Enhancing Multiscale Simulation Data Management with Domain Ontologies and an ELN: Addressing Challenges and Implementing Strategies

RESEARCH PAPER

]u[ubiquity press

HAFIZ MUHAMMAD NOMAN

MICHAEL SELZER

*Author affiliations can be found in the back matter of this article

ABSTRACT

Data plays a fundamental role in advancing knowledge and driving research across all fields. Despite the abundance of data, several challenges still need to be addressed. These challenges include limited accessibility, heterogeneous data, lack of interconnections between associated topics, difficulty retrieving required information, semantic mismatches, and several other challenges. Ontologies provide a structured method to address these challenges effectively.

This study addresses two central aspects of ontology research. First, it details the multidisciplinary ontology development process, highlighting the challenges, mitigation strategies, and impacts on domain data management. It then offers guidelines for beginners and individuals with a background in data management on effective engagement in ontology creation.

Second, it introduces the Ontology for Multiscale Simulation methods (Onto-MS), constructed by following the guidelines from the first part. The ontology, developed in Web Ontology Language (OWL) using Protégé, integrates with other ontologies, such as the Elementary Multiperspective Material Ontology (EMMO), aligning this research with the Linked Data concept. A custom Python script was used to incorporate the ontology into an Electronic Laboratory Notebook (ELN), enabling the automatic creation of knowledge graphs and systematic data organization conforming to the ontology.

This research successfully answers the fundamental questions in interdisciplinary or domain-level ontology development. Onto-MS provides a robust framework for organizing and linking data in multiscale simulations within computational materials science. Furthermore, ontology incorporation into an ELN simplifies its integration into data management practices. While ontology development is ongoing, the current version is functional and continuously refined with new insights and feedback.

CORRESPONDING AUTHOR:

Hafiz Muhammad Noman

Institute of Nanotechnology (INT), Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

hafiz.noman@kit.edu

KEYWORDS:

Ontology; Domain-level ontology; Multiscale Simulation; Web Ontology Language (OWL); Research Data management

TO CITE THIS ARTICLE:

Noman, HM and Selzer, M 2025 Enhancing Multiscale Simulation Data Management with Domain Ontologies and an ELN: Addressing Challenges and Implementing Strategies. *Data Science Journal*, 24: 28, pp. 1–22. DOI: https://doi.org/10.5334/dsj-2025-028

1. INTRODUCTION

The integration of computing and information principles across disciplines is gaining momentum and broader adoption. As a result, it has given rise to various interdisciplinary fields, such as materials and biomedical informatics. This approach substantially complements traditional research and supports the growing demands of scientific advancement.

Material informatics, for example, applies information science principles to improve the understanding of material data, long-term storage, and retrieval, ultimately accelerating data-driven discoveries in materials science.

A critical strategy for organizing scientific information in computational environments is the use of ontology, which formally describes a system through distinct concepts and their interrelationships (Ghedini et al., 2020). A domain ontology, however, formalizes concepts and relationships relevant to a particular community or an area of knowledge (ISO/IEC 21838-1, 2021). Concepts in ontology are simplified and abstract representations of phenomena, entities, or ideas. They are formally defined and serve as the building blocks of an ontology.

The aim of the ontology construction is to foster a shared understanding, enable the reuse of domain data, facilitate interoperability among information systems, and standardize knowledge representation (Pinto and Martins, 2004; Noy and McGuinness, 2001). However, constructing domain or interdisciplinary ontologies, such as those for simulation methods in materials science, presents several challenges. The variety of terminology across disciplines complicates the development of a shared vocabulary containing clear and consistent definitions. These ontologies must accurately represent and logically interconnect concepts from different scientific areas. Simulations in material science span multiple length scales, making it difficult to integrate them within a unified framework. Moreover, developers often struggle to capture domain-specific requirements, leading to overly complex or too abstract designs, ultimately making them unattractive and less reusable.

Since their introduction, ontologies have been explored for potential benefits across various fields of study. The necessity for ontology building, along with strategies and methodologies, has been extensively discussed in the literature (Pinto and Martins 2004; Noy and McGuinness 2001; Jones et al., 1998; Arp et al., 2015). As a result, numerous ontologies have been developed; for instance, more than 40 ontologies have been reported within the domain of material science alone (De Baas et al., 2023). While these studies mainly focus on describing concepts and relationships, they lack details of the challenges associated with building a domain ontology. These research works typically present a finalized ontology and recommend its integration into research. However, the answer to effectively exploiting or integrating ontologies into existing research is missing.

Additionally, ontologies have been employed in modeling and simulation contexts (Lacy and Gerber, 2004; Benjamin et al., 2006; Rubin et al., 2006; Turnitsa et al., 2010; Silver et al., 2011), often supported by semantic query languages (Grolinger et al., 2012) like SPARQL Protocol and RDF Query Language (SPARQL) (Prud'hommeaux and Seaborne, 2008) and Semantic Web Rule Language (SQWRL) (O'Connor and Das, 2009). Although these works provide detailed descriptions of the resultant ontologies, they fall short in terms of their implementation, efficient usage, and creation process. However, such efforts led to initiatives like MODA (CEN Workshop, 2018) (materials modeling data), a standard for describing simulation-related information, and its ontological transformation, Ontology for Simulation, Modelling, and Optimization (OSMO) (Horsch et al., 2021). However, OSMO lacks adequate connections to other ontologies, shows limited ontological clarity in class hierarchies, and presents challenges in constructing knowledge graphs.

Recent efforts like Physics-based Simulation Ontology (PSO) (Cheong and Butscher, 2019) and ontology-based simulation (Ontology-Sim) (May et al., 2022) attempt to address these limitations, but significant gaps remain in practical usability, multiscale representation, and integration strategies.

Thus, there is a clear need for an interdisciplinary ontology that supports multiscale capabilities in materials science simulation, overcomes challenges in its construction, offers user-friendly interaction, and complies with linked data and accepted ontology standards.

This research addresses these knowledge gaps by exploring the development and utilization of domain ontologies and answers the following key research questions in the first part.

1. What are the primary challenges in multidisciplinary ontology development, and what strategies are employed to overcome them?
2. How can individuals effectively contribute to ontology development and research data management?
3. How do ontological practices influence and enhance data management and utilization within the domain?

The second part of the study presents Onto-MS, the developed ontology, constructed based on the directions provided in the first part. Onto-MS concepts are defined with agreed-upon definitions and preferred labels. The concept hierarchy is structured through "is-a" or "kind-of" logic for reasoning. The ontology incorporates linked data principles and reuses established ontology elements by strictly adhering to ontological guidelines from the literature. The ontology is centered around three fundamental concepts in simulation and accommodates modeling across different scales (electronic, atomic, nano, micro, and macro).

Furthermore, this study explores the transformation of the Onto-MS into the Karlsruhe Data Infrastructure for Materials Science (Kadi4Mat) (Brandt et al., 2021), an ELN and repository, to enhance accessibility and usability. Practical insights are provided to create knowledge graphs based on Onto-MS and organize data related to their simulation use cases.

Onto-MS improves upon existing ontologies by effectively addressing core challenges such as domain ontology building, semantic interoperability, logical structuring, and practical reusability. Through its integration into an ELN environment, it supports streamlined, intuitive use of ontology concepts in multiscale simulations, making semantic technologies more accessible to researchers in materials science. Thus, it provides a more holistic and usable framework to document simulations.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature; Section 3 outlines the ontology development challenges, mitigation strategies adopted in Onto-MS, and impacts on data management; Section 4 presents guidelines on performing ontological research data management; Section 5 presents Onto-MS methodology and its integration into Kadi4Mat; Section 6 presents and discusses the results, focusing on the finalized ontology and its multiscale capabilities; and Section 7 concludes with key findings and future directions.

2. LITERATURE REVIEW

Ontologies emerged in computer science in the 1990s (Gruber, 1995). Over the years, there have been numerous attempts to apply ontologies to fields beyond computer science. The usage of ontologies became prevalent and famous in some fields, i.e., bioinformatics, due to the early development of some comprehensive ontologies (e.g., Gene Ontology) (Ashburner et al., 2000). However, the modeling and simulation (M&S) community has not widely adopted ontology practices and procedures. Researchers in M&S have undertaken the following efforts to explore the ontology potential. These selected efforts can broadly be divided into two categories.

The first aspect concerns ontology development, different approaches, and their importance in managing scientific data.

Foundational concepts related to the motivations, methodologies, and challenges of ontology development have been discussed by several authors (Pinto and Martins, 2004; Noy and McGuinness, 2001). In this research, their insights, particularly regarding best practices, guided the methods in section 4.2 and helped avoid common mistakes during ontology development. Similarly, Arp et al. (2015) explained best practices in ontology development and presented the Basic Formal Ontology (BFO) with its implications, further supporting the methodological perspective of this study.

Additionally, Jones et al. (1998) reviewed various ontology-building methodologies, providing a comparative discussion and outlining five key issues that were addressed in this research by ensuring reuse potential, selecting an appropriate development model, aligning informal and formal descriptions, and providing clear guidelines for extending ontologies.

Subsequently, Fathalla et al. (2018) introduced the SemSur ontology to enhance the communication of research publications. It shifts from traditional text-based formats to a transparent and comparable structure that uses a unified knowledge base to capture information from survey and review articles. Following this, Matentzoglu et al. (2018) provided community-reviewed Minimum Information for Reporting an Ontology (MIRO) guidelines for documenting ontologies. Those guidelines aim to improve ontology documentation quality, consistency, and completeness. MIRO has been taken into consideration during ontology documentation.

In the materials science domain, Himanen et al. (2019) discussed the main challenges related to irrelevance, incompleteness, and non-standardization, presenting ontologies as a remedy that parallels the objectives of the developed ontology (Onto-MS) in addressing similar issues.

Most recently, Bayerlein et al. (2024) demonstrated how the Platform Material Digital Core Ontology (PMDco) bridges semantic gaps across various domains. This is a strategy that aligns with this research effort to ensure interoperability between domain-specific knowledge.

The second aspect focuses on the use of ontologies in modeling and simulation.

Early efforts in this field began when Lacy and Gerber (2004) demonstrated the crucial role of the Web Ontology Language (OWL) in representing information on the Semantic Web for the simulation and modeling domain. Their arguments underlie the choice of OWL as the preferred ontology language in this study.

Similarly, Benjamin et al. (2006) emphasized the value of ontologies in facilitating simulation modeling, while Rubin et al. (2006) described an ontology-based framework for representing physiological dynamic simulations of circulation. Their framework effectively reduced the complexity of the process description and assisted in a more straightforward visualization of the complex models.

Turnitsa et al. (2010) discussed how ontology development in modeling and simulation differs from other disciplines, often being driven by research questions defined by the modeler.

More recently, Cheong and Butscher (2019) introduced the Physics-based Simulation Ontology (PSO), composed of PSO-Physics for physical phenomena and PSO-Sim for simulation solvers. However, these studies primarily focused on solvers and physical equations, leaving other essential aspects of simulation ontology unexplored, a gap this research intends to fill.

3. INTEGRATING MULTIDISCIPLINARY EXPERTISE IN ONTOLOGY DEVELOPMENT: CHALLENGES, MITIGATION STRATEGIES AND IMPACTS

Domain-specific research projects frequently involve collaboration across disciplines. Therefore, ontology development for a domain should incorporate the input and perspectives of all the contributors involved. Although the interdisciplinary approach offers numerous benefits, it also introduces several challenges.

