Dreger *et al. J Mater Inf* 2023;3:2
DOI: 10.20517/jmi.2023.01

Journal of Materials Informatics

Research Article

Open Access

Synergizing ontologies and graph databases for highly flexible materials-to-device workflow representations

Max Dreger[1,2], Mohammad J. Eslamibidgoli[1,2], Michael H. Eikerling[1,2,3], Kourosh Malek[1,2,*]

[1]Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, Jülich 52425, Germany.
[2]Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Lab for Energy Materials, Forschungszentrum Jülich, Jülich 52425, Germany.
[3]Chair of Theory and Computation of Energy Materials, Faculty of Georesources and Materials Engineering, RWTH Aachen University, Aachen 52062, Germany.

[*]**Correspondence to:** Dr. Kourosh Malek, Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, Wilhelm-Johnen-Straße, Jülich 52425, Germany. E-mail: k.malek@fz-juelich.de

How to cite this article: Dreger M, Eslamibidgoli MJ, Eikerling MH, Malek K. Synergizing ontologies and graph databases for highly flexible materials-to-device workflow representations. *J Mater Inf* 2023;3:2. https://dx.doi.org/10.20517/jmi.2023.01

Received: 12 Jan 2023 **First Decision:** 8 Feb 2023 **Revised:** 17 Feb 2023 **Accepted:** 27 Feb 2023 **Published:** 6 Mar 2023

Academic Editors: Xingjun Liu, Yi Liu **Copy Editor:** Ke-Cui Yang **Production Editor:** Ke-Cui Yang

Abstract

The escalating adoption of high-throughput methods in applied materials science dramatically increases the amount of generated data and allows for the deployment and use of sophisticated data-driven methods. To exploit the full potential of these accelerated approaches, the generated data need to be managed, preserved and shared. The heterogeneity of such data calls for highly flexible models to represent the data from fabrication workflows, measurements and simulations. We propose the use of a native graph database to store the data instead of relying on rigid relational data models. To develop a flexible and extendable data model, we create an ontology that serves as the blueprint of the data model. The Python framework Django is used to enable seamless integration into the virtual materials intelligence platform VIMI. The Django framework relies on the Object Graph Mapper neomodel to create a mapping between database classes and Python objects. The model can store the whole bandwidth of the data from fabrication to simulation data. Implementing the database into a platform will encourage researchers to share data while profiting from rich and highly curated data to accelerate their research.

Keywords: FAIR, energy materials, fabrication workflow optimization, ontologies, graph databases

INTRODUCTION

Accelerating the development of clean energy devices is pivotal for the energy transition. A significant proportion of development efforts in this realm is devoted to the complex materials, such as electrocatalysts, multifunctional electrodes and ionic and porous transport media. Integrating these materials into devices necessitates a symbiotic combination of their properties to achieve the target device functionality. The intertwined requirements define a need for energy materials to be comprehensively and thoroughly screened, characterized and fabricated. Consequently, this calls for the development and implementation of a holistic and seamless platform to manage and analyze the rapidly growing datasets along the materials-to-device development workflow[1].

High-throughput methods in materials research have rapidly evolved in recent years and are expected to greatly speed up the rate at which novel materials are developed[2-5]. The screening for new energy materials and the optimization of manufacturing processes are already being conducted using high-throughput computation (HTC) and high-throughput experimentation (HTE). HTC is enabled by the steep growth in computing power along with the robust and efficient implementation of physics-based models[6-7]. HTC paves the way for the automated large-scale screening of materials with the desired combinations of properties[8-13]. HTE allows many experiments to be conducted in parallel, thereby enabling fast materials screening[14,15]. The further acceleration of scientific research can be achieved by automating fabrication workflows. Material acceleration platforms (MAPs) seek to enable closed-loop development by performing HTE in a fully autonomous fashion[16,17]. Most MAPs are deployed to optimize a set of materials or workflow parameters with respect to predefined target properties[18-21]. A high degree of autonomy calls for a sophisticated computational backend where data from previous fabrication cycles must be extracted and used to design the next cycle on-the-fly. Bayesian optimization is the most commonly used method in closed-loop experimentation[18]. These techniques generate a large amount of data along the materials development pipeline, thereby necessitating the need for efficient data management strategies[22,23].

The standard approach to data management is to build rudimentary data infrastructures suited to the needs of a particular project. This method is effective for studies of limited scope, where the collected data can be used for incremental improvements on specific materials classes or target applications. Since the data lack standardization, it becomes, however, challenging to compare them from various sources or to reuse data for other purposes with similar scope. An effective data management system should adhere to the FAIR principles, i.e., the data should be findable, accessible, interoperable and reusable[24,25]. Data FAIRification improves the reproducibility of scientific results and makes data accessible to the whole research community. Thus, FAIR data management enlarges the pool of available highly curated data and allows the application of a wider variety of data-driven methods[26,27]. Standardized formats that originate from following the FAIR principles require less processing to make data machine-readable and AI-ready[27].

In materials research and other fields, database projects have emerged to collect and manage increasing amounts of research data. Data types represented by these projects tend to be specific, with examples including the Materials Project[28], a database containing materials simulation data, and the Cambridge Crystallographic Data Centre (CCDC)[29], which gathers crystallographic data on materials. Enabling the full potential of data-driven approaches for accelerated materials discovery requires databases that include not only simulation data but also suitable fabrication and characterization data.

A suitable database must be flexible to effectively represent heterogenous data that contain insights into fabrication processes, measurements and simulations in materials research. Furthermore, materials, components and devices need to be described on multiple spatial and temporal scales. A database that is capable of storing data with such bandwidth could be the foundation of platforms that can orchestrate and accelerate materials research. These platforms can assist in the screening of materials, experimental design, the optimization of workflows and the orchestration of devices within self-driven labs. Our efforts in this regard tie in with the recently developed VIMI platform and offer data management for simulation and fabrication data, providing data-driven analytics, accelerated characterization and computer-aided materials design to its users[1].

