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

Andreas Eibeck[a,b], Mei Qi Lim[a,b], Markus Kraft[a,b,c,*]

[a] *CARES*
Cambridge Centre for Advanced Research and Education in Singapore,
1 Create Way,
CREATE Tower, #05-05,
Singapore, 138602.
[b] *University of Cambridge,*
Department of Chemical Engineering
and Biotechnology,
West Cambridge Site,
Philippa Fawcett Drive,
Cambridge, CB3 0AS
United Kingdom.
[c] *Nanyang Technological University,*
School of Chemical and
Biomedical Engineering,
62 Nanyang Drive,
Singapore,
637459.

Abstract

The J-Park Simulator (JPS) acts as a continuously growing platform for integrating real-time data, knowledge, models, and tools related to process industry. It aims at simulation and optimization in cross-domain and multi-level scenarios and relies heavily on ontologies and semantic technologies. In this paper, we demonstrate the interoperability between different applications in JPS, introduce new domain ontologies into the JPS, and integrate live data. For this, we utilize a knowledge graph to store and link semantically described data and models and create agents wrapping the applications and updating the data in the knowledge graph dynamically. We present a comprehensive industrial air pollution scenario, which has been implemented as part of the JPS, to show how knowledge graphs and modular domain ontologies support the interoperability

*Corresponding author
Email address: `mk306@cam.ac.uk` (Markus Kraft)

between agents. We show that the architecture of JPS increases the interoperability and flexibility in cross-domain scenarios and conclude that the potential of ontologies outweighs additional wrapping efforts.

Keywords: Industry 4.0, ontology, linked data, knowledge graph, interoperability, agent

---

1. Introduction

Software tools for modeling, 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 pieces of 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 straightforward. 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. 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 etc.

Ambient air pollution is considered a large environmental health risk and causes millions of deaths every year. The industrial air pollution scenario and its variations support the assessment of modeled emissions against air quality standards during the planning phase of industrial plants, for instance, to determine the stack height or to determine the locations of new decentralized power plants in order to comply with legal pollution thresholds for nearby buildings. They also support safety and emergency planning as well as air pollution forecasts during plant operation. While these scenarios are of high importance by themselves, this paper does not focus on a specific detailed case study but instead uses the industrial air pollution scenario to exemplify and discuss how simulation tools and other components referring to different domains may be coupled in a flexible way.

There are several approaches to tackle the problem of interoperability within the context of process engineering. Fillinger et al. [4] give an overview about standards and formats that address this problem. However, these approaches are specific to the data exchange between process simulation and CAD tools and do not consider tools related to other domains. In contrast, ontologies and related technologies 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 [5]. 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 [6] 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 [7]. Wiesner et al. [8] use OntoCAPE to integrate information from different software tools and phases in process engineering.

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.

This shows that ontologies have been widely used for data exchange, control and simulation within the context of process industry. Besides that, ontologies have been successfully utilized in cross-domain applications, e.g. in the areas of Internet of Things [15] and Smart Manufacturing [16], or have been proposed to integrate and link heterogeneous industrial data from different systems and enterprises [17]. But on the other hand, these cross-domain applications do not consider process engineering tools.

In this paper, we will use the J-Park Simulator (JPS) to implement the industrial air pollution scenario. JPS is part of the C4T project (Cambridge Centre for Carbon Reduction in Chemical Technology) [18] and has continuously expanded as a 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 was the reduction of CO_2 emissions from the industrial park on Singapore's Jurong Island. To achieve this, a uniform and holistic approach was applied to different levels of modeling (unit, process, plant, and industrial network level) and networks (for energy, power, waste, and materials) in [19] and [20]. At a later stage, ontologies and related semantic technologies were introduced into JPS successfully: Zhou et al. [21] focused on the process simulation and optimization of chemical plants modeled by means of OntoCAPE as a first step towards a knowledge base for an overall industrial park. Zhou et al. [22] developed a skeletal ontology for multi-level and cross-domain modeling in eco-industrial parks (EIPs) by adapting and extending OntoCAPE; they also introduced an ontological knowledge base for managing information in EIPs in a decentralized manner. However, the authors did not address two important questions: How can different software applications change the knowledge base dynamically and collaborate with each other, and how can live data be incorporated into the knowledge base?