Below are some of the challenges of an interdisciplinary approach, the mitigation strategies employed in Onto-MS, and their impacts on data.

The insights and strategies presented in this section have been derived from ontology-related initiatives in several large collaborative projects, including the Post Lithium Storage Cluster of Excellence (POLIS) ('POLiS - Cluster of Excellence,' 2019), the Multiscale Materials, Process and Device Modeling and Design Platform (MUSICODE) (Konchakova et al., 2022), and the National Research Data Infrastructure for Engineering Sciences (NFDI4Ing) (Schmitt et al., 2020).

In this study, the terms 'domain ontology', 'domain-level ontology', and 'multidisciplinary ontology' refer to the same concept.

3.1. CHALLENGES

Inconsistent terminology: It is observed that researchers from differing disciplines often use different terms to describe the same concept, which results in semantic heterogeneity and misunderstandings.

For instance, the terms 'solid electrolyte interphase' (SEI), 'passivation layer', 'interface layer', and 'surface film' are used interchangeably to describe the same phenomenon in different research scenarios.

In contrast, a term can have multiple interpretations depending on the discipline. A cell is the fundamental unit of life in biology. However, in chemistry, it refers to an electrochemical device used to generate electrical energy.

Communication challenges: When building domain ontologies, researchers from various disciplines contribute their specialized knowledge, which can reveal significant knowledge gaps. These gaps often result in communication challenges, as participants may not clearly understand each other's research perspectives.

The evaluation of the impact of SEI on battery performance across disciplines is an excellent example of this challenge. A material scientist focuses on the chemical properties and formation mechanisms of SEI. In contrast, a data scientist assesses performance based on data analysis, statistical methods, and machine learning, often without understanding the underlying electrochemical processes.

Difference in motivation: As mentioned above, researchers from various fields contribute to domain-level ontology building. However, they may not share equal motivation for engaging in research data management (RDM) activities. Such activities are often seen as an additional burden rather than a vital part of the research. This disparity can delay progress and cause under-participation in ontology development.

Depth and breadth of the ontology: The breadth of the ontology is the extent to which it covers the domain, while depth pertains to the level of detailed knowledge it includes (Yao et al., 2011). Maintaining a balance between the two is essential: a highly detailed ontology may become overly complex and narrowly focused, whereas one emphasizing broad coverage might lack specificity and fail to engage the target audience.

Interdisciplinary expertise: Until this point, it has been established that domain ontology building is an interdisciplinary effort. Thus, it requires a project leader who can understand the data from different fields and effectively communicate across disciplines. However, finding such an expert is often challenging.

Resource constraint and available expertise: Ontology development involving domain experts demands significant time, effort, and resources to achieve project goals. Coordinating interdisciplinary teams and managing project logistics can be resource-intensive, especially for large-scale projects.

3.2. MITIGATION OF CHALLENGES IN ONTO-MS

Below are mitigation strategies adopted during the development of Onto-MS to address the challenges. These are presented in the same order as the challenges above and are addressed individually.

- Shared conceptualization is central to ontology development. In Onto-MS, extensive efforts were dedicated to achieving consensus on concept definitions, possible interconnections, and alignment with established vocabularies. These efforts included monthly ontology workshops with researchers engaged in simulation activities at different length scales. As a result, a standardized glossary was developed for foundational concepts like *Task*, *Model*, *Algorithm*, and others. This glossary was initially maintained in a shared document, which was subsequently merged with ontology documentation and published on GitHub.
- To overcome the communication challenges, regular workshops and open discussion sessions were organized. These sessions, led by an ontologist, brought together research data managers and simulation experts from diverse backgrounds, such as materials science, electrochemistry, physics, and mechanical engineering. The structured discussions helped clarify domain-specific viewpoints, resolve misunderstandings, and bridge knowledge gaps.
- Outreach programs, such as introductory ontology sessions, were offered at the beginning of the project to educate researchers about the long-term benefits of

ontologies. Subsequent webinars demonstrated how domain-specific ontology approaches can support tasks like model validation and data integration across scales. A strong emphasis was placed on treating research data management as an integral part of the research rather than extra work.

- User-friendly tools such as the Kadi ecosystem (Brandt et al., 2021), detailed guidelines (as provided in Section 5), and automated workflows were introduced to integrate data management into daily research routines. The core Onto-MS structure has been incorporated into Kadi4Mat (as specified in Section 6.3) to ensure project-wide consistency and reusability. All project members are currently adopting the ontology structure within Kadi4Mat to describe their specific use cases.
- The breadth of the ontology was defined to cover essential concepts for multiscale simulation based on the set requirements (see Section 6.1 for detailed requirements). The perspective of participating simulation users was carefully considered when specifying the depth of the ontology, which ensured the inclusion of all necessary concepts to describe a simulation process from start to end. This balance reduces complexity and increases ontology acceptance.
- Onto-MS was developed iteratively, where core concepts are initially focused on, and then the ontology is gradually expanded. Domain experts and end users were engaged in a continuous feedback loop through structured review sessions to refine depth and coverage.
- A multidisciplinary team comprising members with domain-specific knowledge and ontology development expertise was formed. The author coordinated these efforts, drawing on relevant experience in both ontology engineering and material science, thereby acting as an interdisciplinary expert (Keestra, 2017).

An interdisciplinary expert comprehends the intricacies of specialized fields and possesses skills across multiple disciplines, allowing them to approach problems and challenges comprehensively.

- The Onto-MS development was monitored through regular meetings, and no formal project management tools were employed. This approach proved sufficient for this work. However, for similar large-scale interdisciplinary projects, applying project management principles and best practices (Project Management Institute, 2021) is recommended to effectively plan, execute, monitor, and achieve all project goals.

3.3. ADVANTAGES OF INTERDISCIPLINARY ONTOLOGY APPROACH

While the general benefits of ontology development are discussed in the introduction and literature review, this section focuses on the domain-specific impacts of Onto-MS.

Eradication of Isolated Domain Concepts: Onto-MS eliminates isolated concepts by logically linking related entities in the multiscale simulation domain. This connectivity ensures meaningful clustering and supports improved reasoning across the domain.

Unified Understanding and Interoperability: The development of Onto-MS fosters shared conceptualization and standardized domain information. This shared understanding enhances semantic interoperability and bridges disciplinary gaps within the domain.

Knowledge Sharing and Collaboration: The development of Onto-MS has strengthened community ties and fostered a culture of collaborative research, which, in turn, promotes knowledge exchange, creates a coherent research environment, and creates opportunities for collaboration on other relevant research avenues.

Broader Impact: Collaboration across disciplines has expanded the potential impact and applications of Onto-MS beyond individual research fields. By spanning multiple disciplines, Onto-MS facilitates interdisciplinary research, cross-domain problem-solving, and efficient data exchange.

4. CONNECTING RESEARCHERS WITH ONTOLOGY DEVELOPMENT

The following steps help researchers with limited ontology or data management knowledge to bridge the gap between research and data management by using ontological methods.

By following these instructions, researchers can fulfill most ontology-building requirements without the need for direct support from an ontologist. Although these steps are sequential, they may also be iterative or concurrent based on the research context.

1. **Experiment documentation:** Carefully record all the relevant data about an experiment so that no critical details are overlooked.
2. **Digital storage of data:** Whenever data documentation is done physically, secure it digitally, preferably in an electronic notebook that allows easy access and retrieval for future analysis and reference.
3. **Data classification:** Data should be organized into meaningful categories to facilitate understanding and logical clustering. For example, temperature, pressure, and time can all be categorized as *process parameters* based on the specific process or use case. Data classification can co-occur during documentation in step 1.
4. **Metadata documentation:** Metadata provides indispensable information about data, enhancing the understanding and usability of a data resource. A metadata vocabulary, such as the Dublin Core Vocabulary (Dublin Core Metadata Initiative, 2019), can be incorporated to describe any resource type using its established metadata elements.
    - **Defining terminology:** Define the terms used to describe the recorded data to reduce ambiguity. This practice ensures that the data is understandable across researchers from multiple fields.
5. **Standard data format:** Formulate a consistent (standardized) format for documenting information across similar experiments to maintain uniformity and comparability. A literature review and feedback from domain experts can help to choose an appropriate format.
6. **Data format refinement:** The data recording format should be revised at regular intervals to incorporate user feedback and advancing research.
7. **Holistic approach:** Consider the holistic impact of the research. Identify the connections between distinct related topics within the domain under consideration. Holistic thinking helps eliminate isolated pieces of information and facilitate linked data.

These steps enhance data management and can be mapped to the first three layers of data evolution, or the Data Information Knowledge Wisdom Hierarchy (DIKW) (Frické, 2019), as shown in Figure 1 below.

The data layer (steps 1–3) is the foundational layer, consisting of raw, unstructured data that originates with documentation, storage, and basic classification. The information layer (steps 4–6) is the layer where data is enriched with context, standard terminologies, and a consistent format to make it more meaningful. The knowledge layer (step 7 and beyond) is the layer where information transforms into valuable knowledge by using structured data, holistic thinking, and the linking of relevant datasets within a domain. This layer then supports decision-making and reasoning and represents the operational domain of the semantic web and ontologies.

Figure 1 represents the progressive transformation of data. It shows how data can evolve through structured steps to enable ontology-based intelligent systems.

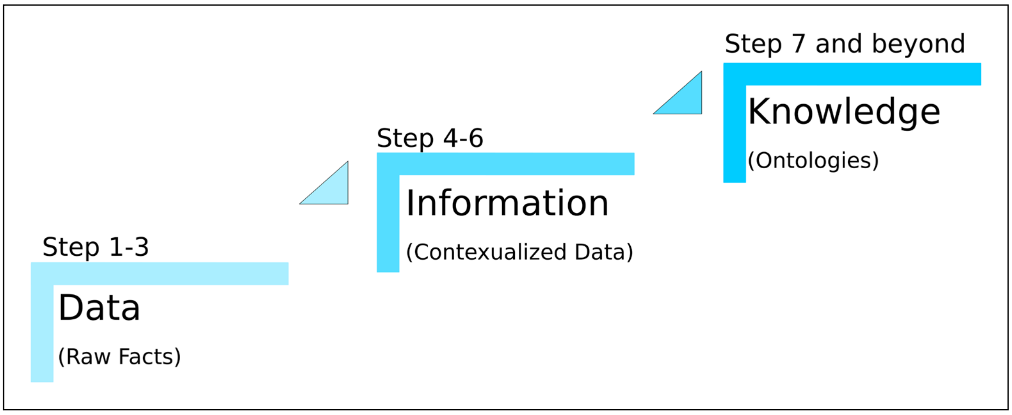


Figure 1 The DIK hierarchy as it aligns with the ontological data management process. The color-coded stair-step model maps data management actions (steps 1–7) onto the levels of data and information, ultimately leading to the knowledge and ontologies.

5. METHODOLOGY

5.1. ONTOLOGY DEVELOPMENT LIFECYCLE AND METHODOLOGY

In contrast to section 4, which introduced beginners to ontological data management, this section presents the actual development process of Onto-MS, which was developed through collaboration between an ontologist and domain experts to achieve the intended goal.

Ontology development is an iterative process that requires repeating specific steps to produce a robust version. These iterations ensure that the ontology is up-to-date and incorporates relevant knowledge.

