D1.3

Report on INTERSECT developed ontologies and MODA

Matthias Büschelberger, Joana Francisco Morgado, Kathrin Frei,
Christian Eichheimer, Jana Boehm, Arrigo Calzolari, and Adham Hashibon

Document information

Project acronym:	INTERSECT
Project full title:	Interoperable Material-to-Device simulation box for disruptive electronics
Research Action Project type:	Accelerating the uptake of materials modelling software (IA)
EC Grant agreement no.:	814487
Project starting / end date:	1st January 2019 (M1) / 31st January 2022 (M37)
Website:	www.intersect-project.eu
Final version:	27/07/2021
Deliverable No.:	D1.3
Responsible participant:	Fraunhofer (participant number 5)
Contributing Consortium members:	CNR, FRA, AMAT
Due date of deliverable:	31/07/2020
Actual submission date:	29/07/2021
Dissemination level:	PU - Public

Authors: Matthias Büschelberger, Joana Francisco Morgado, Kathrin Frei, Christian Eichheimer, Jana Boehm, Arrigo Calzolari, and Adham Hashibon

To be cited as: M. Büschelberger, J. F. Morgado, K. Frei, C. Eichheimer, J. Boehm, A. Calzolari, and A. Hashibon (2021): Report on INTERSECT developed ontologies and MODA. Deliverable 1.3 of the H2020 INTERSECT project (final version as of 27/07/2021). EC grant agreement no: 814487, Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V., Freiburg, DE.

Disclaimer:

Versioning and Contribution History

Version	Date	Modified by	Modification reason
v.01	22/01/2020	Joana F. Morgado and Adam Hashibon	First Version
v.02	30/01/2020	Arrigo Calzolari	Revision
v.03	01/07/2021	Matthias Büschelberger	Main revision and update to new EMMO release
v.04	20/07/2021	Arrigo Calzolari	Revision
v.05	23/07/2021	Matthias Büschelberger and Kathrin Frei	Update MODA-chapter and individual figures

Content

[1] Acronyms are marked in purple in the text and defined at the end of the document.

1. Executive Summary

The strategy used for the INTERSECT ontology development, and for the foundation of the ontology-based data structures (Common Universal Data Structures - CUDS) described in D2.5 is documented in this report. The developed INTERSECT ontology is compliant with the European Materials Modelling Ontology (EMMO) that aims at being the European standard in representing all aspects of materials modelling and characterization, and can also be applied to other domains. This report documents the initial EMMO extension towards the Material-to-Device (M2D) application field. In this deliverable, we provide information about the generic ontology application developed for the INTERSECT Interoperable Material to Device (IM2D) simulation toolbox. The INTERSECT specific ontology-based data structures populate a dedicated ontology module (namely "im2d_toolbox") that is compliant with the EMMO 1.0.0-beta version on the public GitHub. The presented ontology is currently being used in the AiiDA wrapper development for semantic interoperability and it is being enhanced since new extensions are required in order to represent all the IM2D workflows.

An original version of this document has been delivered at M13 of the project (January 2020). This revised version updates the INTERSECT-driven ontology development to the latest version of the EMMO (version 1.0.0-beta) that has been officially released after the publication of the original D1.3 document. Newest INTERSECT-related ontology features implemented in the last year have also been included to update and improve the original work.

2. Description of the work done

The interoperability Hub (iHub) developed within the IM2D box provides rich standardized metadata that enables a seamless communication and integration among different components of the platform, and other external repositories and marketplaces. Moreover, it enhances the consistency of information (by adopting the same standards), and will favor the re-use of existing data. The semantic interoperability is achieved in INTERSECT by adopting a shared ontology that is compliant with the EMMO developed within the European Material Modelling Council (EMMC), and that functions as a common language understood by all the components of the platform, and by any other EMMO compliant engine.

Ontologies are the backbone of semantic web services, and, according to Gruber (1993) [1], can be defined as *an explicit, formal specification* of a *shared conceptualization*. This means that expressing knowledge through an ontology requires:

- an explicit, well-defined description of all the concepts and their meaning based on rules of syntax, context and pragmatics;
- the use of a formal language, ontology modelling language like the Web Ontology Language (OWL, used by EMMO), that makes the described concepts machine readable;

- shared concepts, and agreed terminology.

The current report documents the strategy used for the ontology development but also its application on a specific feature to be handled by the IM2D box.

It is important to note that the ontology development is a thorough iterative process with several feedback loops. The development of INTERSECT ontologies follows the steps below:

Analysis of the domain

Currently, the workflows of the post.BandGap, post.DielectricConstant, and post. Defects from the AiiDA-post package, which run all available major computations for the IM2D toolbox in this project, have been chosen as starting point for the INTERSECT ontology development work.

Identification of the relevant entities and individuals

The identification of the relevant entities for the INTERSECT ontology is not independent from the application. Therefore, the identified entities are based on requirements to execute each specific AiiDA workflow. Materials Modeling Metadata (MODA) documentation has been used as starting point for the identification of these entities and has been complemented by further code-specific information from input and output files.

Create the taxonomy and identification of other relations among the entities

Once the required entities have been identified they are classified in a hierarchical way by defining an explicit hierarchy of classes and subclass relationships of ***is_a*** (equal to rdfs:subClassOf) type (taxonomy). This classification defines the main skeleton of the ontology branch to each of the other entities (classes or individuals) and relationships (such as parthood and connection relations – primitive relations in mereotopology).

Populate the ontology with additional entities and relations

Additional entities or classes can be merged to the developed taxonomy by different kinds of relationships (including subclass relationships) and axioms.

Details of ontology development are on the application or software development procedures. Therefore, the presented ontology is continuously extended and tailored to the Quantum ESPRESSO (QE) standards. However, the ontology structure can be straightforwardly extended to the implementation of other codes (e.g., SIESTA). The goal is to build a generic module that can satisfy the description of any other code. The ontology development procedure, and its application by using the EMMO-based data structures CUDS, is described in the following sections.

INTERSECT partners are involved in the EMMO development, contribute to the EMMO implementation and validation, and work on its extension for the electronics application field. Since October 2020, INTERSECT partners have been attending weekly collaborative hackathon sessions together with other participants from DT-NMBP-09 sister projects. This allowed to extend the EMMO core 1.0.0-beta version to the overall entities needed for the

domain ontology describing the field of quantum mechanics and Density Functional Theory (DFT, see public GitHub-branch: https://github.com/emmo-repo/EMMO/tree/DT-NMBP-09).

More specific implementations for the IM2D-ontology are hosted on a private GitLab-repository (https://gitlab.cc-asp.fraunhofer.de/ontology/applications/intersect/im2d-electronic-calculations/-/tree/2-using-only-emmo-1-0-0-alpha2).

3. Deviation from planned work in the DoA

The main deviation to the plan of work in the DoA relies on the fact that the presented ontology addresses only one single feature within the IM2D workflows. Delays and a massive revision of the EMMO core top-level development (independently from INTERSECT), together with the setup of a common ontology development environment, are the main reasons for such a deviation and make the documented ontology a starting point for further developments in INTERSECT. Indeed, the construction of the EMMO itself is still in progress and involves several groups.

4. Results

The outcomes of the ontology development are separated into several parts which will be presented in the following subsections.

Please note: In the following graphs

- greyish boxes represent classes that were already present in the upstream EMMO v1.0.0-beta master branch (https://github.com/emmo-repo/EMMO/tree/master - last update: 3rd of March, 2021);
- yellowish boxes represent new classes on the downstream DT-NMBP-09 branch (https://github.com/emmo-repo/EMMO/tree/DT-NMBP-09 - last update: 25th of February, 2021);
- orange boxes are concepts already available in the isolated but unmerged EMMO-atomistic, and EMMO-Crystallographic Information File (CIF)-ontology (https://github.com/emmo-repo/domain-atomistic/tree/emmocheck_conf - last update: 18th of March, 2021 and https://github.com/emmo-repo/CIF-ontology/tree/cif-data – last update; 12th of March, 2021),
- the original EMMO taxonomy (grey boxes) is displayed in an oversimplified manner in some graphs in order to save space.

The following chapter is organized in different parts having varying relevance in relation to semantic interoperability formulation (D1.1) and implementation (D2.5). Section 4.1 describes the MODA requirements for standard IM2D calculations. The top-level classes (Section 4.2) are fundament for a large range of mid-level classes described in Sections 4.4-4.7. Section 4.3 covers newly introduced object properties. The material-related domains (Section 4.5) fulfill a more crystallographically interpreted, and software-agnostic description

of the computed properties. INTERSECT specifically dedicated classes concerning the software-syntax, and its interpretation through EMMO are introduced in Section 4.6 in order to meet different *persona* requirements and expertise levels (D1.1), and in order to control the computational outcomes of the AiiDA-QE and AiiDA-SIESTA workflows. Section 4.7 handles parts of the ontology which have been identified (*a posteriori*) as non-functional for the wrapper-execution (D2.4) but that have a potentially valuable side-information for the knowledge-transfer of processes in the DFT-codes.

4.1 Recapitulation of developed MODA

Figure 1 shows the MODA diagram describing a coupling-and-linking electronic model's chain (DFT) for the calculation of the defect formation enthalpy in crystalline solids (see D1.1). Figure 2 shows a simplified representation of a workflow for the band gap energy calculation, and Density of State (DOS).

