Neural Computing and Applications (2025) 37:11737–11753
https://doi.org/10.1007/s00521-024-10674-5

S.I.: INNOVATIONS IN AI-BASED SYSTEMS AND SOFTWARE

Modeling dislocation dynamics data using semantic web technologies

Ahmad Zainul Ihsan[1] · Said Fathalla[1] · Stefan Sandfeld[1,2]

Received: 14 September 2023 / Accepted: 5 July 2024 / Published online: 14 December 2024

Abstract
The research in Materials Science and Engineering focuses on the design, synthesis, properties, and performance of materials. An important class of materials that is widely investigated are crystalline materials, including metals and semiconductors. Crystalline material typically contains a specific type of defect called "dislocation". This defect significantly affects various material properties, including bending strength, fracture toughness, and ductility. Researchers have devoted a significant effort in recent years to understanding dislocation behaviour through experimental characterization techniques and simulations, e.g., dislocation dynamics simulations. This paper presents how data from dislocation dynamics simulations can be modelled using semantic web technologies through annotating data with ontologies. We extend the dislocation ontology by adding missing concepts and aligning it with two other domain-related ontologies (i.e., the Elementary Multi-perspective Material Ontology and the Materials Design Ontology), allowing for efficiently representing the dislocation simulation data. Moreover, we present a real-world use case for representing the discrete dislocation dynamics data as a knowledge graph (DisLocKG) which can depict the relationship between them. We also developed a SPARQL endpoint that brings extensive flexibility for querying DisLocKG.

Keywords Ontology · Knowledge graph · Reasoning · Dislocation · Crystallographic defects · Semantic web

1 Introduction

Plastic deformation in metals and other crystalline materials can be attributed to a one-dimensional lattice defect type known as dislocation. The concept of dislocation was introduced in the 1930 s by Taylor [1] and Polanyi [2]. Dislocations determine the mechanical properties of materials, such as strength, hardness, and ductility. For instance, materials engineers have discovered the strengthening mechanism of crystalline materials by studying the relationship between dislocation motion and the mechanical behaviour of metals [3]. By controlling the motion of dislocations in crystalline materials, materials engineers can build, for example, an aircraft turbine blade that can withstand an operation temperature of $\sim 1000°C$ and creep deformation due to centrifugal forces whereas the turbine is rotating [4]. Significant efforts have been made to understand dislocation systems using dedicated microscopy techniques and simulation methods. These simulation methods, along with other techniques, have been created to predict dislocation evolution.

In recent years, data-driven approaches have brought new methods and tools for analyzing and understanding the evolution of dislocation systems [5–10]. This intensely transforms Materials Science and Engineering (MSE), combining simulations, data mining, and experiments, making the digital transformation possible [11, 12]. However, a digital transformation without being supported by the appropriate data infrastructure often ends up with isolated and inaccessible data repositories, the so-called "data silos". In this regard, materials informatics plays a significant role in materials science research to overcome the data silos problem. This is because materials informatics

✉ Stefan Sandfeld
s.sandfeld@fz-juelich.de

Ahmad Zainul Ihsan
a.ihsan@fz-juelich.de

Said Fathalla
s.fathalla@fz-juelich.de

[1] Institute of Advanced Simulation–Materials Data Science and Informatics (IAS-9), Forschungszentrum Jülich, Jülich, Germany

[2] Faculty of Georesources and Materials Engineering, RWTH Aachen University, Aachen, Germany

combines two discourses of materials science and information technologies to tackle the major problems in materials science, such as data management and analysis. Moreover, it helps to develop intelligent systems to, e.g., explore materials, find novel material properties, or study the behaviour of a specific material phenomenon.

To fully understand the behaviour of materials and, in particular, dislocations, aspects from different length scales need to be considered. This variety of length scales makes the knowledge representation of systems of dislocations challenging, even though this has yet to be perceived as a significant research hindrance in materials science. Generally, the schematic representation of knowledge, i.e., the representation through ontologies, can significantly boost data management and analysis; it helps to extract knowledge from data. Ontologies also allow the domain knowledge to be machine-understandable, meaning that machines can read and interpret this knowledge efficiently. Furthermore, it has become an essential part of achieving FAIR (Discoverable, Accessible, Interoperable, and Reusable) data [13].

This paper presents how Discrete Dislocation Dynamics (DDD) data can be enriched using semantic web technologies, such as the Resource Description Framework (RDF) [14], the Web Ontology Language (OWL) [15] and SPARQL [16]. The first step we have taken is to adapt and extend the dislocation ontology (DISO) [17, 18] so that it can model various concepts and relationships in the DDD domain. The adaptation includes adding missing concepts, improving class definition, exploring additional relationships between concepts, and finally aligning it with other domain-related ontologies, including the Elementary Multi-Perspective Material Ontology (EMMO) and the Materials Design Ontology (MDO). This allows for representing the dislocation simulation data efficiently. DISO is one of the Dislocation Ontology Suite (DISOS)[1] ontologies that represent the concepts and relationships of linear defects in crystalline materials. DISOS comprises several modules describing the material's scientific concepts, representations of dislocations, and different simulation models in the dislocation domain. The adapted version of DISO is developed and maintained in the DISOS GitHub repository. The ontology is available in several RDF serializations via a persistent identifier (i.e., https://purls.helmholtz-metadaten.de/disos/diso) provided by PIDA (Persistent Identifiers for Digital Assets).[2] PIDA employs content negotiation [19] to serve different versions of the ontology (i.e., the HTML documentation or an RDF representation) via its IRI. DISO has been syntactically validated by the W3C RDF validation service[3] to conform with the W3C RDF standards. The documentation of the ontology is available via its IRI.

The next step after adapting the ontology is to annotate the data gathered from multiple DDD simulations with the adapted version of DISO resulting in a knowledge graph (DisLocKG) of DDD data (more details can be found in Sect. 6). This knowledge graph relates DDD data concepts, thus facilitating machine actionability [20], semantic querying, inference of implicit knowledge, and data consistency and integrity. The objective is to convert the unstructured DDD data to linked data with dereferenceable IRIs that adhere to W3C standards and best practices. This will enable not only reasoning about the dislocation data but also integrating it into other MSE-related fields. We have made DisLocKG publicly available via its GitHub repository.[4]

2 Related work

Over the past few years, many researchers have given particular attention to developing ontologies to represent scientific data in different fields of science, such as physics [21], agriculture [22], and pharmaceutical science [23]. Specifically in the MSE field, several efforts have been made to create ontologies representing materials-related notions or semantically presenting actual materials data as knowledge graphs. This section will discuss related works of knowledge graphs with and without semantic web technology (including RDF, OWL, and SPARQL). Two examples from the latter are the *Propnet* Knowledge Graph [24] and the Materials Knowledge Graph (*MatKG*) [25].

The Propnet is a knowledge graph enhancing materials properties data from the Materials Project [26] Repository.[5] It augments base properties data (e.g., lattice, basis, chemical formula, band gap, and total energy) resulting from the ab-initio calculation into derived properties, e.g., Debye temperature, bulk modulus, and shear modulus. The workflow and input for generating the augmented data are subsequently stored in the knowledge graph.

On the other hand, the MatKG stores metadata from over 2.9 million materials science articles. This metadata includes abstracts, titles, keywords, and author data (e.g., name, email, affiliation, and ORCID). By accessing the MatKG, we can retrieve information such as the milestones of a material developed by multiple authors.

[27] have developed a "Materials Ontology" which is an ontology describing substances, processes,

[1] https://purls.helmholtz-metadaten.de/disos.

[2] https://purls.helmholtz-metadaten.de/.

[3] https://www.w3.org/RDF/Validator/.

[4] https://purls.helmholtz-metadaten.de/dislockg.

[5] https://next-gen.materialsproject.org.

environments, and properties. This ontology has also been used to exchange data between three different thermal property databases.

In the solid-state physics domain, [28] have developed the *Materials Design Ontology* (MDO) which is an ontology covering knowledge in the field of materials design, e.g., with regards to ab-initio methods. MDO is used to represent materials' data related to ab-initio calculations over disparate materials data repositories as RDF triples. At the time of writing the paper, a total of ≈ 4.3K RDF triples had been collected in their repository.[6] While the work is related to the representation of crystalline material through the crystal structure, MDO does not represent data related to crystalline defects.

Another effort in the experimental materials science community that uses semantic web technologies is the NanoMine Knowledge Graph[7] [29]. It is a knowledge graph for polymer nanocomposite materials, integrating diverse data from more than 1,700 polymer nanocomposite experiments. Moreover, the authors of the NanoMine knowledge graph have developed the NanoMine ontology,[8] which is a backbone ontology to describe polymer nanocomposite experiments.

In conclusion, it is evident that even though several efforts and groups utilizing the semantic web in various MSE-related fields have progressed significantly, work for semantically representing dislocations simulation data is still missing. We believe this work is a pioneering attempt to create a knowledge graph in an MSE-related domain, specifically dealing with dislocation data. As a result, the unstructured dislocation data is transformed into linked data with dereferenceable IRIs using persistent URLs, adhering to W3C standards and best practices. This enables not only the annotation of dislocation data by an ontology but also the integration of dislocation data into other MSE-related fields.

3 Description of the domain

This section briefly describes the relevant notions and concepts of line defects within the crystalline materials domain.

3.1 Representation of crystalline materials

Most metals and metallic materials have a crystalline structure, which implies that the atoms are arranged in a periodic structure with a high degree of symmetry. This periodic arrangement is at the basis of the *crystal structure* model, idealizing the physical concept of crystalline materials. For example, in Fig. 1 atoms are shown in an idealized manner as small spheres.

The crystal structure is represented by the *lattice* together with a *motif*: the lattice is a mathematical concept of an infinite, repeating arrangement of points in space (3D), in a plane (2D), or on a line (1D), in which all points have the same surrounding and coincide with atom positions. The motif (or base) consists of an arrangement of chemical species, which can be atoms, ions, or molecules in crystalline materials. By putting a motif of one or more atoms at every lattice point, the crystal structure can be represented.

It is now possible to identify the smallest atom pattern that can be repeated along all spatial directions to cover the entire structure. This pattern is called a *unit cell*, shown as the black cube in Fig. 1. The lattice parameters of the unit cell consist of the angles between the edges and the edge lengths.

