Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

J-Park Simulator: An ontology-based platform for cross-domain scenarios in process industry

Andreas Eibeck [1], Mei Qi Lim [1], Markus Kraft [1,2,3]

released: 04 February 2019

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

[2] University of Cambridge,
Department of Chemical Engineering and Biotechnology,
New Museums Site,
Pembroke Street,
Cambridge, CB2 3RA
United Kingdom.

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

Preprint No. 222

UNIVERSITY OF CAMBRIDGE

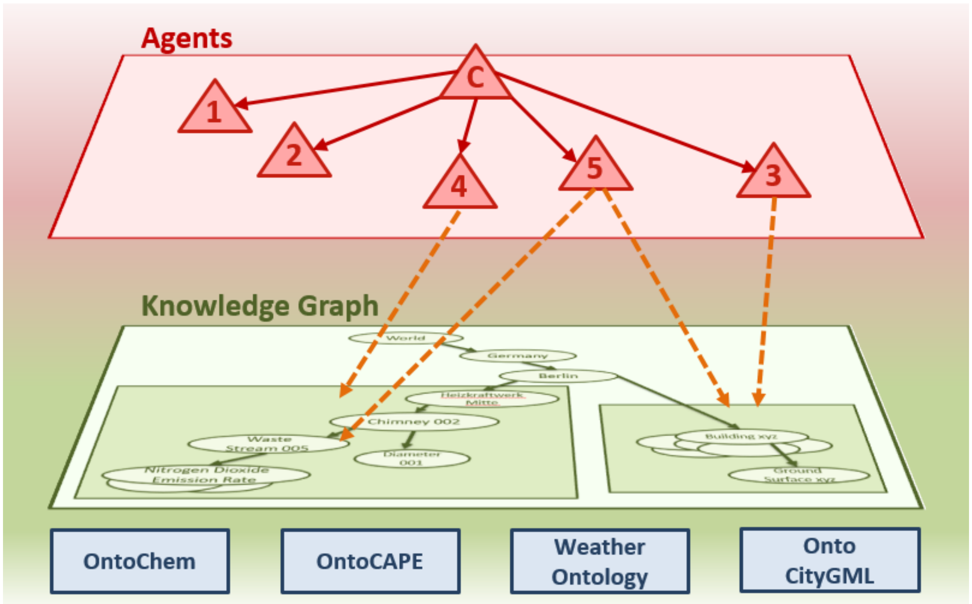
Keywords: Industry 4.0, ontology, linked data, knowledge graph, interoperability, agent, modelling, simulation, optimization, cross-domain, multi-level

Edited by
Computational Modelling Group
Department of Chemical Engineering and Biotechnology
University of Cambridge
West Cambridge Site
Philippa Fawcett Drive
Cambridge CB3 0AS
United Kingdom

Fax: + 44 (0)1223 334796
E-Mail: c4e@cam.ac.uk
World Wide Web: http://como.cheng.cam.ac.uk/

Abstract

In this paper, we illustrate the relevance of ontologies and semantic technologies in process industry and discuss how they can support the interoperability between agents in cross-domain scenarios. We present a comprehensive industrial air pollution scenario, that has been implemented as part of the J-Park Simulator, to analyze questions related to interoperability. The J-Park Simulator utilizes ontologies and acts as platform for integrating real-time data, knowledge, models and tools to fulfill objectives such as simulation and optimization in cross-domain and multi-level scenarios. It utilizes ontologies that are designed in a modular structure from different application domains and a distributed knowledge graph to store and link the data. We conclude that the architecture of JPS supports interoperability in cross-domain scenarios. The final discussion of the industrial air pollution scenario also reveals open questions that should be addressed in future works.



Highlights

- The importance of ontologies and semantic technologies for cross-domain scenarios in process industry is illustrated.
- The main architectural principles of J-Park Simulator are presented.
- A modular structure for domain ontologies and a distributed knowledge graph are proposed to support the interoperability of agents in cross-domain scenarios.
- An implementation of an industrial air pollution scenario is used to analyze the interoperabilty between agents and to discuss open questions for future work.

1 Introduction

Software tools for modelling, optimization and simulation are decisive in process industry. Interoperability between different tools and models has always played an important role in designing and simulating larger composed structures such as chemical plants. Interoperability can be described as the ability of systems to understand each other and to use each other's functionalities [1]. As the complexity and choice of tools increase in the future, interoperability will become even more critical. It is also considered as one of the key factors of Industry 4.0 [2]. To achieve interoperability, components and systems that are involved in the same application scenario "must be capable of automatically interpreting each other's roles and 'understanding' each other", and consequently, semantics and models are important research topics for Industry 4.0 [3].

Interoperability and semantics are especially critical in cross-domain scenarios which we will illustrate with an industrial air pollution scenario. This scenario is used throughout this paper and can be summarized as follows: The emissions of a power plant are estimated and by considering the effects of surrounding buildings and real-time weather conditions, the dispersion profiles for different pollutants are simulated. This short description already contains concepts from different domains such as "power plant", "pollutant", "building" and "weather". We implemented this scenario by utilizing two pieces of commercial software - one for estimating the plant's emissions and another for simulating the dispersion of the emitted pollutants. Consequently, both software have to share data related to emissions and pollutants. In addition, the second software has to process data related to weather conditions and buildings that are in the vicinity of the plant.

If we are only interested in the industrial air pollution scenario for a specific plant at a specific location, its implementation would be straight-forward. But this is not true if we want to vary, extend, generalize and/or combine the scenario with other scenarios. For example, we might want to extend the scenario to include additional emission sources such as chemical plants or vessels in a port or replace the commercial software estimating emissions with other simulation tools or with real-time measurements. We might also want to use the same scenario to determine the locations of new decentralized power plants in order to adhere to emission levels for nearby buildings. In addition, the power plant could be part of an eco-industrial park with chemical plants as its consumers and/or it could be connected to a smart grid where some of the buildings' roofs are equipped with solar panels. A cross-domain simulation would allow predictions for power demand, market prices and consumer behaviour etc. based on factors such as weather conditions. However, such complexity raises the question of how components and systems can access and understand information from different sources, e.g. building data from different locations.

In order to deal with the increasing complexity, a generic approach is required. Ontologies and related technologies can provide a uniform framework to describe data semantically, to share knowledge and to cope with heterogeneity in cross-domain applications. An ontology "is an explicit specification of a conceptualization" [4]. It defines and describes the concepts of an application domain and their relationships to each other in an expressive format that allows for logical reasoning and inference.

