Ref. Ares(2020)4055942 - 31/07/2020

D1.4

Report on INTERSECT high-level requirements, interoperability interfaces for coupling and linking

Joana Francisco Morgado, Arrigo Calzolari, Nicola Marzari, 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:	30/07/2020
Deliverable No.:	D1.4
Responsible participant:	Fraunhofer (participant number 5)
Contributing Consortium members:	EPFL, CNR, FRA, AMAT
Due date of deliverable:	31/07/2020
Actual submission date:	31/07/2020
Dissemination level:	PU - Public

Authors: Joana Francisco Morgado, Arrigo Calzolari, Nicola Marzari, and Adham Hashibon

To be cited as: J. F. Morgado, A. Calzolari, N. Marzari, and A. Hashibon (2020): Report on INTERSECT high-level requirements, interoperability interfaces for coupling and linking. D1.4 of the H2020 project INTERSECT (final version as of 30/07/2020). EC grant agreement no: 814487, CNR, Modena, Italy

Versioning and Contribution History

Version	Date	Modified by	Modification reason
v.01	26/07/2020	Joana F. Morgado	First Version
v.02	30/07/2020	Arrigo Calzolari	Final Version

D1.4 Report on INTERSECT high-level requirements, interoperability interfaces for coupling and linking

Contents

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

Executive Summary

The architecture design of the Interoperable Materials-to-Device (IM2D) infrastructure, developed within the INTERSECT project, is the essence of this deliverable. IM2D comprises the interconnection between the Ginestra™ code, which is an industry ready, mature technology for device design, and emerging, upscaled target materials electronic modelling codes (Quantum ESPRESSO (QE), and SIESTA) and databases (e.g. materialscloud.org). This deliverable reports on the overall IM2D architecture and workflows design focusing on specific target end users, namely those taking the role of device developers that are not necessarily experts in a particular materials modelling approaches but are knowledgeable in device R&D, design and manufacturing. The aim is then to enable an increased utilization of advanced materials modelling in the industry day-to-day activities, by embedding the modelling expertise into the IM2D system.

For this reason, the architecture focuses on the high-level requirements and interoperability interfaces that define this type of user's interaction with the integrated environment design IM2D. This includes the necessary coupling and linking scenarios to support both the classical bottom-up and the top-down designs – *i.e.*, from materials properties to prediction of the electrical device performances and vice versa, from the performance requirements (e.g., casted in Key Performance Indicators (KPI)), to materials selection). In particular this deliverable lays down the requirements and specifications of select user stories identified in Deliverable D1.1 and extends INTERSECT ontology foundations (reported in Deliverable D1.3) with new concepts necessary to enable semantic interoperability across IM2D workflows implemented in WP2. This portfolio of activities carried out along this work resides on the core of IM2D semantic interface development.

1. Description of the work

The high-level requirements consist on specifications of the system or platform that intend to guide the developers through its major issues and functionalities. These requirements are not intended to be at such a technically elaborate level that could be directly implemented, but give the complete understanding of how IM2D features and expected behavior are; so that the developers can pick the right tools and technologies to upscale of the code. The high-level requirements are, therefore, solely dictated by the functionality from the user interaction design point of view and not the technical implementation details that are left to the coding step.

IM2D high level requirements address the main user stories of INTERSECT, which is relevant to a ***persona*** (see definition and characteristics on D1.1) that is typically an engineer expert in materials and devices optimization/characterization but with no experience in materials modelling. This *persona* is interested therefore in the ability to define the basic information on the material, device, and parameters to activate various automatic workflows that rely entirely on the combination of GINESTRA^{TM} [1] and AiiDA [2]. The latter also manages the interactions with the DFT engines (Quantum ESPRESSO [3] and SIESTA [4]). There is a need therefore to provide information passage from the front-end through the user interface with these components. The IM2D should be capable of processing these requirements into a fully-fledged user case that can be thoroughly executed and validated in an automatic manner. In order to facilitate integration with EMMC MarketPlace [5] and/or other open simulation platforms, and to enable curation of the entire user case, IM2D has to support a semantic, EMMO-based [6] description of the entire user case and results. This is however only required to connect to the outside. Internally, the various components, such as AiiDA, Ginestra^{TM}, SIESTA, QE and the data hub, will use their own syntactic language as it is already optimized for providing the needed efficiency.