Figure 2 shows the six lattice parameters needed to characterize the unit cell: three lengths (a, b, c) and three angles (α, β, γ). These parameters also constitute the basis vectors in the crystal coordinate system; they are not necessarily mutually perpendicular. Unit cells are often classified into a systematic based on the lattice parameters (cf. Fig.3).

For instance, the cubic system has $a = b = c,\ \alpha = \beta = \gamma = 90°$ and the orthorhombic system has $a \neq b \neq c,\ \alpha = \beta = \gamma = 90°$. Seven crystal systems are often ordered according to the increasing symmetry: cubic, tetragonal, orthorhombic, hexagonal, rhombohedral, monoclinic, and triclinic.

In the unit cell, we can also define *lattice points*, *lattice directions*, and *lattice planes*: A lattice consists of lattice points where the atoms, ions, or molecules are located (the leftmost cube in Fig. 4). The vector position of lattice points, $\overrightarrow{R}$, is described by the equation

$$\overrightarrow{R} = n_1\mathbf{a} + n_2\mathbf{b} + n_3\mathbf{c}, \tag{1}$$

where n_i are arbitrary integers and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are basis vectors (pointing along the axes in Fig. 2) derived from the lattice

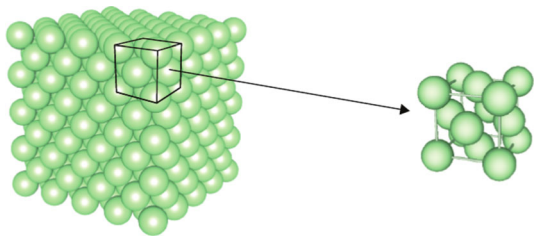

Fig. 1 The crystal structure of face-centred cubic comprises an aggregate of atoms within one unit cell in crystalline materials

[6] https://github.com/LiUSemWeb/materials-design-ontology.

[7] http://nanomine.org/.

[8] https://github.com/tetherless-world/nanomine-ontology.

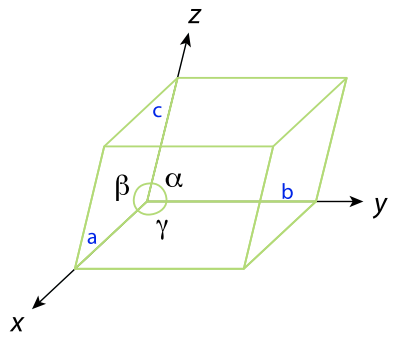

Fig. 2 The geometry of a unit cell is exactly defined through the three lengths, a, b, and c, and the three angles α, β, and γ

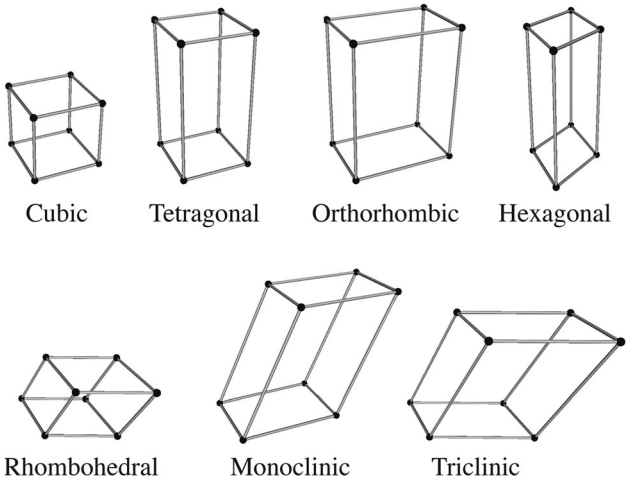

Fig. 3 The seven crystal systems. These seven crystal systems are also seven primitive Bravais lattices. Each of them only corresponds to a single lattice point

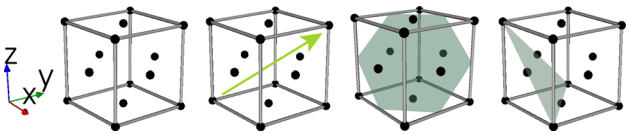

Fig. 4 On the very left, a unit cell is shown that corresponds to a "face-centered cubic" structure. The black points indicate the lattice points, i.e. the positions of the atoms. Second from left: a direction vector that connects to these lattice points. Lastly, two different lattice planes are shown

parameters. As illustrated in Fig. 4, a lattice direction or lattice vector is a vector connecting two lattice points, whereas a lattice plane forms an infinitely stretched plane (characterized through a plane normal) that cuts through lattice points such that a regular arrangement of lattice points in the plane occurs.

As shown in the left panel of Fig. 4, so-called "lattice points" can be thought of as the attachment points of the "unit cells"-the parallelepipeds shown in grey color. The three-dimensional position of the lattice points is described by the vector $\vec{R} = n_1\mathbf{a} + n_2\mathbf{b} + n_3\mathbf{c}$, where n_i are any integers and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are basis vectors derived from the lattice parameters. Eight unit cells are adjoined at every corner, resulting in a so-called primitive lattice: The primitive lattice is a lattice where each unit cell corresponds to exactly one lattice point. In contrast, the non-primitive lattice is the lattice where a unit cell corresponds to more than one single lattice point. The combination of primitive and non-primitive lattices in 3D space defines the 14 Bravais lattices, consisting of 7 primitive and 7 non-primitive lattices.

3.2 Description of linear defects

In crystalline materials, atoms are not always arranged or positioned perfectly. Typically, different kinds of crystallographic defects lead to disruption of the local order in a material (in addition to thermal fluctuations affecting the atomic positions). A common type of such defect is the dislocation, which causes a strongly localized, tube-like region of disorder (illustrated by the dashed circle in the right panel of Fig. 5; this tube-like region stretches along the z-direction).

This region contains the highly disordered dislocation core at the center. Further away from the dislocation core, the perfect lattice structure is restored, even though there is now a row of atoms shifted into the new position as indicated by the red spheres in Fig. 5.

The "Burgers vector" of a dislocation can be defined through the "Burgers circuit", as shown in Fig. 6. A Burgers circuit is an atom-to-atom path that is closed in a perfect crystal (left panel of the figure). The length of the path is given as a multiple of the atomic distances in two directions. In the presence of a dislocation, a path of the same lengths would not be closed (right panel of the figure). The step-by-step procedure is as follows: We define a reference point C as the start point of the path. The line sense of the path is given by $\boldsymbol{\xi}$ assuming the "right-hand convention" (the thumb points along the vector $\boldsymbol{\xi}$ into the picture plane and the other fingers curl around that vector; their fingertips indicate the direction of the path).

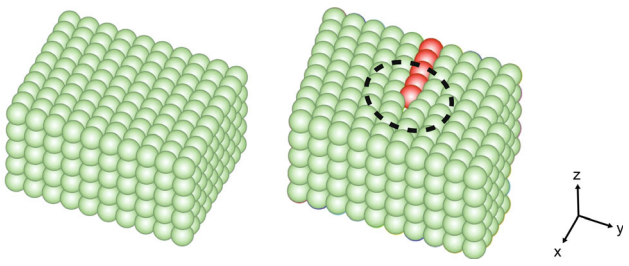

Fig. 5 The left figure shows the perfect order of atoms in a crystalline material. The right crystal contains a dislocation that destroys the local order of the crystal structure. Note that red-colored atoms only show the 'irregularity", and are not different atom types

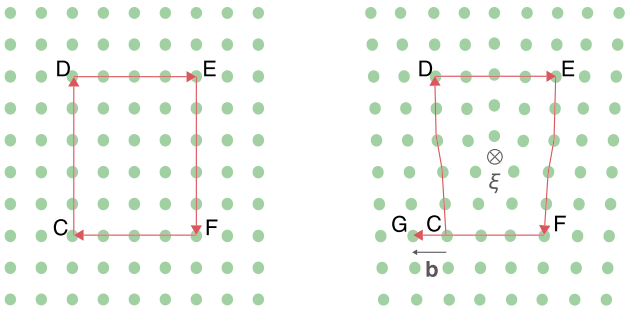

Fig. 6 The Burgers circuit in the crystalline materials. While the left panel has the Burgers circuit in the perfect crystal material. The right panel has the Burgers circuit around the dislocation in the crystalline materials. Due to the closure failure in the defective crystalline materials, we can define the Burgers vector, **b**

The symbol ⊗ in the figure indicates that the vector $\boldsymbol{\xi}$ points into the picture plane. In the perfect crystal, this circuit goes up five atoms from the reference point, four atoms to the left, five atoms down, and four atoms to the left to close a circuit at point *C* again. With the same reference point and the same number of atomic spacings as in the circuit, the perfect crystalline material can be used with the crystal containing a dislocation. However, the circuit does not end at the same point as in the perfect crystal, but rather at point *G* as shown in the right panel of Fig. 6. The vector connecting the start point *C* with the finish point *G* is the Burgers vector, **b**.

There are two fundamentally different types of dislocations: "screw" and "edge" dislocations. A screw dislocation has a line sense parallel to its Burgers vector, $\boldsymbol{\xi} \| \boldsymbol{b}$, whereas for an edge dislocation, the line sense is perpendicular to its Burgers vector, $\boldsymbol{\xi} \perp \boldsymbol{b}$. Thus, Fig. 6 shows an edge dislocation.

In reality, screw and edge dislocations are extreme cases, while the most general case of a dislocation type in a crystalline material is a "mixed dislocation." This is a dislocation with the line sense, $\boldsymbol{\xi}$, neither parallel nor perpendicular to its Burgers vector, **b**.

Since the atoms around dislocations are not positioned at the perfect lattice points, the lattice is distorted near a dislocation. This distortion results in a stress field in the crystalline material around the dislocation which is the reason why dislocations move: they try to minimize energy. In the context of plastic deformation, a dislocation is defined as the boundary of a slipped area within which atoms are displaced by the size of an elementary unit translation given by the Burgers vector.

In materials science, often the question arises on which "granularity level" a dislocation should be defined. Clearly, if we are interested in phenomena on the nanometer scale, then we should resolve individual atoms (e.g., through high-resolution transmission electron microscopy or molecular dynamics simulations). When taking the *mesoscopic* perspective, typically the individual atoms cannot be seen anymore and are not of interest (e.g., done through regular transmission electron microscopy or dislocations dynamics simulations). However, the dislocation line itself is still observable: the tube-like defect "region" is reduced to an idealized mathematical line, as demonstrated in the right panel of Fig. 7.