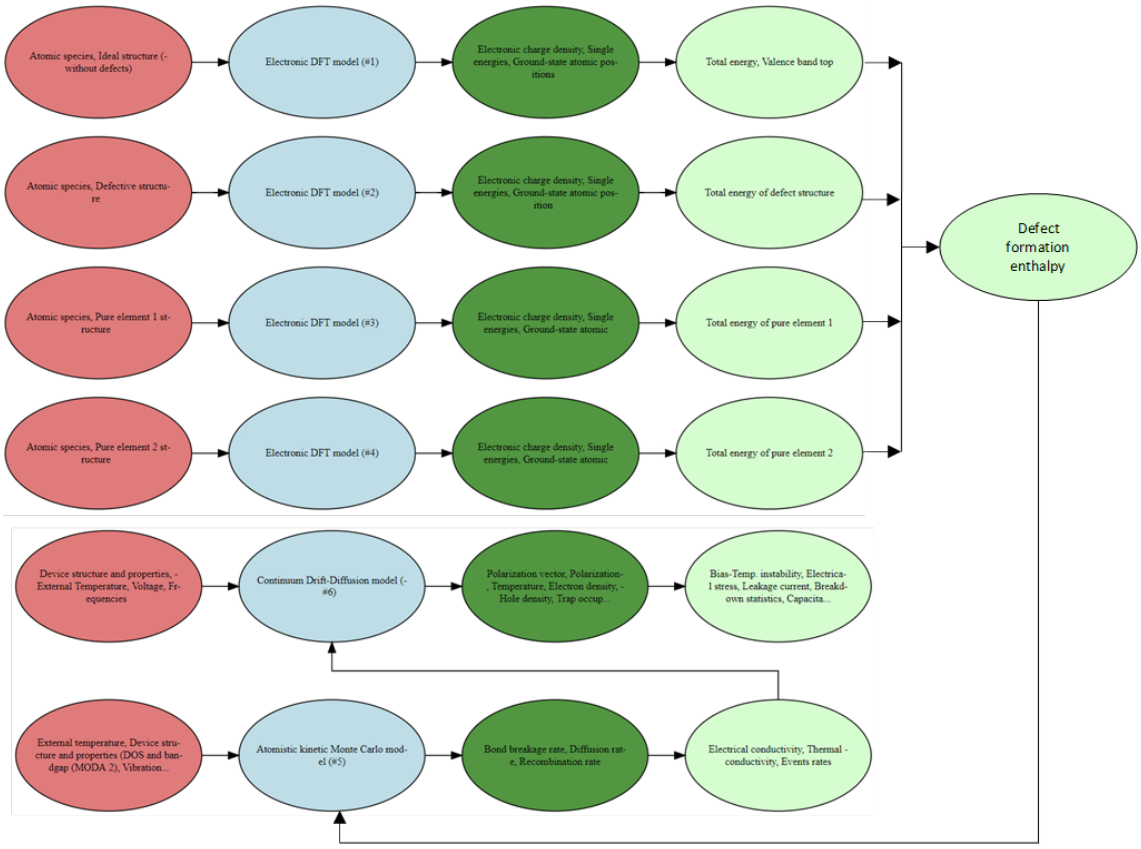

Figure 1: MODA workflow diagram for the evaluation of defect formation energy, within the M2D design modelling approach.

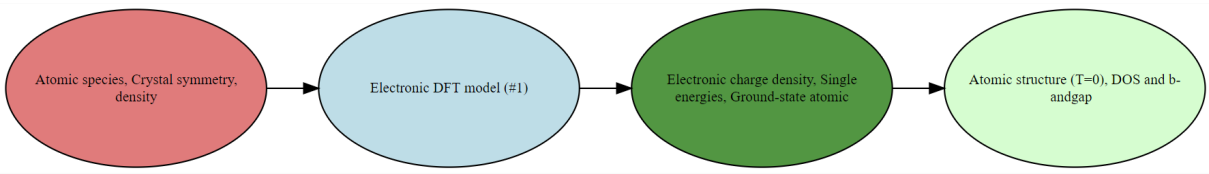

Figure 2: MODA workflow for the DOS modelling and the band gap energy.

The developed MODA forms the basis for the fundamental understanding of the entities needed to describe the use cases and the ontology development in the Protégé code. The according diagram (Figures 1 and 2), as well as the following tables (Tables 1, 2, and 3), hold piecewise information about potential key performance indicators (KPIs), fundamental equations, and properties of interest as output of the computations. Remarkably, most of these concepts from the MODA tables and workflows were not available in the EMMO at the beginning of the ontology development, hence had to be categorized and implemented into OWL.

However, since there is limited information about potential requirements or inputs for different user expertise, additional analysis and literature research by the INTERSECT consortium have been unavoidable in order to understand potential requirements in relation to different familiarities in the model-complexity.

Since the corresponding codes of AiiDA-defects and AiiDA-POST (see D2.4) have been continuously maintained and updated since the MODA-diagram from Figure 1 was created in 2019 (D1.1), the focus has been set on the investigation of the code itself. The inputs of the host-, defect-, and unit cell-structure remained as the main inputs but they have also been extended to further KPIs, such as a correction scheme in the defect formation energies, dielectric constant, defect charge, etc. It has to be mentioned here that these AiiDA-packages are unreleased repositories, and are still in a partial development/validation stage. The direct collaboration among developer partners allowed us to ontologize the relevant entities and concepts within these computations.

As a starting point, the focus has been set on the basic workflows of the band gap and band structure calculation, since the corresponding AiiDA-plugins have already a stable release on public GitHub.

Notably, MODAs do not specify all the possible information about material-scientific relations and high-level crystallographic descriptions. In the following chapters, only a small set of parameters has been considered in terms of ontology formalization. This work is in progress to enlarge the list of parameters, and their numerical definitions.

1	**ASPECT OF THE USER CASE/SYSTEM TO BE SIMULATED**	
1.2	**MATERIAL**	*Semiconductor materials such as ferroelectric HfO_2, ovonic chalcogenides (GeSe, GST) with various distributions of defects.*
1.3	**GEOMETRY**	*3D bulk crystalline and amorphous phases*

1.5	**MANUFACTURING PROCESS OR IN-SERVICE CONDITIONS**	*External temperature (T1) in the range of 300 to 700 K*

Table 1: MODA - Aspect of the user case/system to be simulated.

2	GENERIC PHYSICS OF THE MODEL EQUATION		
2.1	MODEL TYPE AND NAME	*Electronic Models - Density Functional Theory*	
2.2	MODEL ENTITY	*Electron*	
2.3	MODEL PHYSICS/ CHEMISTRY EQUATION PE	**Equation**	*Kohn–Sham equation*
		Physical quantities	*Electronic charge density; Single particle energies; Ground-state atomic position*
2.4	MATERIALS RELATIONS	**Relation**	*Effective potential; Exchange correlation potential; External potential; Hartree potential*
2.6	SIMULATED INPUT	*Atomic species, Defective structure*	

Table 2: MODA - Generic physics of the model equation.

3	SOLVER AND COMPUTATIONAL TRANSLATION OF THE SPECIFICATIONS	
3.1	NUMERICAL SOLVER	*Iterative self-consistent field (scf), eigensolver for DFT problem, Davidson or Conjugated gradient - electronic structure BFGS quasi-newton algorithm - optimization of atomic structure*
3.2	SOFTWARE TOOL	*Quantum-Espresso and SIESTA*
3.5	COMPUTATIONAL BOUNDARY CONDITIONS	*Periodic boundary conditions*
3.6	ADDITIONAL SOLVER PARAMETERS	*Atomic structures are relaxed until forces on all atoms become lower than 0.03 eV/Ang-1. Kinetic cut off and PAO basis set depend on specific material simulation.*

Table 3: MODA - Solver and computational translation of the specifications

4.2 Top level classes

Before introducing the mid-level classes for the description of the application, and its physical interpretation in terms of quantum mechanics, more generic entities have been introduced (see Figure 3). ***Calculation*** and ***Computation*** have been added as subclasses of a process and therefore have ***Software*** as participant. Another important entity stands for the ***Workflow*** that is the subclass of an ***Algorithm,*** which is the subclass of an ***Icon***. The ***Icon*** is a semiotic entity in the EMMO top level that describes the interpreted meaning of a real-world object, here called ***Physicalistic***. A ***Physicalistic*** is experienced through a ***Perceptual*** within a ***Process*** by the ***Interpreter***. In this process, the interpreted ***Sign***, which is the superclass of the ***Icon***, and the ***Interpreter*** are acting as ***Participant***. However, depending on the process, also a ***Perceptual*** or a ***Physicalistic*** can take the role of a ***Participant,*** without being taxonomically a ***Holistic.*** This is expressed via the relationships of ***hasParticipant*** or ***hasSign***, as well as via their according subclass-relationships. Both the ***Chemical*** and the ***Crystallographic*** classes are considered as a ***Language*** and therefore as a ***Perceptual***. The reason of this choice relies on the capability of EMMO to understand the according subclasses as terms and concepts (e.g. ***UnitCell***, ***ChemicalFormula***), which are formalizations and vocabularies for portraying and concretizing real-world objects within a specific field of science.

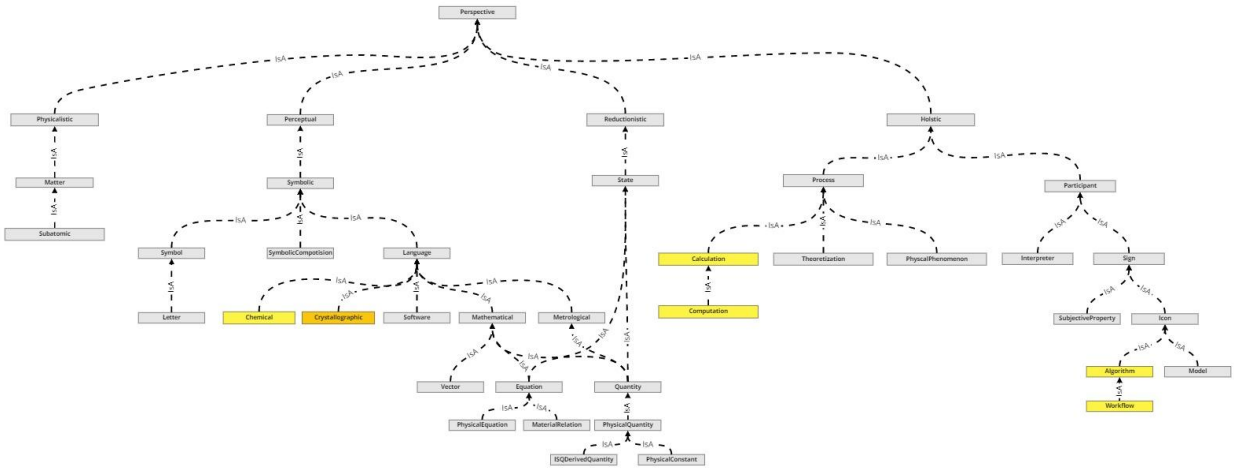

Figure 3: Top level entities needed for further description of the IM2D-workflows.

