (W3C). OWL is built on top of the Resource Description Framework (RDF), which represents information as triples consisting of a subject, predicate, and object. RDF provides the foundational data model for the Semantic Web, enabling the publication of linked data on the web, while OWL adds richer semantics including cardinality constraints, disjointness, and property characteristics such as transitivity and symmetry [14].

An OWL ontology consists of several key components. Classes represent sets of individuals sharing common characteristics (for example, 'CreepTestProcess' class). Individuals or instances are the concrete members of classes (like 'CreepTestProcess_1', a creep test performed on a given date). Object properties relate individuals to other individuals (e.g., 'hasInput' linking a test to a specimen), while data properties relate individuals to literal values (e.g., 'hasValue' linking a test parameter to a value). Axioms are logical statements that constrain the interpretation of classes and properties, enabling automated reasoning to infer new knowledge and detect inconsistencies. Together, these components enable ontologies to serve as machine-processable semantic descriptions of domain knowledge [15].

1.2.3. Ontology levels

The ontology engineering community has proposed ontologies with different levels of abstraction to ease reuse and to layer ontologies upon each other. Although no single agreed-upon ontology hierarchy exists, adapting the widely cited ontology classification of Guarino [16], one can largely distinguish four different levels of abstraction in ontology design, illustrated in Fig. 1.1. These levels form a spectrum from the most general and most reusable ontologies at the top to the most specific and least reusable at the bottom [17].

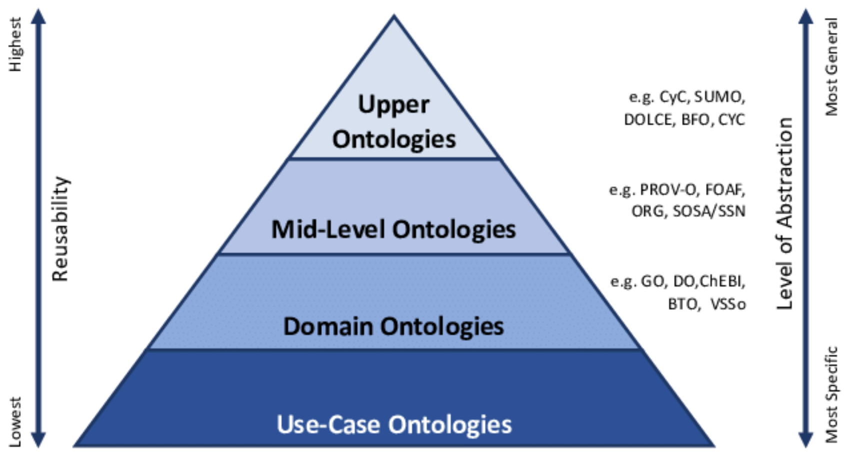

Fig 1.1. Levels of abstraction in ontology design [17].

- **Top-Level Ontologies (TLOs):** Top-level ontologies define very general terms that are common across all knowledge domains, such as object, process, quality, and relation. They establish the highest level of abstraction and provide the semantic foundation upon which all other ontologies can be built, ensuring that ontologies developed independently can nonetheless be integrated through a shared conceptual root. Prominent examples include Cyc [18], the Suggested Upper Merged Ontology (SUMO) [19], the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [20], and the Basic Formal Ontology (BFO) [21]. Because they capture only the most general categories, upper ontologies are the most reusable but also the furthest removed from any specific application [17].
- **Mid-Level Ontologies (MLOs):** Mid-level ontologies act as a bridge between the abstract content of an upper ontology and the richer detail of various domain ontologies. They capture concepts that are shared across many domains (for example, space, time, provenance, and organizational structure) and define them generally enough that their semantics can be further narrowed by a domain ontology. Widely used examples include the OWL Time Ontology [22] and Geonames for spatial and temporal concepts, as well as PROV-O for provenance [23], FOAF for agents and social relations [24], ORG for organizations [25], and SOSA/SSN for sensors and observations [26].
- **Domain-Level Ontologies (DLOs):** Domain ontologies define concepts and relations that belong to a specific field of knowledge, modelling domain-specific definitions of terms. They inherit their high-level structure from upper and mid-level ontologies while adding the detailed vocabulary needed to represent a particular subject area. Examples span many disciplines: the Gene Ontology

[27], the Disease Ontology [28], and ChEBI [29] in the life sciences; the Building Topology Ontology (BTO) in architecture [30].

- **Application-Level Ontologies (ALOs):** Application ontologies include a set of detailed classes and relations that are highly dependent on a specific use case. They typically extend the terms of a domain ontology to serve the concrete needs of a particular application or dataset. For example, in a smart-home environment for an apartment building, a use-case ontology might extend terms in a domain ontology so that they can be applied to several similar units across an apartment complex [17]. Being the most specialized, use-case ontologies offer the least reuse potential but the greatest expressiveness for their intended task.

This four-level classification provides the conceptual framework for the present work. The Creep Testing Ontology (CTO) developed in this thesis is a use-case-oriented application ontology: it builds on the Basic Formal Ontology (BFO) as its upper-level foundation (Section 1.5), reuses the Platform MaterialDigital Core Ontology (PMDco) as a mid-level bridge, and specializes the general vocabulary of the materials-testing domain to the specific requirements of creep testing. In this way, CTO inherits the reusability and interoperability of the higher levels while providing the detailed semantics needed to represent creep experiments in full.

1.3. Basic Formal Ontology (BFO)

The Basic Formal Ontology (BFO) is a small, rigorously axiomatized top-level ontology designed to support information retrieval, analysis, and integration across scientific and other domains [31]. It is standardized as ISO/IEC 21838-2:2021, making it the only top-level ontology to receive formal international standardization [32]. BFO is currently used by more than 800 ontology-driven projects worldwide and serves as the upper-level foundation for the Open Biomedical Ontologies (OBO) Foundry, a collective of ontologies adhering to shared principles of openness, collaborative development, and interoperability [33].

At its core, BFO is organized around a single fundamental distinction: the partition of all entities into continuant and occurrents [21], [31]. Continuants are entities that persist through time while maintaining their identity; they are wholly present at every moment at which they exist. Examples include material objects (a cell, a rock, a measuring instrument), immaterial entities (a database entry, a geometric boundary), and qualities or dispositions that depend on their bearers (the temperature of a substance, the fragility of a glass). Occurrents, by contrast, are entities that unfold in time and have temporal parts; they are never wholly present at a single moment. Examples include processes (a chemical reaction, a surgical procedure, a measurement act), process boundaries (the beginning of a process, the moment of death), and temporal and spatiotemporal regions (the duration of an experiment, the volume occupied by a reaction over time). This distinction, illustrated in Fig. 1.2, is not merely taxonomic, it reflects a deep ontological commitment: continuants are the bearers of properties that change, while occurrents are the changes themselves [21].

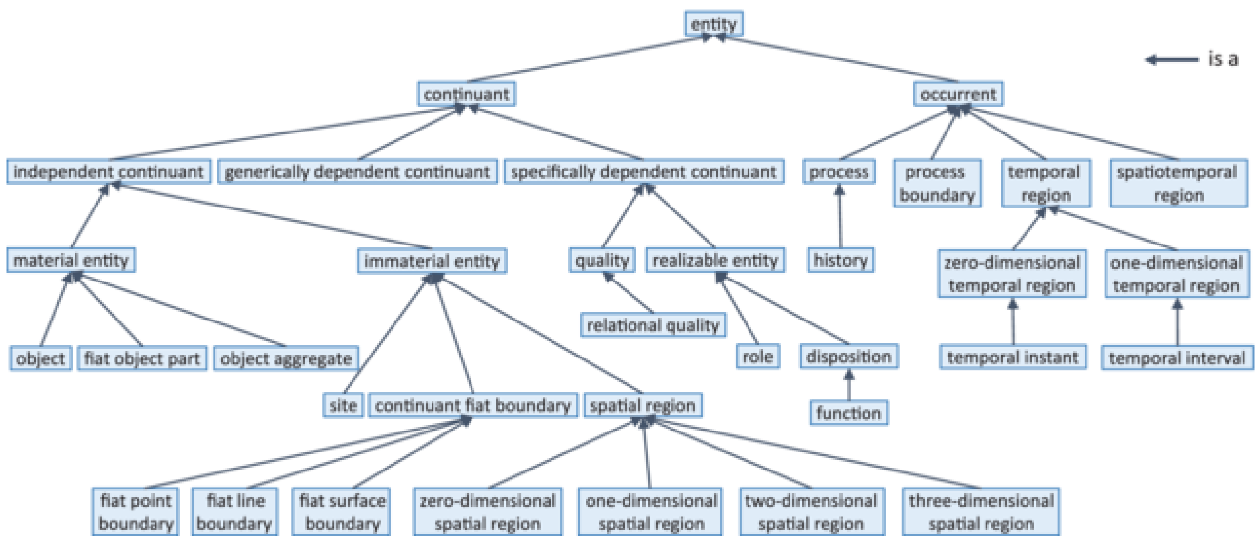

Fig 1.2. The Basic Formal Ontology (BFO) class hierarchy. Shows the top-level partition into Continuant and Occurrent, with major subcategories [32].

1.4. Ontology Development Pipeline

The development of a domain ontology follows established methodologies that have been refined through decades of practice in the ontology engineering community. Several methodologies have been proposed, including the Ontology Development 101 method [34], the NeOn Methodology [35], and the practices codified by the Open Biomedical Ontologies (OBO) Foundry [33]. While these methodologies differ in detail, they share a common core of activities: defining the ontology's scope and purpose, formulating competency questions (CQs) that the ontology must be able to answer, collecting and organizing domain terminology, designing the class hierarchy and relations, implementing the formal ontology using an editor such as Protégé, and evaluating the ontology through logical reasoning and CQs testing. Fig 1.3 shows a general pipeline for ontology development, structured around four sequential phases:

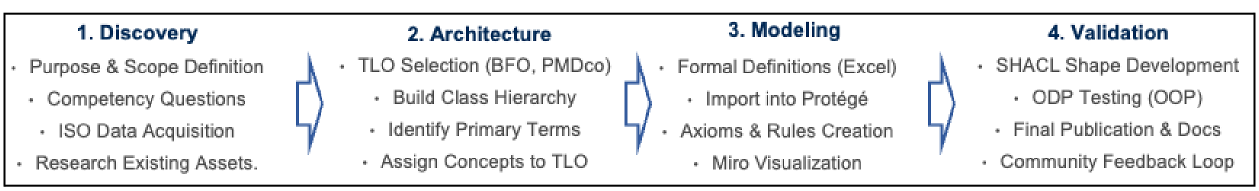

Fig 1.3. General pipeline for ontology development, comprising four sequential phases: Discovery, Architecture, Modeling, and Validation.

- **Phase 1- Discovery:** The Discovery phase establishes the purpose, scope, and functional requirements of the ontology. The first step is to define the ontology's purpose, the semantic problem it is meant to solve and the community of use it is meant to serve. The scope is then bounded by the domain of interest, the entities and processes to be represented, and the intended knowledge sources. CQs are formulated to capture the queries the ontology must support, serving as verifiable functional requirements throughout development. Terminology resources are collected from multiple complementary sources: (i) relevant domain standards, which provide normative vocabulary and procedural definitions; (ii) the domain literature itself, from which key terms, property names, and descriptors are extracted; (iii) existing ontologies in the same or neighboring domains, which provide reusable classes and relations; and (iv) any existing data schemas or databases whose structure reflects the main knowledge categories of the domain. Finally, existing ontological assets are systematically reviewed to avoid duplication and identify opportunities for reuse and alignment.

- **Phase 2- Architecture:** The Architecture phase defines the high-level semantic structure of the ontology. The first architectural decision is the selection of a top-level ontology, a domain-neutral foundation that provides a rigorously axiomatized distinction between categories such as continuants and occurrents, enabling interoperability with the broader ecosystem of ontologies built on the same foundation and supporting formal reasoning. At the mid-level, an existing mid-level ontology is typically adopted as a bridge between the top-level ontology's abstract categories and the domain-specific concepts to be modeled, contributing reusable structures for processes, materials, devices, specifications, and values wherever applicable. The class hierarchy is then designed by identifying the primary domain terms from the collected terminology and assigning each to the appropriate top-level category. This phase also defines the ontology's namespace and repository structure.
- **Phase 3- Modeling:** The Modeling phase transforms the architectural design into a formal, machine-readable ontology. Formal class definitions — including labels, textual definitions, logical axioms, and restrictions — are first drafted in a structured spreadsheet (e.g., an Excel template) to facilitate review and community feedback before implementation. These definitions are then imported into an ontology editor such as Protégé, where classes, object properties, data properties, and annotations are created and organized. SHACL (Shapes Constraint Language) shapes are developed to define data-level constraints that complement the OWL axioms, ensuring that instance data populating a knowledge graph conform to the expected structure and value ranges. Each Ontology Design Pattern (ODP) is tested in isolation using the OWL ontology profiling approach before being integrated into the full ontology.
- **Phase 4- Validation:** The Validation phase ensures that the ontology is logically consistent, semantically adequate, and practically useful. Automated reasoning with a reasoner such as HermiT is applied to check logical consistency, detect unsatisfiable classes, and infer the complete class hierarchy. CQs are translated into SPARQL queries and executed against populated knowledge graph instances to verify that the ontology provides the necessary vocabulary and structure to answer each question. Expert review by domain specialists and the relevant ontology community provides domain-level validation of the ontology's coverage, correctness, and alignment with community standards. Feedback from each validation cycle is fed back into earlier phases (Discovery through Modeling) in an iterative loop, refining the ontology until it meets all quality criteria. The final phase typically concludes with ontology release through a public repository, comprehensive documentation, and publication under an open license.

This four-phase pipeline ensures that a domain ontology is developed according to recognized ontology engineering best practices (including OBO Foundry principles such as openness, collaborative development, and common formatting) and that the resulting ontology is both semantically rigorous and fit for its intended purpose.

1.5. Ontology Design Patterns

Ontology Design Patterns (ODPs) are reusable, modular solutions to recurrent modeling problems in ontology engineering. Rather than designing every class, relation, and axiom from first principles, ODPs provide semantically coherent templates that have been tested and validated across multiple ontology projects. ODPs promote faster development through modular reuse, consistency across ontologies that share the same patterns, reduced modeling errors since patterns are community-vetted, and improved maintainability as improvements to shared patterns propagate to all ontologies that reuse them [36], [37].

Beyond providing generic reusable templates, ODPs scholarship distinguishes several pattern types according to the modeling problem they address. Content patterns capture recurring conceptual structures within a single ontology, while alignment patterns address a different but related problem: recurring correspondences and mismatches that arise when reconciling multiple ontologies with each other. Systematic analysis of ontology mismatches distinguishes three levels of abstraction for expressing such correspondences (executable transformation rules, concrete pairwise correspondences, and generalized alignment patterns) and shows how a library of these patterns can support both manual alignment design and automated pattern-detection algorithms [38]. This distinction underlines that ODPs are not a single monolithic tool but a family of complementary techniques suited to different stages of ontology engineering.

At the same time, adoption of ODPs in practice remains uneven. ODPs are characterized as a reusable solution to a recurrent modeling problem, and discuss both the promises of pattern-based design (faster development, cross-ontology consistency, reduced error rates) and persistent barriers to wider adoption, including difficulties in pattern discovery, inconsistent documentation quality, and limited tool support for instantiation and validation [37].

In the MSE domain, the PMDco pattern library provides a set of documented, BFO-compliant ODPs covering common modeling scenarios such as process chains, chemical compositions, mechanical properties, and value specifications. These patterns serve as the building blocks from which domain and application ontologies are constructed by adding domain-specific subclasses and relations while inheriting the semantic structure and BFO alignment of the parent pattern [3].

The PMDco pattern library [39] documents a broader set of patterns than those directly reused in CTO, organized around two complementary use cases. A first group, illustrated through a running example of a steel ingot's lifecycle, addresses structural modeling problems: the duality of an object and the portion of matter it consists of, the temporal boundaries and states of a process, chains of manufacturing subprocesses with their inputs and outputs, and the realization of roles such as an evaluant role in a measurement assay. A second group of "frequently applied patterns" targets the representation of material properties and their values: fundamental, behavioral, and relational qualities of materials, the distinction between a measured value and a specified setpoint, and the expression of scalar and categorical values themselves. Fig 1.4 illustrates this last case, the “Scalar Value Specification” pattern: a property is linked via pmd:specified_by_value to a generically dependent information content entity typed as obi:scalar_value_specification, which carries the numeric magnitude through obi:has_specified_numeric_value and the associated unit through iao:has_measurement_unit_label — here, a value of 42.0 in the unit angstrom [39].

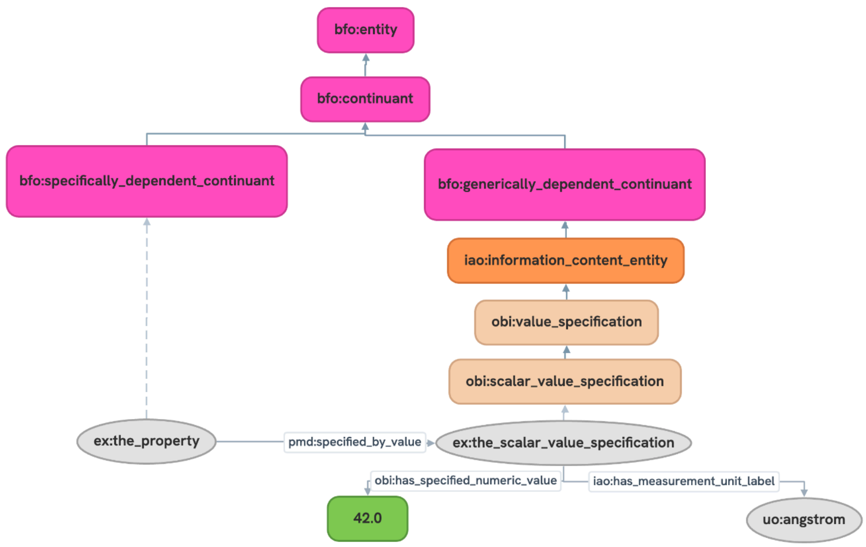

Fig 1.4. The PMDco Scalar Value Specification pattern for expressing a quantitative value [39].

The MWO ontology also employs ODPs for representing resource contact points, resource implementation, and RDM metadata structures, demonstrating that ODP-based development enables scalable, reusable, and semantically robust knowledge representation [40].

1.6. Ontology-Driven Knowledge Graphs

A knowledge graph (KG) is a structured, graph-based representation of knowledge in which entities (nodes) are connected by typed relationships (edges), forming an interlinked network of facts that can be queried, reasoned over, and traversed by machines [41]. Unlike traditional relational databases, which rely on rigid schemas and table-based storage, KGs represent information as subject–predicate–object triples following the Resource Description Framework (RDF) data model, enabling flexible integration of heterogeneous data from multiple sources without requiring upfront schema alignment [42].

When a KG is built upon an ontology, the ontology provides the semantic schema (the controlled vocabulary of classes, properties, and logical constraints) that governs which types of entities and relations are permissible and how they should be interpreted [43]. This is the defining characteristic of an ontology-driven KG: rather than encoding facts using ad-hoc labels that vary across datasets, every entity is typed with an ontology class (e.g., cto:CreepTestingProcess), every relation uses an ontology property (e.g., cto:hasInput), and every value is associated with a unit and provenance. This semantic normalisation enables three capabilities that are unattainable with schema-free KGs: (i) expressive querying via SPARQL, the W3C-standard query language for RDF data, allowing questions such as 'retrieve all specimens tested above 650 °C with a rupture time exceeding 1,000 hours' to be answered across multiple publications in a single query; (ii) automated logical reasoning, in which a reasoner such as HermiT or Pellet infers new facts from explicitly stated axioms (for example, inferring that a test conducted at 700 °C under 500 MPa belongs to the class of high-temperature creep tests; and (iii) interoperability, since ontologies aligned with a common top-level foundation (such as BFO) enable data from different KGs (tensile data from TTO, fatigue data from FTO, creep data from CTO) to be federated and queried together [4], [44].

The benefits of ontology-driven KGs are particularly pronounced in scientific domains such as materials science, where data are generated by diverse experimental methods, reported in heterogeneous formats, and distributed across thousands of independent publications. By anchoring the KG in a well-defined ontology, extracted facts become FAIR-compliant: Findable through persistent identifiers and SPARQL endpoints, Accessible via standard web protocols, Interoperable through shared ontology classes and alignment with upper ontologies, and Reusable because the semantics of every entity are explicitly defined [8].

The mechanism of ontology-driven KGs is fundamentally straightforward; the ontology provides the semantic schema, a formal vocabulary of classes, Connective properties, and logical axioms that prevent the ingestion of inconsistent data. The knowledge graph populates this framework with concrete individuals, such as specific heat treatments, rupture time values, or grain size measurements. Because each node is typed with an ontology class and every edge utilizes a formal property, a researcher executing a SPARQL query remains insulated from whether the original source reported temperatures in Celsius or Kelvin, or identified the material as Inconel 718 versus its elemental constituents. The ontology normalizes these decisions at the point of data entry, ensuring the query is successful across disparate sources.

In the MSE domain, the MSE Knowledge Graph (MSE-KG) is a consortium-wide semantic infrastructure developed within NFDI-MatWerk to make materials science and engineering (MSE) research data findable, interoperable, and queryable across institutional boundaries [40]. Rather than centralizing raw datasets, MSE-KG federates metadata contributed by participating institutes into a shared, ontology-governed representation, built on the NFDI-MatWerk Ontology (MWO) as its semantic backbone. This allows datasets originating from different labs, instruments, and data management practices to be described using a common, BFO-aligned vocabulary, so that entities such as materials, processes, and properties can be linked and queried consistently regardless of where or how the underlying data was produced (Fig. 1.5). Beyond the technical infrastructure, MSE-KG depends on active governance: coordinating with data and ontology contributors across the consortium, aligning modeling decisions with product owners, and maintaining a demo and onboarding pipeline so new institutes and use cases can be integrated smoothly. Its value lies less in the size of the graph itself than in the semantic normalization it enforces — turning otherwise siloed, heterogeneous datasets into a genuinely federated resource that supports cross-institutional scientific queries within the NFDI-MatWerk community [40].

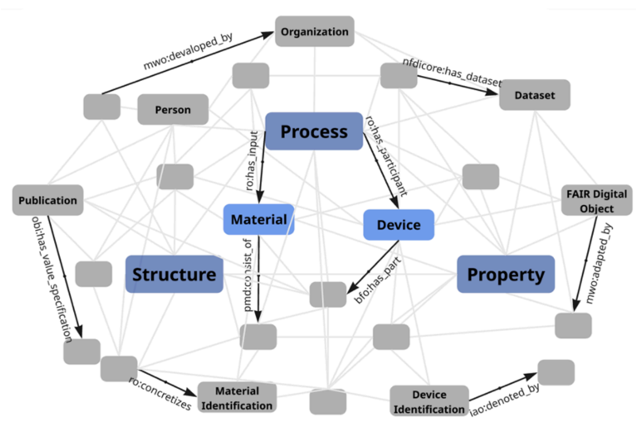

Fig 1.5. MSE-KG; semantic representation of MSE concepts in relation with RDM entities [45].

Further example ontology-driven KGs in MSE domain include the Vickers hardness knowledge graph utilizing the VTO ontology [45], MGED-KG, which algorithmically derives terminology hierarchies from text corpora [46], and MatKG, which employs ontologies to normalize entities extracted from a broad sweep of materials literature for downstream discovery [47].

1.7. Large Language Models for Populating Knowledge Graphs with Literature Data

Large language models (LLMs) (neural networks trained on vast corpora of text using self-supervised learning) have emerged as powerful tools for information extraction from unstructured text. Models such as GPT, LLaMA, and Gemini can identify entities, extract relations, and structure information directly from natural language, making them well-suited for the task of populating KGs from scientific literature [46], [48]. Recent surveys report that LLM-based pipelines now match or exceed traditional information extraction models in recall for named entity recognition and relation extraction, although concerns about precision and hallucination remain [49], [50].

While ontologies address the challenge of data representation, large language models (LLMs) address the challenge of scale. The volume of materials science literature far exceeds the capacity of manual human annotation; LLMs provide a scalable methodology for identifying facts and structuring them from unstructured text. The critical question remains whether these models can achieve the precision required to populate a formal knowledge graph without necessitating extensive manual correction of the output.

Two paradigms for LLM-based KG construction can be distinguished;

i- Schema-free extraction: In schema-free extraction, the LLM identifies entities and relations without a predefined vocabulary, producing ad-hoc triple structures (e.g., 'IN718 → has rupture time → 245 hours'). The paradigm of schema-free extraction is enticingly simple, yet it suffers from significant limitations when merging data across publications. Although recent surveys indicate that LLM-based pipelines can match or exceed traditional NER models in recall [46], [48], the resulting data often lacks semantic consistency. A single property, such as rupture time, may be extracted with inconsistent labels or units that cannot be automatically equated, while relations like "has rupture life" or "exhibits time to failure" proliferate without a shared vocabulary. While sufficient for exploratory search, this lack of precision causes schema-free knowledge graphs to fail when answering complex, cross-study research questions [51].

ii- Ontology-guided extraction: In ontology-guided extraction, by contrast, the LLM is instructed through schema-constrained prompts to map every extracted entity and relation to a specific ontology class or property. The ontology thus constrains the LLM's output space, ensuring that 'IN718' is always typed as cto:NickelBasedSuperalloy, that 'rupture time' is always represented via standard axioms, and that every value carries its unit and measurement context[52], [53]. This approach transforms the LLM from an uncontrolled text generator into a schema-aware extraction engine.

Several implementations have demonstrated the potential of this approach in the MSE community. Durmaz et al. introduced a benchmark dataset for extracting process–structure–property entities [54], followed by the development of ontology-conformal models designed to map text segments directly to formal ontology terms [55]. Ghosh and Tewari demonstrated agentic workflows for harvesting thermoelectric properties from ten thousand articles [56], while Nasrabadi et al. achieved the full loop of extracting hardness data, transforming it into RDF, and querying the resulting graph with full provenance [57].

Despite these successes, a systematic, quantitative comparison of ontology-guided versus schema-free extraction remains a significant research gap. While it is widely accepted that ontological constraints improve label consistency and entity linking, the magnitude of this improvement has not been rigorously measured across different parameter categories or LLM architectures. This thesis addresses this gap by conducting a controlled experiment using a manually annotated benchmark dataset.

1.8. Use Case — Creep Testing Data of Inconel Materials

The development and evaluation of an ontology-guided LLM extraction framework requires a focused, well-bounded use case, since targeted knowledge extraction from scientific literature depends critically on a clearly defined domain and scope. This thesis selects the extraction of creep-related metadata of Inconel nickel-based superalloys as its use case.

Inconel: Nickel-based superalloys of the Inconel family (like IN718, IN625, IN740H, IN617, and IN713C) are the materials of choice for the most demanding high-temperature structural applications. They are used extensively in gas turbine engines (blades, discs, casings), nuclear reactor components, and industrial gas turbines, where they must withstand temperatures exceeding 650 °C under sustained mechanical load for tens of thousands of hours [58]. The key to their performance is their microstructure: a nickel-chromium matrix strengthened by coherent precipitates, primarily γ′ ($Ni_3(Al,Ti)$) and γ″ (Ni_3Nb), whose size, volume fraction, and distribution are controlled through carefully designed heat treatment sequences. Creep, the time-dependent plastic deformation of a material under constant load at elevated temperature, is the life-limiting mechanism for these components. Understanding and predicting creep behaviour (particularly rupture time, minimum creep rate, and the influence of microstructure on these properties) is therefore of both fundamental scientific interest and enormous industrial importance.

