Semantic Interoperability and Characterization of Data Provenance in Computational Molecular Engineering

Semantic interoperability and characterization of data provenance in computational molecular engineering

Martin Thomas Horsch,*,† Christoph Niethammer,*,‡ Gianluca Boccardo,¶ Paola Carbone,§ Silvia Chiacchiera,† Mara Chiricotto,§ Joshua D. Elliott,§ Vladimir Lobaskin,‖ Philipp Neumann,⊥ Peter Schiffels,# Michael A. Seaton,† Ilian T. Todorov,† Jadran Vrabec,@ and Welchy Leite Cavalcanti#

†*STFC Daresbury Laboratory, UKRI, Keckwick Ln, Daresbury, Cheshire WA4 4AD, UK*

‡*High Performance Computing Center Stuttgart, Nobelstr. 19, 70569 Stuttgart, Germany*

¶*Department of Applied Science and Technology, Institute of Chemical Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*

§*School of Chemical Engineering and Analytical Science, University of Manchester, Oxford Rd, Manchester M13 9PL, UK*

‖*School of Physics, University College Dublin, Dublin 4, Ireland*

⊥*High Performance Computing, Helmut-Schmidt-Universität, Holstenhofweg 85, 22043 Hamburg, Germany*

#*Fraunhofer Institute for Manufacturing Technology and Advanced Materials, Wiener Str. 12, 28359 Bremen, Germany*

@*Thermodynamics and Process Engineering, Technische Universität Berlin, Ernst-Reuter-Platz 1, 10587 Berlin, Germany*

E-mail: martin.horsch@stfc.ac.uk; niethammer@hlrs.de

Abstract

By introducing a common representational system for metadata that describe the employed simulation workflows, diverse sources of data and platforms in computational molecular engineering, such as workflow management systems, can become interoperable at the semantic level. To achieve semantic interoperability, the present work introduces two ontologies that provide a formal specification of the entities occurring in a simulation workflow and the relations between them: The software ontology VISO is developed to represent software packages and their features, and OSMO, an ontology for simulation, modelling, and optimization, is introduced on the basis of MODA, a previously developed semi-intuitive graph notation for workflows in materials modelling. As a proof of concept, OSMO is employed to describe a use case of the TaLPas workflow management system, a scheduler and workflow optimizer for particle-based simulations.

1 Introduction

Hans Hasse, to whose achievements this special issue is dedicated, is among those who have contributed to the success of modelling and simulation by computational molecular engineering. Building on previous efforts in molecular model characterization and simulation method development, e.g., by Möller and Fischer[1] as well as Lotfi et al.,[2] Vrabec and Hasse introduced the grand equilibrium simulation method by which vapour-liquid equilibria can be efficiently sampled.[3] This workflow, implemented in the *ms2* code,[4–6] was the basis for a period of increased productivity in molecular model design during which Hasse, Vrabec, and collaborators parameterized a multitude of reliable intermolecular pair potentials[7–12] and applied them to predict the thermodynamic properties of pure components and mixtures.[8,13–16] Using their code *ls1 mardyn*,[17] a molecular dynamics (MD) system size world record with four trillion particles was achieved,[18] which has recently been pushed towards twenty trillion particles.[19] This work in model and software development, in combination with the increase

in accessible computational resources, played a role in establishing molecular modelling as a branch of simulation-based engineering.

Improving the interoperability of methods and codes, providing simulation metadata in an agreed way, and specifying simulation workflows that integrate multiple model granularity levels have become key challenges at combining computational molecular engineering with the other simulation-based engineering approaches that are already widespread in industrial practice, e.g., computational fluid dynamics (CFD) and process simulation. This requires a coordinated effort in data technology. With this perspective, Burger, von Harbou, and Hasse, jointly with industrial partners, worked towards interfacing experimental and simulation data with model design,[20,21] an objective that the ongoing virtual marketplace initiatives promise to pursue systematically.[22,23] Within a collaborative research centre (CRC 926 MICOS), Hasse and collaborators introduced the concept of domain-specific processing-morphology-property relationships for component surfaces, referred to as OMEB from the German expression.[24] This facilitates an approach that can be employed to connect molecular and phenomenological modelling to decision support by multicriteria optimization,[25–28] translating problems of industrial end users to solutions based on quantitatively reliable modelling and simulation.[16,29,30] Recent works by Hasse and Lenhard address the philosophy of modelling, formulating an engineering-oriented perspective on the role of computational methods.[31,32] These contributions have advanced data technology in materials modelling and created opportunities to address further challenges, some of which are discussed in the present work.

Where databases and platforms using different data structures and file formats interoperate, or where data and metadata from various sources are combined, agreement on semantics becomes a necessity,[33] supporting the effort to construct a universal web into which any linked data can be integrated[34] to become FAIR, i.e., findable, accessible, interoperable, and reusable.[35,36] For this purpose, metadata (i.e., data about data) need to be provided to characterize the context of any relevant data items so that they can be found and accessed

easily and reused properly. This includes information on data provenance, i.e., the process by which the data have been obtained.[33,37] Interoperability applies to three major aspects of data stewardship:[36] Syntax (formats), semantics (meaning), and pragmatics (procedures). It is achieved by establishing a common intermediate standard to which all users and contributors to a data infrastructure can map their own internal approach.[38,39]Accordingly, data technology solutions that aim at facilitating interoperability and data integration require the definition of semantic assets, i.e., documents that codify semantics.[40] For this purpose, it is crucial to develop and maintain community-governed semantic standards, facilitating the systematic annotation of pre-existing dark data, i.e., data for which machine-processable metadata are absent or insufficient,[41] by a variety of data and metadata owners and infrastructure providers. In data technology, an ontology is a formal machine-processable representation of knowledge within a certain domain. It consists of a definition of classes, individuals (objects that belong to the classes), and rules for the possible relations between them:[42–45] For instance, concerning simulation workflows in materials modelling, a "workflow node" can be defined as a "workflow graph that contains exactly one workflow resource." Therein, *workflow node*, *workflow graph*, and *workflow resource* are classes, and *contains* is a relation. In this way, an ontology can be employed to standardize the semantic space belonging to a particular application domain.

A variety of applications in simulation based engineering can benefit from a machine-readable way of specifying a simulation workflow;[46] thereby, the characterization of workflows is relevant in two major ways. First, workflows are designed and communicated within simulation environments where materials models are evaluated to generate data by simulation.[47] Second, in order to integrate data obtained in different ways (e.g., from simulation and experiment, or from simulations with different models or solvers), simulation results need to be stored together with metadata that describe their provenance. If experimental and other data are meant to be integrated with simulation results in a common infrastructure,[20,21] workflow descriptions can be combined with domain-specific provenance description ontolo-

gies which, e.g., already exist in genetics[48] and nanosafety.[49] Specified workflow metadata, supplemented by an extensive technical documentation, can be employed to reproduce data by repeating the same workflow. Furthermore, certain aspects of the data uncertainty (and uncertainty propagation), such as the sensitivity with respect to specific model parameters or the choice of a particular solver implementation, can be quantified by varying individual values, parameters, or elements of a workflow;[50] e.g., round-robin studies can be conducted, where various simulation software environments are employed to carry out the same (or closely related) algorithms in combination with the same models, comparing the outcome.[51] Other technologies that can profit from well-defined workflow semantics include high performance computing (HPC) and scheduling environments where computational requirements may be automatically predicted[52] and optimized by workflow autotuning and task-based parallelization.[53]

In computational molecular engineering, two major organized efforts toward achieving an agreed coherent semantic-technology framework have been conducted: With a focus on process simulations, CAPE-OPEN was developed in *computer aided process engineering* as an *open* interface standard.[54,55] CAPE-OPEN is in use both in academic and industrial engineering practice.[56,57] At present, ongoing work within a series of projects associated with the European Materials Modelling Council (EMMC) aims at going beyond this by achieving interoperability for all physical modelling and simulation methods, including quantum mechanics, molecular modelling and simulation, and continuum methods up to the macroscopic and process level.[45,58] Within this line of work, a Review of Materials Modelling (RoMM) was conducted. This review, which is now available in its 6th edition,[58] resulted from work done within the European Commission Directorate for Research and Innovation. Its long term goal is to increase the competitiveness of European industries thanks to a stronger uptake of materials modelling techniques for the different stages of manufacturing. Given the vast diversity of approaches and vocabularies used in subdomains of the modelling world, RoMM proposes a harmonized language and a classification of models to support communication

across subdomains and across roles (software developers, theoreticians, and industrialists). A detailed explanation and discussion of this harmonized language, containing numerous examples, is given in the RoMM document.[58] On this basis, MODA (Model Data), a semi-intuitive graph language for simulation workflows,[59] was introduced jointly with a collection of further semantic assets,[60] including the European Materials and Modelling Ontology (EMMO) which is under development by Ghedini et al.[61] Compared to generic workflow notations, MODA is tailored to optimally address aspects that are specifically relevant to materials modelling, and it is based on the RoMM terminology that was developed for the same purpose. Annex II of the RoMM document[58] includes MODA workflow description examples contributed by 36 projects from the LEIT-NMBP line of the European Union's Horizon 2020 research and innovation programme.

The Virtual Materials Marketplace (VIMMP) is a platform, presently under development, where services related to materials modelling such as expertise, translation (from an industrial problem to a modelling solution), software, model parameters, training, computing resources, validation data, etc., will be provided to end users. Accordingly, agents on the market include individuals, groups, and institutions from the industrial and academic world, such as modellers, software owners, and data providers. By design, VIMMP is open to any interested provider, and the basis for connecting their heterogeneous resources is a common language – hence the significance of a semantic approach. In this context, the present work discusses the state of the art in semantic asset development for simulation workflows in computational molecular engineering and introduces a formalism based on ontologies which can be employed to represent workflow metadata. Thereby, it addresses the need for a formalized, machine-readable representation of simulation workflows. This is done in a way that facilitates an integration with the previous and ongoing work done within the EMMC community, in particular with MODA, which is the previous EMMC standard for describing a simulation workflow. To increase the expressive capacity and eliminate ambiguities inherent in the semi-intuitive graph notation from MODA, logical resources are introduced as entities

related to the flow of information. On the basis of an improved graph notation for simulation workflows (cf. Section 4.1), an ontology for simulation, modelling, and optimization (OSMO) is formulated (cf. Section 4.2) which goes beyond MODA by being machine processable, amenable to automated reasoning by semantic technology, and by which workflow semantics in materials modelling are captured in a way that is closely aligned and interoperable with the whole family of semantic assets presently under development in the context of the same infrastructures and projects.

To characterize software, in general, it is possible to describe many different aspects for a variety of purposes including, e.g., to identify, to understand, to trade, or to use a given tool, and these descriptions can be provided at multiple levels of detail. Finding appropriate ways to cite software, recognize authorship, and give scientific credit to the developers is a concern for different communities and key to making software development sustainable. Along these lines, principles for software citation have been proposed,[62,63] and the metadata schema CodeMeta[64] as well as the citation file format CFF[65] have been developed. To describe simulation software within VIMMP, the VIMMP Software Ontology (VISO) is presented here (cf. Section 2.2), complementing OSMO. The main focus of VISO is to facilitate the description of software capabilities in computational molecular engineering. Ontologies with a similar purpose, which describe the software from the point of view of a scientist end user, have been developed in other fields, e.g., the Software Ontology (SWO) in the area of life sciences[66] and OntoSoft for geosciences;[67] in particular, among other aspects, SWO also covers the implemented algorithms. Here, with VISO, a comparable ontology is made available for the area of materials modelling. OSMO and VISO will be used by the VIMMP marketplace, its components, and all interoperable platforms and environments, to represent simulation workflows at a logical (i.e., non-technical) level and assist in the selection of suitable software components and simulation platforms.[22] In particular, this work aims at facilitating the description of information from model databases and parameterization environments, such as Bottled SAFT[68,69] or the infrastructures designed by Hasse and col-

laborators,[12,28] as well as workflow management systems (WMS) and workflow repositories, e.g., TaLPas[52] and exabyte,[70] in a well-defined way to make such platforms interoperable with VIMMP. Accordingly, the present work was conducted as a collaboration of the TaLPas and VIMMP project consortia.