Therefore, the transition from the atomic scale to the mesoscale requires a conceptual and mathematical idealization that significantly reduces the amount of information. These idealizations require to be accompanied by further details and definitions from the atomic scale, including the crystal structure, the lattice, the lattice plane, and the lattice direction information, all of which have an impact on the dislocation's motion. For example, the motion of a dislocation line through a crystal is constrained to a specific crystallographic or lattice plane. Thus, it still requires crystallographic information, even though the "defect region" is now only represented as a mathematical line. These two different levels of information require particular attention when designing the dislocation ontology.

The particular crystallographic or lattice plane constraining the dislocation motion is called the *slip plane* (see the green plane in Fig. 8). There are specific *slip directions*, which are lattice directions along which plastic deformation occurs within the slip plane, given by the Burgers vector. A *slip system* is a set of slip planes with the same unit normal vector and the same slip direction. Thus, the unit normal vector and the slip direction or the Burgers vector (where the latter is not a unit vector) determine the slip system.

The mathematical representation of a mesoscale dislocation, as shown in Fig. 8, is an oriented curve with a start point and an endpoint. The local line orientation changes along the line, while the Burgers vector is constant for each point. Since the dislocation is a directed curve, it has a line sense. Unlike the local line orientation, it is a property of the whole line.

Various computational and experimental techniques are leveraged to predict and observe dislocations in crystalline

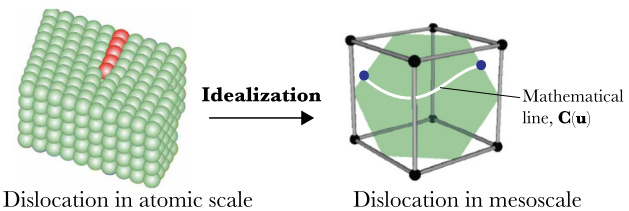

Fig. 7 The idealization represents the dislocation in the mesoscale. Here, the individual atoms are no longer visible. This idealization reduced the tube-like defect "region" to a mathematical line. Note that the line on the right does not correspond to the dislocation on the left (this would be a vertical, straight line)

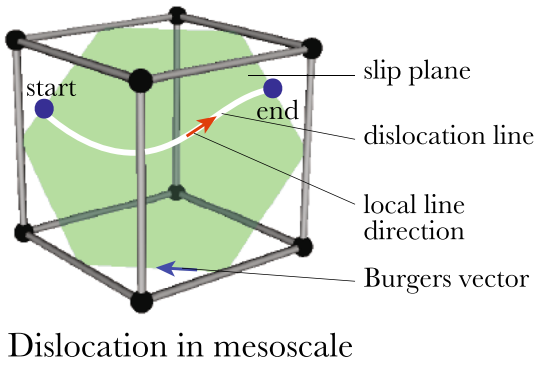

Fig. 8 Depiction of the mathematical dislocation line on the mesoscale as a mathematical object that has start and end points. The object is characterized by the Burgers vector and the line sense. Furthermore, the dislocation motion is constrained by the slip plane

materials, some of which were already mentioned above. For instance, high-resolution transmission electron microscopy (or field ion microscopy) is used on the atomic scale to image the arrangement of atoms. On the mesoscale, the focus is on examining the characteristics of individual dislocations and analyzing the distribution, arrangement, and density of dislocation in materials. Transmission Electron Microscopy (TEM) and Discrete Dislocation Dynamics (DDD) simulations are techniques for investigating these properties and simulating dislocation behaviour, respectively.

TEM is a microscopy technique that generates a highly-magnified image of a material specimen. This technique involves an electron beam passing through the specimen and several lenses. In strongly simplified terms, if the electron beam hits an atom, then it is deflected. As a result of the deflection, the intensity of the transmitted beam is reduced, and the intensity of the diffracted beam is increased. The dislocation can be seen as a dark line in such a bright-field image.

Above, it was already mentioned that the displaced atoms around a dislocation result in stresses, because the atoms are no longer in their preferred equilibrium position. Dislocations move in such stress fields which are mainly described by the governing equations of (linear) elasticity theory. DDD simulations employ mathematical lines (polygons or splines) to represent dislocations, which are moved based on elastic interactions and further "local rules".

The numerical schemes used in DDD simulations require to numerically discretize the mathematical line, e.g., by a number of straight line segments. The discretization steps are illustrated in Fig. 9. Further details can be found in [30].

The discretization process can be easiest described based on the example of a polygonal chain. There, the smooth mathematical line is approximated by a polygonal

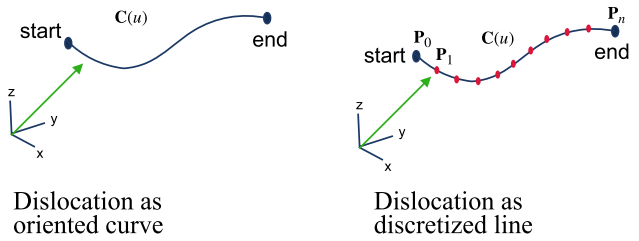

Fig. 9 The discretization of dislocation to a numerical representation. The oriented curve dislocation line shown in the left panel is discretized into several segments shown in the right panel

chain, **C**. **C** is a curve defined by a sequence of points ($\mathbf{P}_0$, $\mathbf{P}_1$,..., $\mathbf{P}_n$), and these points are called vertices. In addition, the curve consists of segments connecting consecutive pairs of vertices. In general, we can define the shape of a segment through the *shape function* which allows to have not only straight line segments but also spline curves of different order.

4 The dislocation ontology

The dislocation ontology (DISO) is developed using several well-known ontology development methodologies, such as [31]. The process is iterative, starting with an initial version and continuously revising and refining the evolving ontology. The development process is outlined in Fig. 10, which includes the main phases, their sub-tasks, and the roles involved.

4.1 Metadata

It is essential to provide a systematic and comprehensive description of the ontology, also known as ontology metadata, thus supporting its reusability and findability [32]. When ontology metadata is missing, several potential issues can occur. These include reduced accessibility for potential users, decreased reusability, and ontologies not being recognized as relevant for specific use cases. Accordingly, several DCMI Metadata Terms[9] have been added to the ontology, involving `terms:contributor`, `terms:created`, `terms:title`, and `vann:preferredNamespacePrefix`.

4.2 Reuse of existing models

When developing an ontology, one of the first steps is to utilize or reuse terms (i.e., classes or properties) from existing ontologies that describe the same domain or subject matter. Deciding which ontologies are appropriate for reuse is a challenging task for ontology engineers.

[9] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

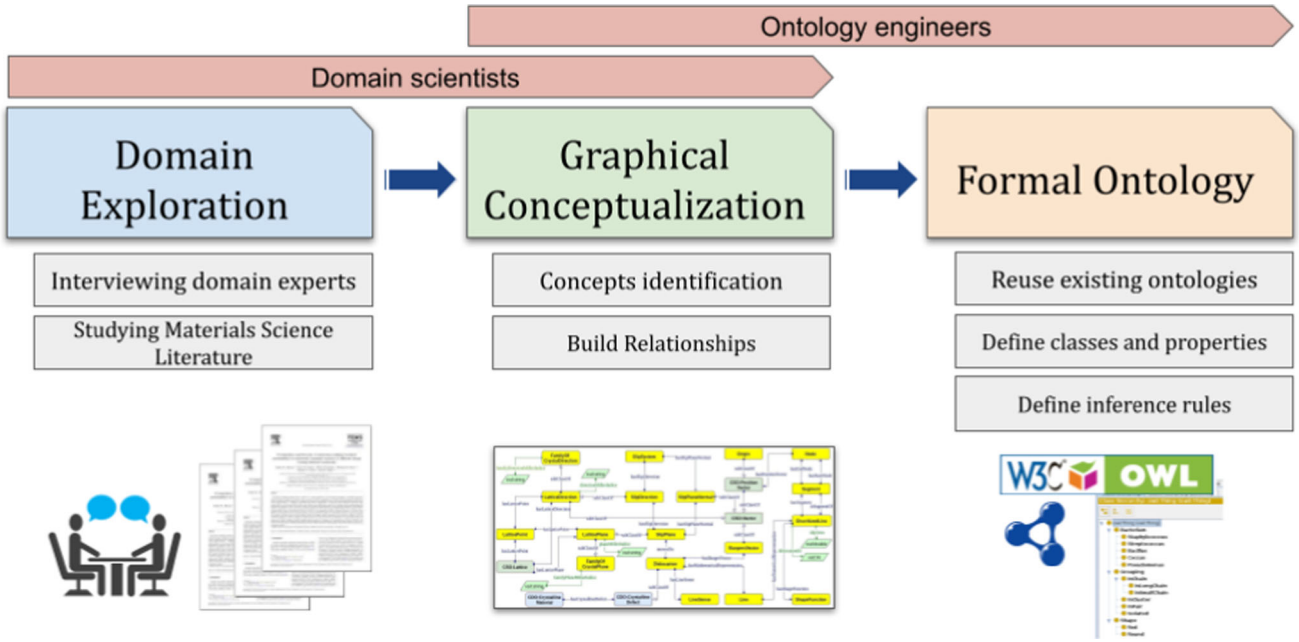

Fig. 10 The workflow of the dislocation ontology development which illustrates the main phases, subprocesses, and roles involved in the whole process

Ontology reuse involves several activities, including merging, extending, specializing, or adapting other ontologies. In DISO, we reuse concepts from two related ontologies in the MSE domain: the *Crystal Structure Ontology*[10] (CSO) and the *Crystalline Defect Ontology*[11] (CDO). CSO describes crystallographic data related to dislocations, while CDO links physical material entities to crystal structures and different defect types within a crystal, such as point defects, dislocations, and planar defects.

In CDO, the `EMMO:Crystal` class (from the EMMO[12] ontology) is reused to describe the physical entity of crystalline materials. The `CDO:CrystallineMaterial` class is defined as a subclass of `EMMO:Crystal` that is used to represent crystalline materials.

In CSO, several MDO [28] classes are reused to describe the crystal coordinate system, the motif in a crystal structure, point groups, and space groups. Furthermore, CSO defines the unit quantity of a property by reusing several classes from QUDT (Quantities, Units, Dimensions, and Data Types Ontologies) [33]. Overall, the semantic data value of the developed ontology increases as more ontologies are included, making the reuse of terms from other ontologies a worthwhile undertaking [34].