Creep testing process: Creep testing is standardised internationally through ISO 204:2023 (Metallic materials — Uniaxial creep testing in tension — Method of test) [59], which specifies the test procedure, specimen geometry, loading and heating protocols, and the quantities to be reported. A typical creep experiment records: the material identity and chemical composition; the full processing history (manufacturing route, solutionizing, aging); the initial microstructural state (grain size, precipitate characteristics); the test conditions (temperature, applied stress, test environment); and the resulting creep response (rupture time, minimum creep rate, elongation at rupture, reduction of area). When multiple tests are performed at different stress and temperature levels, derived quantities such as the stress exponent n and the apparent activation energy for creep (Qc) can be extracted. This rich, multi-parameter data structure (the processing–structure–property chain) makes creep an excellent domain for ontology-based representation, because every parameter has a defined meaning, a known relationship to other parameters, and a standardised measurement procedure.

Inconel creep properties in literature: The scientific literature on Inconel creep provides a sufficiently rich yet manageable corpus for developing and testing the extraction methodology. It is estimated that hundreds of research papers have been published in this thematic area, spanning journals such as Materials Science and Engineering A, Acta Materialia, Metallurgical and Materials Transactions A, and Materials at High Temperatures. These papers report experimental creep data in a variety of formats: tables of measured properties, curves of creep strain versus time, micrographs of initial and post-creep microstructures, and textual descriptions of processing routes. The diversity of reporting conventions (different units, different alloy naming schemes, different levels of detail in processing and microstructure description) makes manual data extraction slow and error-prone, yet it is precisely this diversity that makes the corpus an ideal testbed for evaluating whether an ontology-guided LLM can produce consistent, machine-readable metadata where schema-free methods fail.

Chapter 2: Related Work

2.1. Ontologies in Materials Science and Engineering

To date, a wide range of ontologies have been introduced in the MSE domain [3], [40]. The first generation of MSE ontologies (including MASON, MatOWL, the materials ontology by Ashino, ONTORULE for steel, SLACKS for laminated composites, PREMaP, and MMOY) mainly remained unused due to their narrow and specific focus fields, inappropriate levels of abstraction, insufficient representation of instances, and poor interoperability with other ontologies [60]. To address these limitations, more recent ontology development in MSE has converged on the principle of building ontologies as extensions of standardized upper-level ontologies, ensuring semantic consistency and cross-domain interoperability [4], [60].
Fig 2.1 provides a general overview of the different levels of ontologies used in MSE domain:

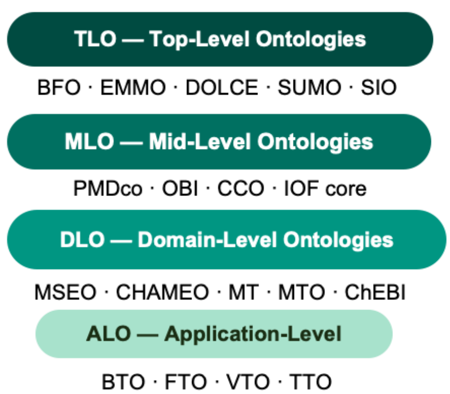

Fig 2.1. Overview of different level ontologies in materials science and engineering.

TLOs: Basic Formal Ontology (BFO) [31], [61], Elementary Multiperspective Material Ontology (EMMO) [62], and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [63] are the most applicable TLOs in the materials science domain [4]. Other TLOs reused in MSE include the Provenance Ontology (PROV-O), the Suggested Upper Merged Ontology (SUMO), and the Semantic Science Integrated Ontology (SIO) [60].
MLOs: The Common Core Ontologies (CCO) [64] and the Industrial Ontologies Foundry (IOF) core [REF] are MLOs established based on BFO. In the MSE domain specifically, the Platform MaterialDigital Core Ontology (PMDco) has been developed as an MSE-centric MLO that bridges semantic gaps between high-level, MSE-specific, and other science domain semantics through community-based curation [3]. NFDIcore is another BFO-based MLO focused on research data management concepts [40].
DLOs: Several domain-level ontologies populate the MSE landscape. The Materials Science and Engineering Ontology (MSEO), built on BFO and the IOF Core, represents extensive semantics for experimental data and processes in MSE [60]. The Characterization Methodology Ontology (CHAMEO) was

developed to harmonies characterization methodologies across techniques and explicitly includes mechanical testing within its taxonomy of characterization methods [40], [65]. The NFDI-MatWerk Ontology (MWO) is a BFO-compliant ontology designed to standardize metadata, link distributed datasets, and support digital RDM in MSE. MWO is developed as a modular extension of NFDIcore, reuses PMDco, and models key RDM-in-MSE elements such as researchers, organizations, projects, software, workflows, datasets, instruments, and services [40].The Mechanical Testing Ontology (MTO) is another important DLO that provides a cross-test vocabulary for mechanical testing processes [4]. Other domain-level ontologies include the Tribology Ontology (tribAIn), the Materials Ontology (MatOnto), and various process-specific ontologies for additive manufacturing and battery research [40], [60].

Given their high conceptual relevance to MSE, EMMO and PMDco are introduced specifically in the next sub sections.

2.1.1. European Materials Modelling Ontology (EMMO)

The European Materials Modelling Ontology (EMMO), developed by the European Materials Modelling Council (EMMC), offers an alternative top-level framework grounded in mereocausality (a first-order logic theory combining mereology (parthood) with causality) and supports multiscale, multiperspective representation of materials and physical phenomena. EMMO provides high semantic richness and domain coverage, though its complex structure can impact query efficiency and integration with other ontologies [60], [66].

Beyond its top-level physicalist commitments, EMMO's distinctive contribution to MSE ontology engineering lies in the deliberate unification of three domains that are conventionally treated separately. As illustrated in Fig 2.2, EMMO provides the connection between the physical world of real materials and phenomena, the materials characterisation world in which those phenomena are observed and measured, and the materials modelling world in which they are represented, simplified, and computed. This unification is achieved through EMMO's semiotic branch, which formalizes characterisation as an observation process linking an observer, an object, and the resulting property, and through its models branch, which formalizes modelling as a semiotic process in which a theory, physics equation, or material relation stands for the same underlying physical phenomenon. Because both characterisation and modelling are grounded in the same physicalist, mereotopological top level, EMMO can express, within a single logical framework, how an experimentally measured property and its simulated or modelled counterpart relate back to one and the same physical entity, a capability of particular relevance to materials science, where experimental and computational workflows must ultimately be reconciled. This tri-domain integration is a key reason EMMO is frequently discussed alongside BFO as an alternative top-level candidate for MSE ontologies, even though, as noted above, its richer axiomatization can come at the cost of reasoning and query efficiency relative to BFO's leaner structure [62].

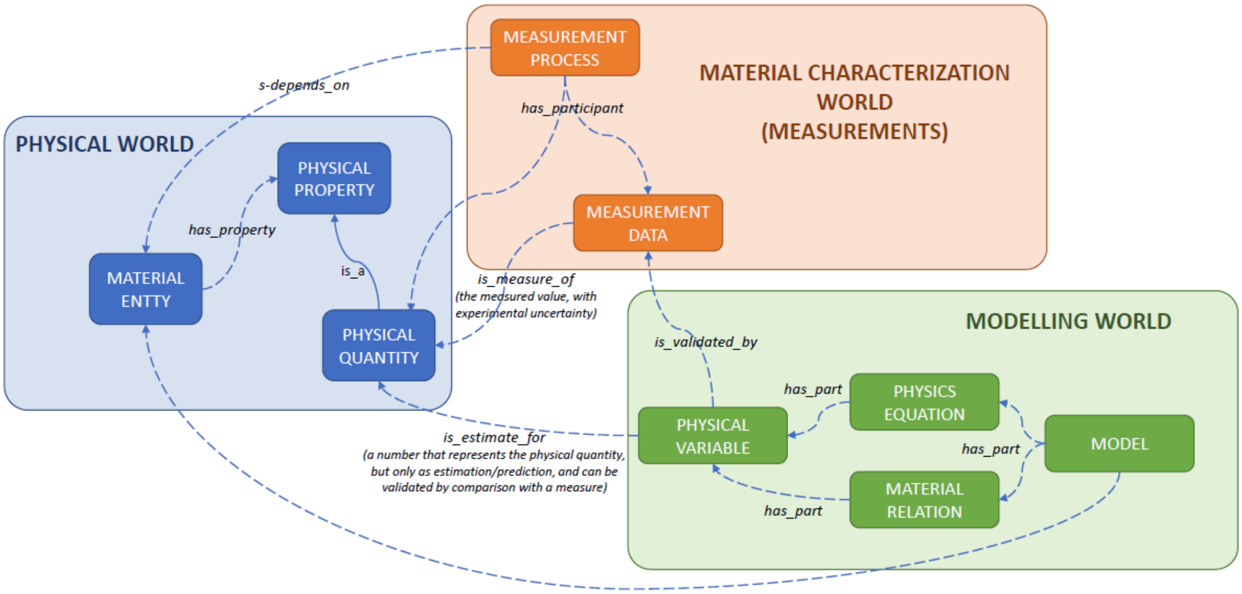

Fig 2.2. EMMO provides the connection between the physical world, materials characterisation world and materials modelling world [63].

2.1.2. Platform Material Digital Core Ontology (PMDco)

The Platform MaterialDigital Core Ontology (PMDco) is a MLO, developed to promote semantic interoperability in materials science and engineering. It was introduced to bridge the gap between very abstract TLOs and highly specific application ontologies used to describe experiments, simulations, or industrial processes. PMDco provides a shared MSE-centered vocabulary that helps align heterogeneous data, terminology, and domain perspectives across the materials life cycle. In this way, it functions as an intermediate semantic layer that is easier for domain experts to understand and use, while still enabling structured and interoperable knowledge representation [3].

A central strength of PMDco is that it supports the consistent description of processes, processing nodes, objects, and associated value objects, thereby enabling the modelling of complete process chains and the traceability of generated materials data. The ontology is aligned with the BFO framework, which strengthens provenance representation and reproducibility, and it is designed to be extended by domain-specific application ontologies. Thus, as a mid-level ontology, PMDco not only lowers the threshold for ontology development in MSE, but also supports FAIR data creation, knowledge exchange, and the integration of experimental, simulation, and industrial materials data within a coherent semantic framework [3]. As illustrated in Fig. 2.3, PMDco represents the entire materials process-chain, intending to make relevant materials information consistently available across entire value chains, rather than isolated to individual experiments or processing steps, so that informed decisions can be made at any point along the chain. This continuous, chain-based traceability is a key reason PMDco is well suited as the mid-level foundation for CTO: the same principle (linking heat treatment, mechanical testing, and characterization steps into a single traceable chain) is exactly what is required to preserve the processing–structure–property relationships central to creep research [3].

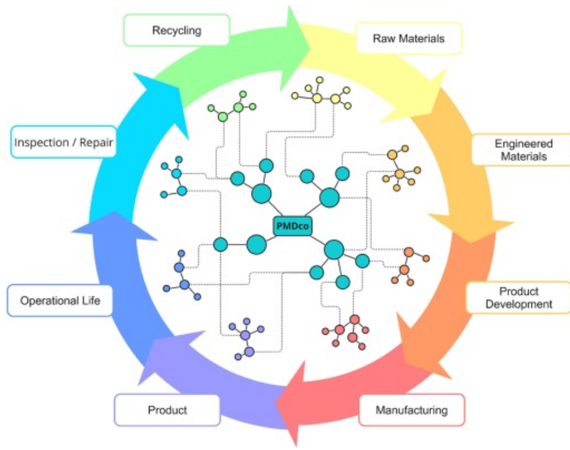

Fig 2.3. Materials information across entire value chains is made consistently available through continuous process representations based on PMDco [3].

2.2. Ontologies in the Mechanical Testing Domain

Mechanical testing occupies a distinct position in the ontology landscape because mechanical property data are uniquely context-dependent. A value such as tensile strength, fatigue life, hardness, or creep rate only becomes meaningful when connected to specimen geometry, test procedure, loading conditions, temperature, environment, instrumentation, and data processing [4]. As a result, ontology development in this domain has progressed test-by-test, producing a family of standard-specific application ontologies built on shared upper-level foundations. These test-specific ontologies are compared in Table 2.1, which summarizes each ontology's upper-level foundation, domain scope, and supported testing standards.

Table 2.1. Comparative survey of existing domain and application ontologies in materials testing.

Acronym	Full Name	Upper-Level Base	Scope / Coverage	Key Tests Covered
MT	Mechanical Testing (domain ontology)	EMMO	Physical models & simulation of mechanical tests	Tensile, hardness, fatigue, creep (model description)
MTO	Mechanical Testing Ontology (domain ontology)	BFO+IOF+MSEO / PROVO+PMDco /	Vocabulary from ISO 23718; data management focus	Tensile, Brinell, Vickers, stress relaxation, fatigue
MSEO	Materials Science & Engineering Ontology (domain ontology)	BFO+IOF	Experiments data & processes in MSE; CCO stack	Tensile, fatigue, hardness — semantic data representation
BTO	Brinell Testing Ontology	BFO+IOF+MSEO / PROVO+PMDco / EMMO+CHAMEO+MT	Application-level knowledge graph for hardness testing	Brinell hardness (DIN EN ISO 6506-1:2015)
VTO	Vickers Testing Ontologies	BFO + IOF + MSEO	Application-level knowledge graphs	Fatigue, Vickers hardness
TTO	Tensile Test Ontology	BFO+IOF+MSEO / PROVO+PMDco / EMMO	Semantic representation of tensile testing workflows	Tensile testing (ISO/DIN standards)
FTO	Fatigue Testing Ontology	BFO+IOF+MSEO / PROVO+PMDco /	Semantic data for LCF experiments	Low-cycle fatigue

The Mechanical Testing Ontology (MTO) [4] is the most comprehensive cross-test ontology in the mechanical testing domain. MTO was developed based on ISO 23718 (Metallic materials — Mechanical testing — Vocabulary) for the semantic representation of mechanical testing experiments, quantities, artefacts, and report data. It covers tensile, Brinell hardness, Vickers hardness, stress relaxation, and fatigue testing processes. MTO served as the semantic basis for a 'digital materials mechanical testing lab' in which heterogeneous experimental test data of varying structures, languages, and formats were converted to standardized RDF. SPARQL queries were successfully used to evaluate data retrieval and response to CQs, demonstrating that the approach enables industries to access trustworthy and traceable mechanical testing data across organizational boundaries [4]. However, as a domain-level ontology, MTO intentionally remains at a level of generality that does not capture the full depth of any single test, delegating detailed modelling to application-level ontologies [4].

Building on MTO and related resources, a family of standard-specific application ontologies has emerged.

The Tensile Testing Ontology (TTO) models tensile testing according to ISO 6892-1 and is based on PMDco 3.0, reusing BFO, the Relation Ontology (RO), MWO, and NFDIcore [4], [40].

The Fatigue Test Ontology (FTO) represents low-cycle fatigue (LCF) testing according to DIN EN ISO 12106. FTO was developed to semantically structure LCF testing data (covering test parameters, specimen information, measured quantities, and metadata) and was integrated with the data management system to enable FAIR data publication, semantic querying, and reuse [67].

The Vickers Testing Ontology (VTO) models Vickers hardness testing according to ISO 6507 and was developed with the Ontopanel approach for mapping real experimental data into knowledge graphs. VTO reuses BFO, RO, PMDco, NFDIcore, and MWO, and has been demonstrated with SPARQL-based querying of experimental hardness datasets [45].

The Tensile Stress Relaxation Testing Ontology (TSRTO) models tensile stress relaxation testing according to DIN EN ISO 10319-1 and follows the same BFO/PMDco alignment strategy, with testing datasets for several copper alloys mapped into the knowledge graph and converted to machine-readable RDF [57].

The Brinell Testing Ontology (BTO) was developed as part of a comparative study evaluating the performance of different top-level ontologies (BFO, EMMO, PROV-O) for building application ontologies. Three versions of BTO were created using different TLO combinations and evaluated on ten parameters including semantic richness, domain coverage, extensibility, complexity, mapping efficiency, query efficiency, and integration capability. The results showed that all candidate TLOs can effectively develop application ontologies, with BFO providing the best balance of understandability, query efficiency, and cross-ontology integration [60].

Taken together, these application ontologies represent a significant achievement: the community can now model several important mechanical tests in semantically rich, standard-aware ways. They also establish a methodological template (BFO alignment, PMDco reuse, standard-specific vocabulary, ODP-based modelling, and ODK-based development) that this thesis follows and extends for creep testing.

2.3. Ontologies for Creep Testing

Compared to tensile, fatigue, and hardness testing, the ontology coverage of creep testing is noticeably less developed. The recent MSE ontologies review, which provides the most comprehensive recent survey of MSE ontologies, lists TTO, FTO, VTO, and TSRO among application ontologies for mechanical testing but contains no corresponding creep testing ontology [37]. Three existing efforts are worth examining in detail.

OntoFNCT — Full-Notch Creep Test Ontology: OntoFNCT is the most mature creep-related ontology currently available. Developed for full-notch creep tests (FNCT) on polyethylene according to ISO 16770, it is based on the PMDco framework and provides structured representation of test standards, methodologies, materials, primary data (force, elongation, time to failure), secondary data, and metadata. Schilling et al. demonstrated OntoFNCT in a data-transformation pipeline that converts raw FNCT files from CSV and Excel formats into machine-actionable RDF structures. The approach normalizes equipment metadata into generic, human-understandable categories such as machine type, model name, calibration date, and load cell specification, enabling semantic alignment of data originating from various test machines and software platforms [68]. A subsequent paper extended this work by implementing the full OntoFNCT ontology within a FAIR data framework for polymer mechanical testing. The ontology was designed to transform diverse and integrated data associated with FNCT into a well-structured, normalized, semantic exchange format, with an emphasis on interoperable connection through higher-level ontologies [69]. OntoFNCT is important because it proves that creep ontology engineering is viable and delivers measurable improvements in data management. However, it is a highly specialized ontology for one creep test configuration (FNCT), one material class (polyethylene), and one standard (ISO 16770). It does not provide a general vocabulary for metallic creep testing under ISO 204.

NIMS Creep Data Ontology: Ashino and Fujita developed an early ontology for representing the National Institute for Materials Science (NIMS) creep data sheets, a large, publicly available collection of creep rupture data for heat-resistant steels and superalloys. The ontology was part of a broader effort to define a standardized XML schema (MatML) for material properties and to establish an ontology that represents the structure of concepts related to materials (including definition, classification, and properties) for knowledge management in material selection and processing. The authors demonstrated that using OWL, knowledge about creep data analysis could be integrated with material databases via standardized XML schema data representation [70]. While pioneering for its time, this work predates the widespread adoption of BFO, OWL 2, and modern ontology engineering practices, and was built as a standalone schema rather than as part of an interoperable ontology ecosystem.

Creep Void Analysis Ontology: Zeb and Mätäsniemi [71] demonstrated an ontological approach to modelling creep void analysis data for automating the training of machine learning models that detect creep voids in scanning electron microscope (SEM) images. Their work focused on the ontological representation of creep void analysis data (including SEM images, ML model predictions, cropped creep-void images, and the feedback of the materials engineer) within a single coherent information model. They aligned concepts from EMMO, the ML-Schema, and PROV-O to connect experimental data, the analysis workflow, and expert interpretation in a transparent and traceable way. The ontology enabled quality control, querying, and retraining of ML models more systematically than manual data handling [71]. This work is notable because it demonstrates semantic integration of characterization data with ML workflows. However, its scope is limited to one specific characterization task and does not extend to the broader processing–structure–property chain.

Creep reference dataset schema: More broadly, Ávila Calderón et al. published a FAIR reference dataset for creep of single-crystal CMSX-6 Ni-based superalloy, documenting the full experimental procedure, specimen geometry, test conditions, and measured creep curves in a structured, reusable format. While not an ontology per se, this work demonstrates the growing recognition in the creep community that provenance-rich, machine-readable data are essential for reproducibility and data-driven research [72].

2.4. Evaluation of Existing Approaches and Research Gap

Based on research survey done, the existing approaches to representing creep-related knowledge fall short along several interconnected dimensions:

1. **BFO alignment and interoperability:** None of the available creep-related resources are built on the BFO. The earlier efforts such as Ashino's materials ontology predate BFO and OWL 2 entirely, while more recent frameworks, including OntoFNCT and the work of Zeb and Mätäsniemi on creep void analysis, were developed independently of any standardized top-level ontology. This absence of a common BFO foundation undermines interoperability, since these resources cannot be logically integrated with the growing ecosystem of BFO-aligned MSE ontologies (such as PMDco, MWO, and the family of mechanical-testing application ontologies) without substantial re-engineering.
2. **Standard basis:** Existing efforts are either tied to standards outside the metallic-creep domain, such as ISO 16770 for polyethylene in the case of OntoFNCT, or predate the current ISO 204:2023 standard for uniaxial creep testing altogether, leaving no ontology whose class hierarchy is grounded in the standard actually used by creep laboratories today.
3. **Semantic richness:** Available frameworks tend to model isolated characterization techniques rather than the full processing–structure–property narrative that creep research requires., e.g., material identity, processing history, microstructure, test conditions, and deformation mechanisms in one coherent schema.
4. **Lack of matching ODPs:** No ODPs can describe the creep concepts via available ontologies, and the PMDco pattern library itself provides no ready-made pattern for microstructural characterization or for distinguishing planned from measured test parameters, forcing any creep-modeling effort to design these structures from scratch rather than reuse a validated template.
5. **Usability:** Without a BFO-aligned, standard-grounded, semantically comprehensive, and pattern-supported ontology, researchers have no off-the-shelf vocabulary for creep data curation, forcing each project toward ad-hoc, non-reusable schemas.
6. **No precedent for ontology-driven LLM literature mining:** No existing creep or mechanical-testing ontology has been coupled with an LLM-based extraction pipeline for literature mining, so there is no established methodology, or evaluation benchmark demonstrating how such an ontology could actually guide automated extraction from scientific text.
7. **Unproven usability for ontology-driven KG construction:** Even where BFO-aligned mechanical-testing ontologies exist, none has been demonstrated in practice as the schema for populating and querying an ontology-driven KGs at scale, leaving open whether their class structures are practically usable for real KG construction workflows rather than remaining a documentation-only formalism.

Chapter 3: Research Objectives and Approach

3.1. Research Objectives

The primary objective of this thesis is to develop a BFO-compliant Creep Testing Ontology (CTO) that directly closes the gaps identified in the preceding evaluation of existing approaches. Ontology engineering is treated as the central research contribution of this work, rather than one of two co-equal aims; the ontology's subsequent use in LLM-guided extraction and knowledge graph construction serves to validate its practical usability rather than to define the thesis's primary scope.

To this end, the thesis pursues five interconnected ontological aims, each addressing a specific gap identified above:

1. Design and implement CTO following OBO Foundry best practices, aligned with BFO (ISO/IEC 21838-2) as the top-level foundation and PMDco as the mid-level bridge, ensuring logical integration with the existing ecosystem of BFO-aligned MSE ontologies.
2. Ground CTO's class hierarchy and competency questions in the ISO 204:2023 standard for uniaxial creep testing, rather than in outdated or domain-mismatched standards.
3. Model the complete processing–structure–property narrative of creep research (material identity, processing history, microstructure, test conditions, creep properties, and deformation mechanisms) within a single coherent, logically consistent schema.
4. Establish an ODP-based development workflow, reusing existing PMDco patterns where applicable and contributing novel patterns as reusable templates for future mechanical testing ontologies.
5. Deploy CTO as the semantic backbone of an LLM-based extraction pipeline that populates a FAIR-compliant creep KG, demonstrating, rather than merely asserting, that the ontology is usable in practice for both automated literature mining and KG construction.

3.2. Research Questions

The thesis is structured around three research questions that span mainly on ontology engineering, and its application for LLM-based extraction and knowledge graph construction:

- ***RQ1:*** *How can a BFO-compliant, ISO 204:2023-grounded ontology represent the full processing–structure–property knowledge of creep testing in a semantically consistent, logically sound, and interoperable manner?*

  This question targets the core ontology-engineering contribution, and asks whether the BFO/PMDco framework provides sufficient semantic coverage for the creep domain. It tracks whether the producing CTO's class hierarchy, relations, and axioms so that material identity, processing history, microstructure, test conditions, creep properties, and deformation mechanisms are represented within one logically sound, BFO/ISO 204-grounded schema. The RQ1 answer can be verified through concrete, executable methods; SPARQL queries built from CQs test semantic coverage, HermiT reasoning checks logical consistency and detects unsatisfiable classes, and structured expert review from supervisors and the PMDco community validates domain correctness and alignment with community standards.

- ***RQ2:*** *To what extent can an ODP-based development workflow (reusing existing PMDco patterns and contributing novel patterns) produce a template applicable to future mechanical testing ontologies?*
  RQ2 targets the ODP-based development methodology itself, whether reusing PMDco's existing patterns and contributing new ones yields a workflow generalizable beyond CTO. Each pattern can be isolated and tested individually, and its reusability can be assessed by checking whether the same patterns and workflow steps transfer cleanly to other mechanical-testing ontologies without requiring domain-specific redesign.
- ***RQ3:*** *To what extent can CTO support ontology-guided LLMs for extracting creep knowledge from unstructured scientific text, and constructing an ontology-driven creep literature knowledge graph?*
  This question targets CTO's practical usability as the semantic backbone for an applied pipeline; guiding LLM-based extraction from creep literature and organizing the results into a traceable KG. The extraction quality can be quantified against a human-annotated benchmark using precision, recall, F1-score, and entity-linking accuracy relative to a schema-free baseline, while the KG's validity can be checked through SHACL shape conformance and SPARQL queries derived from the CQs, confirming the graph is both correctly structured and queryable.

3.3. Research Methodology

The research methodology is organized to directly fulfill the five ontology-engineering objectives set out above and to answer the thesis's three research questions. Methodologically, the work divides into two complementary strands of CTO development and its application for an ontology-guided LLM extraction and KG construction pipeline.

3.3.1. Ontology development methodology

Crafting a domain-specific ontology necessitates adherence to rigorous methodologies shaped by extensive professional practice within the semantic engineering field. Various frameworks have achieved prominence, most notably the Ontology Development 101 approach [34], the NeOn Methodology [35], and the standard practices established by the Open Biomedical Ontologies (OBO) Foundry [33]. Despite their specific procedural variances, these methodologies converge on a fundamental sequence of operations: delineating the intended scope and utility, establishing CQs as verifiable requirements, synthesizing relevant domain terminology, architecting the taxonomic hierarchy and associative relations, formalizing the model within a specialized environment like Protégé, and validating the semantic integrity through both automated logical reasoning and targeted query execution.