Ontologies have been designed and used by many application domains. However, we will only mention a few references that are relevant for the process industry below. Batres [5] gives a comprehensive overview about ontologies that have been used in process systems engineering. OntoCAPE is a large-scale ontology developed for computer-aided process engineering [6]. Wiesner et al. [7] use OntoCAPE to integrate information from different software tools and phases in process engineering. Zhou et al. [8] extend OntoCAPE for management of information in eco-industrial parks.

ISO 15926 is a standard that supports the exchange and integration of information from all phases of the life cycle of chemical plants [9]. It allows the use of different formats such as XML (Extensible Markup Language). Parts 8 and 12 of the standard refer to the semantic description with OWL (Web Ontology Language). Moreover, the data model of ISO 15926 is proposed as an upper ontology [10]. The DEXPI (Data Exchange in the Process Industry) initiative [11] supports interoperability with respect to ISO 15926. Fillinger et al. [12] illustrate a prototype for an XML-based data exchange between P&ID tools from different vendors by using ISO 15926 and tools from DEXPI.

Bramsiepe et al. [13] analyze methods to reduce lead time in chemical engineering process and plant design. They identified proprietary formats for data exchange as one of the key challenges in plant design and speculate that OntoCAPE, ISO 15926 and ISO 10303 (STEP, Standard for the Exchange of Product model data) could increase the interchangeability of data in chemical industry. Muñoz et al. [14] present a batch control ontology that is structured according to ANSI/ISA-88, a standard for batch control, and applied it successfully to the optimization of a simulated plant scenario.

According to [15], Industry 4.0 envisages an increasing number of machines, devices and services that are dynamically connected and make decentralized decisions by accessing, comprehending and combining a variety of local and global information (such as sensor data, electronic documents and models). Pötter et al. [16] illustrate that many ideas of Industry 4.0 are applicable to process industry despite that the nature of batch and continuous processes is different from processes in manufacturing industry. Consequently, challenges associated with heterogeneity and interoperability will also arise in process industry in the context of Industry 4.0. Ontologies seem to be particularly suitable in scenarios where querying, reasoning and inference on heterogeneous data is required. For example, Graube et al. [17] present an approach to integrate and link heterogeneous industrial data from different systems and enterprises using ontologies and related technologies.

In this paper, we will use the J-Park Simulator (JPS) to analyze how to tackle heterogeneity and interoperability in complex cross-domain scenarios. JPS is part of the C4T project (Cambridge Centre for Carbon Reduction in Chemical Technology) [18] and acts as platform for integrating real-time data, knowledge, models and applications from different domains to fulfill objectives such as simulation and optimization. The initial goal of JPS is the reduction of CO_2 emissions from the industrial park on Jurong Island. To achieve this, a uniform and holistic approach was applied to different levels of modelling (unit, process, plant and industrial network level) and networks (for energy, power, waste and materials) [19], [20].

The use of ontologies and related semantic technologies in JPS for multi-level and cross-domain modelling and for decentralized management of data and knowledge has been successfully proven [21], [8]. The purpose of this paper is to illustrate how ontologies can also support interoperability between different applications in cross-domain scenarios. We will use JPS as a research platform to analyze this question along with the industrial air pollution scenario.

The remaining parts of this paper are organized as follows. Section 2 summarizes the main ideas of ontologies and linked data in an incomprehensive manner. Sections 3, 4 and 5 present the main architectural principles of JPS: Section 3 recaps the modular structure of domain ontologies that are currently used in JPS. Section 4 presents the JPS knowledge graph that is used to store and link concrete data and information and can be distributed over the World Wide Web. Section 5 illustrates how agents can operate on the JPS knowledge graph and collaborate with each other. In JPS, the term "agent" is used in a very broad context to refer to applications and services that utilize semantic technologies and are accessible on World Wide Web. In order to facilitate better understanding of JPS agents, we will first present the current implementation of these agents for the industrial air pollution scenario. This is followed by discussions of open questions concerning the current implementation and an outlook on how to address these questions in the future. Section 6 outlines the conclusions for this paper.

2 Semantic Web Stack

In 2012, Google added information boxes on the right side of its search result pages. When searching e.g. for "Marie Curie", the information box will present precise information about the scientist such as where she was born, when she died and whom she was married to etc. To achieve this, Google annotated and interlinked data semantically and stored them into an internal graph that represents knowledge in the form of grammatical triples. For example, the expression "Marie Curie was born in Warsaw" consists of the subject "Marie Curie", the predicate "was born in" and the object "Warsaw" where the subject and the object are nodes in the graph and the predicate is a directed edge from the subject to the object node. The expression "Marie Curie died on 04/07/1934" can be represented in the same way where the subject node coincides with the former example and the object node represents a concrete date. Similarly, further triples from expressions such as "Marie Curie is married to Pierre Curie", "Warsaw is capital of Poland" and "Pierre Curie was born in Paris" can be added to obtain a larger knowledge graph. "Marie Curie", "Warsaw", "Poland" etc. are not just strings but entities that can be related to each other. As Google has pointed out, the knowledge graph "also models all these inter-relationships. It's the intelligence between these different entities that's the key" [22].

The main principles behind knowledge graphs can be traced back to ideas of the semantic web and linked data that aim at the semantic description, understanding and integration of data on the World Wide Web [23], [24]. The World Wide Web Consortium (W3C) has published several standards and formats, the so-called semantic web stack, as basis for the semantic web [25]. Since the J-Park Simulator (JPS) relies heavily on these ideas, standards and related technologies, we will give an incomprehensive overview of some of its key concepts below.

An ontology formalizes the idea of grammatical triples and defines a vocabulary to de-