4.3 Object properties (relationships)

In the OWL language, the relationships are called object properties, and are linking classes and individuals whose interactions are related by logical rules. In EMMO, there are two main kinds of relationship for the domain ontology: the ***semiotical*** relationship through ***hasSign,*** and the ***mereological*** relationship through ***hasPart***.

The former is used to describe the meaning of an entity in relation to another one, and without being directly connected to each other.

For example, a ***Computation*** is executed as ***Participant*** by a computer using a ***Fortan***-code running an implementation of a ***PhysicsEquation***. Hence, the equation is not in its original perceptible form but in the representation of the code syntax. Therefore, the equation is neither a direct ***Participant*** nor a direct part of the ***Process,*** and it has to be considered as another interpretation in this assertion. In order to give the code a meaning of a ***PhysicalEquation***, its perceptible mathematical formulation is set as a ***Sign***, standing for the actual code implantation via the ***hasCalculatingEquation***-relationship (see Figure 4 where this relation is asserted to be a subclass of the ***hasIcon*** and ***hasSign***-relation). This procedure allows the equation to be connected to another interpretation without changing its superclass from ***Perceptual*** to ***Sign***.

The second important top level object property expresses the parthood of an entity with ***hasPart***, and with all of its subclass-relations. This holds true for all ***Perceptual*** and ***Physicalstic*** that can be considered as being connected in the real-world.

For example, the mathematical logic of a ***PhysicalsEquation*** can include a ***PhysicalQuantity*** via the ***hasPhysicalQuantity***-relation, which is a subrelation of the ***hasSpatialDirectPart.*** It is important to notice that the logical comprehension of a ***PhysicsEquation*** and of its ***PhysicalQuantity*** are separated from the ***SymbolicComposition*** and the ***Symbol*** in its pure written form, e.g., the symbol for π. A way to establish this connection would be via the ***hasFormulaSymbol***-relation. Similar arguments hold for the numerical data that can be asserted for a quantity, e.g., 3.14 for π. This is achieved by the ***hasQuantityValue*** (already available in the EMMO v1.0.0-beta version, and not displayed in Figure 4).

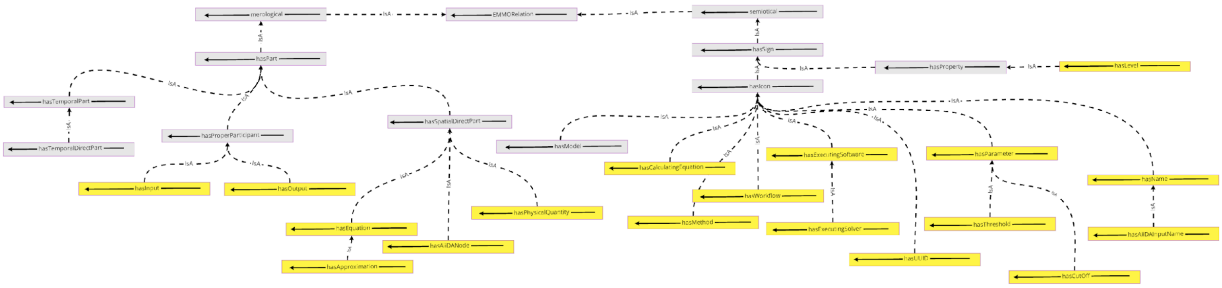

Figure 4: Newly introduced relationships (object properties).

4.4 Generic material-related classes

Coupling of material entities

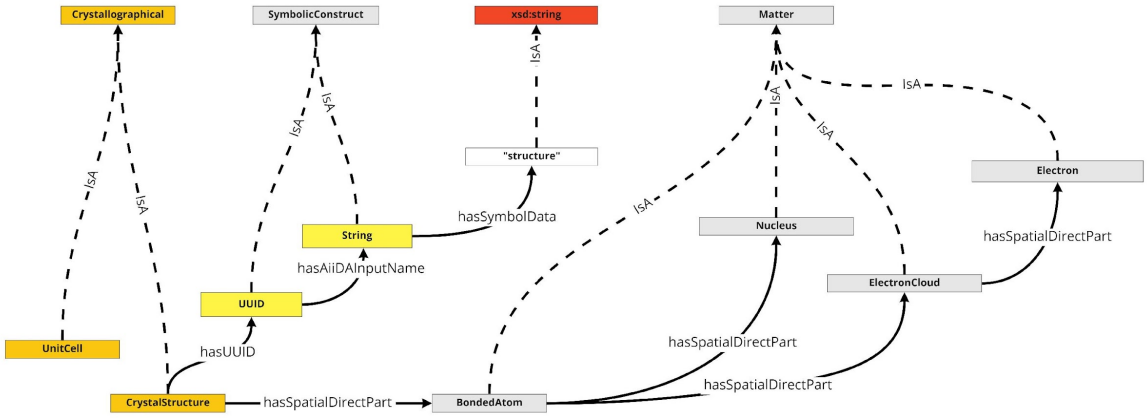

Figure 5: Coupling of material entities.

A minor work has been done on the description of the material entities, such as ***CrystalStructure*** and ***ElectronicBandStructure*** (see Figure 5). One reason for this choice is that the structure of crystals and their chemical composition are already syntactically described by the StructureData from the MaterialsCloud-database of AiiDA. They might potentially be described on a high-level (e.g., crystal system and chemical formula the host structure) in the future. The most generic link is currently realized through the Universally Unique IDentifier (UUID) index (Universally Unique Identifier), which is assigned by AiiDA when a CIF-file is imported or uploaded from the Crystallography Open Database (COD), and the ***"structure"*** input of the submitted workflow for AiiDA (see D2.4 and D2.5). Other EMMC activities are currently working on the CIF-files ontologization (see the public GitHub: https://github.com/emmo-repo/CIF-ontology), therefore they might be coupled through the semantic OSP-interface afterwards if needed.

The activity planned for the upcoming months includes the definition of further classes for a proper crystallographic description of the simulated properties of materials. One relevant example is reported in Figure 6, which shows the conceptualization of an atomic ***Defect*** located in the ***CrystalLattice*** of a certain ***CyrstalSystem*** with a certain ***ChemicalFormula***. These entities are considered to be a subclass of ***Crystallographic***. More specifically, the kind of ***CrystalSystem***, the kind of ***Defect*** and the defect species will be defined in the next steps of the development. This ontology part will be merged afterwards with the semantic representation of the ***ElectronicBandStructure***. The ***DefectFormationEnthalpy*** is assigned via the ***hasQuantitativeProperty***, which is a semiotical object property. Overall, this whole domain specification will be the most important part to be developed in the semantic data model within the end of the project.

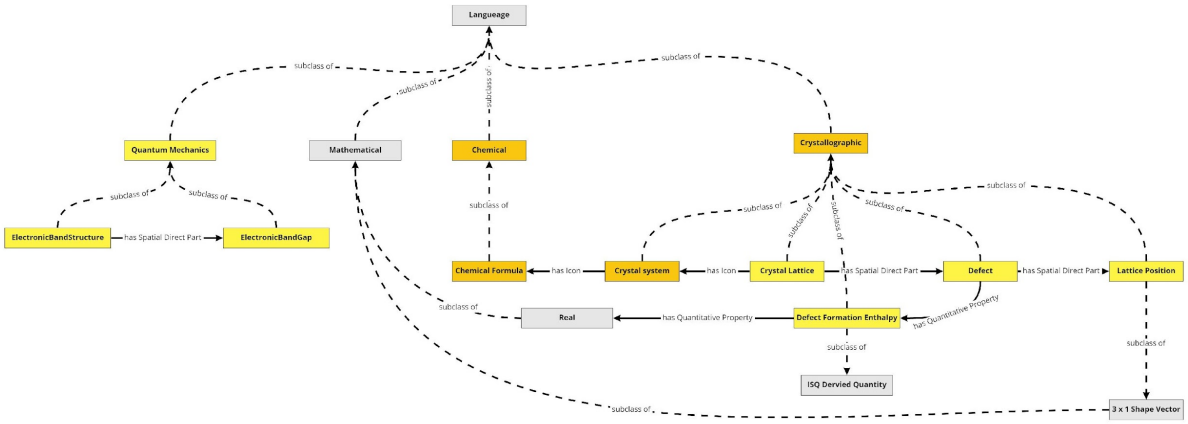

Figure 6: Crystallographic entities and descriptions for defect formation enthalpy and band structure models.

Continuum Drift Diffusion Model

The ***ContinuumDriftDiffusionModel*** (Figure 7) is one of the major workflow drivers within GINESTRA™ and enables the numerical modelling of current characteristics in disruptive electronic devices. Quantitative properties of the ***CrystalLattice*** such as the ***DefectDensity***, the ***ElectricCharge*** of a ***Defect***, the ***TotalElectronicEnergy***, and the ***DefectDistribution*** are generally considered as some of the most important parametric inputs (see D1.1) to calculate KPIs like the ***Capacitance***, the ***ElectricConductance***, and the ***ElectricPotential*** of a simulated ***ElectricCurrentFlow*** through an electronic ***Device*****.** The semantic data model might be enriched by additional entities describing the mesoscopic composition and structure, as well as the type of device (e.g. PCR, RRAM, FET, selectror, capacitor, etc.).