The remainder of this article is structured as follows: Section 2 discusses the challenge of achieving interoperability of diverse tools and environments at the levels of syntax, semantics, and pragmatics; VISO is introduced as a formalism for simulation software descriptions. In Section 3, workflow management systems are discussed, and the environment developed within the TaLPas project is presented; an example workflow is introduced, concerning the parameterization of an equation of state (EOS) by molecular simulation. This application scenario is subsequently employed to illustrate the concepts from the present work. Section 4 comments on existing formalisms by which simulation workflows can be represented at a logical level, in particular the graph notation from MODA. It is shown how an improved graph notation can be employed to denote the flow of information and dependencies between components of a workflow less ambiguously, and the ontology OSMO is introduced, which provides an additional layer of formalization to the characterization of the involved classes of objects and the relations between them. Finally, a conclusion is given in Section 5.

2 Semantic interoperability

2.1 Development of semantic assets

Interoperability is the capacity of multiple codes or platforms, which are not immediately compatible, to interact automatically by means of a common representational system; i.e., whereas for compatible environments, the sender needs to be familiar with the concepts and data structures employed by the recipient, interoperability does not require any bespoke tailoring to a specific target environment. For a large number of (actual or potential) diverse interacting systems, interoperability is the more scalable approach, since it does not force

the developers of each software or infrastructure to implement all the formats required by a multitude of different codes. Instead, data transferred between interoperable environments need to be transformed to a single agreed intermediate stage by the sender, and it is the duty of the recipient to implement the common representational system adequately on his own side.

To facilitate interoperability, a common framework needs to be established at three levels: Syntax, semantics, and pragmatics.[71,72] Thereby, syntactic interoperability refers to the standardization of data formats and technical protocols for data transmission. However, beside the need for a sender and the recipient to implement input/output functionalities for the same format, they also need to agree on the meaning of the communicated contents; this is semantic interoperability. Only on this basis, full interoperability can be achieved, which additionally requires an agreement on pragmatics, i.e., the use of data,[73] including minimum standards for data and metadata curation, research data management, validation, and assessment of data. Pragmatic interoperability also concerns what to expect from an individual agent with a particular social role,[74] e.g., a *translator* who maps a problem from industrial practice to viable solutions by computational molecular engineering; significant efforts need to be devoted to negotiating agreement on such expectations. Along these lines, in case of the translator role, the EMMC has developed a pragmatic asset, the Translators' Guide.[75]

Syntactic and semantic interoperability are closely related and usually co-developed. If the focus is on file formats (hence, syntactic interoperability leads the development), underlying assumptions on the interpretation of the contents often remain implicit; guidance on semantics is usually, if at all, provided in human-readable form, e.g., in a user manual. Obversely, if semantic interoperability leads the development, standard serializations of data exist by which syntactic agreement can be achieved in a straightforward way, such as the RDF/XML format, the terse triple language (TTL), the hierarchical data format HDF5, or the Allotrope data format.[43,76,77] The semantic assets usually take the form of metadata

schemas or ontologies, stating what classes of objects exist (in a certain domain, i.e., the application field for which the schema or ontology is designed) and how they can relate to each other.[42–45] The approach based on semantic interoperability has the advantage that the agreement on both the format and the meaning is codified on the basis of definitions that can be processed computationally, e.g., by automated logical reasoning. In this way, the internal consistency of data sets can be checked, and data from multiple sources can be integrated,[40] facilitating more effective decision support systems.[78] Besides, the experience available so far suggests that the development of ontologies can be a major step towards achieving interoperability at all three levels, including pragmatics.[72,79,80]

As a prerequisite for such solutions, pre-existing dark data need to be amended with appropriate metadata, in agreement with the established semantic assets. This is a personnel-intensive task, for which dedicated expertise is required, and which has to be repeated whenever the semantic assets are replaced or undergo a major update.[81] Accordingly, it is important to reduce the risk that significant changes become necessary, which might disrupt backwards compatibility, at a point when an ontology has already been employed to classify great amounts of data and metadata. Multiple perspectives, representative of the envisioned community of future users, need to be involved in the development of semantic assets from the first design onward. Accordingly, requirements and experiences from the VIMMP, TaLPas, and SmartNanoTox projects (cf. Acknowledgment) were taken into account for the present work, and ontology drafts were made available to participants of the Horizon 2020 projects MarketPlace and EMMC-CSA within the European Virtual Marketplace Ontology working group.

2.2 Software metadata at the Virtual Materials Marketplace (VIMMP)

The ontology VISO was developed to support the identification of suitable software tools and to standardize the description of software tools as well as modelling and simulation approaches, with the eventual aim of assisting users at accessing the VIMMP marketplace

infrastructure. In particular, VISO will be used to structure the data ingest about software tools at the VIMMP marketplace frontend. The same keywords will then be available to the users to browse the tools and compare them. Accordingly, the main purpose of VISO is to describe materials modelling software, mostly addressing features and capabilities of models and solvers, but also licensing, requirements (e.g., with respect to libraries and operating systems), and compatibilities with other tools; a pre-release version of VISO (`viso-all-branches.ttl`) and an example of its use (`example-viso.ttl`) are included as Supporting Information.

The approach from RoMM, which is followed here, requires a separation of the governing (i.e., constitutive) equations of a model into one or multiple physical equations (PE) which pertain to the basic modelling approach and, by definition, do not depend on the considered material, and one or multiple materials relations (MR) which capture the characteristics of the considered material. Tab. 1 lists the main model types considered by VISO. Therein, the PE type ID refers to a property from OSMO, cf. Section 4.2, where PEs that often occur within models are classified into 25 categories on the basis of RoMM; examples for this are provided as Supporting Information. While the distinction between the PE and the MR may appear to be straightforward from an abstract philosophical point of view, its application to concrete models is often non-unique, and imperfect to a certain extent, since the form and the content of a model cannot normally be separated from each other completely. Similarly, RoMM is also based on a strict distinction between the model (i.e., the theoretical appproach) and the solver (its numerical implementation); accordingly, a `model_feature` here characterizes the underlying physical representation, whereas a `solver_feature` characterizes the implementation and computational representation of the modelling approach by a numerical algorithm. In practice, applying the split between model and solver features to a concrete scenario poses similar challenges as in the case of the PE and the MR. A prototypical example of this are thermostats: Depending on the modelling perspective, they can either be seen as solver features or, e.g., in dissipative particle dynamics (DPD), as fun-

damental ingredients of the model. Moreover, in the latter case, there are arguments both to include them in the PE, since they are necessary and their functional form is not material dependent, or in the MR, since their parameters are related to the material transport properties. The challenges mentioned above are unavoidable when logically decomposing pre-existing complex models and software into a logical and simple structure. Designed to combine information from a wide community of prospective contributors and users, VISO provides a systematic approach for tackling any ambiguities in this context.

Below an upper level (`viso-general`) that addresses general aspects common to all software (e.g., the programming languages), we split VISO into three branches, i.e., electronic (EL, `viso-el`), atomistic-mesoscopic (AM, `viso-am`) for the two molecular granularity levels from RoMM, and continuum (CO, `viso-co`). These branches expand on the model and solver features for each class. The present formulation of these hierarchies was designed by evaluating a representative set of software packages for CFD simulation as well as quantum-mechanical density functional theory (DFT), Monte Carlo (MC), MD, and DPD simulation. Given that many model types can be described from several points of view, VISO allows its users to represent certain approaches in multiple ways; in such cases, the equivalence relation `is_modelling_twin_of` is employed to express that despite being distinct in the ontology, certain instances of different classes can be employed as representations of the same concepts.

Beside features, the other upper classes defined in VISO are `software` (including operating systems, compilers, and software tools), `agent`, `license`, `programming_language`, `modelling_related_entity` (including high level concepts related to modelling, such as model types), `software_interface` (based on the analogous class from SWO;[66] it includes, e.g., graphical, command line, and application programming interfaces), and `software_update`. The latter, in particular, allows to describe the addition/removal of features across versions of a tool.

The main relations defined in VISO to connect these components are briefly described in Tab. 2, and the direct subclasses of the `solver_feature` class are listed in Tab. 3.

Table 1: Models currently considered in developing the VIMMP Software Ontology (VISO), associated physical equation (PE), materials relation (MR), and physical equation type identifier (PE type ID); there, PE and MR are concepts from the Review of Materials Modelling[58] (RoMM), and the PE type ID is introduced in the present work, cf. Tab. 9.

Model type	Physical Equation (PE)	Materials Relation (MR)	PE type ID
DFT	Kohn-Sham eq.	Exchange-correlation functional, composition, etc.	EL.1
MD	Newton's II. law	Inter-particle potentials, composition, connectivity	A.3, M.3
MC	Partition function and ensemble-average expressions	Inter-particle potentials, composition, connectivity	A.4, M.4
DPD	Newton's II. law (conservative force) + drag and random forces	Soft DPD + other potentials, composition, connectivity	M.3
CFD	Mass, momentum, and energy transport eqs. (e.g., Navier-Stokes)	Constitutive relations (e.g., linear transport coefficients)	CO.2
EOS	Fundamental or thermal EOS	Functional form and parameters of the EOS	CO.5

The `model_feature` class has generally a richer structure, and we subdivide it into the (non-disjoint) classes `physical_equation_trait`, `materials_relation_trait`, and `external_condition_trait`. As an example, Fig. 1 includes the upper levels of the class hierarchy for the model features in particle-based models (i.e., in `viso-am`). It can be seen that one of the categories for the MR traits is `force_field`, to be used for statements referring to popular transferable group-contribution based methods (AUA,[83] OPLS,[84,85] TraPPE,[86] etc.); additionally, a finer level of description is available, explicitly identifying the functional forms of the inter-particle potentials that are needed for the model of interest. A possible use of VISO would be, given a force field or a set of MR traits, to identify a code that has them in its set of features.