Before going into detail about the ontology development process, it is important to emphasize that there is no single "correct" method to model a system; approaches vary according to the user needs and intended purpose. Nonetheless, this research predominantly drew inspiration from the methodologies outlined in two previously cited articles (Pinto and Martins, 2004) (Noy and McGuinness, 2001). Different modeling strategies, such as top-down and bottom-up strategies, are reported in the literature (Martins, 2004). However, the following steps and procedures have proven beneficial and relevant in the development of Onto-MS.

Domain and scope: The domain addressed by Onto-MS is multiscale simulation methods in computational materials science. In terms of scope, the ontology will encompass terminology across various simulation scales (atomic, molecular, and macroscopic) and support use cases including model selection, interdisciplinary communication, and the integration and standardization of simulation data.

Modelling strategy: The choice of modelling strategy is often driven by the specific needs and intended applications of the ontology. Onto-MS was developed using a middle-out modeling approach, which is particularly appropriate for a domain with well-established foundational concepts and flexible, evolving boundaries. This approach focuses on identifying familiar domain-relevant mid-level concepts that are neither overly abstract (as in upper-level ontologies) nor excessively specific (as in application ontologies). Concepts such as Task, Model Entity, and Solver were chosen because they are well understood and broadly applicable across users working on various simulation length scales and in different application areas. A key reason for choosing this strategy is the flexibility it provides. It enables the seamless upward mapping of concepts with established upper-level ontologies while accommodating the downward extension to incorporate more details such as types, parameters, and implementation-related details. Thus, it provides a stable and extensible core upon which an ontology can be expanded in both directions.

Purpose and requirements of ontology: The need for the ontology, the knowledge gaps it aims to address, and the essential features it must incorporate should be clearly defined. The primary purpose of Onto-MS is to capture and standardize the core simulation data in multiscale materials simulations to improve interdisciplinary communication and data management practices. A key objective is to identify and establish semantic connections across various simulation length scales to enable users to understand when outputs from one scale can serve as inputs for another. Onto-MS is intended to support the creation of knowledge graphs in an ELN. Additionally, it must align with existing upper-level ontologies and concepts to ensure interoperability and semantic consistency. The ontology must be usable and extensible to include new knowledge and evolving requirements.

Requirements are based on the purpose and involve deciding on the Onto-MS's features and the structure, such as:

- A shared vocabulary of formally defined and agreed-upon terms.
- Connection between various simulation length scales with appropriate relationships to represent a comprehensive and coherent domain model.
- Compatibility with ontology tools and standards, such as Protégé, and the use of OWL and SPARQL as the de facto ontology and query languages.
- Reuse of semantically equivalent concepts from upper-level ontologies such as EMMO (Ghedini et al., 2020) and others.

- Clear, user-friendly explanation and detailed documentation accessible through an open platform (e.g., GitHub) to support version tracking and future updates.

Domain knowledge acquisition and analysis: Domain knowledge was gathered through a comprehensive literature review (see section 2) and by examining existing relevant ontologies. To support the analysis and selection of available knowledge, domain experts provided valuable input through discussions and feedback to the author. While ontologists may not be experts in a specific field, their foundational understanding facilitates effective communication with domain experts and the integration of relevant domain concepts.

Ontology design: Classes, hierarchies, and relations: Building on the relevant literature identified earlier, the next step involved specifying ontology classes, subclasses, hierarchy, and restrictions. Core entities were modeled as classes based on their functional roles in simulation workflows. The class hierarchy in Onto-MS reflects 'is-a' or 'kind of' relationships, where subclasses represent more specific types of their parent classes. For example, *Physical model* and *Mathematical model* are subclasses of the *Model* class.

Figure 2 illustrates three fundamental ontology classes and their hierarchical organization in Onto-MS. The simulation component class, through its subclasses *Model, Task,* and *Algorithm,* captures the core concepts related to simulations. These concepts are further refined into more specific subclasses to represent concrete use cases in curly braces. To support multiscale representation, Onto-MS introduces two additional classes: Model Entity and Model Scale. The *Model Entity* specifies the physical entities being modeled, whereas the *Model Scale* characterizes the length scales addressed by the simulation methods.

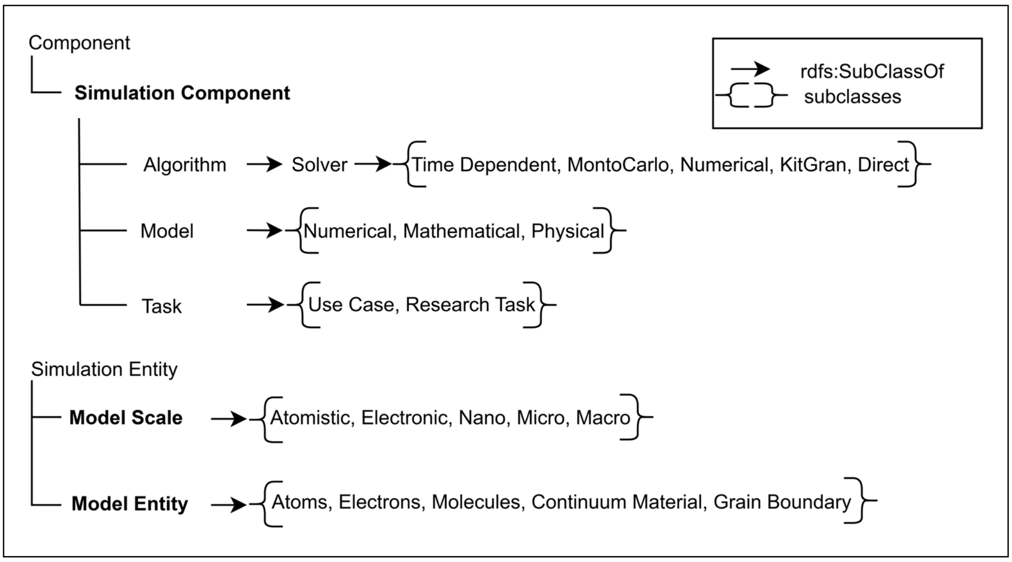

Component
Simulation Component
rdfs:SubClassOf
subclasses
Algorithm
Solver
Time Dependent, MontoCarlo, Numerical, KitGran, Direct
Model
Numerical, Mathematical, Physical
Task
Use Case, Research Task
Simulation Entity
Model Scale
Atomistic, Electronic, Nano, Micro, Macro
Model Entity
Atoms, Electrons, Molecules, Continuum Material, Grain Boundary

Figure 2 Selected class hierarchies in Onto-MS. The upper part illustrates the *Simulation Component* class and its subclasses. The lower part depicts the *Model Entity* and *Model Scale* classes with their subclasses.

Relationships between concepts were formalized using object properties, which were designed to capture the real-world connections derived from domain use cases and multiscale simulation workflows. Examples include determine (*Task, Model*), define (*Model, Input for Simulation*), is Realized Through (*Model, Solver*), etc. These object properties represent logical and functional relationships among different simulation components.

The actual data from simulation use cases (referred to as *individuals* in the OWL paradigm) are not included as part of the ontology. Instead, they are intended to be stored within Kadi4Mat, using the ontology structure as a guiding schema.

Each class in the ontology includes a formal definition to convey its meaning unambiguously. While class names may vary based on developer preference (e.g., Initial Measurement Results or Direct Measurement Results vs. Raw Measurement Results), the underlying semantics should remain consistent.

Reusing and extending concepts: An essential aspect of ontology development is the reuse of existing concepts that accurately represent the intended domain. The decision to adopt particular upper-level concepts was based on their semantic compatibility and alignment with concepts and relations in Onto-MS. Equivalent classes in external ontologies were identified using the TIB terminology service (TIB and Leibniz, 2022). These classes were then integrated and extended, where necessary, by adding subclasses to meet the requirements of Onto-MS while adhering to linked data standards. In this work, upper-level concepts are drawn predominantly from EMMO, with additional concepts adopted from Semantic Science Integrated Ontology (SIO) (Dumontier et al., 2014), the provenance ontology (PROV-O) (Lebo et al., 2013), an ontology for describing the generation of research data within a scientific activity (Metadata4Ing) (Iglezakis et al., 2023), and Schema.org (Schema.org, 2019), as needed.

Table 1 lists a representative subset of concepts to illustrate the main reuse and extension patterns in Onto-MS. The columns show concepts from upper-level ontologies, their corresponding Onto-MS extensions, and the rationale for reuse or adaptation. The table provides an overview of how each concept is integrated into Onto-MS and the reasoning behind its inclusion. Here, 'C' denotes a concept reused or considered as a class, and 'SC' indicates a concept reused as a subclass.

Table 1 External ontologies and their concepts reused or extended in Onto-MS, along with the rationale for their inclusion. (C) denotes class and (SC) denotes subclass.

ONTOLOGY NAME	CONCEPT REUSED	ONTO-MS EXTENSION	RATIONALE
EMMO	Component $^{(C)}$	Simulation Component $^{(SC)}$	To represent constituents of simulation systems
EMMO	Parameter $^{(C)}$	Parameter for Solver $^{(SC)}$	Solver-specific needs
EMMO+PROV-O	Participant $^{(C)}$ Agent $^{(SC)}$	-	Aligned with PROV Agent to represent responsible participant in processes
PROV-O	Organization $^{(SC)}$	Research Group ^{SC}	To represent research teams within institutions
OSMO	Granularity level $^{(C)}$	Used as Model Scale ^{C}	To represent multiscale simulation methods
SIO	Objective $^{(C)}$	Task Objective $^{(SC)}$	To specify the objectives of simulation tasks
Metadata4Ing	Method $^{(C)}$	Method $^{(C)}$	To describe the method used in simulation
Schema.org	Research Project $^{(C)}$	POLIS $^{(SC)}$	To support interoperability at the project level

The ontology was implemented in Protégé (Musen, 2015) and serialized in both OWL (.owl) and Turtle (.ttl) formats. Protégé's built-in 'Refactor > Move/Copy Axioms' feature was used to transfer relevant axioms from external ontologies into Onto-MS, facilitating concept reuse. To visualize the Onto-MS offline version, the OntoGraf plugin in Protégé was employed. OntoGraf offers flexibility to generate customized views by filtering specific relationships and node types.

The HermiT reasoner (integrated within Protégé) was used to validate the ontology and infer implicit axioms. The Wizard for documenting ontologies (Widoco) (Garijo, 2017) was utilized to generate a structured template for ontology documentation, while WebVOWL was adopted to provide an interactive visualization of the online version. Onto-MS documentation is maintained on GitHub, ensuring version control, transparency, and open access.

Additional tools used for ontology conversion and manipulation are discussed in Section 6.2.

Implement and Evaluate Ontology: After completing all the steps, Onto-MS was implemented and evaluated against the criteria defined in step 2 (Purpose and Requirements). The evaluation confirmed that Onto-MS met all the requirements and successfully fulfilled its intended purposes, such as standardizing terminology in the multiscale simulation domain, enhancing interdisciplinary communication, and supporting the construction of knowledge graphs or ontology instantiation in an Electronic Notebook (ELN).