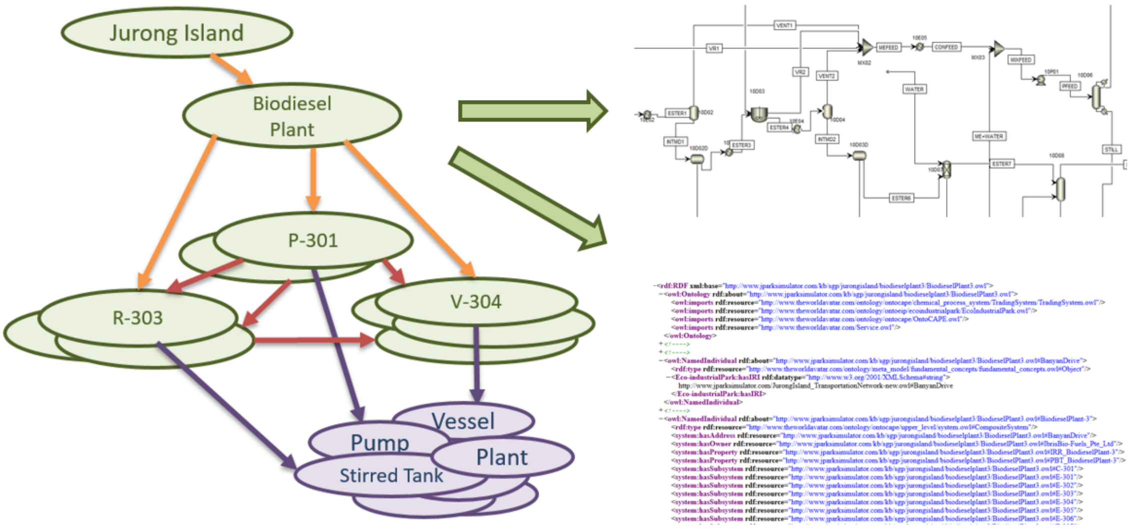


Figure 1: *A Biodiesel plant on Jurong Island represented in different formats: a) graph with instances (green nodes), relations (orange and red arrows), individual assertions (purple arrows) and classes (purple nodes), b) Aspen Plus process flow model, and c) machine readable format in RDF/XML.*

scribe an application domain in a semantic way. It distinguishes between classes, instances and relations: Classes denote concepts that constitute the application domain, instances represent concrete individuals of a given class and relations define which classes and instances can be linked to each other. The domain of a relation is specified by a concrete class and its range by a concrete class or data type (such as date). An instance can be linked to another instance or data value (such as "04/07/1934") using a relation only if the instances and values are consistent with the domain and range of that relation. In the expression "Marie Curie was born in Warsaw" the terms "Marie Curie" and "Warsaw" could be defined as instances of classes "Person" and "City" respectively, and "was born in" as relation with domain "Person" and range "City". In order to define classes, instances and relations, ontologies provide particular axioms and assertions.

The left side of figure 1 illustrates another example: It is greatly simplified and only presents some aspects of an existing Biodiesel plant on Jurong Island; the complete example was realized in detail as part of JPS in combination with simulations in Aspen Plus [21]. The upper right side of figure 1 shows the corresponding process flow model. The Biodiesel plant itself is represented as an instance of the class "Plant" and contains equipment instances of various classes such as "Pump", "Stirred Tank" and "Vessel" that are connected to each other. The relations "contains" and "is connected to" are represented by orange and red arrows respectively; purple arrows denote the individual assertion to define instances of a given class. Consequently, the graph in figure 1 represents expressions such as "P-301 is a Pump", "R-303 is a Stirred Tank" and "P-301 is connected to R-303".

JPS uses the Web Ontology Language (OWL) which is a powerful language for expressing ontologies. Its functionalities go far beyond the presented examples and allow the definition of sub classes, synonyms, properties of relations (such as symmetry, reflexivity and transitivity) etc. that can be used for reasoning and inference. Ontologies in OWL can be serialized, stored and exchanged in different formats, e.g., the lower right side of figure 1 shows a machine readable snippet in RDF/XML format that describes the Biodiesel plant. For purpose of illustration, we will use human readable names for all entities, e.g. "Plant", "R-303" and "is connected to", throughout this paper. Instead, OWL makes use of URLs (Uniform Resource Locators) to identify and resolve entities as web resources in a globally unique way[1]. Consequently, data that are distributed over the World Wide Web can be described and linked in a semantic way to eventually form a "giant global graph". OWL, RDF (Resource Description Framework) and XML (Extensible Markup Language) are parts of the semantic web stack. Moreover, it contains SPARQL, a language for semantic queries, and SWRL, a language to define rules.

3 Modular Cross-Domain Ontologies

JPS utilizes various ontologies that define classes and relations for different application domains at different levels. They are often designed and developed independently from each other and can be combined in a modular manner. These ontologies usually do not contain instances[2] and are applicable to different concrete scenarios. Table 1 lists examples of ontologies currently used in JPS.

OntoCAPE [6] is a large-scale ontology describing different aspects and levels for the domain of Computer Aided Process Engineering. It consists of four layers that are subdivided into more than fifty modules: the upper layers define more general concepts, e.g. from systems theory, while the lower layers are application-oriented and define concepts such as process units and plant equipment. OntoCAPE was the first ontology that was integrated into JPS and can be used to model chemical plants [21].

In general, JPS aims to integrate existing ontologies and standards as much as possible. Some of these existing ontologies have been slightly adapted to meet the dynamic requirements of JPS. Other ontologies such as OntoEIP, OntoPowSys and OntoKin have been designed and developed by researchers involved in the JPS project.

OntoEIP is a skeletal ontology to describe eco-industrial parks, inter-plant connectivities and networks for energy, power, water and materials. It has been applied as the framework for creating a decentralized knowledge graph of Jurong Island [8]. OntoKin consists of approximately 50 classes and 120 relations to represent reaction mechanisms [26]. OntoPowSys defines classes and relations for electrical power systems. OntoCityGML is an ontology to describe 3D models of cities and landscapes and was generated from the XML standard CityGML [27] by researchers from University of Geneva [28]. The Weather Ontology was created by researchers from Technical University of Vienna [29].

[1]For example, the instance "Biodiesel Plant" from figure 1 is identified in JPS by the URL http://www.theworldavatar.com/kb/sgp/jurongisland/biodieselplant3/BiodieselPlant3.owl. Requesting this URL will return an OWL file in RDF/XML format with information about the plant. Actually, OWL allows the use of IRIs (Internationalized Resource Identifiers) which are more general than URLs.

[2]The separation is similar to the use of T-box and A-box in description logic where the A-box contains statements that use conceptual models from T-box.

Table 1: *Examples of ontology from different domains used in JPS.*

Ontology	Domain
OntoCAPE	Computer aided process engineering [6]
OntoEIP	Eco-industrial park [8]
OntoKin	Reaction mechanisms [26]
OntoPowSys	Electrical power system [8]
OntoCityGML	Cities and landscapes [28]
Weather Ontology	Weather [29]
DBpedia	Cross-domain knowledge extracted from Wikipedia [30]

4 Distributed Knowledge Graph

JPS uses a knowledge graph to store and link information semantically. There is no standard definition for knowledge graphs but the number of instances in a knowledge graph is usually much larger than the number of its classes [31]. To allow for a high degree of reusability, many classes and relations in JPS are defined and bundled in modular ontologies as described in section 3. However, the JPS knowledge graph is defined as a structure which contains both the modular ontologies and the large number of instances, data values and their relations. Entities in the JPS knowledge graph are usually expressed in OWL (Web Ontology Language). But JPS can also utilize published linked data that are not expressed in OWL, e.g., JPS uses classes and instances from DBpedia, an RDF knowledge graph that contains extracted knowledge from Wikipedia. For the sake of completeness, DBpedia is added to table 1.