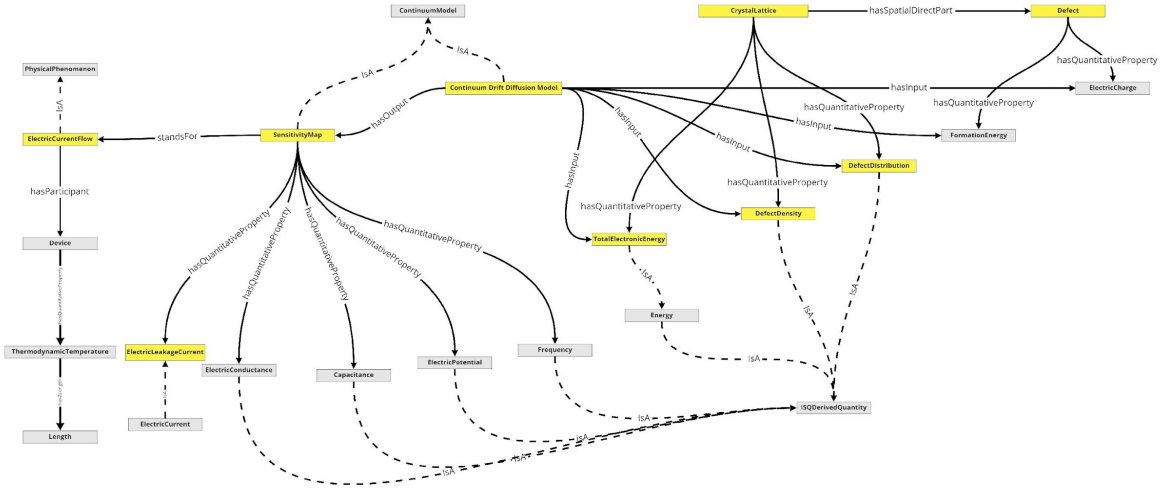

Figure 7: Rough description of the continuum drift diffusion model with its in- and output entities.

4.5 Software-related low- and mid-level classes

Taxonomy of Quantum Espresso Inputs

In order to meet the *persona* requirements for intermediate and advanced user-experience with DFT-computations within the IM2D, it is necessary to derive more mid- to low-level classes, which control the accuracy of the calculated properties (i.e., setting of input parameters). Since QE and SIESTA assume different Kohn-Sham-Equations implementation schemes (e.g., basis set, numerical algorithms, optimization solvers, etc.), these entities are intended to be code-specific, and hence they need to be separated in the logical axioms of the ontology.

The corresponding low-level classes relative to QE (Figures 8-10) have been implemented and include different ***Occupation***-methods, ***SpinPolarization***-types, and ***GroundStateApproximationParameters***. A few of those, such as the ***KineticEnergyCutOffs***, the ***ForceConvergenceThresholds*** and ***SelfConsistencyConvergenceThresholds*** are considered the most important accuracy-drivers for QE. These refer to physical quantities, such as ***KineticEnergy***, ***IonicForce*** and ***ChargeDensity.*** In addition, other entries related to the physical dimensions and the units of the single properties (e.g. ***HarteeAtomicUnits,*** ***RybergAtomicUnits***, ***RydbergUnitofEnergy***, ***AtomicUnitofForce***, ***AtomicUnitofEnergy*** and ***PhysicalDimension***) need to be introduced (Figure 11). The next steps will consider a similar set of parameters for SIESTA.

It is important to note that these input descriptions are code-specific, and they are valid for specific versions of the software only. They might be described through semantic entities introduced in the following chapters.

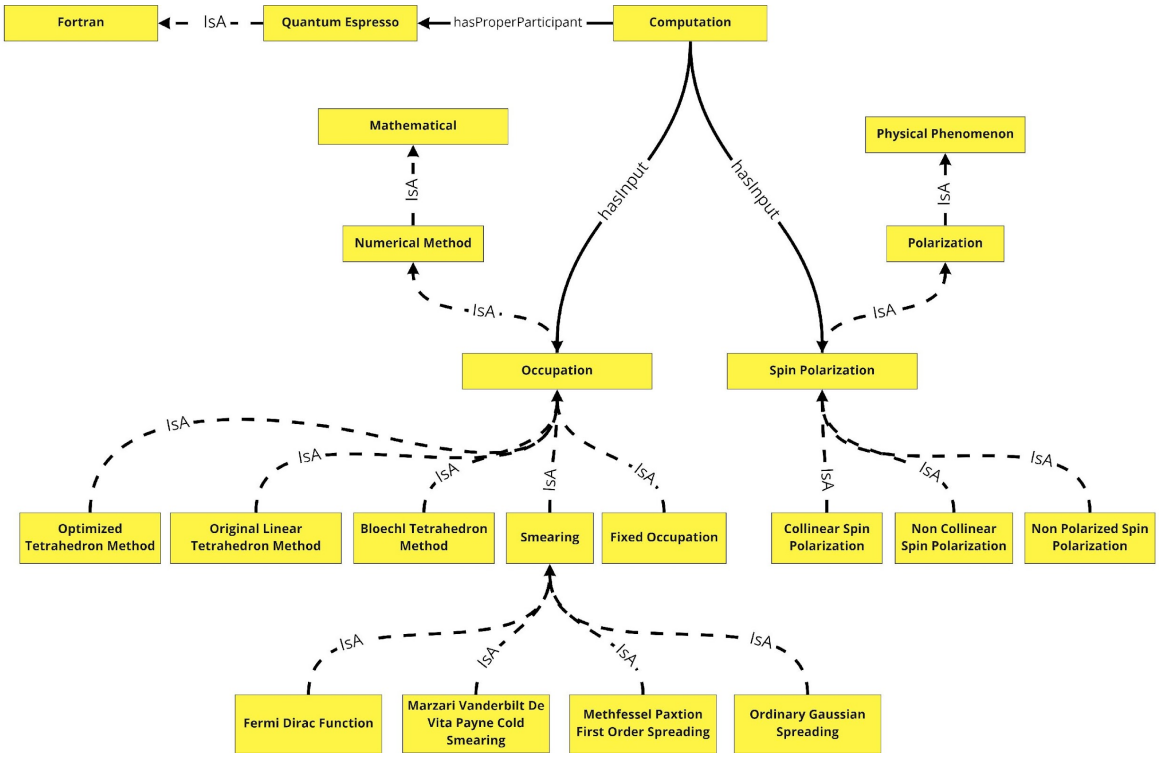

Figure 8: Taxonomy input parameters for QE (part 1 of 3).

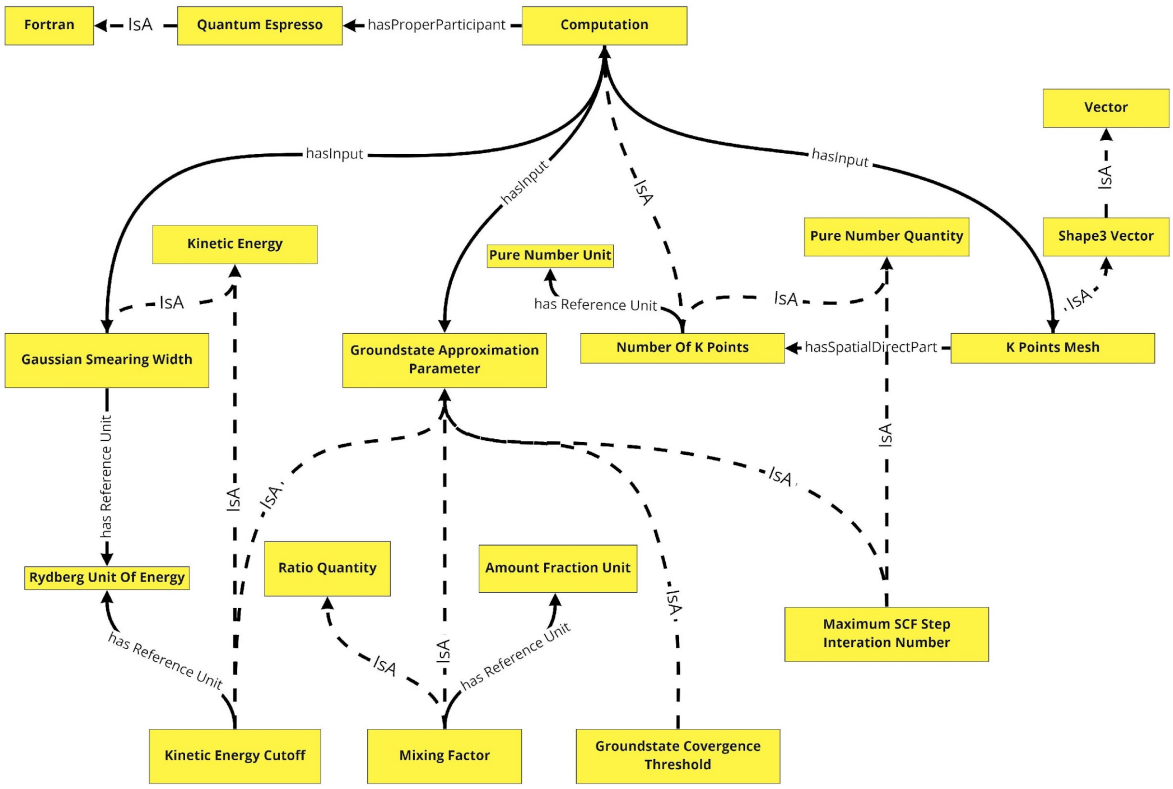

Figure 9: Taxonomy input parameters for QE (part 2 of 3).