This work focuses on the framework workflows design and on its alignment with the user requirements addressed in D1.1 to make sure IM2D completely resonates the INTERSECT targeted persona and material modelling workflows. Hence, part of this work deals with the specification of the framework behavior when the user interacts with it. In particular, the identification of information that the user will be able to retrieve and have access, as well as how the interaction user-interface will look like. Additionally, these specifications are complemented with the architecture/workflow that comprises the elaboration on the interconnection between the different tools involved in the workflow definition.

The identification of the high-level requirements of IM2D framework is facilitated by the results from D1.1 together with analysis of the MODA workflows [7] and discussions with the partners about the information that should be accessible to the user (*persona* of interest). The use of the identified high-level requirements on the development of the interoperability interface is also demonstrated for AiiDA and takes into account electronic modelling calculations.

1.1 Deviation from Planned work in the DoA

After the start of the project, MDlab was acquired by Applied Materials (AMAT, now official partner of the consortium), a large enterprise in the domain of electronic device design and validation. As result the business requirements from AMAT with respect to user interfaces shifted more towards reusing the existing GUI. Hence, instead of focusing currently on creating a new GUI system, the integration of the relevant part(s) of the Ginestra™ GUI (e.g. the DFT plug-in) into the existing MarketPlace and IM2D has been prioritized. This required more efforts and affected also D1.5 which is available currently only in preliminary form (see also risk R1 in Deliverable D5.2). The high-level requirements here have also been affected resulting in more efforts needed. An alternative, native and simplified GUI is still in plan, but is not the main component to be delivered.

2. Results

2.1. High-level requirements and workflow design

Based on the user story and the expected behavior of the IM2D the following high-level requirements are proposed for the IM2D:

1. IM2D has a front-end comprising a user interface that expresses the user case in a semantic, EMMO compliant manner, by using the EMMO-CUDS (Common Universal Data Structures) and the SimPhoNy OSP (Open Simulation Platform) core engine.
2. IM2D provides a user interface element enabling the user to specify all needed parameters, as described in Table 1 in D1.1.
3. IM2D has a back-end that consists of several integrated simulation codes (simulation hub) and databases (data hub).
4. IM2D provides a semantic, EMMO based description of the user case to the back-end.
5. The user can choose from two main operation modes: device-to-materials (D2M) or materials-to-device (M2D).
6. The user launches the workflow.
7. IM2D analyses the input and decides internally which device level and electronic level modelling is to be done.
8. The electronic modeling is orchestrated by an embedded AiiDA platform within the interface [8].
9. AiiDA infrastructure controls the electronic modelling engines, SIESTA and QE.
10. AiiDA infrastructure controls the data management and storage via the data hub, which is based on AiiDA back-end.
11. SimPhoNy-OSP core enriches the workflows data with EMMO-based metadata based on CUDS that provide means to store all aspects of a workflow in a format compliant with the Materials MarketPlace and other compatible repositories in addition to the AiiDa back-end.

12. SimPhoNy is the core of the interoperability layer of IM2D and analyses the specific use case from user specification and identifies what kind of data and information needs to be exchanged and selects the most appropriate workflow that will be operated by the simulation hub (sHub) through AiiDA.

Figure 1 outlines IM2D components' interconnection (workflow design) and highlights in light-blue the target component for this deliverable – semantic interface.

