Materials Graph Ontology

Sven P. Voigt[1] and Surya R. Kalidindi[1,2,3*]

1. School of Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0245, United States
2. George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405, United States
3. School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, United States

**corresponding author, surya.kalidindi@me.gatech.edu, GWW School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Abstract

To maximize the use of the materials data being generated by various researchers and organizations, it is necessary to store the data such that it is findable, accessible, interoperable, and reusable (FAIR). Although current materials data repositories and databases partly address the FAIR principles, they do not adequately capture the critical metadata that represents the contextual information (e.g., relationship between materials data and terms typically used by materials scientists such as process, structure, and property). The collection and organization of this metadata along with the original data would allow advanced queries that implicitly improve FAIR characteristics. Recent work has attempted to define this necessary metadata through the development of materials ontologies. This paper introduces a new materials graph and develops the associated materials graph ontology needed to address shortcomings of the current materials ontologies. This novel ontology can be combined with existing ontologies to standardize the inter-relationships between materials data elements and related materials concepts.

This paper demonstrates how the proposed materials graph ontology enables the conceptual description of a broad variety of materials data, improves the findability and usability of the different graph-connected material concepts and data, and formalizes a materials data ingest framework that is amenable for the extraction of process-structure-property relationships.

Keywords: Artificial Intelligence

1 Introduction

An astonishingly increasing amount of data is being generated in the materials science field due in part to the advent of novel high throughput experimental assays, increased computational power for simulations, and national initiatives such as the Materials Genome Initiative (MGI) [1]. Leveraging this data in materials development efforts requires the design and deployment of a suitable data infrastructure [2,3] that allows the addition of the critical metadata needed to interpret the data correctly. For example, a microstructure image sitting on a remote server has very limited utility without the proper context. The metadata should include information describing the image, the type of image, relationships between this microstructure image and the material it describes, what/how additional data is derived from the microstructure image, what other information is available about the material structure at different length scales, or how the microstructure may change if the material is subjected to a new processing step. In this paper, all of the metadata about a materials data point and how it may be connected to other related data points will be collectively captured using a suitably defined materials ontology [4,5]. Finding connected materials data not only improves FAIR characteristics, but is also an essential step

towards extracting process-structure-property (PSP) surrogate models needed to drive materials innovation [6,7]. Identifying related datasets or finding existing models that could be applied to a new dataset would dramatically improve the re-use of existing materials datasets, and is likely to produce significant cost and time savings in materials innovation efforts. Using a properly designed materials ontology can address this critical need.

Databases have been employed by the materials research community to realize some of the FAIR goals [8–11]. Some databases, such as the Materials Project [12], Automatic Flow for Materials Discovery [13], Materials Data Facility [9], and database technologies, such as Automated Interactive Infrastructure and Database for Computational Science [14], add indexable materials data directly to the database to allow users to quickly search by commonly used terms by materials specialists (e.g., attributes related to structure, property, and process). As a specific example, materialsproject.org [12] allows a researcher to search by elements, chemical formula, id, crystallographic information file (CIF), or an mpquery entry that lets users search over arbitrary database keys. All of the related terms used in the database (i.e., database keys) have precise meanings, as established by the database schema. The only exception is mpquery, which allows any arbitrary database query string. The database schema, therefore, plays an important role in adding the contextual information to the data, improving its FAIR characteristics. Ontologies go beyond database schemas to enrich the metadata by including many features of language, such as subjects, predicates, objects, synonyms, etc., to describe the contextual information with desired precision. Ontologies are particularly known for their ability to describe complex heterogeneous information and integrate disparate data sources [15]. As such, ontologies are expected to play an important role in improving the

FAIR characteristics of materials data, by connecting the typically dispersed data among the multiple databases and repositories.