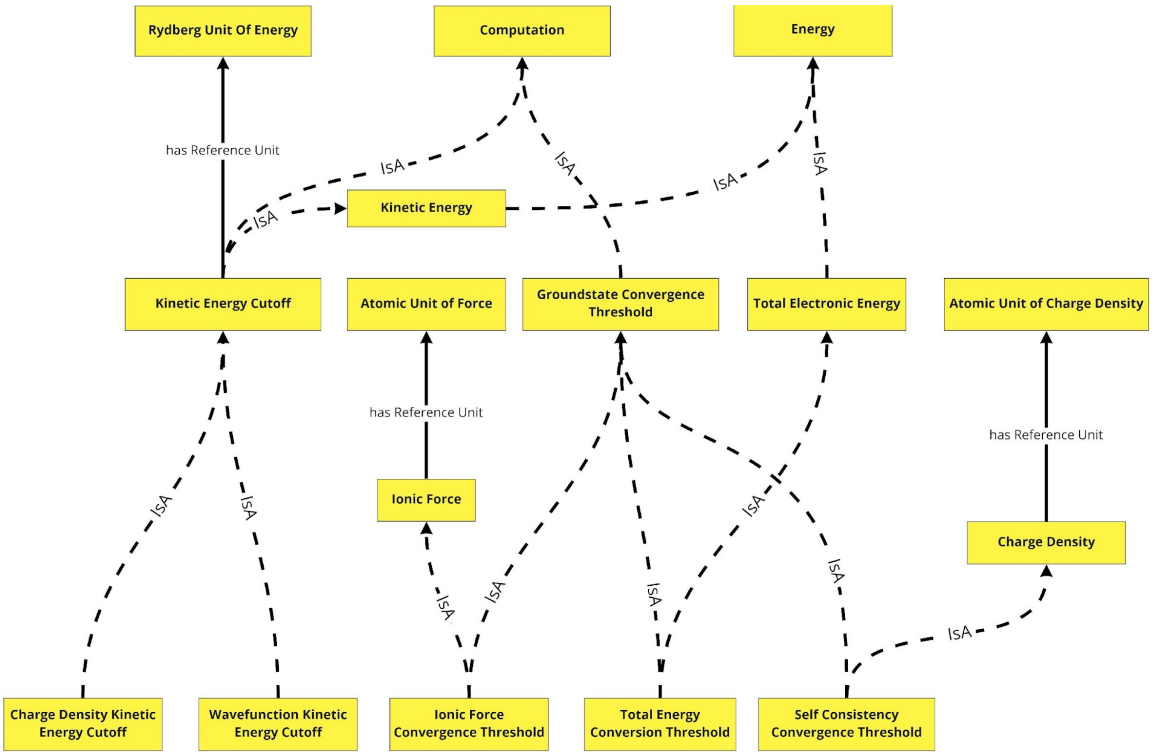

Figure 10: Taxonomy input parameters for QE (part 3 of 3).

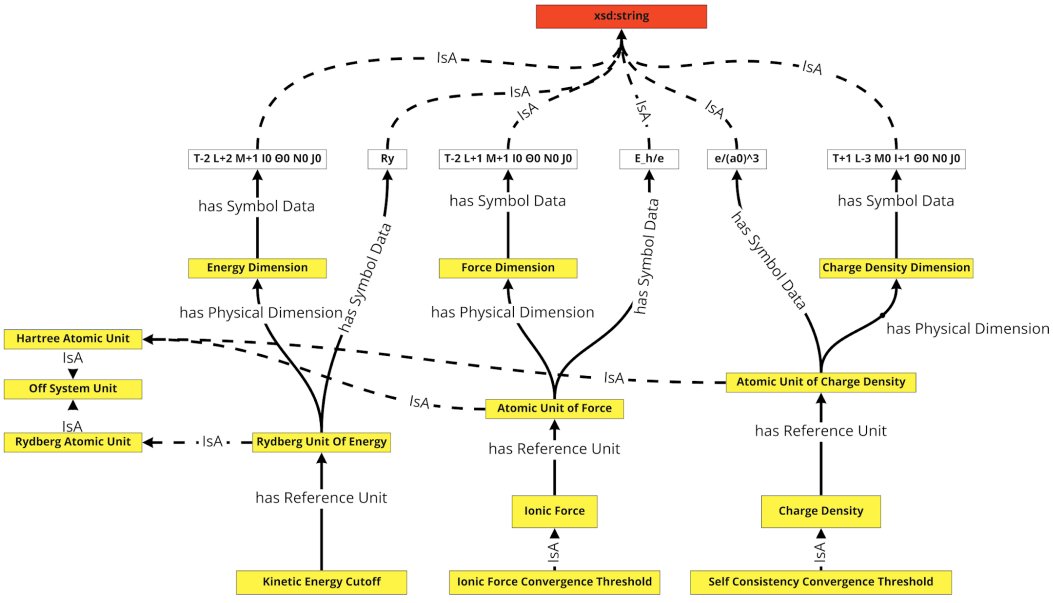

Figure 11: Units and physical dimensions of Quantum ESPRESSO input parameters.

4.6 Customization to user profiles (persona)

User profile and knowledge levels

One of the major uses of the IM2D-ontology is the introduction of several knowledge levels relative to the *persona*'s experience. The user levels are categorized into ***Basic***, ***Intermediate*** and ***Advanced*** (Figure 12), which correspond to different levels of complexity in the definition/access of the input parameters. This gives to basic users the opportunity to manage only a minimal set of input parameters (e.g., chemical species, compound name, crystal structure, etc.), while it allows more experienced users to access a set of more advanced parameters. The individual set of parameters for each AiiDA-workflow shall also be chosen in view of their influence on the quality of the simulation results. Default values will be queried from the CIF-Import workchain from AiiDA.

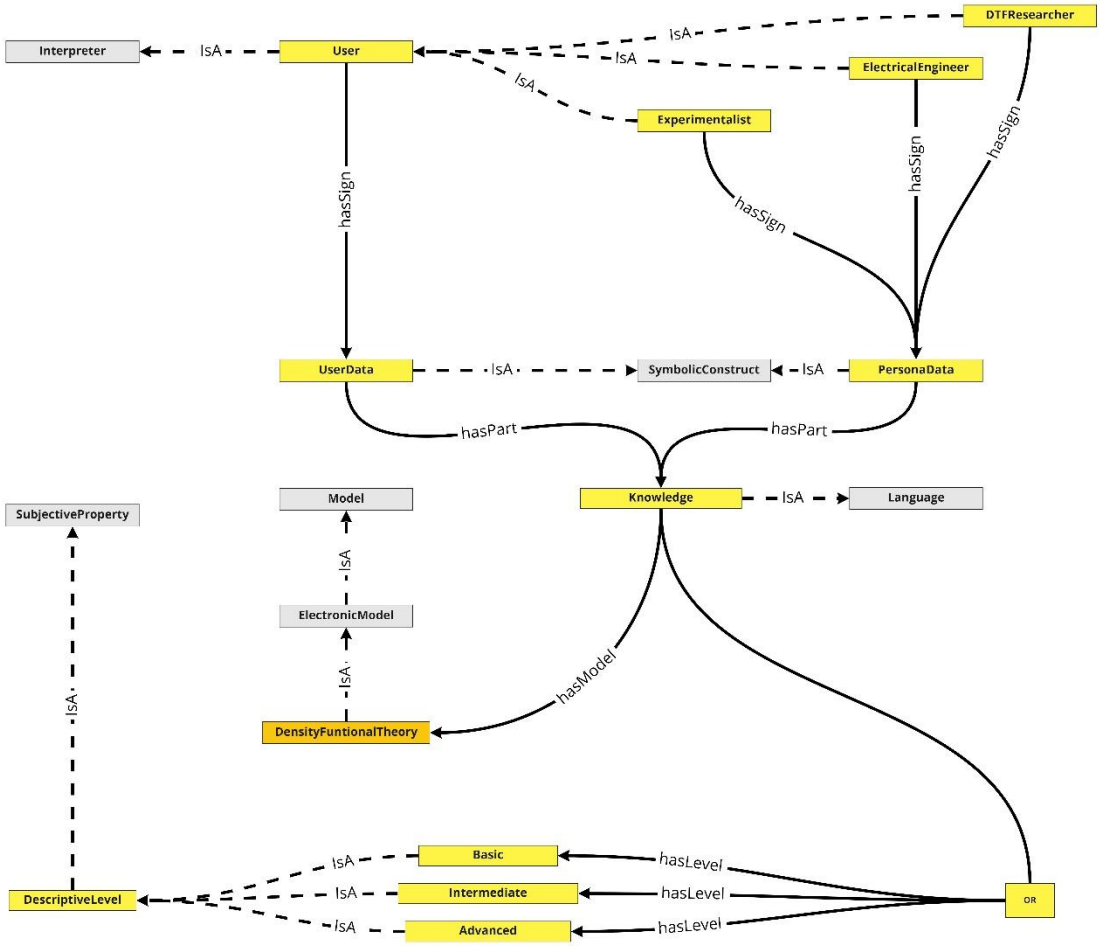

Figure 12: Ontologization of user profiles as a function of the knowledge level relative to DFT simulations.

Example for band_gap.pw-workflow

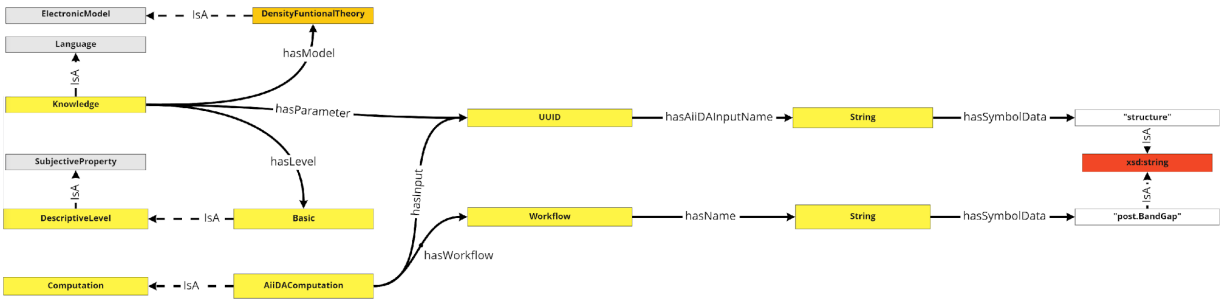

Figure 13: Basic knowledge level with AiiDA parameters for post.BandGap-workflow.

This subsection provides an example on how the parameters of AiiDA are connected to ontology through a ***String*** pointed by the ***hasAiiDAInputName***-relation. Accuracy-related quantities (e.g. ***Maximum***, ***Threshold***, ***CutOff***) are linked by the ***hasInput***-relation to the according ***AiiDAComputation*** (with a ***Workflow*** having a name through ***String***, such as ***"post.BandGap"***) and to the selected knowledge level through the ***hasParameter***-relation. The most relevant AiiDA-parameters have been described in the previous subsections. Other workflows, such as the ones for the DOS and the formation energy, can also be customized according to the user's level of knowledge. On the contrary, a few advanced workflows (such as those for the defect formation energy, and for the Nudged Elastic Band - NEB calculations for atomic defects), which are at the-state-of-the-art of quantum modelling, are too complex to be blindly managed by basic and intermediate users. For the time being, their use is restricted to advanced users only. The main advantage of the knowledge level and input-parameter representation in Figures 13-15 is that the class schema is distinctively flexible, and individual entities can be simply exchanged without further causing any maintainability-issues for the OSP-interface.