In this communication, we present a flexible data management approach for energy materials platforms to accelerate the search for advanced materials by exploiting the full potential of data-driven research. Our approach contains the following:

- An extension of the European Materials Modelling Ontology (EMMO) to create a standardized representation of the energy materials domain for mandating FAIR data generation.

- A new graph data model based on the classes manufacturing, measurement, matter and property, as well as the relations between them. The data model provides an intelligible, flexible and extendable representation of fabrication workflows, measurements and simulation data.

- Data storage within the native graph database neo4j for efficient access to its highly connected content.

- An encapsulation of the database in a Django framework to allow a straightforward integration into VIMI or other platforms.

- A mapping from objects within the database to Python classes *via* an Object Graph Mapper (OGM).

METHODOLOGY

Ontology

Ontologies are a formal representation of knowledge that connect various metadata and make them machine-readable[30]. As shown in Figure 1, an ontology employs classes, which can have properties stored as key-value pairs. The classes within an ontology are connected *via* relationships and rules and constraints can be specified. Ontologies are crucial for representing domain knowledge in various scientific fields and provide the basis for data and knowledge exchange among researchers within a specific domain. Ontologies are particularly useful in the context of FAIR data generation since they standardize the knowledge representation of a domain.

The interdisciplinary nature of materials science renders the standardization of information imperative for communication. The European Materials Modelling Council[31] implements EMMO[32], which is a versatile ontology for materials sciences. EMMO consists of three levels. The top level defines real-world objects and introduces "perspectives" to reflect their pluralistic nature (e.g., materials can be defined *via* their composition or function). The middle level contains specific perspectives that make EMMO applicable to various domains. Each of the perspectives represents a different branch that defines objects from a holistic, physicalistic, semiotic or mereotopological perspective.

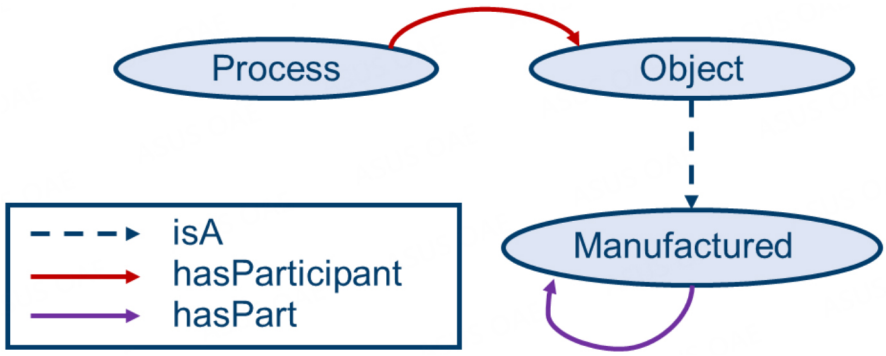

Figure 1. A minimalistic example of an ontology. Instances of the Process class are connected to Object instances *via* the has Participant relationship. The Manufactured class is a child class of the Object class, meaning that all properties and relationships can be inferred from Object to Manufactured. A Manufactured object is composed of other Manufactured parts. This mereological perspective is represented by the hasPart relationship.

The different perspectives enable EMMO to represent the fabrication, characterization and simulation of materials up to the device scale on a very general level, as illustrated in Figure 2. The bottom level contains ontologies of specific materials science domains. Other projects that require domain-specific ontologies can extend the bottom level of EMMO, while the higher levels of EMMO offer a ruleset that can be inferred from these extensions. Projects that use EMMO-based ontologies include NOMAD[33,34], CHAMEO[35] and BigMap[36,37]. Building on EMMO to introduce an ontology substantially simplifies its creation since the basics for most application domains are already contained in the EMMO. An EMMO-based ontology can therefore rely on a variety of existing relationships/classes and constraints. Extending the EMMO branches allows properties, relationships and constraints from the parent classes, e.g., Manufactured is a subclass of Object, to be inferred and it therefore also has the hasParticipant relationship to Process classes [Figures 1 and 2]. Creating an ontology relying on EMMO also improves interoperability with other EMMO-based ontologies, since they have the same structure and are following the same basic rule sets. In many aspects, ontologies are comparable to languages, centered around narrow domains, and like languages, the power of an ontology strongly relies on its level of adoption.

Database

At the core of the data infrastructure, a database is required to store fabrication, simulation and measurement data in a FAIR format. Furthermore, the metadata need to be sufficient to ensure the reproducibility of each entry in the database. Relational databases, such as MySQL or Postgres, are the current industry standards[38,39]. These databases contain tables, each of them representing a class of real-world objects, e.g., materials or measurements. Each row within a table refers to a specific instance of these real-world objects, e.g., a specific material. Relationships between different objects are represented by joining tables with foreign keys. Relational databases excel when highly structured and sparsely connected data need to be stored. However, the usage of foreign keys makes processing these relationships slow and their table structure makes the data model rigid[40]. The differences in performance between relational databases and NoSQL databases have been investigated and benchmarked previously[41,42].