Materials data is also stored and shared through data repositories. For example, the NIST materials data repository [9] allows searching for records by community collections, author, subject, title, date issued, and whether the record has associated files or not. However, materials specific information contained in repositories' files is inaccessible and cannot be searched. Further, these repositories do not enforce a schema; the files stored in repositories can include heterogeneous data in any format, limiting their interpretability and utility to anyone other than the original creator. Some data standards such as CIF [16], the chemical markup language [17], and Universal Spectroscopy and Imaging Data [18] aim to standardize file formats and address this problem. However, this standardized metadata has not yet been implemented as part of a schema or a materials ontology. Additionally, the materials repositories often save the materials data in zip or hdf5 format, which capture heterogeneous files in a hierarchical file structure. However, file structures can at best imply relationships, but these could easily be misinterpreted or could be sufficiently ambiguous hindering the re-usability of the data. A materials ontology using clearly defined terminology that can be accessed by a variety of software and software platforms would precisely capture the connections between materials data, and dramatically improves the FAIR characteristics of data in comparison to the typical schemas found in current materials databases and repositories. Furthermore, the emergent tools in AI reasoning and web ontologies can be leveraged in developing and deploying such a materials ontology.

2 Knowledge Representation and Ontology

Knowledge representation (KR) is a subfield of artificial intelligence concerned with the digital representation of human understanding. KR is a model of the real world and captures as much of reality as is needed for reasoning and inference [19]. Reasoning differentiates knowledge and data, where reasoning allows inferring new connections between data points based on rules [20], as depicted in Figure 1. An ontology provides a vocabulary to define concepts and relationships [21], which can be used for KR. Ideally, an ontology should be designed to be capable of defining any concepts and rules needed to describe the real world; particular forms of ontology should be designed to represent the features most relevant or important to the specific field of implementation.

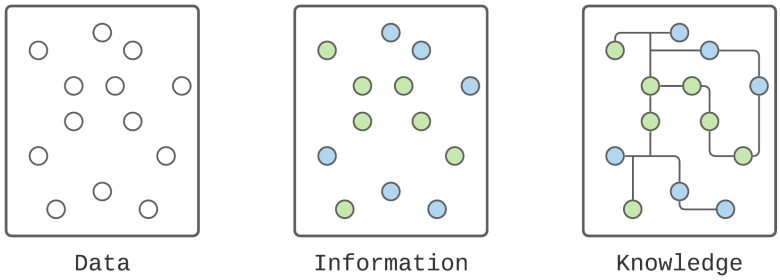

Figure 1: KR deals with adding contextual information to data that allows us to reason about the data, and understand the patterns and connections present in the data.

Ontologies are formally prescribed using ontology modeling languages, which take inspiration from *logics* (i.e., systems of logic) in which mathematical axioms are used to define rules and inference is computed within an established framework. First-order logics allow axioms to be declared with prepositions, which allow making statements about any

number of variables. Second-order logics allow prepositions to make statements about other prepositions, but do not see use in ontology modeling languages. Instead, all logical statements must be mapped to first-order logic, if possible. Among the various ontology modeling languages, description logics (DL) [22] and the web ontology language (OWL) [21] restrict ontologies to decidable first-order logic statements, where decidable means that there exists a method to prove a new statement from existing statements (i.e., infer additional information). Decidable first-order logics constitute only a subset of all first-order logics. DL constructs ontologies using *concepts* (usually entities such as a Material or a Process or a Structure), *individuals* (specific instantiations such as Ti-6Al-4V alloy), and *roles* (relationships) [22]. Analogous to these, OWL uses *classes*, *individuals*, and *properties*, respectively. Both of these ontological modeling techniques envision ontologies as graphs, where a graph is comprised of nodes and relationships [23]. The nodes reflect physical entities (i.e., nouns, objects, concepts, classes), while the relationships capture the contextual actions (i.e., verbs, predicates, roles, properties). A graph is easily visualized as shapes and arrows, where shapes indicate the nodes and arrows indicate the direction of a relationship. Additionally, data that follows an OWL ontology can be represented in the resource description framework (RDF) model, where all data is represented as a <subject><predicate><object> triples [24]. Each of these would be associated with defined classes, individuals, or properties in the OWL ontology. Further, the RDF is a directed, labeled graph. Here, the subject and objects are nodes and the predicate is the relationship that points from the subject to the object.