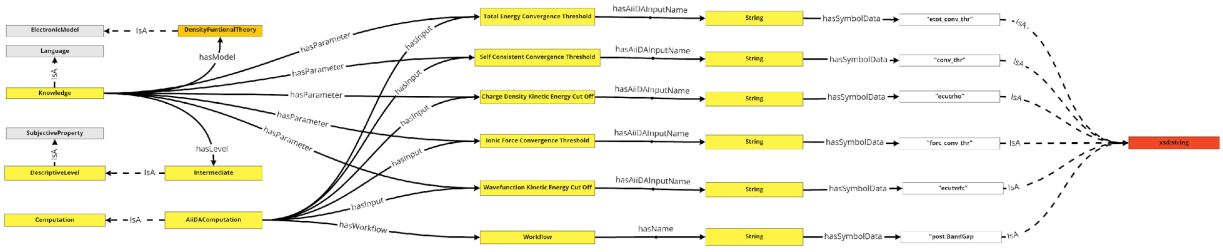

Figure 14: Intermediate knowledge level with AiiDA parameters for post.BandGap-workflow.

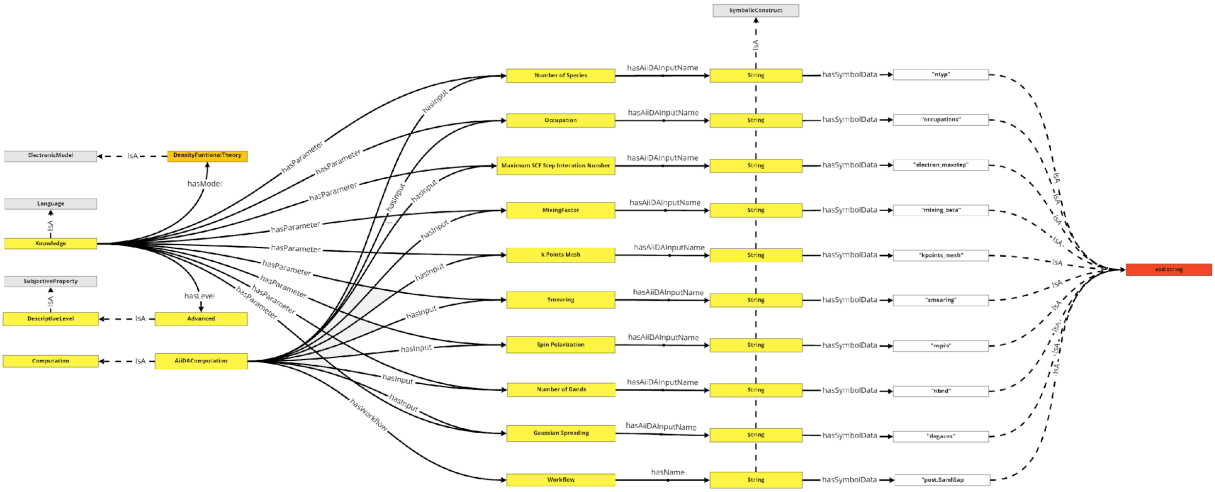

Figure 15: Advanced knowledge level with AiiDA parameters for post.BandGap.

4.7 Additional (optional) classes

Even though developed for the specific case of QE code, most of the classes described in this section are common to any DFT simulations and can be straightforwardly extended to SIESTA.

Calculation entities

The most important results of a DFT simulation are the charge density, the total energy, the atomic forces, the single particle energies, and the band structure. The related entities are shown in Figure 16 and are linked to a ***Simulation*** entity by the ***DensityFunctionalTheory***-class, which is considered as ***Model*** and thus as an ***Icon.*** Therefore, the assertion to relate all ***Process***-subclasses mentioned to ***DensityFunctionalTheory*** via the ***hasModel*** relation is valid. Additionally, it will be stated that ***QuantumEspresso*** or ***SIESTA*** are an ***Icon*** of the ***Simulation*** via the ***hasExecutingSoftware***-relation.

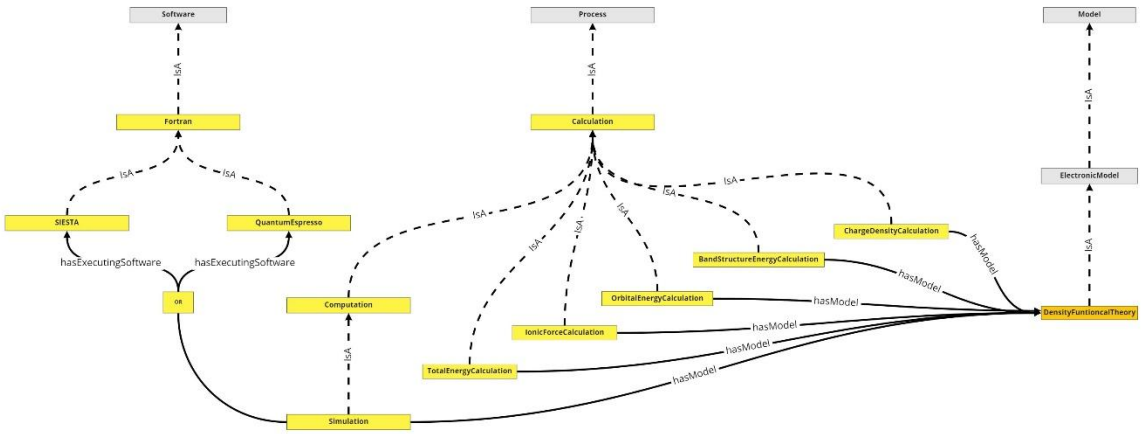

Figure 16: Coupling of a simulation entity to different calculations via the entity for DFT.

Coupling of simulation entities

In order to specify the executed simulation in more detail, we introduced entities for the ***SCFSolver*** (SelfConsistentField Solver) and AiiDA. Since AiiDA is the workflow manager which launches and controls the simulations in QE and SIESTA**,** workflow specific pieces of information, such as the ***AiiDANode***, which specifies where the task is running at, are ontologized via ***hasAiiDANode***. In the case of ***QE***, information about the ***SelfConsistentField*** as subclass of a ***Theoretization*** (a ***Process***-entity) is added to the ***Simulation***-entity over the ***SCFSolver***-entity by the ***hasExecutingSolver***-relation (see Figure 17). The ***SelfConsistentField***-entity is additionally bearing the ***Threshold*** of the ***ChargeDensity***, used to set the accuracy conditions for the convergence of the scf cycle. The link is made via the ***hasThreshold***-relation.

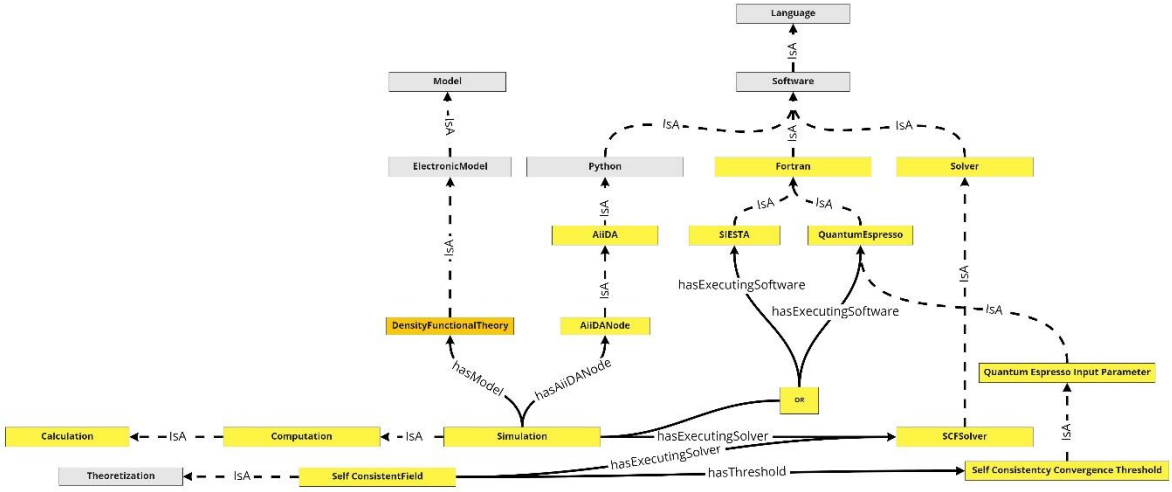

Figure 17: Coupling of simulation entities to software entities and the self-consistent-field.

Minimization

The ***Minimization*** of the ***TotalEnergy*** and of the ***IonicForces*** plays another important role in the convergence criteria of the ***SelfConsistentField*** (see Figure 18). For this purpose, both these quantities are having their own ***Threshold*** that defines the ***TotalEnergyCalculation*** and the ***IonicForceCalculation*** convergence. Minimization is also asserted to be a computation by having the ***SCFSolver*** as executing Software. The ***Groundstate*** is intended to be reached by computing the lowest ***TotalEnergy*** of the system through the ***Minimization*** step**.** The reference is managed by the ***hasSign***-relation.