Ontology Refinement: Onto-MS will be refined at regular intervals through a collaborative process involving domain experts, end-users, and an ontologist. The refinement process will cover validation against evolving research requirements and use cases. It will also consider incorporating new knowledge, a comprehensive review of class hierarchies and relations, and semantic alignment with upper-level ontologies. All modifications will be versioned and documented. They will be made publicly available on the webpage and in the Onto-MS GitHub repository. This approach ensures that Onto-MS remains scientifically current, accurate, interoperable, and aligned with emerging community standards. For detailed documentation and future updates, readers are encouraged to visit the official webpage (Noman, 2023).

Figure 3 illustrates the ontology development lifecycle adopted for Onto-MS. It encompasses all the essential stages and highlights the cyclic and iterative nature of the development process. The cyclic aspect indicates that development is continuous, with each cycle building upon the previous one. In contrast, the iterative nature implies that certain steps should be repeated to achieve high quality and robustness. However, not all stages in the lifecycle are equally iterative. Stages such as ontology design, knowledge analysis, reusing concepts, and evaluation may occur frequently, whereas defining the domain, scope, and requirements typically happens less often. It is crucial to recognize when to conclude the development process, as excessive cycles and iterations can lead to unnecessary complexity, potentially reducing the ontology's acceptance among its intended audience. Therefore, it is essential to ensure that the ontology maintains its ease of use for users.

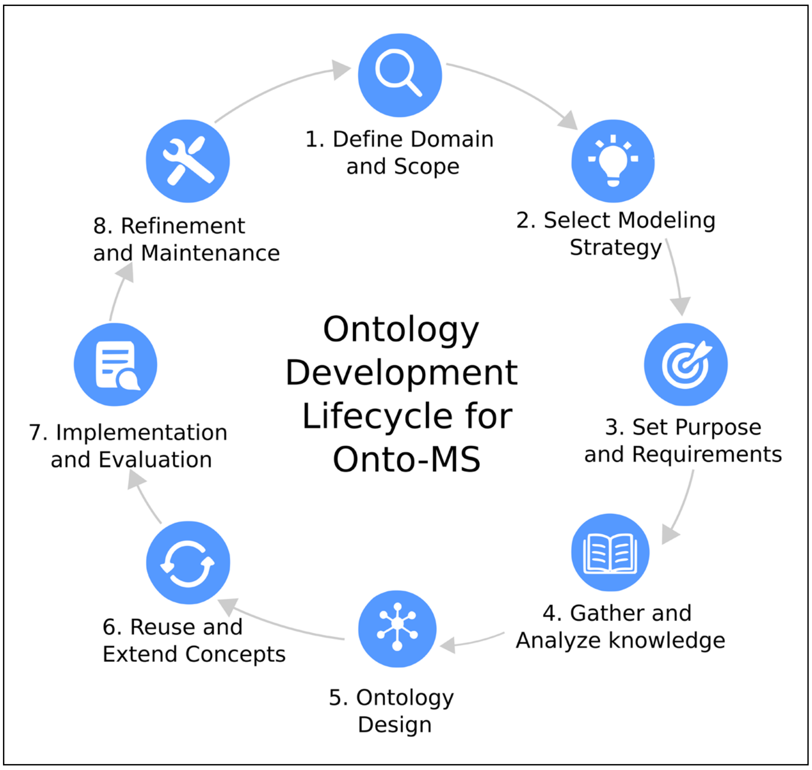

1. Define Domain and Scope
2. Select Modeling Strategy
3. Set Purpose and Requirements
4. Gather and Analyze knowledge
5. Ontology Design
6. Reuse and Extend Concepts
7. Implementation and Evaluation
8. Refinement and Maintenance
Ontology Development Lifecycle for Onto-MS

Figure 3 Ontology development lifecycle followed in Onto-MS, highlighting key stages such as domain definition, modeling strategy, implementation, evaluation, and refinement.

5.2. ONTOLOGY TRANSFORMATION TO ELN (I.E., KADI4MAT)

To construct a knowledge graph based on the developed ontology (ontology instantiation), Onto-MS is converted into Kadi4Mat resources.

Two primary types of Kadi4Mat resources have been utilized in this work: records and collections. In Kadi4Mat, a record is the fundamental building block and represents any type of digital or digitized object, such as research data, experimental setups, or individual processing steps. Each record is described using structured metadata that captures essential information about the entity in question. The record can be logically linked to other records, thereby supporting traceability and providing context.

In contrast, a collection is used to group multiple records. This enables related data entries, such as those belonging to a single experiment or simulation workflow, to be organized into a coherent and manageable unit (Brandt et al. 2021).

This conversion was carried out by using a custom Python script that leverages RDFlib (a Python library for parsing RDF data and executing SPARQL queries) (RDFLib documentation, 2025) and Kadi-APY (which serves as an interface to the Kadi4Mat platform). The complete script is available on Zenodo (Noman, 2024). Key steps in the script are outlined below.

Loading the ontology: The Turtle-formatted (.ttl) ontology file is parsed using the RDFlib `graph` object. The script identifies the main IRI of the ontology, which serves as a reference point for subsequent processing.

```
# See script on Zenodo for full implementation
graph = Graph ()
graph.parse (“OntoMS.ttl”, format=”ttl”)
main_iri = graph.value(predicate=RDF.type, object=OWL.Ontology)
```

Extracting names and descriptions: Once the ontology is loaded (step 1), two helper functions are defined to generate human-readable titles and descriptions for Kadi4Mat resources.

`get_resource_name` selects the first available property to use as the resource name in Kadi4Mat. The selection is done in the following order of priority: SKOS `prefLabel` > RDFS `label` > DCT `title` > IRI fragment.

`get_resource_description` populates the description field of records and collections by selecting the first available property, prioritizing SKOS `definition` > RDFS `comment` > DCTERMS `description` > legacy DC `description`.

By following this approach, the script ensures that each record and collection in Kadi4Mat is annotated with meaningful metadata, as specified by the ontology.

```
Name = get_resource_name (graph, entity)
Description = get_resource_description (graph, entity)
```

Building on this metadata extraction, the next step involves querying restrictions defined in the ontology.

Querying ontology restrictions: All OWL subclass restrictions are retrieved through a SPARQL query. These restrictions include existential (`someValuesFrom`), universal (`allValuesFrom`), minimum cardinality (`minQualifiedCardinality`), maximum cardinality (`maxQualifiedCardinality`), and exact cardinality (`qualifiedCardinality`). The query returns subclass-property-value triples, ensuring that all semantic constraints on subclasses defined in the ontology are available for constructing a knowledge graph in kadi4mat.

```
Restrictions = graph.query(“...SPARQL query capturing all restrictions...”)
```

Creating Kadi4Mat records and collections: At this step, the script establishes a connection with a chosen kadi4Mat instance (here: demo instance) using the kadi-apy library. This setup enables interaction with the Kadi4Mat programmatically.

```
manager = KadiManager(instance=”demo_instance”)
```

The function `create_KGs` first creates a collection whose title and description are sourced from the ontology (using the name and description variables from step 2). It then iterates over subclass-property-value triples to create kadi4Mat resources. This systematic procedure transforms ontology classes into interconnected Kadi4Mat records within a collection.

```
def create_KGs(suffix=”X”):
          # 1. Create a collection
          collection = manager.collection (“collection parameters”)

          For each subclass-property-value triple:
          # 2. Create records for the subclass and the value class
          record_1 = manager.record (“record parameters”)
          record_2 = manager.record (“record parameters”)

          # 3. Link both records by using the property
          record_1.link_record(record_to=record_2.id, name=property_name)

          # 4. Add both records to the collection
          collection.add_record_link (“record_id”)
```

Execution:

The primary knowledge graph function `(create_KGs)` is executed with an optional suffix parameter to ensure unique identifiers.

```
create_KGs(suffix="Provide a unique suffix")
```

The resultant Kadi4Mat collection is titled and described in accordance with the ontology, containing records connected through relevant object properties.

Selection criteria:

- The SPARQL query targets subclass restrictions on terminal (leaf) classes in the ontology.
- Only those leaf classes that are connected to other classes via restrictions are included.

This approach guarantees that the generated knowledge graph encompasses the most relevant classes, while the complete class hierarchy remains accessible within the ontology. Future iterations of the script will expand support for additional ontology axioms and also offer users the flexibility to select classes for knowledge graph construction.

Finally, it is important to note that transforming an ontology into Kadi4Mat resources requires certain prerequisites, as detailed in Appendix A. For comprehensive details on the usage and configuration of kadi-APY, refer to its official documentation (kadi-apy documentation, 2025).

6. RESULTS & DISCUSSION

Although the ontology was constructed in Protégé, the visuals have been reconstructed for clarity, providing a simplified perspective. The ontology focuses on three key concepts: Task, Model, and Algorithm, as illustrated in Figure 4 and explained below.

1. Task

   A *task* is an assignment or activity that requires action to accomplish a particular goal within a specific field.

2. Model

   A *model* in a simulation study represents or abstracts a real-world system, a phenomenon, or a process designed to replicate its behavior and characteristics.

3. Algorithm

   An *algorithm* represents a set of stepwise instructions for solving or accomplishing a task. A *solver* is a specific class of an algorithm.

This section does not cover the definition of every concept in detail; readers are encouraged to study the detailed online documentation (Noman, 2023) for comprehensive definitions, connections between concepts, and applied restrictions.

Figure 4 illustrates the Task class and its relationships with other classes. Arrows marked with 'o' denote optional properties (or zero cardinality), meaning the associated class values are not always required and can be left blank if the information is unavailable. The Task class is linked to *Material*, *Material Shape*, *Method*, and *Time Interval* through appropriate object properties with zero cardinality restrictions. It also connects to the *Task Objective*, *Participant*, and *Model* classes, where at least one value must be provided. The *Participant* concept encompasses two critical aspects of participation. The first one is the *Agent*. An *agent* is not necessarily a human and is responsible for an activity to take place. It can be considered the driver of the process. Meanwhile, the *Tool* includes everything that helps an agent perform the activity. This can be equipment, a device, or a piece of software. Based on purpose, the *Task* concept can be further specified as either *Research Task* or *Use Case*, as shown by the subclass relation.

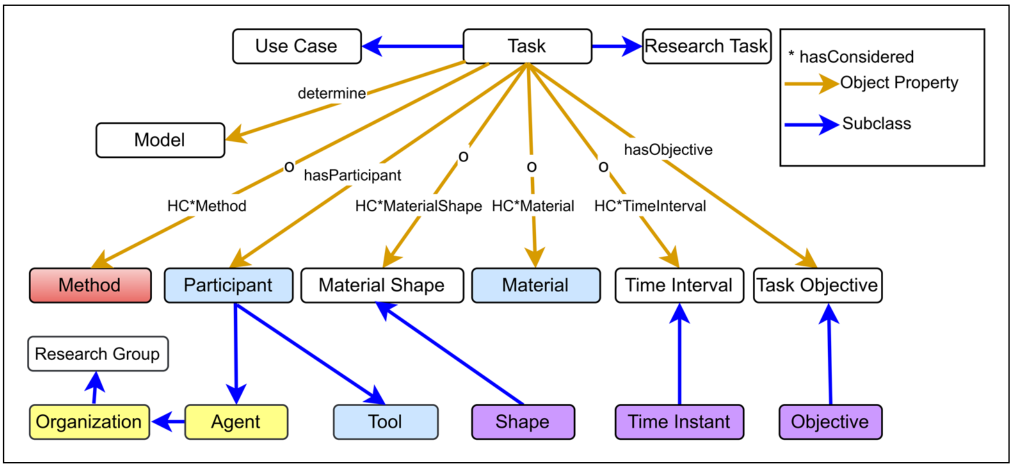


Figure 4 Task view: Shows the different concepts in the Onto-MS. Colored boxes represent the concepts imported from other ontologies, while uncolored boxes are native to Onto-MS. 🟦 EMMO 🟨 PROV-O 🟥 Metadata4Ing 🟪 SIO ⬜ Onto-MS. The color of the arrow signifies object properties or subclass relationships.