There is also a wide variety of NoSQL databases for the development of a more dynamic data model and these can be divided into document-oriented, key-value and graph databases. Document-oriented data storage uses documents in a specific encoding, such as XML, YAML, JSON or BSON. Each document is addressed *via* a unique key and can be related to other documents by joint keys. The difference to relational databases is the versatile structure of the documents, which allows for a more flexible data model[43]. The handling of relationships nevertheless remains inefficient, since documents are still joined *via* foreign keys,

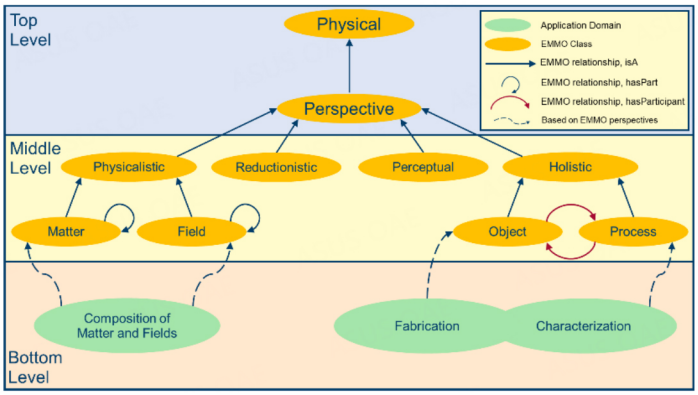

Figure 2. A minimalistic example of the overall EMMO structure. The green shapes on the bottom level represent application domains that can use and extend the respective EMMO classes in the middle level. The top level is strongly simplified since it contains further fundamental classes, relationships and constraints that the rest of the EMMO is based on.

just as in the relational data model[40]. Key-value databases store data as a single opaque collection of key-value pairs. This data structure also enables highly flexible data models, but its primitive structure needs to be extended for complex use cases[44,45].

In this work, we chose neo4j, a native graph database, for data storage. Native graph databases use graph theory to represent and store data. Graphs contain nodes that are connected *via* relationships[46]. They can be used to solve mathematical problems or to represent and store data using adjacency lists. Graphs can naturally represent complex domains inside the real world, allowing for data modelling without imposing layers of abstraction between the natural world and the associated data model. Directed graphs also contain valuable information since they can represent asymmetric relationships. The data structure of native graph databases allows for a highly efficient traversal of the graph along its relationships, which can be described as pointer hopping or dereferencing pointers. Since the adjacent nodes of each node are stored directly at the node itself, queries that require graph traversal *via* a path of relationships exhibit an O(1) complexity[40].

In neo4j, nodes and relationships can have properties stored as key-value pairs. Relationships in neo4j are always directed, which allows for the representation of asymmetric relations. These directed relationships provide more context, e.g., a simple workflow can be represented as several Process nodes, connected *via* followedBy relationships that need to be directed to fully represent the workflow. Relations between nodes of the same class are only unambiguous if they are intrinsically symmetric or if they are represented by directed relationships. Graph databases do not require a data schema since they are naturally additive, making them highly advantageous for storing heterogeneous data.

From a materials research perspective, fabrication workflows underline the heterogeneity of data in materials science research, since fabrication and characterization process data can contain a wide range of parameters, materials and subprocesses. These workflows and measurements can naturally be represented by graphs, since they are mostly sequences of subprocesses that have materials and parameters as inputs and manufactured materials or properties as outputs. A process can also be represented within a table, but it requires its structure to be predefined. If a process changes, its table must be altered or a new table must be created, leading either to massive, sparsely filled tables or many small tables representing variations of the same process. Graph databases store workflows as node sequences, one sequence for every stored workflow. Variations of these workflows do not require any changes to existing ones within the database since each workflow is stored as a separate sequence of nodes. Representing the workflows in Figure 3 within a graph

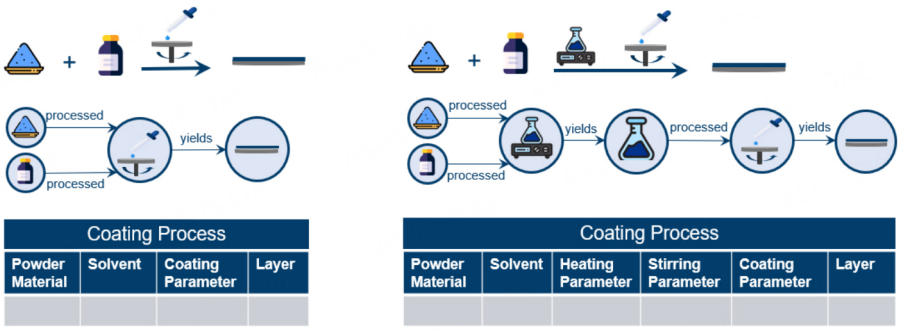

Figure 3. Schematic of fabrication workflows. The first row represents simplistic coating workflows. The workflows differ slightly since the left workflow contains an intermediate stirring and heating step. The fabrication workflows are represented as graphs and tables (bottom).

database would lead to two different sequences, which can both be accessed by queries that ask for workflows, leading from the given precursors to the corresponding product. The database is flexible in that regard since the data model does not predefine how a specific process is structured.

SYNERGIZING ONTOLOGIES AND GRAPH DATABASES

Ontology

Ontologies are naturally represented as graphs since they contain classes represented *via* relationships. The structure of an ontology makes it easily transferrable to a graph data model. The created ontology is an extension of the EMMO, which is already established as a highly sophisticated framework of classes, rules and constraints to represent materials science. The ontology with its central classes is shown in Figure 4.

The classes manufacturing and measurement both share the same parent class, Process. Matter represents all physical objects, from single atoms to manufactured components or devices. Property is a child class of PhysicalQuantity and is yielded by a Measurement process. Meta Data represents a collection of classes to make the scheme more understandable. It contains all classes that contain information stored as metadata to a specific process (e.g., measurement instruments, researchers or institutions, or experimental parameters). The Manufacturing, Measurement and Matter instances can be divided into components. The Matter instances are inputs to the Manufacturing and Measurement instances and outputs of the Manufacturing nodes. Processes can have metadata assigned to them and Measurement can have physical quantities as Property instances as outputs.