Phase 1- Discovery: The Discovery phase establishes the purpose, scope, and functional requirements of the ontology. The first step is to define the ontology's purpose; to provide a BFO-compliant semantic framework for representing creep testing metadata of Inconel superalloys as extracted from the scientific literature. The scope is then bounded by the material system (Ni-based superalloys), the test type (uniaxial creep in tension), and the knowledge sources (peer-reviewed journal articles). CQs are formulated to capture the queries the ontology must support, serving as verifiable functional requirements throughout development. Terminology resources are collected from multiple complementary sources: (i) ISO 204:2023, which provides the normative vocabulary and procedural definitions for uniaxial creep testing; (ii) the domain creep literature itself, from which material-specific terms, property names, and microstructural descriptors are extracted; (iii) existing BFO-based ontologies (particularly PMDco, MTO, TTO, and MWO) which provide reusable classes and relations; and (iv) the data schema of the NFDI-MatWerk project [40], whose column structure for creep data reflects the main knowledge categories (source, material, processing, microstructure, test conditions, creep response). Finally, existing ontological assets are systematically reviewed to avoid duplication and identify opportunities for reuse and alignment.

Phase 2- Architecture: The Architecture phase defines the high-level semantic structure of CTO. The first architectural decision is the selection of BFO (ISO/IEC 21838-2:2021) as the top-level ontology. BFO provides a rigorously axiomatized, domain-neutral foundation that distinguishes between continuants (materials, specimens, microstructural features) and occurrents (creep tests, processing steps, measurement acts). This choice ensures interoperability with the broader BFO ecosystem and enables formal reasoning over the ontology. At the mid-level, PMDco is adopted as the primary bridge between BFO's abstract categories and the domain-specific concepts of creep testing. PMDco contributes reusable structures for processes, process chains, materials, devices, specifications, and values, all of which are directly applicable to the processing–structure–property paradigm of creep research. The class hierarchy is then designed by identifying the primary domain terms from the collected terminology and assigning each to the appropriate BFO category (e.g., 'creep test' as a planned process under BFO:Process, 'specimen' as a material entity under BFO:MaterialEntity, 'rupture time' as a temporal quality). This phase also defines the namespace[1] and the structure the repository by Ontology development kit (ODK)[2]. The Ontology Development Kit (ODK) was the infrastructure layer that tied everything together. It enforced the canonical repository layout, managed imports of external ontologies (BFO, PMDco, RO, IAO, OBI), automated quality control checks via ROBOT, generated multi-format releases (RDF/XML, Turtle, OWL Functional, JSON-LD, OBO), and integrated continuous reasoning validation into the GitHub Actions CI pipeline. The ODK is the recommended development environment for OBO Foundry ontologies, and its use in this thesis follows the precedent established by PMDco and MWO [40]. In this phase, we chose "CTO" as the prefix namespace of creep testing ontology, also registered a stable and resolvable Internationalized Resource Identifier (IRI) for CTO through the w3id.org community service, it is https://w3id.org/pmd/cto/.The registration of the namespace under the /pmd/ path explicitly integrates CTO into the Platform MaterialDigital semantic ecosystem, ensuring alignment with PMDco and associated mechanical testing ontologies like TTO. This design guarantees that entity IRIs—such as cto:CreepTest—remain permanently accessible regardless of changes in physical storage. Furthermore, all identifiers adhere to the OBO Foundry ID policy, utilizing meaningless numeric strings (e.g., CTO_0000001) to maintain stability across versions even as preferred human-readable labels are refined.

Phase 3- Modeling: The Modeling phase transforms the architectural design into a formal, machine-readable ontology. Formal class definitions (including labels, textual definitions, logical axioms, and restrictions) are first drafted in a structured spreadsheet (Excel template) to facilitate review and community feedback before implementation. These definitions are then imported into *Protégé*[3], the primary ontology editing environment, where classes, object properties, data properties, and annotations are created

---

[1] https://w3id.org/pmd/cto/
[2] https://incatools.github.io/ontology-development-kit/
[3] https://protege.stanford.edu/

and organized. A critical aspect of the Modeling phase is the reuse of PMDco ODPs. Conceptual visualization is supported by Miro[4] that map entity groups and relations before formal implementation. Miro served as the visual design tool during the early conceptual phase. Before any class was formalised in OWL, the main entity groups) material, specimen, processing route, microstructure, creep test, creep property, provenance source) were sketched as box-and-arrow diagrams on a shared Miro board. This informal layer acted as a bridge between the free-form discussions of scope and the rigid formalism of Protégé, and it proved particularly useful for communicating the ontology's structure to supervisors and collaborators who were domain experts but not ontology engineers. SHACL (Shapes Constraint Language) shapes are developed to define data-level constraints that complement the OWL axioms, ensuring that instance data populating the KG conform to the expected structure and value ranges.

Phase 4- Validation: The Validation phase ensures that CTO is logically consistent, semantically adequate, and practically useful. Automated reasoning with the HermiT reasoner[5] is applied to check logical consistency, detect unsatisfiable classes, and infer the complete class hierarchy. CQs are translated into SPARQL queries and executed against populated KG instances to verify that the ontology provides the necessary vocabulary and structure to answer each question. Expert review provides domain-level validation of the ontology's coverage, correctness, and alignment with community standards. Feedback from each validation cycle is fed back into earlier phases (Discovery through Modeling) in an iterative loop, refining the ontology until it meets all quality criteria. The comprehensive documentation generated by Widoco[6]. The final phase concludes with ontology release through GitHub[7], and publication under a Creative Commons Attribution 4.0 (CC BY 4.0) license[8]. GitHub provided version control, documentation hosting, and collaborative management. Every change to the ontology was tracked, every release was tagged, and the repository's README, issue tracker, and pull request workflow enabled the kind of transparent, auditable development process.

3.3.2. CTO application for constructing creep literature knowledge graph (CreepLitKG)

Once released, CTO functions as the semantic schema for a high-throughput literature mining pipeline. The core methodology transforms the LLM from an unconstrained text generator into a schema-aware extraction engine by incorporating CTO's ontological terms and definitions directly into the extraction prompt, constraining the model's output to a standardized vocabulary rather than ad-hoc labels. The pipeline processes a literature corpus of 100 scientific publications covering decades of Inconel creep research. Extraction performance is verified against a human-annotated gold-standard benchmark: a representative subset of 20 papers is manually annotated by a domain expert. The LLM outputs are evaluated against the gold-standard using precision, recall, F1-score.

An RML pipeline is reused to convert the extracted creep metadata to RDF triples conforming to CTO. The resulting RDF is loaded into a triplestore behind a public SPARQL endpoint. The resulting graph is evaluated along three lines: HermiT-based reasoning confirms the logical consistency of inferred knowledge; SHACL shapes validate the structural integrity of the populated triples against CTO's modeling constraints; and SPARQL queries derived from the CQs verify that the graph supports the intended scientific inquiries. Finally, the pipeline concludes with an interoperability test against the NFDI-MatWerk Knowledge Graph

[4] https://miro.com/
[5] https://www.hermit-reasoner.com/
[6] https://hosseinbeyginasrabadi.github.io/creep-testing-ontology/
[7] https://github.com/HosseinBeygiNasrabadi/creep-testing-ontology
[8] https://creativecommons.org/licenses/by/4.0/deed.en

(MSE-KG), confirming that CTO's BFO alignment allows the resulting creep data to federate with neighboring materials domains.

3.4. Novelty and Scientific Contribution

The contribution of this thesis lies at the intersection of three fields (ontology engineering, materials informatics, and LLM-based information extraction) and addresses a documented gap in each.

First, CTO is the first BFO-compliant domain ontology for creep testing of metallic materials aligned with ISO 204. While application ontologies exist for tensile testing (TTO), fatigue (FTO), and hardness (VTO, BTO), and while OntoFNCT addresses polymer creep, no BFO-aligned, standard-grounded ontology for metallic creep testing has been published. Furthermore, CTO develops a systematic ODP-based methodology and extends general patterns to be fed back into PMDco for reuse by other mechanical testing ontologies.

Second, the thesis establishes a reproducible, ODK-based ontology development pipeline that combines BFO top-level alignment, PMDco mid-level reuse, ODP-based modular modelling, SHACL constraint definition, and Widoco documentation, all under version control and continuous integration. This pipeline is documented in sufficient detail to serve as a methodological template for other domain-specific MSE ontologies.

Third, by integrating CTO with an LLM-based extraction pipeline, the thesis contributes to the emerging field of ontology-guided LLM extraction. Current LLM-based KG construction suffers from semantic inconsistency: extracted entities and relations are defined ad hoc, fragmented across studies, and impossible to align without manual curation. CTO provides the shared vocabulary that constrains LLM outputs to consistent, BFO-aligned concepts, enabling automated validation, SPARQL querying, and cross-study integration, capabilities that schema-free extraction cannot offer.

3.5. Further Research Output Applications

CTO and the resulting CreepLitKG are designed to support multiple downstream applications across the materials research data lifecycle.

FAIR creep knowledge graph construction: The primary application is the population of a BFO-compliant creep KG from the scientific literature. CTO provides the semantic schema; the LLM pipeline extracts the instance data. The result is a queryable, provenance-rich knowledge base covering 100 Nickel-based creep papers, accessible via a SPARQL endpoint. The use case can be easily extended to a higher number of publications, and also other materials than Inconel.

AI-ready datasets for machine learning: Because CTO enforces consistent terminology, value types, and units, the extracted data is directly usable as training input for machine learning models. Instead of spending weeks manually curating creep datasets from PDFs, researchers can query the KG for structured feature matrices (composition, grain size, γ' fraction, temperature, stress → rupture time, creep rate) with a single SPARQL query. This lowers the barrier to data-driven creep modelling and supports tasks such as remaining-life prediction, alloy design, and creep mechanism classification.

Semantic integration with materials data infrastructures: BFO and PMDco alignment ensure compatibility with the broader semantic MSE resources like those of Platform MaterialDigital ecosystem and the NFDI-MatWerk MSE-KG. This means creep data extracted by CTO-based pipelines can be federated with tensile, fatigue, and microstructure data from other PMD-aligned ontologies, enabling cross-property, multi-scale materials analysis that is currently impossible with siloed datasets.

Ontology reuse and extension: Because CTO is publicly available under CC BY 4.0 with full ODK-based repository structure and Widoco documentation, other researchers can reuse it directly, for example, by extending it to additional alloy systems (steels, aluminum alloys), additional creep test configurations (compression creep, bending creep), or additional characterization techniques (TEM, EBSD). The ODP-based architecture means that such extensions can reuse the same patterns and remain semantically aligned with the parent ontology.

Chapter 4: Creep Testing Ontology (CTO)

The results presented in this chapter follow the development pipeline (left-to-right sequence) shown in Fig 4.1 The pipeline begins with the scope definition, the decision that CTO would cover uniaxial creep testing under ISO 204 standard, and the review of existing ontologies that determined what could be reused. It moves through term identification, the collection of vocabulary from the standard, the related literature and data schema, and into the definition of classes and individuals, the point where each term in the spreadsheet became an OWL class with a label, a textual definition, and a parent in the BFO hierarchy. Relations between those classes were then formalised using RO properties, and SHACL shapes were developed to enforce data-level constraints on the KG instances to come. On the formalisation side, the pipeline runs from CQs, through the selection of BFO as the top-level foundation and the construction of the class hierarchy around the continuant–occurrent split, to the assignment of every domain concept to its TLO category and the addition of logical axioms. The sequence ends with evaluation, SPARQL queries derived from the CQs, and publication, the release of CTO on GitHub with persistent identifiers. The sections that follow walk through each of these steps in order, reporting what was done, what decisions were made, and what the resulting ontology looks like.

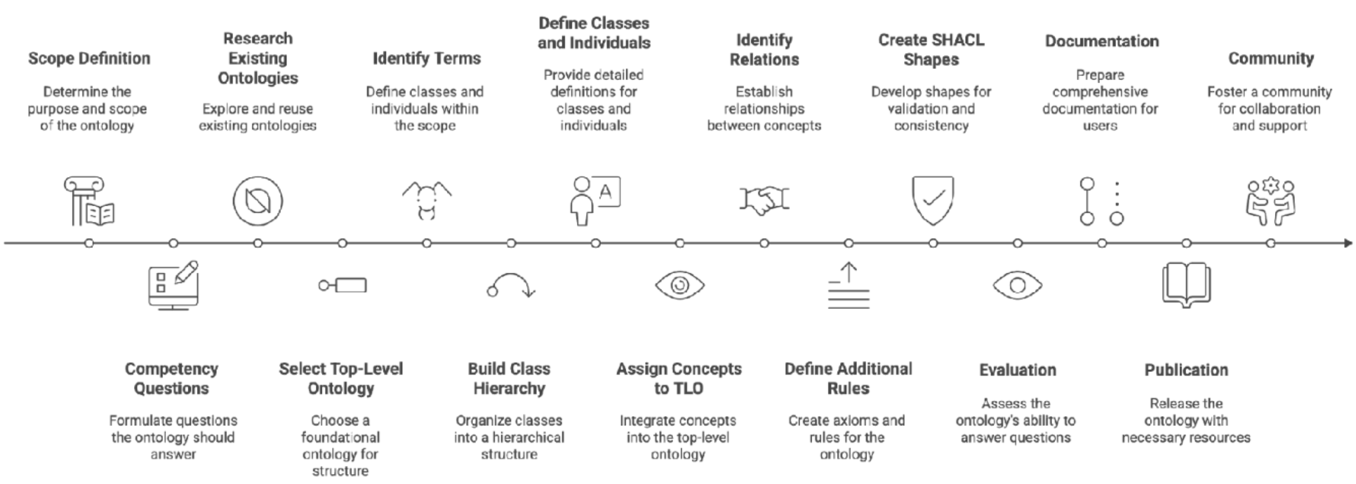

Fig 4.1. Ontology development pipeline followed in this project (left to right).

4.1. Ontology Scope

The Creep Testing Ontology (CTO) is an application ontology designed to model the full processing, structure, property, and equipment chain for uniaxial creep testing. CTO provides a standardized, explicit vocabulary to represent experimental creep data from the scientific literature, tracing the entire workflow from initial material history and microstructural characterization down to final test outcomes. CTO limits its boundary to uniaxial tensile creep tests standardized under DIN EN ISO 204 (ISO 204:2023). The ontology scope is built around four main interconnected domains centered on the core CTO hub, as illustrated in Fig 4.2.

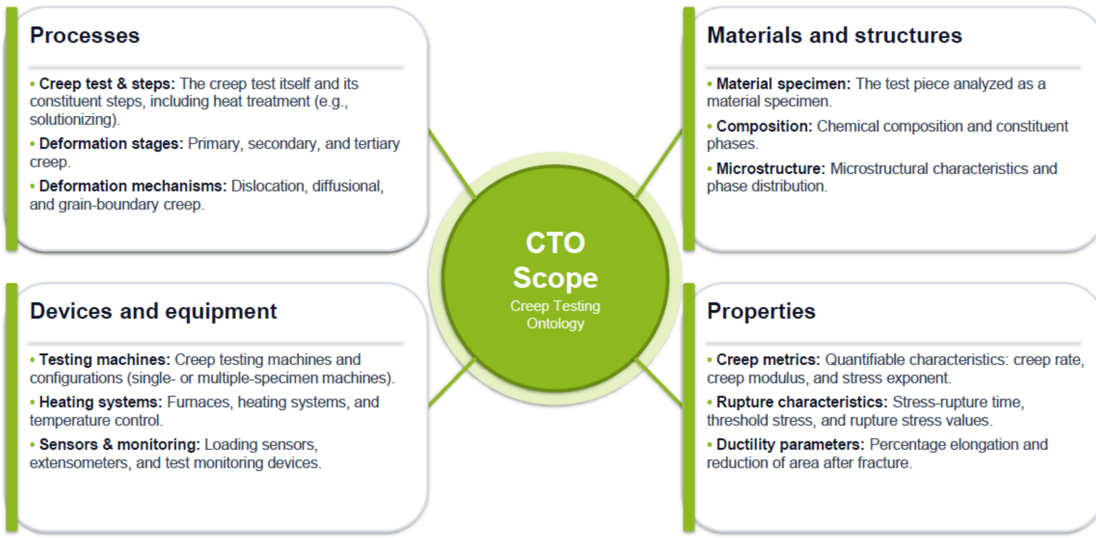

Fig 4.2. Scope of the Creep Testing Ontology (CTO), core domain modules and concepts covering testing processes, materials, properties, and equipment.

- **Processes:** Covers the actual creep test alongside its precursor steps, such as solutionizing and aging heat treatments that define the initial material condition. It represents the primary, secondary (steady-state), and tertiary creep stages, as well as deformation mechanisms like dislocation movement, diffusional flow, and grain-boundary sliding.
- **Materials and structures:** Represents the test specimen itself, captures chemical composition, and key microstructural features, including grain size, grain morphology, and precipitate phases, along with their volume fractions, dimensions, and spatial distributions.
- **Properties:** Focuses on quantifiable mechanical measurements and creep characteristics. This includes minimum creep rate, creep modulus, stress exponent, apparent activation energy, stress-rupture life, threshold stress, and post-test ductility metrics like percentage elongation and reduction of area.
- **Devices and equipment:** Describes the physical machinery and sensors used to run and track the experiment. It includes single- and multi-specimen testing frames (operating under constant load or stress), heating furnaces, temperature controllers, load cells, thermocouples, and extensometers.

In addition to experimental data, CTO captures provenance metadata such as publication authors, year, and source references, allowing every triple in the KG to trace directly back to its original paper. At the top level, CTO aligns with Basic Formal Ontology (ISO/IEC 21838-2:2021) and uses PMDco as a mid-level semantic bridge, maintaining compatibility with related materials ontologies like the Mechanical Testing Ontology (MTO), the Tensile Test Ontology (TTO), and the MatWerk Ontology (MWO).

4.2. Competency Questions (CQs)

A CQ is a natural-language question that a finished ontology must be able to answer; it expresses a functional requirement and is used to scope the ontology's content, guide its design, and later validate that the built ontology actually covers what it was meant to. A good CQ is a single, unambiguous, grammatically clear question rather than a compound question or a statement with a question mark appended. It must be answerable in principle from ontology content, it should avoid negation, vague comparators, and superlatives that resist translation into a query language, and it should target a defined level of granularity matched to its stage of use (scoping, validation, foundational alignment, relationships, or metaproperties). Completeness is judged by whether every requirement elicited from domain experts maps to at least one CQ, and whether every CQ is answerable; gaps in either direction signal the CQ set or the ontology needs revision [73], [74].
The CQs capture the end-to-end processing–structure–property–equipment chain of uniaxial creep testing, covering six primary thematic domains: provenance, equipment, processing history, material and microstructure, test conditions, and creep properties. Table 4.1 outlines the core CQs defined for CTO, along with their expected ontological representations and formal answers across these key categories.
These CQs collectively define the required semantic coverage of the CTO model. Following initial ontology design, population, and formal verification, the ontology will be evaluated against these questions in Section 4.8. This evaluation is conducted by translating each CQ into executable SPARQL queries against the populated KG to verify functional completeness, retrieval accuracy, and practical utility.

4.3. Reused Ontologies

CTO is not developed in isolation, it is built by reusing and extending several established ontologies at different levels of the semantic stack (Fig 4.3). This reuse strategy follows the principle that ontology development should maximize the reuse of existing, community-vetted resources rather than reinventing concepts, ensuring interoperability and reducing duplication of effort. All ontologies listed below are reused for developing CTO through the ODK import workflow and are included as extracted subsets containing only the entities actually referenced by CTO, ensuring that the combined import closure remains tractable for reasoning.

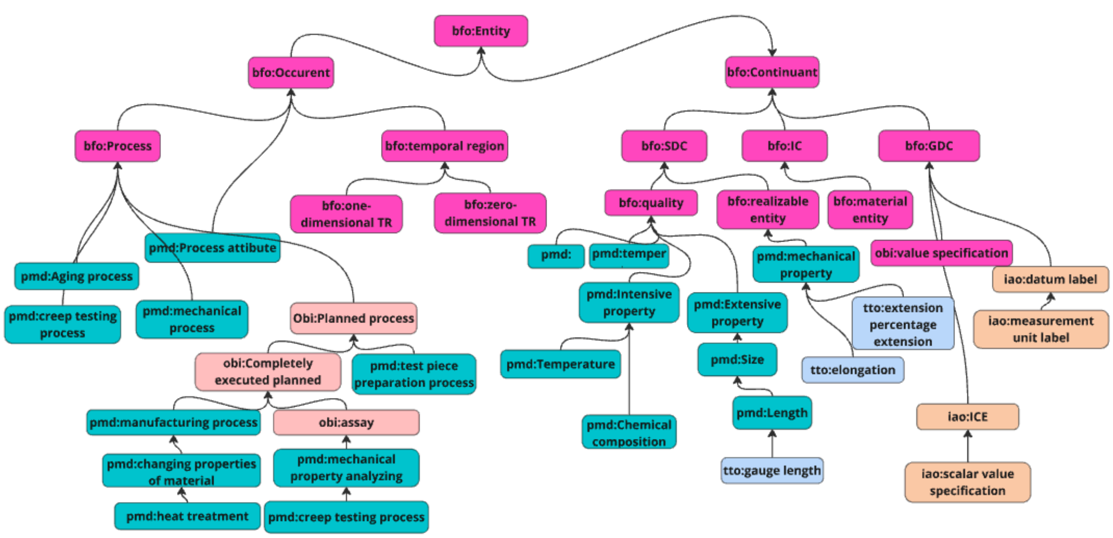

Fig 4.3. Reused ontologies for constructing CTO: TOP-level classes hierarchy basis on BFO.

Table 4.1. Competency Questions (CQs) and expected answers

#	Category	Competency Question	Expected answer
1	Equipment	Which types of creep test machines are distinguished in the ontology?	Single test piece machine; multiple test piece machine
2	Equipment	Which components are formally declared as part of a creep test machine?	Creep test machine heating system; creep test machine loading sensor
3	Equipment	What role does the testing equipment play during a creep test?	"Creep testing machine role"
4	Equipment	Which device is responsible for collecting the measurement signals (load, temperature, extension) during a creep test?	"Data acquisition system"
5	Equipment	What kind of device is used as the heating system and loading sensor in a creep test?	Heating system → furnace; loading sensor → load cell
6	Process	Which processes does the ontology relate temporally to the creep testing process?	manufacturing process, solutionizing heat treatment process, aging process
7	Process	What input conditions should be defined for the creep testing process?	Specified creep temperature datum; specified creep stress datum
8	Process	What is a creep test standard, and how does it relate to the test process it governs?	A standard that is concretized as a plan intended to realize the creep testing process (e.g. ASTM E139, ASTM E 8M-94a, ISO 204)
9	Material	What role does a test piece play during a creep test?	The test piece participates in the creep testing process and has the role "creep test piece role"
10	Material	How is a test piece's chemical composition related to the creep test it's used in?	Chemical composition → quality of a composition-quality → quality of the test piece → test piece participates in the creep testing process
11	Material	How is the grain size of a test piece connected back to the creep test it's used in?	Grain size value → quality of a crystallite → member of a microstructure → part of the test piece → test piece participates in the creep testing process
12	Material	What microstructural information does the ontology capture about a test piece?	Grain size value; delta/gamma/gamma-prime precipitate fraction delta/gamma/gamma-prime precipitate size va
13	Property	What characteristics describe how a material behaves during creep (e.g., creep rate, creep modulus, activation energy, stress exponent)?	Creep rate, creep modulus, creep activation energy, stress exponent
14	Property	Which types of extensions are distinguished as part of a specimen's deformation during creep?	Percentage creep, total, anelastic, elastic, initial plastic, initial total, plastic, and total ultimate extension
15	Property	Which physical mechanisms explain how a material deforms during a creep test?	Grain boundary creep mechanism; dislocation creep mechanism; diffusional creep mechanism
16	Property	Which types of creep stages (regions) are distinguished over the course of a test?	Primary, secondary, and tertiary creep region

Basic Formal Ontology (BFO): BFO (ISO/IEC 21838-2:2021) serves as the top-level ontological foundation of CTO. BFO provides the fundamental distinction between continuants and occurrents that structures the entire CTO class hierarchy. From BFO, CTO inherits core categories (e.g., Material Entity, Quality, Role, Process, Process Boundary, Temporal Region, and Generically Dependent Continuant) that are specialized for the creep domain.

Relation Ontology (RO): RO provides standardized object properties such as has_part, part_of, has_participant, preceded_by, and has_quality, which are used to express the structural and temporal relations between CTO entities [77]. These are the edges of the graph, and using standardised properties rather than inventing new ones is what makes cross-ontology querying possible. CTO imports ro:has_part and ro:part_of for meronomic relations (a specimen has a microstructure as a part, a grain boundary is part of a grain structure). ro:has_participant links processes to the entities that participate in them (a specimen participates in a creep test, a load cell participates in a loading process). ro:preceded_by encodes temporal ordering (aging is preceded by solutionizing, creep testing is preceded by specimen preparation). ro:has_quality links entities to their measurable attributes a specimen has a grain size, a precipitate phase has a volume fraction. These four properties alone express the majority of structural and temporal relations in CTO, and because they are defined by RO rather than invented locally, any other RO-using ontology can traverse CTO's graph without remapping.

Platform MaterialDigital Core Ontology (PMDco): PMDco (version 3.0) is the mid-level bridge, where its classes sit between BFO's abstract categories and CTO's domain-specific vocabulary. PMDco contributes the entire experimental workflow. pmd:ManufacturingProcess covers the primary shaping route, casting, forging, powder metallurgy. pmd:HeatTreatment groups solutionizing and aging under a common parent, with pmd:AgingProcess as a direct subclass. pmd:MechanicalProcess and “ProcessAttribute” provide the generic structures for linking a process to the attributes that characterise it, the bridge to the quality hierarchy. pmd:Temperature captures the thermal conditions of the test and the activation energy derived from it. The split between pmd:IntensiveProperty and pmd:ExtensiveProperty follows the thermodynamic distinction: pmd:ChemicalComposition is intensive, independent of specimen size; pmd:Size is extensive, scaling with the specimen, and pmd:Length inherits from it, covering reference length, gauge length, and every geometric parameter that ISO 204 defines. Under BFO:RealizableEntity, pmd:MechanicalProperty anchors the output side of the experiment. Rupture time, minimum creep rate, elongation, stress exponent, activation energy, each of these is a subclass of pmd:MechanicalProperty, inheriting the structure for linking a measured value to the material that bears it and the process that produced it.

Ontology for Biomedical Investigations (OBI): OBI contributes general-purpose classes for planned processes (OBI:planned process), specimens (OBI:specimen), and measurement devices (OBI:device), which CTO specializes for creep testing [33]. Despite its biomedical name, OBI contains several general-purpose classes that map directly onto the structure of a creep experiment, as the graph illustrates. obi:PlannedProcess is the parent for any protocol-driven procedure. A creep test designed according to ISO 204 is typed as obi:PlannedProcess at the point of experimental design, capturing what was supposed to happen. When the test is actually run and results are recorded, the same instance is additionally typed as obi:CompletelyExecutedPlannedProcess, marking the transition from intention to completed measurement. This dual typing is what enables CTO to distinguish between a test condition that was planned, 650 °C and 625 MPa according to the standard, and a test condition that was actually executed, 648 °C and 628 MPa as recorded by the instrument. Pattern 7, Planned vs. Measured Parameters (Section 4.7), depends on exactly this OBI distinction. The bridge between process and data runs through obi:assay. When the LLM extracts a creep property from a publication and maps it to CTO, the resulting iao:ScalarValueSpecification is linked to the obi:CompletelyExecutedPlannedProcess via the obi:assay relation. This single property encodes the most important provenance link in the entire graph: this value came from this test. Every rupture time, every creep rate, every elongation in the knowledge graph is connected to the process that produced it through obi:assay, making the full experimental chain traversable by SPARQL query.

