More than two years ago, we first published our **dblp Knowledge Graph** as an dblp RDF dump file. We have since been working on expanding and updating our RDF schema, as well as on adding new semantic relations to the graph. Today, we release our first major extension to the dblp KG: We added publication venues (e.g., journals and conference series) as first-class entities to the graph.

Introducing *dblp:Stream*

Until recently, the dblp KG had been mainly an "almost bipartite" graph of publications (*dblp:Publication*) and their creators (*dblp:Creator*). At some point we introduced further reification entities to model more context, but the publication-creator network remained at the core of the graph. With the new *dblp:Stream* class we now introduce a third kind of core entity designed to model any kind of "publication stream" (aka a "publication venue"), i.e., a regular, common source where *dblp:Publication* entities are published.

The interrelation of all dblp KG entities is sketched in the following diagram. Also, we now have a proper ontology reference documentation, which can be accessed at https://dblp.org/rdf/docu/.
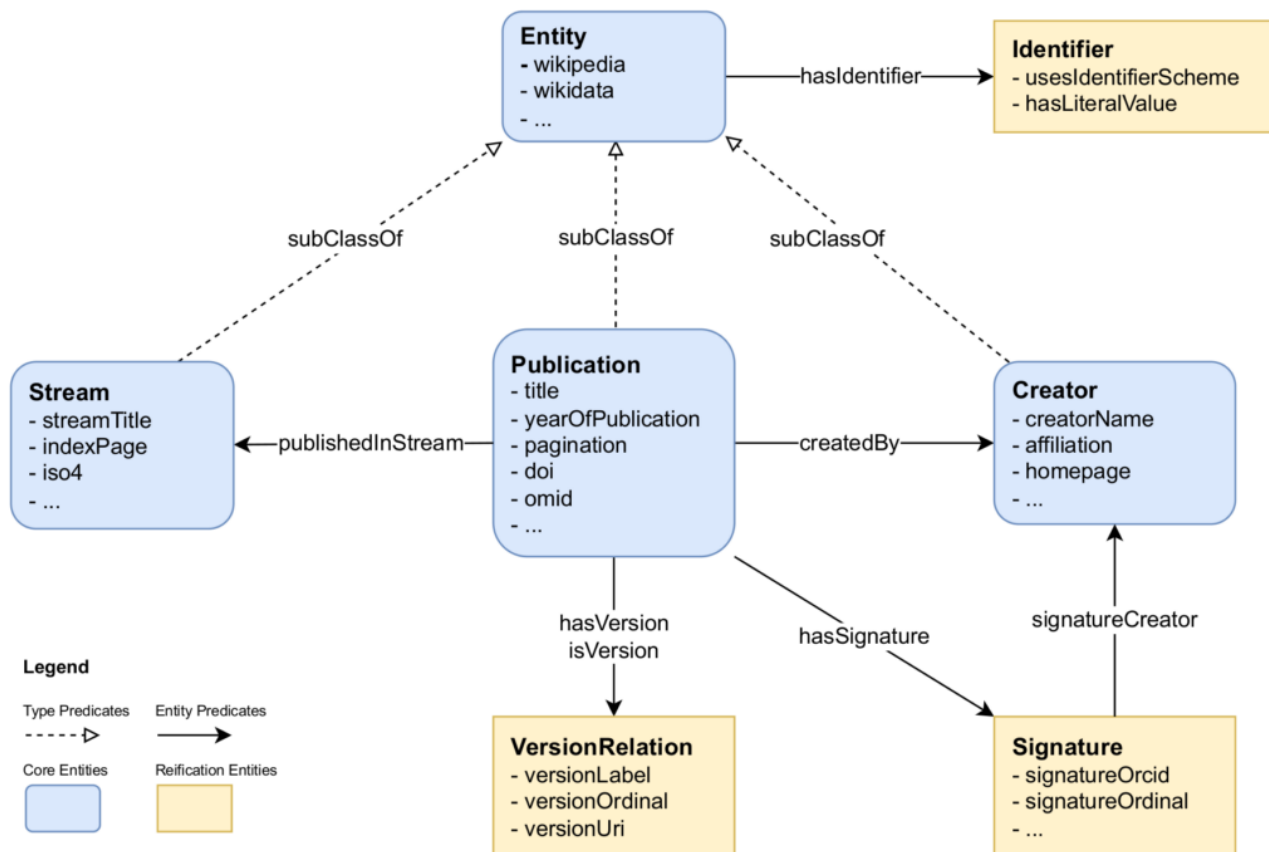
Sketch of the dblp ontology, showing selected relationships between core and reification entities.

