A Multi-Scale Cross-Domain Thermochemical Knowledge-Graph

Sebastian Mosbach,[†,‡] Angiras Menon,[†] Feroz Farazi,[†] Nenad Krdzavac,[‡] Xiaochi Zhou,[†] Jethro Akroyd,[†,‡] and Markus Kraft[*,†,‡,¶]

[†]*Department of Chemical Engineering and Biotechnology, University of Cambridge, Philippa Fawcett Drive, Cambridge CB3 0AS, United Kingdom*

[‡]*Cambridge Centre for Advanced Research and Education in Singapore (CARES), CREATE Tower #05-05, 1 Create Way, Singapore 138602*

[¶]*School of Chemical and Biomedical Engineering, Nanyang Technological University, 62 Nanyang Drive, Singapore 637459*

E-mail: mk306@cam.ac.uk

Phone: +44 (0)1223 762784

Abstract

In this paper, we develop a set of software agents which improve a knowledge-graph containing thermodynamic data of chemical species by means of quantum chemistry calculations and error-cancelling balanced reactions. The knowledge-graph represents species-associated information by making use of the principles of linked data as employed in the Semantic Web, where concepts correspond to vertices and relationships between the concepts correspond to edges of the graph. We implement this representation by means of ontologies, which formalise the definition of concepts and their relationships, as a critical step to achieve interoperability between heterogeneous data formats as well as software. The agents, which conduct quantum chemistry calculations

and derive estimates of standard enthalpies of formation, update the knowledge-graph with newly obtained results, improving data values and adding nodes and connections between them. A key distinguishing feature of our approach is that it extends an existing, general-purpose knowledge-graph, called J-Park Simulator (theworldavatar.com), and its eco-system of autonomous agents, thus enabling seamless cross-domain applications in wider contexts. To this end, we demonstrate how quantum calculations can directly affect the atmospheric dispersion of pollutants in an industrial emissions use-case.

Introduction

Optimising industrial operations is critical for mitigating their environmental impact, that is, maximising energy efficiency and minimising pollution and waste of resources. In order for this to be most effective, systems must not be considered in isolation, but rather connected to one another as part of a network. Furthermore, implementing this requires models of all systems involved. Such reasoning naturally leads to the ideas of Industry 4.0[1] and the Internet of Things (IoT),[2] where digital twins of real objects can communicate with each other via the internet.

In case one or more of the models in the network involve chemistry in some form, chemical models need to be built, using chemical data about species and reactions. Numerous databases exist for such purposes, containing various levels of detail. For instance, some of the most widely used are Reaxys,[3] PubChem,[4] and the CAS Registry,[5] each of which include a wealth of chemical and physical substance information on in excess of 10^8 compounds. Amongst the largest is the Chemical Universe Database GDB-17[6] with more than 10^{11} structures, though it contains SMILES strings[7] only. At a smaller scale, PrIMe[8] is largely focussed on combustion, and includes not only a warehouse of experimental and computational data, but also an associated set of tools for a variety of tasks related to model development. At the most detailed end of the spectrum, and even more focussed on

thermodynamic properties, are collections that include quantum chemistry calculations. For example, Ramakrishnan et al.[9] have conducted quantum calculations for a GDB-17 subset of more than 10^5 structures. NIST's CCCBDB[10] provides extensive experimental and quantum calculation data on the thermochemistry of 1968 small gas-phase species. Active Thermochemical Tables (ATcT)[11] use a statistical approach to synthesise accurate and consistent thermodata from experiments and computations for 1617 species. Highly accurate thermochemistry has been calculated for 219 molecules relevant to combustion.[12]

With so much data available from so many different sources, inconsistencies are ubiquitous, both in terms of naming of species and in terms of data values of thermodynamic, transport, and kinetic properties of species and reactions (see *e.g.* Lambert and West[13]). Combined with the complexity of fuel combustion models, with hundreds or even thousands of species and reactions, it is clear that automation is inevitable. A number of attempts have been made to automate the generation of kinetic mechanisms, and in particular the generation of thermodynamic data. The Reaction Mechanism Generator (RMG)[14] constructs species and reactions based on a known set of rules, which also includes estimation of thermodynamic, kinetic, and transport properties. Building upon RMG, Keçeli et al.[15] have developed a Predictive Automated Computational Thermochemistry (PACT) software package which automatically generates thermodynamic data from quantum calculations for species involved in the combustion of an arbitrary hydrocarbon fuel. Li et al.[16] have implemented a self-evolving thermochemistry machine which uses a convolutional neural network to predict species formation enthalpies. The network is trained on a database of density functional theory (DFT) calculations and estimates its own prediction errors, launches new DFT calculations whenever the error exceeds a given threshold, and includes the new results into its training database, thus continuously and automatically improving its predictions.

As the above examples show, automating specific, well-defined tasks can be achieved through tailor-made code. However, the challenge becomes much harder as soon as one considers more general contexts, where, in order to make substantial progress with automation,

a more fundamental, systematic approach is required. One such approach utilises knowledge-graphs, which represent information by making use of the principles of Linked Data[17,18] as employed in the Semantic Web,[19] where concepts correspond to vertices and relationships between concepts correspond to edges of the graph. This representation is implemented by means of ontologies, which formalise the definition of concepts and their relationships through collections of subject-predicate-object triples. The power of knowledge-graphs at a large scale has become abundantly clear in various applications over the last few years.[20,21] In the field of chemistry, it has been recognised that subject-predicate-object representations of chemical data are of value, in particular for automation,[22] with PubChemRDF[23] being one of many examples.

In previous work on knowledge-graphs in the chemistry domain,[24] we have developed an ontology for quantum chemistry calculations based on Chemical Markup Language (CML[25]), called OntoCompChem.[26] We have furthermore developed an ontology for chemical reaction mechanisms, called OntoKin.[27] In order to connect these two worlds, linking reaction mechanisms and quantum chemistry,[28] we have introduced an ontology for unique chemical species, called OntoSpecies, thus integrating the union of all these concepts into a single knowledge-graph.

The purpose of the present paper is to employ a knowledge-graph approach, specifically with newly-developed software agents, to improve thermodynamic data for chemical species, and apply it to a multi-scale, multi-domain combustion example. Whilst this process can also be automated by means of other methods, our approach achieves this in a fashion that, by design, guarantees interoperability between data and software, and thus allows seamless integration with complex cross-domain applications. As such, the knowledge-graph approach renders automated thermochemistry extensible beyond its original scope.

The paper is structured as follows. The next section explains the context, namely the general-purpose knowledge-graph within which the present work is placed. Subsequently, we outline a generic agent design and describe a number of thermochemical agents and how

they are integrated. We then provide a cross-domain use-case of atmospheric dispersion of industrial pollution, and finally draw conclusions.

The World Avatar

The term 'World Avatar' intends to capture the idea of representing every aspect of the real world in a digital 'mirror' world. This is essentially an extension of the Digital Twin concept, where, taking an example from Industry 4.0, a device or a unit operation in an industrial process has a corresponding virtual representation. Lifting the restriction to industrial devices, thus considering virtualisation of any abstract concept or process, is a logical continuation, similar to extending the Internet of Things to the Internet of Services and beyond.