Information Artifact Ontology (IAO): IAO supplies classes for information entities such as IAO:document, IAO:measurement datum, and IAO:data set, which CTO uses to represent publications, extracted data points, and dataset entries with their provenance metadata [75]. When the extraction pipeline pulls a rupture time out of a publication, the resulting value is typed as iao:ScalarValueSpecification. Above the individual values, iao:MeasurementDatum groups related scalar specifications into coherent data points. A single creep test produces multiple measurements, rupture time, minimum creep rate, elongation, each an iao:ScalarValueSpecification, and they are collected under a shared iao:MeasurementDatum that links them to the specific test run that generated them. At the top of the hierarchy, iao:Dataset aggregates measurement data across publications, forming the gold standard benchmark and the full extraction corpus as queryable collections. These IAO classes ensure that CTO can distinguish between the physical experiment and the information that describes it, a distinction that BFO's realism demands and that provenance tracking depends on.

NFDI MatWerk Ontology (MWO): MWO v3.0 is aligned with BFO and is developed as a modular extension of the NFDIcore v3.0 mid-level ontology, into which PMDco v3.0 is in turn integrated. Beyond representing MSE-specific instruments, facilities, materials, processes, and properties, MWO models the research data management (RDM) concepts related to MSE like peoples, organizations, resources, services, and educational activities these generate [40]. Within the present work, MWO functions as the semantic backbone of the MSE Knowledge Graph (MSE-KG), supplying the shared vocabulary through which CTO-typed creep data can be linked to the broader NFDI-MatWerk RDM landscape [40].

Tensile Testing Ontology (TTO): Regarding the closeness of the topics and covering some general mechanical properties, some of TTO classes were reused in CTO, like tto:gauge_length, tto:percentage_elongation, tto:percentage_extension.

4.4. Terminology Resources

The terminology populating the CTO was not established in a single iteration. Rather, it resulted from the synthesis of several distinct sources, each providing a unique perspective that required reconciliation into a semantically coherent vocabulary suitable for formal knowledge representation.

ISO 204:2023 standard: The primary and most authoritative source is ISO 204:2023, the international standard governing uniaxial creep testing in tension. The testing standard is the most important resource for preparing the ontology vocabulary, since it provides the specific names and definitions for the different concepts of the processes, and orients the structuring and categorization of test process entities based on the information that has already been agreed by an expert standardization committee [5]. In addition, developing the ontologies based on the materials testing standards ensures their acceptance and applicability in the industry [4]. The ISO 204:2023 document defines, in rigorous engineering language, the essential components of a creep experiment, including specimen geometries, thermal and mechanical loading protocols, and the requisite quantities for reporting. Normative terms such as creep strain, creep rate, and rupture time are adopted directly as textual definitions for the corresponding CTO classes.

Creep Reference Dataset Data Schema: The IUC02 team of NFDI-MatWerk, created a creep reference dataset data schema, that categories the main vocabulary (about source, material, processing, microstructure, and test response) related to creep testing datasets [72]. Aligning CTO terminology with these columns ensures that extracted literature data can be integrated into the MatWerk infrastructure without a complex translation layer. This practical alignment serves as a verification mechanism: terms that are essential for the community's research data management almost certainly warrant a formal representation within the ontology.

Existing creep ontologies: A further resource informing CTO's development was the small set of existing creep-related ontologies reviewed in Section 2.3: OntoFNCT (the Full-Notch Creep Test Ontology), the NIMS Creep Data Ontology, and the Creep Void Analysis Ontology. Unlike PMDco or BFO, none of these could be reused directly: none is built on BFO or aligned with a shared upper-level foundation, and each is scoped to a narrow test configuration, material class, or characterization task, so importing their modules or logical axioms wholesale would have introduced conflicting, non-interoperable commitments into CTO. These ontologies were therefore consulted only at the terminological level, relevant classes describing test parameters, materials, and metadata were identified and recreated as native CTO classes, re-defined and re-axiomatized within the BFO/PMDco framework, rather than imported together with their original axioms.

Creep literature: While ISO 204 provides a generic framework, it does not capture the nuances of alloy-specific behavior or the complex processing–structure–property relationships reported in the field. Domain-specific vocabulary, including heat treatment sequences and microstructural descriptors for strengthening phases was harvested iteratively. These terms were cross-referenced with existing semantic resources and refined based on the CQs to ensure the resulting KG captures the necessary conceptual depth for materials discovery.

Domain experts: CQs defined in Section 4.2 serve as the functional requirements for terminology selection. Each concept in CTO must contribute to answering at least one CQ, ensuring that the vocabulary remains lean and focused on practical research utility. Domain experts reviewed the CQs against the terminology already gathered from the reused resources to identify any additional concepts needed to answer a question but not derivable from ISO 204, the domain literature, or existing ontologies. If a query cannot be executed due to a missing concept, that term is prioritized for inclusion; conversely, interesting but irrelevant terms are deferred to future versions. This tight coupling transforms the ontology from an abstract catalog into a specialized tool for semantic inquiry.

Terminology acquisition was not a static phase but a dynamic conversation maintained throughout the development cycle. As terms were identified from literature, they were validated against ISO 204, mapped to BFO/PMDco classes, and tested against the CQs. The creep reference dataset schema provided the structural boundaries, while the CQs provided the filter for scientific relevance. The result is a vocabulary that is both standards-grounded and literature-informed, yet structurally compatible with broader materials informatics infrastructures. This multi-source methodology mirrors the terminology development process reported for MWO, where the conceptualization was driven by community requirements and the practical demands of constructing interoperable knowledge graphs [37].

Fig 4.4 shows part of the terminology spreadsheet that was used to formalise every term before any OWL code was written. Each row is a concept CTO needs to represent. Each column captures a different piece of its definition: the term itself, its textual definition, where that definition came from, what the equivalent class is in TTO or PMDco if one exists, where it sits in the class hierarchy, and what logical constraints apply to it.

Term	Definition	Definition source
# From ISO204		
reference length	Base length used for the calculation of either percentage elongation or pe	ISO 204:2023(en) Metallic materials
original gauge length	Length between gauge length marks on the test piece measured at ambie	ISO 204:2023(en) Metallic materials
extensometer gauge length	Distance between the measuring points of the extensometer.	ISO 204:2023(en) Metallic materials
parallel length	Length of the parallel reduced section of the test piece.	ISO 204:2023(en) Metallic materials
final gauge length after fracture	Length between gauge length marks on the test piece measured after frac	ISO 204:2023(en) Metallic materials
original cross-sectional area	Cross-sectional area of the parallel length as determined at ambient temp	ISO 204:2023(en) Metallic materials
minimum cross-sectional area after fracture	Minimum cross-sectional area of the parallel length as determined at amb	ISO 204:2023(en) Metallic materials
initial stress	Applied force divided by the original cross-section area, So, of the test pie	ISO 204:2023(en) Metallic materials
extension	Increase of extensometer gauge length, Le, at time t and at test temperatu	ISO 204:2023(en) Metallic materials
elongation	Increase of original gauge length, Lo, at time t.	ISO 204:2023(en) Metallic materials
percentage extension	Extension at test temperature expressed as a percentage of the reference	ISO 204:2023(en) Metallic materials
percentage elongation	Elongation expressed as a percentage of the reference length, Lr.	ISO 204:2023(en) Metallic materials
percentage elastic extension	Extension at test temperature expressed as a percentage of Lr, proportior	ISO 204:2023(en) Metallic materials
percentage initial total extension	Extension at test temperature as a percentage of Lr at end of loading with	ISO 204:2023(en) Metallic materials
percentage initial plastic extension	Extension at end of loading determined as difference between eti and ee.	ISO 204:2023(en) Metallic materials
percentage total extension	Extension at the test force at time t and at test temperature, as a percenta	ISO 204:2023(en) Metallic materials
percentage plastic extension	Extension at time t determined as difference between et and ee.	ISO 204:2023(en) Metallic materials
percentage total ultimate extension	Total extension at rupture and at test temperature, expressed as a percen	ISO 204:2023(en) Metallic materials
percentage creep extension	Extension at loading determined as difference between ep and ei.	ISO 204:2023(en) Metallic materials
percentage anelastic extension	Negative extension at end of unloading at test temperature.	ISO 204:2023(en) Metallic materials
percentage permanent extension	Extension at end of unloading determined as difference between et and s	ISO 204:2023(en) Metallic materials
percentage permanent elongation	Elongation expressed as a percentage of Lr at end of unloading and at ro	ISO 204:2023(en) Metallic materials
percentage elongation after creep fracture	Permanent elongation after fracture, Lu−L0, expressed as a percentage o	ISO 204:2023(en) Metallic materials
percentage reduction of area after creep fracture	Maximum change in cross-sectional area after fracture, S0−Su, as a perc	ISO 204:2023(en) Metallic materials
creep extension time	Time required to obtain a specified percentage creep extension, x.	ISO 204:2023(en) Metallic materials
plastic extension time	Time required to obtain a specified percentage plastic extension, x.	ISO 204:2023(en) Metallic materials
creep rupture time	Time to rupture for a test piece maintained at specified temperature and s	ISO 204:2023(en) Metallic materials
single test piece machine	Testing machine that permits straining of a single test piece.	ISO 204:2023(en) Metallic materials
multiple test piece machine	Testing machine that permits straining of more than one test piece simulta	ISO 204:2023(en) Metallic materials

Fig 4.4. Part of the terminology spreadsheet used during the discovery phase of CTO development.

4.5. Classes Hierarchy (Taxonomy)

To ensure interoperability and formal semantic rigor, CTO is structured as a three-tier ontological hierarchy. As shown in Fig 4.5, the architecture connects top-level formal concepts down to application-specific creep testing parameters through established mid-level materials ontologies. Building the hierarchy followed a strict sequence. Each term from the terminology spreadsheet was assigned to its BFO parent category, decisions made during architecture and validated against the CQs.

The class hierarchy of CTO follows the two main branches of BFO: Continuant and Occurrent.

CTO organises under continuant branch:

i- Material Entity covers the physical objects; Material for the alloy itself, Specimen for the test piece extracted from it, MicrostructuralConstituent for grains and precipitate phases.

ii- Quality covers the measurable attributes that inhere in those objects: CreepProperty for rupture time and creep rate, MicrostructuralFeature for grain size and precipitate fraction, TestCondition for temperature and stress.

iii- Role captures the functions entities take on: a Specimen bears a TestSpecimenRole during the test, and the distinction between PlannedParameter and MeasuredParameter.

iv- InformationContentEntity covers the documentary layer; Publication with DOI and authorship, DatasetEntry, MeasurementDatum, all subclasses of BFO:GenericallyDependentContinuant, capturing that a data point is not the physical measurement but information about it.

CTO organises under occurrent branch:

i- Process is the central class, like ManufacturingProcess, SolutionizingHeatTreatment, AgingHeatTreatment, TestPiecePreparation, CreepTestingProcess, HeatingProcess, and LoadingProcess.

ii- ProcessBoundary marks the instantaneous transitions: the start of loading, the moment of fracture.

iii- TemporalRegion captures the intervals: RuptureTime is the one-dimensional temporal region that is the duration of the loading process.

iv- ProcessAttribute that has no meaning independent of the process it characterizes: like for creep rate and activation energy.

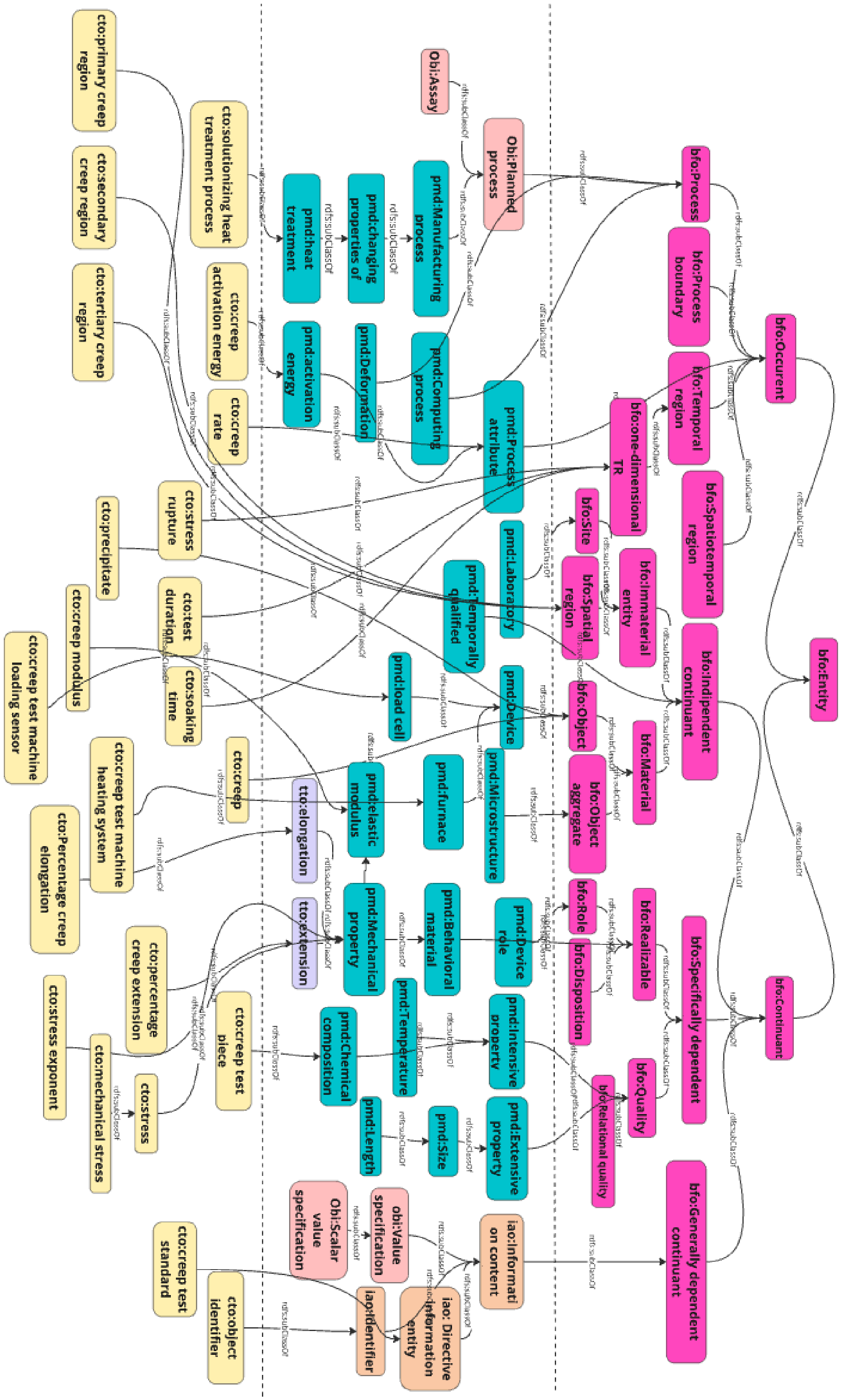

Fig 4.5. Hierarchical architecture of CTO.

Fig 4.6 shows part of the class hierarchy in Protégé. On the left, the class hierarchy browser displays the asserted taxonomy, each class placed under its BFO or PMDco parent, each carrying a label, a textual definition drawn from ISO 204 or the literature, and, where applicable, logical axioms that constrain how it can be used. On the right, the annotation panel for a single class, creep reference dataset, shows the full set of metadata that every CTO class carries: an rdfs:label, a skos:definition written in natural language, an equivalent class expression, subclass axioms, and general class axioms. This is the pattern that repeats for every class in the ontology. Taxonomy without definitions is just indentation. Definitions without axioms are just comments.

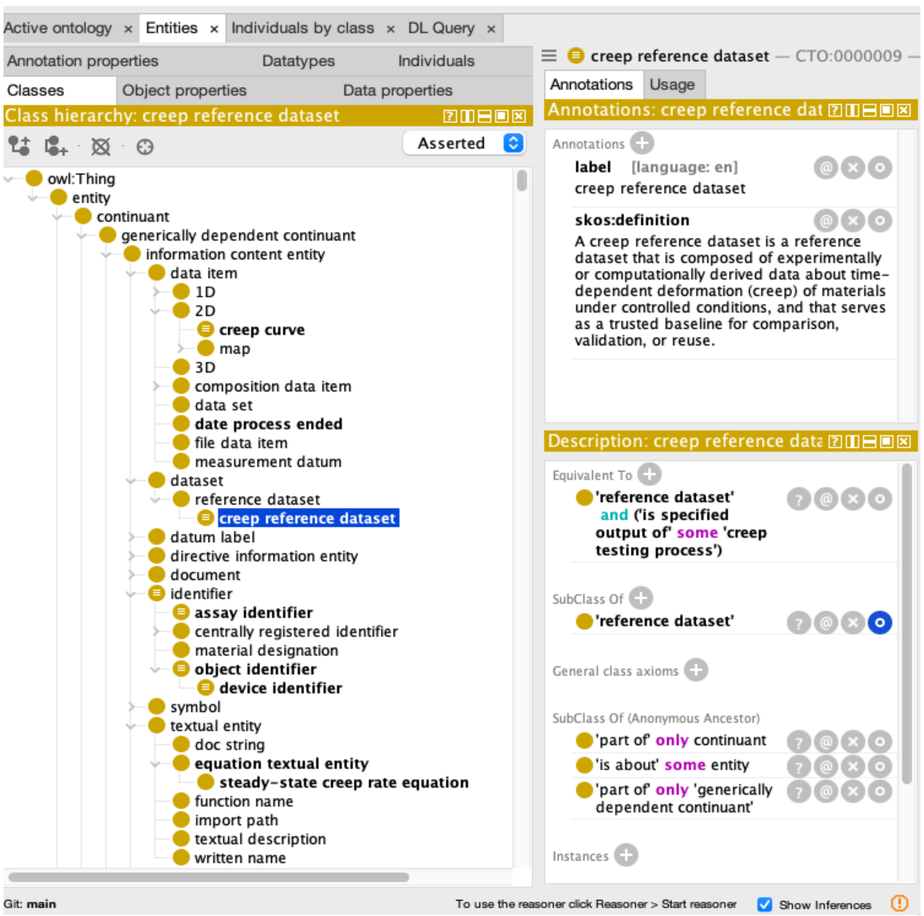

Fig 4.6. A representative CTO classes hierarchy and annotation in Protégé.

4.6. Axioms and Ontology Design Patterns (ODPs)

A taxonomy tells you what things are called and how they nest. An ontology tells you how they relate to each other and what must be true for the whole structure to make sense. Moving from one to the other requires axioms, logical statements that constrain how classes and properties can be combined, and writing good axioms from scratch is error-prone. This is where Ontology Design Patterns (ODPs) earn their place. An ODP does not just give you a class hierarchy, it gives you the relations that hold between those classes, the restrictions that govern them, and the inferences that follow from them, all packaged as a tested, reusable template.

The axioms required to formally represent CTO were derived through a requirement analysis grounded in the ontology's defined scope (Section 4.1) and its competency questions (Section 4.2). Rather than being authored ad hoc, each axiom traces back to a specific CQ or scope statement it is intended to satisfy, ensuring that every formal commitment in the ontology serves a documented purpose. This analysis grouped the resulting axiom requirements into four categories, mirroring the thematic structure of the CQ table (Table 4.1):

I- Processes axioms and capturing their temporal and causal relationships: *Pattern 1*, Process Chain, models the full experimental sequence, manufacturing, solutionizing, aging, specimen preparation, creep testing, as an ordered series of steps, each linked to the next by temporal precedence axioms. This directly addresses CQ6, what processing history did the sample undergo, and CQ8, how can a reported property be traced back through that history. The pattern provides the structural axioms: every process has an input and an output, every output of one process is the input of the next, and the chain is temporally ordered.

II- Materials and equipment axioms: *Pattern 2*, Chemical Composition, links a material to its elemental constituents, each with a concentration value and measurement unit. *Pattern 3*, Microstructure, the first of the two novel patterns, links a specimen to its grain structure and precipitate phases, each characterised by size, fraction, and morphology. Together they answer CQ10, what is the material and its composition, and CQ12,CQ11, what microstructural features are reported. The Microstructure pattern provides the axioms that a specimen has exactly one grain structure and that precipitate characteristics inhere in the specimen, not in the abstract material, a distinction that matters when multiple specimens from the same alloy receive different heat treatments.

III- Properties axioms for capturing the creep-response quantities: *Pattern 4*, Mechanical Property, models the six creep outputs demanded by CQ3, rupture time, minimum creep rate, elongation, reduction of area, stress exponent, activation energy, each as a quality of the tested specimen with a specified value type and unit. The pattern provides the axiom that every mechanical property is realised through some testing process, linking the output to the experiment that produced it.

IV- Process-Structure-property axioms: *Pattern 5*, Temporal Region, models rupture time as the duration of the loading process, providing the axiom that connects a temporal measurement to the occurrent it measures. *Pattern 6*, Process Attribute, models minimum creep rate as a quality of the ongoing deformation process, providing the axiom that distinguishes a process attribute from a terminal property. *Pattern 7*, Planned vs. Measured Parameters, the second novel pattern, models each test condition as a pair, a planned value linked to the standard and a measured value linked to the instrument, providing the axioms that prevent a planned temperature of 650 °C from being treated as identical to a measured temperature of 648 °C. Together, Patterns 5, 6, and 7 answer CQ1-5, under which conditions was the sample tested, and CQ7, which samples can be compared because they were tested under genuinely similar conditions.

Viewed this way, the planned patterns form a single semantic network.: processes act on materials, materials bear properties, and properties are measured by processes. For most of the common materials science scenarios mentioned above, the PMDco library [3]provides a set of BFO-compliant ODPs. One of the main research questions in this project is to evaluate whether PMDco ODPs are enough for representing CTO axioms, and what patterns are required that have not already been modeled by PMDco. The ODPs provide the axioms that make those relationships logically explicit, and because the patterns were adapted from PMDco rather than invented locally, those axioms are consistent with every other ontology in the Platform MaterialDigital ecosystem.

4.6.1. Pattern 1: Process Chain

Purpose: Represent processes, consisting of simultaneous and serial subprocesses.

Example Use Case: The process chain pattern is the foundational structural pattern of CTO, as it models the entire experimental workflow from raw material to final creep measurement. The important part is that the creep properties depend on the full chain of prior steps: how the material was manufactured, how it was heat-treated, how the test specimen was prepared, and how the creep test itself was conducted. The example use case for this pattern is describing the processes before the creep testing process (such as the primary manufacturing route, casting, forging, or powder metallurgy, followed by solutionizing heat treatment and single- or multi-step aging heat treatment, and finally specimen machining and preparation) and the creep testing process itself that consists of a number of simultaneous and serial subprocesses (such as heating the specimen to the target test temperature, applying the constant mechanical load via the loading process, and continuously monitoring strain accumulation throughout the primary, secondary, and tertiary creep stages).

Related PMDco ODP:
https://github.com/materialdigital/core-ontology/tree/main/patterns/process%20chain

How pattern extended to CTO: As seen in Fig 4.7, the process chain pattern was extended with the following domain-specific process classes: Manufacturing Process; Solutionizing Heat Treatment Process, Aging Heat Treatment Process, Test-Piece Preparation Process, Creep Testing Process, Heating Process, and Loading Process. By adding these subclasses under the appropriate BFO and PMDco parent classes (e.g., Creep Testing Process as a subclass of PMDco:Process and ultimately BFO:Process), CTO preserves full semantic alignment with the upper ontology while providing the domain-specific vocabulary needed for literature-based knowledge extraction. The temporal relations between processes are modeled using BFO:preceded_by and BFO:has_participant, ensuring that the complete experimental chronology (from manufacturing through aging to creep testing) is explicitly represented.

Is PMDco ODP completely covering the purpose? Yes.

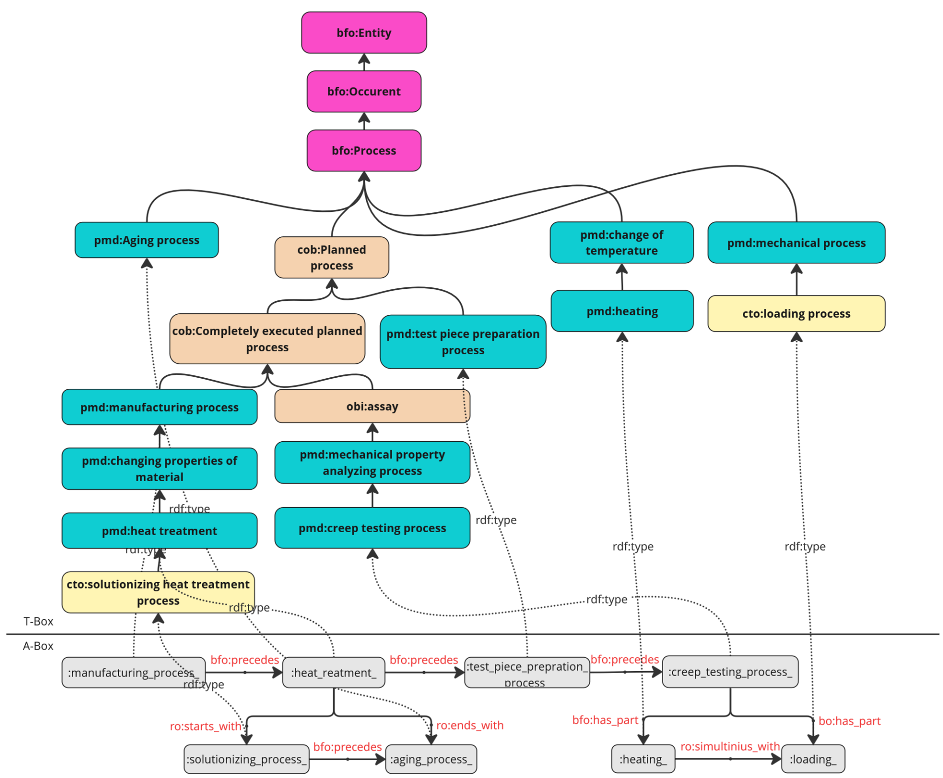

Fig 4.7. Pattern #1, for "Process Chain".

4.6.2. Pattern 2: Chemical Composition

Purpose: Description on how to represent the chemical composition of a (typical metallic) material

Example Use Case: In the creep testing domain, chemical composition is a critical material variable: small variations in key alloying elements such as Al, Ti, Nb, Mo, and Cr directly govern the volume fraction, morphology, and thermal stability of strengthening precipitates (γ′, γ, δ), which in turn control high-temperature creep resistance and rupture life. Competency question CQ10 explicitly requires the ontology to represent "How is a test piece's chemical composition related to the creep test it's used in?" making this pattern indispensable for correlating alloy chemistry with experimental creep performance.

Related PMDco ODP:
https://github.com/materialdigital/core-ontology/tree/main/patterns/chemical%20composition

How pattern extended to CTO: As seen in Fig 4.8, the base PMDco chemical composition pattern provides the core structure by linking a bfo:MaterialEntity (or pmd:Material) to its constituent chemical element entities through a composition quality relation (pmd:ChemicalComposition), with each element's concentration value (typically weight percent) represented via a obi:ScalarValueSpecification (or pmd:ValueObject). For CTO, this pattern was extended to accommodate the specific reporting conventions of the Inconel creep literature. Specifically, the pattern was extended to distinguish between nominal composition (the standardized alloy specification provided by manufacturers or standards, e.g., nominal IN718) and measured composition (the actual chemical composition determined by experimental analysis of the specific test sample or heat). This distinction is vital because creep publications frequently report nominal compositions for alloy identification alongside measured compositions for precise property correlation. Additionally, the pattern was extended to support partial composition reporting (a common occurrence in published literature where only key alloying elements (such as Ni, Cr, Fe, Nb, Ti, and Al) are listed)without requiring all trace elements to be present, ensuring the ontology does not impose unrealistic completeness constraints during automated data extraction.

