Using Knowledge Graphs to Manage a Data Lake

Henrik Dibowski [1], Stefan Schmid[2]

Abstract: Knowledge graphs as fundamental pillar of artificial intelligence are experiencing a strong demand. In contrast to machine learning and deep learning, knowledge graphs do not require large amounts of (training) data and offer a bigger potential for a multitude of domains and problems. This article shows the application of knowledge graphs for the semantic description and management of data in a data lake, which improves the findability and reusability of data, and enables the automatic processing by algorithms. Since knowledge graphs contain both the data as well as its semantically described schema (ontology), they enable novel ontology-driven software architectures, in which the domain knowledge and business logic can completely reside on the knowledge graph level. This article further introduces such a use case: an ontology-driven frontend implementation, which is able to fully adapt itself based on the underlying knowledge graph schema and dynamically render information in the desired manner.

Keywords: Artificial Intelligence; Ontology; Knowledge Representation; Knowledge Graph; Semantic Data Lake; Data Catalog; Semantic Search; Semantic Layer; Ontology-Driven UIs

1 Introduction

Artificial intelligence (AI) is one of the biggest topics in computer science, and we have seen a big development of machine learning (ML) and deep learning (DL) during the past ten years. As many other enterprises in the world, Bosch is heavily investing in becoming one of the world-leading AI companies. The foundation of the Bosch Center for Artificial Intelligence in 2017 was an important milestone and meanwhile employs over a hundred AI experts. With ML and DL, data-driven subsymbolic approaches have dominated the past decade. Ontologies and knowledge graphs are another fundamental pillar of the AI landscape and they are emerging from the shadows. These approaches are fundamentally different, as knowledge is not represented by the weights of a neural network (black box), but in an explicit, symbolic way by means of logic languages with formal semantics (white box). Their inferences are deterministic and their results are traceable and explainable, which constitutes an essential requirement for a wide range of applications, such as autonomous driving. At Bosch, we are facing a high demand for knowledge graphs from various business units and are seeing a strong momentum. That is not much of a surprise, as Gartner recently rated knowledge graphs as one of the most promising emerging AI technologies of the next five to ten years [Br19]. We believe that knowledge graphs have a way bigger potential

[1] Robert Bosch GmbH, Corporate Research, 71272 Renningen, Germany, henrik.dibowski@de.bosch.com, https://orcid.org/0000-0002-9672-2387

[2] Robert Bosch GmbH, Corporate Research, 71272 Renningen, Germany, stefan.schmid@de.bosch.com

doi:10.18420/inf2020_02

for being a truly disruptive technology than ML and DL, because they do not depend on large amounts of available data for training the networks, and because they are applicable for representing knowledge of practically any domain, be it engineering, sciences, the humanities, medicine, etc.

The shift from subsymbolic to symbolic approaches goes along with a fundamental paradigm shift: from the use of raw data to the use of knowledge in AI. Purely subsymbolic approaches assume that with more and more data, AI can beat even the best algorithms. On the contrary, symbolic approaches emphasize that the quality of the data is more important than the quantity, and weigh knowledge over raw data. Data without description of its meaning is of low value, and it takes a lot of effort to turn such data into business value. Knowledge and semantically well-described information however is a key for many kinds of novel applications, machine understanding and truly smart AI.

2 Semantic Data Lakes

With the development of cloud computing, *data lakes* have emerged as new kind of cloud-based repositories for storing large amounts of data. Data lakes can store data in its native format in a flat architecture and run different types of analytics on the data. However, just "dumping" large amounts of data into a data lake does not provide value on its own. Without a contextual, semantic description of the data and without provenance information, the data stored in a data lake is unlikely to be usable by people and machines others than the ones that stored it and can still remember what was stored where. We reckon that this accounts for a majority of all existing data lakes, which are rather "*data swamps*" than data lakes. Data in a data lake is only (re)usable by people and machines, if it is semantically well described, and if the data provenance is clearly defined and tracked. *Semantic data lakes* are a specific form of data lakes in which a semantic layer on top enriches and connects the data semantically. The semantic layer overcomes data silos and enables semantic search across all data. This facilitates completely new, advanced use cases and analytics on the entirety of stored data. In our opinion, the best technology for realizing semantic data lakes are semantic technologies and knowledge graphs.