Despite being able to declare rules and reason about them, decidable first-order logics, and therefore the ontology modeling languages OWL and DL, have technical limitations that limit the type of rules that may be defined. In fact, many common first-order logic

statements are in fact undecidable and may not be represented in the current ontological modeling languages [25]. However, there are many other graph analysis techniques, the latest of which being deep learning methods [26], which can address the problem of inferring new relationships in graphs. However, these statistical methods need to specify the probability of a relationship, which is not possible with the current RDF syntax, although RDF* [27] is under development and seeks to solve this issue.

Despite the drawbacks in defining rules mentioned above, OWL is still an extremely powerful ontological modeling language. OWL has many associated tools for ontology analysis and inherently allows the integration of any number of other ontologies, which are specified globally through the web using internationalized resource identifiers (IRI). The ability to extend existing ontologies allows new ontologies to draw on already defined classes and properties. This work plans to define a unifying ontology that can merge existing materials ontologies and make use of the extensive quantity of already defined classes and properties in the materials science domain.

3 Materials Graph Ontology

In materials science, ontologies have been developed for several specific applications such as general materials knowledge from wikidata [28], functionally graded materials [29], and additively manufactured materials [30]. Another ontology integrates two computational materials databases [15]. These ontologies are merged on the identical structure and spacegroup classes, which creates explicit links between the types of properties that can be found in each database. For example, one database schema has x-ray diffraction data

associated with structures and another schema has associated prototype structures, which are now linked through their relationship to the structure class in the newly developed ontology. Another materials ontology has been developed for materials synthesis [31], which defines a process as a sequence of a precursor material node followed by a variable number of process operation nodes. This materials ontology is unique because it allows process operations to link to themselves, essentially creating a long chain of process steps that can define an arbitrarily complex processing history. However, despite these developments, there is very little consensus on what a material actually is, and “material” may not even be defined in the specific ontology. Callister [32] defines a material as having four components: Processing, Structure, Properties, and Performance. This implies that the characteristics of a material define it. In an ontology we can then define a material as anything having relationships to the four mentioned classes. Further, materials science is often concerned with how materials are related to each other. Therefore, a materials ontology should be able to answer how two materials are related, and if they are similar or dissimilar. This information can be captured by a materials graph that can link any two materials together in a connected network. The materials graph ontology could also be used to integrate other existing materials ontologies by merging them on the material class.

The proposed materials graph is designed to relate materials to other materials based on the concept of processing history, as discussed later. The ontology specifies the types of nodes (i.e., classes) and relationships (i.e., properties) that are permitted in constructing the materials graph to capture the desired contextual information (i.e., metadata). In other words, one can establish the desired graph by using any of the allowed nodes and relationships in any sequence, repeatedly as needed. Also, it is noted that we will be using

the graph terminology node for OWL classes and relationship for OWL properties, as the word property has a different meaning in the materials domain.

Figure 2 depicts the nodes and relationships of the proposed materials graph ontology. This materials graph ontology is also provided in the OWL syntax in the supplement to this paper, where WebProtégé [33] was used to generate the ontology. The four types of nodes identified in Figure 2 as Material, Process, Data, and Tool and the six distinct relationships identified as *next_in_process*, *composed_of*, *describes*, *input_to*, *yields*, and *used_in* are proposed to be the minimal set needed to build a materials graph for any given materials dataset or database. These can be used to broadly connect all materials concepts and data. Additional node types and relationships can also be added from other ontologies as needed.