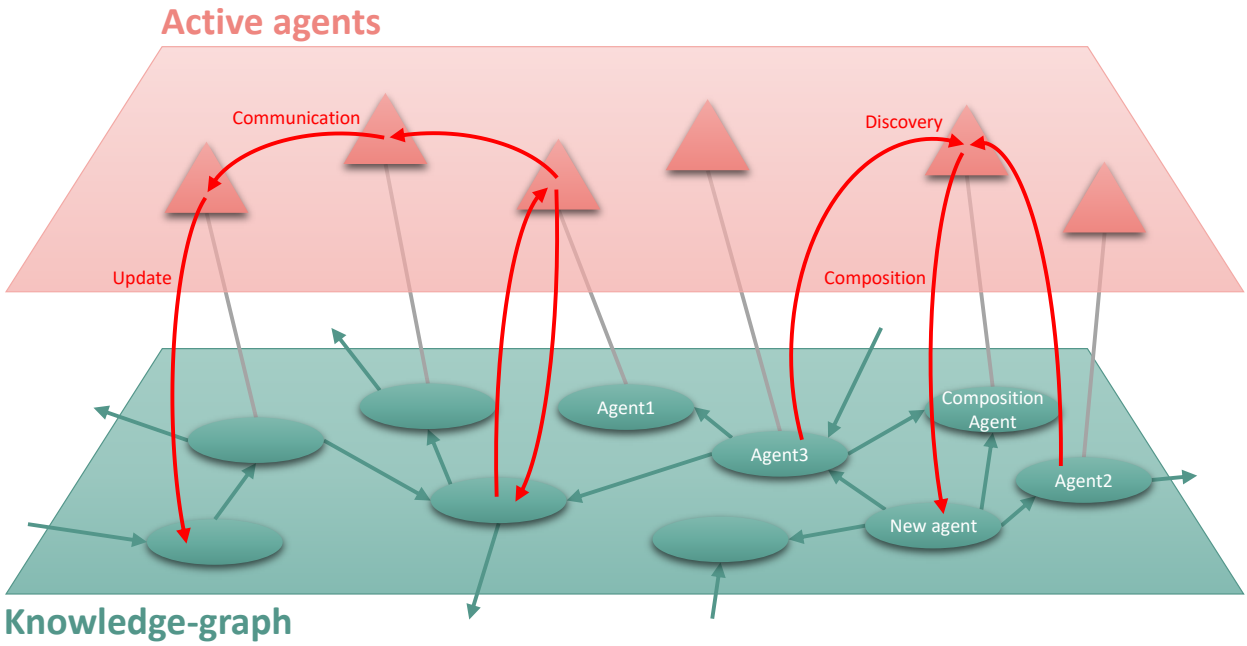


Figure 1: Current design of the J-Park Simulator (JPS) as an implementation of a World Avatar. Agents are part of the knowledge-graph and operate on it.

The J-Park Simulator (JPS)[29,30] is an implementation of the World Avatar concept. Figure 1 illustrates its main underlying principles. At the heart of the JPS lies a knowledge-graph that is intended to be general-purpose and all-encompassing. As a representation of data, the key distinguishing feature of a knowledge-graph is that individual items of information

are linked to each other. In the JPS, this Linked Data is achieved through the use of Internationalised Resource Identifiers (IRIs), essentially generalised web-addresses, in line with the Semantic Web. The concepts in the knowledge-graph and the links between them are implemented by means of ontologies for various domains. These include process engineering (OntoCAPE[31]), Eco-Industrial Parks (EIPs) (OntoEIP[32–34]), electrical power systems (OntoPowerSys[35]), and 3D buildings and landscapes (OntoCityGML[30]). In the chemistry domain, we have developed ontologies for the subdomains of quantum chemistry (OntoCompChem[26]), species (OntoSpecies[28]), and kinetic reaction mechanisms (OntoKin[27]). In addition, various subgraphs of the Linked Open Data (LOD) Cloud[36] are also connected to the JPS knowledge-graph, in particular DBpedia.[37,38]

Beyond mere data representation, the JPS contains an eco-system of software agents that act autonomously and operate on the knowledge-graph, constantly updating it, as also illustrated in Fig. 1. Crucially, the agents themselves are part of the knowledge-graph, governed by the OntoAgent[39] ontology. As agents are continuously operating on the knowledge-graph, it evolves in time. In particular, we have developed agents for automatic agent discovery and composition,[39] *i.e.* agents that create new, composite agents for more complex tasks. And further, in order to facilitate the usage of agents and simplify identification of an agent suitable for a specific task in an agent-rich environment, with an abundance of services available, we have established an agent market place based on block-chain technology and Smart Contracts.[40]

The JPS started with a focus on virtualising industrial operations within the Jurong Island EIP in Singapore,[33,41,42] but has since expanded well beyond this original scope. It has been employed in a variety of cross-domain applications that require interoperability between heterogeneous software and data formats, such as optimal site selection of nuclear power plants[43] and industrial pollution prediction of ships and power plants.[30] In addition, scenario planning is possible using the Parallel World Framework,[44] which generically allows asking what-if questions and exploring alternatives in complex multi-domain applications.

Design and implementation

In this section, we describe the agents that form the basis of the present paper. We give some theoretical background of what they do and outline their architecture as well as some key features of their implementation. We also explain how they are integrated, *i.e.* how they work with each other and the knowledge-graph, and thus how they are part of the wider World Avatar agent eco-system as described in the previous section.

The chemical knowledge-graph

In order to provide the necessary context of what environment the agents are operating in, we briefly recall here what chemical entities are represented in the knowledge-graph and how. As mentioned above, we have previously created three ontologies for a number of chemical concepts, populated triple-stores with collections of instances, and established links between them:

The OntoKin[27] ontology covers chemical reaction mechanisms – collections of chemical elements, species, and reactions together with their associated thermodynamic, kinetic, and transport data. A web-interface is available[45] that allows up- and download, and basic queries.[46] In addition, we have developed an API that allows programmatic up- and downloading, and import of, export of, and conversion between widely used mechanism file formats.

The OntoCompChem[26] ontology is concerned with representing quantum calculations. Whilst the ontology itself is not intrinsically specific to a particular piece of software, the parser and API we have written for file import is designed for Gaussian 09/16 calculations. A web-interface is also available.[47] Representable quantities include empirical formula, InChI and SMILES strings, molecular geometry, level of theory and basis set, rotational and vibrational frequencies, rotational constants, and electronic energies.

Thirdly, we have introduced the OntoSpecies ontology,[28] primarily for the purpose of

identifying chemical species uniquely. The ontology captures basic physical and chemical information about species, such as elemental composition, connectivity between atoms, molecular geometry, and in particular standard enthalpy of formation. In itself, OntoSpecies is not meant to and does not need to store much information – its main value lies in providing links via IRIs, connecting quantum calculations with each other as well as species in mechanisms, which are labelled using arbitrary strings. These IRI links are used to disambiguate arbitrary chemical labels, distinguishing different species with identical labels as well as recognising identical species with different labels, thus circumventing naming inconsistencies.

The agents described in the following critically depend upon this infrastructure, with its links between relevant concepts, being in place.

Agent template

When developing agents for various tasks, one quickly arrives at the realisation that many design features are entirely generic, *i.e.* independent of the specific task. Such design features include most importantly listening and responding to requests via the internet (implemented through HTTP), submission and monitoring of jobs to a resource manager (*e.g.* SLURM) on an HPC system, and managing input and output files associated to a job as well as their transfer between the hardware platforms involved. We have thus developed a template that is, within reason, applicable to 'any' (Linux) executable.

Figure 2 illustrates the main components of this generic agent and how it interacts with the knowledge-graph. The agent (the triangle shaded in red) consists of two executable elements: An asynchronous watching process and a generic executable. The asynchronous watcher handles HTTP requests and responses, through which other agents can request jobs. Secondly, it manages input and output files, which are being stored in separate folders associated to each job. And thirdly, it takes care of submitting jobs to a resource manager tool on an HPC system as well as monitoring any jobs that are running. It does this using a status file associated to each job in its corresponding folder. This design is robust to