Figure 5 represents the central simulation components, described at the start of the results section, and extends the *Model* class to show its relationships with other concepts.

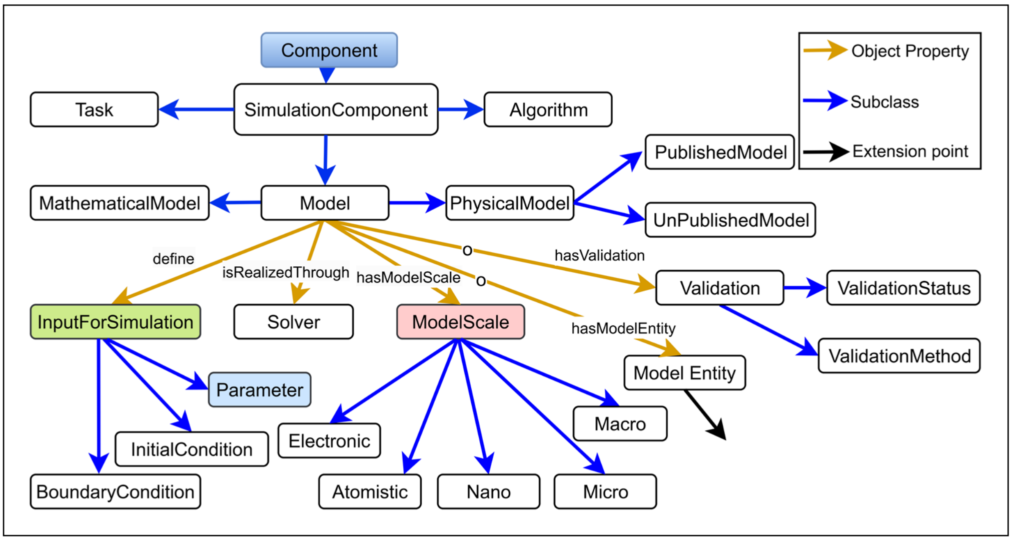


Figure 5 Model view: Shows the different concepts in the Onto-MS. Colored boxes represent the concepts imported from other ontologies, while uncolored boxes are native to Onto-MS. 🟦 EMMO 🟥 OSMO/MODA. 🟩 MUSICODE ⬜ Onto-MS. While the color of the arrow signifies object property or subclass relationship.

It elaborates on the *Model* class and its relationships with other concepts. A model can be specified as *Mathematical*, *Physical*, or other type. There are further subdivisions called *Published* and *Unpublished Model*. For simplicity, only up to three subclasses are shown in the images, but additional subclasses, like a *Numerical Model*, can be added depending on the specific use case. A model specifies the *Input for Simulation* and is implemented through a *Solver*. The *Input for Simulation* comprises three main concepts: *Boundary Condition*, *Initial Condition*, and *Parameter*. The *Parameter* concept defines constants, material properties, or environmental values for simulation input that the model relies on to simulate real-world conditions. Through the *Model scale* and *Model Entity*, different length scales and entities can be accommodated. The *Model* class can also be connected to the *Validation* class with zero cardinality, which means that validation information is optional.

Figure 6 depicts the *Solver* class, a subclass of *Algorithm*, which can be specified as *Numerical*, *Monte Carlo*, etc. Additional subclasses like *kit Gran*, *Time-dependent*, and *Direct Solver* are documented. The *Solver* requires software for specific implementations with different access restrictions. The various access scenarios can be expressed through the *Access Condition* concept. The *Solver* operates with defined parameters as described in *Input for Simulation*, often using dedicated *software for Solver* to solve a task, and produces results classified as Raw or *Inferred Simulation Measurement Results*. *Raw Measurement Results* are direct measurements or observations requiring further analysis and interpretation. Therefore, they require a *Post Processor* or a *Tool* to do so. Alternatively, *Inferred Measurement Results* are derived from raw results and typically do not need additional processing.

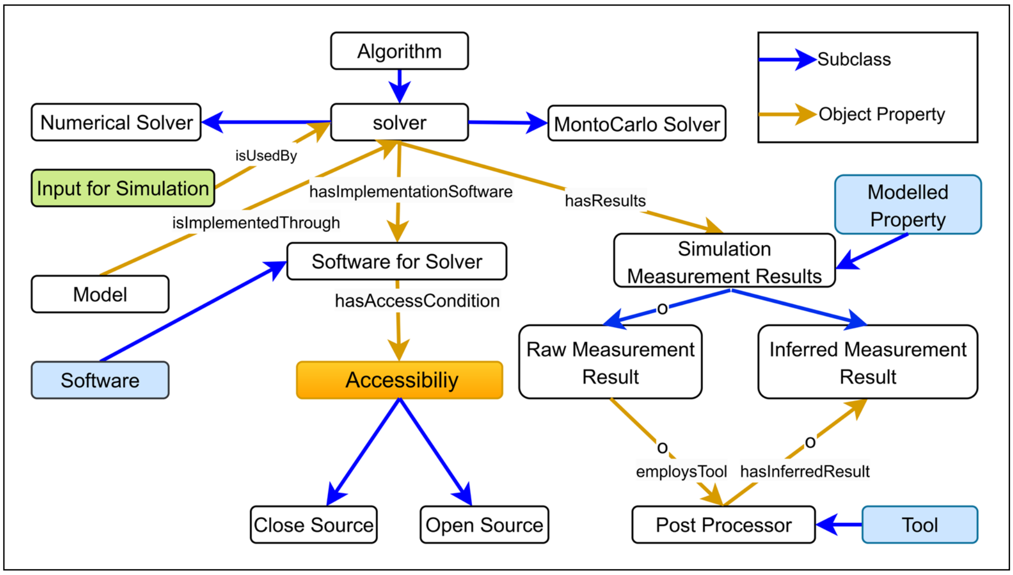


Figure 6 Solver view: Shows the different concepts in the Onto-MS. Colored boxes represent the concepts imported from other ontologies, while uncolored boxes are native to Onto-MS. 🟦 EMMO 🟨 TEMA □ Onto-MS. The arrow's color signifies object property or subclass relationship.

As detailed in the methodology (see Section 6.2), the ontology has been transformed into Kadi4Mat resources. Figure 7 illustrates these conversion results, where each colored circle represents a selective ontology class, arrows indicate directional relationships, and link names correspond to object properties in Onto-MS.

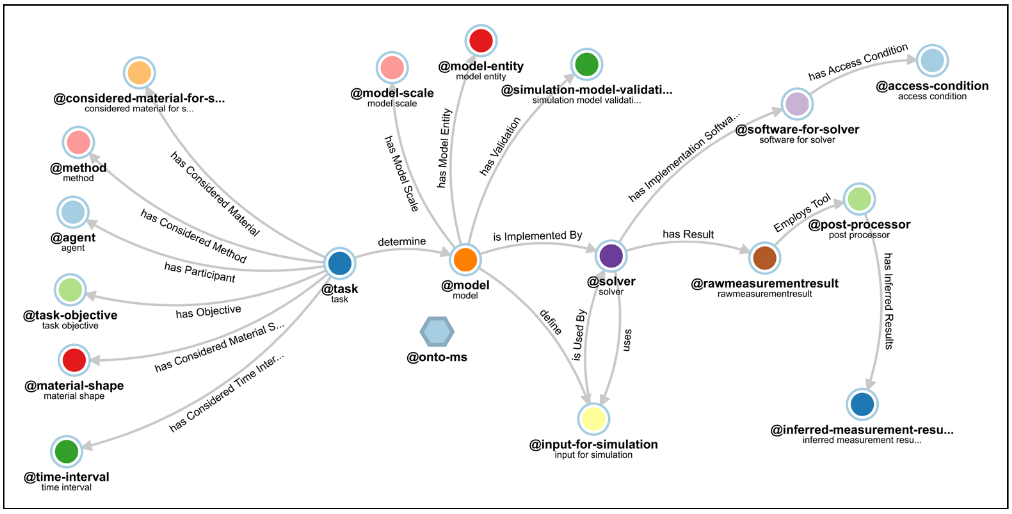


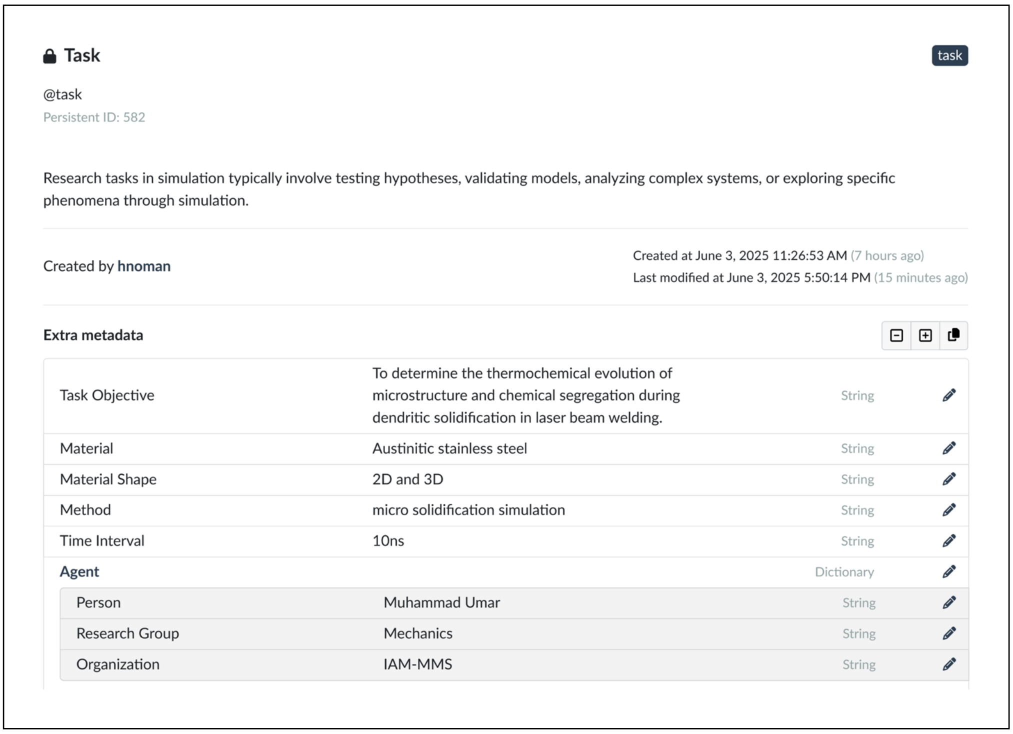
Figure 7 Onto-MS as 'onto-ms' collection in Kadi4Mat, where specific ontology classes are represented in records (shown in colored circles), and object properties are links between records in this collection.