For the industrial air pollution scenario, two power plants, "Heizkraftwerk Mitte" in Berlin and "Energiecentrale" in The Hague, have been added to the JPS knowledge graph. We will focus on the power plant "Heizkraftwerk Mitte" in the following description. Figure 2 illustrates some related instances such as the power plant's chimney and waste stream in a simplified manner. The scenario mainly utilizes classes from OntoCAPE, OntoKin, OntoCityGML, DBpedia and weather ontology which are represented as blue boxes. We have migrated publicly available CityGML data for Berlin and The Hague to OntoCityGML and loaded them into two "triple stores" that allow for high-performance semantic queries with SPARQL. The two resulting knowledge bases for Berlin and The Hague can be considered as part of the entire JPS knowledge graph. Figure 2 sketches the ontological representation of a building in Berlin in the lower right corner. Moreover, the OntoKin knowledge bases provide detailed information about concrete reaction mechanisms in OWL. For example, one of the reaction mechanisms used in the scenario is a mechanism involving 109 species and 543 elementary reactions as proposed in [32].

Figure 2 also gives an idea of the current structure of JPS knowledge graph. Previous works for the JPS included description and optimization for chemical plants, power plants and various networks on Jurong Island as depicted by the upper left corner. Actually, the electrical network consists of nearly five hundred instances of classes such as "Bus" and "Transmission Line" from OntoPowSys which are coupled with chemical plants and waste heat recovery networks.

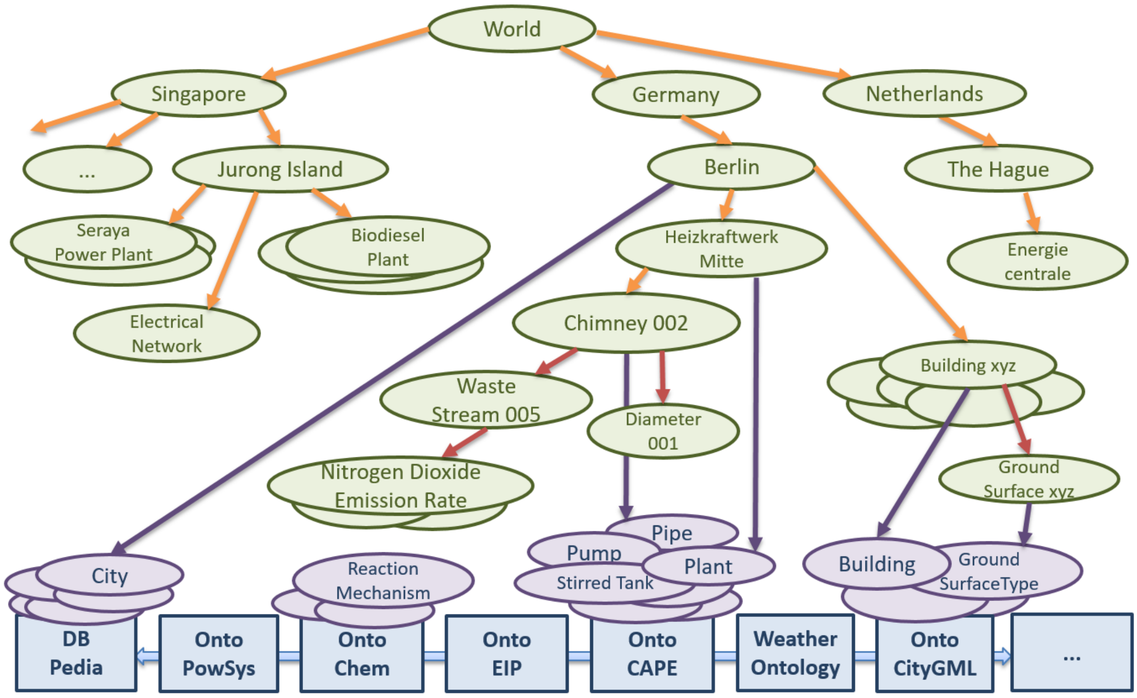


Figure 2: *Simplified part of JPS knowledge graph: Blue boxes denote reusable domain ontologies, purple nodes correspond to classes and green nodes to instances. Orange and red arrows denote different types of relations and purple arrows individual assertions.*

Since the JPS knowledge graph is based on the principles of linked data, it can be distributed over the web. Zhou et al. [8] have demonstrated how this could be used to establish a decentralized information management system of Jurong Island. In a real-world industrial air pollution scenario, the detailed ontological representation of "Heizkraftwerk Mitte" proprietary technology would be kept private, while only the waste stream becomes part of an external interface with controlled access. In contrast, the knowledge base for buildings in Berlin could be published as part of a governmental open data strategy and be queried using SPARQL in a similar way as DBpedia.

5 Agents and Interoperability

Figure 3 summarizes the main principles of JPS: The lower layer (blue boxes) denotes the modular and reusable domain ontologies. The middle layer (green) stores and links instances and data values and uses the ontologies for their semantic descriptions. Both the lower and the middle layer form the JPS knowledge graph which can be distributed over the World Wide Web, i.e. its sub graphs can be distributed on different web nodes (represented by green rectangles). The upper layer (red) consists of agents (represented by triangles) that interact with each other and operate on parts of the knowledge graph, depending on their granted access privileges.

This section consists of two parts: The first part presents the current implementation of