[10] https://purls.helmholtz-metadaten.de/disos/cso.

[11] https://purls.helmholtz-metadaten.de/disos/cdo.

[12] https://github.com/emmo-repo/domain-crystallography.

4.3 Classes

Our ontology classes are separated into two groups: 1) those imported from existing ontologies (as explained in Sect. 4.2) and 2) newly created classes that are not already defined in any existing ontologies.

Imported classes. DISO reuses several classes from CSO: `CSO:Lattice` represents the periodic arrangement of one or more atoms, and `CSO:Vector` represents quantities with both magnitude and direction. Additionally, DISO reuses classes from CDO, including `CDO:CrystallographicDefect`, which represents lattice irregularity or lattice defects.

Newly defined classes. For new classes, we focus on specific classes of crystalline materials and line defects, including 1) `Dislocation`, the focal class in DISO which represents a linear or one-dimensional defect that causes some atoms to be displaced, 2) `SlipPlane`, which models the lattice plane to which the dislocation is constrained to move in, 3) `SlipDirection`, which models the lattice direction where the slip occurs in the crystalline materials, 4) `LatticePlane`, which represents the lattice plane where it forms an infinitely stretched plane that cuts through the lattice points, 5) `LatticeDirection`, which models the direction inside the lattice that connects two lattice points, and 6) `DiscretizedLine`, which provides a numerical representation of the dislocation line as a mathematical line, such as an oriented curve, that is discretized into several segments (cf. Fig. 11).

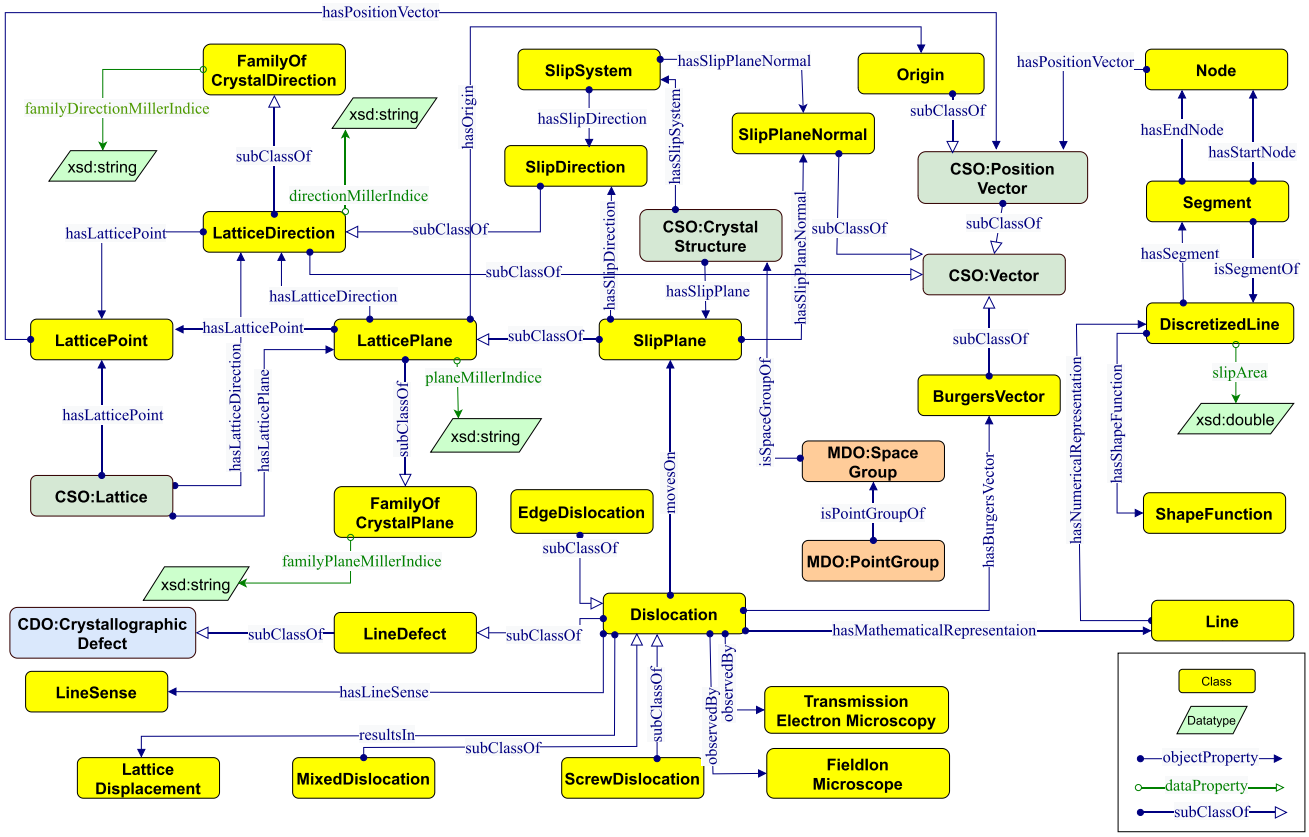

Fig. 11 Core concepts and interconnected relationships in DISO. Arrows with open arrowheads denote `rdfs:subClassOf` properties between classes. Regular arrows represent *rdfs:domain* and *rdfs:range* restrictions on properties and coloured boxes represent classes belonging to different ontologies, e.g., yellow boxes represent DISO's classes

4.4 Properties

Similarly, both data and object properties in DISO are divided into two categories: newly defined properties and reused ones.

Newly defined properties. Object properties constitute the relationship between various concepts in the ontology. For instance, the relationship between `TransmissionElectronMicroscopy` and `Dislocation` classes can be represented through the `observedBy` object property. Similarly, the `hasLineSense` object property represents the relationship between `Dislocation` and `LineSense`. Additionally, several data properties, including `directionMillerIndice` and `planeMillerIndice` are defined, which typically provide a relation to attaching an entity instance to some literal datatype value, such as a string or a date.

Reused properties. Several properties from the reused ontologies have been used, e.g., `cso:hasPositionVector`, `cdo:hasCrystallographicDefect`, `mdo:hasComposition`, and `emmo:hasProperty` from the CSO ontology, CDO ontology, MDO ontology, and EMMO ontology, respectively. Moreover, we reused several data properties from DCterms for adding ontology metadata (see Sect. 4.1). After defining new properties and identifying reused ones, the domain and range for each property using `rdfs:domain` and `rdfs:range` are defined, respectively. For instance, the domain of the data property `diso:planeMillerIndice` is `diso:LatticePlane` and the range is `xsd:string`. while the domain of the object property `diso:hasLatticePoint` is `diso:LatticePlane` and the range is `diso:LatticePoint`.

Restricting properties. In DISO, several classes use property restrictions, e.g., value constraints. For example, the `resultsIn` property which connects `Dislocation` and `LatticeDisplacement` is restricted by a value constraint of `owl:someValuesFrom` representing the fact that every dislocation individual results in *some* or at least one lattice displacement individual(s). The `hasLineSense` property which connects `Dislocation` and `LineSense` is restricted by a value constraint of `owl:allValuesFrom` representing that every dislocation individual can *only* have a line sense individual.

4.5 Reasoning

DISO's inference capability is increased through the use of several property characteristics, such as functional relations, transitivity, and the inverse property [35]. `hasMathematicalRepresentation` is a functional property because it means that a dislocation can be represented by exactly one mathematical line, i.e., it can not have any different mathematical representation than that. The transitive property can be demonstrated through the `hasRepresentation` relationship. This relationship refers to the connection between a dislocation and its representation. For instance, if a dislocation has a line representation, and this line has a discretized line representation, it can be inferred that the dislocation also has a discretized line representation.

To enable bidirectional navigation between two classes in the ontology, inverse properties are established for each corresponding property. For instance, the `isSegmentOf` property is the inverse property of `hasSegment`. This means that if a discretized line A *has a segment* B, then B *is a segment of* A.

5 Ontology alignment

Ontology alignment is the process of identifying relations between entities among different ontologies to establish connections between them [36]. These entities include classes, properties, and individuals. For successful ontology alignment, it is crucial to identify similarities between source and target ontologies. The analysis entails examining concepts that overlap but may have different names (i.e. synonyms) or types in the ontologies [37].

This section will cover the extension of DISO, which involves aligning two ontologies, namely EMMO and MDO. This alignment plays a crucial role in allowing DISO to annotate the DDD data and transform it into linked data while also facilitating knowledge graph generation.

5.1 Alignment with EMMO

EMMO is a continuous initiative aimed at establishing semantic standards that can be implemented at the highest level of abstraction. This makes it possible for all potential domain ontologies, especially in the MSE field, to be integrated and to work together seamlessly. EMMO offers two essential modules: a top-level module and a mid-level module. The former includes the fundamental axioms that constitute the philosophical foundation of the EMMO, while the latter consists of a set of perspectives to develop more specialized domain ontologies. These two ontologies serve as the basis for building further domain and applications ontologies, e.g., the application of EMMO in the domain of mechanical testing [38].

The starting point to align DISO with EMMO is by aligning with the Crystallography Domain Ontology,[13] a domain ontology based on EMMO and the CIF core dictionary.[14] As shown in Fig. 12, `CDO:CrystallographicDefect` subsumes `Dislocation`, while also being a subclass of `EMMO:Crystallographical` class. Similarly, `EMMO:CrystalStructure`, an equivalent class to `CSO:CrystalStructure`, is also a subclass of `EMMO:Crystallographical`. Overall, `EMMO:Crystallographical` is a class that ideally represents the physical concepts associated with crystalline materials.

As we mentioned in Sect. 3, on the mesoscale, a dislocation is represented by a mathematical line, which can be further idealized as a pixel or discretized line depending on the application (e.g., microscopy or simulation). To align the dislocation mathematical line concept with an EMMO class, `EMMO:MathematicalModel` subsumes `Line` and the discretized representation of the mathematical dislocation line, `DiscretizedLine`, is subsumed `EMMO:Numerical`.

5.2 Alignment with MDO

The MDO is a domain ontology that defines concepts and relations to cover the knowledge of materials design, especially in the ab-initio calculation. MDO consists of several modules, a *Core*, the *Provenance* module, and two domain-specific modules: *Structure* and *Calculation*.

