Ontologies for Digital Twins in Battery Manufacturing: Re-use and Development

Silvia Chiacchiera[1,*], Martin Gebser[2,*], Gerhard Goldbeck[3,*], Daniele Toti[3,4,*], Fadi Al Machot[5], Maria Bashir[5,†], Mohamed El Bahnasawi[2], Ivano E. Castelli[6], Sara Drvarič Talian[7], Palina Dubatouka[2], Nicole Fleck[8], Liyuan Liu[8], Carlos Nieto-Draghi[9], Martin Petit[10], Liyanapathiranage Sudeepika Wajirakumari Samarathunga[5], Simon Stephan[11], Ilian T. Todorov[1], Noel Vizcaino[1], Xueqi Zhang[11] and Martin T. Horsch[1,5,*]

[1]*Scientific Computing, UKRI Science and Technology Facilities Council, Daresbury Laboratory, Sci-Tech Daresbury, Warrington WA4 4AD, United Kingdom*

[2]*Department of Artificial Intelligence and Cybersecurity, University of Klagenfurt, Klagenfurt, Austria*

[3]*Goldbeck Consulting Ltd, St John's Innovation Centre, Cambridge CB4 0WS, Cambridge, United Kingdom*

[4]*Faculty of Mathematical, Physical and Natural Sciences, Catholic University of the Sacred Heart, 25133 Brescia, Italy*

[5]*Department of Data Science, Norwegian University of Life Sciences, P.O. Box 5003 REALTEK, 1432 Ås, Norway*

[6]*Department of Energy Conversion and Storage, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

[7]*Department of Materials Chemistry, National Institute of Chemistry, Hajdrihova 19, Ljubljana 1000, Slovenia*

[8]*Centre for Process Innovation, Darlington DL1 1GL, United Kingdom*

[9]*IFP Energies nouvelles, 1–4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France*

[10]*IFP Energies nouvelles, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize, France*

[11]*Laboratory of Heat and Mass Transfer, Otto von Guericke University, Magdeburg, Germany*

Abstract

Within the Horizon Europe project BatCAT (Battery Cell Assembly Twin), a set of ontologies has been developed to cover use cases on vanadium redox-flow batteries and lithium-ion batteries. Users of our tools will be practitioners and operators in manufacturing, modelling and characterization, as well as business managers. The ontologies in OWL will be the basis for conceptualization at lower and higher logical expressivity (as the YAML data model used within the LinkAhead database and the answer set programming axioms for logic-based optimization). The ontologies will support data ingest and retrieval and semantic interoperability within BatCAT tools and with external components. Here we provide an overview of these ontologies, and of their alignment or mapping to related ones (including EMMO and a number of domain ontologies from its ecosystem, such as those previously known as BattINFO). Finally we present work on the translation of the ontologies (axioms and facts) from OWL to answer set programming.

Keywords

Answer set programming, Applied ontology, Battery manufacturing, Digital twin, Ontology alignment

1. Introduction

BatCAT[1][1] considers three major and interconnected aspects of batteries' life cycle: Battery design, battery manufacturing, and battery operation ("in-use"). The central focus of the project is providing a

Proceedings of FOIS 2026 Satellite events co-located with the 16th International Conference On Formal Ontology in Information Systems (FOIS 2026), 21–25 September 2026, Vitória, Brazil

*Corresponding authors.

†Current address: CESSDA ERIC Research Services, Bergen 5058, Norway

silvia.chiacchiera@stfc.ac.uk (S. Chiacchiera); martin.gebser@aau.at (M. Gebser); gerhard@goldbeck-consulting.com (G. Goldbeck); daniele.toti@unicatt.it (D. Toti); martin.thomas.horsch@nmbu.no (M. T. Horsch)

0000-0003-0422-7870 (S. Chiacchiera); 0000-0002-8010-4752 (M. Gebser); 0000-0002-9668-6961 (G. Goldbeck); 0000-0002-4181-2852 (D. Toti); 0000-0002-1239-9261 (F. Al Machot); 0009-0000-5452-4764 (M. Bashir); 0000-0001-9600-3182 (M. El Bahnasawi); 0000-0001-5880-5045 (I. E. Castelli); 0000-0003-1355-2843 (S. Drvarič Talian); 0009-0000-0216-754X (P. Dubatouka); 0000-0001-5956-9259 (C. Nieto-Draghi); 0000-0002-4226-6689 (M. Petit); 0009-0006-6799-7193 (L. S. W. Samarathunga); 0000-0002-4578-3569 (S. Stephan); 0000-0001-7275-1784 (I. T. Todorov); 0000-0001-7608-9484 (N. Vizcaino); 0000-0002-9464-6739 (M. T. Horsch)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]https://www.batcat.info/

CEUR-WS.org/Vol-4342/paper1.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-27

digital twin of battery manufacturing, integrating (physics-based and data-driven) modelling, characterization and sensor data. Two very different types of batteries are considered, vanadium redox-flow batteries (RFBs) and lithium-ion batteries (LiBs).

The main user-facing components will be a knowledge base for batteries, digital twins of both the manufacturing line and batteries in-operando, and an interpretable industrial decision support system (IIDSS). These in turn rely on physics-based and data-driven models, validated and parameterized by characterization data.

Within the IIDSS module, two major types of optimization will be used: Multi-criteria optimization (MCO) and answer set programming (ASP). The first one is suited for numerical optimization, whereas the second targets problems formulated in logical terms, with discrete variables and in a finite domain (*e.g.* task scheduling, resource allocation, procedural compliance). For a high-level overview of BatCAT architecture, including the relation between its various components, see [2] (cf. Fig. 4.1 therein).

A requirements analysis was performed [3] leading to a collection of user stories and a first version of competency questions [2].[2] To cover all needed concepts, we identified a set of ontology modules, that have been developed [5]: (a) Higher level; (b) Characterization; (c) Manufacturing; (d) Modelling and simulation; (e) Data and metadata; (f) Optimization, decision support, design; (g) Twinning; (h) Battery (LiB and RFB). In parallel, we have evaluated coverage by relevant pre-existing ontologies and standards: these have been used to design mappings and alignments.

