Linked Data Schema Repositories for Interoperable Data Spaces

Stier, S. (Speaker)[1]*; Räder, A.[1]; Gold, L.[1]; Popp, M.[1]; Triol, A.[1]

[1] Department Digital Transformation, Fraunhofer ISC, Würzburg

*simon.stier@isc.fraunhofer.de

In this position paper we discuss the improved interoperability and enlarged tool landscape of data spaces by stronger coupling between semantic web, general web technologies, and programming languages. We propose hierarchical self-description of data assets by JSON-LD annotated JSON-SCHEMAs in combination with direct coupling with object-oriented programming through a linked data schema repository. Overall, this approach combines the hierarchical concept of GAIA-X with the graph structure of IDS and provides an interactive schema catalogue along search catalogue for detail or domain levels beyond their core metadata models.

In engineering and science, complex relationships exist between the properties of objects, their composition and processing defined by a variety of heterogenous data sources. Therefore, digital transformation and acceleration represents a particularly big challenge in this domain. Although it is generally agreed that data must be linked by means of semantics and ontologies to form holistic data spaces, there is still a lack of suitable tools for integrating the necessary structures into the everyday work of engineers, programmers and scientists.

This challenge must be addressed with a broad-based strategy that closely links activities at all relevant levels, including automated data acquisition infrastructure, machine-readable specification and documentation of research, and production processes. This also involves the harmonization of the generated data structures in accordance with the data spaces.

For this purpose, both IDS and GAIA-X propose[1,2] RDF and it's JSON-LD serialization as a common data representation, and consequently RDFS, OWL, and SHACL for data schemas and validation. While the expressiveness of RDF is theoretically unlimited and thus fulfills all requirements, there are practical limitation and barriers:

- While SHACL[3] as a schema language for RDF graphs exists, it requires the full dataset to be mapped to such a graph. This requires the existence of full ontology coverage and leads potentially to very large and complex graphs if mapping on demand is not possible.
- Graph shaped schemas with recursive constraints may lead to long runtimes
- Object-oriented programming languages have no build-in support of open data models and typically require classes to be known at compile / interpretation time

[1] https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Position-Paper-GAIA-X-and-IDS.pdf
[2] https://library.oapen.org/handle/20.500.12657/57901
[3] https://www.w3.org/TR/shacl/

We therefore propose the closer integration between RDF/SHACL and object orientated programming. Since JSON as serialization is already widely agreed, JSON-SCHEMA is a suitable broad-established candidate for a language and implementation independent object model serialization. In addition to data validators, a large number of code and UI generators exists[4].

By now, JSON-SCHEMA covers similar to SHACL: property presence, data types, nested structures (arrays, sub-objects) and hierarchy/inheritance. Especially the hierarchical view on information models reflects well the intuitive conventionalization of humans. To make JSON-SCHEMAs and generated JSON data ready for linked data, it is sufficient to couple them with a JSON-LD context. The context provides a mapping of properties and their values to vocabulary term and the JSON-LD flattening-algorithm[5] allows transforming of any nested object structure to a flat graph and consequently RDF triples and vice-versa. The major limitation is that in both JSON-SCHEMA and object-oriented programming, at some point an object reference must be replaced by a symbolic reference such as an IRI to avoid creating a full world model. While this marks domain or use case specific break downs of a large graph, JSON-SCHEMA currently lacks in contrast to SHACL the expressiveness to put constrains, e. g. on the type of the entity targeted by the symbolic reference (typical reflected in an UI by an autocomplete field). However, we are convinced that it is worthwhile addressing these gaps to include the established and intuitive object model perspective in linked data spaces (cf. Fig. 1).