To align DISO with the MDO, we reused several classes in the MDO Core module. The MDO Core module describes the structure or the virtual specimen of interest via `MDO:Structure` class. As shown in Fig. 13, we defined a `DislocationStructure` class as a subclass of `MDO:Structure`. This class describes a dislocation (micro)structure, which is a virtual specimen used by a DDD simulation to study the mechanical properties of a crystalline material. Furthermore, `DislocationStructure` as an idealized representation relates to a physical concept called `CDO:CrystallineMaterial`.

In the MDO Core module, an instance of the `MDO:Structure` class is used as a virtual specimen input or output for a simulation. Here, the simulation concept is represented as the `MDO:Calculation` class. We subsumed the `MDO:Calculation` class to define the

[13] https://github.com/emmo-repo/domain-crystallography.

[14] https://www.iucr.org/resources/cif/dictionaries/cif_core.

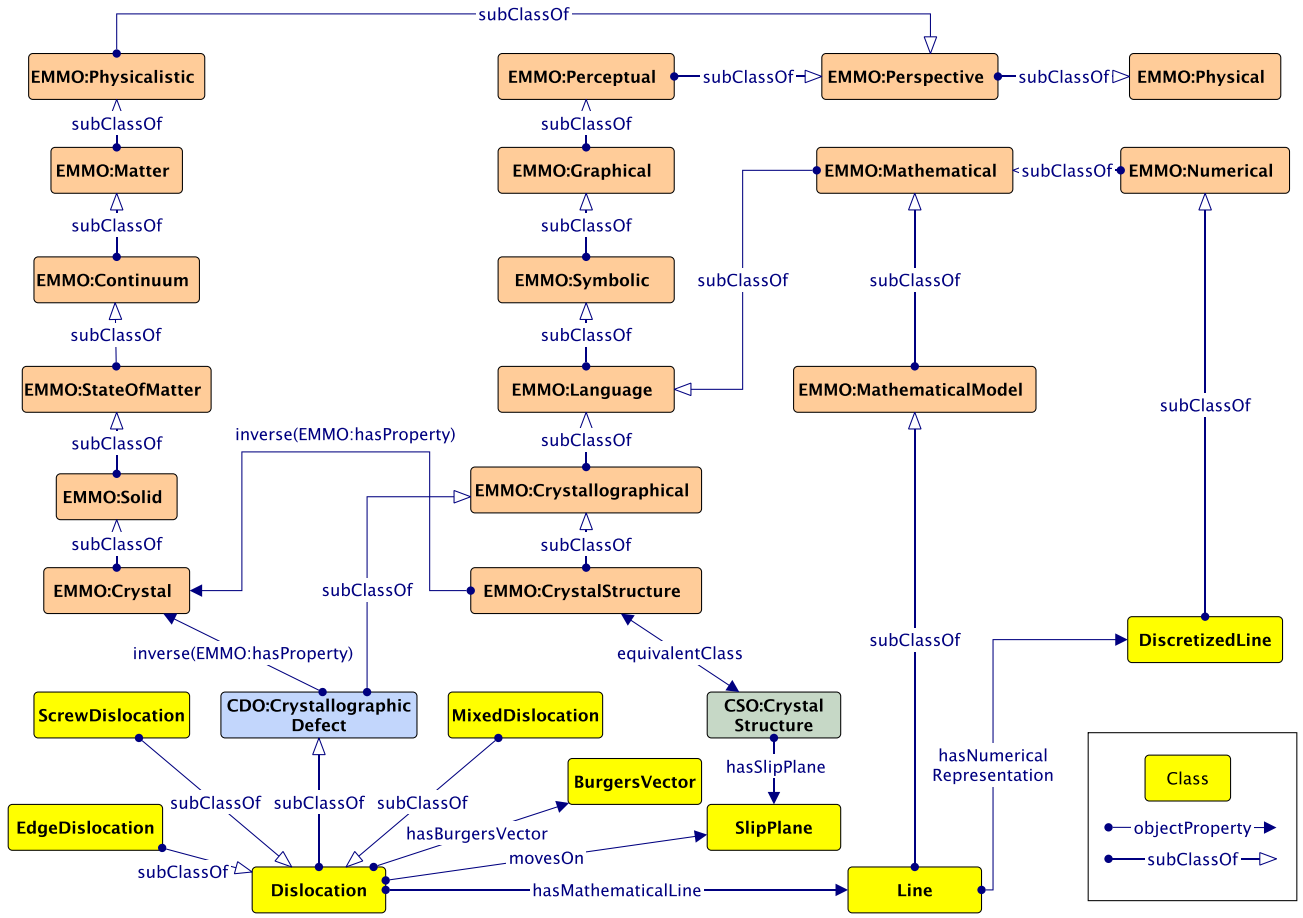

Fig. 12 DISO alignment with EMMO. The starting point to align DISO with the EMMO by importing the domain ontology crystallography developed by EMMO

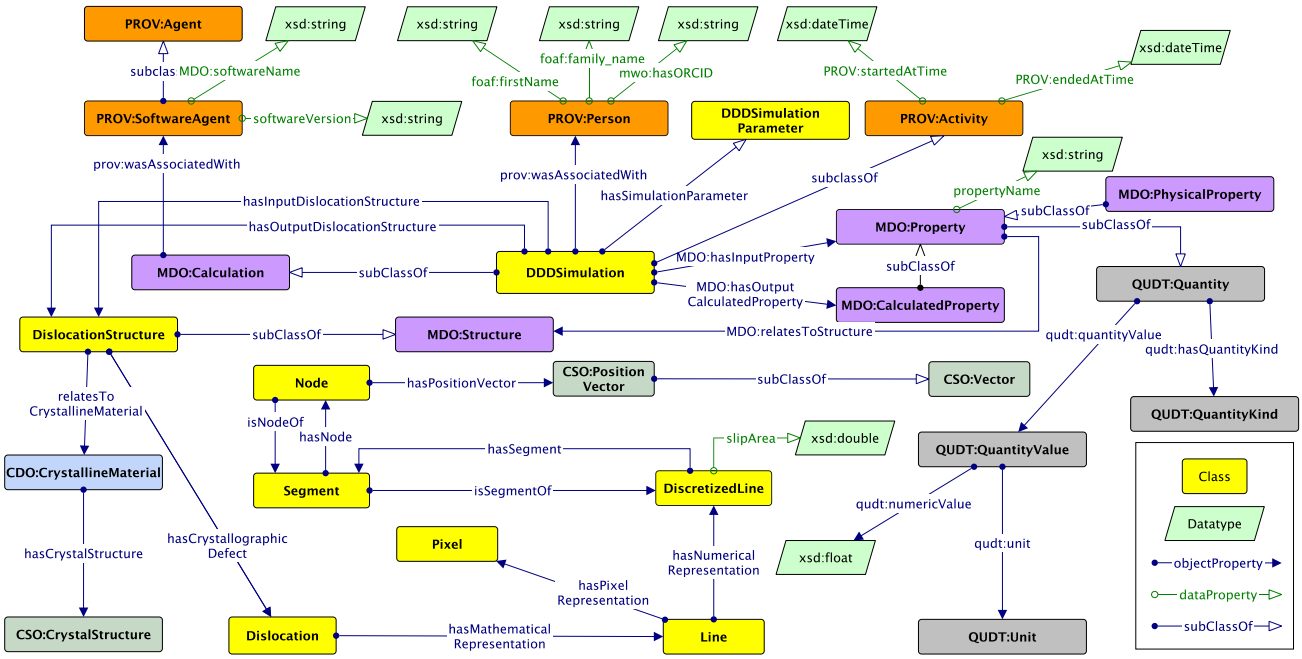

Fig. 13 DISO alignment with the MDO core and provenance module

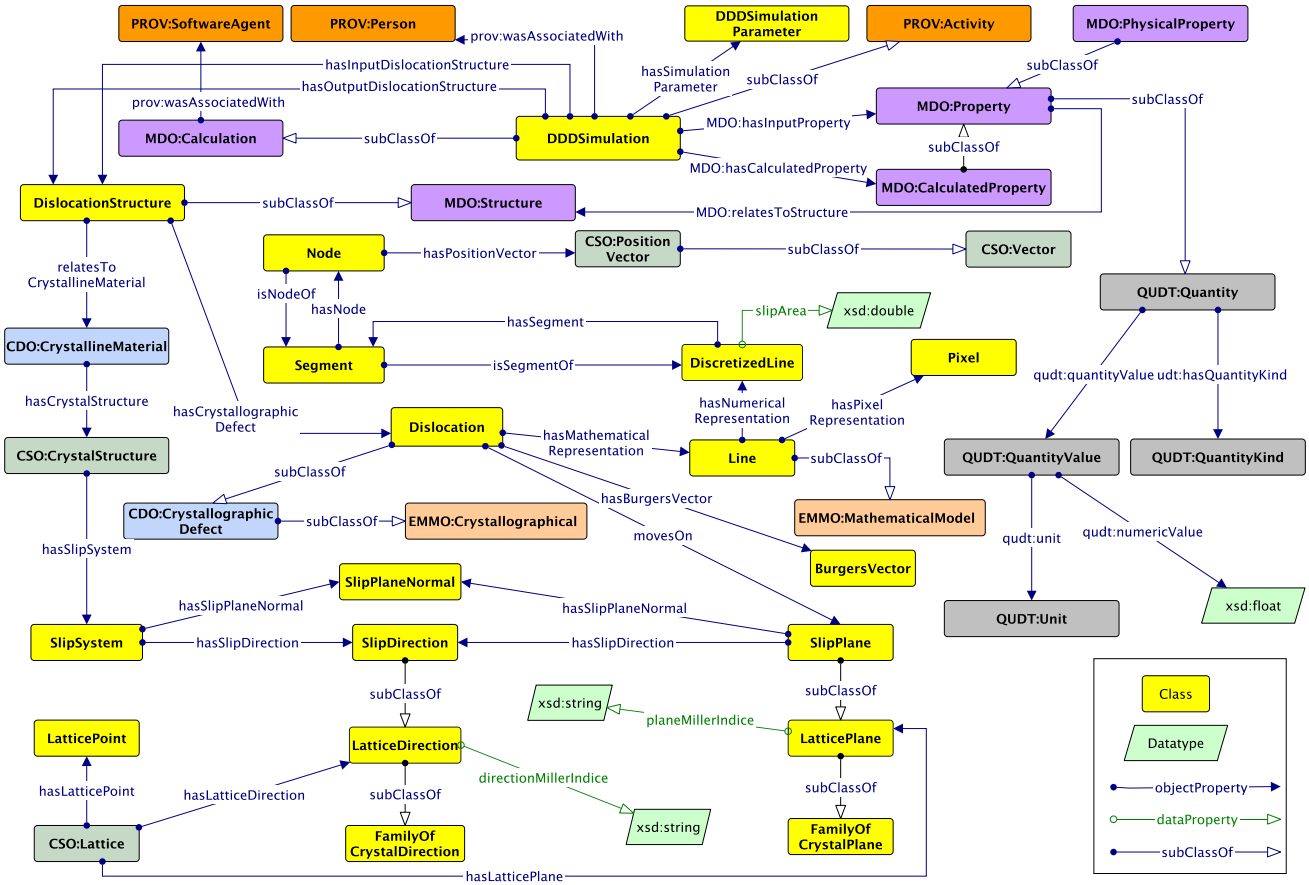

Fig. 14 The core concepts in DISO ontology after the alignment with MDO and EMMO. Arrows with open arrowheads denote `rdfs:subClassOf` properties between classes, while regular arrows represent the relationships between them. Classes that belong to the same ontology share the same color

