D7.1 – Report on Ontology Standards and Development Strategy

VERSION

VERSION	DATE
1.0	30.11.2020

PROJECT INFORMATION

GRANT AGREEMENT NUMBER	957189
PROJECT FULL TITLE	Battery Interface Genome - Materials Acceleration Platform
PROJECT ACRONYM	BIG-MAP
START DATE OF THE PROJECT	1/9-2020
DURATION	3 years
CALL IDENTIFIER	H2020-LC-BAT-2020-3
PROJECT WEBSITE	big-map.eu

DELIVERABLE INFORMATION

WP NO.	7
WP LEADER	Simon Clark
LEAD BENEFICIARY	SINTEF
CONTRIBUTING PARTNERS	WUT, DTU, EPFL, CSIC, NIC
NATURE	Report
AUTHORS	Anna Szczęsna-Chrzan, Marek Marcinek, Simon Clark, Jesper Friis, Francesca Lønstad Bleken, Roberto Scipioni, Robert Dominko, Rosa Palacin
CONTRIBUTORS	Martin Uhrin, Casper Andersen, Miran Gaberscek
CONTRACTUAL DEADLINE	30.11.2020
DELIVERY DATE TO EC	27.11.2020
DISSEMINATION LEVEL (PU/CO)	PU

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957189.

ABSTRACT

This report describes the standards and development strategy that will be used to create the Battery Interface Ontology (BattINFO) in the project BIG-MAP. The report includes an introduction to the fundamental concepts of ontology with a review of current standards focusing on the European Materials & Modelling Ontology (EMMO). A short description of batteries and their interfaces is provided to give context to the BattINFO development objectives. The initial strategy and tools for the BattINFO development is presented. The content of the report should serve as an orientation for new contributors to the BattINFO.

TABLE OF CONTENTS

1. Introduction

The purpose of this report is to describe the standards and development strategy that will be used to create the Battery Interface Ontology (BattINFO) in the project BIG-MAP. The BattINFO is a knowledge-based digital representation of battery interfaces, that will serve as a descriptive, formalized scheme to improve interoperability and reproducibility of battery experiments and simulations.

BIG-MAP links together the main European actors contributing to the development of new battery materials. This include both experimental and theoretical actors working at many different scales, from electronic via atomistic and discrete mesoscopic to continuum scale. To support the efficient reuse and interoperability of research data between all these actors, it is essential to have a common way to describe the real world entities constituting batteries and their interfaces, their associated physical properties, and how these properties are measured and/or modelled. This common representational framework should be easy to read and understand by humans. At the same time, it must be highly formalised and accurate to be consumed by machines supporting flexible automated workflows that connect different domains and provide rich content to descriptors used for artificial intelligence. Another application of such a common representational framework is to enable semantic search across multiple databases. The battery interface ontology will serve exactly this purpose of being a highly formalised common language for describing battery interfaces.

The structure of this report is laid out as follows. To begin, a general discussion of ontologies in Section 2 is provided to introduce the fundamental concepts and frame the goals of the BattINFO. This includes a review of the current state of ontology standards with a specific focus on the European Materials & Modelling Ontology (EMMO), which will be used as a foundational library for the BattINFO. A general overview of batteries is given in Section 3 to provide context for the objects, properties, and processes that will need to be included in the ontology. This is followed by a discussion of the strategy for the ontology development, with specific proof-of-concept use cases described in Section 4. Finally, Section 5 summarizes the content of the report.

2. Ontology

An ontology is a data model that represents knowledge as a set of concepts within a domain and the relationships between the concepts. By creating a standardized representation of a system, including its constituent concepts with properties and relations, an ontology provides a means not only for classifying data but also for inferring associations. Put simply, the ontology defines some basic concepts and relationships between them from which it is possible to gain new knowledge. One practical benefit of this is that linking an ontology with rich machine-processable semantic descriptions to a database enables AI-based tools to accelerate the discovery of new appropriate materials.

2.1 Ontology Concepts

Readers with a programming background will recognize many of the fundamental concepts of ontology from object-oriented programming (OOP). Indeed, OOP definitions of class-subclass hierarchies and declarations of individuals as instances of a class are essentially the same in ontology development. However, ontology goes further by also defining relations, restrictions, and axioms to enable a richer description of the system.

Some primitive elements of ontologies include:

- **Individuals**: are distinct basic entities in an ontology. Examples are a specific person, a specific version of a model, a specific dataset, etc.
- **Classes**: are the collection of individuals that belong to the class. One can also think about individuals as instantiations of a class, for example, a specific person is an instantiation of the class Person, the specific model is an instance of the class Model, etc.
- **Relations**: specify how classes and individuals are related to each other. An important relation is the *isA* or *isSubclassOf* relation, which is used to provide a classification of classes into a hierarchy of subclasses (the *taxonomy*).
- **Restrictions**: provide a way to define a class by restricting which individuals that can belong to the class. They are often expressed as a relation combined with an existential, universal or cardinality requirement.
- **Annotations**: provides additional content to the entities in the ontology, without being a part of the logical framework itself. They are very important for making the ontology human understandable.
- **Axioms**: are logical propositions that define the relations between the individuals and classes.

Figure 1 shows a simple example of an ontology to illustrate the concept. Information can be expressed in the form of a *subject-predicate-object triple*. In ontologies, the subject and object are classes or individuals, and the predicate is a relation. For example, in the triple *Rose isA Flower, Rose* is the subject, *isA* is the predicate, and *Flower* is the object. Linking multiple *isA* triples together creates a tree-like taxonomy (*Rose isA flower isA Plant isA Thing*). Individual objects can be introduced by declaring them as instances of a class (*RoseA isAnInstanceOf Rose).* An ontology provides relations that are richer than the simple isA triples of a taxonomy. For example, it can be said that the Rose has a leaf, which can be declared through *RosaA hasPart some LeafA* and *LeafA isAnInstanceOf Leaf.* Taking this one step further, it can be said that *Leaf isA Appendage* and *Appendage isA Thing*. In addition to the *isA* or *isSubclassOf* relations, classes can be further refined by *restrictions* that restricts which individuals that belong the class. An example of an existential restriction is *Plant hasPart some Appendage*, from which it can be reasoned that having an appendage is a necessary part of being a plant. Other types of restrictions are listed in Table 1. One must also take care in the definition of relations to avoid false reasonings. For example, one might be tempted to include the relation that *Rose hasPart some Leaf*, however, this would imply that if all the leaves were removed from the stem, then it would no longer be a rose. The opposite, *Leaf isPartOf some Rose* is not true either, because there are members of the class *Leaf* that are not a part of a rose, e.g. a marble leaf.

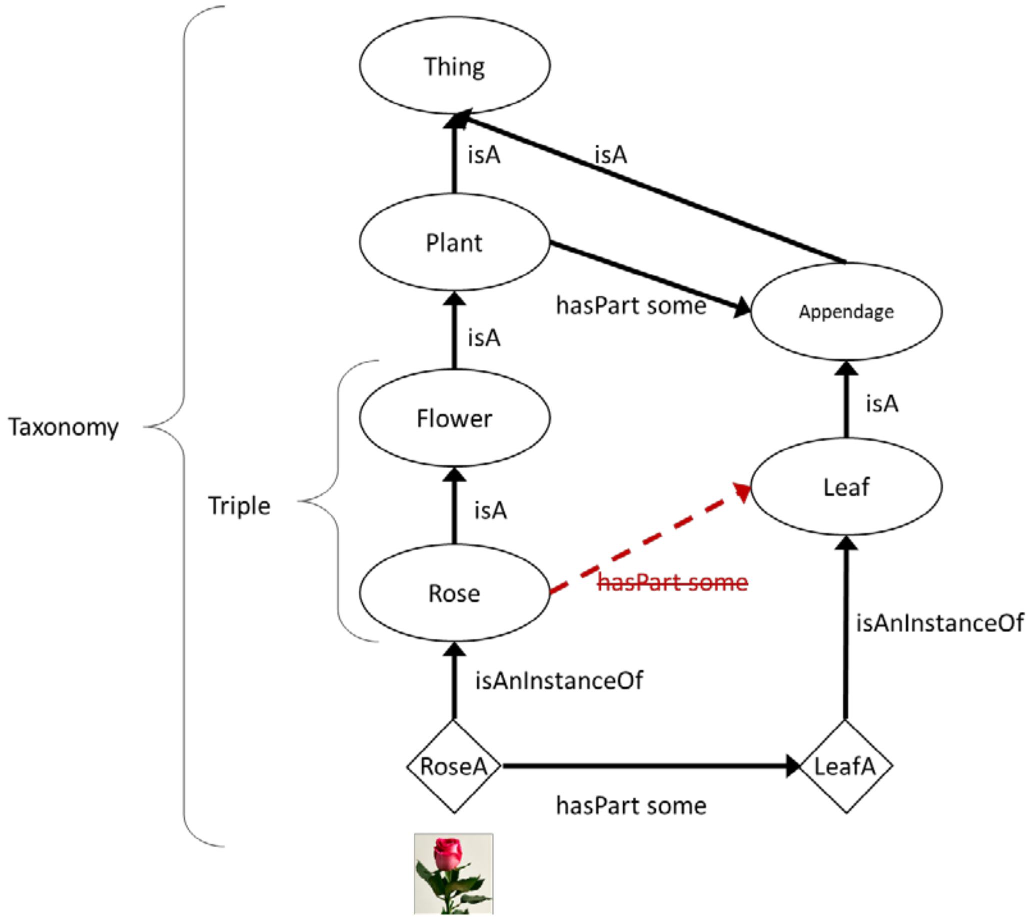

Figure 1. Example of a simple ontology of a rose.