2.1 The role of data catalogs

In the era of big data, *data catalogs* emerged as the standard for metadata management. In the last few years, new application areas have appeared and the volume and richness of metadata required has grown significantly. Data lakes constitute one such important new application for data catalogs, besides warehouses, master data repositories, etc. According to Gartner, a data catalog "... maintains an inventory of data assets through the discovery, description, and organization of datasets. The catalog provides context to enable data analysts, data scientists, data stewards, and other data consumers to find and understand a relevant dataset for the purpose of extracting business value." [Ed17].

2.2 State of practice and criticism

Current vendors offer a wide range of commercial data catalog software. A sample of such vendors includes Alation Data Catalog, Atlan Enterprise Data Catalog, Talend Data Catalog, Collibra Data Catalog, Informatica Enterprise Data Catalog, Microsoft Azure Data Catalog, Oracle Cloud Infrastructure Data Catalog, and even Google is joining the market with its Google Data Catalog. To our knowledge, however, none of these data catalogs uses or supports standard semantic technologies (W3C recommendations), nor do they allow for using existing ontology vocabularies. Rather, they are closed, propriety systems with their own metadata languages and glossaries.

Our solution differs from existing solutions by proposing a semantic data lake architecture that incorporates a semantic data catalog, built with standard semantic technologies, and that addresses provenance and access control for resources in the data lake. This solution is described in detail in the following sections.

3 Semantic data lake catalog ontology

As a primary contribution, this section describes a data catalog ontology for semantic data lakes: the DCPAC ontology (Data Catalog, Provenance, and Access Control). The DCPAC ontology can be applied for adding a semantic layer to a data lake, which provides semantic description of the content, provenance, and access control permissions of the resources in a data lake. This ontology was created by combining several common, (predominantly) standardized ontology vocabularies and by aligning and extending them where necessary.

3.1 Ontology layer architecture

A big benefit of semantic technologies is the possibility to reuse, combine and extend existing ontology vocabularies, instead of reinventing the wheel. There is a large amount of open and commonly used ontologies available, which cover many domains and specify the domain knowledge and expertise of thousands of domain experts and knowledge engineers. One can only benefit from reusing such vocabularies, and further contribute and extend them. Building on open and standardized vocabularies leads to interoperable metadata representations and enables the exchange of information between different systems and applications. This is the approach that we followed, and the result can be seen in Fig. 1. The figure shows a layer architecture diagram of the DCPAC ontology. Boxes in this figure each represent an ontology, and the arrows between them define the vocabularies that are imported. The DCPAC ontology is shown at the bottom, and recursively imports all other ontologies. Additionally, it defines SHACL constraints for validating instance data (ABox).

In the following subsection, the primary ontologies utilized by DCPAC are introduced.