`DDDSimulation` class, which is a class to describe the DDD simulation. Thus, the `DDDSimulation` can have `DislocationStructure` as an input or an output. Moreover, the `DDDSimulation` has an input and output relationship with `MDO:Property` to run a calculation. In addition, the `DDDSimulation` is related to the `DDDSimulationParameter`, a simulation parameter concept configuring the DDD simulation, e.g., the activation parameter for cross-slip, junction formation, and external load.

To preserve the provenance information of a DDD simulation, we reused several classes from the MDO Provenance module and the PROV ontology [39]. Running a DDD simulation requires specific software to solve materials science problems. It is quite helpful to store information about the software used and its version, as this can help scientists reuse data through post-processing methods specific to DDD software. In this regard, `DDDSimulation` has a relationship with `PROV:SoftwareAgent`, which has two data properties: `softwareVersion` and `MDO:softwareName`. Furthermore, to preserve the provenance information related to when a DDD simulation starts and ends, `PROV:Activity` subsumed the `DDDSimulation` and inherited two data properties: `PROV:startedAtTime` and `PROV:endedAtTime`. Apart from that, we reused `PROV:Person` to annotate the person running or responsible for the simulation. It has three data properties to define a person: `FOAF:firstName`, `FOAF:family_name`, and `MWO:hasORCID`. The latter is a data property that we reused from the MatWerk Ontology (MWO)[15].

To summarize, core concepts and interconnected relationships in DISO after the alignment can be seen in Fig. 14. The advantages of ontology alignment for DISO are promoting knowledge transfer from other ontologies when describing the DDD simulation. Furthermore, ontology alignment fosters interoperability between ontologies in MSE-related domains. The objective is to assist in building a knowledge graph for the dislocation domain.

[15] http://purls.helmholtz-metadaten.de/mwo/

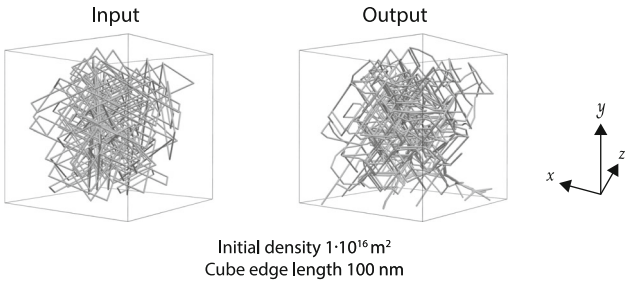

Fig. 15 Sample of dislocation microstructures used as input for simulation as well as yielded by the simulation as output

6 The dislocation knowledge graph

In the field of materials science, researchers utilize a numerical method known as "DDD simulation" to analyze the behaviour of dislocations within crystalline materials. This technique helps to identify the specific characteristics of each dislocation, as well as their interaction, arrangement, and collective behaviour within the material. The simulation observes the motion and interaction of many dislocations which ultimately creates the relationship between the microstructure, loading conditions, and the mechanical properties of a crystalline material. For simulations of dislocations, there are various software options available such as MoDELib [40], ParaDiS [41], and microMegas [42]. Every software is equipped with a specific set of metadata that effectively organizes the inputs and outputs of the simulation.

DISO was utilized in this specific scenario to accurately annotate the information collected from various DDD simulations. The ultimate goal is to generate a comprehensive dislocation knowledge graph (DisLocKG) using this data. The DDD data used in this work was generated through the MoDELib software and took different initial dislocation densities and specimen sizes into account. The cube-shaped Copper specimen, with an edge length of either 50 or 100 nanometers, was randomly filled with dipolar edge loops on all slip systems until the initial density of either $1 \cdot 10^{16}$ m^{-2} or $5 \cdot 10^{16}$ m^{-2} was reached. A sample of the generated cube-shaped Copper specimen can be seen on the left panel of the Fig. 15. During the simulation, the dislocation microstructure was allowed to relax without any external load, meaning that internal stress and image forces solely influenced the dislocation evolution. The simulation resulted in the relaxed dislocation microstructure shown on the right-hand side of Fig. 15. An important aspect for a materials scientist is that some simulations do not have cross-slip or junction formation. This has significant implications when it comes to analyzing simulation results. E.g., [43] investigated the influence of cross-slip on the evolution of dislocation structures and therefore could benefit from this information. Such relaxation calculations are also important for creating a realistic microstructure. For example, [44] investigated how the relaxed dislocation microstructure influences the plasticity in subsequent tensile test simulations. Furthermore, several authors [9, 45–47] conducted machine learning and data mining studies utilizing the dislocation relaxed microstructure to classify the structure and express the strain energy density of a dislocation microstructure, respectively. This explains why there is a strong need for a detailed and formal representation of such simulations, i.e., the class `DDDSimulation`.

For our example, we have collected a total of 25 data points, where each data point is one DDD simulation consisting of an initial and final microstructure. Each of those was annotated with DISO. Any data point gives information about the simulation details, such as the parameters used for the simulation, the initial dislocation microstructure used as input, and the resulting dislocation microstructure produced by the simulation. Additionally, each dislocation microstructure includes information about the crystal structure, Bravais lattice, dislocation, slip plane, Burgers vector, and numerical representation of dislocation. The results of the simulations are stored and parsed in the HDF5[16] format. Subsequently, we utilize our in-house Python scripts (using the RDFLib 6.0 [48] Python library) to create a knowledge graph called *DisLocKG* from this data using DISO as a reference ontology (cf. Fig.16). The DisLocKG is a semantic network that holds information about dislocations in crystalline materials. DisLocKG preserves the provenance information associated with the data, including the creator's data, software, and the corresponding software version employed in data generation. In total, we have generated $\sim$2.2M RDF triples, which are available through its persistent identifier.[17]

Publishing DDD data as linked data has several benefits [49], including 1) establishing links between dislocation-related datasets, enabling machines to understand and discover new information, 2) supporting semantic querying via the SPARQL query language, 3) supporting data enrichment, where machines can infer implicit knowledge that does not exist, and 4) promoting semantic validation of the data, ensuring consistency and accuracy.

We have listed some competency questions in Table 1 to illustrate the vast information available in DisLocKG and which questions it can answer. For instance, CQ1 can retrieve the history and origin information of DDD data generated by the MoDELib software, and CQ2 and CQ3 can retrieve information on the specimen geometry and the initial dislocation of each dislocation simulation. These

[16] https://www.hdfgroup.org/solutions/hdf5/.

[17] https://purls.helmholtz-metadaten.de/dislockg.

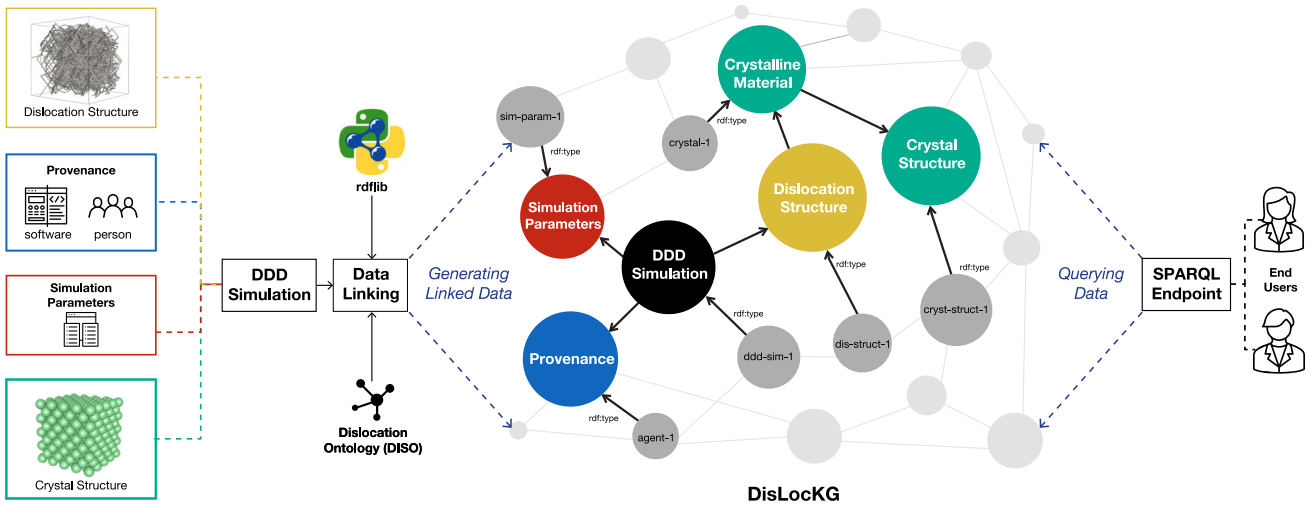

Fig. 16 DDD simulation data as linked data. Colored rectangles on the left depict data types in the DDD simulation: dislocation structure, provenance, simulation parameters, and crystal structure data. The data subsequently is linked using DISO as a reference ontology. The rdflib Python module supports data linking and generates the *DisLocKG*. Via the SPARQL Endpoint, end users can query the data to retrieve the information in the DisLocKG

Table 1 A sample of competency questions for DisLocKG

No.	Question
CQ1	Provide detailed information on the dislocation structures simulated using the MODELIB software, including the software version and creator associated with these simulations
CQ2	Which dislocation structures possess a specimen shape resembling a cube with an edge length greater than 30 nanometers?
CQ3	List all DDD simulations that have an initial density of dislocation $= 5e16$ m^{-2}
CQ4	List all DDD simulations that do not activate the cross slip formation and junction formation
CQ5	What are dislocation structures generated by the relaxation calculation? List also the initial density of a dislocation structure used for a relaxation calculation; the simulation parameters are: cross-slip activation, junction formation activation, and external load activation

CQs are important if one wants to query a dislocation simulation to be reused for the processing step if they need a specific density of a dislocation structure and information concerning the geometry. For instance, [43] used the information about dislocation density from a dislocation structure to study the influence of cross-slip; the geometry information is one of the components used in the so-called "D2C method" used for this type of analysis [50, 51] The D2C method was developed to convert discrete dislocation data to continuous field data. CQ4 retrieves the input parameters to run the simulation, while CQ5 queries all dislocation structures generated by the relaxation calculation. The SPARQL query corresponding to CQ3 is shown in Listing 1, and the complete set of the competency questions and the corresponding SPARQL queries can also be found in DISO's GitHub repository. Figure 17 visualizes the results of CQ3, which contains three individuals (shown as the red markers) of the DDD simulation class. Each of the DDD simulation individuals has a relationship with dislocation structure individuals. Moreover, the dislocation density data relates to the dislocation structure individual.