In this manuscript, which builds on [5], we first briefly present the LBCO-Lightweight BatCAT Core Ontologies [6], including accessibility and evaluation aspects (Sec. 2). We discuss relevant previous work (Sec. 3) and considerations for reuse which shaped our approach (Sec. 4). Then we present the connection of LBCO to the EMMO – Elementary Multiperspective Material Ontology [7, 8] and its ecosystem (Sec. 5). Finally we summarize work to translate OWL axioms and facts into ASP ones, and give a concrete example in ASP (Sec. 6), before concluding in Sec. 7. While LBCO integrates knowledge aspects needed in BatCAT, its relation to other ontologies enables interoperability with results from the broader community, and its link to ASP provides a bridge to the logic-based optimization component of BatCAT architecture. As compared to our previous technical report (cf. Annex of [5]), here the content is extended and thoroughly updated.

During the course of the project, and as ontology implementation proceeded, the initial competency questions have been updated: a live version of the competency questions, up-to-date with development, and their mapping to LBCO ontology entities is available at [6] under the `doc` folder[3].

2. LBCO modules

The set of modules presented in this paper is called LBCO – Lightweight BatCAT Core Ontologies. They are written in OWL (in Turtle format), starting by design from lower logical expressivity, then increasing complexity iteratively and only if motivated by application needs. The development is done in a public GitHub repository [6]. All modules (b) - (h) are aligned with (a), however some are vertical [(b) - (d)] and other transversal [the product-agnostic (e) - (g) and the battery specific one (h)]: this means that, for example, the data module (*e.g.* describing uncertainties and data quality protocols) needs to address corresponding issues from characterization, manufacturing and modelling. In Table 1 we give a list of the modules, with their scope and aim. We also point to relevant external artefacts which overlap with them in scope and either informed the LBCO development or could be linked to (cf. Sec. 3). In Figure 1 we show the class structure of `higher`, and in Table 2 we give some example of classes from the vertical modules below it, to show how they expand the higher module, their analogies and differences.

The understanding of what a digital twin is varies across communities and authors, with a range of

[2] In [2] we collected 56 user stories grouped under 23 epics and about 50 competency questions, following the LOT methodology [4]. Also, the relations between characterization methods, characterization devices and measured quantities was given there in tabular form.

[3] https://github.com/HE-BatCAT/lbco/blob/main/doc/competency_questions.csv

LBCO module	scope / aim	relevant external asset
`higher`	abstract concepts; facilitate structuring of and connection between lower modules; facilitate alignment with TLOs	EVMPO [9]
`characterization`	off-line characterization (experiments)	CHAMEO [10]
`manufacturing`	manufacturing of products, *via* intermediate ones; includes on-line characterization (*i.e.* sensors), actuators, product/process KPIs	SSN [11]
`modelling`	computational modelling and simulation; related processes (*e.g.* parametrization and validation) and data	OSMO, VISO, VOV [9]
`data`	(meta)data aspects, such as uncertainties, data quality protocols, and metadata schemas used for annotations	DQV
`optimization`	optimization, decision support, design; optimization problems, either numerical or logic-based; to state problems and solutions	—
`twinning`	digital twinning (a particular type of modelling, where models are synchronized with their targets); digital twin functionalities	ISO 30173:2023 [12], Ref. [13]
`battery`	battery-specific information (*e.g.* battery manufacturing processes and battery product parts), for LiBs and RFBs	BDO, BVCO and GPO[14]; BPO [15]

Table 1
LBCO modules: name, scope and aim, and relevant external artefacts. These either informed the LBCO development, or are assets with a similar scope that we could map to/import from. Each module corresponds to a file in [6] called `<name>.ttl`. Ordered by: most abstract module, (three) vertical and (four) transversal ones. See also Sec. 2 and Sec. 3.

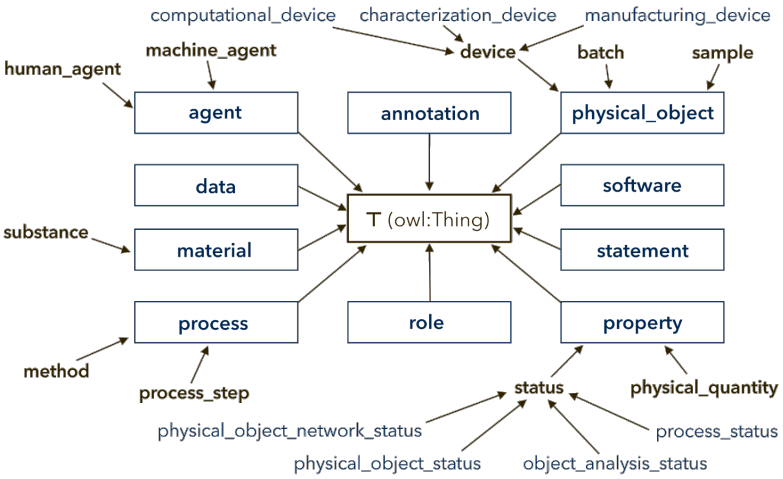

Figure 1: Classes in the LBCO module `higher.ttl`, version 1.0.2. The top classes are represented as (blue) rectangles, and the two orders of direct subclasses below them are in (black) bold text and (blue) normal text, respectively. Arrows represent sub-classing. Note: Here and in the following, classes with no prefix are LBCO classes.

definitions available, from softer to more demanding ones (see *e.g.* [16] for a comparison of definitions). In LBCO we follow the view initiated by Michael Grieves in 2002 [17] and built upon by other authors: in that view, the digital twin central elements are the digital model/representation, its target (typically physical, but not necessarily, see *e.g.* [12]), and the data/information flow between the two. In our approach, the twin target can be a `physical_object` or a `process`. When describing a digital twin, it is important to cover, among others, its functionalities (cf. `digital_twin_action_type`) and the model abstraction level (cf. `generality_level`) [13].

Overall, LBCO contains 285 classes, 47 object properties and 13 data properties. The battery module is by far the largest one, containing about 90% of the overall LBCO classes. Beside the individual modules, the repository also contains a file[4] that serves as an entry point to them all and, for convenience, a file

[4]https://github.com/HE-BatCAT/lbco/blob/main/batcat.ttl

module	process (incl. method)	device	data
char.	char. measurement, protocol and technique	char. device	sample history, char. result
man.	man. process and protocol	man. device, sensor	man. sensor data
mod.	simulation, model parametrization and validation process, computational mod. method	computational device	model (parameters), simulation result

Table 2
Example classes from the characterization, manufacturing and modelling (vertical) LBCO modules.

with all LBCO modules collated[5].

2.1. LBCO Accessibility and persistence

The LBCO ontologies, alongside their mappings discussed in this manuscript, are available in a public GitHub repository [6]. The persistent URL https://w3id.org/batcat{/} has been specifically created within the W3ID community, and redirection rules, complying with W3ID practices, have been defined to point to the relevant ontology files in [6]. There, a container file, `batcat.ttl`, serves as an entry-point for the BatCAT ontologies, and all the ontological URIs are resolvable.