The Material node represents a distinct material with associated properties, material structures at a hierarchy of length scales, process history (sequence of Material and Process nodes), and constituents (also represented as Material nodes). As many additional nodes as needed can be added to the Material node to produce the desired material graph. We also define the rule that any two Material nodes that are related to identical properties, material structures, and process history indeed represent identical materials. Furthermore, any two Material nodes that are related to some identical or similar properties, material structures, and process history, with others undefined, have a probability of being the same material. The graph model does not require that identical materials be merged into a single node; they may exist as different nodes in a graph database.

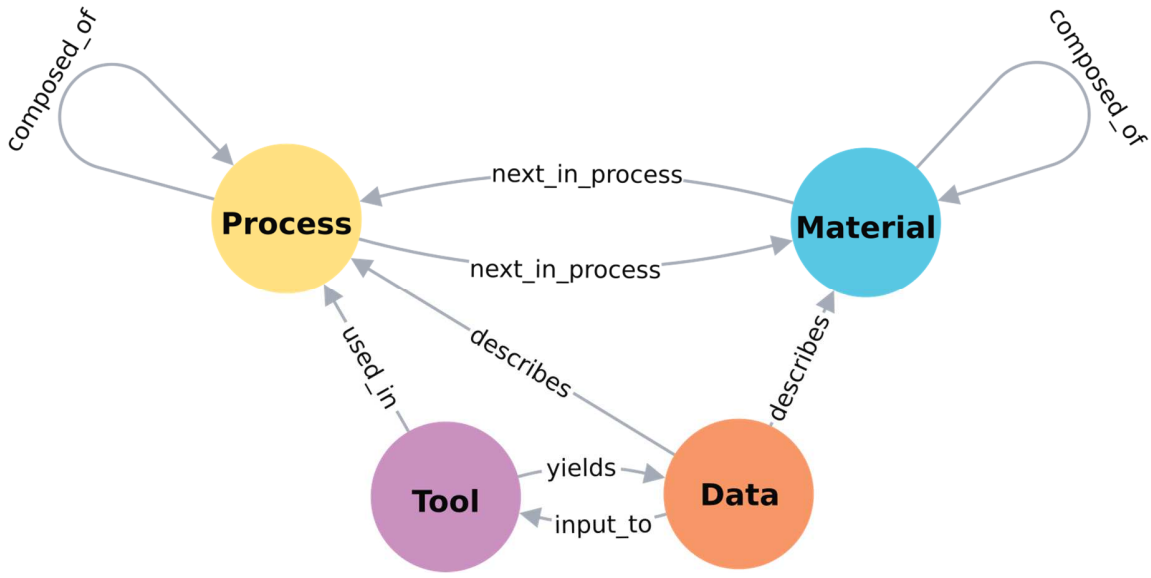

Figure 2: Depiction of the foundational elements for the proposed materials graph ontology.

The Process node together with the *next_in_process* relationship is used to indicate the transformation of a material by a processing step. Additional information about the processing step can be captured using the Tool and Data nodes, along with the allowed relationships specified in Figure 2. The relationship between a Material node and a Process node is defined exclusively by *next_in_process* which stipulates that a material can only change through a processing step and a process always acts on materials. A sequence of alternating and connected Material and Process nodes can then be used to capture a complex manufacturing process. The Material and Process nodes are allowed to relate to themselves, i.e., only these nodes are allowed to be connected to other nodes of the same kind. This is because one can visualize these entities as complex physical systems made of other entities of their type. A Material may have constituents defined at different length scales (e.g., phases, precipitates), which can be treated as distinct materials by themselves (i.e., they exhibit their own unique properties and structure). As

another example, a heat treatment Process may have substep Processes such as ramp, soak, and quench, as shown in Figure 3(a).

The Data node is used to capture all of the associated data and metadata that describe a material in terms of its structure, properties, or performance. For the Material and Process nodes, this is accomplished using the relationship *describes*. Data node can also store the input and output data from a Tool node using the relationships *input_to* and *yields*, respectively. Note that the relations *input_to* and *yields* are special as they can be linked. For example, if Tool acts as a function, it should map the exact input Data to the yielded Data. The Data node can be broadly used to store the results of experiments, simulations, or curated values from literature or domain experts.