Of course, the preceding example is a trivial demonstration of the concept. Real ontologies include many dozens of different types of relations and entities to create versatile descriptions of complex systems.

Another important aspect of ontologies is their logical foundation. EMMO and most other ontologies applied in computer science are based on so-called *description logic*, which is less expressive than first order logic, but with the great advantage of being decidable, which ensures that computers can perform logical reasoning within a finite time (given sufficient memory). EMMO and hence domain ontologies based on it, relies heavily on logical reasoning, because the full taxonomy is not asserted in the declaration of the ontology, but inferred from logical reasoning. In this report we will not go into details of descriptive logics. Instead we summarise the logical constructs that the users will meet in Table 1.

Table 1. Logical constructions in description logic using the Manchester notation (used in Protégé) are shown. Below *A* and *B* are classes, *R* is a relation (object property), *a* and *b* are individuals and *n* is a literal. Inspired by the Great table of Description Logics.

DL	MANCHESTER	READ	MEANING
Constants			
$\top$	Thing	top	A special class with every individual as an instance
$\bot$	Nothing	bottom	The empty class
Axioms			
$A \sqsubseteq B$	A isSubclassOf B	A is a subclass of B	Class *inclusion*
$A \equiv B$	A isEquivalentTo B	A is equivalent to B	Class *equivalence*
$a: A$	a isA A	a is a A	Class *assertion* (*instantiation*)
$(a, b): R$	a R b	a is R-related to b	Object property *assertion*
$(a, n): D$	a D n	A is D-related to b	Data property *assertion*
Constructions			
$A \sqcap B$	A and B	A and B	Class *intersection* (*conjunction*)
$A \sqcup B$	A or B	A or B	Class *union* (*disjunction*)
$\neg A$	not A	not A	Class *complement* (*negation*)
$\{a, b, \dots\}$	{a, b, …}	one of a, b, …	Class *enumeration*
R^{-}	inverse R	Inverse of R	Object property *inversion*
$\exists R.A$	R some A	some A with R	*Existential restriction* (there exists at least one member of A that satisfies R)
$\forall R.A$	R only A	all A with R	*Universal restriction* (R is satisfied for all members of A)
$= nR.A$	R exactly n A	exactly n A with R	*Cardinality restriction* (R is satisfied by n members of A)
$\leq nR.A$	R min n A	minimum n A with R	*Minimum cardinality restriction* (R is satisfied by at least n members of A)
$\geq nR.A$	R max n A	maximum n A with R	*Maximum cardinality restriction* (R is satisfied by at most n members of A)
$\exists R\{a, b, \dots\}$	R value {a, b…}	has value a, b, … with R	*Value restriction* (individuals a, b, … satisfy R)
Decompositions			
$A \sqcap B \sqsubseteq \bot$	A disjoint with B	A is disjoint with B	*Disjointness* (there are no common members of A and B)
$\exists R.\top \sqsubseteq A$	R domain A	domain of R is A	*Domain* (restriction R applies to class A)
$\top \sqsubseteq \forall R.B$	R range B	range of R is B	*Range* (values of restriction R belong to class B)

Property characteristics		
$R \circ R \sqsubseteq R$	R is transitive	*Transitivity*
$\top \sqsubseteq \leq 1R.\top$	R is functional	*Functionality*
$\top \sqsubseteq \leq 1R^{-}.\top$	R is inverse functional	*Inverse functionality*
$R \sqsubseteq R^{-}$	R is symmetric	*Symmetry*
$R \sqsubseteq \neg R^{-}$	R is asymmetric	*Asymmetry*
$A \sqsubseteq \exists R.A$	R is reflexive	*Reflexivity*
$A \sqsubseteq \neg\exists R.A$	R is irreflexive	*Irreflexivity*

Ontologies are commonly organised hierarchically as schematically illustrated in Figure 2. The common definitions shared by all ontologies in the same family is set at the top level. This is typically small and includes only the fundamental concepts. The middle-level ontologies extend the top-level with basic concepts shared between several domains. For materials science this can be the concept of a measurement or a model, how they are related to properties, commonly used physical quantities and units etc. Domain ontologies describe a particular discipline or field. For example, the BattINFO will ultimately be realized as a domain ontology for batteries and battery interfaces. Domain ontologies provide a common language within the domain, but stay generic, such that they can be used by all relevant applications within the domain. This means that they, like top and middle level ontologies, they define classes but rarely individuals. Application ontologies occupy the lowest level and define application-specific concepts and individuals of limited generic interest. Within all of these general categories, there may be sub-hierarchies of ontologies depending on each other.

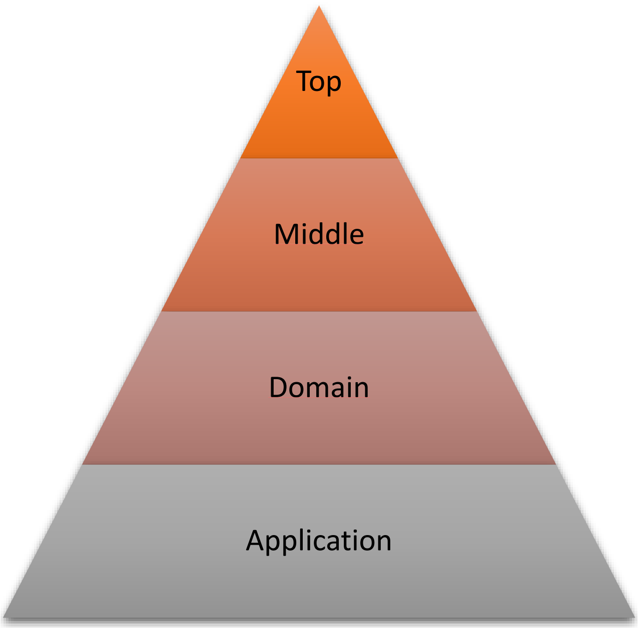

Figure 2. Ontology hierarchy

2.2 Web Ontology Language (OWL) and Protégé

The most common tools used to build ontology-based applications are the web ontology language (OWL) and the open-source editor Protégé.

OWL is the de facto standard language for creating ontologies. Developed and maintained by the world wide web consortium (W3C), OWL is described as a semantic markup language for publishing and sharing ontologies on the World Wide Web. It is an extension of the Resource Description Framework (RDF). Since its initial development in the early 2000s, OWL has grown to include a family of languages based on the specifications released in 2004 (OWL[1]) and 2009 (OWL2[2]). Current development focuses on the utilization of OWL2.

Protégé is a free open-source ontology editor and browser developed by Stanford University[3]. It provides a graphical user interface to define ontologies and includes semantic reasoner to make inferences from the asserted relations and axioms. Protégé is available both as a web-based application and as a desktop application and provides a wealth of documentation and tutorials to help orient new users[4].

2.3 Ontology Standards

The OntoCommons CSA project on standardised documentation of data through taxonomies and ontologies (NMBP-39) that is planned to start Nov. 1st 2020 identifies and brings together three main foundational ontologies with their own communities around them. These are:

- **Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)** is a foundational ontology maintained by ISTC-CNR Laboratory for Applied Ontology that has remained stable since its first release in 2002/2003. It has many applications within engineering and manufacturing.

- **Basic Formal Ontology (BFO)** is an established a top-level 3D ontology developed by Barry Schmit and co-workers. It divides all entities into continuants and occurrents and has been applied to many projects in the area of biomedicine, security and defence.

- **European Materials & Modelling Ontology (EMMO)** is a new top- and middle-level 4D ontology aimed at providing a standard representational ontology framework for applied physical sciences. It is developed and maintained by the European Materials Modelling Council (EMMC). Even though it is still under development (v1.0.0-alpha2 was released Oct. 15, 2020), it is in a useful condition and is currently being actively used and applied in several EU projects.