2.2. LBCO evaluation

The LBCO ontologies have been evaluated using OOPS! [18] and FOOPS! [19] tools: the first one detected no critical warning, and the second one gave a score larger than 80% using the "URI" modality and the `batcat.ttl` container. Use of these tools on previous versions has helped us improve the ontology quality and FAIRness. We note that the current FOOPS! tool is a development (beta) version, so the numerical score is to be viewed in that light.[6]

3. Related work

There is a large amount of literature on digital twins in industry and on battery digitalization: we cite here only the most relevant for our work.

Specific to batteries, we highlight the Battery Domain ontology (BDO) [20] (cf. BattINFO in [14]), which includes and loads a number of battery-related ontologies from the EMMO ecosystem,[7] and the Battery Value Chain Ontology (BVCO) [14], which describes manufacturing steps and their inputs and outputs. For digital twins, the standard ISO 30173:2023 [12] provides a reference vocabulary, and Barros *et al.* [13] analyzes requirements and suggests an ontology addressing them. To describe sensors and actuators, the W3C Semantic Sensor Network (SSN) [11] is available. For characterization data, the CHAMEO ontology [10] is an ontologization of the CHADA CEN Workshop Agreement [21].

Previous work by some of us within the VIMMP - Virtual Materials MarketPlace project includes ontologies for software (VISO - VIMMP Software Ontology) and variables (VOV - VIMMP Ontology of Variables), which are tailored to materials modelling software and algorithms [9], with a focus on physics-based modelling. We note that `higher.ttl` bears similarities with the higher level module developed there (EVMPO - European Virtual Marketplace Ontology).

All of the above has informed our developments at different levels (*e.g.* coverage analysis, drawing direct connections), cf. Table 1.

[5] https://github.com/HE-BatCAT/lbco/blob/main/all_LBCO_modules_combined/all_lbco_modules_combined.ttl

[6] *E.g.* we noted an issue that we believe arises when domain and range are used in parallel with restrictions on some property, causing some misreading of the ontology and an artificial score reduction. This is to be further tested and explored, first on our side, and then in connection with the FOOPS! tool developers if needed.

[7] BattINFO [14] is an older container from the Big-MAP project. This work, led by Simon Clark, has now moved to the EMMO github Domain Battery [20]. The set of BattINFO ontologies includes Battery, Chemical Substance, Electrochemistry and CHAMEO, cf. Sec. 5.2. Note that the difference between BattINFO and the "integrated set" discussed in Sec. 5.2 is that the latter also includes the BTO (Battery Testing Ontology).

Ontologies to be considered include DQV (Data Quality Vocabulary)[8] and the very recent BPO (Battery Production Ontology) [15] (see also its Table 1 and references 49-57 therein).

3.1. Added value/benefits of LBCO

In general, the added value of the LBCO is to draw connections between different aspects of battery manufacturing, modelling and characterization, while remaining lightweight. Also, as it emerges from the needs of a project/application, it captures concrete user requirements. Below we give two examples of where LBCO adds value with respect to relevant battery-specific artefacts.

It is important to underline that the Battery Domain Ontology along with its related domain ontologies is a mature and sizeable asset, with thousands of classes, nice documentation, and part of the larger pool of EMMO-based efforts. That said, as compared to the current BDO, LBCO contains, *e.g.*, additional relations between battery-related objects and their main properties. *E.g.*, for an `electrode` these are: conductivity, exchange current rate, porosity, surface area, surface chemical composition, surface hydrophilicity, surface stability and tortuosity (cf. competency question CQ_CRO_1 at [6]).

Another relevant pattern, and for which LBCO contains additional information as compared to the current BVCO, are manufacturing process steps: in LBCO, beside input(s), output(s), also settings are defined for all the steps of the LiB manufacturing process and for selected ones for RFBs (cf. CQ_MAN_5.1 and 5.2 at [6]). *E.g.*, for `calendering`, the input is a `dried_coated_electrode`, and the output a `calendered_electrode`. The settings of this step are: calendering speed, temperature and thickness.

The first pattern serves as a pivot to connect manufacturing, characterization and modelling, via physical properties that are observed (by sensors), measured (by characterization devices) or predicted (via computational modelling methods, cf. the property `gives_modelling_value_for`). On the other side, knowing the process settings is important for process/product optimization purposes, and this is captured by the second pattern.

4. Mapping and alignment considerations for reuse

Ontology reuse [22] as well as alignment to mid- and top-level ontologies (TLOs) are far from trivial [23]. As noted in [22], the specificity of many application ontologies means that reusing an ontology may require (extensive) modifications, entailing a time-consuming effort to resolve potential issues such as disparities in structure, naming conventions and modelling choices. On the other hand, the classification of domain entities under upper ontology classes is highly inconsistent, even if carried out by experts [23].

Within the BatCAT project we used a bottom-up approach for development, and opted for a light-weight modular set of ontologies, while at the same time keeping track of potential alignments between concepts required for the BatCAT cases and those available in existing ontologies, in particular EMMO-based ontologies. This facilitated the speed of development required in a relatively small project, while keeping alignment options open. We refrained from selecting a single TLO from the outset of the development process, to avoid strong ontological commitments and allow connections to multiple TLOs. This is both convenient for us when reusing external artefacts, and in view of reuse of LBCO by other initiatives.

We hence keep the alignment to TLOs and to external assets separate from the LBCO itself. The development process proceeded in parallel with the analysis of coverage of the required concepts by existing ontologies.

Our case shows a typical situation in which 1) many different facets of a problem need to be brought together, 2) relevant artefacts exist with different levels of maturity and machine readability, and 3) agility (including the possibility to make changes and the timescales for those) are relevant.

Subsequent alignment with an established TLO (*e.g.* EMMO) and mappings to external assets enable connections to existing work. While there may be some duplication of existing work, this is minimised

[8] https://www.w3.org/TR/vocab-dqv/

and the approach has the advantage of feasibility, usability and independence from external developments. Of course, ideally, efforts at the application level should eventually contribute to (community) efforts at the domain level and above (such as BDO, EMMO): that is where the provided mappings can serve as a base to give concrete feedback to various initiatives.

Focusing on EMMO and EMMO-related domain ontologies, we provide an alignment of LBCO to EMMO (via `lbco:higher`). We also extract from EMMO-based ontologies a consistent set of battery-relevant modules, and provide mappings (*e.g.* of characterization techniques from LBCO to CHAMEO).