Is PMDco ODP completely covering the purpose? Partially. While the base PMDco ODP provides the fundamental structural link between a material entity, its chemical elements, and concentration values, it required extension in CTO to explicitly represent the distinction between nominal vs. measured compositions and to support partial elemental reporting typical of literature datasets.

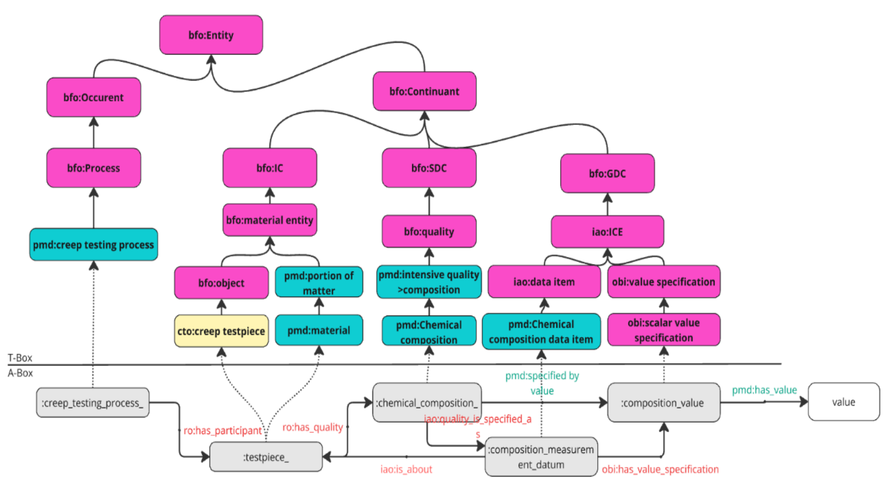

Fig 4.8. Pattern #2 for chemical composition.

4.6.3. Pattern 3: Microstructure

Purpose: Represent microstructural features, including grains and precipitate phases, along with their quantitative physical characteristics (such as phase size, volume fraction, and spatial distribution) and their relationship to the parent material specimen.

Example Use Case: In creep research, microstructure serves as the central physical mediator between processing history and mechanical properties. The primary rationale for designing heat treatments (such as solutionizing and aging) is to engineer a specific microstructural state that imparts high-temperature creep resistance. Competency questions CQ11, and CQ12 explicitly require the ontology to represent "what microstructural features are reported for the sample, such as grain size and precipitate characteristics", and to link these features to processing parameters and resulting creep behavior. The example use case for this pattern is describing the initial microstructural state of an Inconel specimen prior to creep testing, including grain structure (average grain size and morphology) and constituent precipitate phases (δ, γ′, and γ phases, along with their respective volume fractions and average precipitate dimensions).

Related PMDco ODP: None (Not available in the core PMDco pattern library; newly developed as an original contribution in CTO).

How pattern extended to CTO: As illustrated in the T-Box/A-Box architecture diagram (Fig 4.9), the Microstructure pattern models two core families of microstructural entities as bfo:MaterialEntity subclasses that are part of the test specimen (:creep_testpiece_ --ro:has_part--> :microstructure_):

1. Grain Structure: Represented by individual crystallite members (:grain_, inheriting from pmd:crystallite under pmd:crystal and bfo:object), which bear extensive physical qualities such as average grain size (:grain_size_, inheriting from pmd:size under bfo:quality).
2. Precipitate Phases: Represented by specific precipitate members (cto:precipitate subclasses: cto:delta, cto:gamma, cto:gamma prime), which are linked to the microstructure instance via ro:has_member. Each precipitate phase is modeled as bearing two types of quantitative characteristics:
    - Extensive Qualities: Physical dimensions such as precipitate size (:delta_precipitate_size_, :gamma_precipitate_size_, :gamma_prime_precipitate_size_), which inherit under pmd:size.
    - Relational Qualities: Relative phase proportions such as volume fractions (:delta_precipitate_fraction_, :gamma_precipitate_fraction_, :gamma_prime_precipitate_fraction_), which inherit under pmd:volume proportion via pmd:has_relational_quality.

By anchoring these microstructural entities directly to BFO continuant classes and linking them to scalar value specifications, CTO establishes a formal, quantitative representation of multi-phase microstructures that bridges material processing with observed creep deformation.

Is PMDco ODP completely covering the purpose? No. PMDco version 3.0 lacks a dedicated ontology design pattern for multi-phase microstructures and quantitative characterization results. The Microstructure pattern designed for CTO fills this major gap in the materials science ontology landscape. Built strictly according to BFO alignment principles and PMDco naming conventions, this pattern represents an original methodological contribution that can be generalized to other mechanical testing domain ontologies (such as tensile, fatigue, and hardness testing) and proposed for future integration into the core PMDco pattern library.

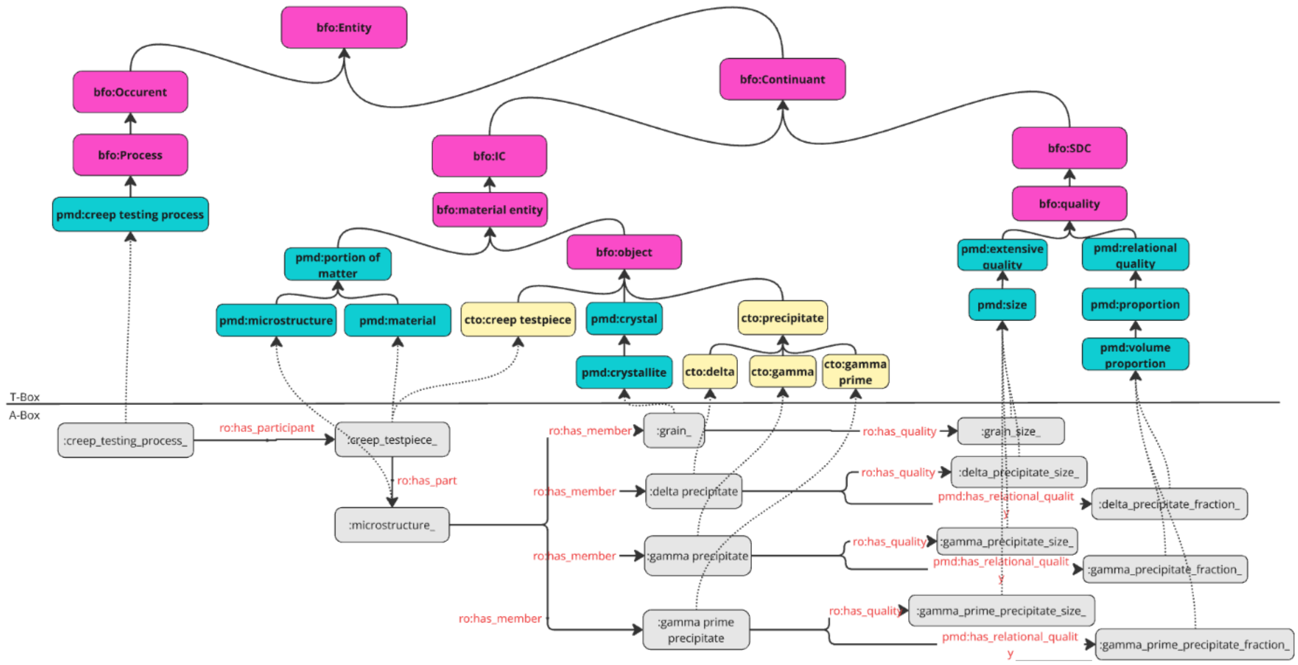

Fig 4.9. Pattern #3 for microstructure. A novel ODP designed in this thesis; no corresponding pattern exists in the PMDco library. The T-Box (upper panel) defines the class structure: a creep test piece is linked to its microstructure via ro:has_part, with grain structure and precipitate phases (δ, γ′, γ) modelled as quality-bearing entities, each characterised by size and fraction values. The A-Box (lower panel) shows an instantiated example populated with data from a specific specimen. This pattern was designed with generic labels so that it can be reused across alloy systems beyond Inconel.

4.6.4. Pattern 4: Mechanical Property

Purpose: Represent measured mechanical properties (such as strain extension and percentage creep extension) as physical qualities that inhere in a specimen participant, are realized during a testing process, and are quantified by scalar value specifications with QUDT unit annotations.

Example Use Case: In creep testing, the primary objective of an experiment is to measure and quantify how a material specimen deforms over time under load. Competency question CQ14 explicitly requires the ontology to represent deformation extension types ("which types of extensions are distinguished as part of a specimen's deformation during creep?"). The example use case for this pattern is representing percentage creep extension (cto:percentage creep extension) measured on an Inconel restpiece specimen (:creep_testpiece_) during a creep test (:creep_testing_process_), linking the mechanical property quality to its scalar value specification (:percentage_creep_extension value specification), numerical magnitude (value), and QUDT percentage unit label (qudt:PERCENT).

Related PMDco ODP:
https://github.com/materialdigital/core-ontology/tree/main/patterns/material%20property%20(quality)

How pattern extended to CTO: As illustrated in the T-Box/A-Box architecture diagram (Fig. 4.10), Pattern 4 extends the base PMDco property pattern to model mechanical deformation qualities across three BFO continuant branches:

1 - Specimen & Participant Branch (bfo:IC): The material test piece (cto:creep testpiece → bfo:object → bfo:material entity → bfo:IC) participates in the creep testing process (:creep_testing_process_ → ro:has_participant--> :creep_testpiece_).

2 - Mechanical Property Quality Branch (bfo:SDC): The mechanical deformation property is modeled under bfo:SpecificallyDependentContinuant (bfo:SDC), specializing through realizable dispositions, material mechanical properties (pmd:mechanical property), and Tensile Test Ontology extension classes (tto:extension → tto:percentage extension) down to cto:percentage creep extension. At the instance level, :percentage_creep_extension:

- Inheres in the specimen participant via ro:has_characteristic (:creep_testpiece_ --ro:has_characteristic--> :percentage_creep_extension).
- Is realized during the creep test process via bfo:realizes (:creep_testing_process_ --bfo:realizes--> :percentage_creep_extension).

3 - Scalar Value & Unit Specification Branch (bfo:GDC): The numerical result is captured under bfo:GenericallyDependentContinuant (bfo:GDC → iao:ICE). The property instance is linked via pmd:specified_by_value to :percentage_creep_extension value specification (instance of obi:scalar value specification), which carries the numerical magnitude via pmd:has_value and its measurement unit via iao:has_measurement_unit_label → qudt:PERCENT (instance of iao:measurement unit label).

Is PMDco ODP completely covering the purpose? Partially. PMDco supplies the underlying generic architecture for connecting material entities, mechanical properties, and scalar value specifications. However, CTO required extension of this pattern to integrate Tensile Test Ontology extension hierarchies (tto:extension → tto:percentage extension), define domain-specific creep extension classes (cto:percentage creep extension), and enforce QUDT unit bindings (qudt:PERCENT) for standardized knowledge graph extraction.

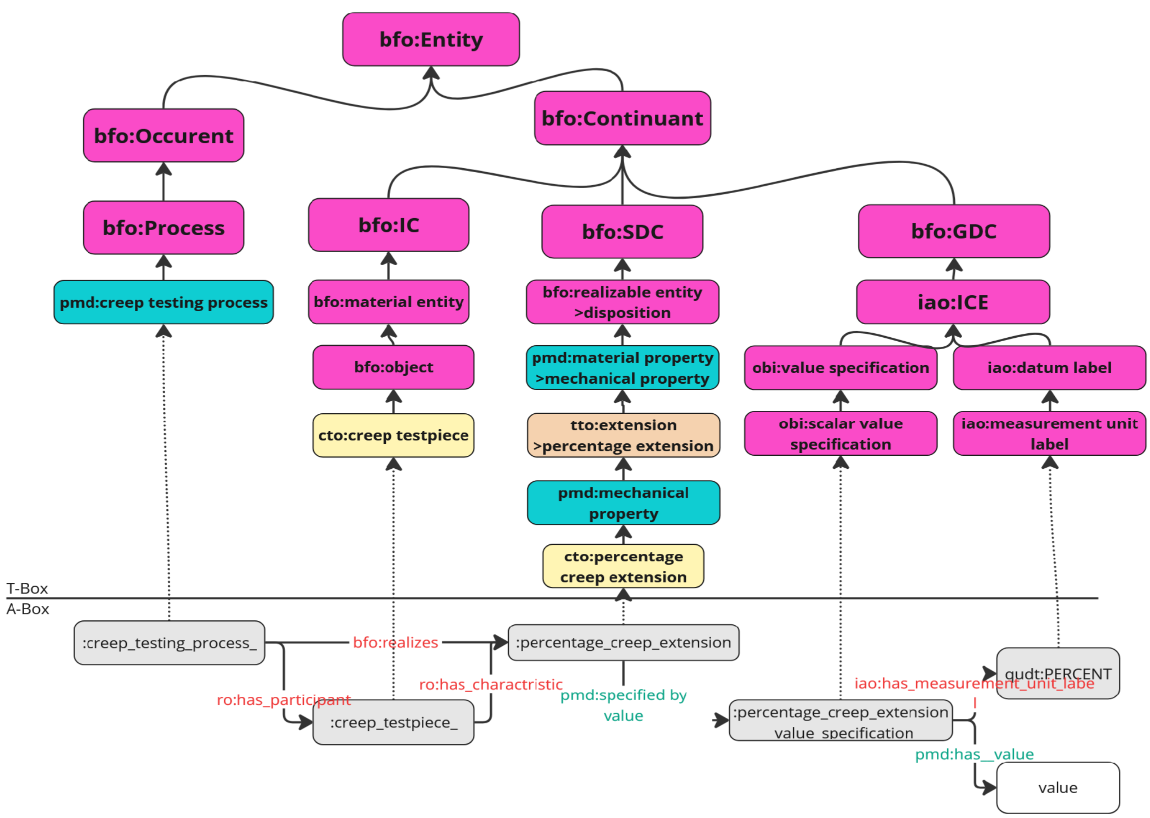

Fig 4.10. Pattern #4 for mechanical property.

4.6.5. Pattern 5: Creep Rupture Time

Purpose: Represent stress-rupture time and process durations not merely as static property values, but as discrete, one-dimensional temporal regions (bfo:one-dimensional TR) that bound active mechanical processes over time, bounded by zero-dimensional temporal instants (bfo:zero-dimensional TR) representing exact starting and ending points.

Example Use Case: Creep rupture time (tr) is the total elapsed time from initial load application to specimen fracture, serving as a critical performance metric for a material's long-term resistance to high-temperature failure. Competency question CQ13 explicitly mandates the representation of stress-rupture time. The example use case for this pattern is representing the stress-rupture life of an IN718 specimen tested at 650 °C and 600–700 MPa, linking the active loading process (:loading_process_) directly to its occupied one-dimensional temporal region (:stress_rupture_time_), which is bounded by its starting instant (:start_loading), ending instant (:end_loading), and quantified numerical duration value (:creep_rupture_time_value).

Related PMDco ODP: None (Not available in the PMDco pattern library; PMDco models properties as static qualities rather than bounding process lifespans as native temporal entities).

How pattern extended to CTO: While PMDco provides extensive patterns for static material properties, it lacks a native design pattern for bounding process lifespans as discrete, queryable temporal entities. Therefore, rather than forcing a structural property pattern from PMDco, CTO leverages the native temporal architecture of the Basic Formal Ontology (BFO). As illustrated in the T-Box/A-Box architecture diagram (Fig 4.11), the pattern is modeled across three interconnected pillars:

Process-Temporal Occupancy: An active mechanical loading process (:loading_process_, inheriting under cto:loading process → pmd:mechanical process → bfo:process) is linked to its temporal span via bfo:occupies_temporal_region. Temporal Region Bounding: The rupture duration is declared as an instance of cto:stress rupture time (specializing bfo:one-dimensional TR under bfo:temporal region). This temporal region is formally bounded by zero-dimensional time points via:

bfo:has_first_instant → :start_loading (instance of bfo:zero-dimensional TR). bfo:has_last_instant → :end_loading (instance of bfo:zero-dimensional TR).

Scalar Value Attachment: The temporal region (:stress_rupture_time_) is connected to its numerical magnitude and unit (e.g., hours or seconds) via pmd:specified_by_value →:creep_rupture_time_value (instance of iao:scalar value specification under iao:ICE → bfo:GDC).

By modeling rupture time as a temporal region rather than an isolated property quality, CTO semantically binds the temporal duration directly to the loading process, which in turn links to the creep test, specimen, processing history, and microstructure. This enables complex SPARQL queries such as "retrieve all rupture times for IN718 specimens tested at 650 °C and 600–700 MPa" with complete provenance traceability across the knowledge graph.

Is PMDco ODP completely covering the purpose? No. PMDco lacks a native design pattern for representing process durations as formal temporal regions. CTO solves this by utilizing BFO's native upper-ontology temporal region framework (bfo:temporal region, bfo:one-dimensional TR, bfo:zero-dimensional TR), establishing a semantically rigorous temporal model for process lifespans that preserves full traceability and enables precise time-dependent SPARQL querying across literature datasets.

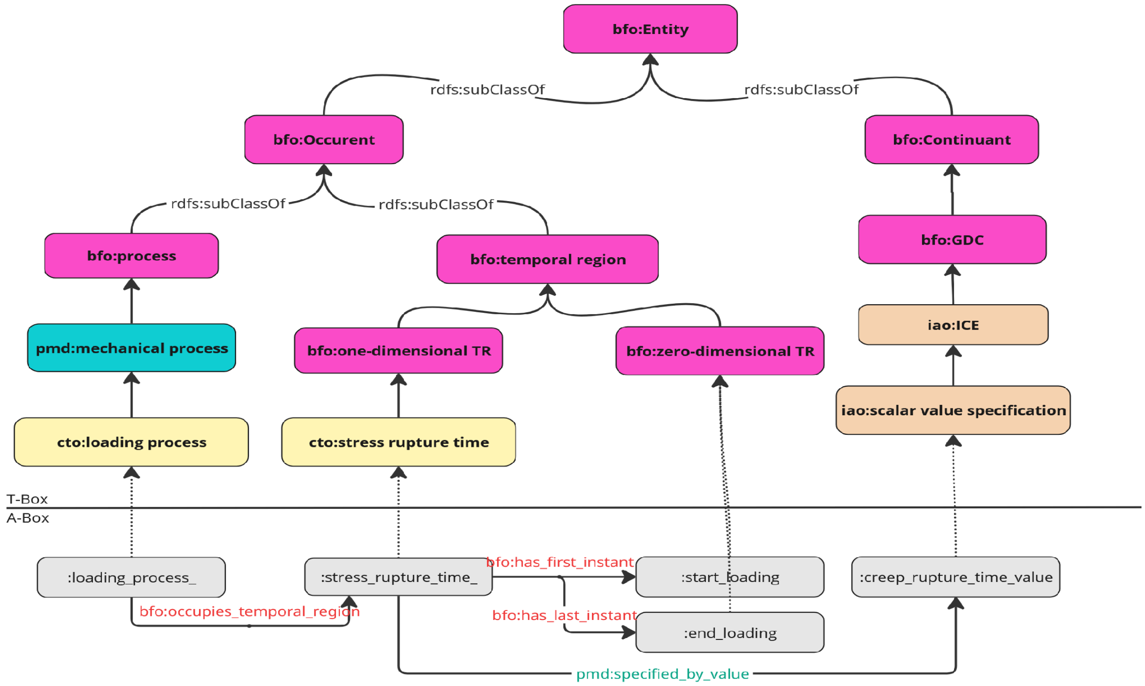

Fig 4.11. Pattern #5 for creep rupture time.

4.6.6. Pattern 6: Process Attribute — Creep Rate

Purpose: Represent minimum (steady-state) creep rate as a dynamic process attribute that inheres in the ongoing secondary creep deformation stage, rather than treating it merely as a static post-hoc terminal property.

Example Use Case: The minimum or steady-state creep rate ($\dot{\varepsilon}_{min}$) is a fundamental parameter in high-temperature engineering design, characterizing the rate of strain accumulation during the extended secondary creep regime before tertiary creep acceleration and rupture occur. Ontologically, the minimum creep rate is a process quality that inheres in the active deformation process itself over a specific time interval, rather than a static property of the specimen. Competency question CQ6 explicitly requires the representation of creep rate, while CQ16 mandates distinguishing between different creep stages over the course of a test. The example use case for this pattern is modeling the steady-state strain rate ($\dot{\varepsilon}_{min} = 1.2 \times 10^{-7}$ s^{-1}) determined over the secondary creep stage (e.g., between 100 and 500 hours) of an Inconel 718 creep test, linking the rate quality directly to the secondary creep process segment (cto:secondary creep region), its parent testing process (cto:creep testing process), and its scalar value specification.

Related PMDco ODP: https://github.com/materialdigital/core-ontology/tree/main/patterns/process

How pattern extended to CTO: The PMDco process attribute pattern provides the generic structural template: a quality entity that inheres in a process (bfo:inheres_in), with its value quantified through a scalar value specification (obi:ScalarValueSpecification). For CTO, this base pattern was specialized across three main domain features (Fig. 4.12):

1. Process Quality Specialization: Defining cto:creep rate as a direct subclass of pmd:ProcessAttribute (and bfo:Quality / bfo:SpecificallyDependentContinuant), with unit bindings restricted to reciprocal time units (qudt:PER-SEC or qudt:PER-HR).
2. Creep Stage Association: Linking the creep rate quality directly to the specific secondary creep

deformation segment (cto:secondary creep region / bfo:Process), which exists as a temporal part of the overall creep testing process (bfo:has_part / bfo:precedes).

3. Interval Segment Traceability: Capturing the precise temporal segment or time window from which the creep rate was calculated (e.g., secondary regime interval), acknowledging that reported minimum creep rates in literature are engineering approximations determined over finite testing intervals.

By modeling minimum creep rate as a process attribute attached to specific creep stages, CTO enables the knowledge graph to distinguish between strain rates measured across different stages of the same test (primary, secondary, tertiary), supporting nuanced SPARQL queries regarding creep curve shape, secondary stage duration, and deformation stage transitions.

Is PMDco ODP completely covering the purpose?: Partially. PMDco supplies the underlying generic architecture for connecting process entities to process attributes and scalar value specifications. However, CTO required extension of this pattern to define specific creep rate subclasses (cto:creep rate), enforce reciprocal time unit constraints (s^{-1} or h^{-1}), and bind the attribute directly to sub-process deformation regimes (cto:secondary creep region) for detailed stage-by-stage creep curve modeling.

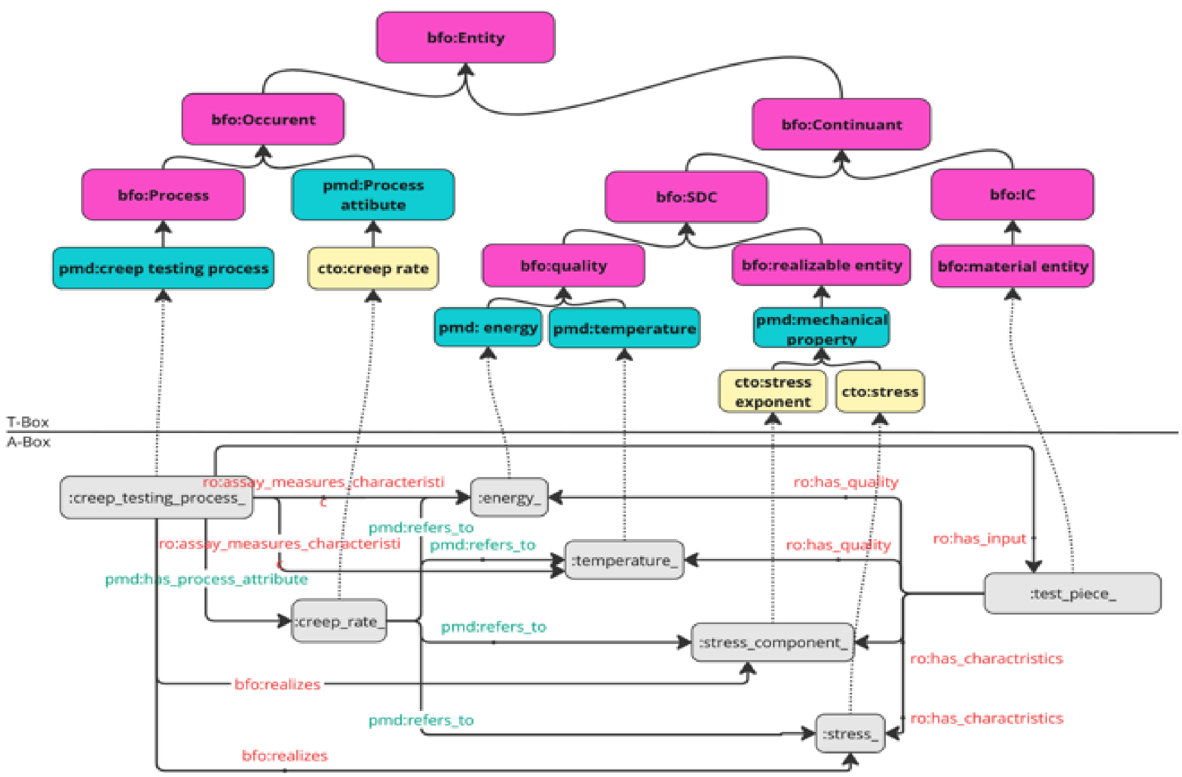

Fig 4.12. Pattern #6 for creep rupture time and its formal semantic linkage to thermal activation energy, stress exponent, and specimen input.

4.6.7. Pattern 7: Parameters — Planned vs. Measured Values

Purpose: Distinguish between intended experimental target parameters (planned test conditions specified by experimental design or test standards like ISO 204) and actual recorded test parameters (measured test conditions captured during process execution), preventing ambiguity during literature-based data extraction.

Example Use Case: In creep testing publications, authors often report target conditions (e.g., a test planned for 650 °C and 625 MPa under ISO 204), but actual recorded conditions during testing may vary slightly due to furnace thermal gradients, thermocouple accuracy, or machine load fluctuations (e.g., an actual measured temperature of 650 ± 2 °C and an actual applied stress of 627 MPa). Competency questions CQ5 (test conditions) and CQ9 (cross-study comparability) depend on knowing whether two dataset entries are directly comparable based on planned targets or measured actuals. The example use case for this pattern is representing a creep test where the intended stress condition (cto:specified creep stress datum = 625 MPa) is linked via has_planned_parameter to the test process, while the actual recorded stress (cto:measured creep stress datum = 627 MPa) is linked via has_measured_parameter, accompanied by measurement device metadata and uncertainty bounds where reported.
Related PMDco ODP: None (Not available in the PMDco pattern library; existing PMDco parameter patterns model a single value specification without distinguishing experimental intent from actual physical measurements).
How pattern extended to CTO: As illustrated in the T-Box/A-Box architecture diagram (Fig 4.13), Pattern 7 addresses the intent-vs-actuality gap by modeling each key test parameter (such as temperature and stress) as a dual-entity pair connected to the same creep testing process (cto:creep testing process):

1. Planned Parameter Entity: Modeled as a Planned Value Specification (inheriting under iao:DirectiveInformationEntity / iao:ICE --> bfo:GDC), linked to the process via has_planned_parameter. This entity is further linked to the governing test standard (e.g., ISO 204) or experimental plan.
2. Measured Parameter Entity: Modeled as a Measured Value Specification (inheriting under obi:ScalarValueSpecification / iao:ICE --> bfo:GDC), linked to the process via has_measured_parameter. This entity carries physical measurement metadata, including sensor device references (pmd:load cell, pmd:furnace) and experimental uncertainty ranges where reported.