3 Simulation workflows in materials modelling

3.1 Workflow management systems (WMS)

There is a great variety of environments dealing with workflows. A large number of workflow management systems (WMS) has been implemented over the years, originating mainly from

Table 2: Main relations, i.e., owl:ObjectProperty instances, defined in VISO, for which software_tool is the domain (i.e., class of X). For more details, cf. the Supporting Information. By convention, the namespace owl is employed for keywords of the Web Ontology Language (OWL).

relation (between X and Y)	range (i.e., class of Y)	*brief description*
X has_feature Y	model_feature or solver_feature	*points to features of a tool*
X is_compatible_with Y	software_tool	*compatibility between tools*
X is_tool_for_model Y	model_type	*associates tools with models*
X requires Y	software	*required operating system or library*

Table 3: Classes of solver features defined within viso-el, viso-am, and viso-co. The namespace prefixes are shown in the upper row. Therein, 'el' represents the *electronic* granularity level, 'am' represents *atomistic and mesoscopic*, and 'co' stands for *continuum*; these concepts are defined and discussed in the RoMM document.[58] For further details, cf. the Supporting Information.

subclasses of el_solver_feature (prefix: viso-el)	subclasses of am_solver_feature (prefix: viso-am)	subclasses of co_solver_feature (prefix: viso-co)
basis_set	barostat	continuum_mesh
electron_diagonalization	integrator	divergence_scheme
electron_mixing	electrostatic_solver	gradient_scheme
electron_smearing	geometric_constraint_algorithm	spatial_discretization_scheme
ionic_relaxation	parallelization_scheme	temporal_discretization_scheme
kpoint_mesh	sampling_algorithm	
symmetry_adapted_solver	thermostat	

the fields of data analysis and bioinformatics which in many cases need to rely on large-scale automated computational pipelines. The WMS are meant to facilitate an improved maintainability and robustness compared, e.g., to plain shell scripts. For this purpose, computations and data dependencies are linked logically, leaving details of the task submission – in many cases also including HPC load balancers – to the WMS. By abstracting from all the logistics of file manipulation, copying procedures, and data handling, the management systems thus allow researchers to concentrate on improving the simulation or data-analysis workflow instead of reimplementing standard procedures.[87]

Popular packages include Apache Airflow which allows users to author workflows as directed acyclic graphs;[88] in FireWorks, workflows can be described in Python or markup languages and can be monitored in web interfaces.[89] Luigi, pionieered by Spotify, works on

Figure 1: VISO class `am_model_feature` and its subclasses. This figure and similar ones in the present article have been generated using OWLViz.[82]

a similar basis and employs Python classes for its workflow definition and task scheduling.[90] Snakemake, which is mainly aimed at bioinformatics, has its own domain-specific language to define workflows, including many features oriented towards HPC;[91] beside, generic building environments like GNU make, which also underlies snakemake, can be used directly to automate task dependence and workflow management for data analysis and simulation, as in the case of the main component of the HOPS solver.[92] Moreover, several WMS, including AiiDA,[37,46] Salome/YACS,[47] and the present WMS for *task-based load balancing and auto-tuning in particle-based simulation*[52] (TaLPas), cf. Section 3.2, have been designed particularly for simulation workflows in materials modelling.

3.2 WMS for task-based load balancing and auto-tuning in particle-based simulation (TaLPas)

The TaLPas WMS was developed with the specific needs of the computational molecular engineering community in mind. Accordingly, it was designed to facilitate complex workflows, potentially consisting of a great number of individual simulation runs and data processing steps. Moreover, the molecular simulations performed within these workflows often need to be executed on HPC facilities due to their high computational demands, and the computational costs of single simulations (or tasks, in the case of task-based workflows) vary significantly depending on the simulation input parameters. Typical challenges hence include the management of a great amount of individual tasks, the organization of the results as well as the setup and execution of simulations on diverse and heterogeneous computer system environments and architectures.[52]

The TaLPas WMS addresses these problems. Its overall architecture is shown in Fig. 2. The main core of the environment is the definition of a workflow model. The model defines tasks, which are evaluated by the TaLPas workflow manager; a task includes information about the simulation parameters $\vec{p}$ as well as the simulation program and the commands required to execute it. The WMS comes with a set of selectable task schedulers which handle dependencies between tasks and facilitate their execution on a variety of different HPC systems. To access available computational resources on a HPC system, the scheduler uses a resource manager that keeps track of the availability and usage status of the provided nodes; it also processes the details of the execution for parallel applications via MPI (message passing interface). For the determination of the task execution order, the WMS provides an interface allowing it to be extended by a performance efficiency provider such as Extra-P,[93,94] which estimates the performance and computational resource requirements using the simulation parameters specified in the task. Subsequently, the efficiency provider receives information about the actual time requirements $t_{\vec{p},\vec{N}}$ of any completed tasks, by which Extra-P can refine its performance model, cf. Fig. 2.

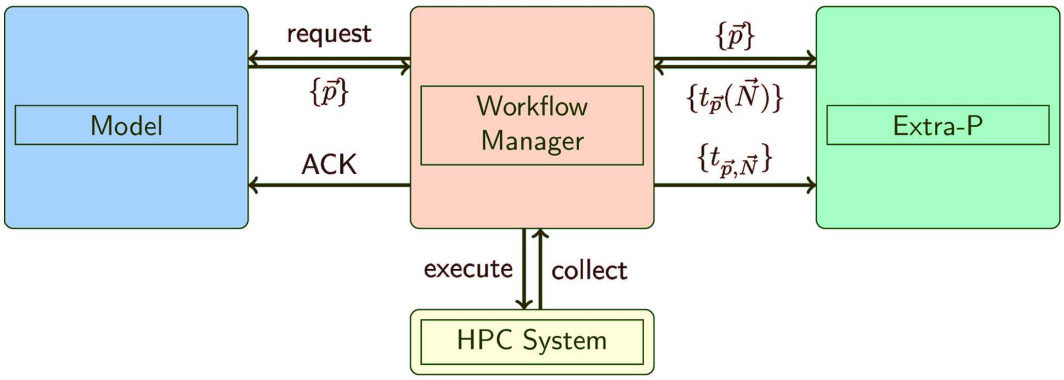

Figure 2: Architecture of the workflow management system (WMS) for *task-based load balancing and auto-tuning in particle-based simulation* (TaLPas): The workflow manager requests tasks from the workflow model. The model responds with a set of parameters $\vec{p}$ specifying the task to be executed. Tasks are then scheduled, and eventually, the results of the execution are collected by the workflow manager in combination with metadata on, e.g., the execution time and the error status. As soon as a task finishes, the model receives an acknowledgment (ACK) about the finished execution to decide on further execution steps. At the same time, performance-related information is communicated to the external performance model provider (here, Extra-P[93,94]). During the scheduling process, the workflow manager can query the performance provider for an estimate of the runtime $t_{\vec{p}}(\vec{N})$ with a given amount of resources $\vec{N}$. This performance model is used to improve the scheduling process.

The TaLPas WMS handles data and files related to all tasks, automatically keeping them seperated by a configurable directory structure. Once the workflow has terminated, this makes it easy for the user to retrieve the simulation outcome. The WMS also collects additional information at runtime, which may help in the case that errors occur during the task execution.

The TaLPas WMS is immediately compatible with the molecular simulation codes *ms2*, cf. Rutkai et al.,[6] and *ls1 mardyn*, cf. Niethammer et al.[17] Beyond case-by-case efforts at achieving compatiblity with individual software architectures, however, TaLPas aims at integrating a multitude of components for the development and optimization of complex task-based auto-tunable workflows. For this purpose, it is advantageous to achieve interoperability with the infrastructures developed on the basis of RoMM, MODA, and EMMO, and to describe simulation software and simulation workflows in terms of semantic assets formalized

as ontologies.

3.3 TaLPas WMS application scenario

EOS parameterization on the basis of high-throughput MC simulations was identified as a proof-of-concept application scenario for the development of the TaLPas WMS and its interoperability with other platforms, such as the VIMMP marketplace. To demonstrate the viability of the present approach, this is applied to phosgene (using the model by Huang et al.[11]), building on previous work by Rutkai and Vrabec;[96] there, the same problem was addressed without employing a dedicated WMS, and without characterizing the provenance of the EOS parameterization as well as the data obtained by molecular simulation.

The present implementation addressing this class of problems uses sampling of state points and fitting with the method developed by Shudler et al.[52] The corresponding workflow can be implemented using the *ms2* simulation program. The data flow and steps to be performed are depicted in Fig. 3, with a focus on technical input/output using files; cf. Section 4 and the Supporting Information for a representation at the logical level, abstracting from the technical implementation of data transfer. A set of thermodynamic states, each of which is defined by the density and the temperature, is simulated in the canonical ensemble. The output of the simulations is processed to obtain multiple derivatives of the Massieu potential, following the formalism proposed by Lustig.[97,98] The Massieu potential derivatives are used to generate the input for an EOS fitter. The result of the fit is not very accurate at the beginning. To increase the accuracy, additional state points are simulated in a series of iterations. The choice of the state points has a considerable influence on the convergence behaviour; in particular, state points close to the vapour-liquid coexistence curve are good candidates to consider for additional simulations. Therefore, intermediate evaluations are performed to refine the state points in an efficient way.

A corresponding workflow model was created for the TaLPas WMS. The workflow model implements the steps from Fig. 3 as well as the application programming interface of the

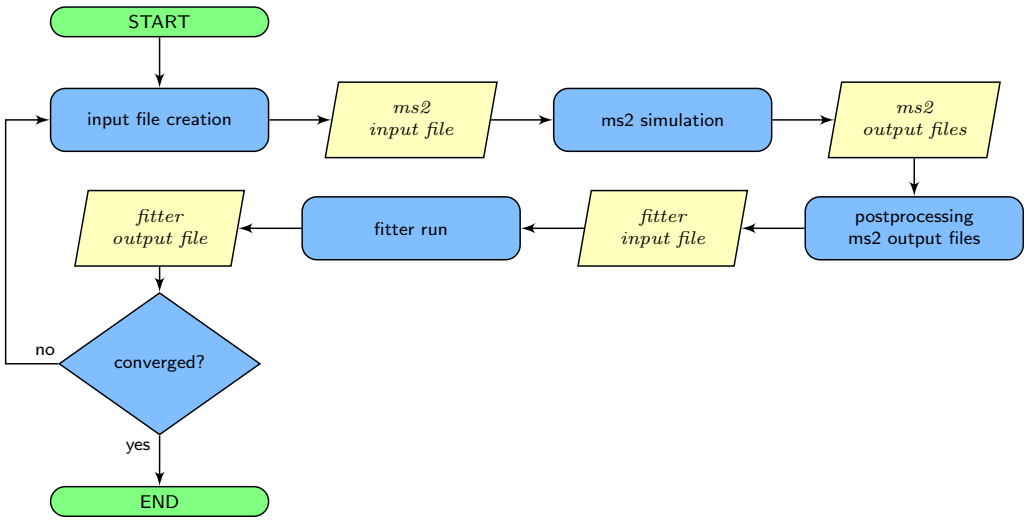

Figure 3: Data flow and program execution of the present equation of state (EOS) parameterization workflow. Yellow boxes represent files, blue ones represent program executions, and diamonds represent conditions and branching.

TaLPas WMS. The model is handed over to the WMS to be executed. The structure and the most important parts of the workflow model are outlined below:

```
class Model:
   def name(self)
      """Returns the name of the workflow"""
   def __init__(self)
      """Initializes the workflow parameters for the
~     refinement as well as an initial set of state
~     points to be processed."""
   def get_task(self)
       """Returns task objects which are intended to
~      be executed on the HPC resources. Ends the
~      workflow by returning an final task object
~      when convergence is achieved."""
   def deploy(self, task, np, mpi)
       """Generates all necessary ms2 input files and
```

```
~       constructs the final MPI command to execute ms2
~       on the HPC system."""
    def record_result(self, task)
        """Records the result of a ms2 simulation
~       run."""
    def createEosInputFromResults(self)
        """Implements the post processing step
~       converting the ms2 output files into an EOS
~       fitter input file"""
    def fitVleCurve(self)
        """Executes the EOS fitter with the generated
~       EOS input file"""
    def refine_around_critical_point(self)
        """Performs refinement around the current
~       critical point creating new state point to be
~       evaluated."""
    def refine_around_VLE(self)
        """Performs refinement around the VLE curve
~       creating new state point to be evaluated."""
```

The method `get_task()` returns the task object, which is prepared for execution on the available HPC resources by the TaLPas WMS. Thereby, a task object is communicated in JavaScript Object Notation (JSON). A typical example for a task is given below:

```
{
  "ID": 53,
  "params": {
    "T": 1.5,
    "rho": 0.01,
```

```
    "step": 0
  },
  "taskdir": "workflow/results/T_1.5/rho_0.01/step_0",
  "deploy": {
    "NP": 4,
    "cmd": ["mpirun", "-np", "4", "./ms2",
~   "EOS_phosgene.par"],
    "nodes": [...]
  },
  "env": "...",
  "starttime": "2019-08-13T15:49:37.938883",
  "endtime": "...",
  "returncode": ...
}
```

The method `deploy()` creates all necessary input for the execution of *ms2* as well as the final MPI command and stores the execution information in a task object. The method `get_task()` hands those task objects over to the WMS for execution. The WMS checks for available resources and starts *ms2* using MPI according to the task object. As soon as the task finishes, the method `record_result()` is called so that the workflow model can record the result for the fitting and iterative refinement step:

4 Representation of simulation workflows

4.1 Graph and diagram notation approaches

The present section discusses how the simulation workflow graphs from MODA can be improved to account for logical data transfer (LDT) and dependencies between workflow elements in a more explicit way; on this basis, in Section 4.2, OSMO is introduced as an

ontology that formalizes the relations visualized by the LDT graph notation and is closely aligned with MODA in its description of the elementary parts of the workflow.