The Tool node is designed to represent the different machines used by the materials experts. These may include a broad range of equipment such as processing equipment, characterization equipment, and simulation/analysis software. Multiple Tools nodes may be connected to a single Process node to capture the desired metadata on how a specific materials processing step was achieved. For example, Figure 3(b) shows a small materials graph that captures details of a heat treatment, where the temperature control information is *input_to* the furnace and the furnace *yields* temperature history. As another example, the use of multiple tools to measure a material property can be captured using multiple Tool nodes and a single Data node and a single Material node. This is illustrated in Figure 3(c) for the measurement of the thermal expansion coefficient.

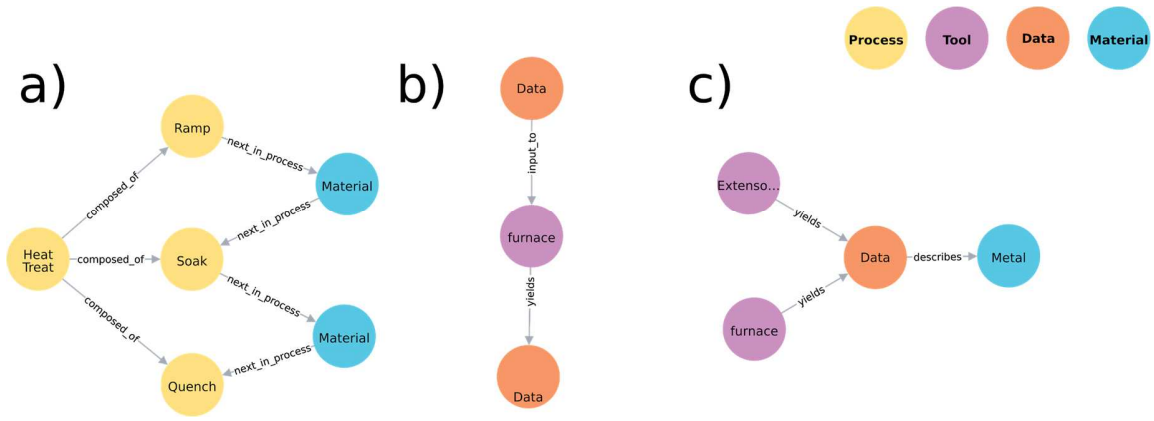

Figure 3: Example Materials Graphs showing the (a) hierarchy of Processes, (b) conversion of Data by Tools, and (c) generation of Data from multiple Tool sources.

4 Case Study

This case study examines an available dataset (containing both process history and material properties) taken directly from Ref. [34]. In this example, there are many discrete materials listed as rows in tables, without any notion of connection between those materials. One of the main benefits of the materials graph ontology is that these materials can be inter-related in a materials graph. Additionally, the implementation of the materials graph ontology would also allow us to store all of the original data in a connected manner together with the contextual information.

Figure 4 describes the connected dataset generated by Ref. [34] as several disjoint materials graphs, using the ontology proposed in this work, where each material graph corresponds to a row in a table provided in Ref. [34]. The graphs start with a starting material and track their transformations through the different imposed process histories, while capturing the details on the tools employed and the data collected at different stages.