The **purpose of this paper** is to, firstly, dynamically update the JPS, demonstrating interoperability between different applications solving a cross-domain problem, secondly, to introduce new domain ontologies into the JPS and, thirdly, to integrate live data into the knowledge base. We do this by implementing the industrial air pollution scenario for a power station in Berlin and in The Hague while keeping in mind that the implementation should easily be adapted to other emission sources and more complex scenarios.

The remaining parts of this paper are organized as follows. Section 2 summarizes the main ideas of ontologies and linked data in a incomprehensive manner.

Sections 3 and 4 present the main architectural principles of JPS along the industrial air pollution scenario: Section 3 presents the JPS knowledge graph that is used to store and link data and information in a distributed manner and emphasizes the modular structure of domain ontologies. Section 4 illustrates how agents can operate on the JPS knowledge graph, collaborate with each other, and use live data. 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 the 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 5 outlines the conclusions for this paper.

2. Semantic Web Stack

The World Wide Web Consortium (W3C) has published several standards and formats, the so-called Semantic Web stack [23], that aim at the semantic description, understanding, and integration of data on the World Wide Web [24]. Since JPS relies heavily on these standards and related technologies, we will give a incomprehensive overview of some of its key concepts.

An ontology defines a vocabulary to describe 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 double). An instance can be linked to another instance or data value (such as "45.3") using a relation only if the instances and values are consistent with the domain and range of that relation.

The left side of Figure 1 illustrates an example: It is greatly simplified and

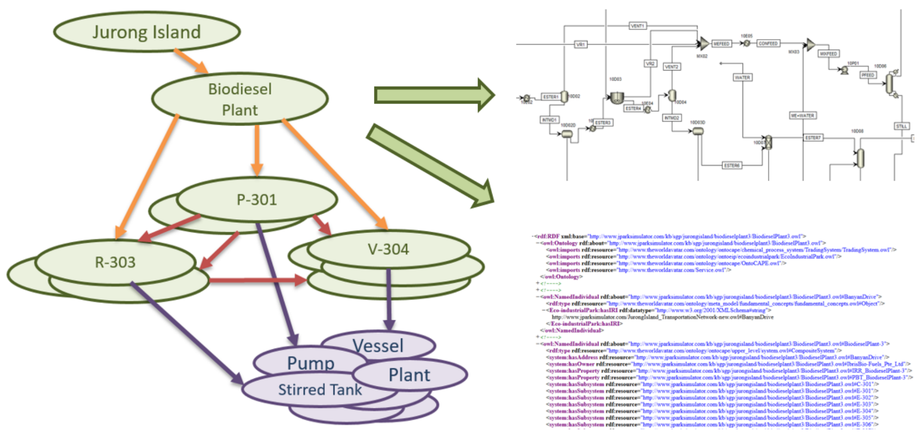

Figure 1: A biodiesel plant on Jurong Island represented in different formats: a) graph with instances (green nodes), relations (orange and red arrows), individual assertions (purple arrows), and classes (purple nodes), b) Aspen Plus process flow model, and c) machine readable format in RDF/XML.

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 grammatical triples such as "P-301 is a Pump", "R-303 is a Stirred Tank", and "P-301 is connected to R-303". Each triple consists of a subject, a predicate, and an object, where the subject and object are presented as nodes and the predicate as directed edge.

JPS uses the Web Ontology Language (OWL), which is a powerful language for expressing ontologies. It provides particular axioms and assertions to define classes, instances, and relations. Its functionalities go far beyond the presented example 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 the 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 “global data graph” [25]. OWL, RDF (Resource Description Framework), XML (Extensible Markup Language), and URLs are parts of the Semantic Web stack. The Semantic Web stack also contains SPARQL (SPARQL Protocol and RDF Query Language), a language to query and update such semantic data graphs.

3. Knowledge Graph

JPS makes use of SPARQL to query, insert, change, and delete triples in its data graph. In the following, we will denote this data graph as the JPS knowledge graph. The JPS knowledge graph may be regarded as part of the “global data graph” since it links entities that are defined and published “outside” JPS, and vice versa, publishes entities that may be linked from elsewhere. There is no standard definition for the term “knowledge graph”, but the number of instances stored in a knowledge graph is usually much larger than the number of classes [26], which is also the case for the JPS knowledge graph.