For simulation workflows (and workflows more generally), highly developed formal descriptions exist, including ontologies and graph languages.[57,99,100] Diagram-like notations, which in most cases can be represented as graphs – in the sense employed in graph theory, i.e., as structures that consist of a) nodes and b) edges that connect the nodes – or similar structures such as hypergraphs,[101,102] exist at various degrees of elaboration. At an informal level, this may include, e.g., intuitive sketches drawn on a board to assist a discussion, whereas a great degree of standardization and formalization can yield highly elaborate systems such as machine-readable representations of process flow diagrams. Ontologies and relations between objects can be visualized as graphs;[42] syntactically, graph languages can be defined by graph grammars[103] or other formal approaches such as type graphs.[104] In particular, such approaches have been applied to specify and visualize concurrent and distributed algorithms and workflows.[94,95,105] Often, however, semi-formal specifications of diagram-like notations are provided, which are not machine-processable, but intelligible to human users and standardized to an intermediate extent.

The level of formalization of MODA, a core building block of the EMMC approach to interoperability in materials modelling, is at an intermediate stage: It is defined by a CEN Workshop Agreement[59] (CWA) of the European Committee for Standardization (CEN), and Annex II of RoMM includes a catalogue of MODA examples.[58] However, the descriptors for use cases, models, solvers, and processors in MODA are restricted to plain-text entries, which cannot be easily integrated with other elements of the EMMC-governed semantic technology framework. Moreover, the semantics of the characteristic blue-arrow edges that connect the *sections* (i.e., nodes) of a MODA workflow graph are not defined by the CWA; arrows can represent any association between elements. This is illustrated here by a simple MODA graph consisting of four sections, cf. Fig. 4a); n.b. that in MODA this graph would supplemented by a structured plain-text description of the associated use case, model, solver, and processor

entities. However, the semantics of the blue arrows is subject to the interpretation by a human reader.

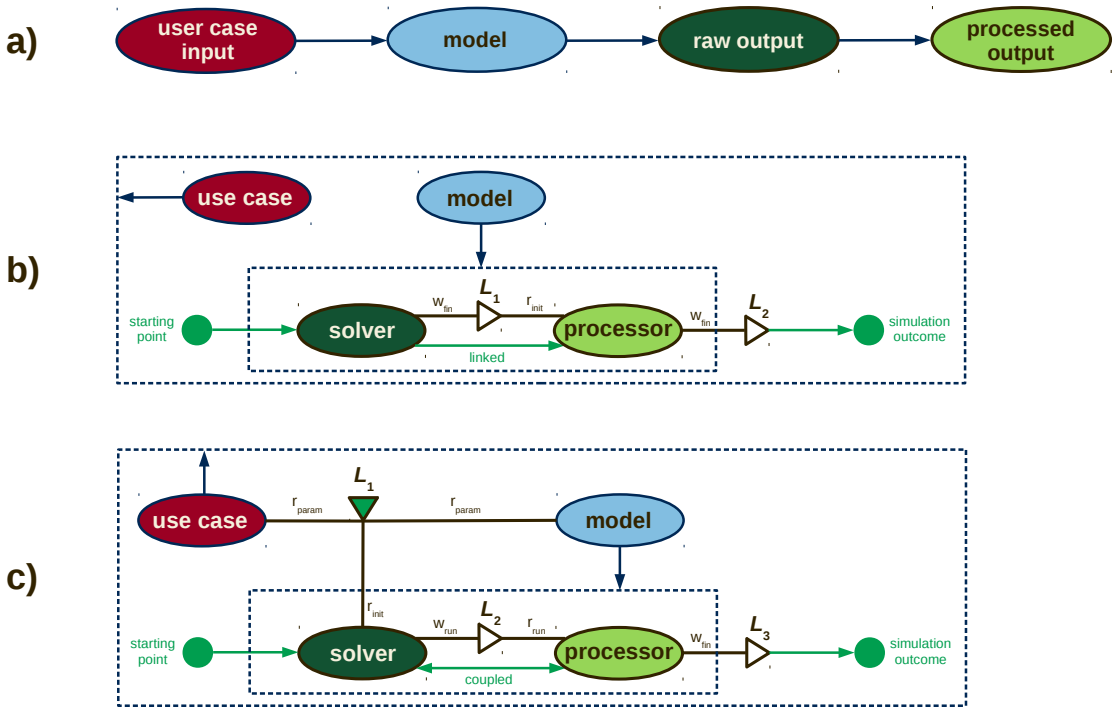

Figure 4: Comparison between the Model Data[59] (MODA) and logical data transfer (LDT) workflow graph notations: a) MODA graph where input characterizing a use case, a model, the raw output of a solver, and the processed output of a postprocessor are connected by blue arrows. b), c) Two LDT graphs corresponding to different scenarios which, in MODA, would both be represented by a).

Therefore, MODA is not sufficiently unambiguous at the level of the workflow graph notation; moreover, it is not an ontology, which would be needed to combine it with the EMMO, other ontologies, and semantic-technology driven infrastructures. On the other hand, existing approaches from the literature cannot be mapped to MODA in a straightforward way; this also holds for OntoCAPE,[42] the ontology that was developed to support CAPE-OPEN.[54,56] It is hence a necessity to develop a more elaborate graph notation and an ontology on the basis of MODA.

The LDT notation clarifies how the use case, model, solver, and processor entities in a workflow relate to each other, cf. Fig. 4b) and c). Therein, ellipses represent sections

(i.e., use cases, models, solvers, and processors); green circles and green arrows represent coupling and linking of elements (as per RoMM[58]), dependencies concerning the order of execution, and aspects related to concurrency and synchronization. Blue arrows point from use cases and models to the part of the workflow to which these entities apply; in particular, if a model applies to a part of a workflow that contains solver entities, these solvers are numerical implementations of this model. Triangles are *logical resources* which are employed to describe how information is transferred between the sections. The triangles point from the source of data to the destination of data. If a triangle is filled (green colour), this implies that a user interaction can occur concerning the data stored at the respective logical resource; this interactivity can consist of any potential steering or input by a user at workflow initialization, execution, or finalization time.

In this way, different workflows, which in MODA would be ambiguously represented by the same graph, e.g., by Fig. 4a), can be distinguished:

- In the case of the workflow represented by the LDT graph from Fig. 4b), the model *applies to* (blue arrow), i.e., is solved and taken into account by, a solver and a processor. The use case *applies to* the entire workflow. The *starting point* (green bullet) of the workflow is the solver, which *is linked to* (green arrow) the processor. Linking here refers to a sequential dependency, i.e., the solver needs to terminate for the processor to start; therefore, in this case, the processor is a *postprocessor*. Upon termination, the solver *writes finally* ($\mathrm{w_{fin}}$) information to the logical resource labelled L_1, which is *read initially* ($\mathrm{r_{init}}$) by the processor; n.b., writing and reading here represents any mechanism of dealing with information, irrespective of the way in which this is implemented. Eventually, the results computed by the postprocessor and written to the logical resource L_2, constitute the overall *simulation outcome* (green bullet).

- The workflow from Fig. 4c) deviates from this in ways that would be hard or impossible to make explicit in MODA notation. Here, the solver *is coupled with* the processor (bidirectional green arrow), i.e., the execution of the two sections is synchronized.

Accordingly, in this case, the processor is a *coupled processor* instead of a postprocessor. Moreover, the use case and the model are parameterized, i.e., they *read parameters* (r_{param}) from a logical resource (here L_1) that is *interactive* (green triangle). Input from L_1 is also used by the solver upon initialization.

An LDT graph for the EOS parameterization example scenario from Section 3.3 is shown in Fig. 5; see also the internal representation from the TaLPas workflow environment, cf. Fig. 3. As in the case of a MODA graph, a description of the use case, model, solver, and processor entities entities always needs to be provided additionally, which can be done at the ontological level following OSMO as outlined in Section 4.2 and Tab. 5.

4.2 Ontology for simulation, modelling and optimization (OSMO)

4.2.1 OSMO, the ontology version of the Model Data (MODA) standard

The ontology OSMO, which is introduced here, is based on the vocabulary and the approach from RoMM;[58] its representation of use cases, solvers, models, and processing is directly based on MODA,[51] and the representation of workflows is based on the LDT notation, cf. Section 4.1, which is itself also an extension of MODA. The class hierarchy for the part of OSMO related to simulation workflows is shown in Fig. 6, including some of the relations that correspond to the visual features of the LDT graph notation; these relations are summarized in Tab. 4. By providing a common semantic basis for workflows that were designed with different tools, OSMO can be employed to consistently integrate data provenance descriptions for materials modelling data from diverse sources.

The detailed description of the four types of section entities (use cases, models, solvers, and processors) in OSMO follows the specification from MODA closely, cf. Tabs. 5 – 8 for the list of aspects (i.e., section descriptors) and Fig. 7 as well as the Supporting Information for technical details.

For example, metadynamics and its variants are solver algorithms; cf. RoMM,[58] p. 59