This structural separation is particularly critical for the ontology-guided LLM extraction pipeline, instructing the LLM to identify and disambiguate planned target values from measured experimental readings in publication text, tables, and figure captions.
Is PMDco ODP completely covering the purpose?: No. PMDco version 3.0 lacks a dedicated ontology design pattern for distinguishing planned experimental targets from actual recorded measurements. Pattern 7 was developed from scratch for CTO as an original methodological contribution. Built in strict alignment with BFO and PMDco conventions, this pattern supports data-quality assessment across the knowledge graph (enabling queries that filter results by the agreement between planned and measured conditions) and represents a key candidate for future inclusion in the PMDco pattern library.

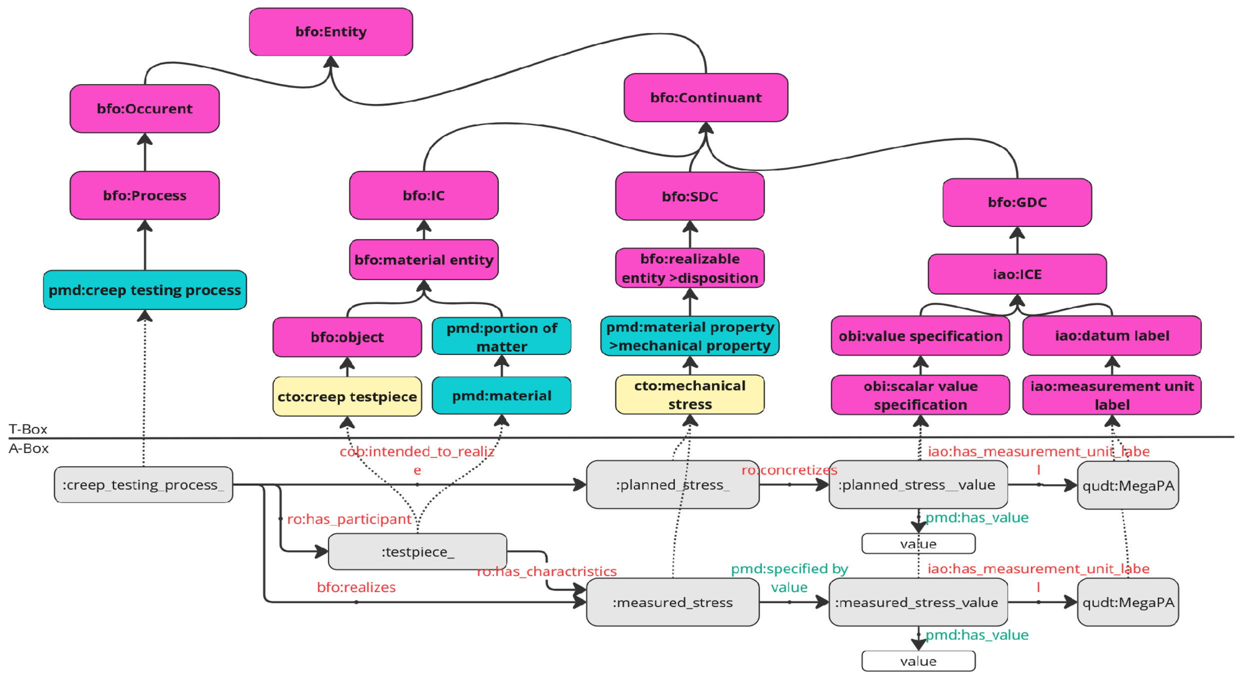

Fig 4.13. Pattern #7 for modeling differences between planned and measured parameters like mechanical stress.

4.6.8. ODPs Coverage and Contributions

As seen in Fig 4.14, for the defined ontology scope and CQs, the seven patterns described above were completely enough for representing the axioms for all classes defined in ontology (section 4.5). Although, other user stories can also be interesting for creating ontology patterns (like time-dependent microstructure evolution during creep, precipitates variations, equipment calibration, etc), but all such cases are beyond scope of this ontology.

Of the seven patterns introduced in this section, five were directly reused or adapted from the PMDco pattern library, demonstrating that PMDco provides substantial coverage of the modeling needs for a mechanical testing domain ontology. However, two patterns (Microstructure and Planned vs. Measured Parameters), required original design, revealing gaps in the current PMDco library. Microstructure characterization is equally relevant to tensile, fatigue, and hardness testing; the planned-vs-measured distinction applies to any experimental domain where instrument tolerances create a gap between protocol and reality. Both new patterns were designed following PMDco naming conventions, BFO alignment principles, and the documentation standards of the PMDco pattern library, so that they can be proposed as community contributions and reused by future ontology development efforts in MSE.

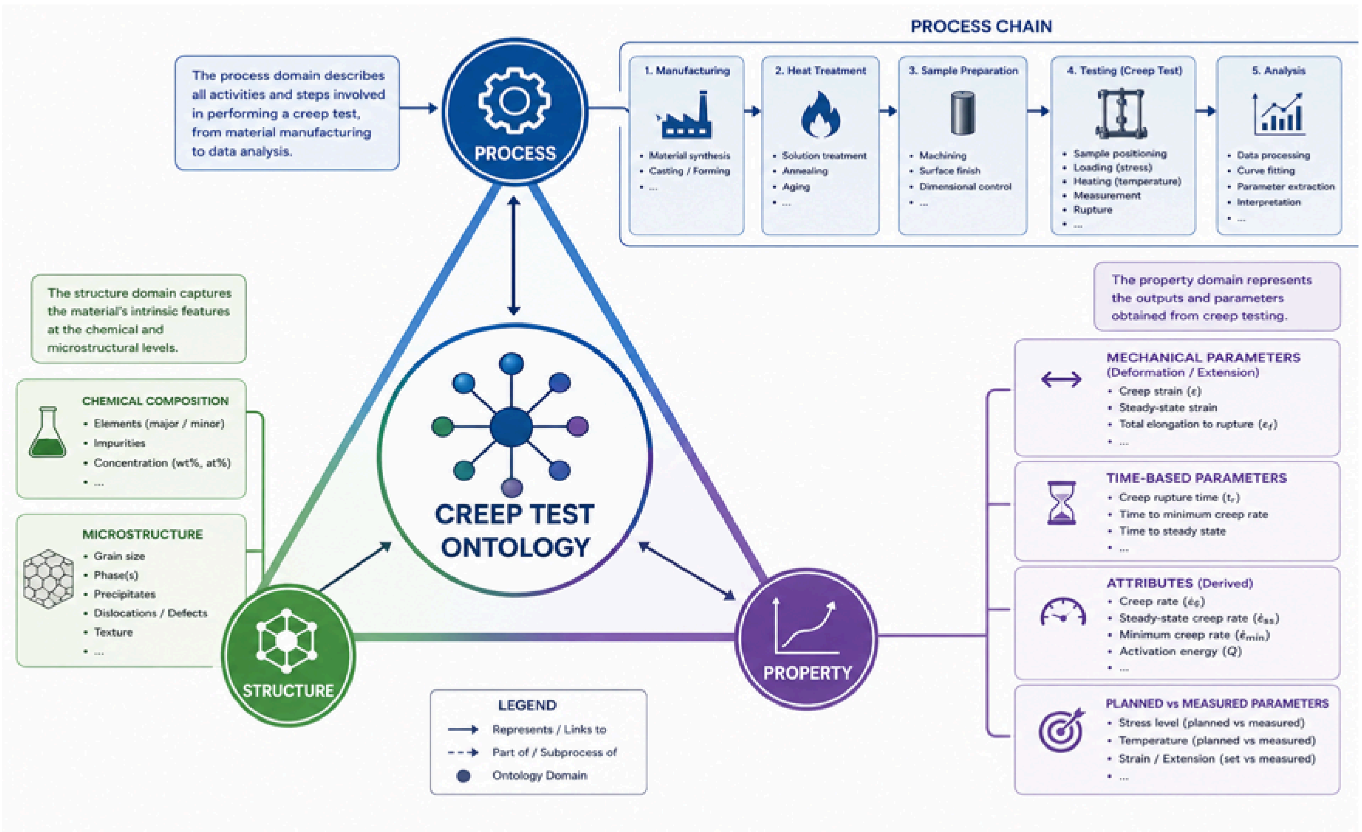

Fig 4.14. ODPs coverage for representing CTO axioms.

4.7. SHACL shapes

As the output of the previous section, CTo with complete taxonomy and axioms was prepared. In this step, the ontology quality assurance was performed using the Shapes Constraint Language (SHACL), which enables formal, machine-checkable validation of an ontology's structural and logical consistency independent of instance data. Since CTO does not yet contain populated individuals, validation was restricted to the T-box (schema) level: SHACL shapes were authored as SPARQL-based constraints (sh:sparql) that inspect class-level axioms directly, rather than relying on native SHACL constraint components that presuppose instance data.

Thirteen shapes were defined, covering two complementary concerns (Table 4.2). The first group (Shapes 1–7) enforces general ontology-engineering hygiene: label and definition completeness and non-redundancy, absence of duplicate labels, presence of a parent axiom for every native class, adherence to the CTO_<digits> IRI naming convention, and absence of subsumption cycles. The second group (Shapes 8–13) verifies domain-specific alignment with the Basic Formal Ontology (BFO) and Relation Ontology (RO) upper-level patterns used throughout CTO, including participation of the test piece in the creep testing process, inherence of creep-related qualities in the test piece, inherence of device-related roles in the creep testing machine, single-bearer constraints on roles, concretization of standards in the testing process, and occurrent-part relations between temporal regions and the process.

Table 4.2. List of SHACL shapes.

#	**Title of SHACL shape**
1	Every CTO-created class has exactly one rdfs:label per language tag (no duplicates, no missing label).
2	Every CTO-created class has exactly one textual definition (skos:definition).
3	A CTO-created class's definition string must not be identical to its label.
4	No two distinct CTO-created classes share the same label.
5	Every CTO-created named class (except owl:Thing) has at least one asserted rdfs:subClassOf or owl:equivalentClass axiom.
6	Native CTO classes follow the CTO_<digits> IRI naming pattern.
7	The CTO-created class hierarchy contains no cycles.
8	Each creep test piece participates in some creep testing process.
9	Each creep-related quality inheres in some creep test piece.
10	Each CTO device-related role inheres in some creep testing machine.
11	Each CTO role is the role of exactly one independent continuant.
12	Each CTO standard/plan is concretized in some creep testing process.
13	Each creep testing process has some temporal region as an occurrent part.

Each shape was scoped to classes native to the https://w3id.org/pmd/cto/ namespace, excluding imported BFO, RO, OBI, IAO, and PMDco terms, so that only CTO's own modeling decisions were assessed. Full shapes can be found in the Github repository[9].

Shapes were executed against the ontology using pySHACL, a Python-based SHACL validation engine, with SHACL Advanced Features enabled to support the SPARQL-based constraints. Validation of all thirteen shapes against cto.ttl returned a positive validation report (Conforms: True; Fig. 4.15), confirming that the ontology's asserted class hierarchy and relational axioms are internally consistent with its intended modeling patterns.

```
(base) hbn@KA000638 cto % cat shacl_shapes/*.ttl > all_shapes.ttl
pyshacl -s all_shapes.ttl -f human -a cto.ttl
Validation Report
Conforms: True
(base) hbn@KA000638 cto %
```

Fig. 4.15. Successful SHACL validation

4.8. Ontology Evaluation

4.8.1. Reasoning

Ontology evaluation by reasoning uses an OWL DL reasoner to check the logical soundness of the T-box: consistency (no class is provably unsatisfiable), coherence (no contradictory axioms), and correct classification of the inferred class hierarchy against the asserted one.

Protégé's built-in HermiT reasoner was used throughout implementation as a live consistency check: after each batch of changes, the reasoner was invoked to verify that no unsatisfiable classes had been introduced and that the inferred class hierarchy matched the intended design. This immediate feedback loop, model, reason, correction, was essential for maintaining logical soundness during the iterative development process.

[9] https://github.com/HosseinBeygiNasrabadi/creep-testing-ontology/tree/main/shacl_shapes

For CTO, the HermiT reasoning completed successfully; the ontology classified without errors, no unsatisfiable classes were detected, and the inferred subsumption hierarchy matched expectations for classes such as creep test piece, testing machine types, and creep properties. A successful HermiT run thus confirms the ontology is logically consistent and ready for further competency-question-based validation.

4.8.2. SPARQL queries

In addition to reasoning-based evaluation, the ontology's competency was validated empirically by translating each CQs formulated for CTO into a corresponding SPARQL query and executing it against the populated ontology. This step, reported in Section 4.2, follows the established practice of using CQs not only to scope but also to validate ontology content: a CQ is considered satisfied only if its SPARQL-OWL formalization returns a correct, non-empty, and semantically appropriate answer when run over the ontology together with sample instance data.

Full list of CQS and related SPARQL query files can be found in the Github repository[10]. Queries covering scoping questions (e.g., retrieving the type of creep testing machine used, the chemical composition of a creep test piece, or the applicable testing standard) as well as validation questions (e.g., checking that inferred subclasses of creep test piece correctly participate in a creep testing process, or that recorded creep rate and stress-rupture values are correctly linked to the corresponding test piece and process) were executed using Protégé's built-in SPARQL query tab.

The results of this evaluation are shown in Fig 4.16, which presents representative queries alongside their retrieved bindings. In every case, the expected entities, properties, or property values were retrieved without errors, missing results, or spurious matches, confirming that the ontology's asserted and inferred axioms are sufficient to answer the full set of competency questions defined for its scope.

Taken together with the successful HermiT reasoning results reported earlier, this SPARQL-based evaluation provides empirical confirmation that CTO satisfies its intended competency questions and is fit for its stated purpose of representing the creep testing process, equipment, materials, and properties.

[10] https://github.com/HosseinBeygiNasrabadi/creep-testing-ontology/tree/main/sparql

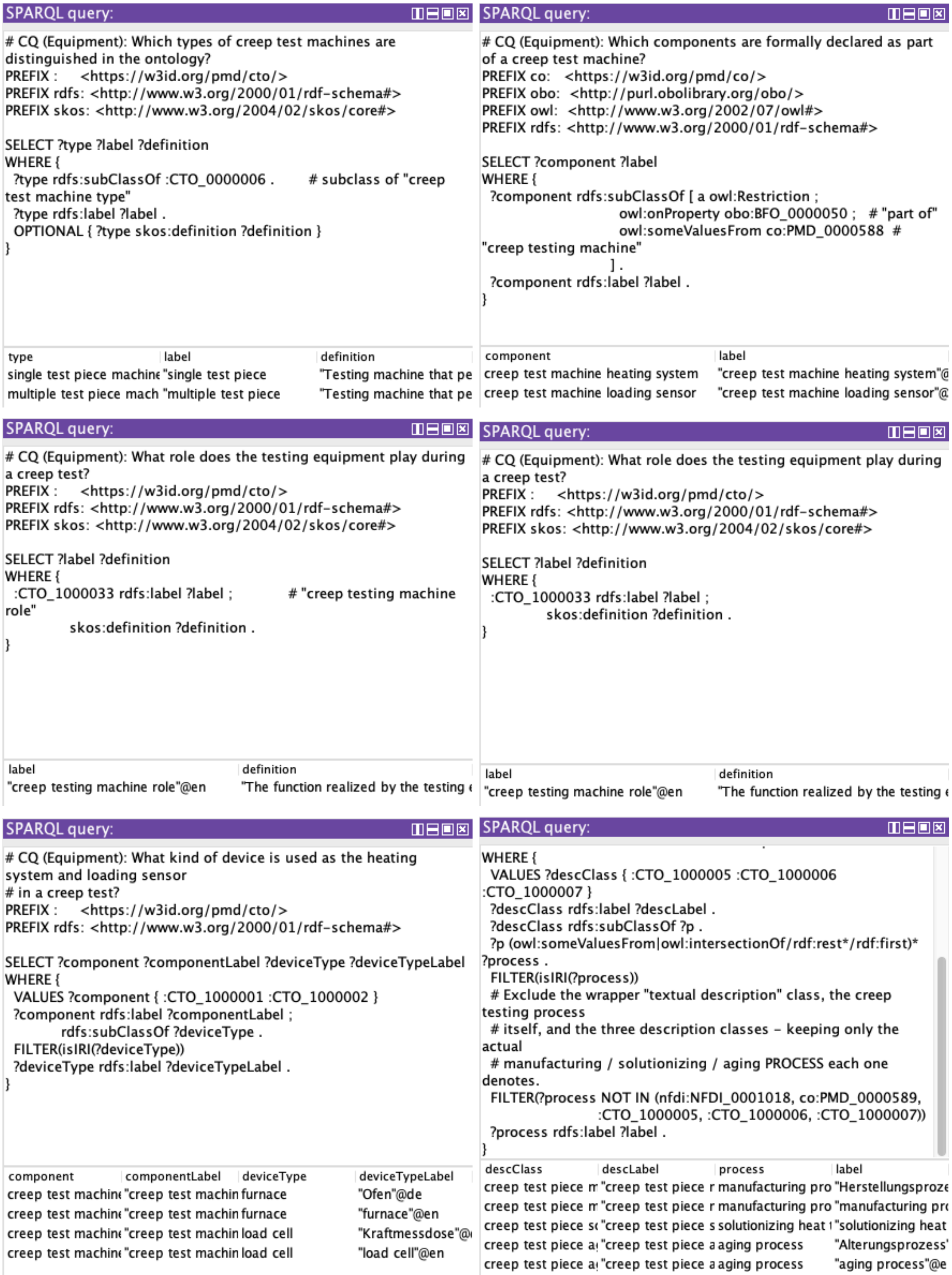

Fig 4.16. Response to some SPARQL queries.

4.9. Ontology Metrics

To evaluate the formal quality, modularity, and structural complexity of CTO, quantitative metrics were computed across the unified ontology graph. Evaluating an ontology through structural and richness metrics provides objective criteria for assessing its design quality, maintainability, and alignment with FAIR data principles. Table 4.3 summarizes the quantitative characteristics of CTO.

Table 4.3. Quantitative Structural and Quality Metrics for CTO

Evaluation Dimension	Metric Name	Measured Value	Description / Interpretation
Reuse & Composition	Native CTO Classes	79 (6.5%)	Application-specific domain concepts
	External Reused Classes	1,129 (93.5%)	Reused from BFO, RO, OBI, IAO, PMDco, NFDIcore, MWO, TTO
	Native Axioms	433 axioms	Axioms asserting properties of native CTO entities
Taxonomy Structure	Max Hierarchy Depth	10	Maximum subsumption path from root
	Average Hierarchy Depth	5.46	Mean depth across all leaf classes
	Tangled Classes	46 classes	Nodes with multiple named superclasses
	Max Node Breadth	77 children	Maximum direct subclasses under a single parent
	Average Node Breadth	4.49 children	Mean subclasses per non-leaf parent
OntoQA Richness	Relationship Richness (RR)	0.069	90 non-taxonomic object properties / 1,217 subclass links
	Attribute Richness (AR)	0.006	7 datatype properties / 1,208 classes (reflects OBI value pattern)
	Inheritance Richness (IR)	1.008	Average subclasses per class (balanced taxonomy)
	Annotation Richness (ANR)	2.48	Average label, definition, and comment annotations per class

Reuse and composition: A foundational design principle of CTO is maximizing the reuse of established upper-level and mid-level domain ontologies rather than inventing duplicate terms. Out of 1,208 total classes in the compiled CTO graph, 79 classes belong natively to the CTO_ namespace, while 1,129 classes are reused from external ontologies (Native CTO classes constitute approximately 6.5% of the total class hierarchy). This high proportion of external reuse confirms that CTO functions as a specialized domain extension embedded directly within the BFO and PMDco ecosystems. Furthermore, 433 axioms explicitly feature a native CTO entity as their subject, representing the application-specific domain assertions of the ontology.

Taxonomy Structure: As detailed in Table 4.3, CTO prioritizes modular interoperability by heavily reusing established upper and mid-level ontologies (BFO, PMDco, OBI, IAO, TTO, MWO), reserving native classes strictly for application-specific creep testing concepts. The taxonomic topology demonstrates a well-developed vertical specialization path from top-level bfo:Entity down to specific parameters, while maintaining a balanced average node breadth across non-leaf classes. Selective multiple inheritance (tangled classes) is restricted to poly-hierarchical classifications where concepts act across multiple functional roles.

OntoQA Richness: The OntoQA richness metrics [76] further highlight the design characteristics of the model. Four quantitative richness metrics highlight the structural profile and semantic expressiveness of the model:

- Relationship Richness (RR = 0.069): RR measures the proportion of non-taxonomic object properties (90 relations) relative to the total number of relations, including 1,217 subclass links. An RR value of 0.069 reflects a strongly taxonomy-driven architecture, where formal subsumption (rdfs:subClassOf) forms the primary structural backbone (~93.1% taxonomic links vs. ~6.9% non-taxonomic relations). In BFO-aligned materials ontologies, a taxonomy-heavy structure is an expected and desirable characteristic, as hierarchical classification provides the formal foundation for description logic reasoners (such as HermiT and ELK). Non-taxonomic object properties (e.g., ro:has_participant, bfo:realizes, pmd:has_process_attribute, ro:has_part) are introduced selectively at key interaction points to link processes, continuants, qualities, and information entities without cluttering the taxonomy.
- Attribute Richness (AR = 0.006): AR is computed as the ratio of datatype properties (7 properties) to total classes (1,208 classes). The low AR value of 0.006 is an intentional architectural feature of BFO/OBI-aligned ontologies rather than a structural deficiency. Instead of attaching primitive data types (such as literal floats or strings) directly to classes using datatype properties—a practice that strips away measurement context, units, and uncertainty—CTO implements the OBI scalar value specification pattern (obi:ScalarValueSpecification / pmd:ValueObject). Numerical values are modeled as structured information entities linked to QUDT unit URIs (qudt:PERCENT, qudt:DEG_C, qudt:MegaPA) and measurement provenance metadata, prioritizing semantic expressiveness and unit traceability over primitive data typing.
- Inheritance Richness (IR = 1.008): IR measures the average number of subclasses per class across the entire graph. An IR value close to 1.00 (1.008) confirms a highly structured, vertical taxonomy tree where knowledge is detailed across multiple hierarchical levels, avoiding 'horizontal' ontologies that lump large numbers of classes under a few shallow parent nodes.
- Annotation Richness (ANR = 2.48) & Documentation Coverage: ANR measures the average number of annotation axioms (rdfs:label, skos:definition, rdfs:comment) per class. With an average of 2.48 annotation axioms per class and 100% label coverage (1,208 / 1,208 classes possessing an explicit rdfs:label), CTO guarantees complete human readability, high-quality prompt generation for LLM extraction pipelines, and seamless inspection in knowledge graph visualizers and SPARQL query endpoints.

4.10. Ontology Documentation and Release

Repository: The ontology repository is constructed in GitHub[11]. The repository follows the standard ODK layout:

- src/ontology/ contains the
  - primary OWL source file (cto-edit.owl)
  - catalogue file (catalog-v001.xml) that maps ontology IRIs to local files
  - imports/ contains the extracted modules of external ontologies (bfo_import.owl, ro_import.owl, iao_import.owl, pmdco_import.owl, etc.);
- docs/ contains the Widoco-generated HTML documentation;
- ODK configuration (cto-odk.yaml),
- README with usage instructions and citation guidelines,
- LICENSE file,

[11] https://github.com/HosseinBeygiNasrabadi/creep-testing-ontology

- CITATION.cff file for machine-readable citation metadata.

Continuous integration via GitHub Actions runs automated quality checks on every push, including HermiT consistency verification, ROBOT report generation, and documentation regeneration, ensuring that the repository's main branch always contains a logically sound, well-documented ontology.

Release Formats: Each release ships in five formats to cover the most common ontology workflows: RDF/XML (.owl) for Protégé editing, Turtle (.ttl) for SPARQL endpoints and readability, OWL Functional Syntax (.ofn), JSON-LD (.jsonld) for web APIs, and OBO Format (.obo) for OBO Foundry compatibility. Release notes accompany each version, summarizing what changed and what was added.

License: CTO is publicly available as an open-source ontology under the Creative Commons Attribution 4.0 license (CC BY 4.0). This section provides the essential information needed to find, access, cite, and reuse the ontology.

Documentation: Full documentation is generated with Widoco and published at: https://materialdigital.github.io/creep-testing-ontology/. The documentation includes an ontology overview, the complete class hierarchy with definitions, property descriptions, namespace declarations, diagrams, and cross-references to imported ontologies (BFO, PMDco, RO, IAO, OBI, COB). It is regenerated with every release so it always matches the current version.

FAIR Status: CTO is designed to meet the FAIR principles: it is Findable through a persistent w3id IRI; Accessible via the public GitHub repository with no authentication required; Interoperable through BFO alignment and reuse of PMDco and community-standard vocabularies; and Reusable under CC BY 4.0 with comprehensive documentation and a standard ODK repository structure.

Chapter 5: CTO application for LLM literature mining and creep KG construction

This chapter describes the CTO application for ontology-guided LLM literature mining and KG construction. Fig. 5.1 shows the pipeline we follow in this section. It covers the full pipeline; from the preparation of the creep literature database, through the creation of a gold standard benchmark dataset, to the LLM-based extraction, the construction of the KG from extracted triples, and the evaluation of the KG through SPARQL queries, SHACL validation, and logical reasoning. As mentioned already, we are not developing LLM and KG construction methods in this thesis, but reusing the available approaches to check if CTO is fitting for these cases.

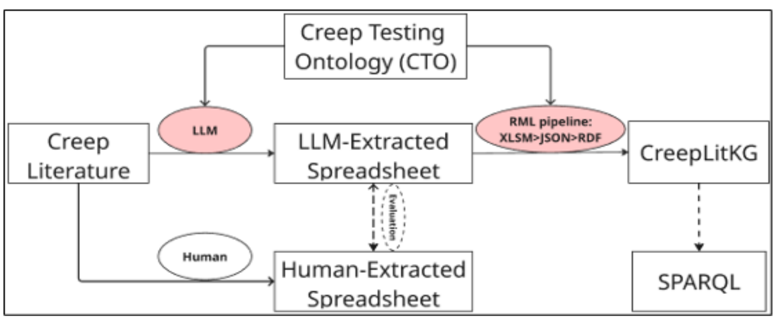

Fig. 5.1. Pipeline for ontology-guided literature extraction, benchmarking evaluation, and KG construction framework.

5.1. Creep Literature Database and Data Extraction Instruction

5.1.1. Preparing a dataset of 100 papers on “Inconel Creep”

To build a realistic literature corpus for testing ontology-guided extraction, a systematic literature search was conducted across major scientific databases, including Scopus, Web of Science, and Google Scholar, alongside open institutional data repositories such as the National Institute for Materials Science (NIMS) Creep Data Sheets and NASA technical reports. The search strategy combined domain terms like ("Inconel” or “nickel superalloy” or “Ni-based superalloy” or “IN718”) and (“creep" or "rupture time" or "stress rupture" or "creep rate"). After removing duplicates and screening abstracts, the search yielded a final corpus of approximately 100 research publications.