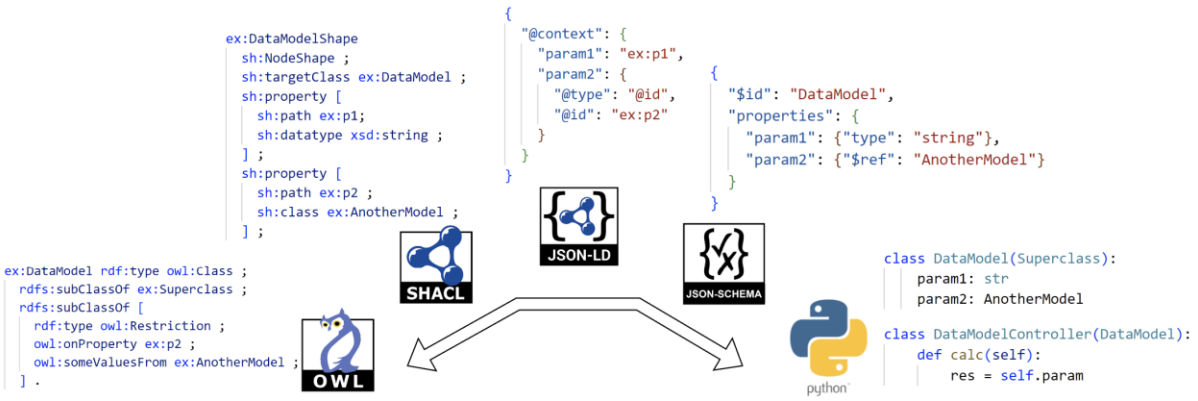

Figure 1. Coupling / Transformation between RDF-based linked data concepts focusing on expressiveness (left) and object-oriented models focusing on implementation (right). As the example *DataModel* shows there's significant overlap between the languages, but finally implementation layers like the example *DataModelController* must be simple to provide a fast-growing ecosystem of linked services within data spaces.

Therefore, we develop with the OpenSemanticWorld (OSW) / OpenSemanticLab (OSL)[6] stack a reference toolset for the domain-specific extension of data spaces to close this gap. Part of the resulting open source solution is document store based on Semantic MediaWiki. The document store provides multiple versioned content slots[7] per entity allowing to store JSON data (all entities) along with JSON-LD annotated JSON-SCHEMAs (class entities) as well as markdown documentations and rendering templates (all entities) with multiple processing routes (see Fig. 2).

[4] https://json-schema.org/implementations.html
[5] https://www.w3.org/TR/json-ld11-api/#flattening-algorithm
[6] https://github.com/OpenSemanticLab
[7] https://opensemantic.world/wiki/Item:OSWdb485a954a88465287b341d2897a84d6#OpenSemanticWorld_Concept