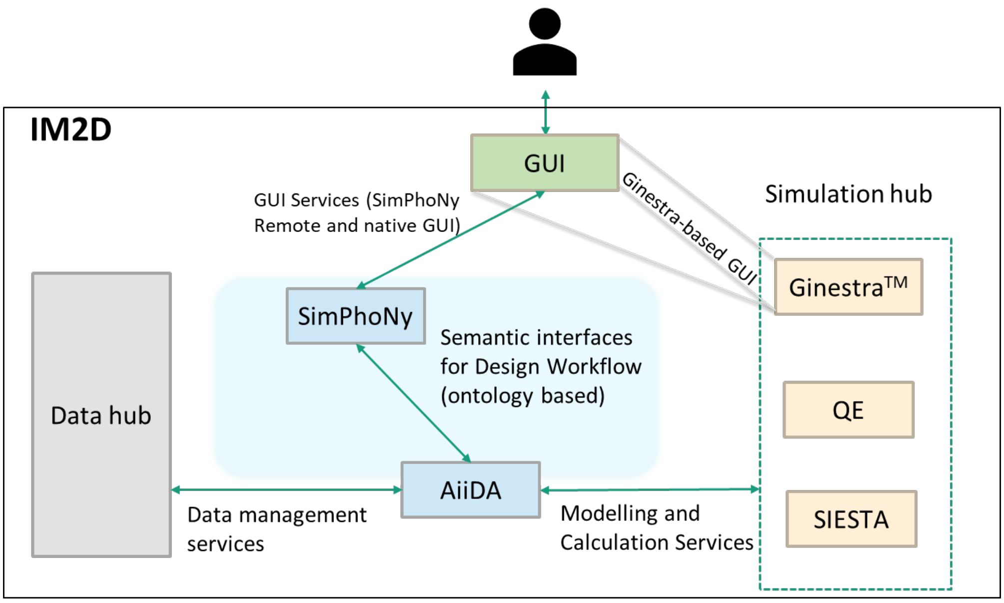

Figure 1 – IM2D components' interconnection

As represented in Figure 1, the semantic interface enables interoperability between the GUI and the modelling and data services within entire infrastructure. Basically, it maps the user needs specified at the GUI level into more specific workflows that are executed by AiiDA [8]. The obtained results are casted into semantic objects that can be seamlessly shared across the IM2D box components. Ultimately, the semantic interface will also facilitate the interoperability of IM2D with the emerging Materials Modelling Marketplaces (in particular the MartketPlace [9]).

The semantic interface builds on INTERSECT high level requirements, which describe the platform architecture and workflows design as well as the user interface (user's experience when using the framework).

2.1.1 Functional requirements

In the following, we report a brief description of the IM2D response to the user interaction focused on its functionality (based on the user story described D1.1).

- The user fires up the application via:
    - MarketPlace web platform or

    - Extended Ginestra™ GUI or
    - Materials Cloud/AiiDa Lab
- Then the user can choose from two main features:
    - M2D modelling service or
    - D2M modelling service
- Once the user selected the suitable modelling approach several workflows will be available to the user. Figure 2 represents the selection of workflows available for each of the available modelling routes.

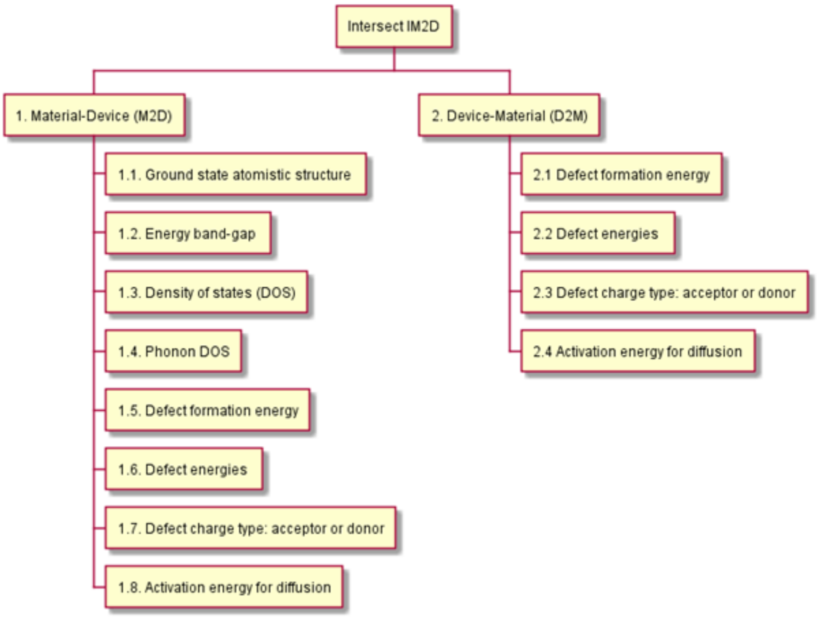

Figure 2 - IM2D main features

- User can choose the suitable workflow (represented in Figure 1 as M2D: 1.1-1.8; D2M: 2.1-2.4)
- Once the user chooses the desired workflow, she/he is presented with an interface in which:
    - User enters the input parameters then
    - User submits the simulation
- Once calculation is over:
    - User can review results and relevant plots and
    - User can choose whether to store in the local database or discard calculation and
    - User can choose to donate the results to open databases

2.2. Interoperability Interfaces – semantic layer

Interoperability interfaces can be understood as "bridges" in software development that enable not only the intercommunication between different tools, but also the exchange of meaningful information that can be understood by any tool that "speaks" the same language.