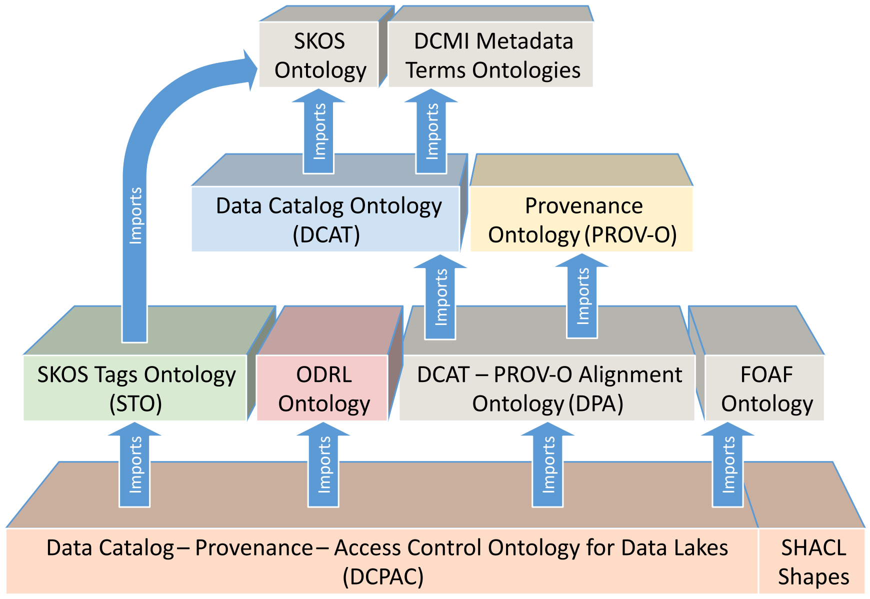

SKOS Ontology
DCMI Metadata Terms Ontologies
Imports
Data Catalog Ontology (DCAT)
Provenance Ontology (PROV-O)
SKOS Tags Ontology (STO)
ODRL Ontology
DCAT – PROV-O Alignment Ontology (DPA)
FOAF Ontology
Data Catalog – Provenance – Access Control Ontology for Data Lakes (DCPAC)
SHACL Shapes

Fig. 1: Layer architecture of the semantic data lake catalog ontology

3.2 Utilized ontology vocabularies

The *Data Catalog (DCAT) ontology* (prefix: dcat) is a recent W3C recommendation and constitutes "... an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web." [Al20]. The DCAT ontology imports and uses the widely recognized SKOS [MB09] and DCMI Metadata Terms [Du20] ontologies. Its primary purpose in the context of the DCPAC ontology is the semantic description of the content of resources in a data lake.

The *Provenance Ontology (PROV-O)* (prefix: prov) is another W3C recommendation that "... provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts." [Le13]. Its purpose in the context of DCPAC is to describe the provenance of the data lake resources.

The *Open Digital Rights Language (ODRL) Ontology* (prefix: odrl) "... is a policy expression language from W3C that provides a flexible and interoperable information model ... for representing statements about the usage of content and services." [Ia17]. In our data lake scenario, ODRL is applied to defining access control permissions for the data lake resources.

The *DCAT – PROV-O Alignment (DPA) ontology* (prefix: dpa) [WW19] was created by the

W3C Dataset Exchange Working Group and contains alignment axioms between DCAT ontology and PROV-O. Thereby, it enhances the DCAT ontology with the ability to use PROV-O for expressing advanced provenance information.

The *Simple Knowledge Organization System (SKOS)* (prefix: skos) is “a common data model for sharing and linking knowledge organization systems” [MB09]. We use SKOS to define separate domain-specific *SKOS Tags Ontologies (STO)* that comprise sets of *semantic tags* and their semantic relationships. By assigning each dataset a set of relevant semantic tags, they provide semantic description of the content of data lake resources and enable semantic search on the resources in the semantic data lake.

3.3 Data Catalog – Provenance – Access Control (DCPAC) Ontology

The *Data Catalog – Provenance – Access Control (DCPAC) ontology* (prefix: dcpac) [Di20] is our primary contribution to the ontology layer architecture (see Fig. 1). It combines, aligns and extends the other ontologies. The ontology imports the ODRL, DPA, and FOAF (“Friend of a Friend”) ontology [BM14] and optionally one or more STO ontologies, and recursively imports all other shown ontologies.

The DCPAC ontology aligns the DCAT ontology with the ODRL ontology by declaring the classes `dcat:Distribution` and `dcat:Resource` to be subclasses of `odrl:Asset`. This enables the definition of access control permissions for these DCAT classes and subclasses with the ODRL vocabulary. Another contribution is the alignment of PROV-O with the ODRL ontology by declaring the PROV-O class `prov:Agent` as subclass of `odrl:Party`, hence enabling all instances of `prov:Agent` to undertake roles in access control permissions. Additionally, the DCPAC ontology defines new subclasses of `dcat:Dataset` and `prov:Activity`, which allow for distinguishing different types of datasets and activities.

The DCPAC ontology is associated with a SHACL shapes definition file that defines a comprehensive set of SHACL constraints [KK17]. They can define cardinalities and type restrictions on properties, and regular expressions on the allowed values of string properties. Also complex constraints can be defined as SPARQL-based graph patterns. A SHACL engine can process the constraints and validate the consistency of the knowledge graph (ABox). That improves the integrity and quality and prevents issues.