Paper selection was guided by clear inclusion and exclusion criteria to maintain a cohesive domain focus and ensure that the extracted data remains semantically comparable:

- Material and Alloy Coverage: Selected publications focus on nickel-based superalloys from the Inconel family, specifically covering polycrystalline wrought and cast grades such as IN718, IN625, IN740H, IN617, and IN713C. These alloys represent the primary materials used in high-temperature creep applications.
- Parameter Coverage: Publications were required to report on the core experimental workflow, ensuring that the corpus collectively covers all 32 target parameters. This includes material composition, heat treatment steps, microstructural features, test conditions, and resulting creep properties.
- Pure Creep Testing Focus: Eligible papers reported on uniaxial tensile creep tests under constant load or constant stress. Studies focusing on low-cycle or high-cycle fatigue, creep-fatigue interactions, fatigue crack growth, or creep-feed grinding were excluded. These processes involve different failure mechanisms and reporting conventions that would introduce ambiguity into both the extraction pipeline and the ontology evaluation.

5.1.2. Preparing a gold dataset of 20 papers

To evaluate automated LLM extraction performance against human accuracy, a subset of 20 papers was selected from the 100-paper corpus to serve as a gold-standard ground truth benchmark. The 20 gold-standard papers were chosen through stratified sampling to ensure they accurately represent the variability of the broader 100-paper corpus:

- Diverse Sources and Publishers: The benchmark includes articles from major materials science journals (such as Acta Materialia, Materials Science and Engineering: A, and Metallurgical and Materials Transactions A published by Elsevier, Springer, and Wiley), conference proceedings, and standardized data repositories like NIMS.
- Comprehensive Parameter Density: The 20 selected papers were chosen because they densely cover the full spectrum of our 32 parameters, ensuring that complex fields like multi-step aging treatments, precipitate phase fractions, and secondary creep rates are well-represented in the evaluation.
- Varied Formatting Styles: Papers were selected across different publication decades (from legacy reports to recent 2020s literature) to test how extraction handles varying layouts, table structures, and textual descriptions.

5.1.3. Information extraction instruction

Identical instructions for information extraction, parameter definitions, and output schemas were provided to both the human annotator and the LLM extraction pipeline:

I- Target extracting parameters: The extraction schema targets 32 parameters organized into five primary domain categories: Material, Processing, Microstructure, Testing Parameters, and Creep Properties (Fig. 5.2). These specific categories were selected for three core reasons:

- Completeness of the Processing–Structure–Property–Equipment Chain: These five domains represent the minimum necessary parameter set required to capture the full physical causality of a creep experiment—from initial material identity and heat treatment history through microstructural state to controlled test conditions and final mechanical creep response.
- Direct Mapping to Competency Questions: Every entity in these five categories directly maps to one or more competency questions (CQs 1–16) established during ontology design (Section 4.2). Omitting any of these categories would create gaps in the knowledge graph, preventing automated reasoning across heat treatments, microstructural precipitate features, or test conditions.

- Cross-Study Comparability: By extracting parameters across these five standardized categories, test results from different publications can be directly filtered, grouped, and compared under equivalent thermal, mechanical, and microstructural conditions.

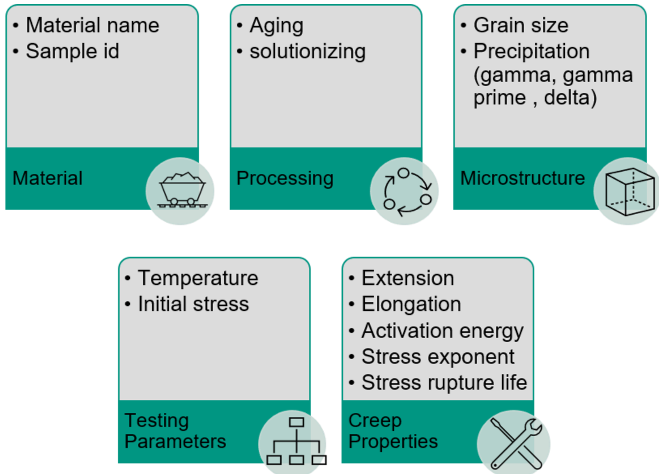

Fig. 5.2. Target creep domain categories and parameters extracted from scientific literature to populate CreepLitKG.

II- CTO-based information extraction: All 32 target parameters have related classes in CTO. We listed these classes and their definitions for all 32 parameters in Fig. 5.3. This excel file is the input for both human and LLM to guide what information exactly should be extracted.

#		Related ontology class	Related ontology class (definition)
1	Paper_DOI	cto:creep publication DOI	A creep publication DOI is a Digital Object Identifier (DOI) that denotes a publication which is about a
2	material_name	cto:creep material identifier	A creep material identifier is a material identifier that denotes a creep test piece which participates in
3	material_chemical_composition	cto:creep test piece composition	Creep test piece composition is a chemical composition data item that specifies the chemical
4	sample_id	cto:creep test piece identifier	A creep test piece identifier is an object identifier that denotes a creep test piece which participates in
5	processing-method	cto:creep test piece manufacturing description	A creep test piece manufacturing description is a textual description that denotes a manufacturing
6	solutionizing-process	cto:creep test piece solutionizing description	A creep test piece solutionizing description is a textual description that denotes a solutionizing heat
7	aging-process	cto:creep test piece aging description	A creep test piece aging description is a textual description that denotes an aging process performed
8	Grain size value specification_X	cto:creep test piece grain size value	A creep test piece grain size value is a scalar value specification that specifies the value of the grain
9	delta-precipitates-fraction_X	cto:creep test piece delta precipitate fraction value	A creep test piece delta precipitate fraction value is a scalar value specification that specifies the
10	delta-precipitates-size_X	cto:creep test piece delta precipitate size value	A creep test piece delta precipitate size value is a scalar value specification that specifies the value of
11	gama-precipitates-fraction_X	cto:creep test piece gamma precipitate fraction value	A creep test piece gamma precipitate fraction value is a scalar value specification that specifies the
12	gama-precipitates-size_X	cto:creep test piece gamma precipitate size value	A creep test piece gamma precipitate size value is a scalar value specification that specifies the value
13	gama-prime-precipitates-fraction_X	cto:creep test piece gamma prime precipitate fraction valu	A creep test piece gamma prime precipitate fraction value is a scalar value specification that specifies
14	gama-prime-precipitates-size_X	cto:creep test piece gamma prime precipitate size value	A creep test piece gamma prime precipitate size value is a scalar value specification that specifies the
15	test_standard_X	cto:creep test standard	A creep test standard is a standard that is concretized as a plan which is intended to realize a creep
16	specified_temperature_X	cto:specified creep temperature datum	A specified creep temperature datum is an input specification that is concretized as a temperature
17	initial_stress_X	cto:specified creep stress datum	A specified creep stress datum is an input specification that is concretized as a mechanical stress
18	creep_rupture_time_X	cto:creep rupture time value	A creep rupture time value is a scalar value specification that specifies the value of the stress rupture
19	percentage_permanent_elongation_X	cto:creep percentage permanent elongation value	A creep percentage permanent elongation value is a scalar value specification that specifies the value
20	Percentage_elongation_after_creep_fracture	cto:Percentage elongation afte creep fracture value	A percentage elongation after creep fracture value is a scalar value specification that specifies the
21	percentage_reduction_of_area_after_creep_	cto:percentage reduction of area after creep fracture valu	A percentage reduction of area after creep fracture value is a scalar value specification that specifies
22	percentage_total_extension_X	cto:percentage total extension value	A percentage total extension value is a scalar value specification that specifies the value of the
23	percentage_initial_total_extension_X	cto:percentage initial total extension value	A percentage initial total extension value is a scalar value specification that specifies the value of the
24	percentage_elastic_extension_X	cto:percentage elastic extension value	A percentage elastic extension value is a scalar value specification that specifies the value of the
25	percentage_initial_plastic_extension_X	cto:percentage nitial plastic extension value	A percentage initial plastic extension value is a scalar value specification that specifies the value of
26	percentage_plastic_extension_X	cto:percentage plastic extension value	A percentage plastic extension value is a scalar value specification that specifies the value of the
27	percentage_creep_extension_X	cto:percentage creep extension value	A percentage creep extension value is a scalar value specification that specifies the value of the
28	stady_state_creep_rate_X	cto:stady state creep rate value	Steady state creep rate value is a scalar value specification that specifies the value of the creep rate
29	stress_exponent_X	cto:stress exponent value	A stress exponent value is a scalar value specification that specifies the value of the stress exponent
30	activation_energy_X	cto:creep activation energy value	A creep activation energy value is a scalar value specification that specifies the value of the activation

Fig 5.3. Target extraction parameters and their related CTO classes labels and definitions for ontology-guided literature extraction.

III- Literature extraction template: For the identical and comparable LLM- and human- extracted information, tabular spreadsheet template of Fig. 5.4 has prepared. Here, the first row collects all 32 parameters, and the extracted information from papers are collected in the next rows. It should be noted that for each paper, more than one raw information can be extracted. For example, if a paper reports preparation of several samples and mentions processing-structure-properties information for each sample. For this reason, the first column of template assigned with an ID which is not collected directly from paper, but is an identifier for each dataset in the form of “Paper DOI+Sample ID”.

ID	Paper DOI	Material name	Material chemical composition	Sample ID	Processing method	Solutionizing	Aging	Grain size	Delta precepitates fraction	Delta precepitates size	Gama precepitates fraction	Gama precepitates size	Gama prime precepitates fraction	Gama prime precepitates size
10.1016/j.msea.2008.04.097_HT1	10.1016/j.ms	Inconel 718 s	wt.%: C 0.04,	HT1	Standard hea	1095°C, 1 h/	955°C, 1 h/A	170 μm						
10.1016/j.msea.2008.04.097_HT2	10.1016/j.ms	Inconel 718 s	wt.%: C 0.04,	HT2	Solutionizing	1095°C, 1 h/	720°C, 8 h/F	160 μm						
10.1016/j.msea.2008.04.097_HT3	10.1016/j.ms	Inconel 718 s	wt.%: C 0.04,	HT3	Solutionizing	1095°C, 1 h/	955°C, 3.5 h/	155 μm						

ID	Test standard	Temperature	Initial stress	Creep rupture time	Percentage permanent elongation	Percentage elongation after creep fracture	Percentage reduction of area after creep fracture	Percentage total extension	Percentage initial total extension	Percentage elastic extension	Percentage initial plastic extension	Percentage plastic extension	Percentage creep extension	Steady state creep rate	Stress Exponent (n)	Activation Energy (Qc)
10.1016/j.msea.2008.04.097_HT1	ASTM E 8M-9	650°C	625 MPa	159.74 ±19.18 h		1.1542 ±0.0577%								0.96 ×10−4 ±5.16×10−6		
10.1016/j.msea.2008.04.097_HT2	ASTM E 8M-9	650°C	625 MPa	331.18 ±28.93 h		5.3565 ±0.2678%								1.42 ×10−4 ±2.05 ×10−6		
10.1016/j.msea.2008.04.097_HT3	ASTM E 8M-9	650°C	625 MPa	196.75 ±16.48 h		1.4892 ±0.0745%								1.04 ×10−4 ±3.49 ×10−6		

Fig 5.4. Template for extraction creep literature.

IV- Cell filling instruction: Another rule applied uniformly to both human annotators and the LLM concerns the format of extracted values; for ease of extraction, each field may be filled as a single string rather than as a strictly typed, unit-separated quantity. For example, a stress value can be entered as "160 MPa" rather than being split into a numeric field (160) and a separate unit field (MPa). This relaxation was adopted deliberately to lower the extraction burden and reduce the parsing errors that a rigid, unit-separated schema would introduce given the varying reporting conventions across papers. It does not, however, license careless extraction; both the LLM and human annotators are instructed to extract values as cleanly as possible (e.g., avoiding transcription errors, and limiting each string to the single value actually reported rather than surrounding prose or ambiguous ranges).
V- Empty cell policy: A strict rule applied to both human annotation and LLM extraction is that missing values must be left completely empty. If a paper does not report a specific parameter, such as grain size or reduction of area, the corresponding spreadsheet cell remains blank. Placeholder entries like "N/A," "None," "Not Reported," or numerical zeros ("0") are not permitted.

5.2. Human Extraction - Preparing Gold Standard Dataset

To enable quantitative evaluation of the LLM extraction pipeline, a manually annotated gold standard dataset was prepared from a subset of the literature corpus. A sample of 20 publications was selected to represent the diversity of the full corpus: different alloy grades, different journals and publishers, different reporting conventions (tables-only, text-only, mixed), and different levels of microstructural detail. Each selected publication was read in full by two domain experts, and information extraction performed according to instruction of section 5.1.3.
The manual annotation of the gold standard dataset revealed a wide range of recurring challenges that span the entire extraction workflow, from publication retrieval and text parsing through entity recognition, value normalisation, and entity linking. Here we listed some of these challenges:

- **Identifier, unit, and terminology heterogeneity:** Creep data are reported with substantial heterogeneity in naming conventions, units, and symbols. The same creep property appears under multiple names: rupture time is variously called 'Tr,' 'time to rupture,' 'rupture life,' or 'creep ductility.' Temperature is reported in both degrees Celsius (°C) and Kelvin (K), requiring unit recognition and normalization to a common standard. Chemical composition is presented sometimes in structured tables with weight percentages (wt.%), sometimes as unstructured running text, requiring the extractor to handle both formats. Special characters such as '±' (indicating tolerance or standard deviation) are frequently mangled during PDF text extraction, causing values like '650 ± 5 °C' to be parsed as separate numbers or garbled symbols. The term 'Inconel' — a trademark — is common in practice, but many papers use the broader 'nickel-based superalloys' or only alloy designations (IN718, IN625), requiring the extraction pipeline to recognize and normalize multiple naming conventions to the correct CTO material class.

- **Processing and heat treatment complexity:** The description of processing routes and heat treatments is one of the most challenging extraction tasks because it combines temporal sequences, multiple parameters, and long unstructured text. A typical heat treatment specification reads: 'aging at 955 °C, 1 h/air cool + 720 °C, 8 h/furnace cool 57 K/h to 620 °C, 8 h/water quench.' This single sentence encodes four sequential steps, each with a temperature, a duration, and a cooling method — information that must be decomposed into separate processing nodes in the CTO process chain. The LLM must therefore perform both entity extraction (identifying temperatures, durations, cooling methods) and temporal structuring (ordering the steps correctly). Furthermore, processing descriptions are frequently spread across multiple paragraphs or sections, requiring aggregation of non-contiguous text fragments. This task is closer to event extraction than to simple named entity recognition and represents one of the most significant accuracy challenges for the extraction pipeline.
- **Multi-sample and multi-condition disambiguation:** Most creep publications report results for several distinct material conditions tested under multiple temperature–stress combinations. Disambiguating which result belongs to which specimen under which condition is a fundamental challenge. In one representative case, two sample identifiers are reported but each sample was tested under multiple temperature and stress conditions — the LLM incorrectly assigned the test conditions as sample identifiers, conflating specimen identity with test parameters. In another case, the samples are correctly distinguished, but the study examines the effect of different applied stresses on creep behavior. The LLM aggregated the results into ranges (e.g., 'rupture time: 120–450 hours') rather than extracting individual data points for each stress level, destroying the stress-dependence information that is the central scientific contribution of the study. These failures motivate three prompt design rules: (i) sample identifiers are unique specimen labels, never test parameters; (ii) each unique combination of specimen and test condition generates a separate dataset entry; and (iii) quantitative values must be extracted as point values, never as ranges.
- **Table- and figure-dominated data:** The majority of creep property values (rupture times, minimum creep rates, and elongations) are reported in tables and figures rather than in running text. Tables present structured data that is theoretically machine-readable, but real-world table formats vary widely: merged cells, multi-level headers, footnotes, and split tables spanning multiple pages all complicate programmatic parsing. Figures, particularly creep strain vs. time curves and Larson–Miller parameter plots, contain the richest temporal and comparative information but are entirely inaccessible to text-only LLMs. It is therefore uncertain whether current text-based extraction pipelines can reliably capture data from these predominant formats. This thesis addresses table extraction through structured prompt templates that instruct the LLM to parse tabular data row-by-row, but the extraction of quantitative values from figures remains beyond the scope of the current pipeline and requires vision-language models for future work.
- **Corpus-level challenges:** Beyond individual publication challenges, the corpus itself presents several structural difficulties. Some publications — particularly older papers and conference proceedings — lack a DOI, complicating automated provenance tracking and requiring fallback to alternative identifiers. A significant fraction of the broader high-temperature materials literature addresses creep–fatigue interaction rather than pure creep, and while these papers were filtered during corpus preparation, the boundary between the two topics is not always clear from titles and

abstracts alone, requiring manual inspection. Finally, the publication set spans several decades, and reporting conventions have evolved considerably — earlier papers use different units, different alloy naming schemes, and less structured data presentation than recent publications, introducing a temporal dimension to the heterogeneity challenge that the extraction pipeline must accommodate.

Taken together, these challenges define the problem space that the ontology-guided extraction pipeline must address. The prompt design principles derived from this analysis (unique sample identifiers, point-value-only extraction, qualitative-vs-quantitative flagging, and ontology-gap logging) are implemented in the extraction templates described in Section 5.3 and evaluated through the metrics reported therein.

5.3. LLM Extraction and Evaluation Against Gold Dataset

Creep information extraction from the gold standard dataset (sample of 20 publications) was also performed by LLM according to instruction of section 5.1.3. The LLM literature mining was done by tool LLM4CreepLitKG[12], its user interface can be seen in Fig. 5.5. LLM4CreepLitKG is working according to the defined extraction instructions. Here, PDFs are uploaded through a drag-and-drop interface and the extraction engine processes each one based on CTO inputs, then mapping the extracted LLM's output to the template. The interface supports multiple AI providers; for this evaluation, Claude (Anthropic) and GPT-4 (OpenAI) were tested, with model selection, API key configuration, and processing options all managed through the same panel.

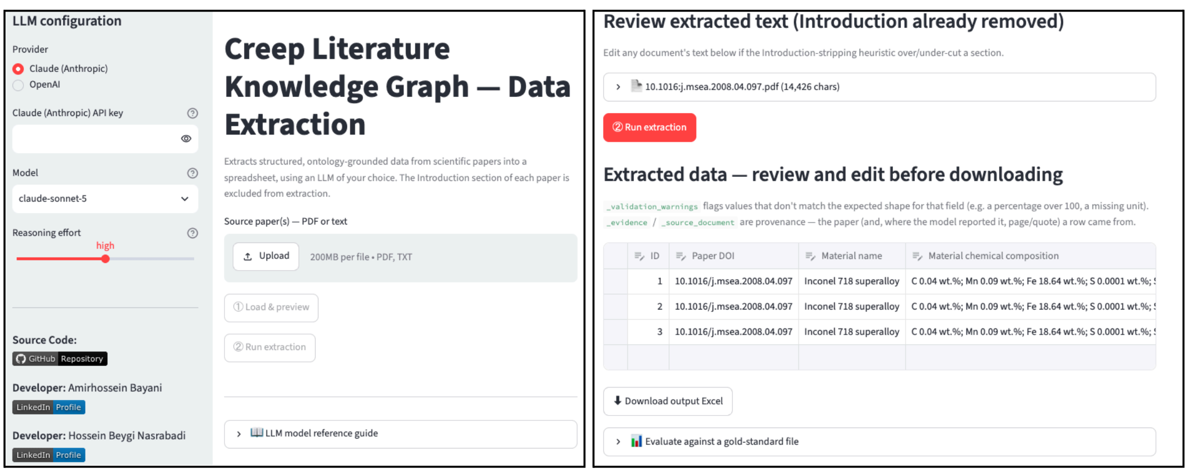

Fig. 5.5. LLM4CreepLitKG interface and example of data extraction from a paper.

The LLM-extracted information was collected in a tabular spreadsheet exactly according to the template that was also used by humans. These similar output structures make the evaluation more straightforward. In this regard, the LLM4CreepLitKG tool has a section for evaluating the LLM outputs against the human-extracted gold standard dataset (See Fig. 5.6). The evaluation follows the field-level matching rules, meaning for each aligned row, every shared column (excluding ID) is compared value-by-value (_values_match / _normalize), checked in this order:

1. Both values blank → match. ("Blank" covers both a missing predicted field and an empty gold Excel cell — the latter comes back from pandas as float NaN, not None, which _normalize treats as blank too, not as the literal text "nan".)

[12] https://github.com/HosseinBeygiNasrabadi/Creep_Literature_Knowledge_Graph/tree/main/LLM4CreepLitKG

2. One blank, one not → no match.
3. Normalized strings identical (lowercased, punctuation/units/symbols stripped down to letters+digits+.) → match. This is why "1095 C, 1 h/AC" and "1095 °C, 1 h/AC" count as the same value — normalization treats °, commas, and slashes as formatting noise, not content.
4. Both sides contain numbers and the extracted numeric tokens are identical (e.g. "159.74 ± 19.18 h" vs "159.74 19.18") → match, regardless of surrounding text/units.
5. Otherwise, fuzzy string similarity (difflib.SequenceMatcher) — match if the ratio is ≥ 0.9, else no match. This catches minor typos/reordering but not paraphrasing (see limitation below).

As the output of this evaluation, there's no single pass/fail verdict, the feature is diagnostic, not a gate. You get:

- Overall field accuracy — correct fields ÷ total fields across all matched rows.
- Per-column accuracy bar chart — which fields the model gets wrong most often.
- Rows matched (X/Y) and a count of extra predicted rows.
- A row-by-row diff table with (gold) / (predicted) / (match) per field, plus "extra"/"missing" row notes — for manual review, not an automated accept/reject.

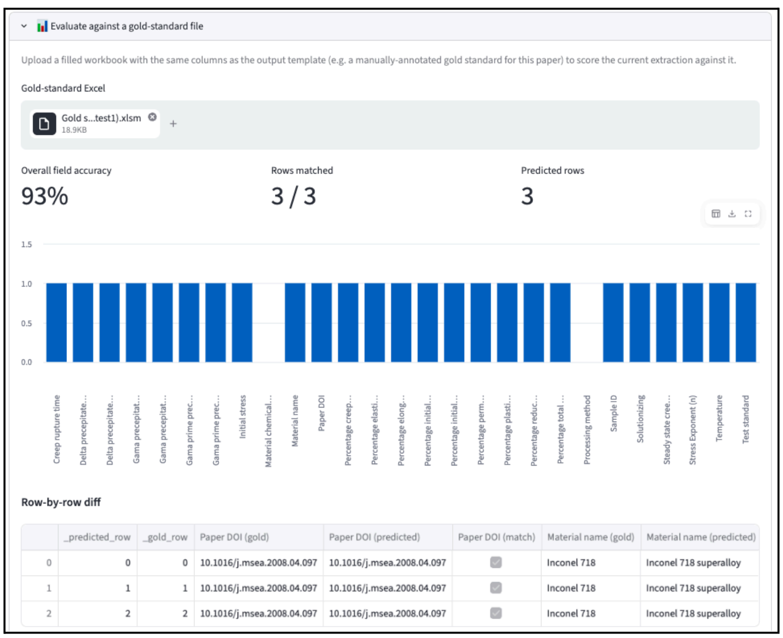

Fig. 5.6. Example evaluation outputs of LLM4CreepLitKG tool.

It should be noted that the evaluation approach is still under progress and needs addressing several issues. Here one important issue is that the evaluation based on matching is lexical, not semantic. Every rule above is string/number-based, there's no embedding or LLM-as-judge check anywhere in the pipeline. This is usually fine for numeric+unit fields (temperature, stress, percentages, times), since the numeric-token rule already ignores formatting. It's weaker for free-text descriptive fields (e.g. processing method), where a correct paraphrase can score below the 0.9 fuzzy threshold and get marked a mismatch even though it describes the same thing.

Based on the current evaluation feature of the LLM4CreepLitKG tool, the LLM outputs achieve an overall field accuracy of more than 85% against the gold standard annotations across the 20-paper dataset. This figure should be read as a snapshot rather than a final result: it reflects the lexical matching rules described above, run on the current extraction prompt and model configuration, and is expected to shift as the prompt, the model, and the gold-standard annotations are further refined over the remaining evaluation iterations.

5.4. Construction of Creep Literature Knowledge Graph (CrepLitKG)

The extraction output (tabular spreadsheet in xlsm format) is transformed into a CTO-compliant RDF using the CreepLitKG approach[13]. The xlsx to RDF conversion follows a four-step pipeline:

- First, the spreadsheet is converted into a structured intermediate JSON format; this step is purely syntactic and reorganizes each row into a machine-readable record, keeping numeric values together with their units, preserving qualifiers and uncertain or purely descriptive values as plain text rather than forcing them into a rigid numeric form, and omitting empty cells outright so that missing data produces no misleading entries in the graph.
- Second, a separate declarative mapping encodes all of the semantic decisions (which CTO class each entity belongs to, which CTO property links each extracted value) and is compiled into an executable set of RML (RDF Mapping Language) mapping rules.
- Third, these rules are applied to the intermediate data to generate the actual RDF triples; each entity is assigned a deterministic identifier derived from its row in the spreadsheet, so that re-running the pipeline after new data is added regenerates identical triples for existing rows rather than duplicating or renumbering them.
- Fourth, the resulting graph is validated against a set of predefined SHACL shapes before it is accepted: any violation halts the process, ensuring that an invalid or malformed graph is never mistaken for a valid one and never loaded into the triple store.

Once the four-step pipeline has run, its output is a single RDF file, creep_literature_rdf.ttl, containing the complete CTO-compliant graph of triples. This file is not consumed directly; instead, it is loaded into a triple store to make its contents queryable. For CreepLitKG, this triple store is provided by the MaterialDigital Dataportal[14] (Fig. 5.7), the PMD infrastructure component used to host and serve semantic research data across the consortium. Hosting the graph there, rather than in a private or ad hoc store, automatically exposes it through a public SPARQL endpoint, so that the data can be queried by anyone (a human through an interactive query interface, or a machine through a standard HTTP request) without needing access to the original pipeline or its intermediate files. At this point, the pipeline's output has become an actual, queryable KG rather than a static RDF file. Fig. 5.7 also shows the live CreepLitKG deployment[15].

[13] https://github.com/HosseinBeygiNasrabadi/Creep_Literature_Knowledge_Graph/tree/main
[14] https://dataportal.material-digital.de/dataset/creep_literature_knowledge_graph
[15] https://hosseinbeyginasrabadi.github.io/Creep_Literature_Knowledge_Graph/

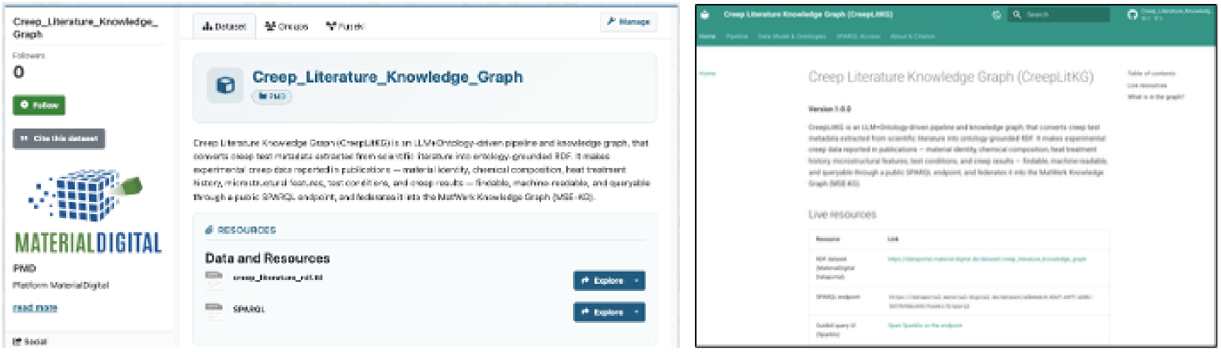

Fig. 5.7. RDF in the MaterialDigital Dataportal and CreepLitKG live deployment.