Typically, this is achieved by adopting a common semantic model (common language) that is presented in IM2D in the form of an ontology (as described in D2.3).

In IM2D, the interoperability interfaces are built on the SimPhoNy OSP-core framework that enables the generation of ontology-based data structures, and CUDS that can communicate with any SimPhoNy aware tools. SimPhoNy facilitates the conversion of opaque data (with hidden meaning/semantics) into transparent data whose semantics are accessible and understood [10] and can be seamlessly exchanged among different components (tool agnostic). In particular, the semantic interfaces are based on the SimPhoNy wrapper structure that encloses IM2D engines with a semantic layer compatible with the current standards for Materials Modelling and thereby compliant with the emerging materials modelling marketplaces (MarketPlace).

As observed in Figure 1, the semantic interface provides the medium for information exchange between the GUI services and the data and modelling services, which are developed based on Ginestra^{TM} GUI and AiiDA framework, respectively. This means that both AiiDA and Ginestra^{TM} need to be integrated with a semantic layer, and thereby integrated within the SimPhoNy environment based on the high-level specifications of the workflows (inputs, outputs, simulation code, type of calculation, etc.). To this aim, we need to extend and adapt the semantic framework (ontology) demonstrated in D2.3 with concepts (e.g. i.e. electron transport) that enable the representation of the information needed to interconnect Ginestra^{TM} GUI and AiiDA in the context of IM2D applications.

In the following section we demonstrate the exercise of converting the high-level requirements into the necessary ontology concepts that will serve as foundation for the semantic interface development. In this work we show the current ongoing efforts applied to electronic calculation workflows.

2.2.1 EMMO-based ontology extension

Based on the discussions with the INTERSECT partners with the electronic modelling as core activity (in particular EPFL, ICN2 and CNR) we concluded that only the high-level description of the inputs, outputs and simulation codes should be accessible to the user and should be sufficient to trigger the suitable workflow at the engine level (AiiDA). All remaining information should be hidden at the wrapper level and encoded based on the engine syntax.

The identified high-level requirements that should be seamlessly exchanged between the user, GUI and AiiDA engine for electronic modelling are the following:

- **Input object** contains all elements which need to be prescribed by the user in order to run the desired workflow via the IM2D framework. This object consists of:
    - **Structure**: represents the crystal structure and therefore describes the arranging of atoms of the system and its chemical composition under study (i.e., material).

- **Code**: represents the name of the code that the user wants to use to run the simulation. In the IM2D context an considering electronic models only two codes can be used QE and SIESTA.
    - **Physics based model:** according to MODA and EMMO represents a solvable set of one Physics Equation and one or more Materials Relations.
    - **Accuracy level:** represents the level or accuracy desired by the user. This level of accuracy determines the numerical method parameters specifications an results in a portfolio of possibilities that range from high accurate, time consuming and computational expensive simulations to low accuracy, fast and computational expensive simulations. Details concerning these different modes of operation are hidden at the interoperability and syntactic layers of the SimPhoNy-based wrappers. As example, in SIESTA the accuracy is defined via Protocols [11].
- **Output object** that consists of:
    - **Structure**: optimized crystal structure obtained from the simulations that should follow the same schema as in inputs
    - **Energy**: represents the energies obtained from the simulation. This entity can be specialized in several relevant types of energies among which are: total energy, adsorption energy

To comply with the functional requirements, more precisely, with the ability to choose the desired workflow, the type of workflow is also defined at the semantic level. This means that classes such as "*Ground state atomistic structure*" or "*Activation energy for diffusion*" need to be represented via ontological concepts. An initial proposal for the ontological representation of the IM2D features is represented in Figure 3.