The alignment of the LBCO "higher" module with the EMMO discussed in the next section illustrates that (a) a strong alignment with the most widely used EMMO Middle level concepts such as Object/process and Data/information can be achieved, but also how (b) this alignment asks questions about the ontological commitment (see *e.g.* the Role alignment).

5. Alignment to EMMO and mapping to EMMO-based ontologies

5.1. Alignment of the BatCAT higher ontology with EMMO

The concepts in the BatCAT LBCO `higher` ontology module have been aligned with the EMMO 1.0.2 ontology, the latter available via https://w3id.org/emmo/1.0.2. We recall that the EMMO [8, 7] is an ontology (including a TLO part) especially relevant for the physical sciences and engineering and it has a built-in approach to accommodate multiple *perspectives*. The EMMO is self-contained and hierarchically structured, with mereo-causality as the foundation, followed by a physicalistic middle level as well as the Perspectives, including "Persistence" (object-process perspective of the world), "Holistic" (whole-role), "Contrast" (patterns, information, data), "Semiotic" (signs, interpretations) [8]. In a nutshell, by taking different, possibly combined, perspectives, the same entity can be seen in different ways. For example, a measurement can be described as a process involving the sample (object) and an instrument (object). The instrument can also be described in terms of its role (*e.g.* carrying out imaging) in the measurement. The resulting image can be seen as an object and also as a pattern (Contrast), providing data that can be interpreted (Semiotics). The alignment of LBCO higher is done to concepts in the EMMO Discipline level[9], and is based either on comparing classes definitions or their subclasses and usage.

The key EMMO concepts to which the BatCAT classes have been aligned are based on the EMMO material backbone as well as on the Persistence, Contrast, Semiotics and Holistic perspectives of EMMO, which cover almost all application cases of EMMO. They include the following key concepts:

1. Objects and Processes, which are the highest level concepts in the EMMO Persistence perspective.
2. Data and Information, which are high level concepts in the EMMO Contrast perspective.
3. Material, which belongs to the EMMO core concepts based on its materialistic foundation. We have chosen Material, rather than the more generic Substance class, as BatCAT deals with specific materials for manufacturing (*i.e.* for a certain purpose).
4. Property, which in EMMO belongs to the Semiotics Perspective.
5. Role, which can be found in the EMMO Holistic Perspective.

For the alignment, we can first of all find an equivalence between `lbco:material` and `emmo:material`, which is "an amount of ordinary matter substance (or mixture of substances) in different states of matter or phases", and "usually means some definite kind, quality, or quantity of matter, especially as intended for use".

Delving deeper into EMMO Perspectives, we identify a close alignment of the `lbco:physical_object` and `lbco:process` classes with the highest-level classes of the Persistence perspective `emmo:Object` and `emmo:Process`. In addition, the `lbco:agent` and `lbco:method` classes can be aligned with the respective EMMO concepts in that perspective. More specific details are described below.

- With the EMMO Persistence perspective, the alignment is as follows:

[9] See section 1.2.2 of https://github.com/emmo-repo/EMMO/wiki/EMMO-Introduction.

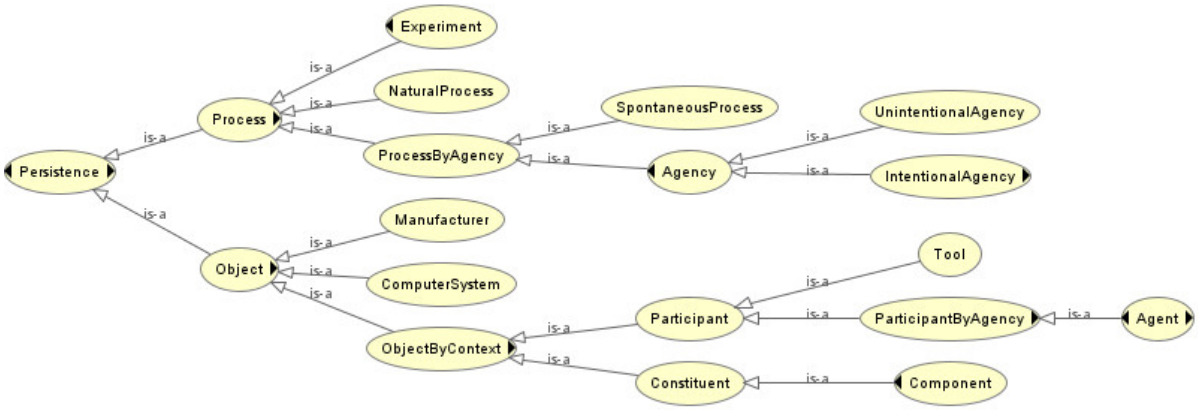

Figure 2: Extract of the EMMO [8] Persistence Perspective. This figure and the subsequent one were generated using the *protégé* tool [24]; see also https://protege.stanford.edu/.

  - `lbco:physical_object` is considered equivalent to `emmo:Object`. As a result, the subclasses of `lbco:physical_object`, *i.e.* `lbco:sample` and `lbco:device`, are also subclasses of `emmo:Object`. That is consistent with similar subclasses in EMMO and its domain ontologies. In particular, `emmo:Device` is a subclass of `emmo:ManufacturedProduct` which is a subclass of `emmo:Object`.
  - `lbco:agent` is deemed a subclass of `emmo:Agent`. The latter is elucidated as "A participant that is the driver of a process." Any such "driver", whether machine, human, animal, *etc.* is an `emmo:Agent`, whereas the `lbco:agent` is the disjoint union of human and machine agents, hence the subclass relation was chosen.
  - `lbco:process` has the same semantic meaning as `emmo:Process`. For the latter, the requirement is that at least two temporal parts must exist which are not of the same type, signifying change. Hence, the equivalence relation is used.
  - `lbco:method` — we interpret the method class in LBCO as subclass of `emmo:Procedure`, where the latter is described as "an intentional process with a plan." Procedure subclasses in EMMO include Measurement and Computation, which seem to align well with the LBCO subclasses of characterisation and modelling/simulation methods.

  The relevant EMMO class hierarchy is shown in Figure 2.
- In the EMMO Contrast perspective (which includes `emmo:Data`, `emmo:Information` and `emmo:Software`), we also identify alignments for the `lbco:data`, `lbco:statement` and `lbco:software` classes:
  - `lbco:data` is interpreted as being equivalent to `emmo:Data`.
  - `lbco:statement` and `lbco:annotation` are deemed subclasses of `emmo:Information`. This means that in an EMMO-aligned and integrated ontology, `lbco:statement` and `lbco:annotation` would become subclasses of `lbco:data` rather than sitting alongside it.
  - `lbco:software` is defined as a subclass of `emmo:Software`. Here we used the subclass relation because `lbco:software` typically refers to application programs. In contrast, `emmo:Software` also includes the source code files, *etc.*

  An image of the relevant EMMO class hierarchy is shown in Figure 3.