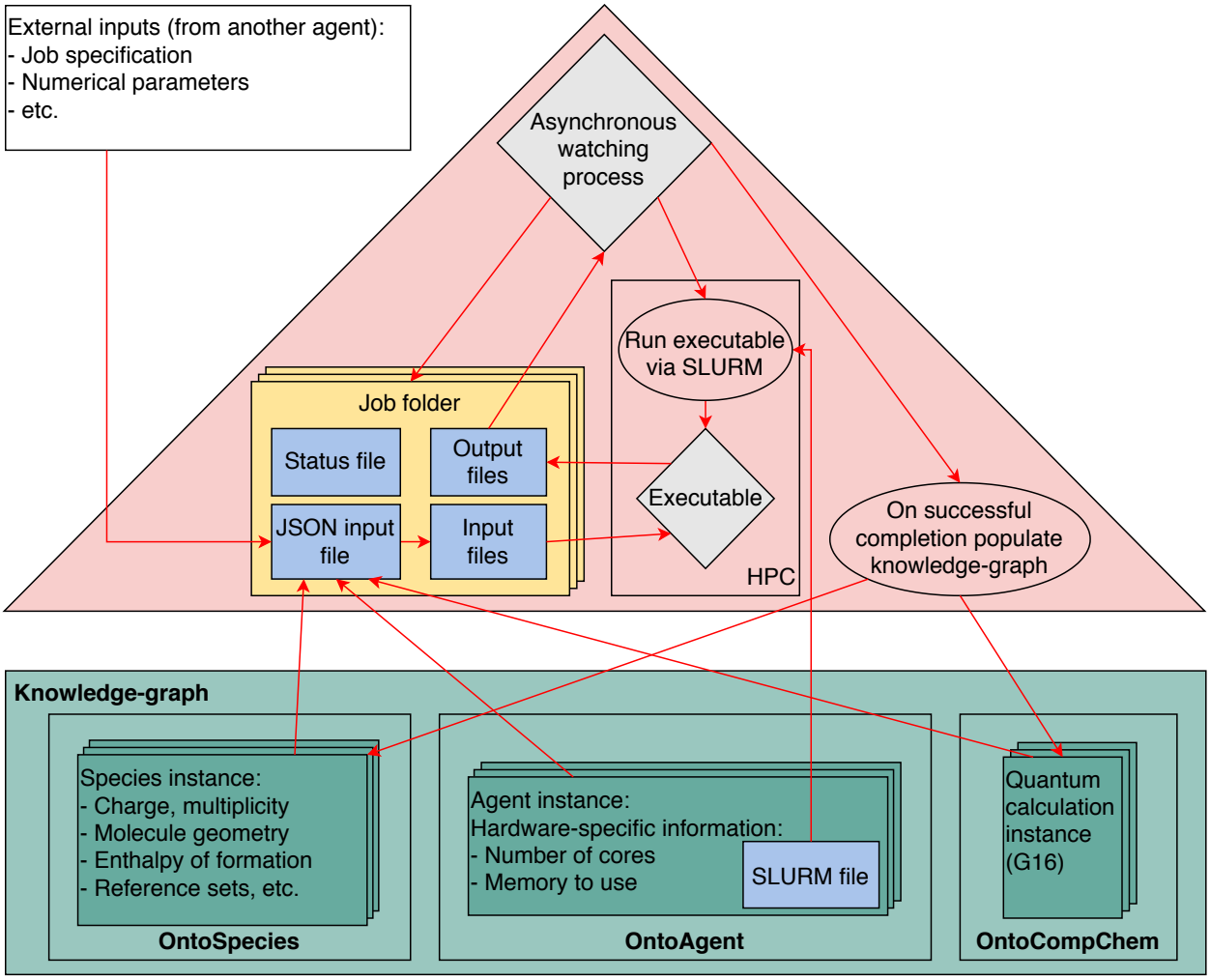


Figure 2: Elements of a generic agent (red triangle) and how they interact with the knowledge-graph (green box). An asynchronous watcher (grey diamond) manages running an executable (grey diamond), with all associated input and output files (blue boxes).

unexpected interruptions such as power-cuts or reboots by a system administrator. Once the hardware and the agent are restarted, the latter will continue its operations unaffected.

When the agent receives a job request, inputs to the agent are passed to it in Semantic Web standard JavaScript Object Notation (JSON) format. These inputs can include 'direct' inputs (from other agents) such as parameters specifying the task to be executed, but more typically will take the form of IRIs, pointing to relevant entities that are represented in the knowledge-graph. Also retrieved from knowledge-graph are parameters that are specific to the HPC hardware on which the executable that constitutes the agent is to be executed,

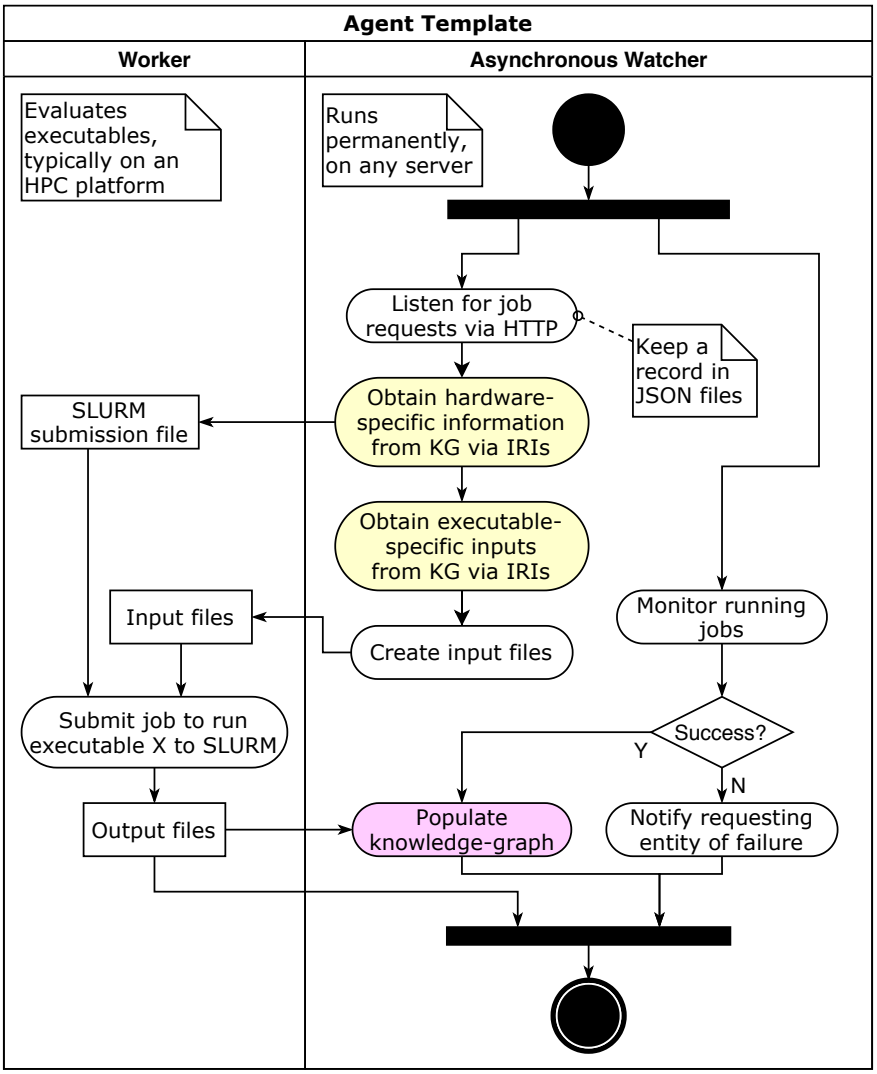


Figure 3: UML activity diagram of a generic agent which enables computational jobs (running of an executable "X") to be executed asynchronously on an HPC system upon request (via HTTP), with inputs obtained from the knowledge-graph (shaded in yellow) and outputs written back into the knowledge-graph (shaded in magenta).

such as the number of CPU-cores and amount of memory to be used, as well as a SLURM file which is used for submission to the resource manager.

As an aside, we note that this design implies that only a single instance of this agent needs to be deployed on a particular piece of hardware, and also in general, if the volume of incoming requests is sufficiently small.

Figure 3 shows a Unified Modelling Language (UML) activity diagram, illustrating the step-by-step activities of the components. Handling job requests and monitoring running

jobs are conducted asynchronously in parallel, as any events can happen at any time, in any order. Most agents will retrieve inputs from the knowledge-graph (shaded in yellow) and write outputs back to the knowledge-graph (shaded in magenta). The knowledge-graph itself is held in triple-stores such as RDF4J, Fuseki, and Blazegraph, which expose SPARQL endpoints. Reading and writing of subject-predicate-object triples is implemented by means of SPARQL queries and updates, respectively, through these endpoints. Which inputs and outputs exactly are exchanged with the knowledge-graph is of course not generic, but depends on the executable in question. Another critical activity relates to job monitoring: In practice, there are many things that can go wrong in the submission and execution of a computational job. It is therefore important to have robust procedures in place to deal with any unexpected failure in a controlled fashion, which in particular involves escalating useful error messages back to the calling entity, and ultimately a human, if the cause of the problem cannot be rectified by means of an algorithm.

Error-cancelling balanced reaction agent

The purpose of this agent is two-fold. Given a set of chemical species, it can produce estimates of standard enthalpy of formation, and in addition, within the given set, identify individual species as either consistent or inconsistent. The agent operates in two steps. The first step computes the standard enthalpy of formation of a species by using error-cancelling balanced reactions (EBRs).[48] At the centre of this method lies Hess' law, which states that the total enthalpy change of a reaction, ΔH_r° is equal to the sum of all individual enthalpy changes, independent of the reaction pathway:

$$\Delta H_r^\circ = \sum_{s \in S^\mathrm{P}} \nu(s) \Delta H_\mathrm{f}^\circ(s) - \sum_{s \in S^\mathrm{R}} \nu(s) \Delta H_\mathrm{f}^\circ(s) \tag{1}$$