In accordance with best practices of the Semantic Web, classes and relations

[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.

are usually defined separately from the instances[2], bundled in modular ontologies and published on the World Wide Web. Many researchers have designed and published ontologies for their specific domains following these principles. This allows for a high degree of reusability and thus reduces design efforts and helps to link data from different sources. JPS follows the same principles and aims to integrate existing modular domain ontologies if suitable – sometimes with slight adaptations. Nevertheless, some ontologies were created from scratch in the past because proper ontologies for previously implemented JPS use cases had been missing; this is the case, for example, for the ontologies OntoEIP, OntoPowSys, and OntoKin listed at the beginning of Table 1.

Table 1: Examples of ontology from different domains used in JPS. The industrial air pollution scenario makes use of the underlined ontologies and of the DBpedia knowledge graph.

Ontology	**Domain**
OntoEIP	Eco-industrial park [22]
OntoPowSys	Electrical power system [22]
<u>OntoKin</u>	Reaction mechanisms [27]
<u>OntoCAPE</u>	Computer aided process engineering [7]
<u>OntoCityGML</u>	Cities and landscapes [28]
<u>Weather Ontology</u>	Weather [29]
DBpedia	Cross-domain knowledge extracted from Wikipedia [30]

Concerning the cross-domain scenario of industrial air pollution, we identified the relevant domains and decided to reuse the ontologies OntoCAPE, OntoKin, and OntoCityGML, and a weather ontology. While OntoCAPE and OntoKin have been used in previously implemented JPS use cases, OntoCityGML and the weather ontology have become part of the JPS knowledge graph for the first time. Figure 2 sketches a part of the entire JPS knowledge graph and shows the modular domain ontologies as blue boxes in combination with

[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 the T-box.

some typical classes as purple nodes. In the following paragraphs, we will briefly introduce the domain ontologies and the actual instances used for this paper.

OntoCAPE [7] is a large-scale ontology describing different aspects and levels for the domain of Computer Aided Process Engineering. It was the first ontology that was integrated into JPS to model chemical plants on Jurong Island's industrial park, such as the biodiesel plant presented in Section 2. 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 for two reasons: While JPS was originally developed for modeling the industrial park on Jurong Island, we also want to show in this paper that the general architecture of JPS is applicable and extensible to other locations, domains, and scenarios. The second reason is that building data for Singapore are simply not available publicly, contrary to Berlin and The Hague. 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.

OntoKin [27] consists of approximately 50 classes and 120 relations to represent reaction mechanisms. The JPS knowledge graph provides many instances of reaction mechanisms with detailed information.

OntoCityGML is an ontology to describe 3D models of cities and landscapes and was generated from the XML standard CityGML [31] by researchers from University of Geneva [28]. 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.

The Weather Ontology was created by researchers from Technical University of Vienna [29] and is used to describe the current weather conditions such as temperature, wind speed, and wind direction for arbitrary locations in a semantic way. Moreover, the scenario makes use of classes and instances from

DBpedia, such as “City” and “Berlin”. DBpedia is a published RDF knowledge graph that contains extracted knowledge from Wikipedia [30]. For the sake of completeness, DBpedia was added to Table 1 and Figure 2.

The JPS knowledge graph may be regarded as a large set of semantic triples containing information and data for the industrial air pollution scenario and further scenarios previously implemented in JPS. These triples are organized in subsets (e.g. domain ontologies, building knowledge base for Berlin or the description of “Heizkraftwerk Mitte” in Berlin) and can be stored on different web nodes. Zhou et al. [22] 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.