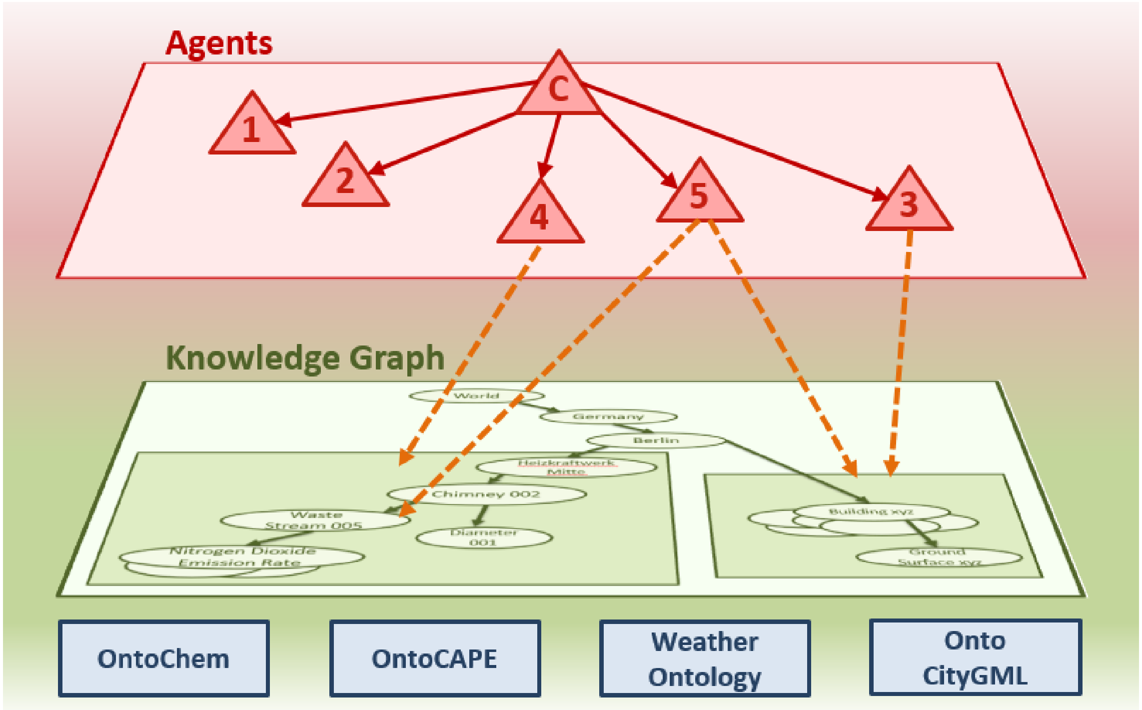


Figure 3: *Main principles of JPS illustrated for industrial air pollution scenario: a) modular domain ontologies (blue), b) distributed knowledge graph (green and blue), c) agents (red) operating on the knowledge graph and interacting with each other.*

the industrial air pollution scenario. It focuses on the agents' operation on the knowledge graph and on the semantic interoperability between agents, i.e. their ability to understand the exchanged data. The second part addresses questions arising from the current implementation and discusses how we could leverage the unleashed potential of semantic technologies to solve some of these questions in the future.

5.1 Implementation of the industrial air pollution scenario

JPS agents apply the semantic web stack, in particular SPARQL for semantic queries. They can read and understand information from the knowledge graph and modify its data values and structure. They can communicate with each other and exchange information via the knowledge graph and semantic input and output parameters. They use HTTP (Hypertext Transfer Protocol) for calling each other and mainly JSON (JavaScript Object Notation) for exchanging input and output parameters. Consequently, they could also run on different web nodes.

In the industrial air pollution scenario, the user selects a plant instance, a reaction mechanism instance and a region as the input parameters and initiates the simulation. The resulting dispersion profile can be viewed in a browser as shown in figure 4. After initiating the simulation, a coordination agent calls the other agents with the required input and output parameters in a consecutive manner; in figure 3 these agents are numbered from 1 to 5. All parameters are expressed in a semantic way, e.g., the selected plant is an instance of OntoCAPE class "Plant" and the selected reaction mechanism is an instance of OntoKin class "Reaction Mechanism"; both are identified by their URLs. The selected region is an instance of OntoCityGML class "EnvelopeType" with a nested structure that specify the coordinate reference system and coordinates of a spatial rectangular area.

The first agent uses Google's Geocoding API in combination with DBpedia's lookup service to retrieve the closest city to the selected region as an instance with its URL. The second agent requests a public web service for the real-time weather conditions close to the selected region and translates the non-semantic response into a semantic format using the weather ontology. The third agent uses the city URL to locate the corresponding building knowledge base, retrieves the coordinates of the selected plant from the JPS knowledge graph and queries for the buildings in the vicinity of the plant.

There are some trade-offs concerning the estimation of emissions from the power plant at this implementation stage: The industrial air pollution scenario focuses on demonstrating cross-domain interoperability rather than on detailed modelling of the power plant. SRM Engine Suite[3] is a tool for simulating exhaust gas emissions from internal combustion engines, and our research groups have comprehensive experience with it. In the industrial air pollution scenario, it is used as a proof of concept for the overall JPS architecture and will facilitate the integration of computational chemistry in JPS in the future. The fourth agent works as an ontological wrapper for SRM Engine Suite: it uses the URL of the selected reaction mechanism to query for the details from the OntoKin knowledge base and stores them into the SRM configuration files. The agent then starts the SRM simulation, annotates the simulation results semantically and modifies the waste stream of the selected plant in the knowledge graph.

Finally, the coordination agent calls the fifth agent, the ontological wrapper for the Atmospheric Dispersion Modelling System[4] (ADMS). This agent reads the waste stream information from the knowledge graph and queries for detailed information of the surrounding buildings, e.g. position and height, from the corresponding building knowledge base. It translates the building details together with the waste stream and weather information into the proprietary format of the ADMS input file and executes the ADMS simulation. ADMS estimates the concentration values in the selected region for all pollutants originating from the waste stream of the selected plant. The resulting output file can be annotated semantically by utilizing the W3C's standard for tabular data [33] and processed for visualization as shown in figure 4.

5.2 Discussion and Outlook

Agents involved in the industrial air pollution scenario map back and forth between ontologies and proprietary formats of utilized software products and web services. The associated additional implementation efforts might not be appropriate for a unique scenario where all involved software components and their communication are established and well-defined in advance. While this is indeed the case for the current implementation