Here, S^P and S^R denote the sets of products and reactants, $\nu(s)$ the stoichiometric coefficients, and $\Delta H_\mathrm{f}^\circ(s)$ the standard enthalpy of formation of species s. The reaction enthalpy

can be estimated from the results of quantum chemistry calculations via

$$\Delta H_r^\circ = \sum_{s \in S^{\mathrm{P}}} \nu(s) E^\circ(s) - \sum_{s \in S^{\mathrm{R}}} \nu(s) E^\circ(s), \tag{2}$$

where $E^\circ(s)$ is the zero-point corrected ground state energy given by quantum chemical calculations. If the enthalpy of reaction is computed as above, then the standard enthalpy of formation of a given target species, s_{T}, assumed here to be a reactant, can be estimated by rearranging Hess' law:

$$\Delta H_{\mathrm{f}}^\circ(s_{\mathrm{T}}) = \frac{1}{\nu(s_{\mathrm{T}})} \sum_{s \in S^{\mathrm{P}}} \nu(s) \Delta H_{\mathrm{f}}^\circ(s) - \frac{1}{\nu(s_{\mathrm{T}})} \Big(\Delta H_r^\circ + \sum_{s \in S^{\mathrm{R}} \setminus \{s_{\mathrm{T}}\}} \nu(s) \Delta H_{\mathrm{f}}^\circ(s) \Big) \tag{3}$$

However, it is known that estimating enthalpies using quantum chemical calculations results in systematic errors.[49] To circumvent this, error-cancelling balanced reactions (EBRs), are employed, which make use of structural and electronic similarities between the species in a reaction to allow cancellation of the systematic errors. Examples of types of EBRs include isogyric reactions, which conserve the number of spin states during the reaction, and isodesmic reactions, which conserves the number of each type of bond during the reaction. Using EBRs enables improved estimates of standard enthalpies of formation of species from quantum chemical calculations.

The second step performed by the agent is a heuristic cross-validation. The algorithm is described in detail elsewhere,[48] and so only a brief summary is provided here. A set of species is given to the agent, each with a reference value for the standard enthalpy of formation. These reference values may be derived experimentally or from high-level computational methods. Next, one species in the set is selected, in a leave-one-out cross-validation method. For the selected species, a user-defined number of EBRs are generated. Each EBR can then be used to estimate the standard enthalpy of formation of the species by using Eqn. (3), an estimate of ΔH_r° derived from quantum calculations, and the reference enthalpies for all other species in the EBR. For each EBR, the estimated standard enthalpy of formation

generated for the species is then compared to the reference value, and an error is computed:

$$\epsilon_r(r, s_{\mathrm{T}}) = \left|\Delta H^{\circ}_{\mathrm{f,reference}}(s_{\mathrm{T}}) - \Delta H^{\circ}_{\mathrm{f}}(r, s_{\mathrm{T}})\right| \tag{4}$$

The EBR is then accepted if this error is below a defined upper limit, $\epsilon_r^{\max}$, and rejected otherwise. If all EBRs for a given species are rejected, the species is flagged as potentially inconsistent. This process is repeated for all of the species in the reference set to sort them as consistent or potentially inconsistent.

The set of consistent and potentially inconsistent species is then refined by selecting the potentially inconsistent species with the highest average error and generating a new set of EBRs for it using only species that were deemed consistent in the initial sorting. The errors are then recomputed for this new set of EBRs. If the average error for the new set of EBRs, $\bar{\epsilon}_r^{\mathrm{new}}$, is lower than the error for the initial set of EBRs, $\bar{\epsilon}_r^{\mathrm{initial}}$, then it is assumed that the original inconsistency for this species is due to another species that appeared in the original EBRs, and the selected species is added to the consistent set. The process is repeated for each potentially inconsistent species in descending order of average error until the set of inconsistent species does not change. At this point, convergence is reached, and the set of inconsistent species is reported as needing an improved estimate of the standard enthalpy of formation. It is noted that, depending on the size of the species set and other parameters, executing the algorithm can be computationally expensive.

The design and implementation of this EBR agent follows that of the generic template illustrated in Figs. 2 and 3. Jobs are requested via HTTP. The JSON-formatted inputs, that are passed as arguments as part of the job request, include a list of pairs of IRIs. Each pair consists of an IRI to a unique species instance in OntoSpecies, and an IRI to a quantum calculation in OntoCompChem. This list defines the species set that the EBR algorithm is applied to. All associated information, like reference values for the standard enthalpies of formation for each species, can be retrieved from the knowledge-graph using the provided

IRIs. Other JSON inputs include for example the type of EBR to be used (isogyric, isodesmic, *etc.*), and algorithm termination criteria (maximum numbers of iterations of various loops). Once all required information has been assembled, input files for the core EBR code are generated and the job is submitted to the resource manager on an HPC platform. In case of successful termination, the resulting new estimates of standard enthalpies of formation as well as the finding of whether or not a species is being deemed consistent or not are stored in the knowledge-graph (as part of the relevant instances in OntoSpecies).

Quantum calculation agent

The purpose of this agent is to conduct a quantum chemistry calculation using the Gaussian[50] software. Quantum chemical calculations are used to derive the molecular properties of a chemical system from first principles by solving the time-independent Schrödinger Equation. Solving this equation for a molecule, *i.e.* for a system consisting of several electrons and nuclei, yields the wavefunction of the system, which provides information about the quantum state of the system, such as the positions of nuclei and electrons and the energy associated with their particular configuration. The Schrödinger Equation can be written as

$$\hat{H}\Psi(\vec{r},\vec{R}) = E\Psi(\vec{r},\vec{R}), \tag{5}$$

where $\hat{H}$ is the Hamiltonian operator, Ψ is the wavefunction of the system, $\vec{r}$ and $\vec{R}$ are the positions of electrons and nuclei, and E is the eigenvalue, representing the total energy of the system. The Hamiltonian operator can be written as

$$\hat{H} = \hat{T}_{\text{n}} + \hat{T}_{\text{e}} + \hat{V}_{\text{n-n}} + \hat{V}_{\text{n-e}} + \hat{V}_{\text{e-e}}, \tag{6}$$

where, $\hat{T}_{\text{n}}$ and $\hat{T}_{\text{e}}$ operators represent the kinetic energy of the nuclei and electrons respectively, and $\hat{V}_{\text{n-n}}$, $\hat{V}_{\text{n-e}}$ and $\hat{V}_{\text{e-e}}$ represent the potential energy of the nucleus-nucleus, nucleus-electron, and electron-electron interactions, respectively.

Typically, the Born-Oppenheimer approximation is invoked, which neglects the coupling between the electrons and nuclei of a system. This results in the electronic Schrödinger equation, which is what modern computational chemistry packages like Gaussian solve to derive molecular properties:

$$\hat{H}_{\mathrm{e}}\Psi(\vec{r},\vec{R})_{\mathrm{e}} = E_{\mathrm{e}}\Psi(\vec{r},\vec{R})_{\mathrm{e}} \tag{7}$$

In this notation, $\Psi(\vec{r},\vec{R})_{\mathrm{e}}$ is the electronic wavefunction, E_{e} are the eigenvalues representing the electronic energies of the system, and $\hat{H}_{\mathrm{e}}$ is the electronic Hamiltonian:

$$\hat{H}_{\mathrm{e}} = \hat{T}_{\mathrm{e}} + \hat{V}_{\mathrm{n\text{-}e}} + \hat{V}_{\mathrm{e\text{-}e}} \tag{8}$$

For chemical systems larger than simple hydrogen-like atoms, the Schrödinger equation can only be solved numerically. Various methods to do so are implemented in modern computational chemistry software such as Gaussian. One very popular method is Density Functional Theory (DFT), which derives the properties of a chemical system by manipulating the electronic Schrödinger equation and solving for the energy as a function of the electron density $\rho(\vec{r})$ of the system.[51] This results in the equation[52]

$$E[\rho(\vec{r})] = T_{\mathrm{s}}[\rho(\vec{r})] + V_{\mathrm{n\text{-}e}}[\rho(\vec{r})] + J[\rho(\vec{r})] + E_{\mathrm{xc}}[\rho(\vec{r})], \tag{9}$$

where $T_{\mathrm{s}}[\rho(\vec{r})]$ is the kinetic energy of the non-interacting model system and $J[\rho(\vec{r})]$ is the Coulomb energy functional, both of which are known exactly. Similarly, $V_{\mathrm{n\text{-}e}}[\rho(\vec{r})]$ is the nucleus-electron attraction potential functional whose form can be derived given the Born-Oppenheimer approximation. The final term, E_{xc} is termed the exchange-correlation functional, and accounts for the difference in kinetic and potential energies between the real interacting-electron system, and the approximate non-interacting electron system.

DFT is a very popular approach, as the electron density is only a function of the spatial coordinates, making it a computationally efficient way to derive molecular properties through

quantum chemistry, and is hence adopted in this work. Nevertheless, such methods still require substantial computational resources, and do typically need to be run on HPC systems.