4. 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 the industrial air pollution scenario. It focuses on the agents' operation on the knowledge graph and on the semantic interoperability between

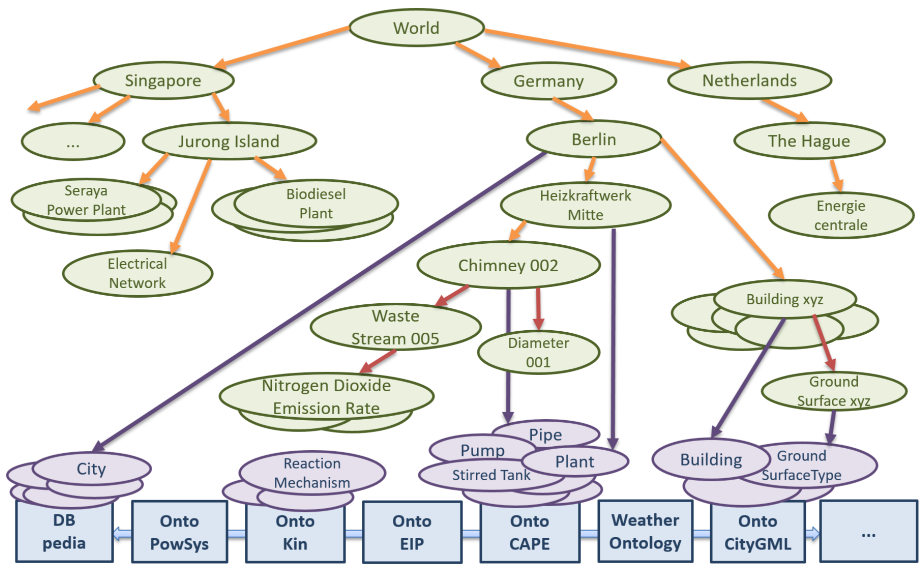

Figure 2: Simplified part of JPS knowledge graph: Blue boxes denote reusable domain ontologies, purple nodes correspond to classes and green nodes to instances. Orange and red arrows denote different types of relations and purple arrows individual assertions.

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.

4.1. 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

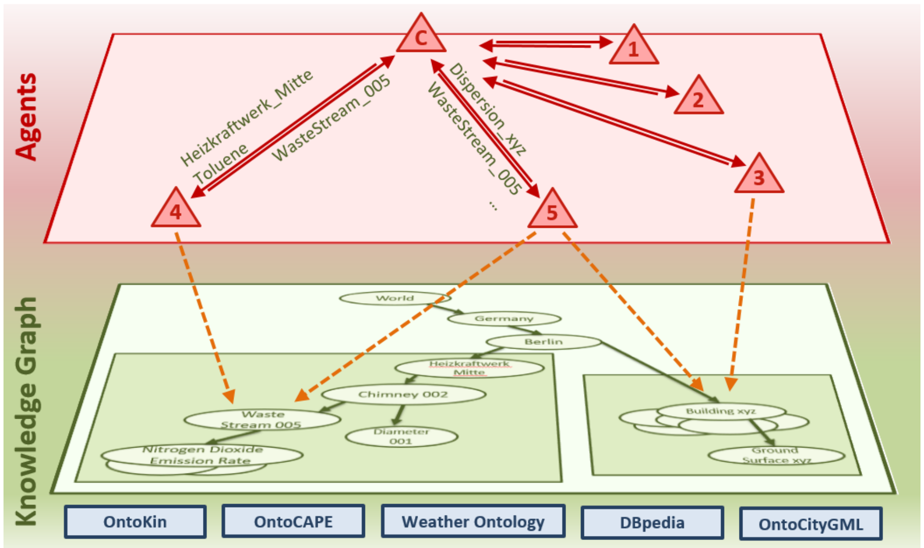

Figure 3: Main principles of JPS illustrated for the industrial air pollution scenario: a) modular domain ontologies (blue), b) knowledge graph (green and blue) with sub graphs distributed on the Web, and c) agents (red triangles) operating on the knowledge graph and interacting with each other.

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 specifies 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 modeling of the power plant. SRM Engine Suite[3] is a tool for simulating exhaust gas emissions from internal combustion engines with which our research groups have comprehensive experience. 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 receives the URLs for the selected power plant and reaction mechanism, e.g. in Figure 3 the URLs for “Heizkraftwerk Mitte” and “Toluene”[4].