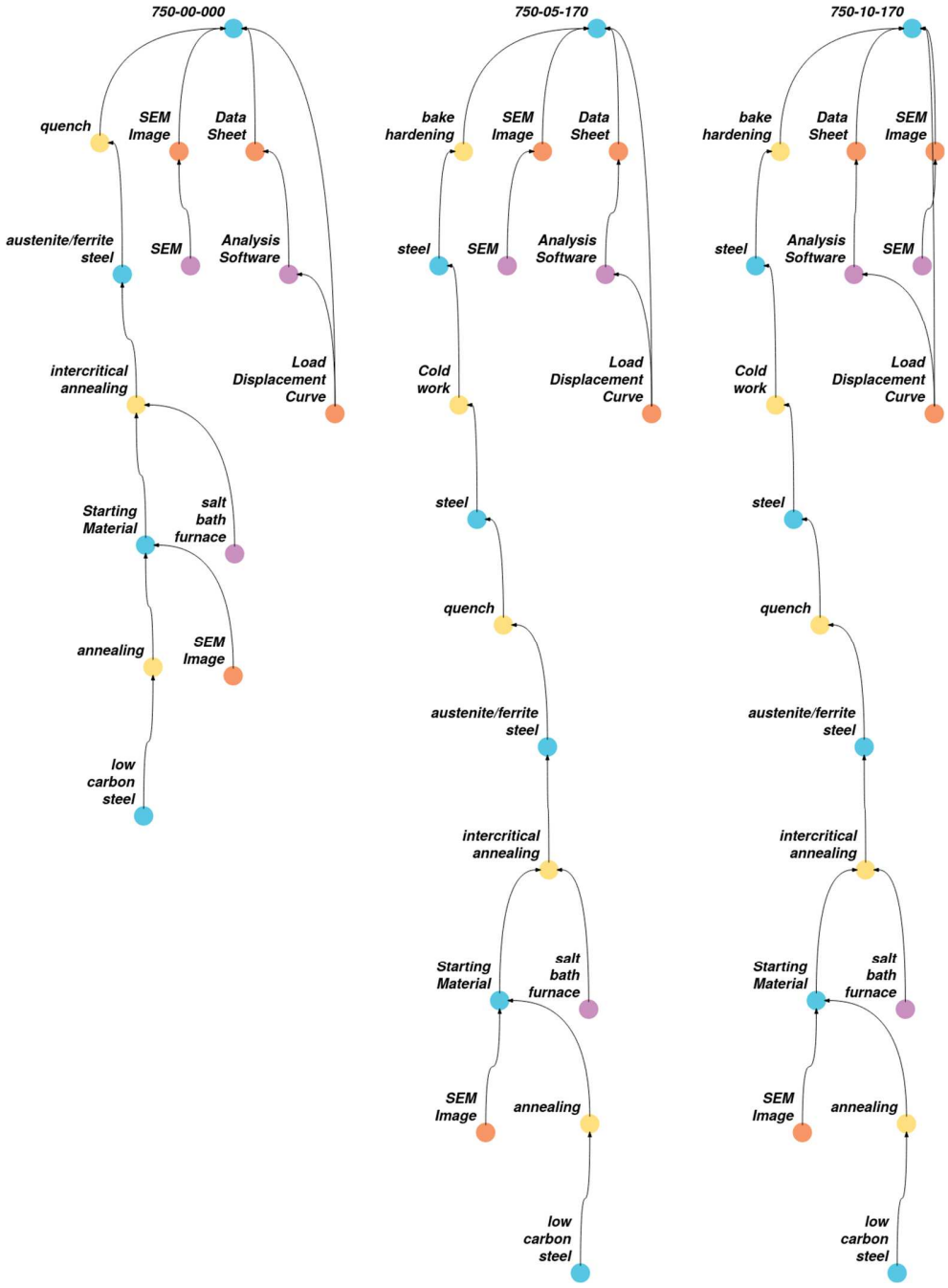

Figure 4: The material graphs for samples 750-00-000, 750-05-170, and 750-10-170 from Ref. [34].

After identifying node equivalencies, the materials graphs in Figure 4 are unified into the single graph shown in Figure 5. The unified graph recognizes that there is a common starting material that was subsequently processed in overlapping process histories. For example, 780-00-000 is an ancestor common to both 780-05-170 and 780-10-170, and this contextual information is captured in an unambiguous manner in the proposed materials knowledge graph.