The ontology is tailored to the specific domain of energy materials by introducing domain-specific child classes. Most properties and relationships of these child classes can be inferred from the parent classes. Only specific properties and constraints need to be added to the class definition.

Currently, the EMMO extension focuses on fuel cell fabrication and characterization, but it can easily be extended to other technology domains. Classes of components, materials and fabrication procedures specific to the fuel cell domain were extracted from several well-cited fuel cell reviews. Therefore, the added taxonomy follows a widely accepted classification.

Data model

The original data model developed in this work is inspired by the open provenance model in which every simulation, measurement or fabrication process acts as a function[47]. It takes an input, e.g., a dataset or a selection of materials, to a function, e.g., a simulation or a fabrication step, and yields an output, e.g., a

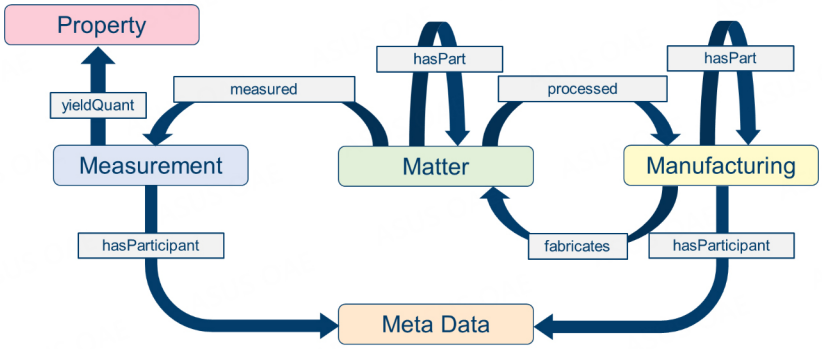

Figure 4. High-level schematic of proposed EMMO-based data model.

manufactured material or a measurement result. Due to the directed nature of the neo4j graph database, each process can naturally be represented by a tuple comprising an input node, a process node and an output node.

The overall data model follows the flowchart in Figure 4, thereby making it highly flexible and adaptable. To improve the findability of the data, we imported specific parts of the ontology into the database, namely, the classes Process, Matter and Quantity, including all their subclasses. These ontology branches represent the abstract concept of the real-world objects we aim to store in our database. The database therefore contains an ontology domain of the abstract concepts and a real-world domain that contains the actual data. Each node of the real-world domain is a specific instance of an ontology class and can be linked to a corresponding node of the ontology domain. The ontology nodes are used as labels for the nodes in the real-world domain. The connectivity of the ontology nodes creates not only a single label but also alternative labels. A real-world node that represents H_2O as a solvent would be connected to the ontology node called PolarSolvent, which is a subclass of Solvent. Therefore, H_2O is labeled with PolarSovent and Solvent can be retrieved as an alternative label [Figure 5]. These alternative labels greatly improve the findability of the data and the ontology domain can be easily extended to maintain the flexibility of the database.

Furthermore, for the holistic perspective on objects, their mereological description is crucial since materials, components and processes span multiple spatial and temporal scales. Device fabrication is a process that contains subprocesses and each of these can be split into a sequence of subprocesses [Figure 6]. The mereological representation of processes and matter allows for the representation of these objects down to an arbitrary degree of precision. This method of fractioning creates tree-structured graphs. Trees are specific graphs in which two nodes are always connected by exactly one path, making it a connected acyclic graph. Each node within a tree can have an arbitrary number of child nodes but must have only one parent node, except for the root node, which has no parent node. Each node can be treated as the root node of its subtree, thereby allowing recursion to traverse a tree.

The data model must represent experimental setups to enforce the reproducibility of the measurements and fabrication workflows. This ability will be crucial when the database is employed as part of a data infrastructure with an interface to automated labs. The scientific setup can be represented as a subgraph of connected nodes representing specific instruments/devices. Steps within fabrication workflows can then be mapped to the corresponding devices of the experimental setup, thereby allowing for precise process representations and enabling specific workflow optimization and troubleshooting. In the field of automation, challenging tasks include the orchestration of different devices and the exact positions of

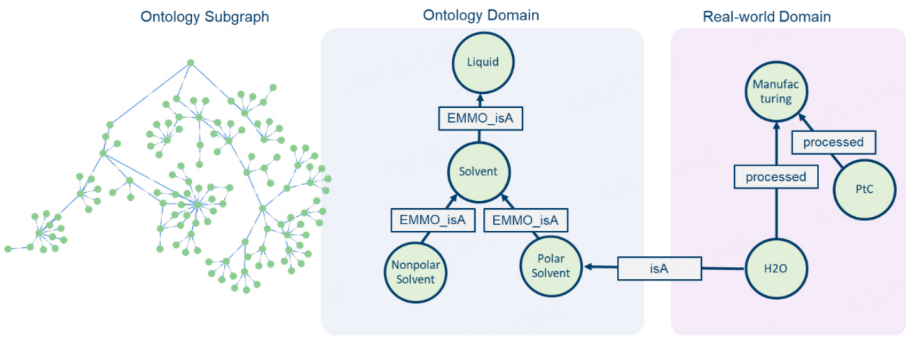

Figure 5. Screenshot of the imported Matter ontology within the database (left). Flowcharts of the ontology and real-world domain within the graph database (right) with the mapping between these domains via the isA relationship. Inheritance in the ontology domain is represented *via* the EMMO_isA relationship. The real-world data represented here shows H_2O and platinum on carbon catalyst (PtC) processed by an arbitrary Manufacturing step.