The *dblp:Stream* entities come in four different concrete classes that mirror the structure we use at dblp: *dblp:Conference* for conference and workshop series, *dblp:Journal* for periodic journals, *dblp:Series* for a periodic series of published book volumes, and *dblp:Repository* for sources of data publications. These entities provide metadata properties for stream titles (*dblp:streamTitle*, *dblp:primaryStreamTitle*, *dblp:formerStreamTitle*), identifiers (*dblp:issn*, *dblp:iso4*), and link to the stream's index webpage on the dblp website (*dblp:indexPage*).

The *dblp:publishedInStream* property links a *dblp:Publication* to the dblp:Stream entity it has been published in. A *dblp:Publication* may be linked to multiple streams. To give a few examples, a conference paper may be linked both to the *dblp:Conference* that accepted the paper as well as the *dblp:Series* its conference proceedings has been published in. Other conferences publish their proceedings in certain *dblp:Journal* streams that host multiple *dblp:Conference* entities. Another example is papers that

may be published in a joint proceedings volume for two different *dblp:Conference* streams and cannot be reasonably be associated with just one of the two.

To model known relationships between *dblp:Stream* entities, we added the *dblp:relatedStream* property. We also added more specialized sub-properties for common cases of relations. We added *dblp:superStream* and *dblp:subStream* to state vertical "part-of" relationships (e.g., between conferences and their satellite workshops) and *dblp:predeccessorStream* and *dblp:successorStream* for temporal relationships (e.g., journals that merged or a conference series that fully replaced another one). In case of any general, unique or unspecified relation, we usually just use the *dblp:relatedStream* property. Please note that modelling the concrete stream relations is an ongoing process at dblp. That is, there may still be many inconsistencies in the data, but we actively try to improve.

Stream entity data from the live API

All *dblp:Stream* entities are referenced by resource IRIs starting with *https://dblp.org/streams/*. RDF fragments of individual entities can be retrieved via the live API in the formats RDF/XML, N-Triples and Turtle. The API can be used to query for streams using their unique dblp stream key. E.g., if you want to retrieve data for the International Semantic Web Conference (dblp key: *streams/conf/semweb*), you may call

- https://dblp.org/streams/conf/semweb.rdf for retrieving RDF/XML,
- https://dblp.org/streams/conf/semweb.nt for retrieving N-Triples,
- https://dblp.org/streams/conf/semweb.ttl for retrieving Turtle,
- https://dblp.org/streams/conf/semweb.xml for retrieving the dblp XML record, or
- https://dblp.org/streams/conf/semweb.html for accessing the dblp HTML website.

Additionally, queries may be directed to any resource IRI (in the case above: https://dblp.org/streams/conf/semweb) via content negotiation using the recognized MIME types for the formats listed above.

Please note that there is at least some rate limiting in place to protect the live API from aggressive crawlers. That means, in case of a rather large number of queries, it is

recommended to download the complete dump and do your queries locally.

Work in progress ... and a new dblp service?!