The reaction mechanism “Toluene” involves 109 species and 543 elementary reactions and was proposed in [32]. It may be replaced by any other reaction mechanism instance from the OntoKin subgraph. The agent uses the URLs to query for the details of the power plant and the reaction mechanism from the knowledge graph and to map and store them into the SRM configuration files. The agent then starts the SRM simulation, annotates the simulation result semantically, modifies the waste stream of the selected plant in the knowledge graph, and returns the URL for the modified waste stream (in Figure 3 “WasteStream 005”) .

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

[4]As mentioned in Section 2, we use human readable entity names instead of complete URLs in this paper. The agents use the corresponding URLs instead, e.g. the fourth agent in Figure 3 receives the URL http://www.theworldavatar.com/kb/ontokin/Toluene.owl#ReactionMechanism_187077735769001 instead of the string “Toluene”.

Finally, the coordination agent calls the fifth agent, the ontological wrapper for the Atmospheric Dispersion Modelling System[5] (ADMS), with weather information and the URLs for the waste stream and buildings. 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.

4.2. 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 of the industrial air pollution scenario, the vision of JPS goes further.

First of all, the implementation was carried out in such a way that it is applicable to any power plant that exhibits 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. The lower picture of Figure 4 shows the simulated dispersion profile for 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 varied vocabularies are

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

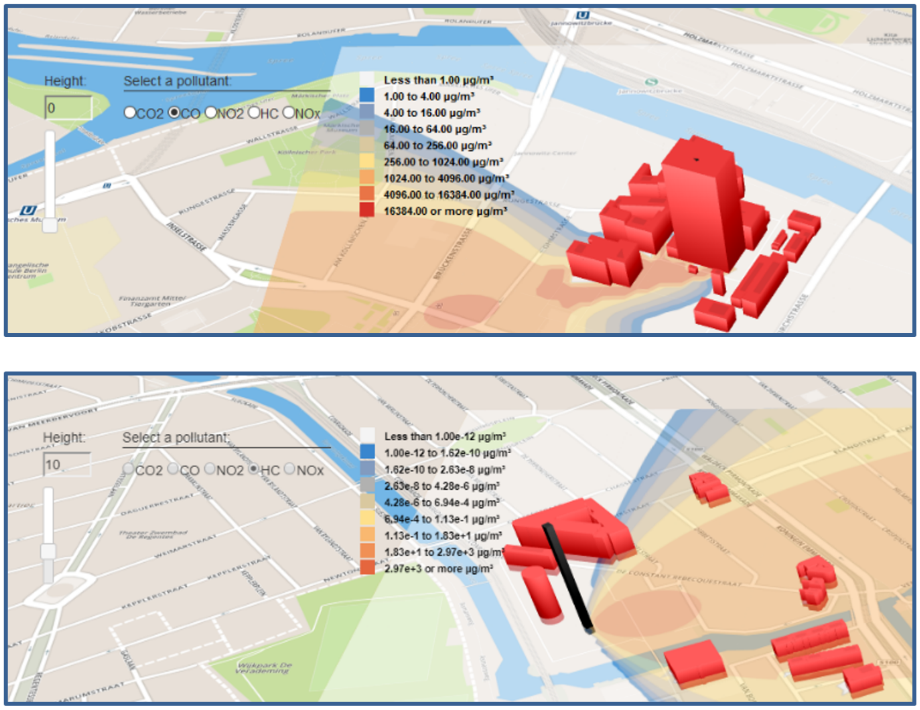

Figure 4: Simulated dispersion profile of CO emissions estimated for "Heizkraftwerk Mitte" in Berlin at a height of zero meters (upper picture) and of unburned hydrocarbons emissions estimated for "Energiecentrale" in The Hague at a height of ten meters (lower picture). Both simulations take into consideration the effects of surrounding buildings and real-time weather conditions.

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 are 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 third 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 modeled in more detail, 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, such as 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 subset 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 in improving 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 Bayesian 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.