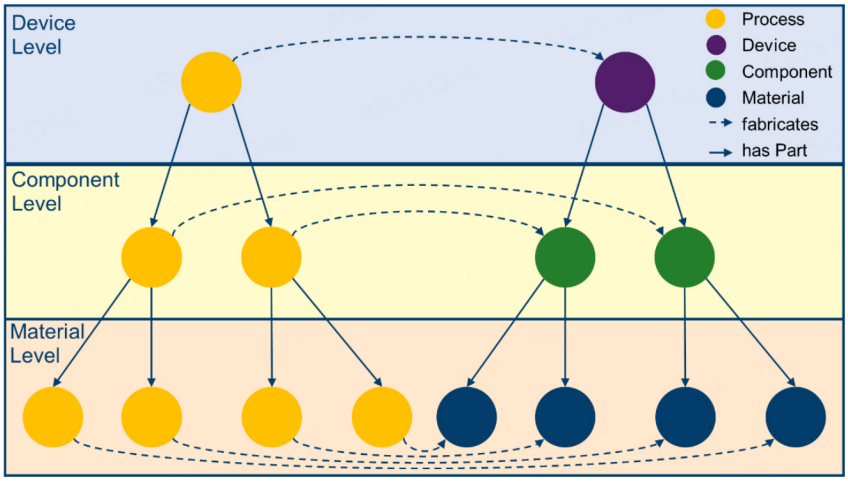

Figure 6. Schematic of the mereotopological structure of a process and its subprocesses (orange nodes) and a device, its components and their materials (purple, green and blue nodes, respectively).

samples and other moving or moveable parts of the setup at different times. This leads to the necessity to represent these workflows in extremely high granularity. The supplementary data generated by publications regarding automated labs show the level of detail that is needed to achieve automation[21,48,49]. Another important aspect of data management in automated labs is tracking the process itself in real-time[16]. Tracking and high-resolution representation could be intuitively accomplished using the proposed graph data model. Although a relational database could also satisfy these needs for a specific automated lab, it would be very complicated to use the same data model for a different setup.

The described data model is intuitive, meaning that the data model and the real-world domain are not separated by layers of abstraction that would add unnecessary complexity to the data model. It uses the middle-level classes and relationships of the EMMO ontology. Furthermore, using trees as data structures allows for the implementation of highly efficient queries along the hierarchical breakdown of objects into their components. Its performance concerning possible user requests must also be evaluated by a data model. The data model allows for intuitive querying for processing sequences or device compositions, as well as requests of fabrication workflows as sequences of input-function-output tuples.

Object graph mapper

Researchers in materials science, as users of the database, should be able to upload and access data from fabrications, measurements and simulations. The data need to be retrievable in different formats, including in the form of Python objects. The usage of layers that connect the database with the platform it is embedded in, are a common practise. Platforms that rely on relational databases use object-relational mappers to map their database object and datatypes to a given programming language, e.g., the AiiDA platform for computational materials science employs an object-relational mapper to map from their SQL database[50].

We utilize the OGM neomodel to facilitate mapping between the objects and data types within a graph database and the Python programming language[51]. The neomodel library is also available as a Django plugin, as a robust Python framework to reduce the complexity of web application creation, making the OGM an ideal key component for an upcoming API implementation due to its Python compatible interface. The OGM allows for the introduction of domain-specific Python classes (e.g., Manufacturing Process). These classes contain properties and member functions, reflecting how they are defined within the ontology. The Python inheritance rules can create a hierarchy of classes. Neomodel enables mapping of the Python classes, including their properties and data types, to the classes within the database [Figure 7].

Neomodel can also be used to generate, modify and query in a high-level interface that is agnostic to database architectural details. Using neomodel enables query formulation following the Python syntax, thereby offering a Python-based interface to the database that makes interacting with it more intuitive due to the broad acceptance of the Python programming language in the scientific community.

Neomodel allows for the creation, deletion and update of the nodes and relationships within the database. For example, it might therefore be used to update the single information of single nodes. Nevertheless, the OGM lacks efficiency for large-scale database operations. Each node/relationship that is added leads to separate requests to the database, resulting in high costs for the ingestion of large datasets. Thus, for large chunks of data, neomodel allows for custom-made cypher queries to conduct complex operations in an efficient manner. To ingest a fabrication workflow, for example, requires a predefined cypher query since such a workflow contains multiple nodes with a high degree of connectivity. Using an OGM has the advantage that complex database operations can be wrapped into Python functions that have well defined input and output interfaces. Neomodel creates an additional abstraction layer between database operations and the data management system itself. Cypher queries for complex database operations are thoroughly tested, wrapped into a Python function and are ready to be used in a Python environment, thereby hiding their underlying complexity.

Database

The proposed data model is centered around the creation of node sequences. Labels are introduced to create sets of nodes that improve the structure within the database. These labels correspond to the Python classes created using neomodel and the classes defined within the ontology. Nodes can have multiple labels, e.g., a fuel cell node might have the labels Matter, Manufactured, Device and Electrochemical Device. The choice of indexing nodes of a specific label mainly depends on the nature of queries containing that label. Each label represents an entity containing attributes as key-value pairs necessary to identify that entity (e.g., unique identifiers or the SMILES representation of a molecule). Nodes that share one label can be indexed, thereby accelerating the node retrieval of a particular label. However, indexing has the drawback of slowing down the write efficiency and requires more storage capacity.