[1] https://www.w3.org/TR/owl-ref/
[2] https://www.w3.org/TR/owl2-overview/
[3] https://protege.stanford.edu/
[4] https://protegewiki.stanford.edu/wiki/Main_Page

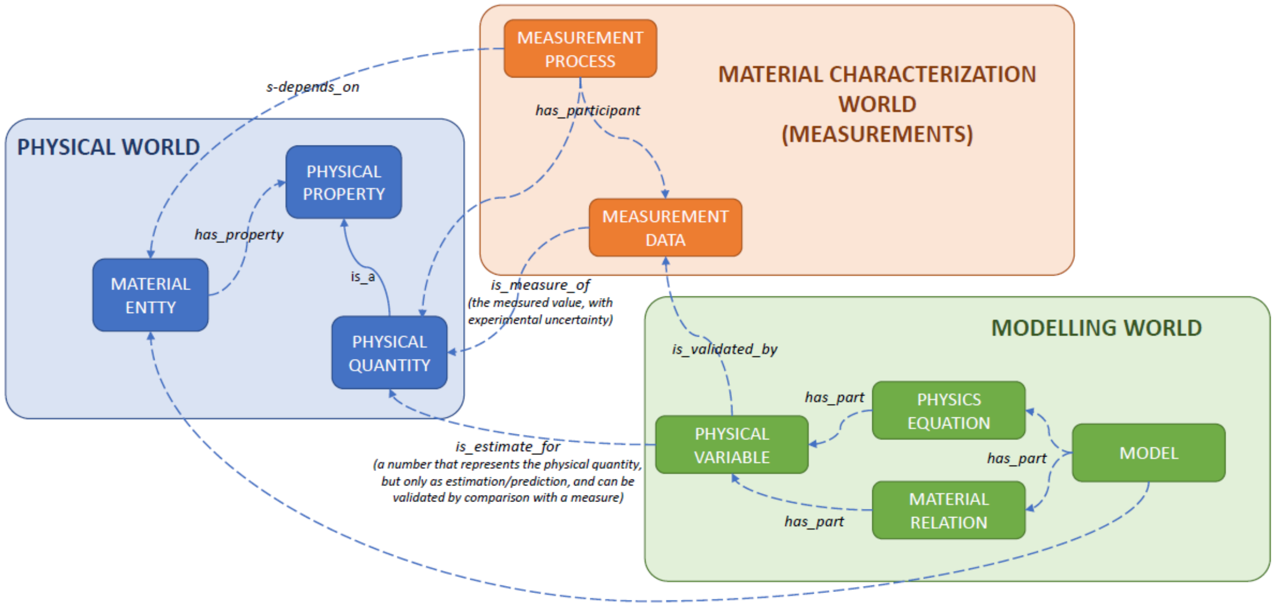

Figure 3. EMMO provides the connection between the physical world, materials characterization world and materials modelling world [1]**.**

The EMMO is most well-suited to serve as the top- and middle-level ontology for the BattINFO. It is described in more detail in the following section.

2.4 European Materials & Modelling Ontology

The European Materials & Modelling Ontology (EMMO) aims at the development of a standard representational ontology framework based on current materials modelling and characterization of knowledge. EMMO starts from the very basic scientific fundamentals and grows to encompass a complex and wide field of knowledge, however it is still functional and clear.

Figure 4 shows the aspects of materials modelling and characterisation that the EMMO is intended to cover. They include:

• the **material** itself,
• the **observation** process,
• the **properties** that is measured or modelled,
• the **physics laws** that describes the material behaviour,
• the **physical models** that approximate the physics laws,
• the **solver** including the numerical discretisation method that leads to a solvable mathematical representation under certain simplifying assumptions,
• the **numerical solver** that performs the calculations, and
• the **post processing** of experimental or simulated data.

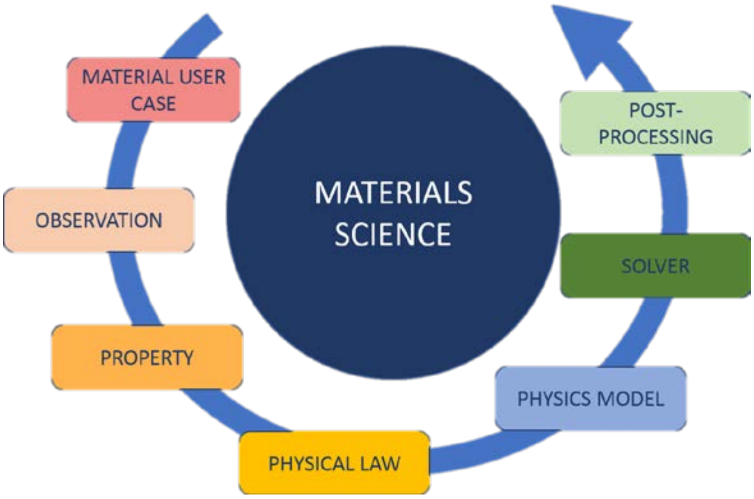

Figure 4. The aspects of materials modelling and characterization covered by EMMO [1].

The EMMO is built on several theoretical foundations including semiotics, set theory, mereology, topology, metrology, and descriptive logic [1]. **Semiotics** is the study of meaning-making, and it is applied in EMMO (via the semion class) to create a physical entity that represents an abstract object. **Set theory** is the theory of membership and is accessed via the set class in EMMO. Sets are defined via the has_member / is_member_of relations. **Mereology** is the science of parenthood. It is introduced via the item class and is based on the mereological has_part / is_part_of relations. **Topology** is the study of geometrical properties and spatial relations. In EMMO, it is included via the substrate class and represents the place in space and time in which every real world item exists. **Metrology** is the science of measurements and is used to introduce units and link them to properties. Finally, description logic is a formal knowledge representation language in which axioms are expressed.

The EMMO top level is the group of fundamental axioms that constitute the philosophical foundation of the EMMO. Adopting a physicalistic/nominalistic perspective, the EMMO defines real world objects as 4D objects that are always extended in space and time.

The middle level ontologies act as roots for extending the EMMO towards specific application domains. In EMMO, the different perspectives that can be adopted when describing a system or model, are included, so that the most appropriate viewpoint can be taken for the specific problem.

In the *Reductionistic perspective* the direct part-hood relation is prominent, providing a means to describing a system throughout all scales and thus also between granularity levels. This perspective is relevant for the BattINFO in for instance considering simulations at a lower granularity level feeding data to a higher level.

In the *Holistic perspective* changes in the system over time can be included by introducing the Participant and Process classes. Such changes over time may be the formation and propagation of the SEI in the battery. In the Semiotics submodules of Holistic the Semiosis process is covered, linking Sign (e.g. a measured physical quantity or a model), Object (real world object) and Interpreter (e.g. a measurement instrument) as participants.

The *Perceptual perspective* introduces things that can be perceived, like a sound graphical symbols written on a paper. This includes the Symbolic branch covering symbols and formal and natural languages.

The *Physicalistic perspective* introduces concepts of real-world objects that are important for applied physical sciences, for instance Matter and Field. A battery clearly falls into the realm of applied physics and the Battery Interface Ontology will also make use of concepts from this perspective.

3. Batteries

A battery is a device that converts the chemical energy stored in the electrode materials into useful electrical energy. Over the last 200 years, many different battery chemistries have been developed with varying degrees of commercial success. Alkaline zinc-based batteries have long dominated the market for primary (non-rechargeable) batteries due to their high energy and low cost. Lead-acid batteries found widespread use as car batteries and for stationary home energy storage (e.g. for solar PV systems). But the development of the Li-ion battery in the late 20th century revolutionized the field of electrochemical energy storage by offering an affordable, energy-dense, and long-lasting secondary (rechargeable) battery chemistry. The development of Li-ion batteries was so significant, it was awarded the Nobel Prize for Chemistry in 2019.

Today, battery development has become one of the most widely researched topics in the world with potential applications spanning a plethora of industries from personal electronics to electric mobility to grid-scale energy storage. Research in Li-ion battery development remains the primary focus of these activities, but there is a growing recognition of the material supply limitations that will face Li-ion battery manufacturing in the future and a movement to research post-lithium-ion batteries.

In this section, the fundamental battery components and processes needed to create the first version of the battery interface ontology are introduced. To support the BIG-MAP goal of chemistry neutrality, the descriptions are kept as general as possible. In cases where specificity is required, then descriptions consistent with Generation 3 Li-ion batteries are used.