5. 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. We draw the following conclusions:

- Domain ontologies specify the domain concepts and their relations to each other. They provide the vocabularies that help agents to understand data and models. We showed that domain ontologies can easily be reused and integrated as a modular base for describing entities in cross-domain scenarios.
- Knowledge graphs allow storage and linking of data and models from different domains and sources in a distributed manner. We showed that agents can easily query the knowledge graph and update simulation results (such as estimated plant's emissions) and live data (such as current weather conditions) dynamically.
- The Semantic Web stack uses URLs to identify and resolve entities. We showed that agents can exchange information by providing URLs as input and output parameters and by using them to query entity-related details

from the knowledge graph. This allows the collaboration of agents performing simulation and optimization tasks related to different models and domains.

On the one hand, the initial effort for realizing agents operating on a knowledge graph might be higher compared to a straight-forward implementation for a specific use case. On the other hand, this approach increases the interoperability and flexibility in cross-domain scenarios where tools and applications are used that have been developed independently of each other. JPS might also be connected to other systems and platforms following a similar approach. However, in complex scenarios and unmanaged environments, JPS agents have to work with various ontologies 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 make up for the additional efforts of wrapping existing applications and tools, 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 Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. MK gratefully acknowledges the support of the Alexander von Humboldt foundation.

References

[1] D. Chen, G. Doumeingts, F. Vernadat, Architectures for enterprise integration and interoperability: Past, present and future, Computers in industry 59 (7) (2008) 647–659.

[2] Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues, Journal of Industrial Information Integration 6 (2017) 1–10.

[3] R. Anderl, K. Bauer, B. Diegner, J. Diemer, A. Fay, J. Firtz, D. Goericke, J. Grotepass, C. Hilge, J. Jasperneite, et al., Aspects of the research roadmap in application scenarios, Federal Ministry for Economic Affairs and Energy, Tech. Rep.

[4] S. Fillinger, E. Esche, G. Tolksdorf, W. Welscher, G. Wozny, J.-U. Repke, Data exchange for process engineering–challenges and opportunities, Chemie Ingenieur Technik 91 (3) (2019) 256–267.

[5] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge acquisition 5 (2) (1993) 199–220.

[6] R. Batres, Ontologies in process systems engineering, Chemie Ingenieur Technik 89 (11) (2017) 1421–1431.

[7] J. Morbach, A. Wiesner, W. Marquardt, Ontocape: A (re) usable ontology for computer-aided process engineering, Computers & Chemical Engineering 33 (10) (2009) 1546–1556.

[8] A. Wiesner, J. Morbach, W. Marquardt, Information integration in chemical process engineering based on semantic technologies, Computers & Chemical Engineering 35 (4) (2011) 692–708.

[9] I. O. 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, `https://www.iso.org/standard/52456.html`, `https://www.iso.org/standard/70695.html` (last accessed 2018-12-18).

[10] R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino, Y. Naka, An upper ontology based on iso 15926, Computers & Chemical Engineering 31 (5-6) (2007) 519–534.

[11] D. E. in the Process Industry, http://www.dexpi.org (last accessed 2018-12-18).

[12] S. Fillinger, H. Bonart, W. Welscher, E. Esche, J.-U. Repke, Improving interoperability of engineering tools–data exchange in plant design, Chemie Ingenieur Technik 89 (11) (2017) 1454–1463.

[13] C. Bramsiepe, N. Krasberg, C. Fleischer, L. Hohmann, N. Kockmann, G. Schembecker, Information technologies for innovative process and plant design, Chemie Ingenieur Technik 86 (7) (2014) 966–981.

[14] E. Muñoz, A. Espuña, L. Puigjaner, Towards an ontological infrastructure for chemical batch process management, Computers & chemical engineering 34 (5) (2010) 668–682.

[15] A. Gyrard, S. K. Datta, C. Bonnet, K. Boudaoud, Cross-domain internet of things application development: M3 framework and evaluation, in: 2015 3rd International Conference on Future Internet of Things and Cloud, IEEE, 2015, pp. 9–16.

[16] I. Grangel-González, L. Halilaj, M.-E. Vidal, O. Rana, S. Lohmann, S. Auer, A. W. Müller, Knowledge graphs for semantically integrating cyber-physical systems, in: International Conference on Database and Expert Systems Applications, Springer, 2018, pp. 184–199.

[17] M. Graube, J. Pfeffer, J. Ziegler, L. Urbas, Linked data as integrating technology for industrial data, International Journal of Distributed Systems and Technologies (IJDST) 3 (3) (2012) 40–52.