3.4 The core vocabulary

This Section provides a short explanation of the core vocabulary of the DCPAC ontology and the primary imported vocabularies, which are explained in the previous sections. For the explanation, we refer to Fig. 2, which shows the main ontology classes as well as the most important object properties and datatype properties. The stereotypes shown for some of the classes in Fig. 2 contain their superclasses and hence their alignment to the other ontologies described in the previous sections.

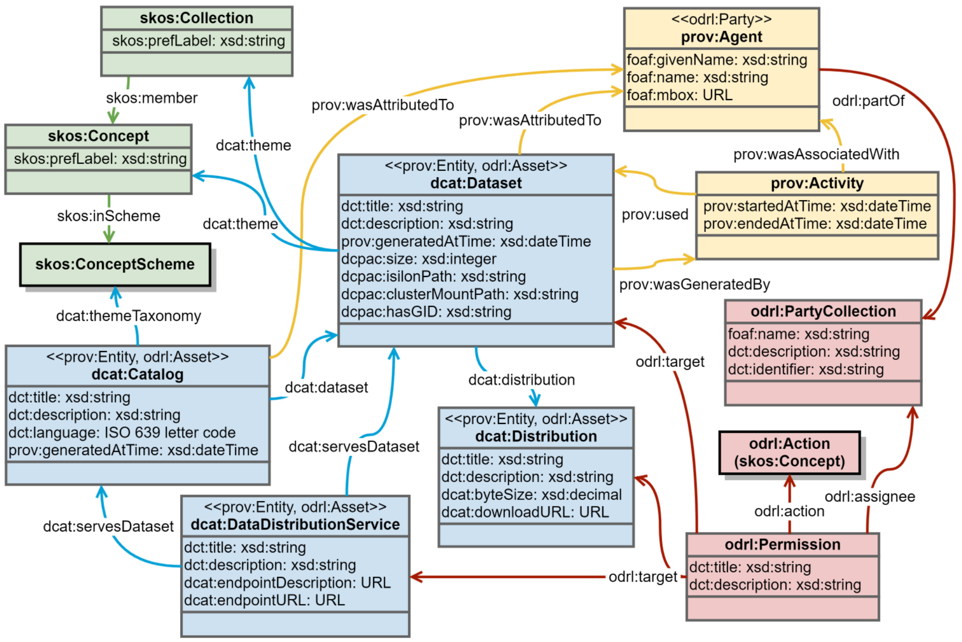


Fig. 2: The main classes and properties of the semantic data lake catalog ontology (TBox)

The *DCAT ontology classes* are shown in the center and bottom left of Fig. 2. The overall data catalog of the data lake is represented by one instance of class `dcat:Catalog`. It can contain many `dcat:Dataset` instances, one per resource in the data lake, e.g. raw data files, HBase or Hive tables, or RDF-based knowledge graphs. An instance of class `dcat:Distribution` models a specific representation of a dataset, comprising a specific serialization or schematic arrangement. Different distributions can exist for the same dataset, and are accessible via a URL (`dcat:downloadURL`). The data catalog and the datasets can each have several data distribution services (`dcat:DataDistributionService`), which are endpoints that provide access. They are accessible via an endpoint URL (`dcat:endpointURL`).

The *PROV-O classes and properties* shown in the top right part of Fig. 2 are used for modeling the provenance of the data catalog and its datasets (both declared as subclasses of `prov:Entity`), and for defining agents (e.g. person, software agent) they are attributed to (`prov:wasAttributedTo`) or that were involved in the activity of creating the dataset. Activities (`prov:Activity`) are initiated by agents (`prov:wasAssociatedWith`), create new datasets (`prov:wasGeneratedBy`), have an start and end time, and can use other datasets as input (`prov:used`).

Access control is defined by *classes and properties from the ODRL ontology*. An `odrl:Permission` can define an access rule for groups of agents (`odrl:Party-Collection`) to datasets, their distributions and/or data distribution services (`odrl:target`). The allowed actions (`odrl:Action`), such as display, read, modify, delete, are defined as `skos:Concept` and attached via `odrl:action`.