When presented in their raw form, ontologies can overwhelm users with large numbers of classes, complex relationships, and intricate graph structures. As a result, many developed ontologies remain theoretical artifacts and rarely find their way into daily research practice. To avoid this outcome, the integration of Onto-MS in Kadi4Mat was indispensable. This integration enables Onto-MS to operate within real research environments and supports its practical adoption by researchers. In addition, it facilitates straightforward ontology instantiation and knowledge graph generation. Onto-MS in Kadi4Mat enhances the accessibility, usability, and structural organization of relevant data resources for end users.

Previously, metadata from various simulation projects, often produced by the members of the same research group, was stored in free-text or inconsistent formats, limiting both reusability and discoverability. By annotating simulation records with Onto-MS, the ELN now imposes a shared**,** semantically rich structure, enabling precise querying and reducing the need for preprocessing.

Onto-MS has already been employed to describe a substantial number of simulation use cases currently stored within Kadi4Mat, and this number is expected to grow. Figures 8–11 illustrate a representative case, where simulations were performed to investigate the thermochemical evolution of microstructure and chemical segregation during dendritic solidification in laser beam welding. Due to space constraints, the original 18 simulation records have been consolidated into four, and metadata fields have been selectively displayed to limit the visual complexity of the figures.

To apply the ontology structure to other similar scenarios, users can replicate the existing collection and modify the record metadata to suit their specific use case requirements.



Task task

@task
Persistent ID: 582

Research tasks in simulation typically involve testing hypotheses, validating models, analyzing complex systems, or exploring specific phenomena through simulation.

Created by **hnoman**

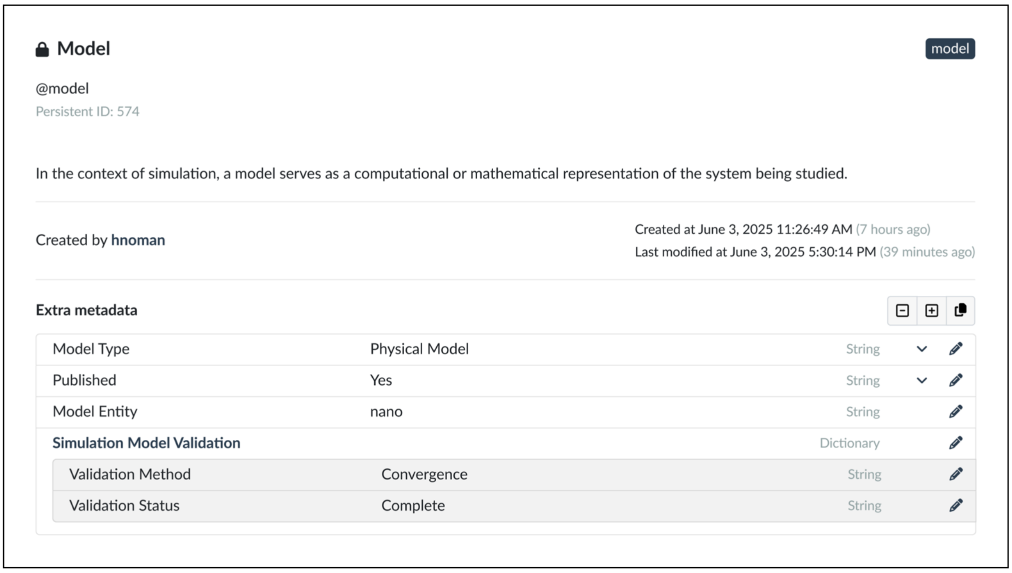
Created at June 3, 2025 11:26:53 AM (7 hours ago)
Last modified at June 3, 2025 5:50:14 PM (15 minutes ago)

Extra metadata ⊟ ⊞

Task Objective	To determine the thermochemical evolution of microstructure and chemical segregation during dendritic solidification in laser beam welding.	String
Material	Austinitic stainless steel	String
Material Shape	2D and 3D	String
Method	micro solidification simulation	String
Time Interval	10ns	String
Agent		Dictionary
Person	Muhammad Umar	String
Research Group	Mechanics	String
Organization	IAM-MMS	String

Figure 8 Representation of a Task record from the Onto-MS ontology collection, where classes such as Method, Material Shape, and Geometry have been consolidated into a single record for demonstration purposes.

The task record, as shown in Figure 8, consists of ontology classes such as Task Objective, Material, Material Geometry, Method, Time Interval, and Agent. Figure 9, in addition, presents the Model record, which includes the Model Entity, Model Type, and Model Validation classes.



Model model

@model
Persistent ID: 574

In the context of simulation, a model serves as a computational or mathematical representation of the system being studied.

Created by **hnoman**

Created at June 3, 2025 11:26:49 AM (7 hours ago)
Last modified at June 3, 2025 5:30:14 PM (39 minutes ago)

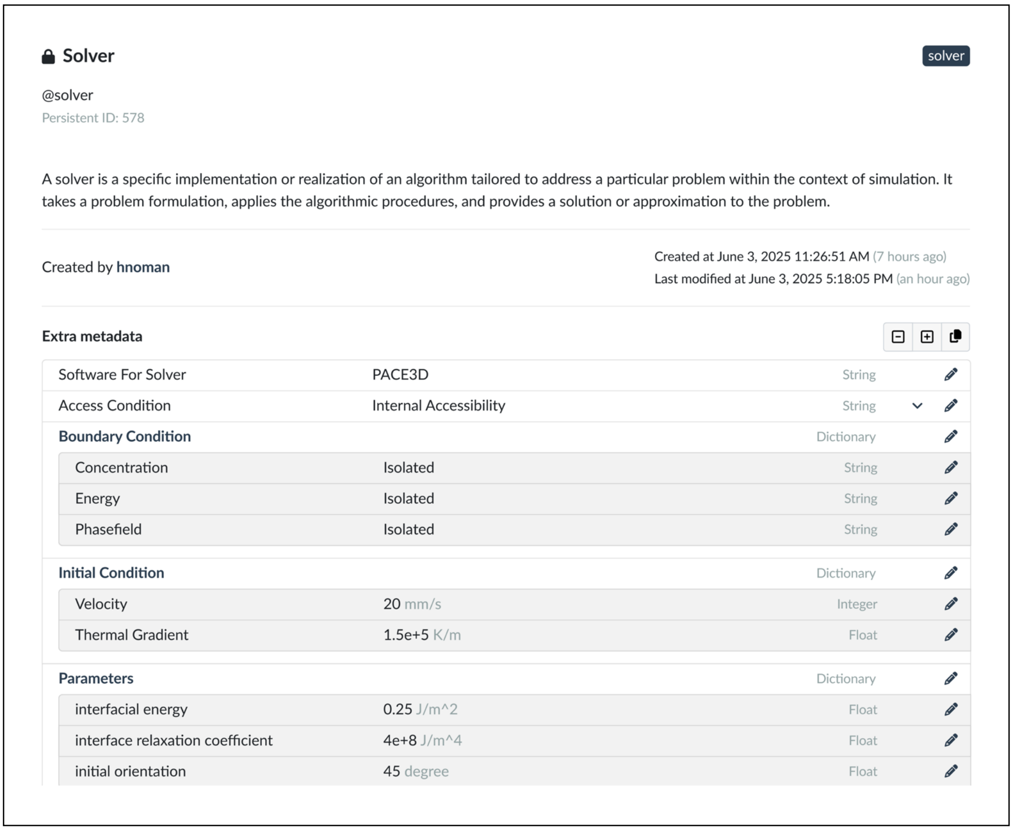
Extra metadata ⊟ ⊞

Model Type	Physical Model	String
Published	Yes	String
Model Entity	nano	String
Simulation Model Validation		Dictionary
Validation Method	Convergence	String
Validation Status	Complete	String

Figure 9 Representation of the Model record from the Onto-MS ontology collection in Kadi4Mat, which consolidates related classes or records such as *Model Type*, *Model Entity*, *Simulation Model Validation*, and others.

The Solver record, as shown in Figure 10, consists of ontology classes such as Software for Solver, Access Condition, and the Input for Simulation class, which is further divided into Boundary Condition, Initial Condition, and Parameter, in accordance with Onto-MS. Finally, Figure 11 presents the Results of the simulation use case, incorporating ontology classes such as Raw Measurement Results, Tool, and Inferred Measurement Result.

Figures 8–11 reinforce that Onto-MS can effectively capture and standardize data across diverse simulation use cases.



Solver

solver

@solver
Persistent ID: 578

A solver is a specific implementation or realization of an algorithm tailored to address a particular problem within the context of simulation. It takes a problem formulation, applies the algorithmic procedures, and provides a solution or approximation to the problem.

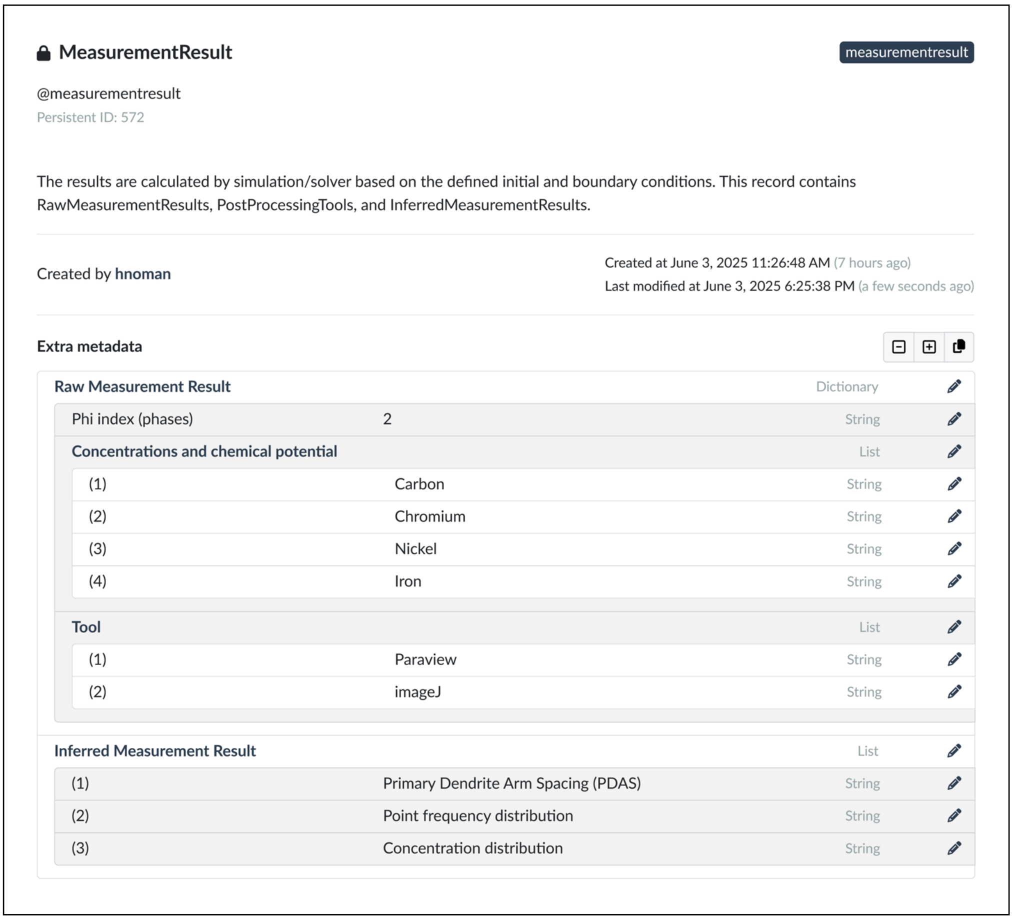
Created by hnoman

Created at June 3, 2025 11:26:51 AM (7 hours ago)
Last modified at June 3, 2025 5:18:05 PM (an hour ago)

Extra metadata

Software For Solver	PACE3D	String
Access Condition	Internal Accessibility	String
Boundary Condition		Dictionary
Concentration	Isolated	String
Energy	Isolated	String
Phasefield	Isolated	String
Initial Condition		Dictionary
Velocity	20 mm/s	Integer
Thermal Gradient	1.5e+5 K/m	Float
Parameters		Dictionary
interfacial energy	0.25 J/m^2	Float
interface relaxation coefficient	4e+8 J/m^4	Float
initial orientation	45 degree	Float

Figure 10 Representation of the Solver record from the Onto-MS ontology collection in Kadi4Mat, encompassing related classes or records such as Input for Simulation (e.g., Boundary Condition), Software, and Access Condition.