Building upon the current iteration of the dblp KG and expanding its capabilities is an ongoing endeavor of the dblp team. The dblp KG aims to (eventually) provide all of dblp's curated semantic information in a machine-actionable format. This includes structured information already available via APIs as well as information that has not been published explicitly before.
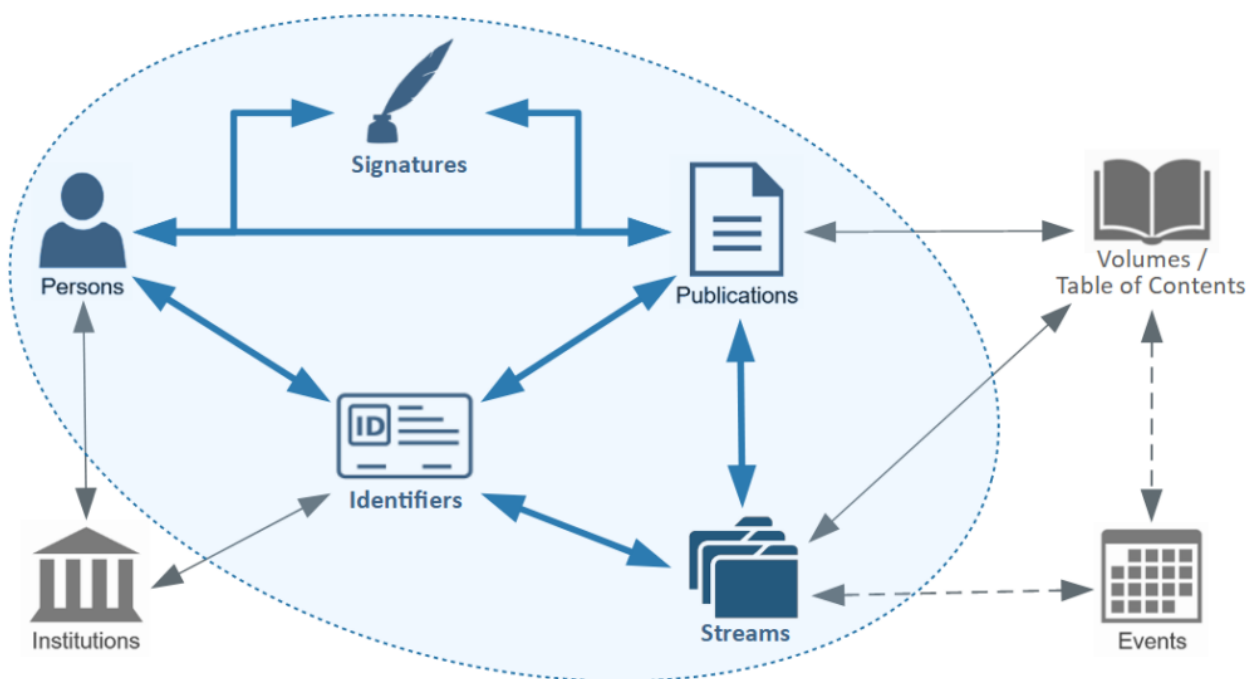
Planned extensions of the dblp Knowledge Graph.

A particular priority is the further utilization of the only weakly structured semantic information listed on the dblp website, as well as making it machine-actionable. The immediate next steps ahead are already clear: In the current DFG-LIS project "SmartER affiliations", the dblp team is intensifying its coverage of author affiliation information.

Future iterations of the dblp KG will expand its model to add institution entities as first-class citizens to the graph and link affiliation information for authors and signatures. Also, the event history of conference and workshop series, together with metadata about the time and date of events, is contained aplenty in the dblp webpages and will be added to the dblp KG.

Having dblp data semantically structured and ready for download so that researchers can make use of it is great. The first research articles using the dblp KG have already been published [1] [2] [3] and we hope many will follow. But with all this semantic data available, the question arises as to how everyday users of the dblp website can make use of the dblp Knowledge Graph. Well, since you are asking, we have actually been working (for quite some time now) on some new service for the dblp community that is almost ready for release. We will be ready to talk about it soon … pretty soon, actually. If you want to know more, you may want to keep an eye on this blog.

Further updates to the RDF schema

In addition to the changes mentioned above, some further minor changes have been made to the dblp RDF schema. A list is given below.

Added owl:Class to all *rdfs:Class*

In order to provide compatibility with OWL2 as well as backward compatibility with earlier standards, classes are now redundantly defined as both *owl:Class* and *rdfs:Class*.

Added *dblp:omid* property

This new property has been added in order to link a *dblp:Publication* to their OpenCitations Meta Identifier (OMID). This ID is crucial for making use of open citation and reference data from OpenCitations.net.

Added *dblp:VersionRelation* class and related properties

A new mechanism for "versions of publications" has been added to dblp. The new *dblp:VersionRelation* entity represents a link between a general, abstract publication concept with its instanced concrete versions. Each version relation is linked to exactly

one abstract (concept) *dblp:Publication* and one specific (instance) *dblp:Publication*, and may list further metadata of the link, like a (mandatory) version label, a (mandatory) ascending sorting ordinal among all concrete versions, and an (optional) URI identifying the linked specific (instance) version. The *dblp:hasVersion* and *dblp:isVersion* properties link a version concept instance *dblp:Publication* to their *dblp:VersionRelation* entities.

Added *dblp:isVersionOf* property

This is a redundant convenience property that directly links between dblp:Publication entities, without any of the additional metadata provided in the *dblp:VersionRelation* entities. You may use this property in queries if you only care whether or not a given version relation exists between two publications.

Removed old deprecated properties completely from schema

We took the liberty to remove all old *owl:DeprecatedProperty* entries from the schema file since they have been out of use for at least two years.