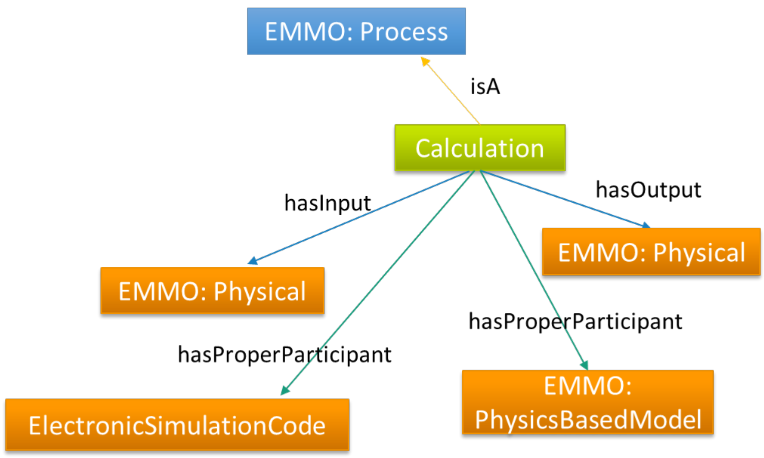

Figure 3 – The "core" structure for IM2D feature (calculation) representation via an ontology

Each electronic modelling feature is represented as a specialization of the *owl:Class* **Calculation**, which is represented as subclass of the *owl:Class* **Process** in EMMO and contains several proper participants. A more detailed description of the specified classes and relationships is given below.

- ***owl: Class*** **Process**: this is a EMMO concept that “restricts the meaning of the word process to 'Physical'-s whose evolution in time have a particular meaning for the ontologist (i.e. every 4D object unfolds in time, but not every 4D object may be of interest for the ontologist).” [6]
- ***owl: Class*** **Calculation**: is defined as type of activity/computation (a process in itself) that participates in a process and involves the transformation of inputs into results (outputs).
- ***owl: Class*** **ElectronicSimualtionCode**: represents the software used to execute the calculation (based on electronic models). The electronic simulation code should contain as proper parts the solver (numerical method), the type of physics-based model (electronic model), the solver parameters (such as accuracy) and the version of the software.
- ***owl: Class*** **PhysicsBasedModel**: this is a EMMO concept based on the MODA standard that represents a solvable set of one Physics Equation and one or more Materials Relations.
- ***owl:ObjectProperty*** **hasInput**: is a specialization of the EMMO *owl:ObjectProperty* hasProperParticipant which represents the physicals that are inputs to the calculation. An input only exists when it is linked to the process, therefore its representation by linking it via the inverse (*hasInput*) relationship to a process is proposed.
- ***owl:ObjectProperty*** **hasOutput**: is a specialization of the EMMO owl:ObjectProperty hasProperParticipant and supports the representation of the outputs of a process (calculation). This relation together with *owl:ObjectProperty* hasInput enables to distinguish between the inputs and outputs.
- ***owl:ObjectProperty*** **hasProperParticipant**: is a EMMO relationship that is a specialization of owl:ObjectProperty hasProperParticipant, which is a way to categorize temporal regions of a process by the interpreters. As defined in EMMO it is the relation between a process and an object participating to it [6].

Figure 4 represents the Ground state atomistic structure feature of IM2D based on the identified high-level requirements.

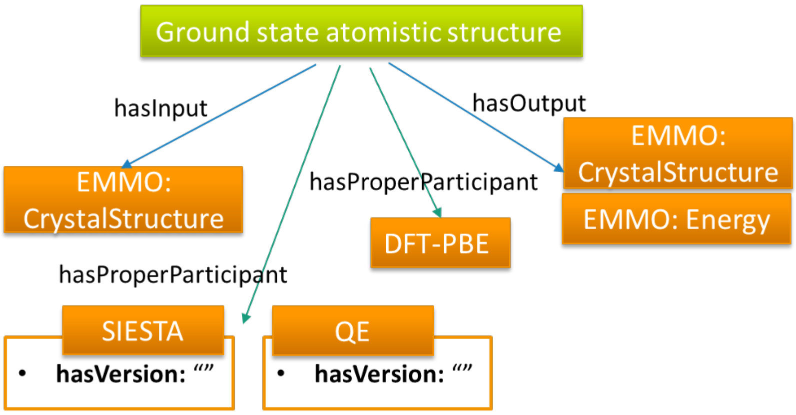

Figure 4 – Representation of the Ground state atomistic structure feature of IM2D based on the calculation generic representation described in Figure 3.