[3] see http://cmclinnovations.com/products/srm/

[4] see http://www.cerc.co.uk/environmental-software/ADMS-model.html

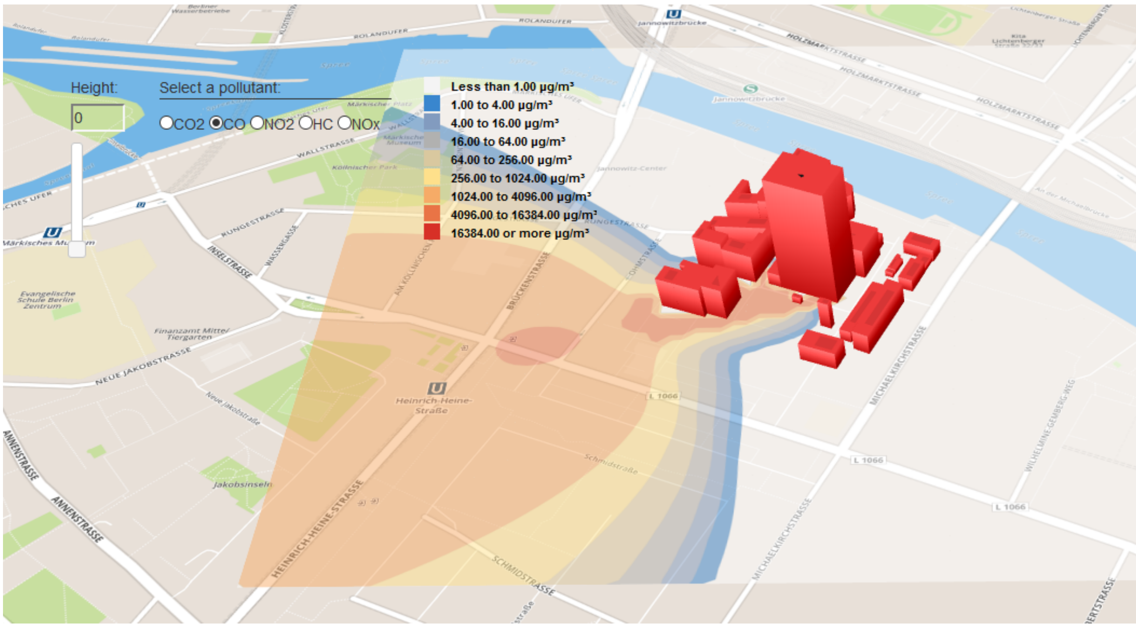


Figure 4: *Simulated dispersion profile of emissions estimated for the power plant "Heizkraftwerk Mitte" in Berlin. The figure shows the concentration profile for the selected pollutant, CO, at a height of zero meter taking into consideration the effects of surrounding buildings and real-time weather conditions.*

of the industrial air pollution scenario, the vision of JPS goes further beyond:

First of all, the implementation was carried out in such a way that it is applicable to any power plant that exhibit the same semantic structure as its waste stream in the knowledge graph. We have proven this for the power plant "Energiecentrale" in combination with migrated CityGML data from The Hague. In fact, this implementation could be easily extended to any emission source with the same waste stream structure. But in unmanaged environments such as the World Wide Web, different or variations in vocabularies are used for describing waste streams (or any other entities, e.g. weather, region, etc.). Consequently, the question of how to ensure interoperability and hence reusablity of agents in a more general context arises. Here, the full potential of semantic technologies comes into play. For example, ontology matching systems [34] can facilitate the semi-automatic alignment of classes from different ontologies that denote the same concept. Once classes have been aligned, they can be declared as synonymous using OWL. This in combination with logical reasoning can be used to automatically transform information into a semantically equivalent form that can be further processed by other agents.

Secondly, we use the term "agent" in JPS in a very broad context as already mentioned. Software agents usually exhibit some degree of intelligence and autonomy and can collaborate with other agents to achieve common goals. In contrast, agents involved in the industrial air pollution scenario provide stateless services which are called by a coordination agent with suitable input parameters. On the other hand, we have started to equip software components with semantic capabilities such that their input and output parameters, functionality and properties could be described using its own ontology. These semantic descriptions would also be part of the JPS knowledge graph in the future. This allows an agent to discover and communicate with other agents and combine their functionalities in an adaptive and automated manner. By doing so, JPS will be able to benefit greatly from research on semantic web services [35], service discovery and service composition [36].

Thirdly, although the JPS knowledge graph is extensible and scalable by design, most data that are available on the web currently are not described semantically. For the industrial air pollution scenario, we migrated from cityGML to OWL representation in advance. Alternatively, there are technologies such as ontology-based data access [37] that can translate semantic queries expressed in SPARQL into queries that act on relational databases and annotate the resulting data semantically on-the-fly. Hence, non-semantic data can also be integrated into the JPS knowledge graph with some additional mapping effort. Once JPS agents understand these data, they can also combine, query and reason on these data that is from different types of sources, i.e. JPS is not restricted to semantically described sources.

Fourthly, this paper mainly illustrates the conceptual principles of the JPS architecture and the use of semantic technologies. As mentioned, JPS agents use HTTP and mainly JSON for communicating with each other. We will now elaborate on the technological details further. Researchers participating in the C4T project work in different domains and explore different aspects of CO_2 emission reduction. JPS integrates their work (simulations, optimization algorithms, models, knowledge bases, experimental data etc.) by combining it with each other and 3rd party simulation software (SRM Engine Suite, ADMS, Aspen Plus etc.). This leads to a large variety of technologies being used for implementation, e.g. diverse programming languages (Java, JavaScript, Python, C++, etc.), web servers (Apache Tomcat, Node.js, nginx), triple stores (Apache Jena Fuseki, Eclipse RDF4J), solvers (GAMS, MATLAB etc.), Ethereum, Docker etc. While technological heterogeneity usually adds complexity for integration and maintenance, it is unavoidable when dealing with cross-domain context and will also facilitate innovation. Since technological heterogeneity does not affect semantic interoperability between JPS agents, the integration capabilities of JPS have been proven.

Fifthly, currently the user has to initiate the simulation for the industrial air pollution scenario manually. In the future, the simulation of the air pollutant's dispersion could be triggered automatically e.g. due to changing weather conditions or periodic real-time measurements of plant emissions. When the power plant is modelled in more details, a changing prognosis of power demand could also trigger the recalculation of the plant's waste stream which in turn would lead to an updated simulation of the pollutants' dispersion. In that sense, the knowledge graph becomes dynamic and evolves with time as changes in one node, e.g. real-time sensor data from a physical device, are propagated progressively by agents to the related nodes.

The above discussion can be used to derive the following categorization for the JPS agents: Type-0 agents operate on the real-world boundary of JPS and facilitate the information exchange via input activities (from users or sensors) or output activities (for reporting and visualizing results or for controlling actuators). Type-1 agents estimate, simulate, optimize and/or query the knowledge graph. Type-2 agents add and/or remove elements of the instance-level of the knowledge graph, i.e. the middle layer (in green) in figure 3. For example a type-2 agent could add a heat exchanger to an existing chemical plant as a result of an energy optimization. Type-3 and type-4 agents unleash the full potential of ontologies by providing higher-level and more generic functionalities. Type-3 agents facilitate the integration of existing vocabularies and domain knowledge into JPS and support ontology matching and the transformation of semantically equivalent structures. In the second above-mentioned point, in the future, the knowledge graph would be complemented by an "agent ontology" and instances describing the agents' functionalities and characteristics. This would allow type-4 agents (with the support of type-3 agents) to provide services for agent discovery and composition and to create new agents that control, simulate and optimize composed structures. Both type-3 and type-4 agents would be able to raise the current level of semantic interoperability to a higher level that allows for automated adaptive behavior in cross-domain scenarios of increasing complexity as described in the introduction.