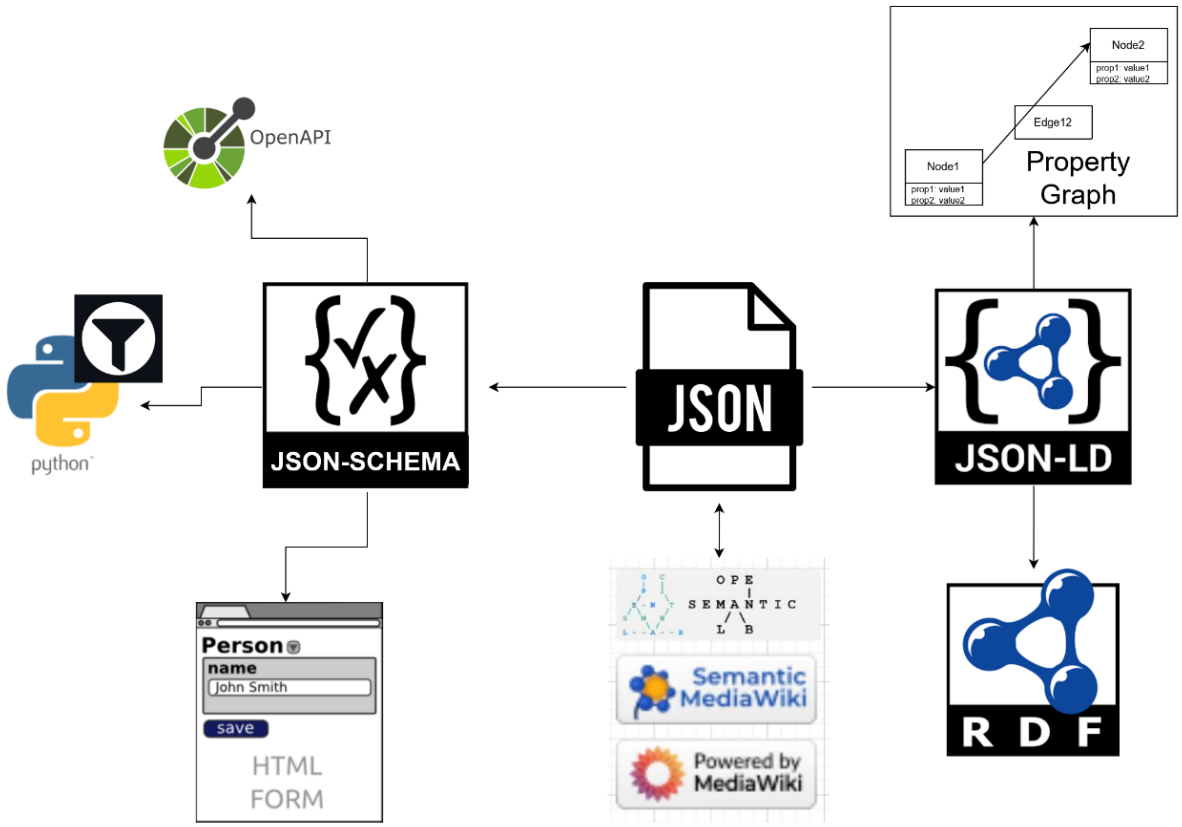

Figure 2. The OpenSemanticLab Software Stack utilizes Semantic MediaWiki as an interactive document store for both unstructured (wikitext) and unstructured data (JSON). JSON data is shaped by a JSON-SCHEMA graph, allowing visual/form editing, API and code generation. JSON-LD context embedded in the schemas allows a mapping to a built-in RDF-store and prepares for usage of property graphs (in combination with JSON-LD-STAR/RDF-STAR) in the future.

A search index, visual renderings and auto-generated edit forms provide a human-centered interface. A published and maintained python package[8] allows the add-hock generation[9] of data classes from data schemas and vice versa, enabling conformance by design of programmatically generated and used data through an abstract knowledge graph interface (cf. Fig. 3). This includes not only pure data classes, but also mapped controller classes that provide corresponding functions/methods with a strong link to data space services/apps via automatically generated APIs.

[8] https://github.com/OpenSemanticLab/osw-python
[9] https://opensemantic.world/wiki/Item:OSW659a81662ff44af1b2b6febeee7c3a25