3.1 Battery Cell Components

A cell is the fundamental unit of a battery. It comprises three basic parts: a negative electrode, a positive electrode, and an ionic conductive interlayer which electronically separates two electrodes and enables ionic conductivity. The ionic conductive medium can take the form of a liquid that impregnates porous separator or the form of solid-state ionic conducting medium. The electrodes are connected through tabs to the cell housing, enabling the flow of electrons through the external sink/supply. An overview of a typical 18650 sized Li-ion battery cell showing the configuration of the negative electrode, positive electrode, and separator is shown in Figure 5.

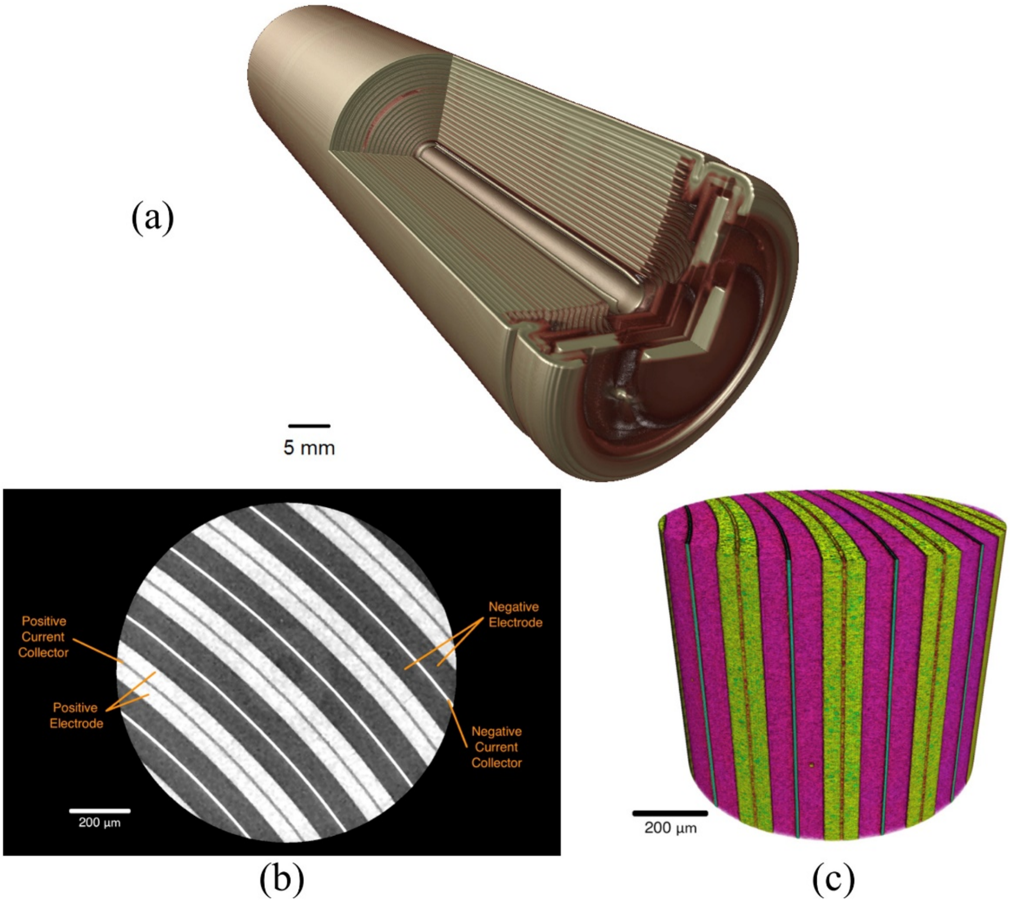

Figure 5. An illustrative example of a typical battery cell configuration showing (a) A 3D image of an entire 18650 Li-ion battery cell (b) Optically enlarging a smaller region from the center of the 18650 battery cell reveals the details of the negative electrode, positive electrode, and separator layers. (c) Virtual slices were assembled into a 3D volume, rendered here for the purposes of visualization. The yellow layers represent the positive electrode and the magenta layers represent the negative electrode, each with their respective current collectors sandwiched between electrode layers. Reproduced with permission from Ref. [2]**.**

In battery jargon, the negative electrode is sometimes called the anode and the positive electrode is sometimes called the cathode. The correct definitions for these terms are:

Anode: electrode at which the oxidation reaction occurs.
Cathode: electrode at which the reduction reaction occurs.

In secondary batteries, the reaction occurring at a given electrode changes depending on whether the cell is being charged or discharged. Therefore, the terms **negative electrode** and **positive electrode** are preferred to avoid ambiguity.

Both the positive and negative electrodes are typically composites consisting of a large part of electrochemically active material, conductive additives like carbon black, carbon fibres and a binder, typically a polymer to enhance mechanical properties. Active materials should have stable structure during cycling to provide good lithium intercalation and deintercalation, which is why the synthesis path is important and still improved. Composite electrodes are deposited and calendared on metal current collectors (typically foils of aluminium for positive electrodes and copper for negative electrodes in Li-ion batteries). The thickness, porosity and contacts between current collector and contacts between particles in the composite electrodes are essential parameters for utilization of the capacity of the active material. The ratio of the components, which builds negative and positive electrodes domains ale optimized to prove the best mechanical and electrochemical parameters.

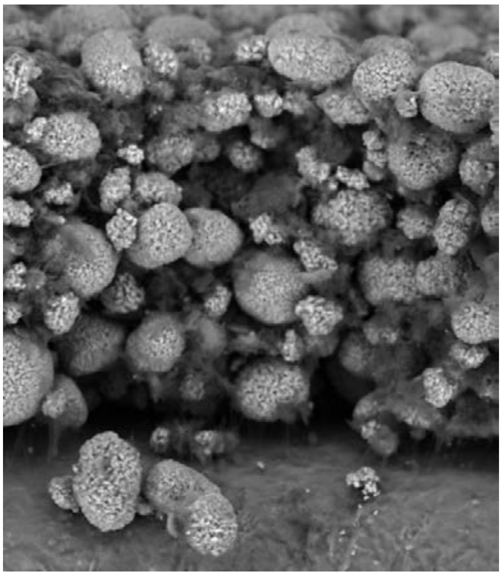

Figure 6. Cross-section image of a Li-ion composite positive electrode showing the active material particles held together with a polymer binder cast on a foil current collector.

Liquid electrolytes consist of one or more salts dissolved in a solvent or more commonly a mixture of solvents. Additionally, the electrolyte can contain of one or several additives (in a few weight percent concentration) for improved operation of the battery cell. Additives can have many functions including limiting thermal runaway, improving SEI formation, or many others. Supply of ions from the electrolyte should enable homogenous diffusion of cation within the solid matrix of the electrode active material.

3.2 Battery Cell Processes

The performance and lifetime of battery cells are determined by a host of processes occurring across a broad span of time and length scales. Dominant processes that describe the fundamentals of battery cell behaviour include: electrochemical reactions, mass transport, charge transport, heat transport.

3.2.1 Electrochemical Reactions

Electrochemical charge-transfer reactions are essential to the working principle of battery cells. Electrochemical reactions are defined as reactions that involve – in addition to molecules and ions – electrons donated from a metal or other substance. By convention, electrochemical reactions are *oxidations* if they result in the liberation of electrons and *reductions* if they consume electrons. A generic description of electrochemical reactions is:

$$\nu_1' M_1' + \nu_2' M_2' + \cdots + n e^- \rightleftarrows \nu_1'' M_1'' + \nu_2'' M_2'' + \cdots \tag{1}$$

where the following condition is fulfilled:

$$\sum \nu M + ne^{-} = 0 \tag{2}$$

This description of electrochemical reactions is valid for all electrode types and chemistries. The electrochemical reactions occur at the electrode-electrolyte interface. The driving force for electrochemical reactions is the change in electrochemical potential between the products and the educts. The most common phenomenological model for electrochemical reaction kinetics on the continuum scale is the Butler-Volmer approximation [3] while Marcus theory is useful to couple atomistic scale descriptions of charge transfer kinetics.

For Li-ion batteries specifically, the negative electrode and positive electrode reactions are:

Negative Electrode: $$\mathrm{LiC_6} \rightleftarrows \mathrm{C_6} + \mathrm{Li^+} + \mathrm{e^-} \tag{3}$$

Positive Electrode: $$\mathrm{MO} + \mathrm{Li^+} + \mathrm{e^-} \rightleftarrows \mathrm{LiMO} \tag{4}$$

Where the forward arrow indicates the direction of the reaction when the cell is discharged, and MO refers to a metal oxide (e.g. CoO_2).

3.2.2 Mass and Charge Transport

The transport of mass and charge in the cell must occur to support the electrochemical reactions. Important processes to consider include transport of ions in the electrolyte and transport of electrons in the electrodes. In the case of intercalation electrodes (e.g. in Li-ion batteries) the solid-state diffusion of charge-neutral molecules in the electrodes must also be considered. The driving force for mass and charge transport processes is the spatial gradient of electrochemical potential.