The quantum calculation agent automates the solutions of the above equations via Gaussian. Similar to the enthalpy of formation agent, the implementation of this agent follows exactly the design outlined in Figs. 2 and 3, *i.e.* calculations can be requested via the internet and are conducted on an HPC system, with key inputs and outputs retrieved from and written back into the knowledge-graph. JSON inputs passed to this agent include in particular an IRI of a unique species instance in OntoSpecies. Species definition information required for a quantum calculation such as charge, spin multiplicity, and geometry of the molecule are retrieved from the knowledge-graph by means of SPARQL queries. Having obtained all necessary inputs from the knowledge-graph, a plain-text Gaussian input file is then populated. Job submission and monitoring are managed generically by an asynchronous watching process. In case of successful completion, the result of the quantum calculation, namely the log file output by Gaussian, is converted to OWL and a new instance of the G16 class, as defined in the OntoCompChem ontology, is created in the knowledge-graph. That is, in this case the newly created OWL file, which is essentially a collection of subject-predicate-object triples itself, can simply be uploaded to a triple-store.

Thermodata agent

The purpose of this agent is to calculate three thermodynamic quantities for a chosen chemical species as functions of temperature T, namely the heat capacity at constant pressure C_p, enthalpy H, and entropy S.[28] These thermodynamic quantities are derived by means of the molecular partition function, $q = q_T q_V q_R q_E$, whose components consist of the translational (q_T), vibrational (q_V), rotational (q_R), and electronic (q_E) partition functions derived from standard statistical mechanics expressions and the rigid-rotor-harmonic-oscillator (RRHO)

approximation:[53]

$$q_T = \left(\frac{mk_\mathrm{B}T}{2\pi\hbar^2}\right)^{\frac{3}{2}} V \tag{10}$$

$$q_V = \prod_{i=1}^{N_V} \frac{\exp\left(-\frac{h\nu_i}{2k_\mathrm{B}T}\right)}{1-\exp\left(-\frac{h\nu_i}{k_\mathrm{B}T}\right)} \tag{11}$$

$$q_R = \frac{(8\pi^3 I_x I_y I_z)^{1/2}(k_\mathrm{B}T)^{3/2}}{\sigma\pi\hbar^3} \tag{12}$$

$$q_E \approx g_0^E \tag{13}$$

Here, h denotes Planck's constant, k_B is the Boltzmann constant, m is the mass, V is the volume, ν_i is the magnitude of the i^{th} frequency, I_k is the moment of inertia about the k^{th} axis, σ is the symmetry number, and g_0^E is the ground state electronic degeneracy. We note at this point that the RRHO approximation can result in errors for species with internal rotors[54] that can be comparable in magnitude with the systematic errors in some DFT methods. However, more advanced hindered rotor treatments can become computationally expensive, so for simplicity such treatments are not applied in the present work but will be considered in the future. The necessary frequencies, rotations, and ground state energies to compute the partition functions are obtained from the results of quantum chemistry calculations. Once the molecular partition function is constructed, the molar heat capacity, entropy, and enthalpy can be derived from the following expressions:

$$C_\mathrm{p} = C_\mathrm{v} + N_\mathrm{A}k_\mathrm{B} \qquad \text{with} \qquad C_\mathrm{v} = N_\mathrm{A}k_\mathrm{B}T\frac{\partial^2(T\ln q)}{\partial T^2} \tag{14}$$

$$S = N_\mathrm{A}k_\mathrm{B}\left[\frac{\partial(T\ln q)}{\partial T} - \ln N_\mathrm{A} + 1\right] \tag{15}$$

$$\Delta H = \int_0^T C_\mathrm{p}\mathrm{d}T = \frac{N_\mathrm{A}k_\mathrm{B}T^2}{q}\frac{\partial q}{\partial T} + N_\mathrm{A}k_\mathrm{B}T \tag{16}$$

Here, N_A denotes Avogadro's number. Note that Eqn. (16) provides instead of an absolute value only an enthalpy difference. In order to obtain meaningful absolute values, it is therefore necessary to provide a reference value for the enthalpy of formation at a known reference

temperature, usually 298.15 K. In practice, the functional dependence of the three thermodynamic quantities on temperature is captured by fitting the widely-used NASA polynomials to the thermochemical data extracted from the partition functions, with seven coefficients $c_1, \ldots, c_7$ for each polynomial, for two contiguous temperature ranges:

$$\frac{C_\mathrm{p}}{R} = c_1 + c_2 T + c_3 T^2 + c_4 T^3 + c_5 T^4 \tag{17}$$

$$\frac{H}{R} = c_1 T + \frac{c_2}{2} T^2 + \frac{c_3}{3} T^3 + \frac{c_4}{4} T^4 + \frac{c_5}{5} T^5 + c_6 \tag{18}$$

$$\frac{S}{R} = c_1 \ln T + c_2 T + \frac{c_3}{2} T^2 + \frac{c_4}{3} T^3 + \frac{c_5}{4} T^4 + c_7 \tag{19}$$

The calculations involved for the thermodata agent are computationally very light and thus do not require submission to an HPC system. Instead, they can be conducted directly by whichever webservice responds to the job request. Therefore, this agent can follow a simpler design than the one outlined in Figs. 2 and 3. When it receives a job request, the JSON arguments need to contain an IRI of a quantum calculation in the knowledge-graph from which thermodata is to be derived, as well as an IRI to a unique species instance in order to retrieve a reference value for the enthalpy of formation.

Integration

Figure 4 sketches in a UML sequence diagram how the three agents described above are being integrated, *i.e.* how they communicate with each other and how they interact with the knowledge-graph. The basic idea is a three-step process: Firstly, the EBR agent commences the process by retrieving previously obtained quantum calculation results and enthalpies of formation for a given set of species from the knowledge-graph via IRIs, and then detecting inconsistent species amongst the given set using cross-validation. Secondly, for each of those species that have been identified as inconsistent, the EBR agent requests quantum calculations from the quantum calculation agent at a higher level of theory than currently available in the knowledge-graph. Lastly, for any quantum calculation job that has finished

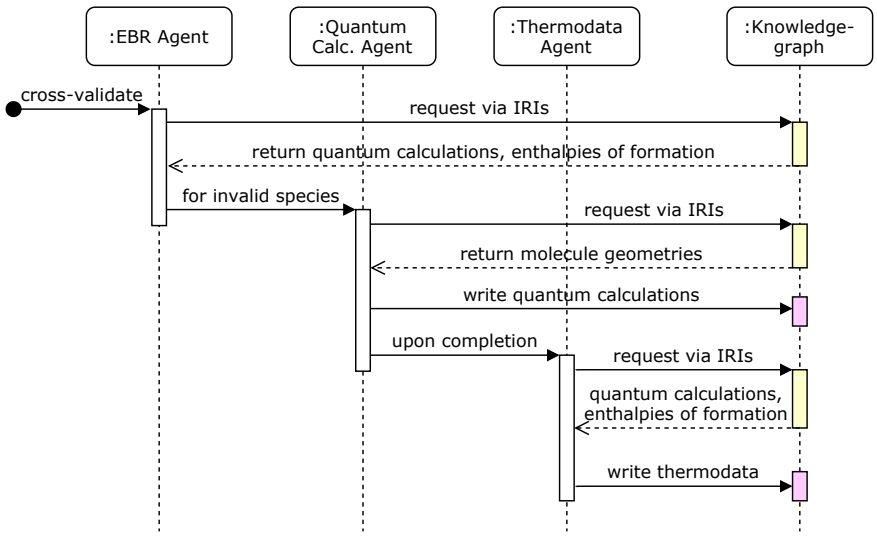


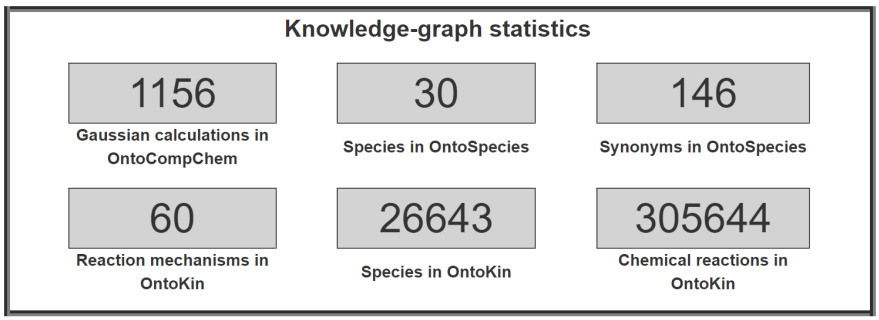
Figure 4: UML sequence diagram illustrating the interaction of the agents and how they act on the knowledge-graph (inputs to agents from knowledge-graph shaded in yellow, outputs back into the knowledge-graph shaded in magenta).

successfully, the thermodata agent derives the corresponding thermodata. This involves retrieving a consistent estimate of the standard enthalpy of formation from the knowledge-graph, as populated by the EBR agent. As the final step, the thermodata agent then updates a chemical mechanism if it contains a species that is linked to the unique species whose thermodata was just updated.