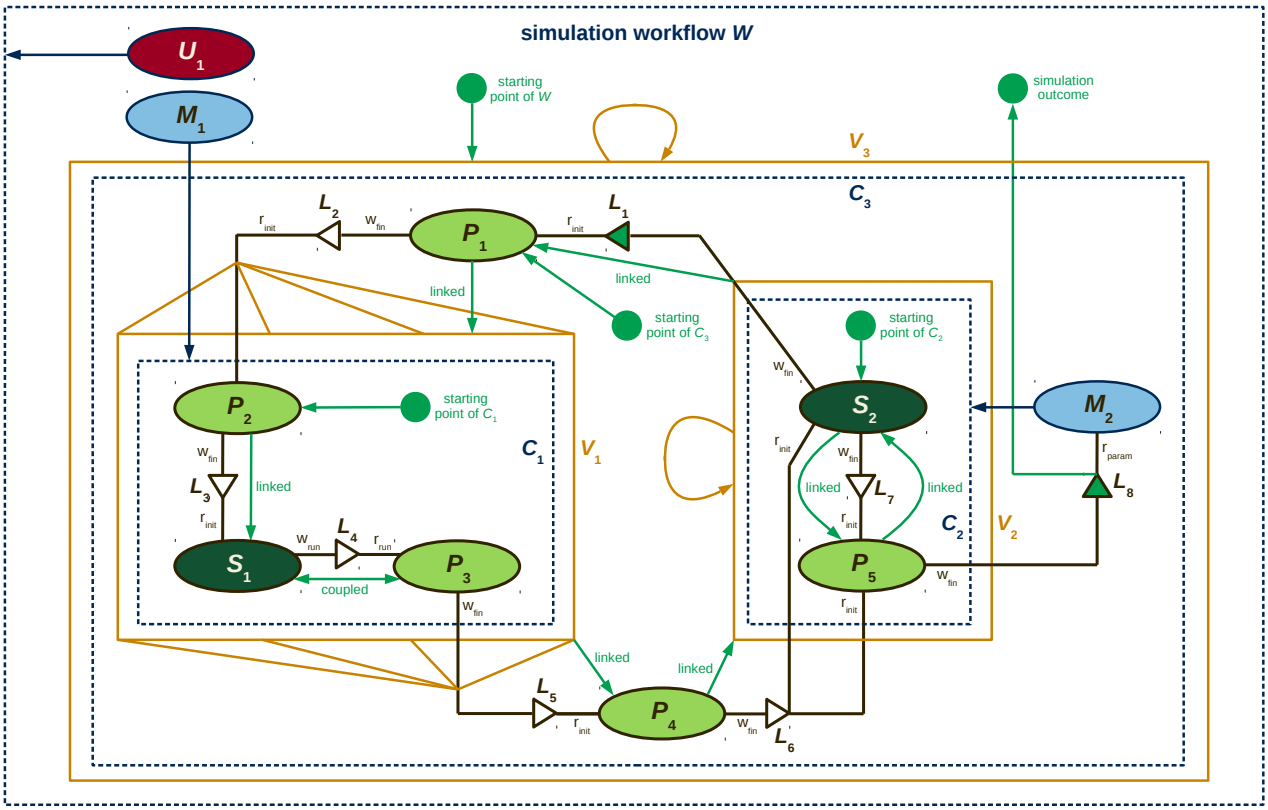

Figure 5: LDT graph representation of the example scenario from Section 3.3, where simulations on the basis of an intermolecular pair potential (model M_1, implemented by the solver S_1) are conducted to parameterize an EOS (model M_2, implemented by the solver S_2) for the purpose of predicting the thermodynamic behaviour of phosgene (use case U_1). The golden solid box with four golden lines at the entry and exit points (virtual graph V_1) represents a concurrent execution of multiple instances of the included blue dashed box (concrete graph C_1), and the golden solid boxes with golden loop-like arrows (virtual graphs V_2 and V_3) represent iterative executions of the included blue dashed boxes (concrete graphs C_2 and C_3). A characterization of this workflow following the ontology for simulation, modelling, and optimization (OSMO), in terse triple language (TTL) format, is included as Supporting Information (`eos-parameterization.ttl`); see also Tab. 4 for the relations from OSMO corresponding to the visual features from LDT graphs.

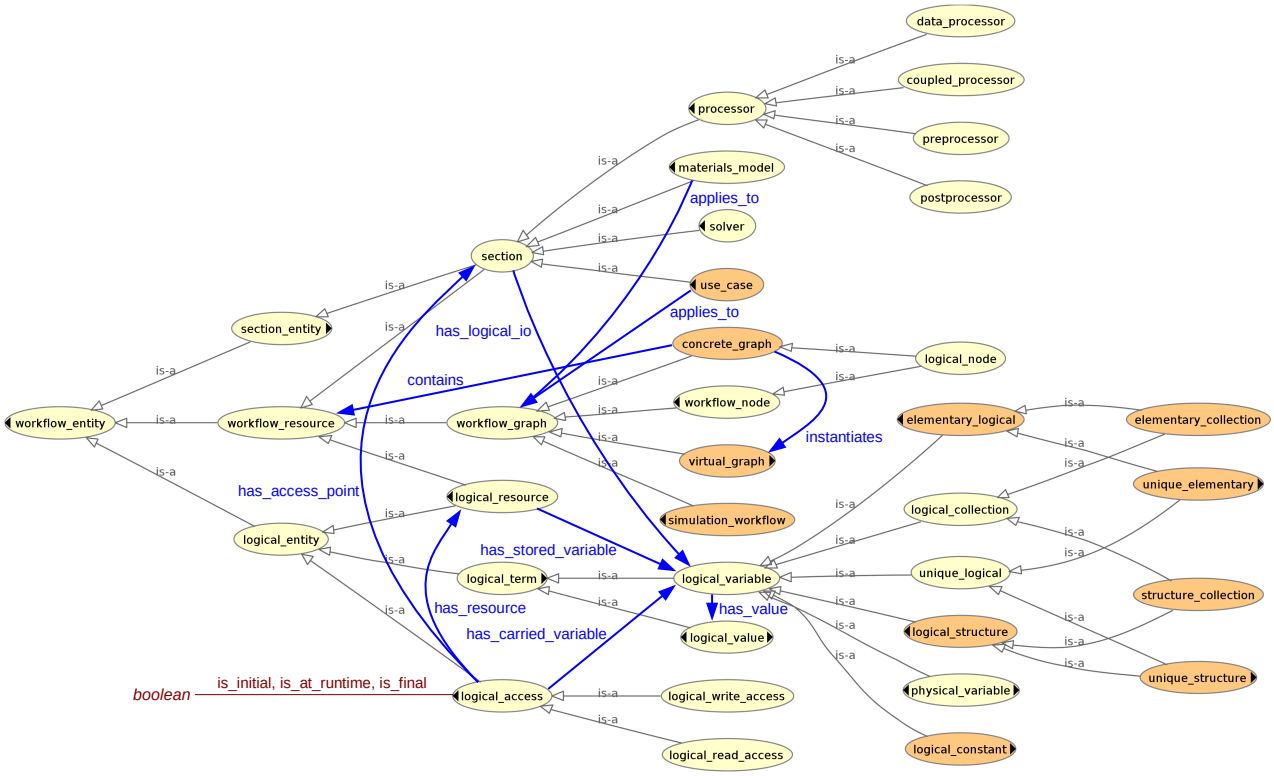

Figure 6: Workflow-related part of the OSMO class diagram, including the `rdfs:subClassOf` relation between classes (grey arrows) and selected additional relations defined in OSMO (blue arrows), as well as boolean features, i.e., instances of `owl:DatatypeProperty`, defined for the class `logical_access` (dark red). By convention, the namespace `rdfs` is employed for RDF Schema (RDFS) keywords.

Table 4: Relations, i.e., `owl:ObjectProperty` instances, defined in OSMO to represent features of simulation workflows, with the corresponding symbols in the LDT graph notation; for a complete specification, cf. the Supporting Information.

relation (between X and Y)	domain (i.e., class of X)	range (i.e., class of Y)	LDT symbol *italics: concise explanation*
X `applies_to` Y	`use_case` or `materials_model`	`workflow_graph`	blue arrow from ellipse X to box Y *Y deals with X*
X `contains` Y	`concrete_graph`	`workflow_resource`	Y is graphically located inside X *Y occurs within X*
X `has_access_point` Y	`logical_access`	`section`	X is a line connected to ellipse Y *LDT by X involves section Y*
X `has_carried_variable` Y	`logical_access`	`logical_variable`	not visualized *LDT by X concerns a transfer of Y*
X `has_internal_lv` Y	`section`	`logical_variable`	not visualized *Y is a logical variable that occurs in X*
X `has_logical_io` Y	`section`	`logical_variable`	not visualized *X reads or writes Y*
X `has_resource` Y	`logical_access`	`logical_resource`	X is a line connected to triangle Y *LDT by X involves resource Y*
X `has_simulation_outcome` Y	`simulation_workflow`	`logical_node`	arrow from Y to a green bullet *resource at Y contains end result of X*
X `has_starting_point` Y	`workflow_graph`	`workflow_node`	green bullet with an arrow to Y *(sub-)workflow X begins at position Y*
X `has_stored_variable` Y	`logical_resource`	`logical_variable`	not visualized *Y can be read from or written to X*
X `has_terminal_point` Y	`workflow_graph`	`workflow_node`	not visualized *(sub-)workflow X ends at position Y*
X `has_value` Y	`logical_variable`	`logical_value`	not visualized *X has the value Y*
X `instantiates` Y	`concrete_graph`	`virtual_graph`	golden solid box around blue dashed box *Y is conditional/multiple execution of X*
X `is_coupled_with` Y	`workflow_graph`	`workflow_graph`	bidirectional green arrow *X and Y are coupled, i.e., synchronized*
X `is_direct_cause_of` Y	`workflow_graph`	`workflow_graph`	green arrow from X to Y *X needs to terminate before Y can begin*
X `is_linked_to` Y	`workflow_graph`	`workflow_graph`	green arrow from X to Y or vice versa (X `is_direct_cause_of` Y or Y `is_direct_cause_of` X)

Table 5: Aspects of a `use_case`, with the corresponding MODA entry numbers[59]

OSMO aspect class name	MODA	aspect and content description (see TTL for details)
`use_case_description`	1.1	*use case summary intended for human readers* content: plain text (elementary datatype `string`)
`use_case_material`	1.2	*characterization of the considered material* content: OSMO/EMMO[61] class `material`
`use_case_geometry`	1.3	*description of the geometry of the considered system* content: plain text, OSMO class `condition`
`use_case_timespan`	1.4	*time interval of a process considered in the use case* content: OSMO class `timespan_information`
`use_case_boundary_condition`	1.5	*thermodynamic, spatio-temporal, or other condition* content: plain text, OSMO class `condition`
`use_case_literature`	1.6	*literature reference related to the use case* content: OTRAS/IAO[106] class `citation`

Table 6: Aspects of a `materials_model`, with the corresponding MODA entry numbers[59]

OSMO aspect class name	MODA	aspect and content description (see TTL for details)
`model_type`	2.1	*PE type following RoMM[58] and Section 4.2.2* content: OSMO class `physical_equation_type`
`model_granularity`	2.2	*granularity level following RoMM[58] and Section 4.2.2* content: `ELECTRONIC`, `ATOMISTIC`, `MESOSCOPIC`, or `CONTINUUM`
`physical_equation`	2.3	*detailed description of the employed PE* content: plain text (i.e., `string`), OSMO class `condition`
`materials_relation`	2.4	*MR following RoMM[58]* (e.g., a pair potential) content: plain text, OSMO class `condition`
`model_boundary_condition`	2.5	*statement on boundary conditions applied to the model* content: plain text, OSMO class `condition`

Table 7: Aspects of a `solver`, with the corresponding MODA entry numbers[59]

OSMO aspect class name	MODA	aspect and content description (see TTL for details)
`solver_method_type`	3.1	*description of the numerical approach* (e.g., MD) content: plain text (i.e., `string`), VISO class `solver_feature`
`solver_software`	3.2	*employed software that implements the approach* content: plain text, VISO class `software_tool`
`solver_timestep`	3.3	*numerical time step employed by the solver* (if applicable) content: plain text, time expressed following QUDT/EMMO[61,107]
`computational_representation`	3.4	*describes how the solver represents the governing equations* content: plain text, OSMO class `condition`
`solver_boundary_condition`	3.5	*numerical boundary conditions applied within the solver* content: plain text, OSMO class `condition`
`solver_parameter`	3.6	*parameter of the solver* content: OSMO class `logical_variable`

Table 8: Aspects of a OSMO `processor`, with the corresponding MODA entry numbers[59]

OSMO aspect class name	MODA	aspect and content description (see TTL for details)
`processor_method_type`	4.2	*describes the methodology employed by the processor* content: plain text (i.e., `string`)
`processor_error_statement`	4.3	*uncertainty, error, or deviation from the most accurate value* content: plain text, VIVO[22] class `accuracy_assertion`