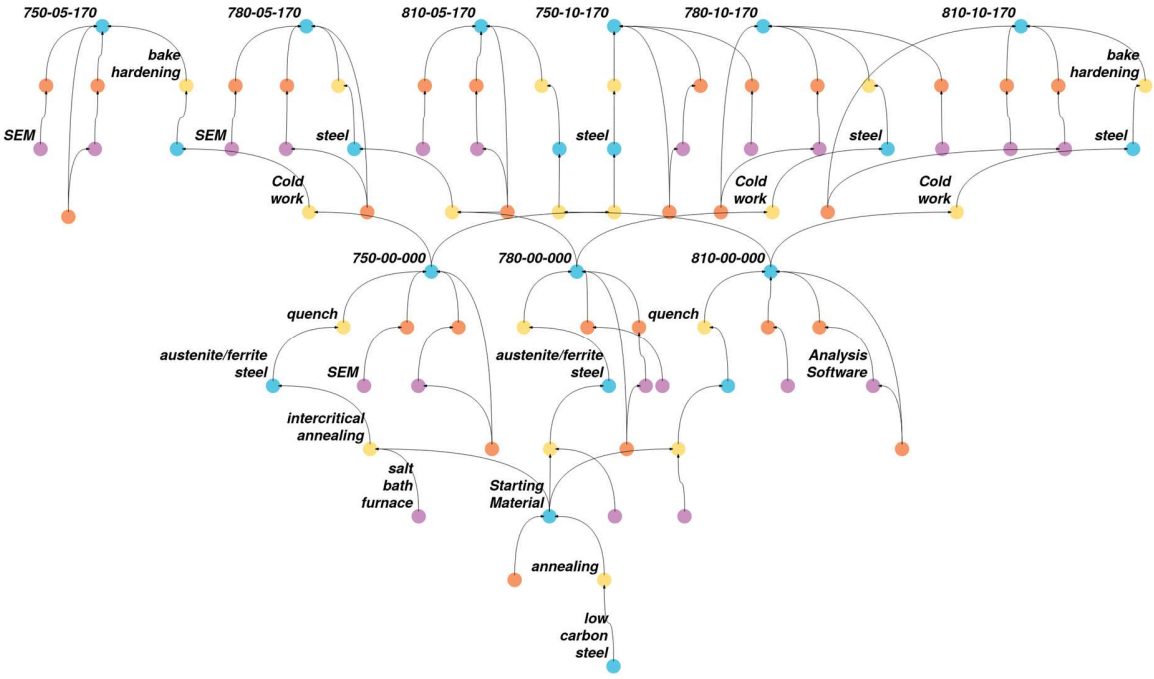

Figure 5: Materials knowledge graph for all the samples processed in Ref. [34].

5 Discussion

Ontologies provide a powerful method for relating information, which would allow us to integrate heterogeneous datasets and make that data more useful by identifying data inter-relationships. However, there are still significant challenges to making the ontologies truly useful. For one, as observed in the case study presented earlier, a direct conversion from existing data to the graphical format produces the same disconnected data that we had before. Ontology rules are essential to generating truly connected data. Defining such rules is quite challenging, as it requires the conversion of higher level statements to first-order logic. For example, the rule that a material is the same as another material if the process history, properties, and constituent materials are the same requires multiple comparisons over a complex network of nodes and edges. Although this comparison is relatively straight-forward to define in graph query languages, defining such a rule in an ontology is quite complex. However, it is important that such definitions are represented in a formal ontology such that they can be automatically inferred by inference tools and defining them formally will be the subject of future work.

Secondly, an advantage of ontologies is that they can be combined by referencing the other ontologies via an Internationalized Resource Identifier (IRI). However, this requires that the other ontologies' IRIs be resolvable either locally or, preferably, through the internet. The current materials ontologies do not provide any IRI through which they could be referenced. There are online tools to help facilitate the sharing of ontologies, such as WebProtégé [33]. However, it is only possible to access ontologies in WebProtégé by making an account and by having the creator make the ontology public.