In liquid electrolytes, the transport dissolved ions – together with their solvation shells – is driven both by gradients in the ionic species concentration as well as the local electric potential. Different theoretical models exist to describe this process, the most relevant for liquid battery electrolytes being concentrated solution theory (CST) [4]. In the solid electrodes, the transport of electrons is driven by the gradient in the local electric potential and can be described by Ohm's law. In the case of intercalation electrodes, the solid-state diffusion of charge-neutral molecules (e.g. Li) is driven by the gradient in concentration and can be described by Fick's law.

3.2.3 Heat Transport

The temperature of battery cells during operation changes due to both thermal conduction and the presence of heat sources. Dominant sources of heat include: Joule heating, the Thompson effect, heat of mixing, and the Soret effect [4].

3.2.4 Degradation

Aging and degradation processes are always present inside battery cells. They can be classified as a chemical aging (the calendar aging that occur during cell storage) and as an electrochemical ageing that take place during the cell operation. Both types of ageing depend on many factors, like temperature (environmental and inside the cell – exothermic reactions) [5], operating voltage and rate and mechanical stress of the battery pack.

Typical aging processes occurring in the bulk of active components are:

- structural and compositional changes,
- consumption of active components,
- active material pulverisation, and
- disintegration of the composite electrode

Typical aging processes occurring at the interface are:

- formation of the interphase films and their thickening,
- corrosion of the active and non-active components,
- cross communication between two electrodes causing poisoning of interface layers, and
- increase of electronic resistance and pore clogging decreasing cell performance.

3.3 Battery Cell Interfaces

Considering the descriptions above, the processes in the battery cell proceed through several interfaces and depend on the ability of the bulk to accommodate large differences in the energetic levels. Those differences have an impact on the bulk properties of the active materials, like structural stability, volumetric changes, morphological changes, and active material degradation. Large differences in the energetic levels compromise interfaces of active materials, additives and current collectors. Thus, chemical reactions often take place at the interfaces, with formation of the solid-electrolyte interphase (SEI) film on the negative electrode/electrolyte interface (Figure 7) and the cathode-electrolyte interface (CEI) film on the positive electrode/electrolyte interface (Figure 8).

Figure 7 provides a schematic of the initial SEI formation process along with proposed long-term growth mechanisms. The initial formation of the SEI is driven by the fact that the low potential of the negative electrode is often outside of the electrochemical stability window of the electrolyte. As a result, the electrolyte is decomposed at the electrode-electrolyte interface. On the one hand, this can have beneficial effects because the SEI can help to remove the solvation shell from Li+ ions and avoid the destructive co-intercalation of solvent. The formation of such films, when they are electronically insulating and have stability and good adhesion to the electrodes, typically prevents further degradation and enables electrolyte operation beyond its thermodynamic stability window. On the other hand, the SEI grows continuously, leading to the slow loss of battery capacity.

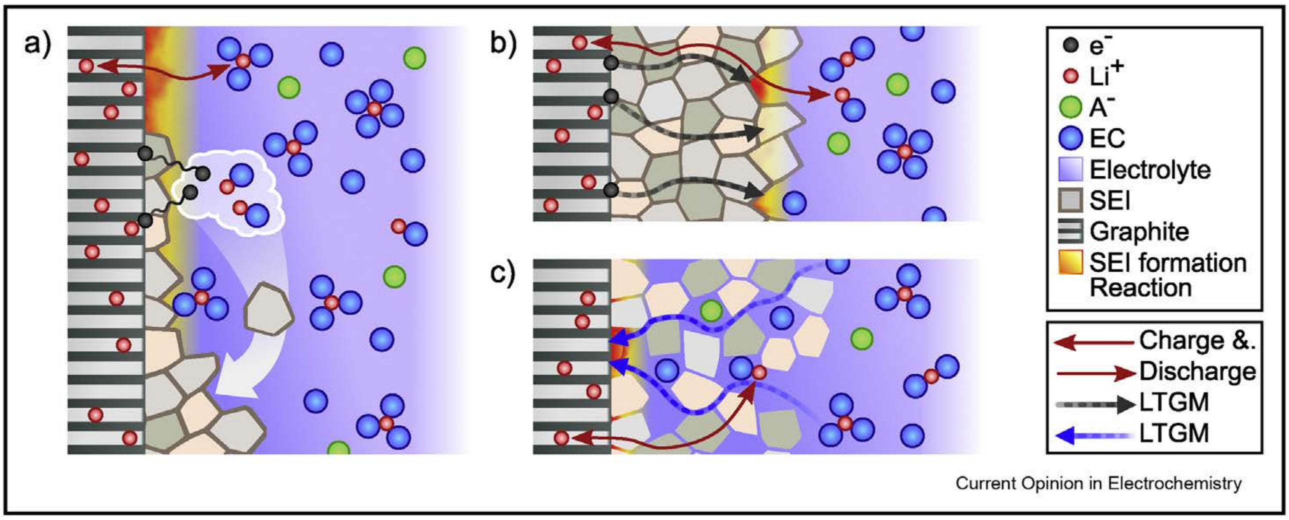

Figure 7. Cross-section through the negative electrode, the SEI, and the electrolyte showing charge, discharge, and long-term growth mechanisms (LTGM). Solvent, Li ions and electrons are mobile species and move as indicated by the corresponding arrows. (a) Initial SEI formation: Electrons tunnel, electrolyte is reduced and reduction products precipitate as solid film. (b) Long-term SEI growth proceeds via a mechanism that transports negative charge to the SEI/electrolyte interface. (c) Alternatively, long-term SEI growth is caused by electrolyte diffusing towards the electrode/SEI interface. Reproduced with permission from Ref. [6]**.**

Formation of such films consumes active components, electrolyte and additives, and can also alter the surface of the active material itself. They may also result in an increase of internal resistance (higher overpotential for electrochemical reactions), clogging the pores in the electrode composite, and corrosion of current collectors. The reactions yielding the formation of such interphases may also involve the formation of gaseous products and result in dissolved species in the electrolyte that can shuttle between two electrodes and affecting the quality of interface layer (either CEI or SEI). For instance, dissolved metal cations can be reduced on the surface negative electrode, inducing some electronic conductivity and hence further formation of passivation layers and consumption of electrolyte.