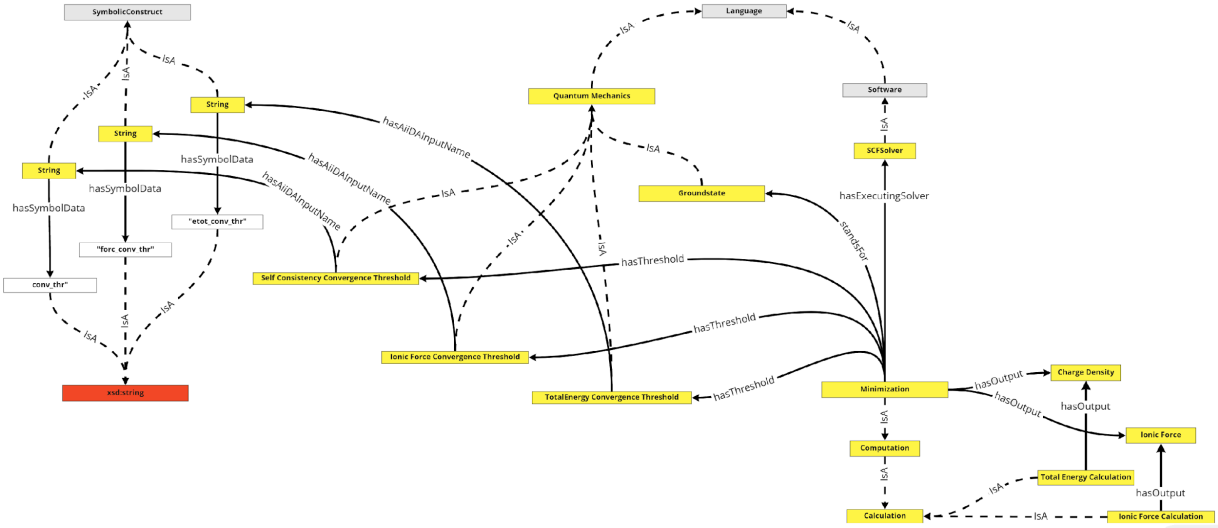

Figure 18: Minimization of the total energy and the ionic forces in relation to its considered threshold. The charge density is not a direct production of the minimization but related to a certain force- and energy field of the simulation.

Coupling of equations

In the IM2D-ontology, the ***Calculation***-entities refer to the original ***PhysicalsEquation*** used to derive the physical quantities. According to the MODA from D1.1, the ***TotalEnergy***, ***ChargeDensity,*** and the ***OribitalEnergy*** are calculated by using the ***KohnShamEquations***. The ***IonicForces*** result from a variance of the ***Hellman-Feynman-Theorem***. Since the ***Calculation*** is executed by the simulation software, the equation is considered to be a ***Sign*** of the derivation process of a quantity, so that the ***hasCalculatingEquation*** (Figure 19) is a subrelation of ***hasSign***, as described above.

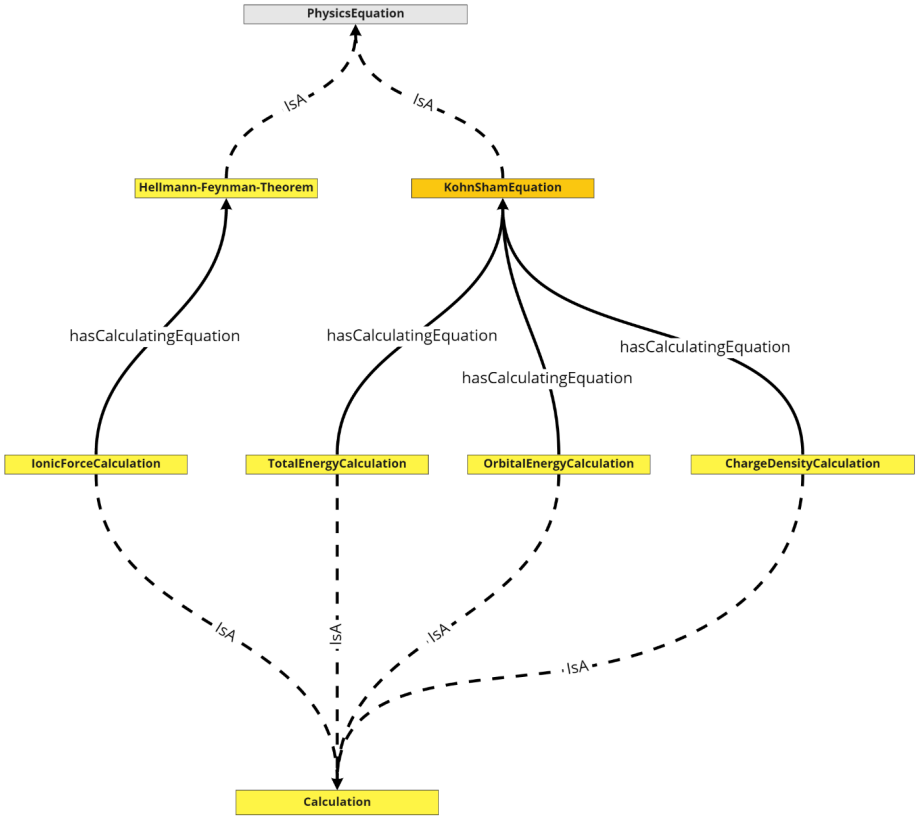

Figure 19: Equations belonging to each of the mentioned calculations.

Coupling of physical quantities

The resulting physical quantities from the ***Calculation***-entities are linked by the ***hasInput***- and ***hasOutput***-relations in order to map their interdependence. According to the equations attached in the MODA from D1.1, the ***OrbitalEnergy*** of an electron on a specific site depends on its ***WaveFunction*** and ***Hamiltonian***. The ***ChargeDensity*** at a specific site can be derived by its ***WaveFunction***. Next to the ***OrbitalEnergy***, the ***TotalEnergy*** also depends on the ***HartreeEnergy*** and on the ***ExchangeCorrelationEnergy***. It is important to mention that the ***HarteeEnergy*** and ***ExchangeCorrelationEnergy*** are both a functional of the ***ChargeDensity*** itself. The ***IonicForce*** is determined by the ***Hamiltonian*** and the ***TotalEnergy*** again. We also remark that the actual calculations with respect to their physical material relations are very complex and have been strongly oversimplified in the graph of Figure 20.

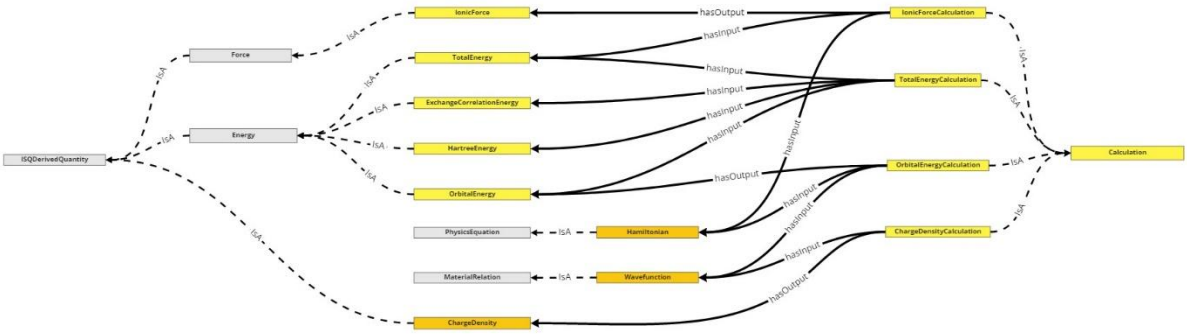

Figure 20: Quantities as a matter of in- and output for each calculation entity.

Wavefunction

The description of the wavefunction is code dependent. The entry below is specific to the planewave implementation of QE, and it cannot be extended *as is* to SIESTA that adopts a different basis set (real-space gaussian) for the representation of the wavefunctions. In crystalline systems, the ***WaveFunction*** of an electron has the form of a Bloch function, i.e., a ***PeriodicWave*** with the lattice periodicity. In the case of QE, the wavefunction is expanded on a ***PlaneWaveBasisSet*** (see Figure 21), and described by a complex exponential function with a ***WaveVector*** and a lattice vector in real space. The ***PeriodicFunction*** depends on ***RealSpaceLatticeVector*** and is labeled by a reciprocal space vector of the first ***BrillouinZone.***

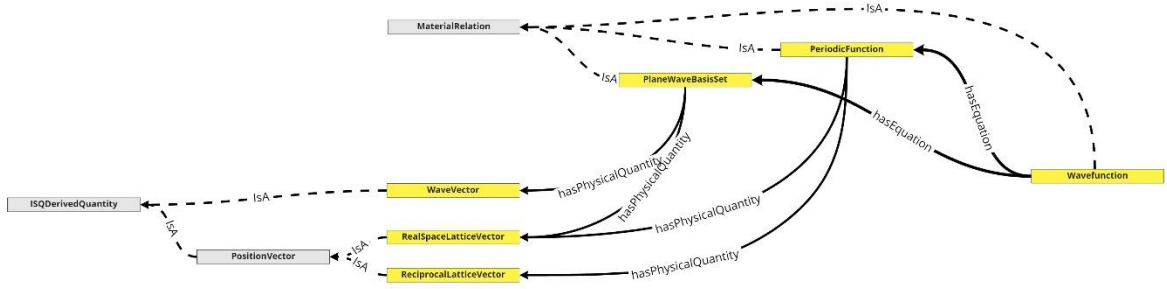

Figure 21: More detailed ontological description of connected material relations and quantities to the mathematical aspects of the Wavefunction.

Hamiltonian

The Hamiltonian in the single-particle Schrödinger equation is expressed by the ***ElectronMass***, the ***PlankConstant*** and the ***ElectronCharge*** constants, as well as by the ***EffectivePotential*** constant. In the Kohn-Sham DFT representation, this potential is expressed in terms of ***HarteePotential***, the ***ExchangeCorrelationPotential*** in terms of ***ExchangeCorrelationFunctional*** and the ***ExternalPotential*** (see Figure 22). This

ontologization of the ***Hamiltonian*** is only of minor importance for the semantic interoperability and might be more simplified in the future.

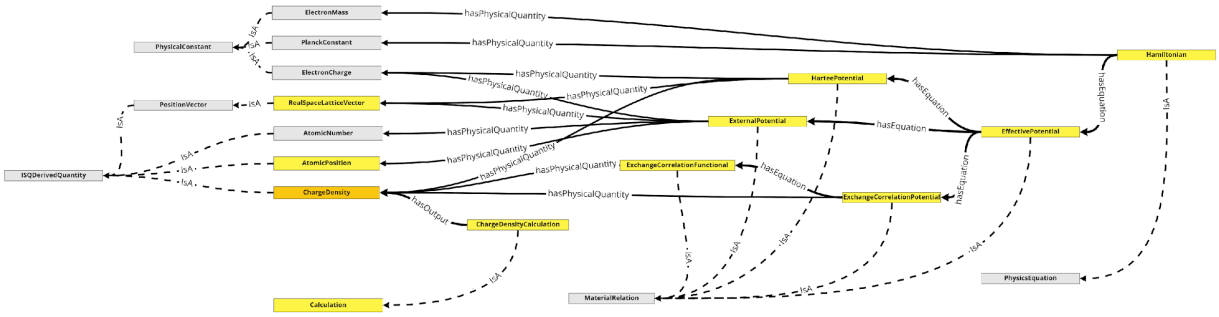

Figure 22: A more detailed ontological description of connected material relation and quantities to the mathematical aspects of the Hamiltonian.