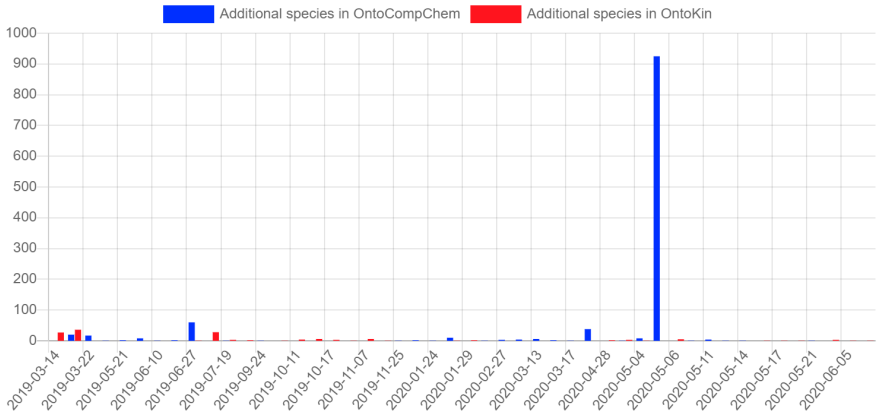
As mentioned above, we note that this sequence of three agents reading from and writing to the knowledge-graph is made possible by and relies upon IRI links being present in the knowledge-graph between quantum calculations (OntoCompChem), unique species (OntoSpecies), and species as they are part of chemical mechanisms (OntoKin).

Monitoring agent

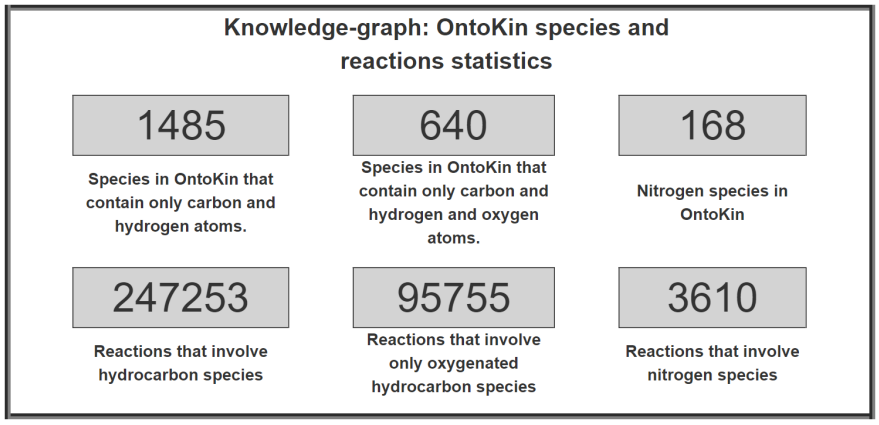
In order to monitor the health status of the knowledge-graph, and to provide a high-level overview of how it evolves over time, we have created a web-page displaying various metrics.[55] Figure 5 displays screen-shots of the relevant parts of that web-page. The page mainly features counters of instances of selected concepts in the OntoKin (mechanisms, species, and reactions), OntoSpecies (unique species), and OntoCompChem (quantum cal-



(a) Counters of selected concepts, including quantum calculations, unique species, mechanisms, species, and reactions.



(b) Time-history chart showing numbers of species added on particular dates.



(c) Counters of selected types of species and reactions within chemical mechanisms (as represented in OntoKin).

Figure 5: Screen-shots of a web-page showing knowledge-graph statistics.

culations) ontologies, as shown in Fig. 5(a), that can be useful for monitoring and diagnostics. A time-history chart of additional species within mechanisms (OntoKin) and as quantum calculations (OntoCompChem), showing how many instances were added on a particular

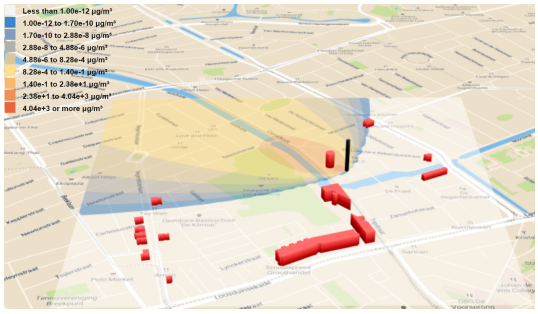
date, is also included (Fig. 5(b)). More specialised statistics on certain types of species and reactions within mechanisms in OntoKin are shown in Fig. 5(c).

All the data displayed on this web-page is ultimately obtained through SPARQL queries from various triple-stores. In this context, it should be noted that subject-predicate-object triple representations of complex data structures such as chemical mechanisms involve considerable numbers of triples: For example, the sixty mechanisms we uploaded for demonstration purposes require several tens of millions of triples. The complexity of a query and the size of the repository both impact the response times. For this reason, in order to reduce loading times, most of the numbers shown on the page are cached. The cache is being refreshed once every 24 hours by this agent.

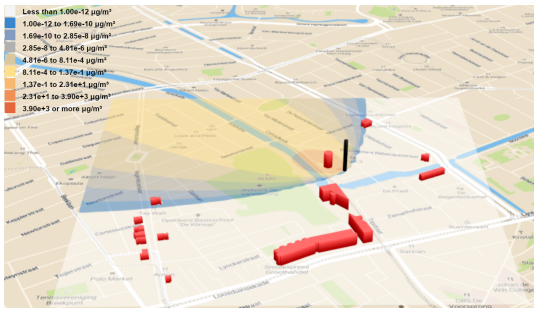
Use-case: Atmospheric pollutant dispersion

As a use-case, we integrate the agents discussed in the previous sections into a cross-domain simulation of the atmospheric dispersion of pollutant emissions from a power plant and display this interactively on a web-page.[56] We consider a power plant that is fictitious, but inspired by the *Energiecentrale* in The Hague (see Eibeck et al.[30] for more details). Three-dimensional representations of selected buildings, which are stored in the knowledge-graph using the OntoCityGML ontology, are overlaid on an OpenStreetMap. The atmospheric dispersion of pollutants is simulated using the Atmospheric Dispersion Modelling System (ADMS),[57] which is based on a fluid-dynamic model that includes Gaussian plume air dispersion. ADMS requires as one of its inputs weather data, which are retrieved in real time from the world wide web and stored in the knowledge-graph by a dedicated weather agent. All relevant data are held in the knowledge-graph and as such are linked through IRIs.

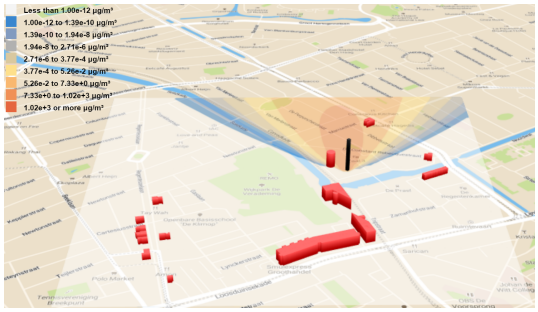
Combustion in the stationary power generator is simulated by means of a Stochastic Reactor Model (SRM)[58] agent. The SRM has been developed primarily as an internal combustion engine (ICE) simulation tool and has been applied as such in numerous studies



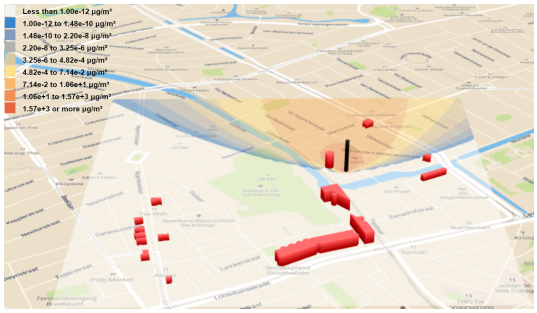
(a) CO emissions predicted using the original mechanism.



(b) CO emissions predicted using a mechanism with updated thermodata.



(c) uHC emissions predicted using the original mechanism.



(d) uHC emissions predicted using a mechanism with updated thermodata.

Figure 6: Screen-shots of a web-interface showing pollutant concentration distributions in the atmosphere over The Hague as predicted by ADMS, with 3D-representation of selected buildings (map data © OpenStreetMap contributors).