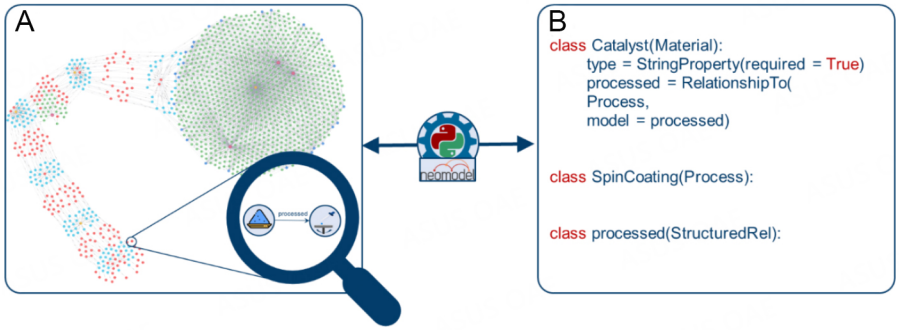

Figure 7. Nodes and relationships within the database (A) can be mapped to Python classes (B).

The property space of materials, components and devices and the parameter space of processes in materials science is high dimensional and constantly expanding. These physical quantities are stored within external Property and Parameter nodes to allow for the representation of all parameters and properties. The externalization of these nodes also allows for the efficient querying of nodes that share the same or similar properties and fabrication workflows that share the same or similar parameters. The database stores measurement results, fabrication parameters or properties in the form of scalars or a scalar array.

The highly efficient traversal of relationships is made possible by the data structure of neo4j, which has a complexity of O(1). To further increase query efficiencies, neo4j physically stores adjacent nodes close to each other and tailors the database architecture specifically to frequently used queries. The caching during queries also improves the efficiency of reading, writing and matching commands.

Large binary data types, such as image data from imaging techniques, will be stored within an external file server and are only referenced as metadata for properties derived from the images (e.g., size distribution). Referencing of the image files is carried out *via* UML links. Externalizing large data types, such as images and videos, is a good practice for data modelling. It keeps the image files connected to their corresponding data while queries do not have to handle clunky image data chunks, thereby improving the overall performance of the database. Furthermore, keeping binary data chunks within the database does not yield benefits since the database itself cannot query, index or compare binary data.

The data management system will be implemented into the VIMI platform[1], so that researchers and users from industry can upload their data *via* well-defined interfaces, e.g., the dragging and dropping of CSV files. Furthermore, cooperations with automated labs are in place for which custom tailored APIs will be created to streamline their generated data directly to the database.

Data representation

To test the data model, a batch of data for the fabrication and characterization of fuel cells was stored. The batch spans data from the materials to the device including the data from various measurements on different length scales. The heterogeneity of the data and its high dimensionality makes them ideal for testing the proposed data model.

The data were ingested into the database *via* CSV files and represented following the proposed data model. Figure 8 shows the fabrication data and how they are stored in the graph database. Figure 8A shows a single fabrication workflow from the materials to the fuel cell device, and Figure 8B presents 25 fabrication workflows within the database. The screenshots were taken from the neo4j browser interface, which offers visual representations of the stored data.

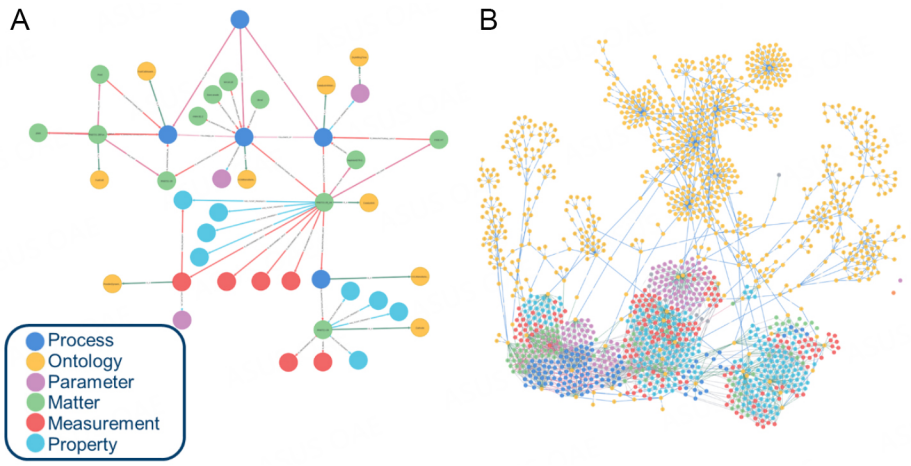

Figure 8. Representation of a single fabrication workflow from the starting materials to the fuel cell device (A) and 25 fabrication workflows and how they are linked to the ontology domain (yellow) within the database (B).

Ingesting sample data shows that the proposed data model is indeed able to represent complex fabrication workflows with a degree of detail. Even though the data model can represent these workflows, it is challenging to make them intuitively retrievable. The user of the data management system should be able to retrieve all variations of fabrication procedures for specific devices or components with a single command. This requires sophisticated queries scanning the database in an efficient manner to find the requested node sequences. The wider the bandwidth of the fabrication data stored within the database, the more challenging it will be to retrieve the requested data. These queries will be wrapped in Python functions and implemented into APIs to make the data accessible. We are currently cooperating with experimentalists from different parts of the energy materials domain to enrich our database, test our data model and especially improve and challenge our queries.

CONCLUSIONS

We propose a new data model as the basis of a highly adaptable data infrastructure for the fabrication, measurements and simulations of energy materials. The data model is designed to represent workflows and processes at an arbitrary level of complexity. It can be modified to incorporate new materials, components or processes. The hydrogen technology domain, with an emphasis on fabrication and characterization, is represented in the data model by introducing an EMMO-based ontology. The data are stored in the native graph database neo4j and its structure allows for the efficient traversal of fabrication processes. To further increase efficiency, tree data structures can be used to represent the fabrication workflows in their subprocesses and the dissection of devices, components or materials into their constituent parts.