As mentioned above type-1 agents can take multiple forms. A sub-set of this agent type plays an important role in JPS. These type-1 agents can be either based purely on data or on physical or chemical insight, i.e. a mathematical model motivated by natural laws. Even if such a model is based on physics and/or chemistry in almost all cases the model contains parameters that need estimating. Hence, as pointed out in [38], a full model is not only defined by its mathematical form but also by the data and methodology that is used in the process of parameter estimation. This needs to be taken into account if one is interested to improve the predictive power, evaluation speed or uncertainty analysis of a particular model. The methods that are used to do this form agents in their own right. For parameter estimation both frequentist or Bayessian methods have been employed. The construction of surrogate models of the original mathematical model often forms an important part of the process. In this paper surrogate model creation, parameter estimation, experimental design and error propagation were carried out using MoDS (Model Development Suite) [39]. The models of a biodiesel plant [40] and an internal combustion engine [41] serve as examples for type-1 agents that are based on surrogates which are currently in use in JPS. In both cases experimental design plays an important role. Both space filling [42] and adaptive methods have been developed [43]. Constructing surrogates from data alone has become more and more popular with the ubiquity of rich data sources. Deep learning methods represent an important and widely used class of methods. How to choose the best method for a particular data set depends on the user requirements. Machine learning algorithms have been used to make this choice [44]. All of these methods mentioned above have been or will be employed in JPS.

6 Conclusions

This paper illustrates the use of ontologies and semantic technologies in process industry and focuses on how they can support the interoperability between agents in cross-domain scenarios. We presented a comprehensive industrial air pollution scenario that utilizes concepts from different domains such as process engineering, reaction mechanisms, weather and buildings. The implementation of this scenario involves two pieces of commercial software, for estimating a power plant's emissions and for simulating the emitted pollutants' dispersion profile, three web services and knowledge bases for buildings and reaction mechanisms. In this paper, we used the scenario as a case study to analyze and discuss questions related to interoperability between agents.

The industrial air pollution scenario was constructed as part of the J-Park Simulator (JPS), an integration platform that relies on the semantic web stack and utilizes domain ontologies that are designed in a modular structure. Instances and their relations are defined using these ontologies. All entities (classes, instances and relations) are identified by their globally unique URLs and form the distributed JPS knowledge graph. This is the basis for semantic interoperability between JPS agents: JPS agents can exchange semantic information through reading from and writing to the knowledge graph and by using entities (identified by their URLs) as parameter values in their HTTP requests and responses. Semantic interoperability is further supported by the modular structure of the domain ontologies and their publication on the World Wide Web as linked data because both encourage the reuse of the ontologies for the same concepts.

However, in complex scenarios and unmanaged environments, JPS agents have to work with various ontologies that can be describing the same concepts and should also be able to adapt to changing requirements. The full potential of ontologies and associated research results on reasoning and inference, service discovery and composition, ontology matching, etc. can be fully unleashed in such situations. We believe that this potential will overcompensate the additional mapping efforts in our current implementation, and we will apply some of these results in our future works to achieve a higher level of interoperability in JPS.

Acknowledgements

This project is funded by the National Research Foundation (NRF), Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. MK gratefully acknowledges the support of the Alexander von Humboldt foundation.

List of abbreviations

EIP	**E**co-**I**ndustrial **P**ark
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol
JPS	**J**-**P**ark **S**imulator
JSON	**J**ava**S**cript **O**bject **N**otation
OWL	**W**eb **O**ntology **L**anguage
RDF	**R**esource **D**escription **F**ramework
SPARQL	**S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage
W3C	**W**orld **W**ide **W**eb **C**onsortium
XML	E**x**tensible **M**arkup **L**anguage

References

[1] David Chen, Guy Doumeingts, and François Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry*, 59 (7):647–659, 2008. doi:10.1016/j.compind.2007.12.016.

[2] Yang Lu. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6:1–10, 2017. doi:10.1016/j.jii.2017.04.005.

[3] Reiner Anderl, Klaus Bauer, Bernhard Diegner, Johannes Diemer, Alexander Fay, Jessica Fritz, Dietmar Goericke, Jürgen Grotepass, Claus Hilge, Jürgen Jasperneite, Johannes Kalhoff, Uwe Kubach, Ulrich Löwen, Georg Menges, Jan Stefan Michels, Wolfram Münch, Dr. Helena Preiß, Olaf Reus, Fabian Schmidt, Ernst-Joachim Steffens, Thomas Stiedl, Michael ten Hompel, and Christian Zeidler. Aspects of the research roadmap in application scenarios. *Federal Ministry for Economic Affairs and Energy, Tech. Rep.*, 2016.

[4] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. doi:10.1006/knac.1993.1008.

[5] Rafael Batres. Ontologies in process systems engineering. *Chemie Ingenieur Technik*, 89(11):1421–1431, 2017. doi:10.1002/cite.201700037.

[6] Jan Morbach, Andreas Wiesner, and Wolfgang Marquardt. OntoCAPE: A (re)usable ontology for computer-aided process engineering. *Computers & Chemical Engineering*, 33(10):1546–1556, 2009. doi:10.1016/j.compchemeng.2009.01.019.

[7] Andreas Wiesner, Jan Morbach, and Wolfgang Marquardt. Information integration in chemical process engineering based on semantic technologies. *Computers & Chemical Engineering*, 35(4):692–708, 2011. doi:10.1016/j.compchemeng.2010.12.003.

[8] Li Zhou, Chuan Zhang, Iftekhar A Karimi, and Markus Kraft. An ontology framework towards decentralized information management for eco-industrial parks. *Computers & Chemical Engineering*, 118:49–63, 2018. doi:10.1016/j.compchemeng.2018.07.010.

[9] International Organization for Standardization. Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities - Parts 8 and 12, last accessed 2018-12-18. https://www.iso.org/standard/52456.html, https://www.iso.org/standard/70695.html.

[10] Rafael Batres, Matthew West, David Leal, David Price, Katsube Masaki, Yukiyasu Shimada, Tetsuo Fuchino, and Yuji Naka. An upper ontology based on iso 15926. *Computers & Chemical Engineering*, 31(5-6):519–534, 2007. doi:10.1016/j.compchemeng.2006.07.004.

[11] Data Exchange in the Process Industry, last accessed 2018-12-18. http://www.dexpi.org.

[12] Sandra Fillinger, Henning Bonart, Wolfgang Welscher, Erik Esche, and Jens-Uwe Repke. Improving interoperability of engineering tools–data exchange in plant design. *Chemie Ingenieur Technik*, 89(11):1454–1463, 2017. doi:10.1002/cite.201700032.