SKOS finally is applied for defining the semantics of the content of a dataset. Therefore, the catalog refers to one or more sets of SKOS concepts (`skos:ConceptScheme`) that can be used for semantically tagging datasets. The defined SKOS tags can be either directly linked to a dataset (`dcat:theme`), or they can be bundled and linked as a collection (`skos:Collection`), which enables the reuse of (large) sets of SKOS tags.

The overall ontology vocabulary accomplished by the DCPAC ontology constitutes a vocabulary developed and harmonized by a broad community and standardized in most of the parts. It enables an interoperable representation and exchange of information beyond the limits of a specific data lake in place. The vocabulary is widely domain independent, with the only exception being the SKOS tags ontologies, which make it customizable towards particular domains of interest.

4 Semantic data lake for automotive data

At Bosch, we have built a semantic data lake for our automotive data as a centralized platform for the engineering and testing of our autonomous driving applications. The architecture of our data lake is shown in Fig. 3. It stores large amounts of data collected from test drives, which involves the logging of hundreds of sensor readings that are being logged each millisecond, but also includes the vehicle configuration, driving maneuvers, weather conditions and other related information. This data quickly accumulates to Petabytes of data, which is stored in large Hadoop-based data stores.

During the ingestion and processing of new incoming data, the data lake catalog population service is triggered, which automatically creates a semantic description and layer on top of the data. The semantic layer is stored and managed as knowledge graph in the semantic data lake catalog using the vocabulary defined by the semantic data lake catalog ontology (see Section 3). Therefore, the data lake catalog population service reads the available metadata on the ingested data assets and constructs the relevant semantic data by aligning, annotating and enriching the input data with DCPAC concepts. The resulting knowledge graph forms a semantic layer on the data lake assets and describes their content, provenance and access rights. This information plays a crucial role in the semantic data lake, as it semantically describes the stored data and hence turns it into valuable information and information assets. It allows an in-depth tracking of provenance related aspects of the data, and a definition of access rights on the data assets.

The semantic data lake catalog is a key-component of the data lake architecture, as it handles and controls the retrieval and access of information. It enables a semantic search of data and improves findability. Data can be found faster, better and automatically even by machines, which enables advanced data analytics use cases. The overall data reuse is much higher, which ultimately results in less test drives and related effort and cost.

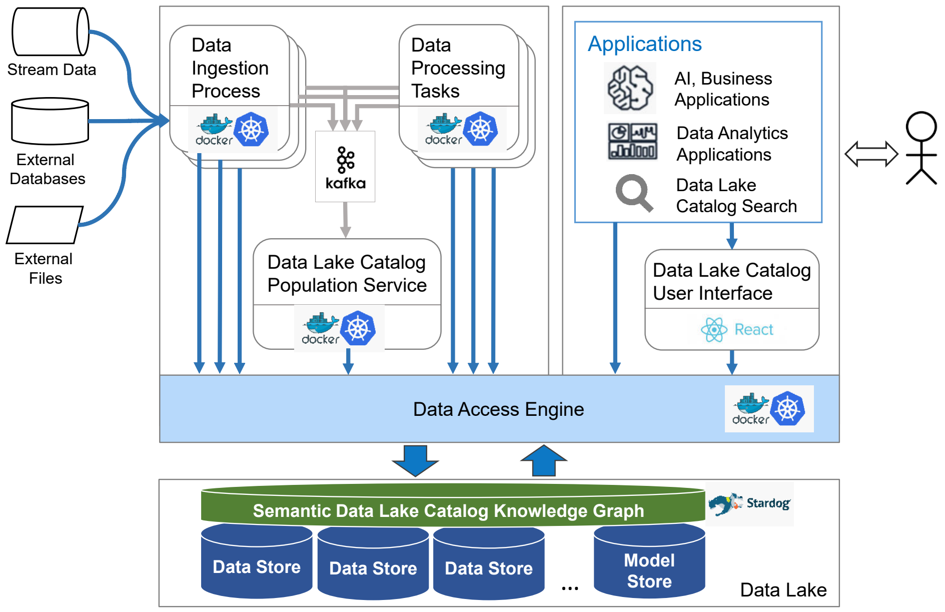
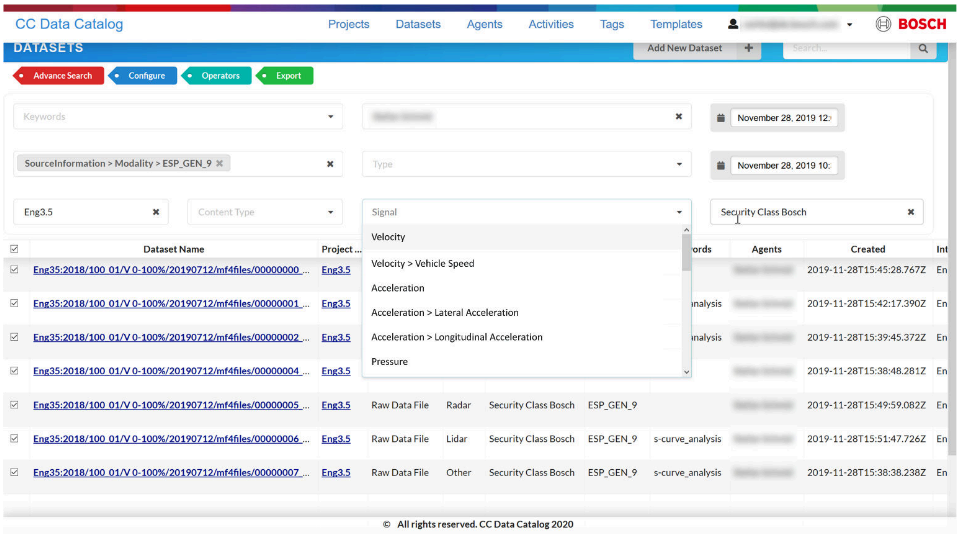