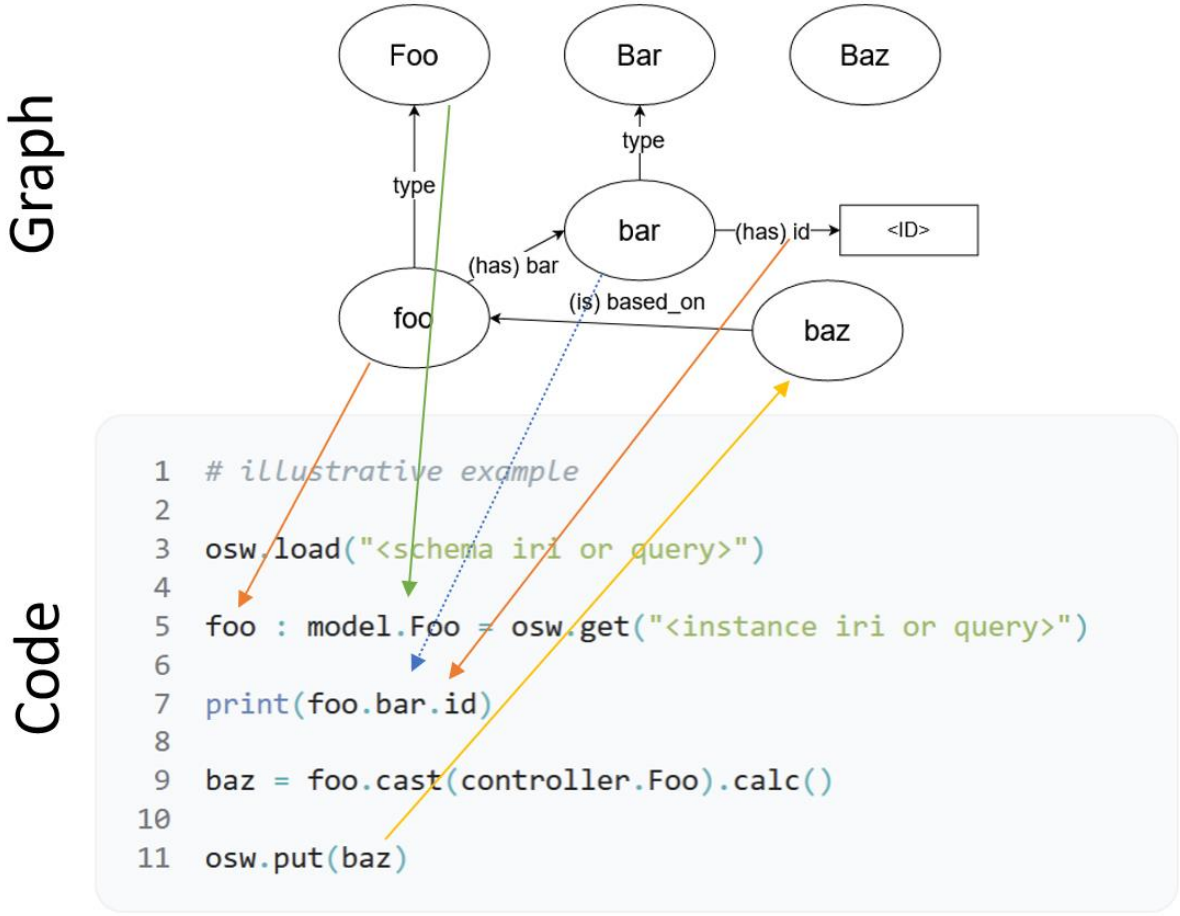

Figure 3. Illustrative example how the osw-python package provides an abstract knowledge graph (KG) interface. First (line 3) primary schemas (*Foo*) and their dependencies (*Bar, Baz*) are loaded from the KG and transformed into python dataclasses. Instantiation of *foo* is handled by loading the respective JSON(-LD) document from the KG and utilizing the type relation to the corresponding schema and dataclass (line 5). Because *bar* is not a dependent subobject of *foo* it is loaded on-demand on first access of the corresponding class attribute of *foo* (*foo.bar* in line 7), while *id* as dependent literal is loaded immediately in the same operation. In line 9 *baz* is constructed by an existing controller class subclassing *Foo* and finally stored as a new entity in the KG in line 11.

An interactive schema and vocabular repository[10] and a demo instance already demonstrate use cases for science in general. By providing a JSON-SCHEMA for OWL classes the documentation, visualization and graphical editing of large and linked ontologies also becomes an important use case[11] for both their development and governance. Thereby OWL classes and data schemas are regarded as complementary and linked concept combining their strength in formal logic on the one hand and closeness to concrete implementations on the other hand. With respect to the existing reference architectures this approach integrates the hierarchical concept of GAIA-X with the graph structure of IDS and provides a user and programmer friendly toolset to extend their application to detail/domain levels beyond the core metadata models[12,13].

[10] https://opensemantic.world
[11] https://onto-wiki.eu/wiki
[12] https://gaia-x.gitlab.io/gaia-x-community/gaia-x-self-descriptions/core/core.html
[13] https://github.com/International-Data-Spaces-Association/InformationModel

Simon Peter Stier received his bachelor's degrees in Aerospace Computer Science and Technology of Functional Materials in 2014 and the consecutive master's degrees in Computer Science and Functional Materials in 2017 at the JMU Würzburg. In 2021 he finished his PhD in the field of interdisciplinary research on materials, processes and sensor applications at the Graduate School for Science and Technology Würzburg. Currently he is Head of the Department for Digital Transformation at the Fraunhofer ISC Würzburg and Consultant for the European Lithium Institute Brussels. Dr. Stier current research interests are the development of ontologies for scientific and industry processes and embedding of semantic technologies is application-ready software stacks. He is a member of the European Materials Modelling Council (EMMC) and Co-Lead of the Batteries European Partnership Association Digitalization Taskforce.