(see for example,[59] and references therein). It uses detailed chemical kinetics and is thus able to predict gaseous and particulate pollutants. We consider a Primary Reference Fuel (PRF) as a surrogate fuel and employ as a mechanism a reduced PRF scheme that was specifically designed for ICE simulations.[60] The mechanism is represented in the knowledge-graph using OntoKin, is retrieved as an OWL file via IRI, and then converted into an SRM-readable format.

The process begins with the EBR agent cross-validating a set of 34 hydrocarbon species and concluding that, based on the given data using isodesmic reactions, 14 of the species are inconsistent. The EBR agent then launches quantum calculations for each of these species at a higher level of theory. As our main focus here is to demonstrate the principle, we restrict ourselves to the relatively basic B97-1/6-311+G(d,p) and M06-2X/6-311+G(d,p) levels of

theory. Once a job finishes, the thermodata agent is launched in order to derive NASA polynomials that are suitable for use in a chemical mechanism. One species in particular whose thermodata is being updated in this way happens to be CO_2, which is part of the mechanism used in this study. As soon as the thermodata agent has deduced new thermodata for a species, the mechanism is updated. This then causes changes in the predicted atmospheric emission distributions.

Figure 6 shows concentration level contours overlaid on a map of The Hague, with some buildings. Distributions of carbon monoxide (CO, Figs. 6(a) and (b)) and unburnt hydrocarbons (uHC, Figs. 6(c) and (d)) are shown, each as predicted using the original mechanism by Wang et al.[60] and as predicted by the same mechanism with updated thermodata. It can be observed that the shape of the contours changes between the images for the original and the modified mechanisms. In addition, it should be noted in the legend that the range of concentration values corresponding to the colours of the contours also differs between the original and modified mechanisms. Taking this into account, the maximum predicted concentration level of CO decreases by about 3% from Fig. 6(a) to (b). Similarly, the maximum predicted concentration level of unburnt hydrocarbons changes by about 50% from Fig. 6(c) to 6(d).

We emphasise that the set of 34 species, with CO_2 being among the species that are identified as inconsistent, has been specifically chosen to demonstrate the principle. The fact that CO_2 is being highlighted does, however, not necessarily imply that the thermodata of CO_2 in the used mechanism[60] are deficient in any way. In fact, those thermodata are close to the currently most accurate values. The EBR algorithm by nature establishes consistency or inconsistency of thermodata of a set of species *with respect to each other*, and thus the notion of consistency of a species depends on the other species in the set, as well as on the EBRs considered. In addition, given the rather basic levels of theory we use for the quantum calculations in this work, one cannot reasonably expect an improvement upon highly accurate values. And further, even if there were an improvement in the thermodata of a particular

species, one could not conclude that a prediction of the concentration of another species within a mechanism would necessarily improve, because again the consistency of (not only thermodynamic, but also kinetic and transport) parameters of different species with respect to each other plays a key role. Whilst the question of what constitutes an improvement is beyond the scope of this work, we believe the linked nature of information in a knowledge-graph approach offers the potential to address such challenges in the future.

Conclusions

We have developed agents that determine the reliability of the thermodata of sets of species based on error-cancelling balanced reactions, that conduct quantum calculations upon request via HTTP, and that deduce thermodynamic data from quantum calculations. These agents seamlessly fit into and extend the existing general-purpose knowledge-graph of the JPS. We have integrated these agents so that species whose thermodata have been identified as inconsistent automatically will be calculated at higher level of theory. Thermodata is automatically deduced from new quantum calculations, and propagated into an updated mechanism. As a proof of concept, and to demonstrate interoperability between heterogenous data formats and software in a wider context, we have integrated this further into a cross-domain application that considers atmospheric pollutant dispersion simulations which utilise 3D geometries of buildings, and live weather data retrieved by agents from the world wide web. This use-case involves simulations across the length scales – from quantum calculations to macroscopic fluid flow. As the knowledge-graph grows, and the quality of the thermodynamic data within it improves, the mechanism used for the combustion simulation automatically updates. We have demonstrated in this multi-scale example how a new quantum calculation for a species can affect the distribution of pollutants in the atmosphere.

Acknowledgements

This work was partly funded by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Markus Kraft gratefully acknowledges the support of the Alexander von Humboldt foundation. The authors are grateful to EPSRC (grant number: EP/R029369/1) and ARCHER for financial and computational support as a part of their funding to the UK Consortium on Turbulent Reacting Flows (www.ukctrf.com).

References

(1) Lasi, H.; Fettke, P.; Kemper, H. G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242.

(2) Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Comput. Netw.* **2010**, *54*, 2787–2805.

(3) Goodman, J. Computer Software Review: Reaxys. *J. Chem. Inf. Model.* **2009**, *49*, 2897–2898.

(4) Kim, S.; Thiessen, P.; Bolton, E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B.; Wang, J.; Yu, B.; Zhang, J.; Bryant, S. PubChem Substance and Compound Databases. *Nucleic Acids Res.* **2016**, *44*, D1202–D1213.

(5) American Chemical Society, CAS Registry. 2019; `https://www.cas.org/support/documentation/chemical-substances`, Accessed 27 May 2020.

(6) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J. L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864–2875.

(7) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comp. Sci.* **1988**, *28*, 31–36.

(8) Frenklach, M. Transforming Data into Knowledge – Process Informatics for Combustion Chemistry. *Proc. Combust. Inst.* **2007**, *31*, 125–140.

(9) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Sci. Data* **2014**, *1*, 140022.

(10) Johnson III, R. D. NIST Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database Number 101, Release 19. 2018; doi:10.18434/T47C7Z.

(11) Ruscic, B.; Pinzon, R. E.; Morton, M. L.; von Laszevski, G.; Bittner, S. J.; Nijsure, S. G.; Amin, K. A.; Minkoff, M.; Wagner, A. F. Introduction to Active Thermochemical Tables: Several "Key" Enthalpies of Formation Revisited. *J. Phys. Chem. A* **2004**, *108*, 9979–9997.

(12) Goldsmith, C. F.; Magoon, G. R.; Green, W. H. Database of Small Molecule Thermochemistry for Combustion. *J. Phys. Chem. A* **2012**, *116*, 9033–9057.

(13) Lambert, V. R.; West, R. H. Identification, Correction, and Comparison of Detailed Kinetic Models. 9th US National Combustion Meeting. 2015.

(14) Gao, C. W.; Allen, J. W.; Green, W. H.; West, R. H. Reaction Mechanism Generator: Automatic Construction of Chemical Kinetic Mechanisms. *Comput. Phys. Commun.* **2016**, *203*, 212–225.

(15) Keçeli, M.; Elliott, S. N.; Li, Y.-P.; Johnson, M. S.; Cavallotti, C.; Georgievskii, Y.; Green, W. H.; Pelucchi, M.; Wozniak, J. M.; Jasper, A. W.; Klippenstein, S. J. Automated Computational Thermochemistry for Butane Oxidation: A Prelude to Predictive Automated Combustion Kinetics. *Proc. Combust. Inst.* **2019**, *37*, 363–371.

(16) Li, Y. P.; Han, K.; Grambow, C. A.; Green, W. H. Self-Evolving Machine: A Continuously Improving Model for Molecular Thermochemistry. *J. Phys. Chem. A* **2019**, *123*, 2142–2152.

(17) Berners-Lee, T. Linked Data – Design Issues. 2006; `https://www.w3.org/DesignIssues/LinkedData.html`, Accessed 10 June 2020.

(18) Bizer, C.; Heath, T.; Berners-Lee, T. Linked Data – The Story So Far. *Int. J. Semant. Web Inf.* **2009**, *5*, 1–22.

(19) Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 34–43.

(20) Noy, N. F.; Gao, Y.; Jain, A.; Narayanan, A.; Patterson, A.; Taylor, J. Industry-Scale Knowledge Graphs: Lessons and Challenges. *Commun. ACM* **2019**, *62*, 36–43.

(21) Fensel, D.; Şimşek, U.; Angele, K.; Huaman, E.; Kärle, E.; Panasiuk, O.; Toma, I.; Umbrich, J.; Wahler, A. *Knowledge Graphs – Methodology, Tools and Selected Use Cases*; Springer Nature, Switzerland, 2020.

(22) Taylor, K. R.; Gledhill, R. J.; Essex, J. W.; Frey, J. G.; Harris, S. W.; De Roure, D. C. Bringing Chemical Data onto the Semantic Web. *J. Chem. Inf. Model.* **2006**, *46*, 939–952.

(23) Fu, G.; Batchelor, C.; Dumontier, M.; Hastings, J.; Willighagen, E.; Bolton, E. PubChemRDF: Towards the Semantic Annotation of PubChem Compound and Substance Databases. *J. Cheminf.* **2015**, *7*, 34.