Listing 1 SPARQL query corresponding to CQ3.

```
PREFIX diso:<https://purls.helmholtz-metadaten.de/disos/diso#>
PREFIX qudt:<http://qudt.org/schema/qudt/>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX mdo_core:<https://w3id.org/mdo/core/>
SELECT ?ddd_simulation ?initial_density ?unit
WHERE{
?ddd_simulation        diso:hasInputDislocationStructure   ?input_dislocstructure.
?initial_density_qu    mdo_core:relatesToStructure         ?input_dislocstructure;
                       qudt:quantityValue                  ?density_qv.
?density_qv            qudt:numericalValue                 ?initial_density;
                       qudt:unit                           ?unit.
FILTER(?initial_density = "5e16"^^xsd:double)
}
```

7 Evaluation

Employing predefined metrics that evaluate an ontology's richness through criteria-based assessment is one way of evaluating its quality [34]. In this section, we evaluate the adapted version of DISO using the OntoQA [52] evaluation model.

This model can assess an ontology based on two dimensions: schema and instances. Here, we focus on the schema evaluation, which evaluates the quality of the ontology's design. We determine the effectiveness of the ontology and its ability to represent rich knowledge using the following metrics:

- *Relationship richness (RR)* shows the diversity of relations and placement of relations in the ontology (Eq. 2).

$$RR = \frac{|P|}{|SC| + |P|} \tag{2}$$

  where P is the number of relationships and SC is the number of sub-classes. The more relations an ontology owns, the richer it is (*is-a* relations are not considered).
- *Attribute richness (AR)* shows that the more attributes are defined, the more knowledge the ontology delivers (Eq. 3).

$$AR = \frac{|AT|}{|C|} \tag{3}$$

  where AT is the number of attributes for all classes and C is the number of classes.
- *Inheritance richness* (*IR*) describes the distribution of information across different levels of the ontology inheritance tree. IR indicates how knowledge is classified into different classes and subclasses in an ontology. (Eq. 4).

$$IR = \frac{|SC|}{|C|} \tag{4}$$

In Table 2, we compare the evaluation outcomes of DISO with MDO [28], CSO[18] and the previous version of DISO. DISO has the most significant value of *RR*, which implies that it has a greater relation diversity. Moreover, DISO has the highest *IR* value, representing a more comprehensive knowledge range than MDO, CSO, and the previous version of DISO. The *AR* value of DISO is lower than that of MDO and higher than CSO and the previous version of DISO. To conclude, DISO possesses the most extensive knowledge representation and diversity in terms of relationships, achieving the highest IR and RR, respectively. Moreover, the adapted version of DISO surpasses its predecessor in all evaluation metrics.

8 Conclusion and outlook

This paper showcases how semantic web technologies can transform unstructured DDD data into well-organized and structured data. Structuring data in the form of a knowledge graph can offer several advantages, such as being able to create links between different datasets related to dislocations, facilitating semantic querying through the SPARQL query language, enabling data enrichment by inferring implicit knowledge, and ensuring consistency and accuracy of the data through semantic validation. Furthermore, we extended the dislocation ontology by aligning it with commonly used materials science ontologies (i.e., EMMO and MDO core) to be able to model simulation data efficiently. Moreover, we presented a real-world use case that utilized DISO to construct a semantic network of DDD data (i.e., linked data) called DisLocKG, where individual entities are connected, enabling semantic query and supporting intelligent tasks. To support querying DisLocKG, the graph has been made publicly available, and steps to set up a SPARQL endpoint are described in its GitHub repository. The evaluation results indicate that the adapted version of DISO is the most comprehensive and

[18] https://purls.helmholtz-metadaten.de/disos/cso.

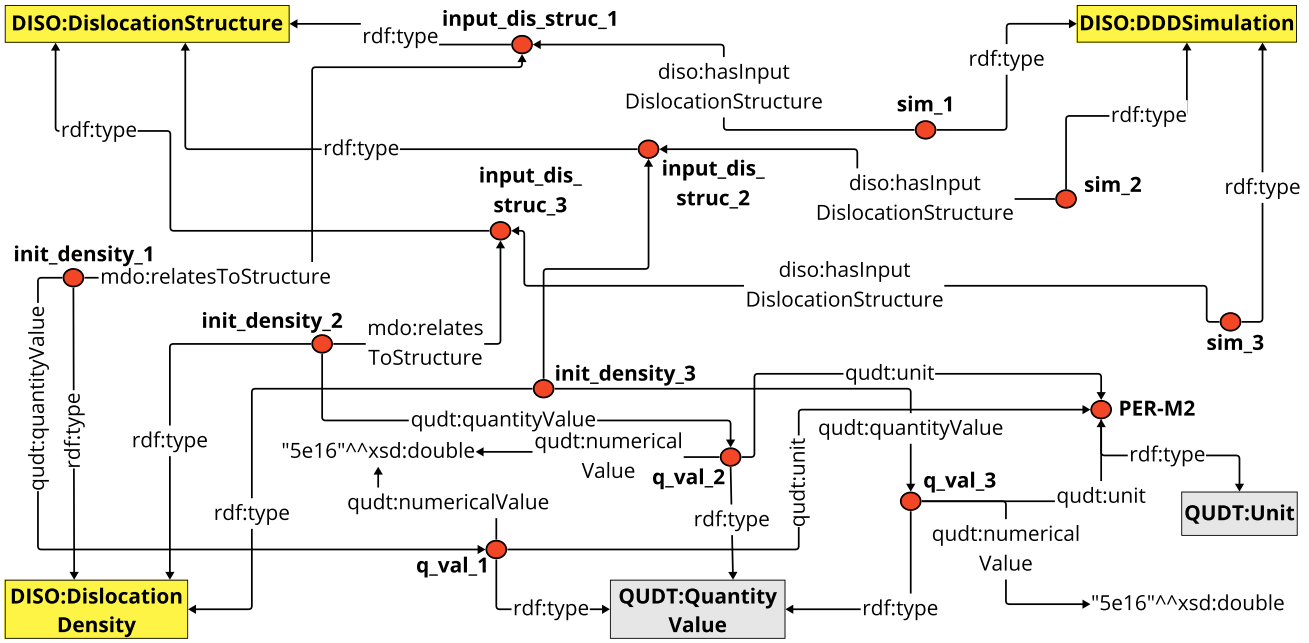

Fig. 17 A visual representation of CQ3 results. Colored boxes represent classes and the red dot represents an individual belonging to that class. Each individual is defined by a directed arrow having the `rdf:type` relationship to the respective class and connected to other individuals by object properties

Table 2 Evaluation of DISO compared to DISO v1.0, MDO and CSO using the OntoQA model. C is the number of classes, SC is the number of sub-classes, AT is the number of attributes, and P denotes the number of relationships

Ontology	C	SC	AT	P	RR	AR	IR
MDO	37	49	32	32	0.40	0.86	1.32
CSO	30	49	19	25	0.34	0.63	1.63
DISO v1.0	33	62	12	33	0.35	0.32	1.63
DISO v1.1	**70**	**116**	**47**	**80**	**0.41**	**0.67**	**1.66**

diverse knowledge representation among the state-of-the-art ontologies.

Materials data from different simulations often show similarities; such datasets typically consist of structure, simulation parameters, and provenance information. We identified that this study can be leveraged at least as a solid starting point in other domains (e.g., point defects and planar defects) and simulation on various scales (e.g., ab-initio calculation, molecular dynamics, and finite-element methods).

In the future, we plan to improve DISO by modelling the linear elasticity theory of dislocations and demonstrating other real-world use cases, including modelling results of other DDD simulation software, such as ParaDiS and MicroMegas, molecular dynamics data, and materials experimental data (e.g., Transmission Electron Microscopy Data). In addition, developing the DisLocKG Application Programming Interface (API) will also be a worthwhile undertaking. The idea is to develop several interactive features via an API, e.g., querying, data mining, visualizing, updating, and deleting data within the DisLocKG. DISO and DisLocKG will continue to be maintained and extended in the context of the Helmholtz Metadata Collaboration (HMC) and NFDI-MatWerk efforts to facilitate machine readability and efficiently handle research data.

Acknowledgements AI and SS acknowledge financial support from the European Research Council through the ERC Grant Agreement No. 759419 MuDiLingo ("A Multiscale Dislocation Language for Data-Driven Materials Science") and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure - NFDI 38/1 - project number 460247524. AI, SF, and SS acknowledge Helmholtz Metadata Collaboration (HMC) within the Hub Information at the Forschungszentrum Jülich (FZJ). We are grateful to Aytekin Demirci for the MoDELib parser that we used.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets used to generate the knowledge graph are available in https://purls.helmholtz-metadaten.de/dislockg.

Declarations

Conflict of interest The authors have no Conflict of interest to declare.