2.2.2 CUDS generation based on the high-level requirements

As already stated in the previous sections, CUDS structures can be shared and understood by any SimPhoNy aware tool enabling the semantic interoperability across an integrated environment. Figure 5 represents a CUDS structure generation for the Ground state atomistic structure feature of IM2D (https://gitlab.cc-asp.fraunhofer.de/ontology/applications/intersect/im2d-electronic-calculations/-/blob/master/MISC/intersect_highlevelreq.py), which builds on the ontology provided at https://gitlab.cc-asp.fraunhofer.de/ontology/applications/intersect/im2d-electronic-calculations/-/blob/master/MISC/intersect.im2d.yml.

intersect_highlevelreq.py 1.27 KB

Edit | Web IDE | Replace | Delete

```
from osp.core import intersect
from osp.core.utils import pretty_print,Cuds2dot


# Create a calculation entity
calculation = intersect.GroundStateAtomisticStructure()

# Create a site

site1 = intersect.Site()
site1.add(intersect.Species(name = "Ti"), intersect.Position(vector=(0,0,0), unit=""), intersect.Occupancy(value=1, unit=""), rel= intersect.hasSpatialDirectPart)
site2 = intersect.Site()
site2.add(intersect.Species(name = "O2"), intersect.Position(vector=(0,0,2), unit=""), intersect.Occupancy(value=0.95, unit=""),rel= intersect.hasSpatialDirectPart)

# Create a unit cell
unitcell = intersect.UnitCell()
unitcell.add(intersect.LatticeVector(vector=(4.0,0.0,0.0), unit=""),intersect.LatticeVector(vector=(0.0,4.0,0.0), unit=""),intersect.LatticeVector(vector=(0.0,1.0,4.0


# Create inputs - Crystal structure

structure_in = intersect.CrystalStructure()
structure_in.add(unitcell,site1,site2, rel= intersect.hasSpatialDirectPart)

# create a code
code = intersect.Siesta(has_version = "4.1-b4")

# create a PhysicsBasedModel

model = intersect.DFT_PBE()

calculation.add(structure_in, rel = intersect.hasInput)
calculation.add(code, model, rel = intersect.hasProperParticipant)

pretty_print(calculation)
Cuds2dot(calculation).render()
```

Figure 5 – CUDS structures generation based on the high-level requirements ontology

A pseudocode for the semantic interface of a SimPhoNy-AiiDA wrapper is represented in Figure 6.

```
from osp.wrappers.simaiida.simaiida_session import SimaiidaSession
import time

# CUDS generation

############### QUANTUM ESPRESSO #########################

print("Starting QE Simulation with input: \n")
wrapper = intersect.SimaiidaWrapper(session=SimaiidaSession())
code = intersect.QuantumEspresso()
calculation = GroundStateAtomisticStructure()
calculation.add(structure_in, rel = intersect.hasInput)
calculation.add(code, model, rel = intersect.hasProperParticipant)
wrapper.add(calculation)
wrapper.session.run()


# check if there is a output and print it
calculation_out = wrapper.get(oclass=intersect.GroundStateAtomisticStructure)
if calculation_out.get(rel=intersect.hasOutput):
    pretty_print(calculation_out.get(rel=intersect.hasOutput)[0])
```

Figure 6 – Pseudocode for the semantic layer for a SimPhoNy-AiiDA wrapper

As described in Figure 6, we need to import the *SimaiidaSession* from the SimPhoNy-wrapper package that can be initialized via the wrapper method run() and to which we can add the necessary CUDS objects via similar methods as in the CUDS API (add()). The initialized session can then be accessed via the wrapper object, from which we can retrieve the generated results in the form of CUDS structures.

All the software-related descriptions are kept at syntactic level and can be specified via the AiiDA common workflow API available at [8] that can be used to develop the interoperability layer of the wrapper. An example of this API is represented in Figure 7.

Taking the example of the structure relaxation, a line of the kind:

```
generator = SiestaRelaxationInputsGenerator()
builder = generator.get_builder(structure, protocol, relaxation_type)
```

returns a builder of the `SiestaRelaxWorkChain` with inputs following a specific `protocol` and `relaxation_type`. A similar code, but using a `QuantumEspressoRelaxationInputsGenerator()` would return inputs for the `QuantumEspressoRelaxWorkChain`. More details on the specific case are in the folder RelaxWorkChain.

Figure 7 - AiiDA-common-workflow API (source: [8])

2.2.3 EMMO vs. OPTiMaDe schema

One of the ongoing topics in the semantic web community is the interoperability among the various available ontologies to facilitate the cross-domain information exchange and reusability. Only by building a sharable "common language" (semantic framework) we can realize the full-potential of interoperability.