Finally, a more complex question arises regarding the alignment of the `lbco:role` class. The lightweight approach of LBCO makes no specific commitments to what defines a role and whether a role is permanent, or whether its instances only exist at certain times when the role is performed. In the case of `emmo:Role`, the latter involves a parthood relation with respect to an overall "whole" in which it participates. For example, hammer is a role that can be fulfilled by many objects (not just those labelled "hammer"), but

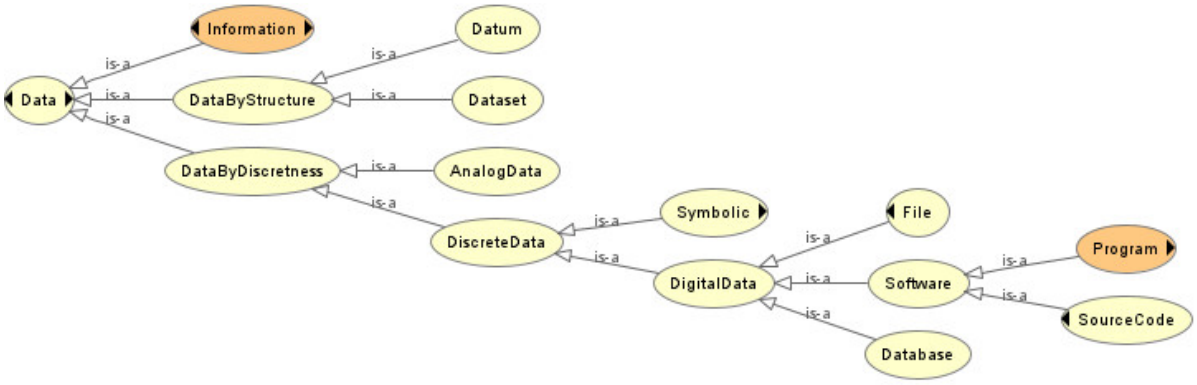

Figure 3: Extract of the EMMO [8] Contrast Perspective.

Ontology (acronym)	level	scope / aim
BDO [20, 14]	domain	describing battery systems, materials, methods, and data
CHEMS [14]	domain	formally representing chemical substances
ECHO [14]	domain	electrochemical systems, materials, experimental methods, and data
CHAMEO [10, 27]	domain	materials characterization as in CHADA template
BTO [28]	application	methodologies and techniques for battery testing

Table 3
Overview of relevant EMMO-based ontology modules, at the domain or application level. Sources of all are available at [8]. See text for the acronyms' expansion and footnote 7 about the relation to BattINFO.

always requires a process ("hammering") in which the hammer participates. Outside of this process, the "hammer" is an object. In BatCAT's case, this more restrictive "role" concept can be used to model devices that can play different roles in different contexts. The alignment file can be found in the LBCO repository `alignment` folder.

5.2. Mapping of LBCO to EMMO-based ontologies for the battery domain

Work within previous European projects, notably BIG-MAP [25] and NanoMECommons [26], has led to the creation of a set of interoperable modules for different facets of batteries and materials characterisation, respectively. These modules cover domain and application use cases and are aligned to the EMMO.

A graphical overview of such ontology modules is shown in Figure 4 and a compact summary of the domain/application modules and their scope is given in Table 3, which also includes the corresponding references. The acronyms there stand for: Battery Domain Ontology (BDO), Chemical Substance Domain Ontology (CHEMS), Electrochemistry Domain Ontology (ECHO), Characterisation Methodology Ontology (CHAMEO) and Battery Testing Ontology (BTO)[10]. Live/development versions of all these modules can be found separately in EMMO's GitHub repository [8]. An entry point into these ontology modules has been created and placed in the BatCAT Github repository's `external` folder as a reference containing a number of relevant domain concepts for further alignment and integration with BatCAT ontologies.

Comparison work has been carried out for key concepts including, for instance, substances from CHEMS as well as characterization techniques from CHAMEO. Such results have already or will be added to the LBCO repository in the future, under the `mappings` folder.

[10]Base URIs of the modules are: BDO: https://w3id.org/emmo/domain/battery; CHEMS: https://w3id.org/emmo/domain/chemical-substance; ECHO: https://w3id.org/emmo/domain/electrochemistry; CHAMEO: https://w3id.org/emmo/domain/characterisation-methodology/chameo; BTO: http://w3id.org/emmo-bto/bto.

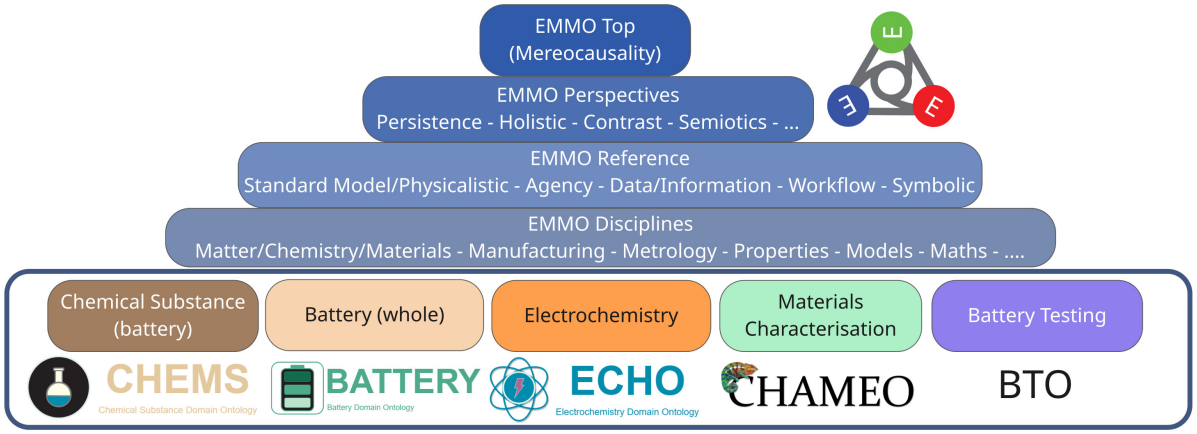

Figure 4: Graphical overview of the integrated EMMO-based ecosystem of ontologies for the battery domain [8]. See also references in Table 3. Ontology logos are from the respective sources.