Fig. 3: Data lake architecture and role of semantic data lake catalog knowledge graph

5 Ontology-driven self-adaptive frontends

Now that more and more applications are using knowledge graphs as representation layer for data and information, new ontology-driven software architectures for self-adaptive frontends become feasible and are gaining momentum. This is doable due to knowledge graphs containing both the data as well as its rich semantically defined schema, which can describe domain knowledge, laws and constraints in a detailed way. Applications and frontends can utilize this knowledge, without the need of duplicating or repeating that inside their code. In an ideal scenario, business logic and expert knowledge can reside completely on the knowledge graph, and the applications and frontends can be independent of the model, domain or even use case.

As a first step into that direction, we have implemented self-adaptive frontends that can dynamically render information from a knowledge graph in a generic way. An example web frontend can be seen in Fig. 4, which shows information from the semantic data lake catalog of our semantic data lake (see Fig. 3). What information is to be shown where and how is defined by views on the knowledge graph. We followed the proposal of Tim Berners-Lee [Ti19] and used a combination of SHACL shapes and forms in order to define different views on a graph in a generic, expressive way. Dynamically at start up, the frontends request the view from the knowledge graph to be rendered. They receive all required information, such as the classes to be shown as tabs in the header and their labels in the required language, the properties to be shown for each class, their type, cardinalities, datatypes, range, supported languages, their ordering in the frontend, as well as the properties that can be used for filtering the instances, along with their filter operators and possible values. But also, who has read access or who can change, create or delete instances is controlled by the knowledge graph.



Dataset Name	Project ...					ords	Agents	Created	Int
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000000 ...	Eng3.5						[illegible]	2019-11-28T15:45:28.767Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000001 ...	Eng3.5					nalysis	[illegible]	2019-11-28T15:42:17.390Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000002 ...	Eng3.5					nalysis	[illegible]	2019-11-28T15:39:45.372Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000004 ...	Eng3.5						[illegible]	2019-11-28T15:38:48.281Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000005 ...	Eng3.5	Raw Data File	Radar	Security Class Bosch	ESP_GEN_9		[illegible]	2019-11-28T15:49:59.082Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000006 ...	Eng3.5	Raw Data File	Lidar	Security Class Bosch	ESP_GEN_9	s-curve_analysis	[illegible]	2019-11-28T15:51:47.726Z	En
Eng35:2018/100 01/V 0-100%/20190712/mf4files/00000007 ...	Eng3.5	Raw Data File	Other	Security Class Bosch	ESP_GEN_9	s-curve_analysis	[illegible]	2019-11-28T15:38:38.238Z	En

© All rights reserved. CC Data Catalog 2020

Fig. 4: Ontology-driven self-adaptive web frontend showing the semantic data lake catalog

Such ontology-driven self-adaptive frontends can dynamically render views on knowledge graphs. We believe that the initial overhead of realizing such next-generation ontology-driven software architectures and applications will pay off quickly, not only since changes of the underlying model do not require any changes at the frontend.