Typically, ontologies are developed from scratch for a very specific application and do not rely on any existing nor available knowledge on the web. This implicitly means that the developed ontologies cannot be reused for different applications within the same domain which hinders interoperability. *The reuse of existing ontologies is one example of a best practice that fosters the development of a "common language"*.

To facilitate the integration of IM2D semantic interface with the available data schemas for electronic modelling and materials databases we started comparing the available entities in EMMO used to describe the crystal structure with the highly adopted OPTiMaDe schema [12]. OPTiMaDe (Open Databases Integration for Materials Design) is an international initiative that aims to make materials databases interoperational by developing a common REST API. The mapping between the EMMO branch domain-crystallography [13] and the OPTiMaDe schema [12] is shown in Figure 8.

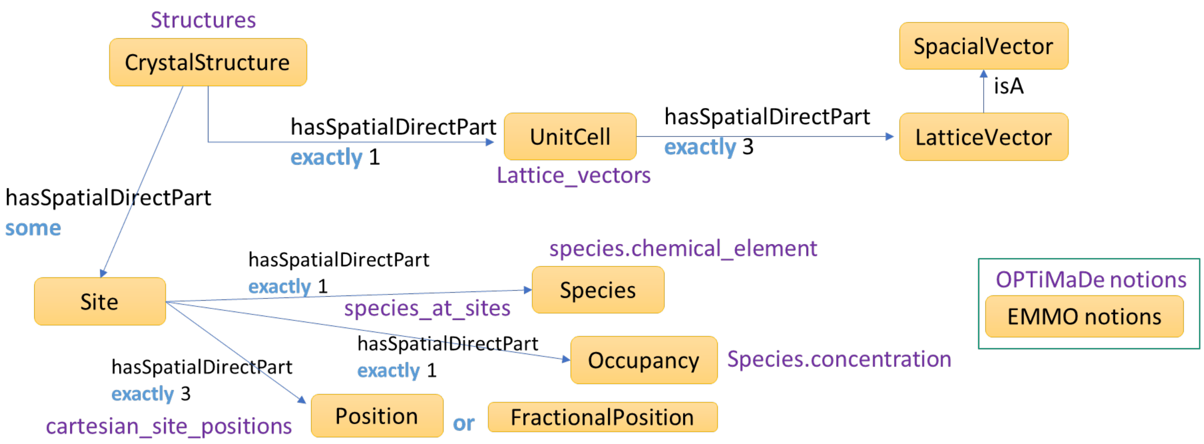

Figure 8 – Comparison between OPTiMaDe and EMMO terminology.

A more detailed description of some of the most relevant classes within INTERSECT context to represent the crystal suture based on the two standards is given in Table 1.

Table 1 - EMMO vs. OPTiMaDe

EMMO		OPTiMaDe	
Class	**Description**	**Class**	**Description**

CrystalStructure	Is a symbolic (subclass of the EMMO class language) that describes of the ordered arrangement of species in a crystal.
	
	This concept has several spatial direct parts that are necessary to define it.
		Structures	Is an object that is used to represent crystal structure with properties defined in "Properties Used by Multiple Entry Types" section at [12].
			
			{
			"data": [
			{
			"type": "structures",
			"id": "example.db:structs:0001",
			"attributes": {
			"chemical_formula_descriptive": "Es2 O3",
			"url": "http://example.db/structs/0001",
			"immutable_id": "http://example.db/structs/0001@123",
			"last_modified": "2007-04-05T14:30:20Z"
			}
			},
			{
			"type": "structures",
			"id": "example.db:structs:1234",
			"attributes": {
			"chemical_formula_descriptive": "Es2",
			"url": "http://example.db/structs/1234",
			"immutable_id": "http://example.db/structs/1234@123",
			"last_modified": "2007-04-07T12:02:20Z"
			}
			}
			// ...
			]
			// ...
			}

UnitCell	A unit cell can be described by either specifying the three lattice vectors that it is spanned by, or by 6 the	Lattice_vectors	Defined as the three lattice vectors in Cartesian coordinates, in ångström (Å) [12].