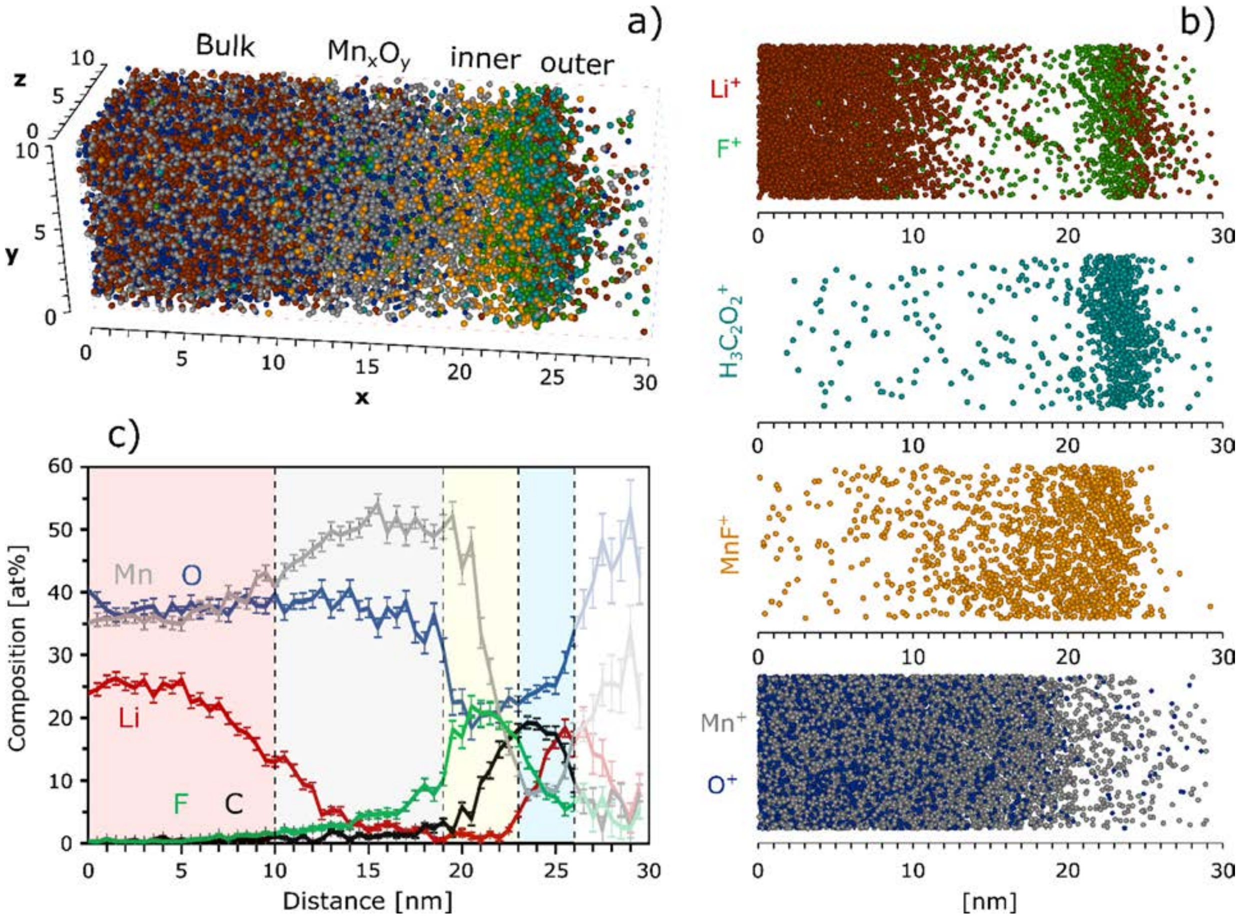

Figure 8. Atomic resolution analysis of the cathode electrolyte interface on LMO. a) 3D APT ion map of a sub-volume (30 ×10 ×10 nm3) of the LMO tip and b) 2D projections, normal to the xy plane, of the single ion maps c) Composition profiles of the atomic fractions for Li, Mn, O, F and C. Reproduced with permission from Ref. [7]**.**

3.4 Battery Characterization Methods and Data

Batteries are complex systems and there exist a plethora of experimental and model-based methods to characterize their performance. Establishing links between experimental and model-based methods of determining the properties required to describe battery objects are essential to the development of the BattINFO. This is a central goal of the ontology development in BIG-MAP, which will build also on the already public work of other current research initiatives including the EU project DEFACTO[5].

Table 2. Example of typical battery characterization data obtained through experimental methods[6].

Data category	What does the data describe	Methods used for collecting the data	Type
Performance degradation	The electrochemical performance of the batteries versus operational time	Potentiostatic amperometry, Galvanostatic potentiometry, Electrochemical impedance spectroscopy	Spectra Key performance

[5] https://defacto-project.eu/

[6] Reproduced from the BIG-MAP data management plan (DMP) for WP5.

			indicators versus time
Structural data (lab.)	Crystal structure of the cathode and the anode before and after cycling	X-ray diffraction	XRD spectra
	Phase distribution and phase transformations, strain & stress	Data analysis	To be defined
Structural data (large-scale facilities)	Relation between the phase transformations of the crystal structures in the electrodes, and the electrochemical activity, i.e. characterising the cells in real time.	X-ray diffraction Neutron diffraction	To be defined
	Relation between the nanoscale structural organisation in the electrodes, and the electrochemical activity, i.e. characterising the cells in real time.	SAXS SANS	2D patterns
Spatially-resolved microscopy-based structural/morphological data (lab.)	2D/3D distribution of phases, nano- and microstructures including porosity, cracks & defects, SEI,…	FIB-SEM STEM-HAADF HR-TEM	Images
Spatially-resolved structural/morphological data (large scale facilities)	2D or 3D distribution of phases, crystalline structures, nano- and microstructures including porosity mapping / cracks & defects	XRD-CT/ SAXS-CT X-ray microtomography X-ray nanotomography X-ray ptychography Neutron Imaging Neutron Depth Profiling	To be defined
Spectroscopic data (lab.)	Local chemical environment, electronic structures, oxidation states, SEI compounds	XPS, IR, Raman NMR Auger EDS/EDX, EELS	Spectra
Spectroscopic data (large scale facilities)	Local chemical environment, electronic structures, oxidation states, SEI compounds	Synchrotron XPS X-ray Raman Scattering XAS (EXAFS / XANES)	Spectra
Inelastic Scattering data (large scale facilities)	Collective excitations, transport mechanisms	QENS, INS RIXS	Spectra
Surface data (large scale facilities)	SEI composition and morphology	Neutron Reflectometry X-rays Reflectivity	Spectra

Table 3. Example of typical battery characterization data obtained through computational methods[7].

Data category and engines	What does the data describe	Methods used for collecting the data	Format
Category: Electronic Structure: WFT,DFT, QMC Engines (molecular): GAUSSIAN, ORCA, MOLPRO, TURBOMOLE, NWChem, QChem, ADF, PSI4, MRCC, NECI Engines (periodic): CP2K, VASP, QUANTUM ESPRESO, Yambo, Castep, GPAW, QuantumATK, Crystal, NECI	Structures, Energy-related data, Wave functions & electronic properties, AIMD trajectories, different types of spectra	A tarball can be created from the calculation folder, including relevant inputs and outputs	- .tar.gz (an archive of input and output text, XML, netctdf, hdf5, or any other machine readable file)
Category: ML-FF (representation & regression) Engines: QUIP & GAP codes	SOAP/ACE + GAP Parameters	As above	As above
Category: Alchemical Exploration & Optimization: Engines: QML & APDFT codes	Property relationships & Compound space search	As above	As above
Category: Atomistic simul. Engines: LAMMPS, GROMACS,QUIP	Atomic trajectories, and associated transport, spectral etc. properties	As above	As above
Category: Platform for calc & workflows Engine: AiiDA	Calculations carried out using AiiDA, including all inputs and outputs in a provenance graph	AiiDA automatically records the full provenance graph as a workflow executes	- AiiDA export file (see specs below) - JSON file with relevant parsed inputs and outputs
Category: Platform for calc & workflows Engine: ASE	Structures generated in or resulting from ASE calculations	Data is manually stored in an ASE database	- sqlite (ASE database format) - xsf file - extended xyz format
Category: Platform for calc &workflows Engine: 3DEXPERIENCE platform / Pipeline pilot.	Structures and computational data from calculations carried out with Pipeline pilot protocols, protocols	The Pipeline pilot protocols can be tailored to store structures and computational data plus any additional meta data required	-The 3D file format. Read & Write components will be created to export data to commonly accepted formats, facilitating exchange with materials databases
Category: Calculation metadata (Nested dictionary with keys and values)	Information about a calculation such as who carried it out and using which version of the software	Users are free to create this manually or have script to autogenerate it	JSON

[7] Reproduced from the BIG-MAP data management plan (DMP) for WP2.

4. Strategy for Battery Interface Ontology Development

This section describes the strategy that will guide the initial development of the BattINFO and inform its long-term development throughout the project. The development strategy first reviews the goals of the BattINFO, then provides an overview of the tools that will be used before discussing the foreseen design procedure. Finally, first use case definitions are proposed.

4.1 Goals for the Battery Interface Ontology

The main goal of the BattINFO is to support the interoperability of data across multiple scales, techniques, and domains in the battery discovery process. This should be applied to help BIG-MAP create a database utilizing a descriptive, formalized scheme to improve the reproducibility of experiments and simulations. The ontology should map the connections between battery data, models, and methods in a machine-readable format to support the use of AI tools to accelerate materials discovery and development. The ontology framework should be general to support to goal of chemistry neutrality.

Other goals for the development of the BattINFO are:

- The BattINFO shall include descriptions of data provenance including quantifiers of uncertainty.
- The BattINFO shall support direct integration into web-based lab tools to be developed in WP8 to support the direct tagging and integration of data though e.g. online lab-books.
- The BattINFO shall be well-documented, versatile, and robust such that the development within the battery community can continue after the end of the project.

To achieve these goals, each part of the battery domain will be carefully considered in the ontology. Objects describing the basic battery components (negative electrode, positive electrode, electrolyte, separator) together with the corresponding material categories and their properties will be ontologized. These objects will then be linked to the dominant battery processes, complete with descriptions of their governing physical laws. The ontology will be implemented in the web ontology language (OWL2) and built using the Protégé editor.

4.2 Ontology Developer Toolbox

The tools necessary to contribute to the ontology development are freely available. The initial concept development will be performed in Microsoft PowerPoint and Excel. Common files for the BattINFO concept development are hosted on the BIG-MAP Sharepoint.

The ontology will be implemented using the open-source tool Protégé[3]. A helpful introductory guide for new users developing their first ontologies in Protégé is provided by Stanford University[8]. Specific instructions for setting up Protégé and the FaCT++ reasoner for use with EMMO are provided on the EMMO GitHub page[9].