5.3. Mappings of LBCO to other external ontologies: the example of GPO for manufacturing steps

During the course of the BatCAT project, we have compared our requirements with existing relevant ontologies. As a result, mappings have been produced, including, for instance, those related to manufacturing steps between LBCO and the General Process Ontology (GPO) [29] (https://gpo.ontology.link/). We remind that GPO is used by BVCO [14] as its upper level. GPO is an ontology, based on an earlier version of EMMO[11], that aims to model processes as holistic entities transforming inputs/educts (matter, energy, information) into outputs/products (matter, energy, information) with the help of tools such as devices or algorithms. For the sake of completeness, we also give a snapshot of the GPO version used for the mappings in the `external` folder of the LBCO repository. The purpose of such mappings is to enable data conversion between data annotated using LBCO and those using GPO. For the specific case of battery manufacturing steps, the LBCO contains information about input, output and settings of each step. We note that, here and in similar cases, we opted for duplicating classes to avoid dependencies on external efforts. The issue of versioning ontologies, not as a whole, but at a finer level of detail, is a known challenge, and term-level versioning would be desirable. For recent efforts addressing this challenge, see for example [30, 31, 32].

6. Mapping of LBCO extracts to answer set programming (ASP)

After presenting LBCO and its connections to other ontologies, in this section we focus on the connection to logical optimization, for ontologies in general and within BatCAT in particular. We make use of ASP as a high-level modelling language based on first-order logic for specifying combinatorial search and optimization problems [33], where ASP systems like `clingo` [34] automate the reasoning process. To this end, a problem representation is split into instance-specific facts and a general first-order encoding of the properties characterizing solutions and objectives to optimize.

For reasoning on LBCO ontologies, we apply the `solver.py` script[12] to convert LBCO modules in Turtle format to ASP facts. Our script utilizes the `rdflib` package[13] for reading OWL axioms and facts, where class, object, and property names are translated into strings to ensure compliance with the input language of `clingo`. As a prototypical example, we discuss here a *method selection problem*. This illustrates a typical situation that could manifest as part of Design of Simulation or Design of Experiment: different (experimental and modelling) methods can be used to assess certain observable quantities of interest. As the available methods differ in their attributes (*e.g.* cost, precision), optimal

[11] GPO is related to a much earlier version of EMMO, and would require some re-alignment to become part of the set in Sec. 5.2.
[12] https://github.com/HE-BatCAT/lbco/blob/main/mappings/map2asp/solver.py
[13] https://rdflib.readthedocs.io/en/stable/

```
1  % Experiment method
2  entity("GITT_TEST","method").
3  property("GITT_TEST","has_output","DIFFUSION_COEFFICIENT").
4  property("GITT_TEST","has_precision_level","HIGH").
5  property("GITT_TEST","has_cost_level","HIGH").

7  % Simulation method
8  entity("DISSIPATIVE_PARTICLE_DYNAMICS","method").
9  property("DISSIPATIVE_PARTICLE_DYNAMICS","has_output","DIFFUSION_COEFFICIENT").
10 property("DISSIPATIVE_PARTICLE_DYNAMICS","has_precision_level","LOW").
11 property("DISSIPATIVE_PARTICLE_DYNAMICS","has_cost_level","MEDIUM").
```

Listing 1: ASP facts for `lbco:method` objects describing an experiment or simulation, respectively.

```
13 % Goal and optimization targets of a method selection task
14 entity("METHOD_SELECTION","logic_based_problem").
15 entity("METHOD_SELECTION","optimization_problem").
16 property("METHOD_SELECTION","has_part","GOAL_1").
17 property("METHOD_SELECTION","has_part","TARGET_1").
18 property("METHOD_SELECTION","has_part","TARGET_2").

20 % Selected method must provide diffusion coefficient as output
21 entity("GOAL_1","qualitative_constraint").
22 property("GOAL_1","has_output","DIFFUSION_COEFFICIENT").

24 % Maximizing precision is of high priority '2'
25 entity("TARGET_1","target").
26 property("TARGET_1","has_precision_level","coarse_amount_level").
27 property("TARGET_1","has_priority",2).
28 property("TARGET_1","has_polarity",-1).

30 % Minimizing cost is of secondary priority '1'
31 entity("TARGET_2","target").
32 property("TARGET_2","has_cost_level","coarse_amount_level").
33 property("TARGET_2","has_priority",1).
34 property("TARGET_2","has_polarity",1).

36 % Mapping of course amount levels to integer weights
37 sameAs("HIGH",3). sameAs("MEDIUM",2). sameAs("LOW",1).
```

Listing 2: ASP facts describing the selection of a method to estimate an electrolyte's diffusion coefficient.

solutions will be suggested based on the user-defined criteria. In particular, we focus on choosing between methods that give access to the same physical property. For a concrete battery-related example, we consider the diffusion coefficient of an electrolyte, a parameter which is required by many models to estimate battery lifetime.

Accordingly, the relevant properties of two `lbco:method` objects are provided by the *facts* in Listing 1, obtained by the syntactic conversion of LBCO information to ASP.[14] The task of selecting an experiment (GITT_TEST in lines 2-5) or a simulation method (DISSIPATIVE_PARTICLE_DYNAMICS in lines 8-11) to estimate the diffusion coefficient of an electrolyte, with the first objective to maximize precision and the secondary target to minimize cost, is expressed by the user-supplied facts in Listing 2. They introduce METHOD_SELECTION as a logic-based optimization problem (lines 14-15) featuring one selection goal (line 16) and two optimization targets (lines 17-18). The goal is a qualitative constraint requiring the

[14]The lines 1 and 7, starting with %, provide comments that are added here to point out the intuition.

```
39 % Selected method must fulfill the goal
40 {select(METHOD) : entity(METHOD,"method"),
41                    property(METHOD,RELATION,VALUE),
42                    property(GOAL,RELATION,VALUE)} = 1
43                 :- entity(PROBLEM,"logic_based_problem"),
44                    property(PROBLEM,"has_part",GOAL),
45                    entity(GOAL,"qualitative_constraint").

47 % Selected method shall maximize or minimize (POLARITY '-1' or '1') optimization targets
48 :~ entity(PROBLEM,"optimization_problem"),
49    property(PROBLEM,"has_part",TARGET),
50    entity(TARGET,"target"),
51    property(TARGET,"has_priority",PRIORITY),
52    property(TARGET,"has_polarity",POLARITY),
53    property(TARGET,RELATION,"coarse_amount_level"),
54    property(METHOD,RELATION,VALUE), sameAs(VALUE,WEIGHT),
55    select(METHOD). [POLARITY*WEIGHT@PRIORITY]
```

Listing 3: First-order ASP encoding of method selection, subject to a goal and optimization targets.