Coupling of formula symbols

The most characteristic formula symbols for the physical quantities mentioned in the previous paragraphs are coupled to the ***Perceptual***-class of ***GreekLetter***- and ***EnglishLetter***-subentities via the ***hasFormulaSymbol***-relation, which is a sub-relation of ***hasPart*** (Figure 23).

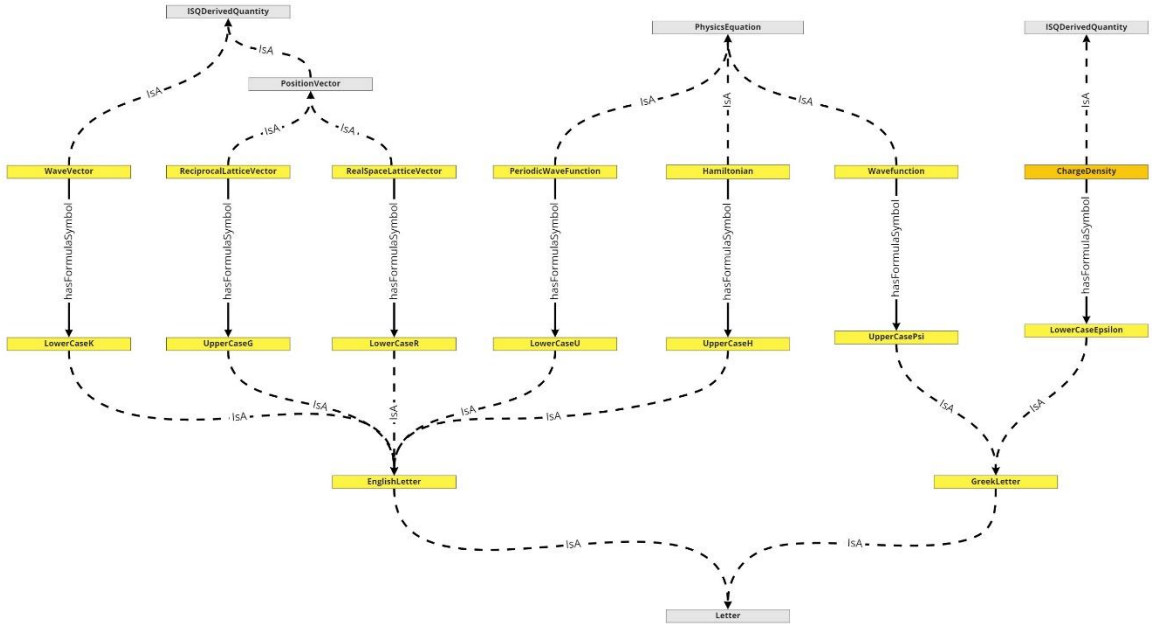

Figure 23: Coupling of formula symbols to the physical quantities used in the calculations.

Coupling to AiiDA concepts

Within INTERSECT, the formalization of the ontological link to AiiDA and its concepts (Figure 24) is particularly relevant in view of the IM2D interoperability upscale. An

AiiDAComputation is understood to have a ***Workflow*** from the AiiDA-post with a ***String*** as a name (e.g. post.BandGap). The workflow triggers a sequence of ***WorkChains*** (e.g., PWBandGapWorkChain), which run on specific ***AiiDANodes***, and are identified by a UUID, a 128 Bit number. The ***AiiDANodes*** sequence with the related ***WorkChains***, as well as theAiiDA-packages version, will be wrapped as individuals to the ontology by SimPhoNy in the semantic interoperability task.

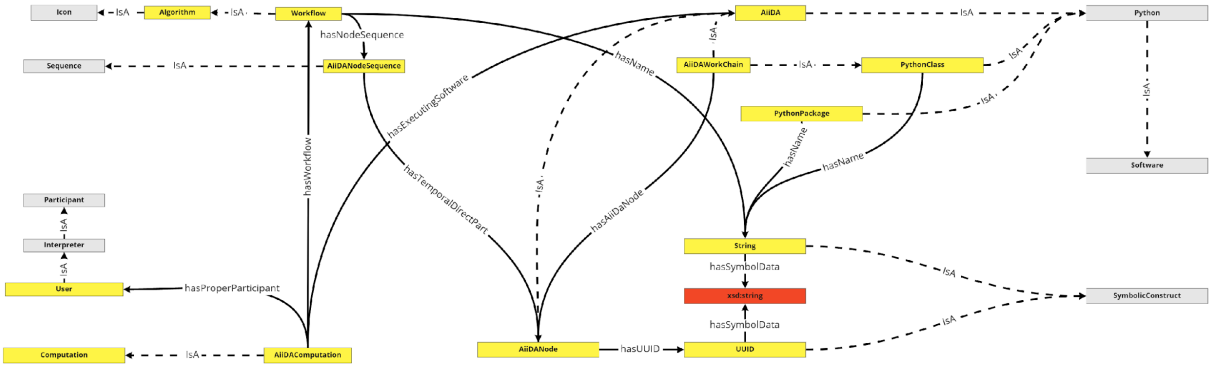

Figure 24: Ontological representation of the AiiDA-workchains which are executing the DFT-simulations.

Plane Wave Truncation (QE)

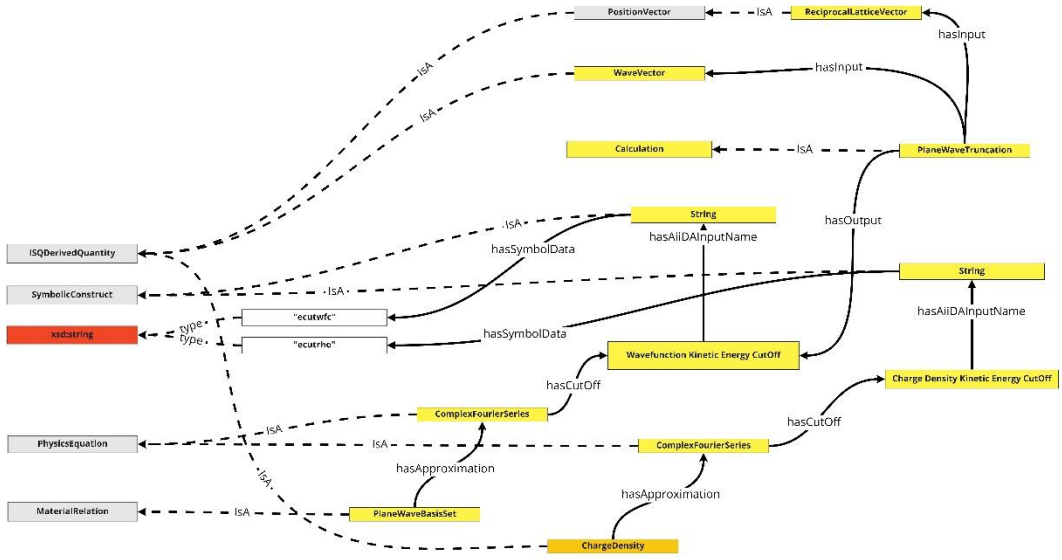

Figure 25: Ontological description for kinetic energy cutoffs.

The ***KineticEnergyCutoffs*** from Section 4.4 can be associated to a ***ComplexFourierSeries*** by the ***hasCutOff***-relation (Figure 25), and can get a name in relation to the AiiDA-workflow via ***hasAiiDAInputName*** to a ***String*** which holds the data property of ***"ecutwfc"*** and of

"ecutrho". For further interest, it can be described that the ***PlaneWaveTruncation*** is a function of the ***WaveVector*** and of the ***ReciprocalLattliceVector***. As mentioned above about the wavefunction, this description is valid for QE only and cannot be extended to SIESTA.

5. Conclusions and outlook

The MODA-tables for the DFT and device simulations have been formalized into an EMMO-1.0.0-beta-ontology, and provide a flexible manner for the maintainability of the OSP-interface in deliverable D2.5. The implementation of the ontological schema for the material-related, and crystallographic interpretation of the simulation properties is still in progress, and will be completed in the next months within the end of the project. One set of input-parameters will be chosen in order to control the accuracy of the SIESTA-workflows similarly to what has been done in the QE case. Furthermore, generic and code-agnostic classes describing the Kinetic Monte Carlo and the Drift Diffusion Model of GINESTRA™ will be implemented. The resulting adaptations and additional schemas will be added until the end of the project.

References

[1] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition,* pp. 199-220, 1993.

[2] "CEN Workshop Agreement," [Online]. Available: https://boss.cen.eu/developingdeliverables/CWA/Pages/default.aspx.

[3] "Description of YAML format Specification," Material Informatics team at Fraunhofer IWM, [Online]. Available: https://gitlab.cc-asp.fraunhofer.de/simphony/osp-core/blob/master/doc/yaml_spec.md.

[4] H. Sack, "Knowledge Engineering with Semantic Web Technologies. Lecture 3.1: Ontologies Basics," Hasso Plattner Institute, 2015.

[5] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," ksl.stanford.edu, 2001.

ACRONYMS

CIF – Crystallographic Information File

COD - Crystallography Open Database

CUDS - Common Universal Unified Data Structures

DFT - Density Functional Theory

DoA – Description of the Action

EMMO - European Materials Modelling Ontology

IM2D – Interoperable Materials-To-Device

KPI – Key Performance Indicator

M2D - Material-to-Device

MODA - Materials Modeling Metadata

NEB - Nudged Elastic Band

OWL - Web Ontology Language

QE – Quantum ESPRESSO

UUID – Universally Unique IDentifier