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

[19] M. Pan, J. Sikorski, J. Akroyd, S. Mosbach, R. Lau, M. Kraft, Design technologies for eco-industrial parks: From unit operations to processes, plants and industrial networks, Applied Energy 175 (2016) 305–323.

[20] C. Zhang, L. Zhou, P. Chhabra, S. S. Garud, K. Aditya, A. Romagnoli, G. Comodi, F. Dal Magro, A. Meneghetti, M. 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 (2016) 88–102.

[21] L. Zhou, M. Pan, J. J. Sikorski, S. Garud, L. K. Aditya, M. J. Kleinelanghorst, I. A. Karimi, M. Kraft, Towards an ontological infrastructure for chemical process simulation and optimization in the context of eco-industrial parks, Applied Energy 204 (2017) 1284–1298.

[22] L. Zhou, C. Zhang, I. A. Karimi, M. Kraft, An ontology framework towards decentralized information management for eco-industrial parks, Computers & Chemical Engineering 118 (2018) 49–63.

[23] W. W. W. Consortium, Semantic web, `https://www.w3.org/standards/semanticweb/` (last accessed 2018-12-18).

[24] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific american 284 (5) (2001) 34–43.

[25] C. Bizer, T. Heath, T. Berners-Lee, Linked data: The story so far, in: Semantic services, interoperability and web applications: emerging concepts, IGI Global, 2011, pp. 205–227.

[26] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic web 8 (3) (2017) 489–508.

[27] F. Farazi, J. Akroyd, S. Mosbach, P. Buerger, D. Nurkowski, M. Kraft, OntoKin: An Ontology for Chemical Kinetic Reaction Mechanisms, submitted (2019).

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

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

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

[31] O. G. Consortium, CityGML, `http://www.opengeospatial.org/standards/citygml` (last accessed 2018-12-18).

[32] H. Wang, M. Yao, Z. Yue, M. Jia, R. D. Reitz, A reduced toluene reference fuel chemical kinetic mechanism for combustion and polycyclic-aromatic hydrocarbon predictions, Combustion and Flame 162 (6) (2015) 2390–2404.

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

[34] P. Shvaiko, J. Euzenat, Ontology matching: state of the art and future challenges, IEEE Transactions on knowledge and data engineering 25 (1) (2013) 158–176.

[35] D. Fensel, F. M. Facca, E. Simperl, I. Toma, Semantic web services, Springer Science & Business Media, 2011.

[36] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, M. Mucientes, An integrated semantic web service discovery and composition framework, IEEE transactions on services computing 9 (4) (2016) 537–550.

[37] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: a survey, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 5511–5519.

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

[39] M. D. of the MoDS toolkit., https://cmclinnovations.com/products/mods/ (last accessed 2019-02-04).

[40] J. J. Sikorski, G. Brownbridge, S. S. Garud, S. Mosbach, I. A. Karimi, M. Kraft, Parameterisation of a biodiesel plant process flow sheet model, Computers & Chemical Engineering 95 (2016) 108–122. doi:10.1016/j.compchemeng.2016.06.019.

[41] G. P. E. Brownbridge, A. Smallbone, W. Phadungsukanan, S. Mosbach, M. Kraft, B. Johansson, Automated IC Engine Model Development with Uncertainty Propagation, SAE Technical Paper No. 2011-01-0237, doi:10.4271/2011-01-0237 (2011).

[42] S. S. Garud, I. A. Karimi, M. Kraft, Smart sampling algorithm for surrogate model development, Computers & Chemical Engineering 96 (2017) 103–114. doi:10.1016/j.compchemeng.2016.10.006.

[43] S. S. Garud, I. A. Karimi, M. Kraft, Design of computer experiments: A review, Computers & Chemical Engineering 106 (2017) 71–95. doi:10.1016/j.compchemeng.2017.05.010.

[44] S. S. Garud, I. A. Karimi, M. Kraft, LEAPS2: Learning based evolutionary assistive paradigm for surrogate selection, Computers & Chemical Engineering 119 (2018) 352–370. doi:10.1016/j.compchemeng.2018.09.008.