DIFFUSION_COEFFICIENT as output of a selected method (lines 21-22), while the targets refer to the maximization of precision (lines 25-28) or the minimization of cost (lines 31-34), respectively. In order to incorporate course amount levels as penalties or rewards in `clingo`'s optimization objectives, we additionally include auxiliary facts for mapping the course amounts to integer weights (line 37).

A general first-order encoding for method selection tasks specified by facts like the above is given in Listing 3, where the (non-string) arguments written in capital letters denote universally quantified variables. The selection of a method fulfilling the qualitative constraint specified by the selection goal is subject to a *choice rule* (lines 40-45). Moreover, any number of optimization targets can be prioritized and selectively maximized or minimized by a *weak constraint* (lines 48-55), considering that the negative polarity `-1` turns penalties into rewards for the (lexicographic) minimization strategies implemented by ASP systems [35]. In our example, the superior precision of the experiment GITT_TEST in comparison to the DISSIPATIVE_PARTICLE_DYNAMICS simulation method leads to an optimal solution represented by the (variable-free) atom `select("GITT_TEST")` in the output of `clingo`.

This example is of course an intentionally simplified version of a realistic use case, with the purpose to demonstrate ASP functioning. However it captures in essence mechanisms that can be used in more general and realistic scenarios. In particular, Listing 3 constitutes a simplified version of our ASP encoding in the `rules.lp` file,[15] which additionally provides functionalities for scheduling tasks and allocating resources to experiments and simulations. While the fact representation of the LBCO ontology, including the excerpt shown in Listing 1, is automatically obtained, this does currently not apply to the facts expressing a specific problem to be solved by the ASP system, such as the ones in Listing 2. Future work includes providing a high-level interface to specify reasoning tasks, e.g., based on requirements formulated in the Shapes Constraint Language (SHACL) [36], which complements OWL and carries a closed-world view.

7. Conclusion

We have presented here the BatCAT ontologies, LBCO, and their relation to other efforts in the ontology landscape. Especially, the alignment to the EMMO has been discussed in detail. Then, work to relate the LBCO in particular and ontologies more in general to answer set programming has been summarized. A concrete example about method selection has been given, and its full implementation in ASP (including the underlying ASP rules, not shown above for simplicity) is provided in the LBCO repository, which

[15] https://github.com/HE-BatCAT/lbco/blob/main/mappings/map2asp/rules.lp

also includes the sources of the related ontology modules (in OWL, Turtle format). In passing, we have discussed and motivated our practical approach to ontology development and reuse. While the first phase of development of LBCO ontologies has concluded, they will be updated at [6] taking into account users' feedback and evolving requirements from the BatCAT project. In fact, with use and as the project advances and more components / tools need to use them, we expect new requirements will arise. Current ongoing work using LBCO includes the development of a JSON-LD export functionality for the metadata of records in the BatCAT dataspace. The ingestion, annotation and export of BatCAT datasets from manufacturing, modelling, and characterization will enable a user-centered evaluation of and valuable feedback on LBCO effectiveness at the domain level. At the same time, the most abstract modules will be reviewed and consolidated as well, taking into account feedback by ontology experts, such as from FOIS.

The LBCO is a lightweight modular set capturing user requirements from a complex application problem bringing together different views by construction. The general captured patterns can be used for digital twins in other contexts (beyond batteries), and, by leveraging the various mappings, LBCO can provide concrete feedback to other (community-driven) ontology development efforts.

Acknowledgments

This work has received funding from the European Union's Horizon Europe research and innovation programme, grant agreement no. 101137725 (BatCAT – Battery Cell Assembly Twin) and from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee, grant agreement no. 10112690 (BatCAT – Battery Cell Assembly Twin).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] BatCAT: Battery Cell Assembly Twin, 2024-2027. doi:10.3030/101137725, Horizon Europe project.

[2] S. Chiacchiera, M. T. Horsch, F. Al Machot, D. Romanov, M. Petit, C. Nieto-Draghi, E. Valseth, D. Senthilkumar, S. Drvarič Talian, B. Likozar, I. Todorov, N. Vizcaino, M. El Bahnasawi, K. Kyamakya, M. Gebser, A. Linhart, K. Roettger, R. Mitchell, D. Mehta, BatCAT Deliverable 4.1: Data landscape and infrastructure related requirements, Technical Report, BatCAT Consortium, 2024. doi:10.5281/zenodo.18059018.

[3] M. T. Horsch, D. Romanov, E. Valseth, S. Belouettar, L. E. Córdova López, J. Glutting, M. A. Janssen, P. Klein, A. Linhart, M. A. Seaton, E. Dypvik Sødahl, N. Vizcaino, S. Werth, S. Stephan, I. T. Todorov, S. Chiacchiera, F. Al Machot, Battery manufacturing knowledge infrastructure requirements for multicriteria optimization based decision support in design of simulation, in: Proceedings of the First International Workshop on Semantic Materials Science, CEUR-WS, 2024, pp. 112–119.

[4] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, LOT: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755. doi:10.1016/j.engappai.2022.104755.

[5] S. Chiacchiera, M. Gebser, G. Goldbeck, M. Petit, D. Toti, F. Al Machot, I. E. Castelli, S. Drvarič Talian, P. Dubatouka, M. El Bahnasawi, L. Liu, C. Nieto-Draghi, S. Stephan, I. T. Todorov, N. Vizcaino, X. Zhang, M. T. Horsch, BatCAT deliverable 4.4: Semantic artefacts, schema alignments and mappings, Technical Report, BatCAT consortium, 2025. doi:10.5281/zenodo.18059132.

[6] S. Chiacchiera, et al., Lightweight BatCAT Core Ontology (LBCO) - GitHub repository, https://github.com/HE-BatCAT/lbco, 2025-2026. (Accessed: 27 July 2026).