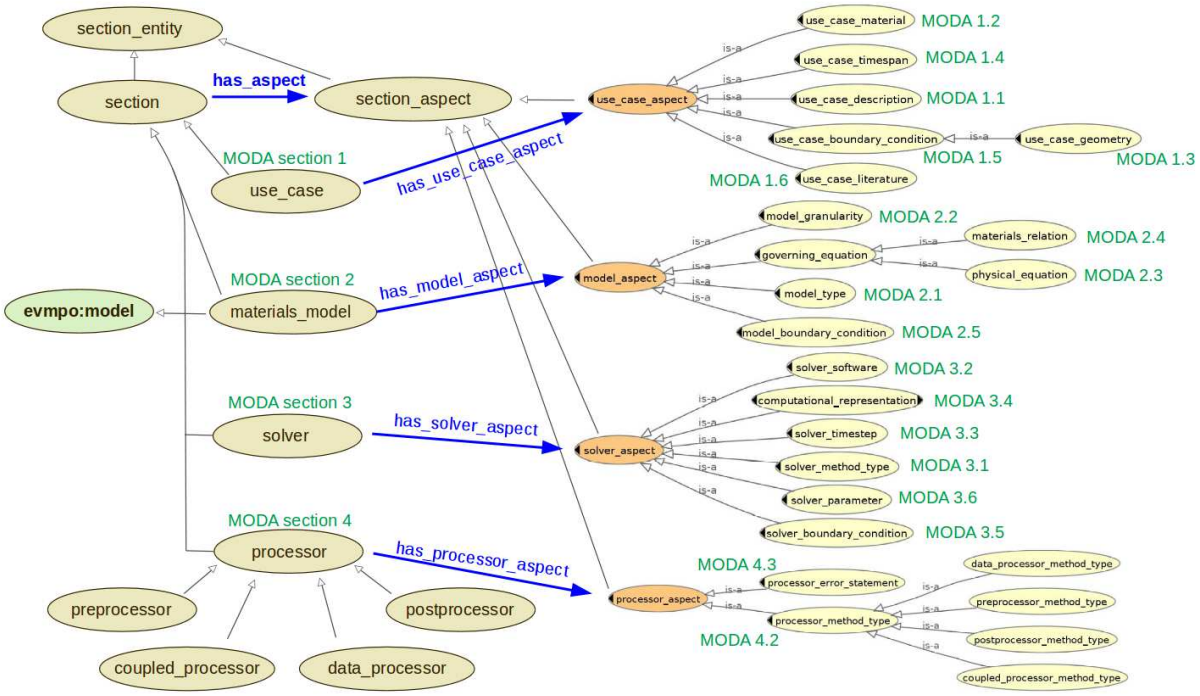

Figure 7: OSMO class `section_entity` and its subclasses. The subclass relation is represented by grey arrows, and `has_aspect` as well as its major subproperties are represented by blue arrows; entry numbers from MODA corresponding to the OSMO aspects, cf. Tabs. 5 – 8, are denoted in green colour. The class `evmpo:model` represents a concept from the European Virtual Marketplace Ontology (work in progress), which defines a model by equivalence to the same concept from the European Materials and Modelling Ontology[61] (EMMO); as a special type of models, `materials_model` from OSMO is a subclass of `evmpo:model`. For further details, cf. the Supporting Information.

("accelerated methods in molecular dynamics"). In MODA,[59] this is specified by entry 3.1. In OSMO, the MODA entry 3.1 corresponds to the aspect `osmo:solver_method_type`, cf. Tab. 7, which points to a `viso:solver_feature` object. For this purpose, the atomistic-mesoscopic branch of VISO provides the class `viso-am:sampling_algorithm`, cf. Tab. 3, which is a subclass of `viso:solver_feature`. Accordingly, the fact that a solver employs well-tempered metadynamics can be denoted as follows:

```
:SX a osmo:solver;
   osmo:has_solver_method_type [
      a osmo:solver_method_type;
      osmo:has_aspect_object_content [
         a viso-am:sampling_algorithm
      ];
      osmo:has_aspect_text_content
         "Well-tempered metadynamics"
   ].
```

4.2.2 Taxonomy of physical equations and relation between OSMO and the Review of Materials Modelling (RoMM)

In OSMO, building on the terminology from RoMM,[58] common PEs in materials modelling are classified into 25 types, represented by subclasses of the OSMO class `physical_equation_type`, at four granularity levels (instances of the OSMO class `granularity_level`), cf. Tab. 9. The characterization of model granularity follows De Baas[58] where the scope of each of the RoMM vocabulary categories is discussed in great detail.

Accordingly, particle-based methods are defined to be *atomistic* if the particles represent single atoms and *mesoscopic* if they represent multiple atoms; by this categorization,[58] e.g., molecular models following the united-atom approach are regarded as mesoscopic. This distinction between atomistic and mesoscopic PEs, however, is only based on the role ascribed

to the discrete particles; therefore, the same equations can be applied at both levels. To ensure that the expressive capacity of OSMO matches that of RoMM, MODA, and EMMO, it is necessary to differentiate between these two levels.[58,59,61] For most purposes, however, this is not a crucial distinction, and they can be jointly referred to as molecular models.

5 Conclusion

The ontologies presented in this work, VISO and OSMO, are intended to play a role as building blocks within a major organized effort toward full interoperability of methods, tools, and environments in computational molecular engineering. This is an ongoing development to which the VIMMP project contributes together with other projects (e.g., MarketPlace). These efforts are coordinated by discussions within the EMMC, an organization open to all modellers, end users, and service providers in the fields of quantum mechanical, molecular, and continuum simulation. By specifying workflows in terms of OSMO, workflow environments such as the TaLPas WMS become interoperable with the VIMMP marketplace and environments from other projects that will be provided at the virtual marketplace frontend. Substantial future work will be needed to develop solutions for facilitating the data ingest into OSMO-compliant infrastructures by providing user-friendly tools to describe simulation workflows in computational molecular engineering according to the approach introduced in the present work.

Acknowledgement

The co-authors M.T.H., G.B., P.C., S.C., M.C., J.E., P.S., M.A.S., I.T.T., and W.L.C. acknowledge funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 760907, *Virtual Materials Marketplace (VIMMP)*, the co-author V.L. from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 686098, *SmartNanoTox: Smart Tools for Gauging Nano Haz-*

Table 9: OSMO physical equation types at four granularity levels on the basis of RoMM[58]

granularity level	PE type ID	RoMM no.	class name and category description
ELECTRONIC	EL.1	1.1	pe_type_electronic_qm_abinitio ab-initio quantum mechanical and first-principle models
	EL.2	1.2	pe_type_electronic_manybody_effective electronic many-body and effective Hamiltonian models
	EL.3	1.3	pe_type_electronic_time_dependent QM modelling of the response to time-dependent fields
	EL.4	1.4	pe_type_electronic_charge_transport statistical charge transport models
	EL.5	1.5	pe_type_electronic_spin_transport statistical electronic spin transport models
ATOMISTIC and MESOSCOPIC, i.e., molecular models	A.1	2.1	pe_type_atomistic_density_functional
	M.1	3.1	pe_type_mesoscopic_density_functional classical-mechanical DFT
	A.2	2.2	pe_type_atomistic_molecular_statics
	M.2	—	pe_type_mesoscopic_molecular_statics energy minimization and molecular statics
	A.3	2.3	pe_type_atomistic_molecular_dynamics
	M.3	3.2	pe_type_mesoscopic_molecular_dynamics MD based on classical equations of motion
	A.4	2.4	pe_type_atomistic_partition_function
	M.4	3.3	pe_type_mesoscopic_partition_function molecular partition-function equations (e.g., for MC)
	A.5	2.5	pe_type_atomistic_spin_model
	M.5	3.4	pe_type_mesoscopic_micromagnetism atomistic spin models (A.5), micromagnetism models (M.5)
	A.6	2.6, 2.7	pe_type_atomistic_statistical_transport
	M.6	3.5	pe_type_mesoscopic_statistical_transport molecular-level statistical transport models
CONTINUUM	CO.1	4.1	pe_type_continuum_solid_mechanics continuum solid mechanics
	CO.2	4.2	pe_type_continuum_fluid_mechanics continuum fluid mechanics
	CO.3	4.3	pe_type_continuum_heat_transfer thermomechanics and continuum modelling of heat transfer
	CO.4	4.4.2	pe_type_continuum_phase_field phase field models and density gradient theory
	CO.5	4.4.1	pe_type_continuum_thermodynamics continuum thermodynamics
	CO.6	4.5	pe_type_continuum_reaction_kinetics continuum modelling of chemical reaction kinetics
	CO.7	4.6	pe_type_continuum_electromagnetism continuum electromagnetism models, including optics
	CO.8	4.7	pe_type_continuum_process_model continuum process models, including flowchart models

ards, and 731032, *NanoCommons*, and the co-authors C.N., P.N., and J.V. from the German Federal Ministry for Education and Research (BMBF) under grant no. 01IH16008B/C/E ('B': co-author P.N., 'C': co-author C.N., 'E': co-author J.V.), *Task-basierte Lastverteilung und Auto-Tuning in der Partikelsimulation (TaLPas)*. The authors thank the EMMC for organizing a series of workshops dedicated to semantic technology, which provided valuable input that contributed to the developments presented in this work, and they thank Y. Bami, A. Bhave, A. Duff, A. M. Elena, A. Fiseni, J. Friis, E. Ghedini, G. Goldbeck, A. Hashibon, P. Klein, R. Kunze, M. Lisal, A. Lister, S. Metz, S. Pařez, B. Planková, G. J. Schmitz, K. Sen, A. Simperler, S. Stephan, G. Summer, and C. Yong for fruitful discussions.

References

(1) Möller, D.; Fischer, J. Vapour liquid equilibrium of a pure fluid from test particle method in combination with *NpT* molecular dynamics simulations. *Mol. Phys.* **1990**, *69*, 463–473.

(2) Lotfi, A.; Vrabec, J.; Fischer, J. Vapour liquid equilibria of the Lennard-Jones fluid from the *NpT* plus test particle method. *Mol. Phys.* **1992**, *76*, 1319.

(3) Vrabec, J.; Hasse, H. Grand equilibrium: Vapour-liquid equilibria by a new molecular simulation method. *Mol. Phys.* **2002**, *100*, 3375–3383.

(4) Deublein, S.; Eckl, B.; Stoll, J.; Lishchuk, S. V.; Guevara Carrión, G.; Glass, C. W.; Merker, T.; Bernreuther, M.; Hasse, H.; Vrabec, J. ms2: A molecular simulation tool for thermodynamic properties. *Comp. Phys. Comm.* **2011**, *182*, 2350–2367.

(5) Glass, C. W.; Reiser, S.; Rutkai, G.; Deublein, S.; Köster, A.; Guevara Carrión, G.; Wafai, A.; Horsch, M.; Bernreuther, M.; Windmann, T.; Hasse, H.; Vrabec, J. ms2: A molecular simulation tool for thermodynamic properties, new version release. *Comp. Phys. Comm.* **2014**, *185*, 3302–3306.

(6) Rutkai, G.; Köster, A.; Guevara Carrión, G.; Janzen, T.; Schappals, M.; Glass, C. W.; Bernreuther, M.; Wafai, A.; Stephan, S.; Kohns, M.; Reiser, S.; Deublein, S.; Horsch, M.; Hasse, H.; Vrabec, J. ms2: A molecular simulation tool for thermodynamic properties, release 3.0. *Comp. Phys. Comm.* **2017**, *221*, 343–351.

(7) Vrabec, J.; Stoll, J.; Hasse, H. A set of molecular models for symmetric quadrupolar fluids. *J. Phys. Chem. B* **2001**, *105*, 12126–12133.

(8) Eckl, B.; Vrabec, J.; Hasse, H. On the application of force fields for predicting a wide variety of properties: Ethylene oxide as an example. *Fluid Phase Equilib.* **2008**, *274*, 16–26.