6 Conclusion

Semantic technologies and knowledge graphs are experiencing a high demand and are gaining strong momentum. The shift from ML and DL to symbolic approaches goes along with a fundamental paradigm shift: from the use of raw data to the use of knowledge in AI. The massive amounts of data in a data lake are only of value if the context of the data is semantically well described and its provenance is defined. We have realized a semantic data lake for automotive data, in which ontologies and knowledge graphs fulfil this role and are of utmost importance. Knowledge graphs enable us to search data based on rich semantics (context and provenance). Moreover, they also allow intelligent agents to automatically search and access the data, and thus, contribute to a significantly higher data reuse, which ultimately results in less test drives, and related efforts and costs.

With knowledge graphs as novel representation layer for data and information, new ontology-driven software architectures become feasible. We have implemented ontology-driven self-adaptive frontends that dynamically render information from a knowledge graph in a generic way. In an ideal scenario, both business logic and expert knowledge can reside completely in the knowledge graph, and the applications and frontends can be fully reused across domains and use cases. This will allow us to move from developing applications and frontends via coding (i.e. writing programs) towards modeling, where we define the views on the information and business logic in the knowledge graph itself.

Bibliography

[Al20] Albertoni, R.; Browning, D.; Cox, S., Beltran, A. G.; Perego, A.; Winstanley, P.: Data Catalog Vocabulary (DCAT) – Version 2. W3C Recommendation, https://www.w3.org/TR/vocab-dcat-2/, 2020, accessed: 14/10/2020.

[BM14] Brickley, D.; Miller, L.: FOAF Vocabulary Specification 0.99. http://xmlns.com/foaf/spec/, 2014, accessed: 14/10/2020.

[Br19] Brant, K.; Hare, J.; Sicular, S.: Hype Cycle for Artificial Intelligence. Gartner Research (ID: G00369840), 2019.

[Di20] Dibowski, H.; Schmid, S.; Svetashova, Y.; Henson, C.; Tran, Tuan: Using Semantic Technologies to Manage a Data Lake: Data Catalog, Provenance and Access Control. In: 13th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2020), Athens, Greece, 2020.

[Du20] Dublin Core Metadata Initiative Board: DCMI Metadata Terms. DCMI Recommendation, https://www.dublincore.org/specifications/dublin-core/dcmi-terms/, 2020.

[Ed17] Edjlali, R.; Duncan, A. D.; De Simoni, G.; Zaidi, E.: Data Catalogs Are the New Black in Data Management and Analytics. Gartner Research, 2017.

[Ia17] Iannella, R. et. al.: ODRL Version 2.2 Ontology. W3C, https://www.w3.org/ns/odrl/2/, 2017, accessed: 14/10/2020.

[KK17] Knublauch, H.; Kontokostas, D.: Shapes Constraint Language (SHACL). W3C Recommendation, https://www.w3.org/TR/shacl/, 2017, accessed: 14/10/2020.

[Le13] Lebo, T.; Sahoo, S.; McGuinmess, D.: PROV-O: The PROV ontology. W3C Recommendation, http://www.w3.org/TR/prov-o/, 2013, accessed: 14/10/2020.

[MB09] Miles, A.; Bechhofer, S.: SKOS Simple Knowledge Organization System Reference. W3C Recommendation, https://www.w3.org/TR/skos-reference/, 2009.

[Ti19] Berners-Lee, T.: Linked Data Shapes, Forms and Footprints. White paper, https://www.w3.org/DesignIssues/Footprints.html, 2019, accessed: 14/10/2020.

[WW19] W3C Dataset Exchange Working Group (DXWG): DCAT-PROV alignment ontology. W3C, https://github.com/w3c/dxwg/blob/gh-pages/dcat/rdf/dcat-prov.ttl, 2019.