References

1. Taylor GI (1934) The mechanism of plastic deformation of crystals. part I.-Theoretical. Proc Royal Soc Lond Ser A Contain papers Math Phys Charact 145(855):362–387
2. Polanyi M (1934) Über eine Art Gitterstörung, die einen Kristall plastisch machen könnte. Z Phys 89(9–10):660–664
3. Callister WD, Rethwisch DG (2018) Materials science and engineering: an introduction, vol. 9. Wiley, New York
4. Murakumo T, Kobayashi T, Koizumi Y, Harada H (2004) Creep behaviour of ni-base single-crystal superalloys with various γ' volume fraction. Acta Mater 52(12):3737–3744
5. Govind K, Oliveros D, Dlouhy A, Legros M, Sandfeld S (2023) Deep learning of crystalline defects from tem images: a solution for the problem of" never enough training data". arXiv preprint arXiv:2307.06322
6. Bertin N, Zhou F (2023) Accelerating discrete dislocation dynamics simulations with graph neural networks. J Comput Phys 487:112180
7. Zhang C, Song H, Oliveros D, Fraczkiewicz A, Legros M, Sandfeld S (2022) Data-mining of in-situ tem experiments:on the dynamics of dislocations in cocrfemnni alloys. Acta Mater 241:118394
8. Yang Z, Papanikolaou S, Reid AC, Liao W-K, Choudhary AN, Campbell C, Agrawal A (2020) Learning to predict crystal plasticity at the nanoscale:deep residual networks and size effects in uniaxial compression discrete dislocation simulations. Sci Rep 10(1):8262
9. Song H, Gunkelmann N, Po G, Sandfeld S (2021) Data-mining of dislocation microstructures: concepts for coarse-graining of internal energies. Modell Simul Mater Sci Eng 29(3):035005
10. Salmenjoki H, Alava MJ, Laurson L (2018) Machine learning plastic deformation of crystals. Nat Commun 9(1):5307
11. Prakash A, Sandfeld S (2018) Chances and challenges in fusing data science with materials science. Pract Metallogr 55(8):493–514. https://doi.org/10.3139/147.110539
12. Adamovic N, Friis J, Goldbeck G, Hashibon A, Hermansson K, Hristova-Bogaerds D, Koopmans R, Wimmer E (2020). The emmc roadmap for materials modelling and digitalisation of the materials sciences. https://doi.org/10.5281/zenodo.4272033
13. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, Silva Santos LB, Bourne PE et al (2016) The fair guiding principles for scientific data management and stewardship. Scientific data 3(1):1–9
14. McBride B (2004) The resource description framework (rdf) and its vocabulary description language rdfs. Handbook on ontologies, pp 51–65
15. Bechhofer S, Van Harmelen F, Hendler J, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein LA et al. (2004) Owl web ontology language reference. W3C recommendation 10(2):1–53
16. Pérez J, Arenas M, Gutierrez C (2006) Semantics and complexity of sparql. In: The Semantic Web-ISWC 2006: 5th International semantic web conference, ISWC 2006, Athens, GA, USA, 5–9 Nov 2006. Proceedings 5, Springer, pp. 30–43
17. Ihsan AZ, Fathalla S, Sandfeld S (2023) Diso: A domain ontology for modeling dislocations in crystalline materials. In: Proceedings of the 38th ACM/SIGAPP symposium on applied computing. SAC '23, pp. 1746–1753. Association for computing ,machinery, New York, NY, USA . https://doi.org/10.1145/3555776.3578739
18. Ihsan AZ, Dessi D, Alam M, Sack H, Sandfeld S (2021) Steps towards a dislocation ontology for crystalline materials. In: 2nd International workshop on semantic digital twins, SeDiT 2021, vol. 2887 . CEUR-WS
19. Berrueta D, Fernández S, Frade I (2008) Cooking http content negotiation with vapour. In: Proceedings of 4th workshop on scripting for the semantic web (SFSW2008), vol. 72 . Citeseer
20. Loibl A, Manoharan T, Nagarajah A (2020) Procedure for the transfer of standards into machine-actionability. J Adv Mech Des Syst Manuf 14(2):0022–0022
21. Say A, Fathalla S, Vahdati S, Lehmann J, Auer S (2020) Semantic representation of physics research data. In: Proceedings of the 12th international joint conference on knowledge discovery, knowledge egineering and knowledge management
22. Hu S, Wang H, She C, Wang J (2011) Agont: ontology for agriculture internet of things. In: Computer and computing technologies in agriculture IV, In: 4th IFIP TC 12 conference, CCTA 2010, Nanchang, China, 22–25 October 2010, Selected Papers, Part I 4, Springer, pp 131–137
23. Say Z, Fathalla S, Vahdati S, Lehmann J, Auer S (2020) Ontology design for pharmaceutical research outcomes. In: International conference on theory and practice of digital libraries, Springer, pp 119–132
24. Mrdjenovich D, Horton MK, Montoya JH, Legaspi CM, Dwaraknath S, Tshitoyan V, Jain A, Persson KA (2020) Propnet: a knowledge graph for materials science. Matter 2(2):464–480
25. Zhao X, Greenberg J, McClellan S, Hu Y-J, Lopez S, Saikin SK, Hu X, An Y (2021) Knowledge graph-empowered materials discovery. In: 2021 IEEE international conference on big data (Big Data), IEEE, pp 4628–4632
26. Jain A, Ong SP, Hautier G, Chen W, Richards WD, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G, Persson KA (2013) The materials project: a materials genome approach to accelerating materials innovation. APL Mater 1(1):011002. https://doi.org/10.1063/1.4812323
27. Ashino T (2010) Materials ontology: an infrastructure for exchanging materials information and knowledge. Data Sci J 9:54–61
28. Li H, Armiento R, Lambrix P (2020) An ontology for the materials design domain. In: International semantic web conference, Springer, pp 212–227
29. McCusker JP, Keshan N, Rashid S, Deagen M, Brinson C, McGuinness DL (2020) Nanomine:a knowledge graph for nanocomposite materials science. In: International semantic web conference, Springer, pp 144–159
30. Ghoniem NM, Sun L (1999) Fast-sum method for the elastic field of three-dimensional dislocation ensembles. Phys Rev B 60(1):128
31. Suárez-Figueroa MC, Gómez-Pérez A, Fernandez-Lopez M (2015) The neon methodology framework:a scenario-based methodology for ontology development. Appl Ontol 10(2):107–145
32. Simperl E, Sarasua C, Ungrangsi R, Bürger T (2011) Ontology metadata for ontology reuse. Int J Metadata Semant Ontol 6(2):126–145. https://doi.org/10.1504/IJMSO.2011.046579

33. Hodgson R, Keller PJ, Hodges J, Spivak J (2014) Qudt-quantities, units, dimensions and data types ontologies. Vol. 156, USA, Available http://qudt. org March
34. Fathalla S, Vahdati S, Lange C, Auer S (2019) Seo: a scientific events data model. In: Ghidini C, Hartig O, Maleshkova M, Svátek V, Cruz I, Hogan A, Song J, Lefrançois M, Gandon F (eds) The semantic web - ISWC 2019. Springer, Cham, pp 79–95
35. Fathalla S, Lange C, Auer S (2023) An upper ontology for modern science branches and related entities. In: European Semantic Web Conference, Springer, pp 436–453
36. Ehrig M (2006) ontology alignment: bridging the semantic gap, vol. 4. Springer
37. Noy NF, Musen MA (2000) *et al.*: Algorithm and tool for automated ontology merging and alignment
38. Morgado JF, Ghedini E, Goldbeck G, Hashibon A, Schmitz GJ, Friis J, Baas A (2020) Mechanical testing ontology for digital-twins:a roadmap based on emmo. SeDiT 2020: Semantic Digital Twins 2020, 3
39. Lebo T, Sahoo S, McGuinness D, Belhajjame K, Cheney J, Corsar D, Garijo D, Soiland-Reyes S, Zednik S, Zhao J (2013) Prov-o: The prov ontology. w3c recommendation. World Wide Web Consortium
40. Po G, Ghoniem N (2014) A variational formulation of constrained dislocation dynamics coupled with heat and vacancy diffusion. J Mech Phys Solids 66:103–116. https://doi.org/10.1016/j.jmps.2014.01.012
41. Arsenlis A, Cai W, Tang M, Rhee M, Oppelstrup T, Hommes G, Pierce TG, Bulatov VV (2007) Enabling strain hardening simulations with dislocation dynamics. Modell Simul Mater Sci Eng 15(6):553
42. Devincre B, Madec R, Monnet G, Queyreau S, Gatti R, Kubin L (2011) Modeling crystal plasticity with dislocation dynamics simulations:the 'micromegas' code. Mech Nano objects 1:81–100
43. Demirci A, Steinberger D, Stricker M, Merkert N, Weygand D, Sandfeld S (2023) Statistical analysis of discrete dislocation dynamics simulations: initial structures, cross-slip and microstructure evolution. Modell Simul Mater Sci Eng 31:075003
44. Motz C, Weygand D, Senger J, Gumbsch P (2009) Initial dislocation structures in 3-d discrete dislocation dynamics and their influence on microscale plasticity. Acta Mater 57(6):1744–1754
45. Fan H, Wang Q, El-Awady JA, Raabe D, Zaiser M (2021) Strain rate dependency of dislocation plasticity. Nat Commun 12(1):1845
46. Steinberger D, Song H, Sandfeld S (2019) Machine learning-based classification of dislocation microstructures. Front Mater. https://doi.org/10.3389/fmats.2019.00141
47. Stricker M, Sudmanns M, Schulz K, Hochrainer T, Weygand D (2018) Dislocation multiplication in stage ii deformation of fcc multi-slip single crystals. J Mech Phys Solids 119:319–333
48. Boettiger C (2018) Rdflib: A high level wrapper around the redland package for common Rdf applications. https://doi.org/10.5281/zenodo.1098478
49. Fathalla S, Lange C, Auer S (2019) Eventskg: A 5-star dataset of top-ranked events in eight computer science communities. In: European semantic web conference, Springer, pp 427–442
50. Sandfeld S, Po G (2015) Microstructural comparison of the kinematics of discrete and continuum dislocations models. Modell Simul Mater Sci Eng 23(8):085003
51. Steinberger D, Gatti R, Sandfeld S (2016) A universal approach towards computational characterization of dislocation microstructure. JOM 68(8):2065–2072
52. Tartir S, Arpinar IB, Moore M, Sheth AP, Aleman-Meza B (2005) Ontoqa: Metric-based ontology quality analysis