5.5. Data Retrieval and Creep KG Evaluation

The HermiT reasoner is applied to the populated KG to verify that no logical contradictions have been introduced by the extraction process. Specifically, the reasoner checks for unsatisfiable classes (classes that, given the accumulated axioms and instance data, cannot have any members) which would indicate that the extracted data violates one or more CTO constraints. The KG passes this test if HermiT reports zero unsatisfiable classes and the inferred class hierarchy remains consistent with the asserted hierarchy.

To validate that the constructed KG functions as intended (and not merely that it is structurally well-formed) some example SPARQL query designed and run against the populated CreepLitKG graph. This provides a direct, testable check of whether the ontology-guided KG can actually answer the questions it was designed to answer, rather than only representing the extracted data correctly.

Example SPARQL queries used for this validation:

- CQ1 — From which literature sources (publications) does the creep data originate?
- CQ2 — Which materials have been creep tested?
- CQ3 — Which testing standards have been used for creep testing Inconel materials?
- CQ4 — At what temperature and applied stress was each creep test performed?
- CQ5 — What heat treatment steps (solutionizing, aging) were applied to a test piece before creep testing?
- CQ6 — What is the grain size of the material for each heat treatment condition?
- CQ7 — What percentage elongation after creep fracture was observed for each test?
- CQ8 — What are the stress rupture time and steady-state creep rate measured for Inconel 718?

Putting the RDF file in MaterialDigital dataportal provides a SPARQL endpoint (Fig. 5.8), where queries can be executed through the guided query UIs (kike Sparklis and YASGUI)[16]. Fig. 5.8 also shows the result returned for example query #4. The retrieved answer was checked directly against the corresponding source publication and found to be correct, confirming that the KG returns the expected values for this competency question rather than an artifact of the query construction. The same verification procedure (manual comparison of the SPARQL result against the source paper) was carried out for each of the remaining example queries, with correct retrieval confirmed in each case.

Together, these results confirm that the KG is not only structurally valid but functionally capable of answering the competency questions it was scoped to address, providing direct evidence that the CTO-guided construction pipeline produces a KG that fulfills its intended purpose.

[16] https://hosseinbeyginasrabadi.github.io/Creep_Literature_Knowledge_Graph/sparql/

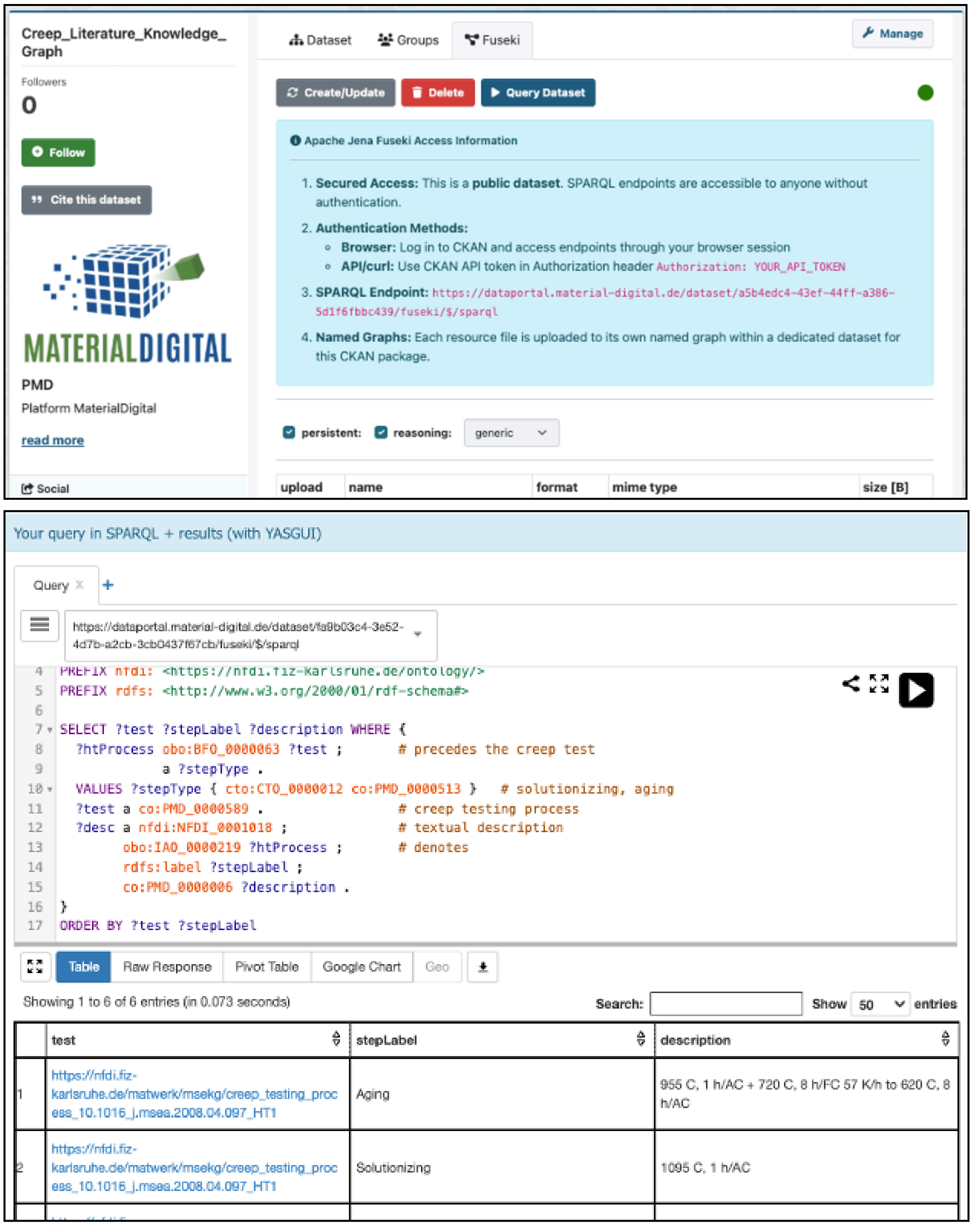

Fig. 5.8. The SPARQL endpoint used to query the CreepLitKG graph, and the result returned for the SPARQL query of CQ4.

Conclusion

This thesis set out to close a documented gap in materials ontology engineering: the absence of a BFO-compliant, standard-grounded ontology for creep testing of metallic materials, and the absence of evidence that such an ontology can practically guide LLM-based literature mining and knowledge graph construction. The following points summarize how the work addressed this gap and answered the three research questions posed in Chapter 3.

1. A BFO-compliant, ISO 204-grounded ontology for creep testing was developed and validated (RQ1). The Creep Testing Ontology (CTO) represents the full processing–structure–property narrative of creep research (material identity and composition, processing history, microstructure, test conditions, creep properties, and deformation mechanisms) within a single, logically consistent schema built on BFO and PMDco. HermiT reasoning confirmed logical consistency and detected no unsatisfiable classes, SPARQL queries derived from the competency questions returned correct results against populated instance data, and SHACL shapes validated successfully, together demonstrating that the BFO/PMDco framework provides sufficient semantic coverage for the creep domain.
2. An ODP-based development workflow produced a template transferable beyond CTO (RQ2). Seven Ontology Design Patterns were developed, reusing existing PMDco patterns for process chains and chemical composition where applicable and contributing novel patterns for microstructural characterization and for distinguishing planned from measured test parameters. Because each pattern was designed and tested in isolation before integration, the same patterns and workflow steps are positioned for reuse by future mechanical-testing ontologies without requiring domain-specific redesign, and the novel patterns have been proposed back to the PMDco pattern library.
3. CTO was demonstrated as the semantic backbone of an ontology-guided LLM extraction and KG construction pipeline (RQ3). Using the LLM4CreepLitKG tool, ontology-guided extraction from the 20-paper gold-standard dataset achieved an overall field accuracy above 85% against the human-annotated benchmark. The extracted data was transformed through a four-step RML-based pipeline into a CTO-compliant RDF graph, validated against SHACL shapes, and deployed on the MaterialDigital Dataportal as CreepLitKG, a live, queryable knowledge graph covering 100 Nickel-based creep papers. Data retrieval via SPARQL queries derived from the competency questions was manually verified against the source publications and confirmed correct in every case, showing that CTO is not only theoretically sound but practically usable for automated literature mining and FAIR knowledge graph construction.
4. Taken together, these results constitute the thesis's scientific contribution. CTO is the first BFO-compliant domain ontology for creep testing of metallic materials aligned with ISO 204, closing a gap left open by application ontologies for tensile, fatigue, and hardness testing and by the polymer-focused OntoFNCT. The reproducible, ODK-based development pipeline (BFO alignment, PMDco reuse, ODP-based modelling, SHACL validation, and Widoco documentation under version control) offers a methodological template for other domain-specific MSE ontologies, and the integration with an LLM-based extraction pipeline contributes concrete evidence to the emerging field of ontology-guided LLM extraction, where CTO's shared vocabulary constrains LLM outputs

to consistent, BFO-aligned concepts and enables automated validation, SPARQL querying, and cross-study integration that schema-free extraction cannot offer.

References

[1] L. D. Xu, E. L. Xu, and L. Li, “Industry 4.0: state of the art and future trends,” *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, Apr. 2018, doi: 10.1080/00207543.2018.1444806.
[2] M. Ghobakhloo, “Industry 4.0, digitization, and opportunities for sustainability,” *Journal of Cleaner Production*, vol. 252, p. 119869, Apr. 2020, doi: 10.1016/j.jclepro.2019.119869.
[3] B. Bayerlein *et al.*, “PMD Core Ontology: Achieving semantic interoperability in materials science,” *Materials & Design*, vol. 237, p. 112603, Jan. 2024, doi: 10.1016/j.matdes.2023.112603.
[4] H. B. Nasrabadi *et al.*, “Toward a digital materials mechanical testing lab,” *Computers in Industry*, vol. 153, p. 104016, Dec. 2023, doi: 10.1016/j.compind.2023.104016.
[5] B. Bayerlein *et al.*, “A Perspective on Digital Knowledge Representation in Materials Science and Engineering,” *Advanced Engineering Materials*, vol. 24, no. 6, p. 2101176, 2022, doi: 10.1002/adem.202101176.
[6] M. Scheffler *et al.*, “FAIR data enabling new horizons for materials research,” *Nature*, vol. 604, no. 7907, pp. 635–642, Apr. 2022, doi: 10.1038/s41586-022-04501-x.
[7] European Commission. Directorate General for Research and Innovation. and PwC EU Services., *Cost-benefit analysis for FAIR research data: cost of not having FAIR research data.* LU: Publications Office, 2018. doi: 10.2777/02999.
[8] M. D. Wilkinson *et al.*, “The FAIR Guiding Principles for scientific data management and stewardship,” *Sci Data*, vol. 3, no. 1, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.
[9] N. Majithia, T. Carey‐Wilson, E. Simperl, and N. Shadbolt, “An actionable framework for AI‐ready data,” *AI Magazine*, vol. 47, no. 1, p. e70054, Mar. 2026, doi: 10.1002/aaai.70054.
[10] N. Reeves-McLaren and S. Moth-Lund Christensen, “Data integrity in materials science in the era of AI: balancing accelerated discovery with responsible science and innovation,” *J. Mater. Chem. A*, vol. 14, no. 1, pp. 276–283, 2026, doi: 10.1039/D5TA05512A.
[11] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data & Knowledge Engineering*, vol. 25, no. 1–2, pp. 161–197, Mar. 1998, doi: 10.1016/S0169-023X(97)00056-6.
[12] D. Fensel, *Ontologies: a silver bullet for knowledge management and electronic commerce*. Berlin Heidelberg: Springer, 2001.
[13] F. J. Ekaputra, M. Sabou, E. Serral, E. Kiesling, and S. Biffl, “Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review,” vol. 4, no. 1, 2017.
[14] G. Antoniou and F. van Harmelen, “Web Ontology Language: OWL,” in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., Berlin, Heidelberg: Springer, 2009, pp. 91–110. doi: 10.1007/978-3-540-92673-3_4.
[15] “OWL Web Ontology Language Reference.” Accessed: Jul. 26, 2026. [Online]. Available: https://www.w3.org/TR/owl-ref/
[16] N. Guarino, “Understanding, building and using ontologies,” *International Journal of Human-Computer Studies*, vol. 46, no. 2–3, pp. 293–310, Feb. 1997, doi: 10.1006/ijhc.1996.0091.
[17] A. Haller and A. Polleres, “Are we better off with just one ontology on the Web?,” *SW*, vol. 11, no. 1, pp. 87–99, Jan. 2020, doi: 10.3233/SW-190379.
[18] D. B. Lenat, “CYC: a large-scale investment in knowledge infrastructure,” *Commun. ACM*, vol. 38, no. 11, pp. 33–38, Nov. 1995, doi: 10.1145/219717.219745.
[19] I. Niles and A. Pease, “Towards a standard upper ontology,” in *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, Ogunquit Maine USA: ACM, Oct. 2001, pp. 2–9. doi: 10.1145/505168.505170.

[20] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, “Sweetening Ontologies with DOLCE,” in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, vol. 2473, A. Gómez-Pérez and V. R. Benjamins, Eds., in Lecture Notes in Computer Science, vol. 2473. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 166–181. doi: 10.1007/3-540-45810-7_18.
[21] P. Grenon and B. Smith, “SNAP and SPAN: Towards Dynamic Spatial Ontology,” *Spatial Cognition & Computation*, vol. 4, no. 1, pp. 69–104, Mar. 2004, doi: 10.1207/s15427633scc0401_5.
[22] S. J. D. Cox and C. Little, “Time Ontology in OWL,” World Wide Web Consortium (W3C), W3C Recommendation, Oct. 2017. [Online]. Available: https://www.w3.org/TR/owl-time/
[23] T. Lebo, S. Sahoo, and D. McGuinness, “PROV-O: The PROV Data Model,” World Wide Web Consortium (W3C), W3C Recommendation, Apr. 2013. [Online]. Available: https://www.w3.org/TR/prov-o/
[24] D. Brickley and L. Miller, “FOAF Vocabulary Specification 0.99.” Jan. 14, 2014. [Online]. Available: http://xmlns.com/foaf/spec/
[25] D. Reynolds, “The Organization Ontology,” World Wide Web Consortium (W3C), W3C Recommendation, Jan. 2014. [Online]. Available: https://www.w3.org/TR/vocab-org/
[26] A. Haller *et al.*, “The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation,” *SW*, vol. 10, no. 1, pp. 9–32, Dec. 2018, doi: 10.3233/SW-180320.
[27] M. Ashburner *et al.*, “Gene Ontology: tool for the unification of biology,” *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000, doi: 10.1038/75556.
[28] L. M. Schriml *et al.*, “Disease Ontology: a backbone for disease metadata integration,” *Nucleic Acids Research*, vol. 40, no. D1, pp. D940–D946, 2011, doi: 10.1093/nar/gkr972.
[29] K. Degtyarenko *et al.*, “ChEBI: a database and ontology for chemical entities of biological interest,” *Nucleic Acids Research*, vol. 36, no. suppl_1, pp. D344–D350, 2008, doi: 10.1093/nar/gkm781.
[30] M. H. Rasmussen, M. Lefranois, P. Pauwels, C. A. Hviid, and J. Karlshøj, “Managing Interconnected Building Data with the Building Topology Ontology,” in *Proceedings of the 17th International Conference on Computing in Civil and Building Engineering (ICCCBE 2018)*, 2019. doi: 10.1007/978-3-030-00220-6_11.
[31] R. Arp, B. Smith, and A. D. Spear, *Building Ontologies with Basic Formal Ontology*. The MIT Press, 2015. doi: 10.7551/mitpress/9780262527811.001.0001.
[32] ISO/IEC JTC 1/SC 32, “Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO),” International Organization for Standardization, Geneva, CH, Standard ISO/IEC 21838-2:2021, 2021. [Online]. Available: https://www.iso.org/standard/74572.html
[33] The OBI Consortium *et al.*, “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration,” *Nat Biotechnol*, vol. 25, no. 11, pp. 1251–1255, Nov. 2007, doi: 10.1038/nbt1346.
[34] N. F. Noy and D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”.
[35] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, “The NeOn Methodology for Ontology Engineering,” in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–34. doi: 10.1007/978-3-642-24794-1_2.
[36] A. Gangemi and V. Presutti, “Ontology Design Patterns,” in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 221–243. doi: 10.1007/978-3-540-92673-3_10.
[37] E. Blomqvist, P. Hitzler, K. Janowicz, A. Krisnadhi, T. Narock, and M. Solanki, “Considerations regarding Ontology Design Patterns,” *SW*, vol. 7, no. 1, pp. 1–7, Nov. 2015, doi: 10.3233/SW-150202.
[38] F. Scharffe, O. Zamazal, and D. Fensel, “Ontology alignment design patterns,” *Knowl Inf Syst*, vol. 40, no. 1, pp. 1–28, Jul. 2014, doi: 10.1007/s10115-013-0633-y.
[39] “Usage Patterns,” PMD Core Ontology (PMDco). Accessed: Jul. 26, 2026. [Online]. Available: https://materialdigital.github.io/core-ontology/docs/patterns.html

[40] H. Beygi Nasrabadi, E. Norouzi, K. Hubaiev, J. Waitelonis, and H. Sack, “NFDI MatWerk Ontology (MWO): A BFO‐Compliant Ontology for Research Data Management in Materials Science and Engineering,” *Adv Eng Mater*, p. e202502331, Nov. 2025, doi: 10.1002/adem.202502331.
[41] A. Hogan *et al.*, “Knowledge Graphs,” *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, May 2022, doi: 10.1145/3447772.
[42] R. Cyganiak, D. Wood, and M. Lanthaler, “RDF 1.1 Concepts and Abstract Syntax,” World Wide Web Consortium (W3C), W3C Recommendation, Feb. 2014. [Online]. Available: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
[43] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” *SW*, vol. 8, no. 3, pp. 489–508, Dec. 2016, doi: 10.3233/SW-160218.
[44] P. Hitzler, M. Krötzsch, and S. Rudolph, *Foundations of semantic web technologies*. in Chapman & Hall/CRC textbooks in computing. Boca Raton (Fla.): CRC press, 2009.
[45] H. B. Nasrabadi and B. Skrotzki, “Ontology-Oriented Modeling of the Vickers Hardness Knowledge Graph,” presented at the 2023 10th International Conference on Mechanics, Materials and Manufacturing, Washington, USA, Apr. 2024, pp. 33–38. doi: 10.4028/p-k8Gj2L.
[46] Y. Zhu *et al.*, “LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities,” *World Wide Web*, vol. 27, no. 5, p. 58, 2024, doi: 10.1007/s11280-024-01289-w.
[47] V. Venugopal and E. Olivetti, “MatKG: An autonomously generated knowledge graph in Material Science,” *Scientific Data*, vol. 11, no. 1, p. 217, 2024, doi: 10.1038/s41597-024-03039-z.
[48] H.-T. Bian, “LLM-empowered knowledge graph construction: A survey,” *arXiv preprint arXiv:2510.20345*, 2025, doi: 10.48550/arXiv.2510.20345.
[49] Y. Ling, Z. Qin, and Z. Ma, “A review of knowledge graph construction using large language models in transportation: Problems, methods, and challenges,” *Transportation Research Part C: Emerging Technologies*, vol. 183, p. 105428, Feb. 2026, doi: 10.1016/j.trc.2025.105428.
[50] M. Trajanoska, R. Stojanov, and D. Trajanov, “Enhancing Knowledge Graph Construction Using Large Language Models,” in *2023 IEEE International Conference on Big Data (Big Data)*, IEEE, 2023, pp. 3288–3297. doi: 10.1109/BigData59044.2023.10386762.
[51] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, and K. Lata, “Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text,” in *The Semantic Web – ISWC 2023*, in Lecture Notes in Computer Science, vol. 14265. Springer, Cham, 2023, pp. 247–265. doi: 10.1007/978-3-031-47240-4_13.
[52] T. V. Ivanisenko, P. S. Demenkov, and V. A. Ivanisenko, “An Accurate and Efficient Approach to Knowledge Extraction from Scientific Publications Using Structured Ontology Models, Graph Neural Networks, and Large Language Models,” *IJMS*, vol. 25, no. 21, p. 11811, Nov. 2024, doi: 10.3390/ijms252111811.
[53] E. Norouzi, N. Jung, A. M. Jacyszyn, J. Waitelonis, and H. Sack, “AI4DiTraRe: Building the BFO-Compliant Chemotion Knowledge Graph,” 2025, *arXiv*. doi: 10.48550/ARXIV.2509.01536.
[54] A. R. Durmaz, A. Thomas, L. Mishra, R. N. Murthy, and T. Straub, “An ontology-based text mining dataset for extraction of process-structure-property entities,” *Scientific Data*, vol. 11, no. 1, p. 1112, 2024, doi: 10.1038/s41597-024-03926-5.
[55] S. T. Potu, R. N. Murthy, A. Thomas, L. Mishra, N. Prange, and A. R. Durmaz, “Ontology-conformal recognition of materials entities using language models,” *Scientific Reports*, vol. 15, no. 1, p. 18597, 2025, doi: 10.1038/s41598-025-03619-y.
[56] S. Ghosh and A. Tewari, “Automated Extraction of Material Properties using LLM-based AI Agents,” *arXiv preprint arXiv:2510.01235*, 2025, doi: 10.48550/arXiv.2510.01235.
[57] H. B. Nasrabadi and B. Skrotzki, “Digital Representation of Materials Testing Data for Semantic Web Analytics: Tensile Stress Relaxation Testing Use Case,” *KEM*, vol. 987, pp. 47–52, Sep. 2024, doi: 10.4028/p-xSmHN2.
[58] R. C. Reed, *The superalloys: fundamentals and applications*, Digital printed version. Cambridge: Cambridge Univ. Press, 2008.

[59] ISO/TC 164/SC 1, “Metallic materials — Uniaxial creep testing in tension — Method of test,” International Organization for Standardization, Geneva, CH, Standard ISO 204:2023, 2023. [Online]. Available: https://www.iso.org/standard/82162.html

[60] H. Beygi Nasrabadi, E. Norouzi, H. Sack, and B. Skrotzki, “Performance Evaluation of Upper‐Level Ontologies in Developing Materials Science Ontologies and Knowledge Graphs,” *Adv Eng Mater*, vol. 27, no. 8, p. 2401534, Apr. 2025, doi: 10.1002/adem.202401534.

[61] J. N. Otte, J. Beverley, and A. Ruttenberg, “BFO: Basic Formal Ontology1,” *AO*, vol. 17, no. 1, pp. 17–43, Mar. 2022, doi: 10.3233/AO-220262.

[62] E. G. Bologne University of, G. G. Consulting Goldbeck, J. F. SINTEF, A. H. IWM Fraunhofer, and G. S. ACCESS, “Europeean Materials Modelling Ontology.” Accessed: Jul. 26, 2026. [Online]. Available: https://emmo-repo.github.io/versions/1.0.0-beta/emmo.html

[63] S. Borgo *et al.*, “DOLCE: A descriptive ontology for linguistic and cognitive engineering1,” *AO*, vol. 17, no. 1, pp. 45–69, Mar. 2022, doi: 10.3233/AO-210259.

[64] R. Rudnicki, “An Overview of the Common Core Ontologies,” CUBRC, Inc., Buffalo, NY, Technical Report, 2019. [Online]. Available: https://www.nist.gov/document/nist-ai-rfi-cubrcinc004pdf

[65] P. Del Nostro, G. Goldbeck, and D. Toti, “CHAMEO: An ontology for the harmonisation of materials characterisation methodologies,” *AO*, vol. 17, no. 3, pp. 401–421, Aug. 2022, doi: 10.3233/AO-220271.

[66] J. Friis, G. Goldbeck, S. Gouttebroze, F. L. Bleken, and E. Ghedini, “Materials science and ontologies,” in *Digitalization and Sustainable Manufacturing*, 1st ed., London: Routledge, 2024, pp. 117–135. doi: 10.4324/9781032693415-9.

[67] H. B. Nasrabadi, T. Hanke, and B. Skrotzki, “Semantic Representation of Low‐Cycle‐Fatigue Testing Data Using a Fatigue Test Ontology and ckan.kupferdigital Data Management System,” *Adv Eng Mater*, vol. 27, no. 8, p. 2400675, Apr. 2025, doi: 10.1002/adem.202400675.

[68] M. Schilling, N. Marschall, U. Niebergall, and M. Böhning, “Modernizing FNCT data handling in polymer labs: Towards efficient management,” *Computational Materials Science*, vol. 259, p. 114085, Sep. 2025, doi: 10.1016/j.commatsci.2025.114085.

[69] M. Schilling, N. Marschall, U. Niebergall, and M. Böhning, “FAIR data in mechanical testing of polymers: Implementation of the full-notch creep test ontology (OntoFNCT),” *Results in Engineering*, vol. 30, p. 110549, Jun. 2026, doi: 10.1016/j.rineng.2026.110549.

[70] T. Ashino and M. Fujita, “Definition of a web ontology for design-oriented material selection,” *Data Sci. J.*, vol. 5, pp. 52–63, 2006, doi: 10.2481/dsj.5.52.

[71] A. Zeb and T. Mätäsniemi, “Ontological modelling of creep void analysis data to automate machine learning training process,” Jul. 08, 2022, *In Review*. doi: 10.21203/rs.3.rs-1807190/v1.

[72] L. Ávila Calderón, S. Schriever, Y. Han, J. Olbricht, P. D. Portella, and B. Skrotzki, “Creep reference data of single-crystal Ni-based superalloy CMSX-6,” *Data in Brief*, vol. 65, p. 112436, Apr. 2026, doi: 10.1016/j.dib.2025.112436.

[73] C. M. Keet and Z. C. Khan, “Characterising Competency Questions for Ontologies”.

[74] C. M. Keet and Z. C. Khan, “On the Roles of Competency Questions in Ontology Engineering,” in *Knowledge Engineering and Knowledge Management*, vol. 15370, M. Alam, M. Rospocher, M. Van Erp, L. Hollink, and G. A. Gesese, Eds., in Lecture Notes in Computer Science, vol. 15370. , Cham: Springer Nature Switzerland, 2025, pp. 123–132. doi: 10.1007/978-3-031-77792-9_8.

[75] IAO Consortium, “Information Artifact Ontology (IAO).” [Online]. Available: http://purl.obolibrary.org/obo/iao.owl

[76] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, and B. Aleman-Meza, “OntoQA: Metric-Based Ontology Quality Analysis,” in *Proceedings of the IEEE ICDM 2005 Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, 2005.