[7] F. A. Zaccarini, C. Masolo, E. Ghedini, S. Borgo, From Causation (and Parthood) to Time: The Case of EMMO, in: Formal Ontology in Information Systems, IOS Press, 2023, pp. 92–106. doi:10.3233/FAIA231120.
[8] E. Ghedini, et al., Elementary Multiperspective Material Ontology (EMMO) - GitHub repository, https://github.com/emmo-repo/EMMO, 2019-2026. (Accessed: 27 July 2026).
[9] M. T. Horsch, S. Chiacchiera, W. L. Cavalcanti, B. Schembera, Data Technology in Materials Modelling, SpringerBriefs in Applied Sciences and Technology, Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-68597-3.
[10] P. D. Nostro, G. Goldbeck, D. Toti, CHAMEO: An ontology for the harmonisation of materials characterisation methodologies, Applied Ontology 17 (2022) 401–421. doi:10.3233/AO-220271.
[11] W3C, Semantic Sensor Network Ontology, https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/, 2017.
[12] ISO/IEC, ISO/IEC 30173:2023: Digital twin — Concepts and terminology, https://www.iso.org/standard/81442.html, 2023.
[13] C. Barros, R. Salles, E. Ogasawara, G. Guizzardi, F. Porto, Requirements for an Ontology of Digital Twins, CEUR workshop proceedings 2941 (2021). Workshop on Ontology-Driven Conceptual Modelling of Digital Twins, Semantics P and Ds 2021.
[14] S. Clark, F. L. Bleken, S. Stier, E. Flores, C. W. Andersen, M. Marcinek, A. Szczesna-Chrzan, M. Gaberscek, M. R. Palacin, M. Uhrin, J. Friis, Toward a Unified Description of Battery Data, Advanced Energy Materials 12 (2022) 2102702. doi:10.1002/aenm.202102702, number: 17.
[15] C. Soufi, A. Ayadi, T. Mesbahi, A. Samet, C. Lallement, BPO—A battery production ontology for traceable, transparent, and sustainable electric vehicle batteries, Journal of Web Semantics 89 (2026) 100877. doi:10.1016/j.websem.2026.100877.
[16] B. R. Barricelli, E. Casiraghi, D. Fogli, A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications, IEEE Access 7 (2019). doi:10.1109/ACCESS.2019.2953499.
[17] M. Grieves, Intelligent digital twins and the development and management of complex systems, Digital Twin 2 (2022) 8. doi:10.12688/digitaltwin.17574.1.
[18] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, International Journal on Semantic Web and Information Systems 10 (2014) 7–34. doi:10.4018/ijswis.2014040102.
[19] D. Garijo, O. Corcho, M. Poveda-Villalón, FOOPS!: An Ontology Pitfall Scanner for the FAIR Principles, in: International Semantic Web Conference (ISWC) 2021: Posters, Demos, and Industry Tracks, volume 2980 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2980/paper321.pdf.
[20] S. Clark, et al., Battery Domain Ontology (BDO) - GitHub repository, https://github.com/emmo-repo/domain-battery, 2023-2026. (Accessed: 27 July 2026).
[21] CEN, Materials characterization - Terminology and structured documentation, https://www.cencenelec.eu/media/CEN-CENELEC/CWAs/RI/2025/cwa17815_2025.pdf, 2025. (Accessed: 27 July 2026).
[22] J. Moreira, A. Donkers, P. Pauwels, E. Bektas, T. van Ee, Onto4Reuse: Towards an Ontology Reuse Framework for Knowledge-intensive Software Engineering, in: I. Oliveira, P. Barcelos, R. Calhau, C. Fonseca, G. Righetti (Eds.), JOWO 2024 : The Joint Ontology Workshops, CEUR Workshop Proceedings, CEUR-WS.org, 2024. 13th International Workshop on Formal Ontologies Meet Industry; Conference date: 15-19 July 2024.
[23] R. Stevens, P. Lord, J. Malone, N. Matentzoglu, Measuring expert performance at manually classifying domain entities under upper ontology classes, Journal of Web Semantics 57 (2019) 100469. doi:10.1016/j.websem.2018.08.004.
[24] M. A. Musen, The Protégé Project: A Look Back and a Look Forward, AI matters 1 (2015) 4–12. doi:10.1145/2757001.2757003.
[25] BIG-MAP, Battery interface genome - materials acceleration platform, 2020-2024. doi:10.3030/957189, Horizon 2020 project.
[26] NanoMECommons, Harmonisation of EU-wide nanomechanics protocols and relevant data ex-

change procedures, across representative cases; standardisation, interoperability, data workflow, 2021-2025. doi:10.3030/952869, Horizon 2020 project.

[27] P. Del Nostro, G. Goldbeck, D. Toti, The CHAMEO Ontology: Exploiting EMMO's Multiperspective Versatility for Capturing Materials Characterization Procedures, in: Proceedings of the 12th International Workshop on Formal Ontologies meet Industry (FOMI 2022), volume 3240 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[28] P. Del Nostro, G. Goldbeck, F. Kienberger, M. Moertelmaier, A. Pozzi, N. Al-Zubaidi-R-Smith, D. Toti, Battery testing ontology: An EMMO-based semantic framework for representing knowledge in battery testing and battery quality control, Computers in Industry 164 (2025) 104203. doi:10.1016/j.compind.2024.104203.

[29] S. Stier, General Process Ontology (GPO) - GitHub repository, https://github.com/General-Process-Ontology/ontology, 2023. (Accessed: 27 July 2026).

[30] F. Neuhaus, M. Glauer, T. Mossakowski, L. Gerlach, C. Heidfeld, Semantic Dependency in Ontologies, in: T. Prince Sales, C. Masolo, C. M. Keet (Eds.), Frontiers in Artificial Intelligence and Applications, IOS Press, 2025. doi:10.3233/FAIA250490.

[31] P. Fabry, A. Barton, J.-F. Ethier, Rethinking Meaning and Ontologies From the Perspective of Ontological Units, Applied Ontology 21 (2026) 3–23. doi:10.1177/15705838251391253.

[32] H. Hegde, J. Vendetti, D. Goutte-Gattat, J. H. Caufield, J. B. Graybeal, N. L. Harris, N. Karam, C. Kindermann, N. Matentzoglu, J. A. Overton, M. A. Musen, C. J. Mungall, A change language for ontologies and knowledge graphs, Database 2025 (2025) baae133. doi:10.1093/database/baae133.

[33] G. Brewka, T. Eiter, M. Truszczyński, Answer set programming at a glance, Communications of the ACM 54 (2011) 92–103. doi:10.1145/2043174.2043195.

[34] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, Theory and Practice of Logic Programming 19 (2019) 27–82. doi:10.1017/S1471068418000054.

[35] M. Alviano, C. Dodaro, J. Marques-Silva, F. Ricca, Optimum stable model search: Algorithms and implementation, Journal of Logic and Computation 30 (2020) 863–897. doi:10.1093/logcom/exv061.

[36] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), https://www.w3.org/TR/shacl/, 2017. (Accessed: 27 July 2026).