A use case for the proposed data management system is automated labs since they require automated data management. Current automated labs usually create data management systems tailored to their specific labs, thus, each automated lab requires a new data management system and a new data model. This creates additional overhead when these labs are set up, and it leads to small unconnected data lakes that lack standardization. The proposed data model is an answer to the growing number of automated labs and their need for data management since it can represent given workflow in an arbitrary level of detail.

For more advanced approaches to data-driven workflow optimization, it is essential to include data FAIRification in experimental workflows. In particular, the data generated in fabrication processes lack FAIR features due to their heterogeneous nature and the absent workflows for standardization. Storing data that are both FAIR and suitable for AI-driven models is possible by abandoning the relational data model and transit to the flexible, graph-based data model. Access to FAIR experimental data further pave the way for data-driven techniques.

The next phases of this project involve the integration of graph databases into the VIMI platform. This will make the database accessible to other users and the data infrastructure will be used to streamline data into the generation of training datasets and the creation of machine learning models. Since the data model is based on the concepts of the EMMO, it can represent characterization and fabrication of other domains in materials science. Its flexibility also allows for applications in other related research domains, such as batteries or solar cells.

DECLARATIONS

Acknowledgements

The authors gratefully acknowledge the cooperation of Prof. Jasna Jankovic (UCONN, USA) and Fabian Tipp (Forschungszentrum Jülich) for providing fabrication and simulation data. The authors also gratefully acknowledge the Gauss Centre for Supercomputing e.V. (http://www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

Authors' contributions

Performed the research and drafted the manuscript: Dreger M
Revised and finalized the manuscript: Dreger M, Eslamibidgoli MJ, Eikerling MH, Malek K

Availability of data and materials

The EMMO-based ontologies as well as the project itself are available at https://github.com/MaxDreger92/MatGraphAI.

Financial support and sponsoring

The authors acknowledge the financial support from the Federal Ministry of Science and Education (BMBF) under the German-Canadian Materials Acceleration Centre (GC-MAC) grant number 01DM21001A and the financial support from the HiTEC graduate school for doctoral candidates at the Forschungszentrum Jülich.

Conflict of interest

All authors declare that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Malek A, Eslamibidgoli MJ, Mokhtari M, Wang Q, Eikerling MH, Malek K. Virtual materials intelligence for design and discovery of advanced electrocatalysts. *Chemphyschem* 2019;20:2946-55. DOI PubMed
2. Maier WF, Stöwe K, Sieg S. Combinatorial and high-throughput materials science. *Angew Chem Int Ed Engl* 2007;46:6016-67. DOI PubMed
3. Xiang XD, Sun X, Briceño G, et al. A combinatorial approach to materials discovery. *Science* 1995;268:1738-40. DOI
4. Koinuma H, Takeuchi I. Combinatorial solid-state chemistry of inorganic materials. *Nat Mater* 2004;3:429-38. DOI PubMed
5. Spivack JL, Cawse JN, Whisenhunt DW, et al. Combinatorial discovery of metal co-catalysts for the carbonylation of phenol. *Appl Catal A-Gen* 2003;254:5-25. DOI
6. Curtarolo S, Hart GL, Nardelli MB, Mingo N, Sanvito S, Levy O. The high-throughput highway to computational materials design. *Nat Mater* 2013;12:191-201. DOI PubMed
7. Martin R. Electronic structure: basic theory and practical methods. Cambridge University Press; 2004.
8. Ceder G, Chiang Y, Sadoway DR, Aydinol MK, Jang Y, Huang B. Identification of cathode materials for lithium batteries guided by first-principles calculations. *Nature* 1998;392:694-6. DOI
9. Jóhannesson GH, Bligaard T, Ruban AV, Skriver HL, Jacobsen KW, Nørskov JK. Combined electronic structure and evolutionary search approach to materials design. *Phys Rev Lett* 2002;88:255506. DOI PubMed
10. Stucke DP, Crespi VH. Predictions of new crystalline states for assemblies of nanoparticles: perovskite analogues and 3-D arrays of self-assembled nanowires. *Nano Lett* 2003;3:1183-6. DOI
11. Curtarolo S, Morgan D, Persson K, Rodgers J, Ceder G. Predicting crystal structures with data mining of quantum calculations. *Phys Rev Lett* 2003;91:135503. DOI PubMed
12. Morgan D, Ceder G, Curtarolo S. High-throughput and data mining with ab initio methods. *Meas Sci Technol* 2005;16:296-301. DOI
13. Curtarolo S, Morgan D, Ceder G. Accuracy of ab initio methods in predicting the crystal structures of metals: a review of 80 binary alloys. *Calphad* 2005;29:163-211. DOI
14. Boussie TR, Diamond GM, Goh C, et al. A fully integrated high-throughput screening methodology for the discovery of new polyolefin catalysts: discovery of a new class of high temperature single-site group (IV) copolymerization catalysts. *J Am Chem Soc* 2003;125:4306-17. DOI PubMed
15. Potyrailo RA, Chisholm BJ, Morris WG, et al. Development of combinatorial chemistry methods for coatings: high-throughput adhesion evaluation and scale-up of combinatorial leads. *J Comb Chem* 2003;5:472-8. DOI PubMed
16. Wagner J, Berger CG, Du X, Stubhan T, Hauch JA, Brabec CJ. The evolution of materials acceleration platforms: toward the laboratory of the future with AMANDA. *J Mater Sci* 2021;56:16422-46. DOI
17. Aspuru-Guzik A, Persson K. Materials acceleration platform: accelerating advanced energy materials discovery by integrating high-throughput methods and artificial intelligence. In. Mission Innovation; 2018.
18. Attia PM, Grover A, Jin N, et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature* 2020;578:397-402. DOI PubMed
19. Tabor DP, Roch LM, Saikin SK, et al. Accelerating the discovery of materials for clean energy in the era of smart automation. *Nat Rev Mater* 2018;3:5-20. DOI
20. Burger B, Maffettone PM, Gusev VV, et al. A mobile robotic chemist. *Nature* 2020;583:237-41. DOI PubMed
21. MacLeod BP, Parlane FGL, Morrissey TD, et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Sci Adv* 2020;6:eaaz8867. DOI PubMed PMC
22. Jose R, Ramakrishna S. Materials 4.0: materials big data enabled materials discovery. *Appl Mater Today* 2018;10:127-32. DOI
23. Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M. Big data of materials science: critical role of the descriptor. *Phys Rev Lett* 2015;114:105503. DOI PubMed
24. Stall S, Yarmey L, Cutcher-Gershenfeld J, et al. Make scientific data FAIR. *Nature* 2019;570:27-9. DOI
25. Draxl C, Scheffler M. NOMAD: The FAIR concept for big data-driven materials science. *MRS Bull* 2018;43:676-82. DOI
26. DeCost BL, Hattrick-Simpers JR, Trautt Z, Kusne AG, Campo E, Green ML. Scientific AI in materials science: a path to a sustainable and scalable paradigm. *Mach Learn Sci Technol* 2020;1:033001. DOI PubMed PMC
27. Scheffler M, Aeschlimann M, Albrecht M, et al. FAIR data enabling new horizons for materials research. *Nature* 2022;604:635-42. DOI PubMed
28. Jain A, Ong SP, Hautier G, et al. Commentary: the materials project: a materials genome approach to accelerating materials innovation. *APL Mater* 2013;1:011002. DOI
29. Allen FH, Bellard S, Brice MD, et al. The Cambridge crystallographic data centre: computer-based search, retrieval, analysis and display of information. *Acta Crystallogr B Struct Sci* 1979;35:2331-9. DOI
30. Guarino N, Oberle D, Staab S. What is an ontology? In: Staab S, Studer R, editors. Heidelberg: Springer; 2009. pp.1-17.
31. Adamovic N, Asinari P, Goldbeck G, et al. European materials modelling council. In Proceedings of the 4th World Congress on Integrated Computational Materials Engineering (ICME 2017): Springer; 2017, pp. 79-92. DOI
32. Council EMM. Report on workshop on interoperability in materials modelling; 2017.
33. Draxl C, Scheffler M. The NOMAD laboratory: from data sharing to artificial intelligence. *J Phys Mater* 2019;2:036001. DOI
34. Lenz-Himmer MO, Ghiringhelli L, Baldauf C, Scheffler M. Ontologies in computational materials science. *APS March Meeting*