Throughout the development process, the code will be hosted in a repository on GitHub (https://github.com/BIG-MAP-ontologies). Hosting on GitHub will support retaining version control and the initial dissemination of the ontology within the BIG-MAP consortium. The Git repository will

[8] https://protege.stanford.edu/publications/ontology_development/ontology101.pdf
[9] https://github.com/emmo-repo/EMMO/blob/1.0.0-alpha2/doc/protege-setup.md

include a readme file containing an up-to-date description of the conventions in place for the ontology development. As the data management and dissemination activities of BIG-MAP progress, the code may be migrated to a dedicated BIG-MAP page on GitLab or a dedicated BIG-MAP App Store.

4.3 Ontology Design

The Battery Interface Ontology will be built using the European Materials Modelling Ontology (EMMO) as a foundation. By utilizing the foundational elements of the EMMO, the development of the Battery Interface Ontology can be designed in a way that is both robust and able to accommodate the very fast development schedule of the BIG-MAP.

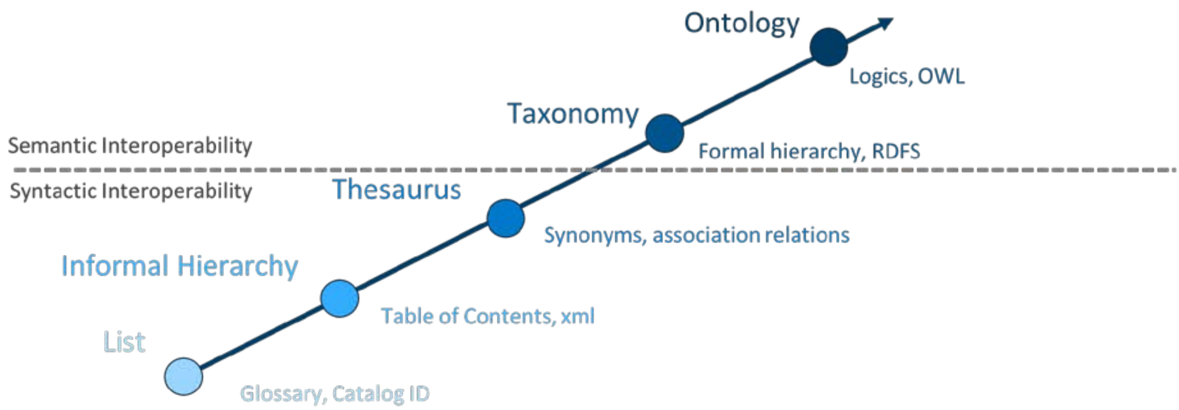

Figure 9. Ontology building blocks

Figure 9 illustrates the generic steps for ontology development. This will serve as a guide for the BattINFO design process. The first step in the ontology design is to create a list of all the objects that will need to be ontologized. This will be accomplished by first identifying the major models of material components necessary to describe the battery. These will be accompanied by a corresponding *isA* statement, a detailed written description, and a list of its parts (informal hierarchy). Then a list of the associate properties will be populated along with information describing how the properties are measured and/or modelled (e.g. from the methods listed in Table 1 and Table 2).

The list of objects, parts, properties, and observations will be compiled using the Microsoft Excel template shown in Table 3. Through the creation of the list, it may occur that the same names or symbols are used to describe different aspects of the battery, as is often the case in complex systems. To address this issue, a Thesaurus is created to avoid ambiguity resulting from synonymous definitions.

Table 4. Example of the BIG-MAP ontology template for defining objects to be included.

Model	Description	Part	Property			Observation Procedure	
			Name	Units	Symbol	Description of Semiotic Process	Interpreter
Object 1	*(natural language description)*	*Parts*	*Property 1*	*(SI unit if it is a quantity, or a reference to the scale used)*	*(e.g. vector, scalar, RGB color scale, names list)*	*(how the observation occurs) (e.g. ISO procedure)*	*(who is assigning the property value to the object) (e.g. customer, measurement device)*
			Property 2	…	…	…	…
			Property 3	…	…	…	…

The main processes governing the behavior of the battery and its interfaces will also be identified and described. A complete description of a process includes a list of the *material entities* and *models* that participate, physical laws that govern the process, and common models used to describe it. These processes will then be diagrammed with an accompanying written description. To create process descriptions, it is necessary to map the process with respect to *causality*, not time. Initial process mapping will be performed in Microsoft PowerPoint. An example process diagram is shown in Figure 10.

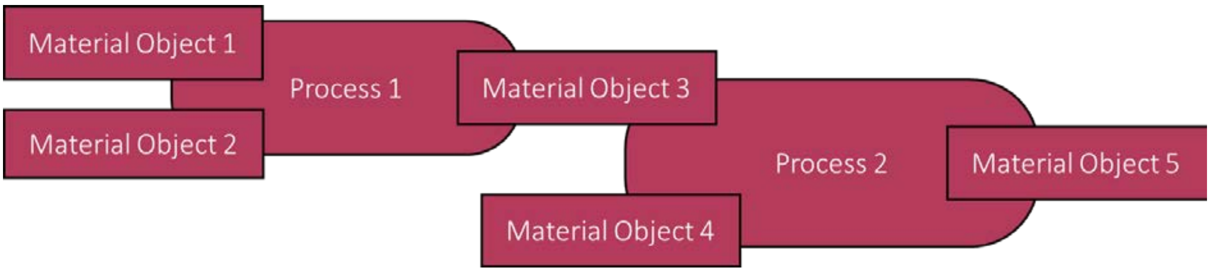

Figure 10. Example of process diagramming in the ontology development.

With the main components and processes are defined, they can then be coded in OWL using Protégé, structured into hierarchical taxonomies, and finally given the appropriate relations to frame the ontology. The following subsection describes the conventions the OWL implementation must consider to be compatible with EMMO.

4.4 Conventions for BattINFO Development

The BattINFO will ultimately be an integrated domain ontology of EMMO. As such it will abide by the conventions set out there. The governance of the EMMO along with a description of repository organization and conventions that contributors are expected to follow can be found in the EMMO repository on GitHub[10].

[10] https://github.com/emmo-repo/

4.4.1 Organizational Conventions

The organization of domain ontology repositories in EMMO is defined in the governance document[10] and replicated here:

- EMMO Domain Ontologies will be managed within separate repositories. These ontologies will follow the conventions outlined for EMMO and never duplicate any class or relation defined in EMMO top and middle level. Consistency and dependencies between these ontologies shall be managed via Domain Group meetings, liaising with EMMO Editor Group and decision making by EMMO Governance Committee where required.

- It is recommended that domain ontologies are organised hierarchically, such that more specialised domain ontologies will import generic domain-level concepts from less specialised domain ontologies. However, as stated above, detailed management of domain ontologies is beyond the remit of EMMO governance. Rather, general best practice should be applied.

Releases are versioned strictly according to the rules of semantic versioning as described on https://semver.org/. Branching the BattINFO repository shall abide by the following set of rules:

- Never pull to master. Master is only changed via pull requests from a release branch reviewed by the EMMO Editors Group. The master branch always hosts the current stable version.

- Each change of the master branch corresponds to a new release, with a unique semantic version number. All versions should be tagged with the version number prefixed with a “v”. For example, the tag for version 1.0.0 should be “v1.0.0”.

- All features should be associated with an issue.

- Features are developed in separate branches derived from a version branch and merged back via reviewed pull requests. Feature branches should be named “issue-”, where is the issue number and is a short message describing the feature.

- Before issuing, the developer should ensure that all unit and other tests pass. At least one EMMO Editor Group member must be assigned.

4.4.2 Naming and Structural Conventions

All OWL identifiers are unique IRIs of the following form:

http://emmo.info/<REPO>/<VERSION>/<PATH>#EMMO_<UUID>

where is the repository name, is the current version, is the path to the owl file in the repository (excluding the .owl file name extension) and is a unique UUID for the entity, usually assigned by Protégé or a similar tool. See Figure 3 for an example for how to configure Protégé to generate correct IRIs for new entities.

Class labels should be singular nouns and CamelCase. Labels for relations should be of the form “hasNoun” (i.e. lowerCamelCase and start with “has” followed by a noun). New relations (i.e. object properties) must be either mereotopological or semiotical and be a subrelation of any of the relations defined in EMMO Core.

4.5 First Use Case Definitions

To facilitate the early development of the ontology, a proof-of-concept use case will be defined. Having a first concrete use case will help to ensure that the ontology is robust and stable, as well as allow the development team to test the links to other WPs. The proof-of-concept use case is defined as a result of discussions with representatives from WPs 2, 3, 8, 9, and 11. The first step of establishing the strategy for ontology development will be the discussion of the use case with other WPs, especially with WP2, which is seen as starting point. Also, the WP9 will define use cases, which must be ontologized.

