DataHives: Swarm-Based Triple Store Enhancement

Pepijn Kroes Wouter Beek [a] Stefan Schlobach [a]

[a] *Vrije Universiteit Amsterdam, 1081HV Amsterdam*

Abstract

The Linked Open Data (LOD) cloud is too big for efficient computation and too heterogeneous for standard materialization techniques to cope with. The purpose of the DataHives system is to solve both of these problems by utilizing swarm intelligence to enhance a curated dataset. The system spawns software agents that traverse the LOD cloud looking for extensions to the curated dataset that are relevant and trusted.

1 Purpose

The Linked Open Data (LOD) cloud is too big to handle queries that are both complex and exhaustive. Moreover, the LOD cloud contains contradictory information.

DataHives provides solutions to both problems. For the first problem, instead of answering queries over the entire LOD cloud, DataHives only answers queries over a locally enriched dataset. The enrichment consist of those triples that are *relevant* for the local dataset. Relevance is defined in terms schema-matching. An RDF triple is a candidate for local dataset enrichment if it contains an RDF term that occurs in the schema of the local dataset (i.e., its classes and properties).

The schema matches are established by scout agents that traverse the entire LOD graph. The scouts start from the local dataset and spread out across the LOD cloud, using existing RDF links (defined in VoID) between those datasets. Once a schema match is established, a group of forager agents is sent straight to this location to perform a more intensive graph traversal in the region. The triples that result from this localized traversal effort are sent back to the curated dataset (i.e., the 'hive').

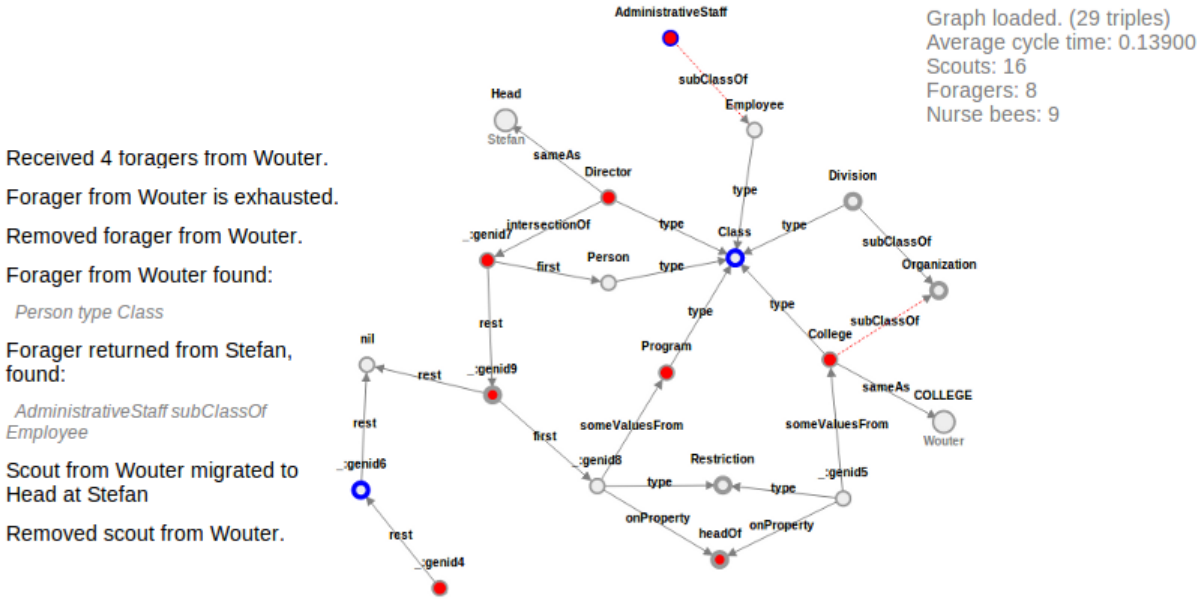
To solve the second problem, the chance that contradictory information is introduced in the curated dataset is lessened by only enriching it with triples that come from *trusted* sources. Trust is defined in terms of the physical graph structure of the LOD cloud. DataHives assumes that a data provider's trust structure is a partial ordering on the collection of LOD sources, that is based on the links that are defined between those sources. Deduction performed in the local graph does not introduce contradictions of its own, since reasoning takes place at the level of RDF(S).

2 User groups

The first intended user group of DataHives consists of data publishers in the LOD cloud. This comprises a reasonably big and continuously growing number of institutions and companies. By setting up DataHives, a data publisher will automatically enrich her curated database with triples that are relevant and trusted. Uptake of the DataHives system is made easy by the system not requiring any changes to existing datasets and by the fact that its enrichments are stored separately from the original dataset.

The second intended user group of DataHives consists of small organizations, institutions, and maybe individuals who want to maintain their own data but do not have the time, money, or proficiency to enrich the

Figure 1: An example of the in-broser graph representation of DataHives. Scouts (red) and foragers (blue) are sent from the hive of data owner *Wouter* to the dataset that is owned by *Stefan*, who is one of his most trusted (i.e, directly linked to) sources.



data themselves. Such organizations may have data they would like to disseminate, but the data itself, taken in isolation, is not interesting enough for data consumers. Demand from this user group is currently very small. However, we expect this demand to rise once data enrichment becomes cheap and fully automated.

3 Practical Information

Project context & Developers
DataHives was developed by Pepijn Kroes, Wouter Beek, and Stefan Schlobach within the context of the Pragmatic Semantics (PraSem) research project at the KR&R group of the Vrije Universiteit Amsterdam.

Technology used
DataHives is built using open and standards-compliant technology exclusively. The Web based user interface is built in JavaScript and HTML5. The communication of agents and data is established by following the new WebRTC standard for between-browser communication. For data format support the W3C standards for triple representation and serialization are used (e.g., N-Triples). For calculating the local deductive closure, the standards-compliant semantics for RDF and RDF Schema are used.

System requirements
One of the characteristic features of DataHives is its low entry level. The only system requirement for DataHives is a recent Web browser (e.g., Google Chrome 12+, Firefox 22+). The configuration of the system consists of having the triple dataset available in a file with read access.

Duration
The duration of the demo is arbitrary. The process of data enrichment takes place continuously. The agents can be visually traced as they move across the data graph. The system has anytime behavior, i.e., there are always some results, but the quality of the results becomes better over time.