6 Conclusions

This work proposed a materials graph ontology that is capable of connecting disparate materials data with related materials concepts typically employed by domain experts. It describes the concept of a material in terms of its relationships to other concepts including process history, structure and property data, and other materials. This will enhance the ability to relate materials data to the actual concept of a material, improve the FAIR characteristics of the data, integrate heterogeneous datasets, and make it easier to define a class of materials in terms of related concepts. In materials science, large quantities of data are being produced by simulations, high throughput testing, and other data generation efforts. However, this data lacks the contextual information to make it re-usable. Being able to add contextual information to the data using the materials graph ontology will not only improve the FAIR characteristics of the data, but give it the potential to be combined with other data, increasing its value. The materials graph ontology presented in this work allows integration with the other existing ontologies.

Acknowledgements

The authors acknowledge support for this work from NIST 70NANB18H039 (Program Manager: Dr. James Warren).

References

[1] J.P. Holdren et al., National Science and Technology Council (2011).
[2] D.L. McDowell et al., MRS Bulletin 41 (2016) 326–337.
[3] S.R. Kalidindi et al., Integrating Materials and Manufacturing Innovation 8 (2019) 441–454.
[4] B. Smith et al., Formal Ontology in Information Systems (2001) 7.

[5] H. Li et al., The Semantic Web – ISWC 2020 12507 (2020) 212–227.
[6] S.R. Kalidindi, MRS Communications 9 (2019) 518–531.
[7] S. Kalidindi, Butterworth-Heinemann (2015).
[8] K. Alberi et al., Journal of Physics D: Applied Physics 52 (2019) 013001.
[9] B. Blaiszik et al., JOM 68 (2016) 2045–2052.
[10] J. Hill et al., Computational Materials System Design (2018) 193–225.
[11] S. Ramakrishna et al., Journal of Intelligent Manufacturing 30 (2019) 2307–2326.
[12] D. Gunter et al., 2012 SC Companion: High Performance Computing, Networking Storage and Analysis (2012) 1244–1251.
[13] S. Curtarolo et al., Computational Materials Science 58 (2012) 218–226.
[14] G. Pizzi et al., Computational Materials Science 111 (2016) 218–230.
[15] S. Zhao et al., AIP Advances 7 (2017) 105325.
[16] S.R. Hall et al., Acta Crystallographica Section A Foundations of Crystallography 47 (1991) 655–685.
[17] P. Murray-Rust et al., Journal of Chemical Information and Computer Sciences 39 (1999) 928–942.
[18] S. Somnath et al., (2019) arXiv:1903.09515.
[19] R. Davis et al., AI Magazine 14 (1993) 17.
[20] L. Ehrlinger et al., SEMANTICS 48 (2016) 4.
[21] S. Bechhofer et al., W3C (2004) www.w3.org/TR/owl-ref/.
[22] M. Krötzsch et al., (2013) arXiv:1201.4089.
[23] M. Needham et al., O'Reilly Media (2019).
[24] G. Klyne et al., eds., W3C (2014) www.w3.org/TR/rdf11-concepts/.
[25] M. Krötzsch, Description Logics (2017) 12.
[26] Q. Wang et al., IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2724–2743.
[27] O. Hartig, (2014) arXiv:1409.3288.
[28] X. Zhang et al., Computer Physics Communications 211 (2017) 98–112.
[29] M. Mohd Ali et al., International Journal of Production Research (2020) 1–18.
[30] E.M. Sanfilippo et al., Computers in Industry 109 (2019) 182–194.
[31] E. Kim et al., Matter 1 (2019) 8–12.
[32] W.D. Callister et al., John Wiley & Sons (2010).
[33] T. Tudorache et al., Semantic Web 4 (2013) 89–99.
[34] A. Khosravani et al., Acta Materialia 123 (2017) 55–69.