4.5.1 Electrode Structure Data

Figure 12 shows a schematic of the proposed proof-of-concept use case. Three-dimensional voxel data of battery material structures – whether it is generated experimentally from CT or FIB-SEM characterization or virtually through stochastic microstructure models [8], [9] - is (i) essential to both battery model development [10] and experimentally optimized material design, (ii) available as already existing datasets [11], and (iii) very large in size and somewhat difficult to manage. As such, it presents a good opportunity as a first use case for the ontology. An example of electrode structure data is shown in Figure 11.

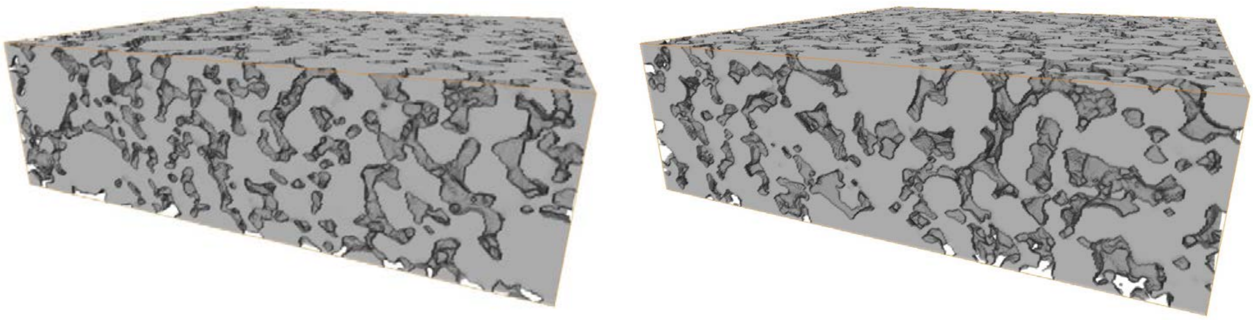

Figure 11. Example of electrode structure data, showing 3D visualizations of (left) a cutout from tomographic image data and (right) a stochastic model realization of a Li-ion battery negative electrode. Reproduced with permission from Ref. [9]**.**

WP11 currently has plans to implement two classes of models: deep learning models and generative models. The deep learning models aim to fit data from different scales to observe properties using explainable AI models. The development of the deep learning models requires e.g. data from battery material microstructures in a standardized format. Similarly, the generative models aim to generate date at multiple scales. The generative model must first learn from data that is physically correct (e.g. real microstructures).

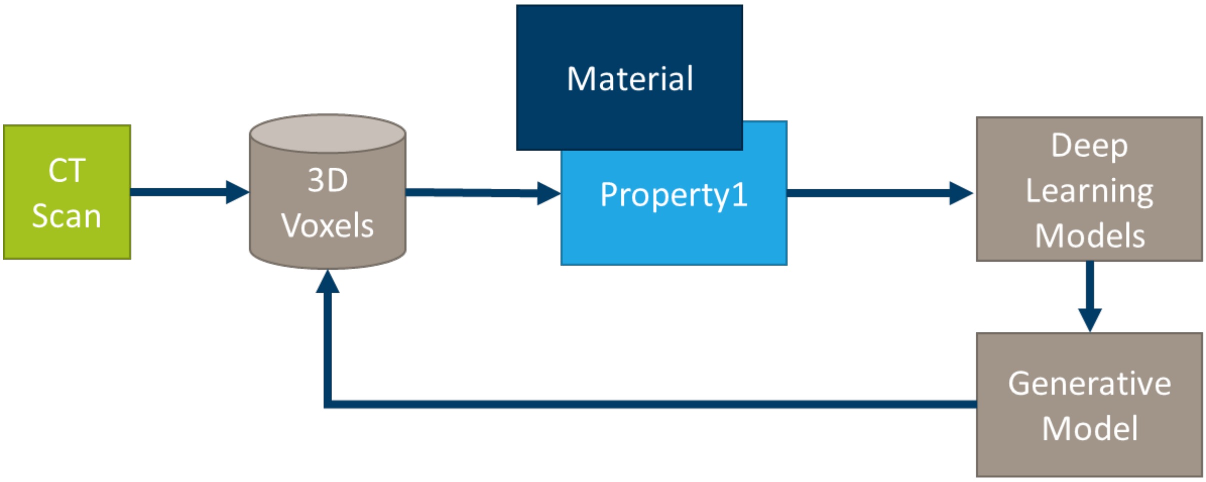

Figure 12. Schematic of the proposed proof-of-concept use case for the battery ontology. The ontology will facilitate access for deep learning models in WP11 to access 3D voxel data from CT scans of battery materials. The deep learning model can then be used to inform a generative model to generate virtual 3D voxel data of the same material.

4.5.2 Other Use Cases

Other early use cases that could help develop the BattINFO and contribute to work in BIG-MAP include model-based material property calculations [12] in WP2 and WP3, such as calculating potentials from DFT. Electrochemical data also offers a promising first use case for the ontology. For example, linking CV data with ab-initio molecular dynamics simulations to feed neural networks with the goal of training potentials. Further development of the concepts for early use cases will be refined with the leaders of the relevant WPs, including WP2, 3, 8, 9, 10, and 11.

5. Summary

This report summarizes the ontology standards and strategy that will guide the development of the Battery Interface Ontology (BattINFO) in the BIG-MAP project. The BattINFO will build on the standards defined by the top and middle-level ontologies developed in the European Materials & Modelling Ontology (EMMO). Battery performance and lifetime is determined mostly by process that occur at the battery interfaces. The BattINFO development will focus first on implementing an ontology for batteries generally and then focusing on battery interfaces specifically. The BattINFO will be implemented in the web ontology language (OWL) and built using the open-source editor Protégé. Initial development will be hosted on GitHub with the option to migrate to a unified BIG-MAP platform when one becomes available. To demonstrate the ontology, a first use case considering the tagging of 3D electrode structure voxel data to support the ongoing activities in WP11 will be demonstrated. Additional use cases including material property calculations and electrochemical data tagging will be discussed with the relevant WP leaders.

References

[1] J. Friis, B. T. Løvfall, E. Ghedini, G. Goldbeck, G. Schmitz, and A. Hashibon, “Documentation on Materials Modelling Ontology in UML,” Brussels, 2020.
[2] J. Gelb, D. P. Finegan, D. J. L. Brett, and P. R. Shearing, “Multi-scale 3D investigations of a commercial 18650 Li-ion battery with correlative electron- and X-ray microscopy,” *J. Power Sources*, vol. 357, pp. 77–86, 2017.
[3] A. Latz and J. Zausch, “Thermodynamic derivation of a Butler-Volmer model for intercalation in Li-ion batteries,” *Electrochim. Acta*, vol. 110, pp. 358–362, 2013.
[4] A. Latz and J. Zausch, “Thermodynamic consistent transport theory of Li-ion batteries,” *J. Power Sources*, vol. 196, no. 6, pp. 3296–3302, 2011.
[5] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu, “A review on lithium-ion battery ageing mechanisms and estimations for automotive applications,” *J. Power Sources*, vol. 241, pp. 680–689, 2013.
[6] B. Horstmann, F. Single, and A. Latz, “Review on multi-scale models of solid-electrolyte interphase formation,” *Curr. Opin. Electrochem.*, vol. 13, pp. 61–69, 2019.
[7] R. Scipioni, D. Isheim, and S. A. Barnett, “Revealing the complex layered-mosaic structure of the cathode electrolyte interphase in Li-ion batteries,” *Appl. Mater. Today*, vol. 20, p. 100748, 2020.
[8] S. Hein, J. Feinauer, D. Westhoff, I. Manke, V. Schmidt, and A. Latz, “Stochastic microstructure modeling and electrochemical simulation of lithium-ion cell anodes in 3D,” *J. Power Sources*, vol. 336, pp. 161–171, 2016.
[9] D. Westhoff, I. Manke, and V. Schmidt, “Generation of virtual lithium-ion battery electrode microstructures based on spatial stochastic modeling,” *Comput. Mater. Sci.*, vol. 151, no. February, pp. 53–64, 2018.
[10] J. Feinauer *et al.*, “MULTIBAT: Unified workflow for fast electrochemical 3D simulations of lithium-ion cells combining virtual stochastic microstructures, electrochemical degradation models and model order reduction,” *J. Comput. Sci.*, vol. 31, pp. 172–184, 2018.
[11] T. M. M. Heenan *et al.*, “Data for an Advanced Microstructural and Electrochemical Datasheet on 18650 Li-ion Batteries with Nickel-Rich NMC811 Cathodes and Graphite-Silicon Anodes,” *Data Br.*, vol. 32, p. 106033, 2020.
[12] S. Bhattacharyya, R. Sahara, and K. Ohno, “A first-principles phase field method for quantitatively predicting multi-composition phase separation without thermodynamic empirical parameter,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–2, 2019.