[13] Christian Bramsiepe, Nicolai Krasberg, Christoph Fleischer, Lukas Hohmann, Norbert Kockmann, and Gerhard Schembecker. Information technologies for innovative process and plant design. *Chemie Ingenieur Technik*, 86(7):966–981, 2014. doi:10.1002/cite.201400029.

[14] Edrisi Muñoz, A Espuña, and Luis Puigjaner. Towards an ontological infrastructure for chemical batch process management. *Computers & Chemical Engineering*, 34 (5):668–682, 2010. doi:10.1016/j.compchemeng.2009.12.009.

[15] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for Industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016. doi:10.1109/HICSS.2016.488.

[16] Thorsten Pötter, Jens Folmer, and Birgit Vogel-Heuser. Enabling Industrie 4.0 - Chancen und Nutzen für die Prozessindustrie. In *Handbuch Industrie 4.0 Bd. 4*, pages 71–83. Springer, 2017.

[17] Markus Graube, Johannes Pfeffer, Jens Ziegler, and Leon Urbas. Linked data as integrating technology for industrial data. *International Journal of Distributed Systems and Technologies (IJDST)*, 3(3):40–52, 2012. doi:10.1109/NBiS.2011.33.

[18] Cambridge Centre for Carbon Reduction in Chemical Technology, last accessed 2018-12-18. http://www.cares.cam.ac.uk/c4t-project.

[19] Ming Pan, Janusz Sikorski, Jethro Akroyd, Sebastian Mosbach, Raymond Lau, and Markus Kraft. Design technologies for eco-industrial parks: From unit operations to processes, plants and industrial networks. *Applied Energy*, 175:305–323, 2016. doi:10.1016/j.apenergy.2016.05.019.

[20] Chuan Zhang, Li Zhou, Pulkit Chhabra, Sushant S Garud, Kevin Aditya, Alessandro Romagnoli, Gabriele Comodi, Fabio Dal Magro, Antonella Meneghetti, and Markus Kraft. A novel methodology for the design of waste heat recovery network in eco-industrial park using techno-economic analysis and multi-objective optimization. *Applied Energy*, 184:88–102, 2016. doi:10.1016/j.apenergy.2016.10.016.

[21] Li Zhou, Ming Pan, Janusz J Sikorski, Sushant Garud, Leonardus K Aditya, Martin J Kleinelanghorst, Iftekhar A Karimi, and Markus Kraft. Towards an ontological infrastructure for chemical process simulation and optimization in the context of eco-industrial parks. *Applied Energy*, 204:1284–1298, 2017. doi:10.1016/j.apenergy.2017.05.002.

[22] Google. Introducing the knowledge graph: things, not strings, last accessed 2018-12-18. http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[23] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[24] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011. doi:10.4018/jswis.2009081901.

[25] World Wide Web Consortium. Semantic web, last accessed 2018-12-18. https://www.w3.org/standards/semanticweb/.

[26] Feroz Farazi, Jethro Akroyd, Sebastian Mosbach, Philipp Buerger, Daniel Nurkowski, and Markus Kraft. OntoKin: An Ontology for Chemical Kinetic Reaction Mechanisms. Submitted, 2019.

[27] Open Geospatial Consortium. CityGML, last accessed 2018-12-18. http://www.opengeospatial.org/standards/citygml.

[28] Centre Universitaire d'Informatique at University of Geneva, last accessed 2018-12-18. http://cui.unige.ch/isi/onto/citygml2.0.owl.

[29] Institute of Computer Engineering at Technical University of Vienna, last accessed 2018-12-18. http://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/WeatherOntology.owl.

[30] DBpedia, last accessed 2018-12-18. https://wiki.dbpedia.org.

[31] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508, 2017.

[32] Hu Wang, Mingfa Yao, Zongyu Yue, Ming Jia, and Rolf D Reitz. A reduced toluene reference fuel chemical kinetic mechanism for combustion and polycyclic-aromatic hydrocarbon predictions. *Combustion and Flame*, 162(6):2390–2404, 2015. doi:10.1016/j.combustflame.2015.02.005.

[33] World Wide Web Consortium. Generating RDF from tabular data on the web, last accessed 2018-12-18. https://www.w3.org/TR/csv2rdf/.

[34] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013. doi:10.1109/TKDE.2011.253.

[35] Dieter Fensel, Federico Michele Facca, Elena Simperl, and Ioan Toma. *Semantic web services*. Springer Science & Business Media, 2011.

[36] Pablo Rodriguez-Mier, Carlos Pedrinaci, Manuel Lama, and Manuel Mucientes. An integrated semantic web service discovery and composition framework. *IEEE Transactions on Services Computing*, 9(4):537–550, 2016. doi:10.1109/TSC.2015.2402679.

[37] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: a survey. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 5511–5519. International Joint Conferences on Artificial Intelligence Organization, 2018. doi:10.24963/ijcai.2018/777.

[38] Markus Kraft and Sebastian Mosbach. The future of computational modelling in reaction engineering. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1924):3633–3644, 2010. doi:10.1098/rsta.2010.0124.

[39] MoDS. Description of the MoDS toolkit., last accessed 2019-02-04. `https://cmclinnovations.com/products/mods/`.

[40] Janusz J Sikorski, George Brownbridge, Sushant S Garud, Sebastian Mosbach, Iftekar A Karimi, and Markus Kraft. Parameterisation of a biodiesel plant process flow sheet model. *Computers & Chemical Engineering*, 95:108–122, 2016. doi:10.1016/j.compchemeng.2016.06.019.

[41] George Peter Edward Brownbridge, Andrew Smallbone, Weerapong Phadungsukanan, Sebastian Mosbach, Markus Kraft, and Bengt Johansson. Automated IC Engine Model Development with Uncertainty Propagation, 2011. SAE Technical Paper No. 2011-01-0237, doi:10.4271/2011-01-0237.

[42] Sushant Suhas Garud, Iftekhar A Karimi, and Markus Kraft. Smart sampling algorithm for surrogate model development. *Computers & Chemical Engineering*, 96: 103–114, 2017. doi:10.1016/j.compchemeng.2016.10.006.

[43] Sushant S Garud, Iftekhar A Karimi, and Markus Kraft. Design of computer experiments: A review. *Computers & Chemical Engineering*, 106:71–95, 2017. doi:10.1016/j.compchemeng.2017.05.010.

[44] Sushant S Garud, Iftekhar A Karimi, and Markus Kraft. LEAPS2: Learning based evolutionary assistive paradigm for surrogate selection. *Computers & Chemical Engineering*, 119:352–370, 2018. doi:10.1016/j.compchemeng.2018.09.008.