Abstracts 2021;2021:X61-002.
35. Nostro P, Goldbeck G, Toti D. CHAMEO: an ontology for the harmonisation of materials characterisation methodologies. *Appl Ontol* 2022;17:401-21. DOI
36. Simon Clark FLB. Available from: https://github.com/BIG-MAP/BattINFO [Last accessed on 2 Mar 2023].
37. Deng Z, Kumar V, Bölle FT, et al. Towards autonomous high-throughput multiscale modelling of battery interfaces. *Energy Environ Sci* 2022;15:579-94. DOI
38. Maier D. The theory of relational databases. Rockville: Computer Science Press; 1983.
39. Halpin T, Morgan T. Information modeling and relational databases. Morgan Kaufmann; 2010.
40. Robinson I, Webber J, Eifrem E. Graph databases: new opportunities for connected data. O'Reilly Media, Inc.; 2015.
41. Holzschuher F, Peinl R. Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j. Proceedings of the Joint EDBT/ICDT 2013 Workshops; 2013. pp. 195-204. DOI
42. Vicknair C, Macias M, Zhao Z, et al. A comparison of a graph database and a relational database: a data provenance perspective. Proceedings of the 48th annual Southeast regional conference; 2010. pp. 1-6. DOI
43. Banker K, Garrett D, Bakkum P, Verch S. MongoDB in action: covers MongoDB version 3.0. Simon and Schuster; 2016.
44. Seeger M, Ultra-Large-Sites S. Key-value stores: a practical overview. Available from: https://blog.marc-seeger.de/assets/papers/Ultra_Large_Sites_SS09-Seeger_Key_Value_Stores.pdf [Last accessed on 2 Mar 2023].
45. Idreos S, Callaghan M. Key-value storage engines. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data; 2020. pp. 2667-72. DOI
46. Berge C. The theory of graphs. Courier Corporation; 2001.
47. Moreau L, Freire J, Futrelle J, Mcgrath RE, Myers J, Paulson P. The open provenance model: an overview. In: Freire J, Koop D, Moreau L, editors. Provenance and annotation of data and processes. Berlin: Springer Berlin Heidelberg; 2008. pp. 323-6. DOI
48. MacLeod BP, Parlane FGL, Rupnow CC, et al. A self-driving laboratory advances the Pareto front for material properties. *Nat Commun* 2022;13:995. DOI PubMed PMC
49. Rooney MB, Macleod BP, Oldford R, et al. A self-driving laboratory designed to accelerate the discovery of adhesive materials. *Digital Discovery* 2022;1:382-9. DOI
50. Pizzi G, Cepellotti A, Sabatini R, Marzari N, Kozinsky B. AiiDA: automated interactive infrastructure and database for computational science. *Comput Mater Sci* 2016;111:218-30. DOI
51. Edwards R. Neomodel documentation. Available from: https://neomodel.readthedocs.io/en/latest/ [Last accessed on 2 Mar 2023].