MeasurementResult

measurementresult

@measurementresult
Persistent ID: 572

The results are calculated by simulation/solver based on the defined initial and boundary conditions. This record contains RawMeasurementResults, PostProcessingTools, and InferredMeasurementResults.

Created by hnoman

Created at June 3, 2025 11:26:48 AM (7 hours ago)
Last modified at June 3, 2025 6:25:38 PM (a few seconds ago)

Extra metadata

Raw Measurement Result		Dictionary
Phi index (phases)	2	String
Concentrations and chemical potential		List
(1)	Carbon	String
(2)	Chromium	String
(3)	Nickel	String
(4)	Iron	String
Tool		List
(1)	Paraview	String
(2)	imageJ	String
Inferred Measurement Result		List
(1)	Primary Dendrite Arm Spacing (PDAS)	String
(2)	Point frequency distribution	String
(3)	Concentration distribution	String

Figure 11 Represents a Measurement Result record from the Onto-MS ontology collection in Kadi4Mat, which consolidates ontology classes or records such as Raw Measurement Results, Tool for Post-Processing, and Inferred Measurement Results.

Onto-MS is designed for multiscale simulations, as illustrated in Figure 12 below. A *model* class in the ontology is linked to the *model entity* (shown in Figures 5 and 9). It organizes model entities across different length scales, such as *Atomistic, Electronic, Nano,* and *Micro*, enabling the integration of information about diverse simulation methods. For instance, simulation methods from the NVPC project in POLIS ('POLiS - Cluster of Excellence', 2019), such as the density functional theory (DFT), the phase-field method (PFM), and pseudo-3D (P3D), can be appropriately categorized under their respective length scales. Although only a few methods are depicted, Onto-MS is highly extensible and can accommodate numerous other simulation techniques.

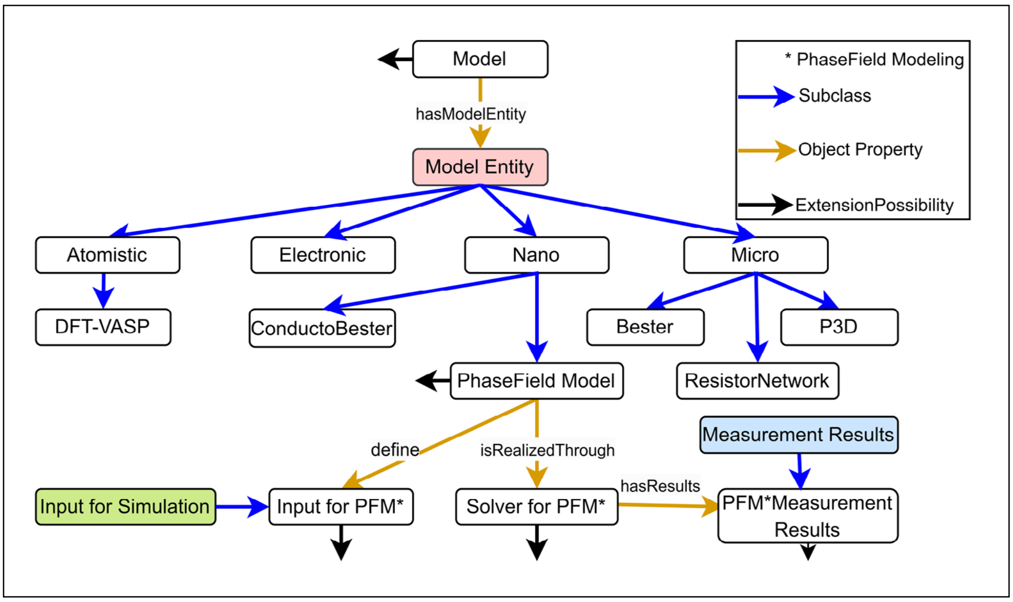


Figure 12 Representation of Onto-MS multiscale capabilities by integrating various length-scale simulation methods for the NVPC project.

The ontology provides detailed specifications for each method through specific classes. For instance, the Phase Field Model (under the nanoscale) is linked to other suitable ontology classes via predefined ontology linkages, with a similar approach used for all methods. The black arrows in Figure 12 indicate areas where further subclass relationships and object properties exist in Onto-MS but are not shown due to space constraints. Readers should refer to the online documentation for a comprehensive overview of how the NVPC use case is structured using Onto-MS.

6.1. LIMITATIONS AND PRACTICAL CHALLENGES

Onto-MS has been designed to address the needs of simulation users who aim to describe the use-case-specific data using an ontology. While it includes a broad range of domain-level concepts, it does not include all application-specific concepts. For example, an application ontology can be built only to cover simulation model selection and parameters.

The concepts and relationships within Onto-MS have been thoughtfully developed to cover key aspects of multiscale simulation in materials science. Nevertheless, users may encounter scenarios where existing Onto-MS do not fully capture their specific data or needs. To address such issues, ontology is designed for extensibility, allowing the seamless addition of new concepts and relations. It is expected to become more comprehensive and robust through iterative refinement and community feedback.

However, it is essential to acknowledge that no ontology can realistically cover all possible scenarios, and a balance must be struck between completeness and practical usability.

Onto-MS integration with Kadi4Mat offers limited flexibility in allowing users to selectively designate ontology classes as either data records or metadata attributes. As a result, this can occasionally lead to excessive records, potentially making navigation and organization within the ELN more effort-intensive. To address this, a future version of the Onto-MS to Kadi4Mat transformation is planned to provide greater flexibility in defining which ontology classes appear as records or metadata.

Moreover, the reverse mapping from Kadi4Mat back to the ontology, preferably in a triple-store environment, has not yet been implemented. This backward linkage will ensure semantic consistency, enable advanced querying and reasoning, and support bidirectional data enrichment between the ontology and the electronic lab notebook system. Integration into triple stores is the subject of ongoing research and will be discussed in more detail in future work.

7. CONCLUSION

Despite the abundance of ontologies across various disciplines, researchers and data managers still need practical, step-by-step guidance on developing, implementing, and utilizing ontologies, particularly in complex, multiscale simulation contexts.

In response to these challenges, the first part of this study provides well-defined procedures and strategies for overcoming obstacles associated with the development of multidisciplinary ontologies. Simple and easy-to-implement guidelines have been provided to encourage beginners to adopt ontology-based data management practices. At the same time, an in-depth ontology development guide addresses the needs of experienced readers.

The second part of this research article introduces Onto-MS as a practical implementation of ontology modeling strategies. It is an ontology designed to formalize multiscale simulation concepts by adhering to established standards and integrating linked data principles. Onto-MS can seamlessly accommodate a variety of simulation methods. It has been incorporated into the Kadi4Mat platform to transform ontology use from a complex theoretical exercise into a simple, intuitive, and user-friendly tool. This enables researchers to organize, retrieve, and reuse simulation data efficiently within their daily workflow. Moreover, this integration directly supports the generation of knowledge graphs based on the developed ontology.

This study successfully presented a consistent and thoroughly structured concept with its practical implementation. This approach will be further refined in future studies and will serve as the basis for building domain ontologies for other multidisciplinary projects, expanding its applicability and utility.

APPENDIX A

Prerequisites for ontology conversion to Kadi4Mat:

To convert Onto-MS into Kadi4Mat resources, the following requirements must be completed.

Visit the Kadi4Mat Demo instance and create a user account. After logging in, navigate to Settings > Access Tokens and generate a Personal Access Token (PAT) for API authentication. Ensure that Python is installed in local environment by running.

```
Python --version
```

Then, install the library via the following cli command:

```
pip3 install kadi-apy
```

Configure kadi-apy for a specific kadi4Mat instance (e.g., demo instance). Use the following commands to create or update host and PAT.

```
kadi-apy config create      # Create a basic config file.
kadi-apy config set-host    # Change a host in the config file.
kadi-apy config set-pat     # Change a PAT in the config file.
```

Run the Conversion Script:

- Download the ontology file and rename it Onto-MS.owl.
- Download the Python conversion script from Zenodo.
- Place both files in the same directory.
- Execute the script to initiate the process.

Outcome:

Upon successful execution, a new collection will be created in the Kadi4Mat demo instance, containing 18 structured records derived from the Onto-MS ontology.

ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (DFG) through the project NFDI4ING (project number 442146713) and as part of the Excellence Strategy of the German Federal and State Governments via the POLiS project (project number 390874152). Additional support was provided by the Helmholtz Metadata Collaboration (HMC) through the MetaSurf project (HMC-0027).

The authors gratefully acknowledge the editorial support of Leon Geisen and Muhammad Umar for his assistance in image refinement and for providing simulation use case data.

Ontology development was conducted using Protégé.

FUNDING INFORMATION

We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Hafiz Muhammad Noman orcid.org/0009-0001-6710-6681
Institute of Nanotechnology (INT), Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

Michael Selzer orcid.org/0000-0002-9756-646X
Institute of Nanotechnology (INT), Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany; Institute for Digital Materials Science (IDM), Karlsruhe University of Applied Sciences, Moltkestraße 30, 76133 Karlsruhe, Germany

REFERENCES

Arp, R., Smith, B. and **Spear, A.D.** (2015) *Building ontologies with Basic Formal Ontology*. Cambridge, MA: Massachusetts Institute of Technology. Available at: https://doi.org/10.7551/mitpress/9780262527811.001.0001

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and **Sherlock, G.** (2000) 'Gene Ontology: Tool for the unification of biology', *Nature Genetics*, 25, pp. 25–29. Available at: https://doi.org/10.1038/75556

Bayerlein, B., Schilling, M., Birkholz, H., Jung, M., Waitelonis, J., Mädler, L. and **Sack, H.** (2024) 'PMD Core Ontology: Achieving semantic interoperability in materials science', *Materials & Design*, 237, p. 112603. Available at: https://doi.org/10.1016/j.matdes.2023.112603

Benjamin, P., Patki, M. and **Mayer, R.** (2006) 'Using ontologies for simulation modeling', *Proceedings of the 2006 Winter Simulation Conference*. Monterey, CA, USA, 3–6 December. Piscataway, NJ: IEEE, pp. 1151–1159. Available at: https://doi.org/10.1109/WSC.2006.323206

Brandt, N., Griem, L., Herrmann, C., Schoof, E., Tosato, G., Zhao, Y., Zschumme, P. and **Selzer, M.** (2021) 'Kadi4Mat: A research data infrastructure for materials science', *Data Science Journal*, 20, p. 8. Available at: https://doi.org/10.5334/dsj-2021-008 Website: https://kadi.iam.kit.edu/ (Accessed: 1 August 2025).