(24) Menon, A.; Krdzavac, N.; Kraft, M. From Database to Knowledge Graph – Using Data in Chemistry. *Curr. Opin. Chem. Eng.* **2019**, *26*, 33–37.

(25) Phadungsukanan, W.; Kraft, M.; Townsend, J. A.; Murray-Rust, P. The Semantics of Chemical Markup Language (CML) for Computational Chemistry : CompChem. *J. Cheminf.* **2012**, *4*, 1–16.

(26) Krdzavac, N.; Mosbach, S.; Nurkowski, D.; Buerger, P.; Akroyd, J.; Martin, J.; Menon, A.; Kraft, M. An Ontology and Semantic Web Service for Quantum Chemistry Calculations. *J. Chem. Inf. Model.* **2019**, *59*, 3154–3165.

(27) Farazi, F.; Akroyd, J.; Mosbach, S.; Buerger, P.; Nurkowski, D.; Salamanca, M.; Kraft, M. OntoKin: An Ontology for Chemical Kinetic Reaction Mechanisms. *J. Chem. Inf. Model.* **2020**, *60*, 108–120.

(28) Farazi, F.; Krdzavac, N.; Akroyd, J.; Mosbach, S.; Menon, A.; Nurkowski, D.; Kraft, M. Linking Reaction Mechanisms and Quantum Chemistry: An Ontological Approach. *Comput. Chem. Eng.* **2020**, *137*, 106813.

(29) Cambridge CARES, J-Park Simulator. 2020; `http://theworldavatar.com/`, Accessed 31 August 2020.

(30) Eibeck, A.; Lim, M. Q.; Kraft, M. J-Park Simulator: An Ontology-Based Platform for Cross-Domain Scenarios in Process Industry. *Comput. Chem. Eng.* **2019**, *131*, 106586.

(31) Marquardt, W.; Morbach, J.; Wiesner, A.; Yang, A. *OntoCAPE – A Re-Usable Ontology for Chemical Process Engineering*, 1st ed.; Springer-Verlag Berlin Heidelberg, 2010.

(32) Zhang, C.; Romagnoli, A.; Zhou, L.; Kraft, M. Knowledge Management of Eco-Industrial Park for Efficient Energy Utilization Through Ontology-Based Approach. *Appl. Energy* **2017**, *204*, 1412–1421.

(33) Zhou, L.; Pan, M.; Sikorski, J. J.; Garud, S.; Aditya, L. K.; Kleinelanghorst, M. J.; Karimi, I. A.; Kraft, M. Towards an Ontological Infrastructure for Chemical Process Simulation and Optimization in the Context of Eco-Industrial Parks. *Appl. Energy* **2017**, *204*, 1284–1298.

(34) Zhou, L.; Zhang, C.; Karimi, I. A.; Kraft, M. An Ontology Framework Towards Decentralized Information Management for Eco-Industrial Parks. *Comput. Chem. Eng.* **2018**, *118*, 49–63.

(35) Devanand, A.; Karmakar, G.; Krdzavac, N.; Rigo-Mariani, R.; Foo, E. Y. S.; Karimi, I. A.; Kraft, M. OntoPowSys: A Power System Ontology for Cross Domain Interactions in an Eco Industrial Park. *Energ. AI* **2020**, *1*, 100008.

(36) Insight Centre for Data Analytics, The Linked Open Data Cloud. 2020; `https://lod-cloud.net/`, Accessed 31 August 2020.

(37) DBpedia, DBpedia. 2020; `https://wiki.dbpedia.org/`, Accessed 31 August 2020.

(38) Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; Bizer, C. DBpedia – A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semant. Web* **2015**, *6*, 167–195.

(39) Zhou, X.; Eibeck, A.; Lim, M. Q.; Krdzavac, N.; Kraft, M. An Agent Composition Framework for the J-Park Simulator – a Knowledge Graph for the Process Industry. *Comput. Chem. Eng.* **2019**, *130*, 106577.

(40) Zhou, X.; Lim, M. Q.; Kraft, M. A Smart Contract-Based Agent Marketplace for the J-Park Simulator – a Knowledge Graph for the Process Industry. *Comput. Chem. Eng.* **2020**, *139*, 106896.

(41) Pan, M.; Sikorski, J.; Kastner, C. A.; Akroyd, J.; Mosbach, S.; Lau, R.; Kraft, M. Applying Industry 4.0 to the Jurong Island Eco-Industrial Park. *Energy Procedia* **2015**, *75*, 1536–1541.

(42) Pan, M.; Sikorski, J.; Akroyd, J.; Mosbach, S.; Lau, R.; Kraft, M. Design Technologies

for Eco-Industrial Parks: From Unit Operations to Processes, Plants and Industrial Networks. *Appl. Energy* **2016**, *175*, 305–323.

(43) Devanand, A.; Karimi, I. A.; Kraft, M. Optimal Site Selection for Modular Nuclear Power Plants. *Comput. Chem. Eng.* **2019**, *125*, 339–350.

(44) Eibeck, A.; Chadzynski, A.; Lim, M. Q.; Aditya, L. K.; Ong, L.; Devanand, A.; Karmakar, G.; Mosbach, S.; Lau, R.; Karimi, I. A.; Foo, E. Y. S.; Kraft, M. A Parallel World Framework for Scenario Analysis in Knowledge Graphs. *Data-Centric Engineering* **2020**, *1*, e6.

(45) Cambridge CARES, OntoKin Web-Interface. 2020; `http://www.theworldavatar.com/ontokin/`, Accessed 31 August 2020.

(46) Farazi, F.; Salamanca, M.; Mosbach, S.; Akroyd, J.; Eibeck, A.; Aditya, L. K.; Chadzynski, A.; Pan, K.; Zhou, X.; Zhang, S.; Lim, M. Q.; Kraft, M. A Knowledge-Graph Approach to Combustion Chemistry and Interoperability. *ACS Omega* **2020**, *5*, 18342–18348.

(47) Cambridge CARES, OntoCompChem Web-Interface. 2020; `http://www.theworldavatar.com/molhub/`, Accessed 31 August 2020.

(48) Buerger, P.; Akroyd, J.; Mosbach, S.; Kraft, M. A Systematic Method to Estimate and Validate Enthalpies of Formation Using Error-Cancelling Balanced Reactions. *Combust. Flame* **2018**, *187*, 105–121.

(49) Weber, R.; Wilson, A. K. Do Composite Methods Achieve Their Target Accuracy? *Comput. Theor. Chem.* **2015**, *1072*, 58–62.

(50) Frisch, M. J. et al. Gaussian 16 Revision B.01. 2016; Gaussian Inc. Wallingford CT.

(51) Ziegler, T. Approximate Density Functional Theory as a Practical Tool in Molecular Energetics and Dynamics. *Chem. Rev.* **1991**, *91*, 651–667.

(52) Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136*, B864–B871.

(53) McQuarrie, D. A.; Simon, J. D. *Molecular Thermodynamics*; University Science Books, Sausalito CA, USA, 1999.

(54) Gao, Y.; He, T.; Li, X.; You, X. Effect of Hindered Internal Rotation Treatments on Predicting the Thermodynamic Properties of Alkanes. *Phys. Chem. Chem. Phys.* **2019**, *21*, 1928–1936.

(55) Cambridge CARES, Knowledge-Graph Statistics. 2020; `http://theworldavatar.com/graph/statisticsAction`, Accessed 31 August 2020.

(56) Cambridge CARES, Atmospheric Pollutant Dispersion. 2020; `http://www.theworldavatar.com/JPS/`, Accessed 31 August 2020.

(57) Cambridge Environmental Research Consultants (CERC), Atmospheric Dispersion Modelling System (ADMS). 2020; `https://cerc.co.uk/environmental-software/ADMS-model.html`, Accessed on 17 March 2020.

(58) CMCL Innovations, SRM Engine Suite, version 11.2.0. 2020; `https://cmclinnovations.com/solutions/products/srm/`, Accessed on 30 April 2020.

(59) Mosbach, S.; Celnik, M. S.; Raj, A.; Kraft, M.; Zhang, H. R.; Kubo, S.; Kim, K.-O. Towards a Detailed Soot Model for Internal Combustion Engines. *Combust. Flame* **2009**, *156*, 1156–1165.

(60) Wang, H.; Yao, M.; Reitz, R. D. Development of a Reduced Primary Reference Fuel (PRF) Mechanism for IC Engine Combustion Simulations. *Energy Fuels* **2013**, *27*, 7843–7853.

Graphical TOC Entry