(9) Schnabel, T.; Vrabec, J.; Hasse, H. Molecular simulation study of hydrogen bonding mixtures and new molecular models for mono- and dimethylamine. *Fluid Phase Equilib.* **2008**, *263*, 144–159.

(10) Engin, C.; Merker, T.; Vrabec, J.; Hasse, H. Flexible or rigid molecular models? A study on vapour-liquid equilibrium properties of ammonia. *Mol. Phys.* **2011**, *109*, 619–624.

(11) Huang, Y.-L.; Heilig, M.; Hasse, H.; Vrabec, J. Vapor-liquid equilibria of hydrogen chloride, phosgene, benzene, chlorobenzene, ortho-dichlorobenzene, and toluene by molecular simulation. *AIChE J.* **2011**, *57*, 1043–1060.

(12) Stephan, S.; Horsch, M. T.; Vrabec, J.; Hasse, H. MolMod – an open access database of force fields for molecular simulations of fluids. *Mol. Sim.* **2019**, *45*, 806–814.

(13) Guevara Carrión, G.; Nieto Draghi, C.; Vrabec, J.; Hasse, H. Prediction of transport properties by molecular simulation: Methanol and ethanol and their mixture. *J. Phys. Chem. B* **2008**, *112*, 16664–16674.

(14) Huang, Y.-L.; Vrabec, J.; Hasse, H. Prediction of ternary vapor-liquid equilibria for 33 systems by molecular simulation. *Fluid Phase Equilib.* **2009**, *287*, 62–69.

(15) Pařez, S.; Guevara Carrión, G.; Hasse, H.; Vrabec, J. Mutual diffusion in the ternary mixture of water + methanol + ethanol and its binary subsystems. *Phys. Chem. Chem. Phys.* **2013**, *15*, 3985–4001.

(16) Werth, S.; Kohns, M.; Langenbach, K.; Heilig, M.; Horsch, M.; Hasse, H. Interfacial and bulk properties of vapor-liquid equilibria in the system toluene + hydrogen chloride + carbon dioxide by molecular simulation and density gradient theory + PC-SAFT. *Fluid Phase Equilib.* **2016**, *427*, 219–230.

(17) Niethammer, C.; Becker, S.; Bernreuther, M.; Buchholz, M.; Eckhardt, W.; Heinecke, A.; Werth, S.; Bungartz, H.-J.; Glass, C. W.; Hasse, H.; Vrabec, J.; Horsch, M. ls1 mardyn: The massively parallel molecular dynamics code for large systems. *J. Chem. Theory Comput.* **2014**, *10*, 4455–4464.

(18) Eckhardt, W.; Heinecke, A.; Bader, R.; Brehm, M.; Hammer, N.; Huber, H.; Kleinhenz, H.-G.; Vrabec, J.; Hasse, H.; Horsch, M.; Bernreuther, M.; Glass, C. W.; Niethammer, C.; Bode, A.; Bungartz, H.-J. 591 TFLOPS multi-trillion particles simulation on SuperMUC. Supercomputing – 28th International Supercomputing Conference (ISC 2013). Springer: Heidelberg, **2013**; pp 1–12.

(19) Tchipev, N.; Seckler, S.; Heinen, M.; Vrabec, J.; Gratl, F.; Horsch, M.; Bernreuther, M.; Glass, C. W.; Niethammer, C.; Hammer, N.; Krischok, B.; Resch, M.; Kranzlmüller, D.; Hasse, H.; Bungartz, H.-J.; Neumann, P. TweTriS: Twenty trillion-atom simulation. *Int. J. HPC Appl.* **2019**, *33*, 838–854.

(20) Asprion, N.; Benfer, R.; Blagov, S.; Böttcher, R.; Bortz, M.; Welke, R.; Burger, J.; von Harbou, E.; Küfer, K.-H.; Hasse, H. INES: Interface between experiments and simulation. *Comp. Aided Chem. Eng.* **2014**, *33*, 1159–1164.

(21) Asprion, N.; Benfer, R.; Blagov, S.; Böttcher, R.; Bortz, M.; Berezhnyi, M.; Burger, J.; von Harbou, E.; Küfer, K.-H.; Hasse, H. INES: An interface between experiments and simulation to support the development of robust process designs. *Chem. Ing. Techn.* **2015**, *87*, 1810–1825.

(22) Cavalcanti, W. L. Virtual Materials Marketplace (VIMMP). **2019**; http://www.vimmp.eu/, date of access: 25th July 2019.

(23) Hashibon, A. MarketPlace. **2019**; http://www.the-marketplace-project.eu/, date of access: 25th July 2019.

(24) Aurich, J.; Schneider, F.; Mayer, P.; Kirsch, B.; Hasse, H. Oberflächenerzeugungs-Morphologie-Eigenschafts-Beziehungen. *Zeitschr. wirtsch. Fabrikbetr.* **2016**, *111*, 213–216.

(25) Burger, J.; Hasse, H. Multi-objective optimization using reduced models in conceptual design of a fuel additive production process. *Chem. Eng. Sci.* **2013**, *99*, 118–126.

(26) Bortz, M.; Burger, J.; Asprion, N.; Blagov, S.; Böttcher, R.; Nowak, U.; Scheithauer, A.; Welke, R.; Küfer, K.-H.; Hasse, H. Multi-criteria optimization in chemical process design and decision support by navigation on Pareto sets. *Comp. Chem. Eng.* **2014**, *60*, 354–363.

(27) Bortz, M.; Burger, J.; von Harbou, E.; Klein, M.; Schwientek, J.; Asprion, N.; Böttcher, R.; Küfer, K.-H.; Hasse, H. Efficient approach for calculating Pareto boundaries under uncertainties in chemical process design. *Ind. Eng. Chem. Res.* **2017**, *56*, 12672–12681.

(28) Forte Serrano, E.; Burger, J.; Langenbach, K.; Hasse, H.; Bortz, M. Multi-criteria optimization for parameterization of SAFT-type equations of state for water. *AIChE J.* **2018**, *62*, 226–237.

(29) von Harbou, E.; Ryll, O.; Schrabback, M.; Bortz, M.; Hasse, H. Reactive distillation in a dividing-wall column: Model development, simulation, and error analysis. *Chem. Ing. Techn.* **2017**, *89*, 1315–1324.

(30) Forte, E.; Jirasek, F.; Bortz, M.; Burger, J.; Vrabec, J.; Hasse, H. Digitalization in thermodynamics. *Chem. Ing. Techn.* **2019**, *91*, 201–214.

(31) Hasse, H.; Lenhard, J. In *Mathematics as a Tool: Tracing New Roles of Mathematics in the Sciences*; Lenhard, J., Carrier, M., Eds.; Springer: Cham, **2017**; pp 93–115.

(32) Lenhard, J.; Hasse, H. In *Technisches Nichtwissen*; Friedrich, A., Gehring, P., Hubig, C., Kaminski, A., Nordmann, A., Eds.; Nomos: Baden-Baden, **2017**; pp 69–84.

(33) Schembera, B.; Iglezakis, D. In *Metadata and Semantic Research*; Garoufallou, E., Sartori, F., Siatri, R., Zervas, M., Eds.; Springer: Cham, **2019**; pp 127–132.

(34) Mühleisen, H.; Jentzsch, A. In *WWW2011 Workshop on Linked Data on the Web (LDOW)*; Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M., Eds.; CEUR-WS: Aachen, **2011**; p 3.

(35) Bicarregui, J. Building and sustaining data infrastructures: Putting policy into practice; **2016**, doi:10.6084/m9.figshare.4055538.v2.

(36) Mons, B. Data Stewardship for Open Science. CRC: Boca Raton, USA, **2018**.

(37) Merkys, A.; Mounet, N.; Cepellotti, A.; Marzari, N.; Gražulis, S.; Pizzi, G. A posteriori metadata from automated provenance tracking: Integration of AiiDA and TCOD. *J. Cheminformat.* **2017**, *9*, 56.

(38) Zelm, M.; Jaekel, F.-W.; Doumeingts, G.; Wollschlaeger, M. Enterprise Interoperability. Wiley: London, **2018**.

(39) Lehne, M.; Sass, J.; Essenwanger, A.; Schepers, J.; Thun, S. Why digital medicine depends on interoperability. *npj Digit. Med.* **2019**, *2*, 79.

(40) Mühleisen, H.; Walther, T.; Tolksdorf, R. A survey on self-organized semantic storage. *Internat. J. Web Informat. Syst.* **2011**, *7*, 205–222.

(41) Heidorn, P. B. Shedding light on the dark data in the long tail of science. *Libr. Trends* **2008**, *57*, 280–299.

(42) Morbach, J.; Wiesner, A.; Marquardt, W. Onto CAPE 2.0: A (re-)usable ontology for computer-aided process engineering. *Comp. Aided Chem. Eng.* **2008**, *25*, 991–996.

(43) Allemang, D.; Hendler, J. *Semantic Web for the Working Ontologist*, 2nd ed.; Morgan Kaufmann: Waltham, USA, **2011**.

(44) Mc Gurk, S.; Abela, C.; Debattista, J. Towards ontology quality assessment. MEPDaW-LDQ 2017 Joint Proceedings. CEUR-WS: Aachen, **2017**; pp 94–106.

(45) Li, H.; Armiento, R.; Lambrix, P. A method for extending ontologies with application to the materials science domain. *Data Sci. J.* **2019**, *18*, 50.

(46) Pizzia, G.; Cepellotti, A.; Sabatini, R.; Marzari, N.; Kozinsky, B. AiiDA: Automated interactive infrastructure and database for computational science. *Comp. Mat. Sci.* **2016**, *111*, 218–230.

(47) Ribes, A.; Caremoli, C. Salomé platform component model for numerical simulation. 31st Annual International Computer Software and Applications Conference, COMPSAC 2007. IEEE Computer Society: Los Alamitos, USA, **2007**; pp 553–564.

(48) Brush, M. H.; Shefchek, K.; Haendel, M. SEPIO: A semantic model for the integration and analysis of scientific evidence. Proceedings of the Joint International Conference on Biological Ontology and BioCreative. CEUR-WS: Aachen, **2016**.

(49) Hastings, J.; Jeliazkova, N.; Owen, G.; Tsiliki, G.; Munteanu, C. R.; Steinbeck, C.; Willighagen, E. eNanoMapper: Harnessing ontologies to enable data integration for nanomaterial risk assessment. *J. Biomed. Semantics* **2015**, *6*, 10.

(50) Burger, J.; Asprion, N.; Blagov, S.; Bortz, M. Simple perturbation scheme to consider uncertainty in equations of state for the use in process simulation. *J. Chem. Eng. Data* **2017**, *62*, 268–274.

(51) Schappals, M.; Mecklenfeld, A.; Kröger, L.; Botan, V.; Köster, A.; Stephan, S.; García, E. J.; Rutkai, G.; Raabe, G.; Klein, P.; Leonhard, K.; Glass, C. W.; Lenhard, J.; Vrabec, J.; Hasse, H. Round robin study: Molecular simuation of thermodynamic properties from models with internal degrees of freedom. *J. Chem. Theory Comput.* **2017**, *13*, 4270–4280.

(52) Shudler, S.; Vrabec, J.; Wolf, F. Understanding the scalability of molecular simulation using empirical performance modeling. Programming and Performance Visualization Tools. Springer: Heidelberg, **2019**; pp 125–143.

(53) Chalk, A. B. G.; Elena, A. M. Task-based parallelism with OpenMP: A case study with DL_POLY_4. *Mol. Sim.* **2019**, doi:10.1080/08927022.2019.1606424.

(54) Belaud, J.-P.; Pons, M. Open software architecture for process simulation: The current status of CAPE-OPEN standard. *Comp. Aided Chem. Eng.* **2002**, *10*, 847–852.