	lattice parameters (the length of and angles between the 3 lattice vectors) [13].		
Site hasSpatialDirectPart Position	This combination of EMMO classes and relationships leads to the concept of position of the site which is equivalent to each of the 3D vectors in OPTiMaDe Cartesian_site_positions object.	Cartesian_site_positions	Cartesian positions of each site in the structure. A site is usually used to describe positions of atoms; what atoms can be encountered at a given site is conveyed by the species_at_sites property, and the species themselves are described in the species property [12].
Species	A chemical species is a well-defined set of one or more atoms/ions. The chemical species can be atom, molecule, ion, radical, and it has a chemical name and chemical formula. The chemical species are associated to the crystalline site in a structure [13].	Species	A list describing the species of the sites of a stricture. Species can represent pure chemical elements, virtual-crystal atoms representing a statistical occupation of a given site by multiple chemical elements, and/or a location to which there are attached atoms, i.e., atoms whose precise locations are unknown beyond that they are attached to that position [12]. It is a dictionary with several keys like name and chemical symbol.
Site hasSpatialDirectPart Species	This combination of EMMO classes and relationships leads to the equivalent the concept of Species at_Sites in OPTiMaDe.	Species_at_sites	Consists of a list of names of the species at each site (where values for sites are specified with the same order of the property cartesian_site_positions) [12].

3. Conclusions

In this deliverable we reported on the definition and the development of the most relevant high-level interoperability requirements, that are the basis of the architecture and workflows design of the overall IM2D infrastructure. This design activity (Task 1.3) is part of WP1, while the corresponding implementation is the part of WP2.

As explained in Sec 1 deviation from DoA, the change of legal entity of a member of the consortium (MDLab acquired by AMAT) and the redefinition of internal business strategies implied an initial re-organization of the GUI design (see D1.5 and D5.2). Since this affects several technical aspects of the overall infrastructure, including the interoperability requirements and workflow design, we faced an initial delay that we are recovering. We are thus confident that results will be obtained as scheduled.

ACRONYMS

API - Application Programming Interface

CUDS - Common Universal Data Structures

DoA – Description of the Action (Grant Agreement, Annex 1)

DFT – Density Functional Theory

D2M - Device-To-Materials

EMMC - European Materials Modelling Council

EMMO - European Materials Modelling Ontology

GUI - Graphical User Interface

KPI - Key Performance Indicator

IM2D - Interoperable Materials-to-Device

M2D - Material-To-Device

QE - Quantum ESPRESSO

References

[1] Larcher, L., et al. A simulation framework for modeling charge transport and degradation in high-k stacks, J. Comput. Electron.12, 658–665 (2013).
Ginestra™ URL: http://www.appliedmaterials.com/semiconductor/products/materials-to-device-simulation

[2] Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N. & Kozinsky, B. AiiDA: automated interactive infrastructure and database for computational science. Comp. Mat. Sci. 111, 218 – 230 (2016).
AiiDA URL: http://www.aiida.net

[3] Giannozzi, P. et al. Quantum ESPRESSO: a modular and open-source software project for quantum simulations of materials. J. Phys. Cond. Matt. 21, 395502 (2009).
Quantum ESPRESSO URL: www.quantum-espresso.org

[4] Soler, J. M. et al. The SIESTA method for ab initio order-N materials simulation. J. Phys. Cond. Matt. 14, 2745 (2002).
SIESTA URL: http://departments.icmab.es/leem/siesta

[5] Materials Modelling Marketplace https://emmc.info/emmc-marketplace

[6] "The European Materials Modelling Ontology (EMMO)," [Online]. Available: https://github.com/emmo-repo/EMMO . [Accessed July 2020].

[7] European Materials Modelling Council (EMMC) MODA https://emmc.info/moda

[8] "aiida-common-workflows," [Online]. Available: https://github.com/aiidateam/ aiida-common-workflows . [Accessed July 2020].

[9] Project MARKETPLACE - Materials Modelling Marketplace for Increased Industrial Innovation www.the-marketplace-project.eu

[10]"SimPhoNy documentation," Materials Informatics team at Fraunhofer IWM, [Online]. Available: https://simphony-docs.readthedocs.io/en/latest/about.html . [Accessed July 2020].

[11] V. M. Garcia-Suarez, A. Garcia, E. Bosoni and V. Dikan, "AiiDA Siesta Plugin Documentation Release 0.11," 2020.

[12] "OPTIMADE API specification v1.0.0," [Online]. Available: https://github.com/ Materials-Consortia/OPTIMADE/blob/develop/optimade.rst . [Accessed July 2020].

[13] "EMMO domain-crystallography," [Online]. Available: https://github.com/emmo-repo/ domain-crystallography/tree/initial-public-version . [Accessed July 2020].