CEN Workshop (2018) Materials modelling – Terminology, classification and metadata. CWA 17284:2018. Brussels: CEN.

Cheong, H. and **Butscher, A.** (2019) 'Physics-based simulation ontology: An ontology to support modelling and reuse of data for physics-based simulation', *Journal of Engineering Design*, 30, pp. 655–687. Available at: https://doi.org/10.1080/09544828.2019.1644301

De Baas, A., Nostro, P.D., Friis, J., Ghedini, E., Goldbeck, G., Paponetti, I.M., Pozzi, A., Sarkar, A., Yang, L., Zaccarini, F.A. and **Toti, D.** (2023) 'Review and alignment of domain-level ontologies for materials science', *IEEE Access*, 11, pp. 120372–120401. Available at: https://doi.org/10.1109/ACCESS.2023.3327725

Dublin Core Metadata Initiative (2019) *DCMI: DCMI metadata terms*. Available at: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (Accessed: 1 January 2024).

Dumontier, M., Baker, C.J., Baran, J., Callahan, A., Chepelev, L., Cruz-Toledo, J., Del Rio, N.R., Duck, G., Furlong, L.I., Keath, N., Klassen, D., McCusker, J.P., Queralt-Rosinach, N., Samwald, M., Villanueva-Rosales, N., Wilkinson, M.D. and **Hoehndorf, R.** (2014) 'The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery', *Journal of Biomedical Semantics*, 5, p. 14. Available at: https://doi.org/10.1186/2041-1480-5-14

Fathalla, S., Vahdati, S., Auer, S. and **Lange, C.** (2018) 'SemSur: A core ontology for the semantic representation of research findings', *Procedia Computer Science*, 137, pp. 151–162. Available at: https://doi.org/10.1016/j.procs.2018.09.015

Frické, M. (2019) 'The knowledge pyramid: The DIKW hierarchy', *Knowledge Organization*, 46(1), pp. 33–46. Available at: https://doi.org/10.5771/0943-7444-2019-1-33

Garijo, D. (2017) 'WIDOCO: A wizard for documenting ontologies', *International Semantic Web Conference*. Cham: Springer, pp. 94–102. Available at: https://doi.org/10.1007/978-3-319-68204-4_9

Ghedini, E., Goldbeck, G., Friis, J., Hashibon, A. and **Schmitz, G.** (2020). *European Materials Modelling Ontology. Version 0.9.9. European Materials Modelling Council (EMMC)*. Available at: emmo-repo.github.io/versions/0.9.9/emmo.pdf.

Grolinger, K., Capretz, M.A.M., Marti, J.R. and **Srivastava, K.D.** (2012) *Ontology–based representation of simulation models*. London, Ontario, Canada: The University of Western Ontario (Electrical and Computer Engineering Publications, 30). Available at: https://ir.lib.uwo.ca/electricalpub/30.

Gruber, T.R. (1995) 'Toward principles for the design of ontologies used for knowledge sharing', *International Journal of Human-Computer Studies*, 43(5–6), pp. 907–928. Available at: https://doi.org/10.1006/ijhc.1995.1081

Himanen, L., Geurts, A., Foster, A.S. and **Rinke, P.** (2019) 'Data-driven materials science: Status, challenges, and perspectives', *Advanced Science (Weinheim)*, 6(21), p. 1900808. Available at: https://doi.org/10.1002/advs.201900808

Horsch, M.T., Toti, D., Chiacchiera, S., Seaton, M.A., Goldbeck, G. and **Todorov, I.T.** (2021) *OSMO: Ontology for Simulation, Modelling, and Optimization*. Zenodo. Available at: https://doi.org/10.5281/zenodo.5084394 (Accessed: 01 April 2025).

Iglezakis, D., Terzijska, D., Arndt, S., Leimer, S., Hickmann, J., Fuhrmans, M. and **Lanza, G.** (2023) 'Modelling scientific processes with the m4i ontology', *1st Conference on Research Data Infrastructure (coRDI) – Connecting Communities, 2023*. Karlsruhe, Germany, 12–14 September. TIB Open Publishing, pp. 1–5., 1. Available at: https://doi.org/10.52825/cordi.v1i.271

Information Technology—Top-Level Ontologies (TLO)—Part 1: Requirements, Standard ISO/IEC 21838-1:2021 (2021) Available at: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:21838:-1:ed-1:v1:en (Accessed: 31 December 2024).

Jones, D., Bench-Capon, T. and **Visser, P.** (1998) 'Methodologies for ontology development', *IT&KNOWS - Information Technology and Knowledge Systems, 15th IFIP World Computer Congress*. Vienna, Austria, and Budapest, Hungary.

kadi-APY documentation (2025) Available at: https://kadi-apy.readthedocs.io/en/stable/ (Accessed: 15 April 2025).

Keestra, M. (2017) 'Metacognition and reflection by interdisciplinary experts: Insights from cognitive science and philosophy', *Issues in Interdisciplinary Studies*, 35, pp. 121–169. http://interdisciplinarystudies.org/docs/Vol35_2017/08_121-169.pdf.

Konchakova, N., Klein, P., Lidorikis, E., Laskarakis, A., Cavalcanti, W.L. and **Friis, J.** (2022) *Position paper: Open innovation in horizon Europe*. Available at: https://doi.org/10.5281/ZENODO.5848551 (Accessed: 15 April 2025).

Lacy, L. and **Gerber, W.** (2004) 'Potential modeling and simulation applications of the Web Ontology Language - OWL', *Proceedings of the 2004 Winter Simulation Conference, 2004*. Washington, D.C., 5–8 December. Piscataway, NJ: IEEE, pp. 257–262. Available at: https://doi.org/10.1109/WSC.2004.1371325 (Accessed: 15 September 2025).

Lebo, T., Sahoo, S.S., McGuinness, D.L., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S. and **Zhao, J.** (2013) *PROV-O: The PROV Ontology*. W3C Recommendation, 30 April 2013. Cambridge, MA: World Wide Web Consortium (W3C). Available at: https://www.w3.org/TR/prov-o/

Matentzoglu, N., Malone, J., Mungall, C. and **Stevens, R.** (2018) 'MIRO: Guidelines for minimum information for the reporting of an ontology', *Journal of Biomedical Semantics*, 9, p. 6. Available at: https://doi.org/10.1186/s13326-017-0172-7

May, M.C., Kiefer, L., Kuhnle, A. and **Lanza, G.** (2022) 'Ontology-based production simulation with OntologySim', *Applied Sciences*, 12, p. 1608. Available at: https://doi.org/10.3390/app12031608

Musen, M.A. (2015) 'The Protégé project: A look back and a look forward', *AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence*, 1(4). Available at: https://doi.org/10.1145/2757001.2757003

Noman, H.M. (2023) *Onto-MS: An ontology for multiscale simulation methods*. Available at: https://kit-mms.github.io/Onto-MS.github.io/ (Accessed: 1 June 2024).

Noman, H.M. (2024) 'Script for converting Ontology to an ELN (i.e., Kadi4Mat)'. *Zenodo* (Version 2.0). Available at: https://doi.org/10.5281/zenodo.13921378

Noy, N.F. and **McGuinness, D.L.** (2001) *Ontology development 101: A guide to creating your first ontology*. Stanford, CA: Stanford University, Knowledge Systems Laboratory.

O'Connor, M.J. and **Das, A.K.** (2009) 'SQWRL: A query language for OWL', in R. Hoekstra and P.F. Patel-Schneider (eds.) *Proceedings of OWL: Experiences and directions (OWLED), fifth international workshop (OWLED 2009),* Chantilly, VA, 23–24 October 2009. CEUR Workshop Proceedings, Vol. 529. Aachen: CEUR-WS, pp. 1–8.Available at: https://ceur-ws.org/Vol-529/owled2009_submission_42.pdf

Pinto, H.S. and **Martins, J.P.** (2004) 'Ontologies: How can they be built?', *Knowledge and Information Systems*, 6, pp. 441–464. Available at: https://doi.org/10.1007/s10115-003-0138-1

POLiS – Cluster of Excellence (2019) *POLiS – cluster of excellence*. Available at: https://www.postlithiumstorage.org/en (Accessed: 12 May 2025).

Project Management Institute (ed.) (2021) *The standard for project management and a guide to the project management body of knowledge (PMBOK guide)*. Seventh edition. Newtown Square, PA: Project Management Institute, Inc.

Prud'hommeaux, E. and **Seaborne, A.** (2008) *SPARQL query language for RDF. W3C recommendation, 15 January 2008*. Available at: https://www.w3.org/TR/rdf-sparql-query/ (Accessed: 30 May 2025).

RDFLib documentation (2025) *RDFLib: A Python library for working with RDF*. Available at: https://rdflib.readthedocs.io/en/stable/ (Accessed: 15 August 2025).

Rubin, D.L., Grossman, D., Neal, M., Cook, D.L., Bassingthwaighte, J.B. and **Musen, M.A.** (2006) 'Ontology-based representation of simulation models of physiology', *AMIA Annual Symposium Proceedings 2006*. Washington, D.C., 11–15 November. American Medical Informatics Association (AMIA), pp. 664–668.

Schema.org (2019) *Home* - schema.org. Available at: https://schema.org/ (Accessed: 1 January 2025).

Schmitt, R.H., Anthofer, V., Auer, S., Başkaya, S., Bischof, C. and **Bronger, T.** (2020) *NFDI4Ing - the National Research Data Infrastructure for Engineering Sciences*. Available at: https://doi.org/10.5281/ZENODO.4015201 (Accessed: 15 June 2025).

Silver, G.A., Miller, J.A., Hybinette, M., Baramidze, G. and **York, W.S.** (2011) 'DeMO: An ontology for discrete-event modeling and simulation', *SIMULATION*, 87, pp. 747–773. Available at: https://doi.org/10.1177/0037549710386843

TIB, Leibniz (2022) *TIB terminology service*. Available at: https://terminology.tib.eu/ (Accessed: 1 June 2025).

Turnitsa, C., Padilla, J.J. and **Tolk, A.** (2010) 'Ontology for modeling and simulation', *Proceedings of the 2010 Winter Simulation Conference*. Baltimore, MD, 5–8 December. Piscataway, NJ: IEEE, pp. 643–651. Available at: https://doi.org/10.1109/WSC.2010.5679124

Yao, L., Divoli, A., Mayzus, I., Evans, J.A. and **Rzhetsky, A.** (2011) 'Benchmarking ontologies: Bigger or better?', *PLoS Computational Biology*, 7, p. e1001055. Available at: https://doi.org/10.1371/journal.pcbi.1001055

TO CITE THIS ARTICLE:
Noman, HM and Selzer, M 2025 Enhancing Multiscale Simulation Data Management with Domain Ontologies and an ELN: Addressing Challenges and Implementing Strategies. *Data Science Journal*, 24: 28, pp. 1–22. DOI: https://doi.org/10.5334/dsj-2025-028

Submitted: 25 November 2024
Accepted: 15 September 2025
Published: 30 September 2025

COPYRIGHT:
© 2025 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/4.0/.

Data Science Journal is a peer-reviewed open access journal published by Ubiquity Press.