(55) Belaud, J.-P.; Pons, M. CAPE-OPEN: Interoperability in industrial flowsheet simulation software. *Chem. Ing. Techn.* **2014**, *86*, 1052–1064.

(56) Lajmi, A.; Cauvin, S.; Ziane, M. A software factory for the generation of CAPE-OPEN compliant process modelling components. *Comp. Aided Chem. Eng.* **2009**, *27*, 207–212.

(57) Koo, L.; Trokanas, N.; Cecelja, F. A semantic framework for enabling model integration for biorefining. *Comput. Chem. Eng.* **2017**, *100*, 219–231.

(58) De Baas, A. F., Ed. *What Makes a Material Function? Let me Compute the Ways*; EU Publications Office: Luxembourg, **2017**.

(59) *Materials modelling: Terminology, classification and metadata*; CEN workshop agreement 17284, **2018**.

(60) Taxonda dashboard. **2019**; http://emmc.info/taxonda-dashboard/, date of access: 25th July 2019.

(61) Ghedini, E.; Friis, J.; Schmitz, G. J.; Goldbeck, G. European Materials & Modelling Ontology (EMMO). **2019**; http://github.com/emmo-repo/EMMO/, date of access: 25th July 2019.

(62) Smith, A. M.; Katz, D. S.; Niemeyer, K. E. Software citation principles. *PeerJ Comput. Sci.* **2016**, *2*, e86.

(63) Katz, D. S.; Bouquin, D.; Chue Hong, N. P.; Hausman, J.; Jones, C.; Chivvis, D.; Clark, T.; Crosas, M.; Druskat, S.; Fenner, M.; Gillespie, T.; González Beltrán, A.; Gruenpeter, M.; Habermann, T.; Haines, R.; Harrison, M.; Henneken, E.; Hwang, L.; Jones, M. B.; Kelly, A. A.; Kennedy, D. N.; Leinweber, K.; Rios, F.; Robinson, C. B.; Todorov, I. T.; Wu, M.; Zhang, Q. *Software Citation Implementation Challenges*; **2019**; arXiv:1905.08674 [cs.CY].

(64) The CodeMeta Project. **2019**; https://codemeta.github.io/, date of access: 25th July 2019.

(65) The Citation File Format (CFF). **2019**; https://citation-file-format.github.io/, date of access: 25th July 2019.

(66) Malone, J.; Brown, A.; Lister, A. L.; Ison, J.; Hull, D.; Parkinson, H.; Stevens, R. The software ontology (SWO): A resource for reproducibility in biomedical data analysis, curation and digital preservation. *J. Biomed. Semant.* **2014**, *5*, 25.

(67) Gil, Y.; Ratnakar, V.; Garijo, D. OntoSoft: Capturing scientific software metadata.

Proceedings of the Eighth ACM International Conference on Knowledge Capture (K-CAP). ACM: New York, **2015**; p 32.

(68) Ervik, Å.; Mejía, A.; Müller, E. A. Bottled SAFT: A web app providing SAFT-γ Mie force field parameters for thousands of molecular fluids. *J. Chem. Inf. Model.* **2016**, *56*, 1609–1614.

(69) Ervik, Å.; Jiménez Serratos, G.; Müller, E. A. raaSAFT: A framework enabling coarse-grained molecular dynamics simulations based on the SAFT-γ Mie force field. *Comp. Phys. Comm.* **2017**, *212*, 161–179.

(70) Bazhirov, T. Data-centric online ecosystem for digital materials science. **2019**, arXiv:1902.10838 [cond-mat.mtrl-sci].

(71) Asuncion, C. H.; van Sunderen, M. J. In *Enterprise Architecture, Integration and Interoperability*; Bernus, P., Doumeingts, G., Fox, M., Eds.; Springer: Heidelberg, **2010**; pp 164–175.

(72) Weidt Neiva, F.; David, J. M. N.; Braga, R.; Campos, F. Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. *Informat. Software Technol.* **2016**, *72*, 137–150.

(73) Schoop, M.; de Moor, A.; Dietz, J. The pragmatic web: A manifesto. *Comm. ACM* **2006**, *49*, 75–76.

(74) Weidt Neiva, F.; David, J. M. N.; Braga, R.; Borges, M. R. S.; Campos, F. SM2PIA: A model to support the development of pragmatic interoperability requirements. 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). IEEE: New York, **2016**; pp 119–128.

(75) Hristova-Bogaerds, D.; Asinari, P.; Konchakova, N.; Bergamasco, L.; Marcos Ramos, A.; Goldbeck, G.; Hoeche, D.; Swang, O.; Schmitz, G.; Klein, P.;

Kraft, T.; Macioł, P.; Iannone, M.; de Baas, A. *Translators Guide, version 2*; **2018**; https://emmc.info/translators-guide-2/, date of access: 25th July 2019.

(76) Schmitz, G. J. Microstructure modeling in integrated computational materials engineering (ICME) settings: Can HDF5 provide the basis for an emerging standard for describing microstructures? *JOM* **2016**, *68*, 77–83.

(77) Oberkampf, H.; Krieg, H.; Senger, C.; Weber, T.; Colsman, W. Allotrope Data Format: Semantic data management in life sciences. 11th International SWAT4HCLS Conference. **2018**.

(78) Asprion, N.; Bortz, M. Process modeling, simulation and optimization: From single solutions to a multitude of solutions to support decision making. *Chem. Ing. Techn.* **2018**, *90*, 1727–1738.

(79) De Leenheer, P.; Christiaens, S. Mind the gap! Transcending the tunnel view on ontology engineering. Proceedings of the 2nd International Conference on Pragmatic Web. ACM: New York, **2007**; pp 75–82.

(80) Gan, M. Enterprise isomorphic mapping mechanism: Towards ontology interoperability in EIS development. 2009 IEEE International Conference on e-Business Engineering. IEEE Computer Society: Los Alamitos, USA, **2009**; pp 340–345.

(81) Schembera, B.; Durán, J. M. Dark data as the new challenge for big data science and the introduction of the scientific data officer. *Philos. Technol.* **2019**, 1–13.

(82) Horridge, M. OWLViz. **2010**, https://protegewiki.stanford.edu/wiki/OWLViz, date of access: 11th November 2019.

(83) Ungerer, P.; Beauvais, C.; Delhommelle, J.; Boutin, A.; Rousseau, B.; Fuchs, A. H. Optimization of the anisotropic united atoms intermolecular potential for n-alkanes. *J. Chem. Phys.* **2000**, *112*, 5499–5510.

(84) Jorgensen, W. L. Optimized intermolecular potential functions for liquid alcohols. *J. Phys. Chem.* **1986**, *90*, 1276–1284.

(85) Jorgensen, W. L.; Maxwell, D. S.; Tirado Rives, J. Development and resting of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

(86) Chen, B.; Potoff, J. J.; Siepmann, J. I. Monte Carlo calculations for alcohols and their mixtures with alkanes. Transferable potentials for phase equilibria. 5. United-atom description of primary, secondary, and tertiary alcohols. *J. Phys. Chem. B* **2001**, *105*, 3093–3104.

(87) Lampa, S.; Alvarsson, J.; Spjuth, O. Towards agile large-scale predictive modelling in drug discovery with flow-based programming design principles. *J. Cheminformat.* **2016**, *8*, 67.

(88) Beauchemin, M. Airflow documentation. **2019**; http://airflow.apache.org/, date of access: 25th July 2019.

(89) Jain, A.; Ong, S. P.; Chen, W.; Medasani, B.; Qu, X.; Kocher, M.; Brafman, M.; Petretto, G.; Rignanese, G.; Hautier, G.; Gunter, D.; Persson, K. A. FireWorks: A dynamic workflow system designed for high-throughput applications. *Concur. Computat. Pract. Exper.* **2015**, *27*, 5037–5059.

(90) Erdmann, M.; Fischer, B.; Fischer, R.; Rieger, M. Design and execution of make-like, distributed analyses based on Spotify's pipelining package Luigi. *J. Phys.: Confer. Ser.* **2017**, *898*, 072047.

(91) Köster, J.; Rahmann, S. Snakemake: A scalable bioinformatics workflow engine. *Bioinform.* **2012**, *28*, 2520–2522.

(92) Robinson, A. R.; Haley, P. J.; Lermusiaux, P. F. J.; Leslie, W. G. Predictive skill, predictive capability and predictability in ocean forecasting. OCEANS '02 MTS/IEEE. IEEE: Piscataway, USA, **2002**; pp 787–794.

(93) Calotoiu, A.; Beckingsale, D.; Earl, C. W.; Hoefler, T.; Karlin, I.; Schulz, M.; Wolf, F. Fast multi-parameter performance modeling. 2016 IEEE International Conference on Cluster Computing. IEEE Computer Society: Los Alamitos, USA, **2016**; pp 172–181.

(94) Shudler, S.; Calotoiu, A.; Hoefler, T.; Wolf, F. Isoefficiency in practice: Configuring and understanding the performance of task-based applications. *SIGPLAN Not.* **2017**, *52*, 131–143.

(95) Cheong, K.; Garijo, D.; Cheung, W. K.; Gil, Y. PSM-Flow: Probabilistic subgraph mining for discovering reusable fragments in workflows. 2018 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE: Piscataway, USA, **2018**; pp 166–173.

(96) Rutkai, G.; Vrabec, J. Empirical fundamental equation of state for phosgene based on molecular simulation data. *J. Chem. Eng. Data* **2015**, *60*, 2895–2905.

(97) Lustig, R. Direct molecular NVT simulation of the isobaric heat capacity, speed of sound, and Joule-Thomson coefficient. *Mol. Sim.* **2010**, *37*, 457–465.

(98) Lustig, R. Statistical analogues for fundamental equation of state derivatives. *Mol. Phys.* **2012**, *110*, 3041–3052.

(99) Pathak, J.; Caragea, D.; Honavar, V. G. In *Semantic Web and Databases*; Bussler, C., Tannen, V., Fundulaki, I., Eds.; Springer: Heidelberg, **2005**; pp 41–56.

(100) Rospocher, M.; Ghidini, C.; Serafini, L. An ontology for the business process modelling notation. Formal Ontology in Information Systems: Proceedings of the Eighth International Conference. IOS Press: Amsterdam, **2014**; pp 133–146.

(101) Comuzzi, M. Ant-colony optimisation for path recommendation in business process execution. *J. Data Semant.* **2019**, *8*, 113–128.

(102) Wiśniewski, R.; Wiśniewska, M.; Jarnut, M. C-exact hypergraphs in concurrency and sequentiality analyses of cyber-physical systems specified by safe Petri nets. *IEEE Access* **2019**, *7*, 13510 – 13522.

(103) Ehrig, H.; Engels, G.; Kreowski, H.; Rozenberg, G. *Handbook of Graph Grammars and Computing by Graph Transformation*; World Scientific: River Edge, USA, **1999**; Vol. 2.

(104) Corradini, A.; König, B. Specifying graph languages with type graphs. *J. Logical Algebr. Meth. Program.* **2019**, *104*, 176–200.

(105) Bauderon, M.; Métivier, Y.; Mosbah, M.; Sellami, A. Graph relabelling systems: A tool for encoding, proving, studying and visualizing distributed algorithms. *Electr. Notes Theor. Comp. Sci.* **2001**, *51*, 93–107.

(106) Ceusters, W.; Smith, B. Aboutness: Towards foundations for the information artifact ontology. Proceedings of the International Conference on Biomedical Ontology. CEUR-WS: Aachen, **2015**.

(107) Hodgson, R.; Keller, P. J.; Hodges, J.; Spivak, J. QUDT ontologies. **2019**; http://www.qudt.org/, date of access: 25th July 2019.