# Towards Semantically Structuring `GitHub`

Dennis Oliver Kubitza[1,2], Matthias Böckmann[1,2], and Damien Graux[2,3]

[1] University of Bonn – Germany
[2] Enterprise Information Systems, Fraunhofer IAIS – Germany
[3] ADAPT Centre, Trinity College of Dublin – Ireland
dennis.oliver.kubitza|matthias.boeckmann|damien.graux@iais.fraunhofer.de

**Abstract.** With the recent increase of open-source projects, tools have emerged to enable developers collaborating. Among these, `git` has received lots of attention and various on-line platforms have been created around this tool, hosting millions of projects. Recently, some of these platforms opened APIs to allow users questioning their public databases of open-source projects. Despite of the common protocol core, there are for now no common structures someone could use to link those sources of information. To tackle this, we propose the SemanGit ontology, the first ontology dedicated to the `git` protocol, which also describes *GitHub*'s features to show how it is extensible to encompass more `git`-based data sources.

## 1 Introduction

Open-Source technology is, once accepted as beneficial, subject to improvement attempts under the premises of commercial marketing, ideological beliefs or just feasibility of implementation. One of the most popular examples is Linux, with its variety of ecosystems, ranging from commercial server distribution to open-source desktop implementations. While not considered as heterogeneous in its implementations, for now, the `git` protocol [10] faces the same development.

Developed in 2005 as a distributed version-control system, `git` [10] is tracking changes in a file system while providing several properties such as data integrity or support for distributed and non-linear workflows. Since the file system represented by `git` can be distributed, developers embed their changes into a local `git` repository and later "push" their contributions to an online repository so that collaborators can then have access to their modifications and contributions. Quickly, the `git` protocol has evolved to provide more and more features dedicated to large open-source communities and projects. In recent years, more and more platforms emerged using the `git` protocol to provide their users with a version-control system, with every one of them using additional features to provide a better user experience, faster distribution or improved maintainability.

Nowadays, *GitHub* seems to be the most popular platform [1] for the external usage of `git`, but other providers like *GitLab* achieve more and more popularity [1,2]. As all these systems share a common infrastructure, based on `git` and principles adopted from social networks, it is possible to agglomerate the different approaches in a unified model and to merge these providers' data into a

common source of information. In this study, we present the SemanGit ontology, an OWL ontology serving three goals. First, it structures data provided by the `git` protocol. Second, it models as much information as possible from *GitHub*. Third, it contains enough abstraction such that similar information from other sources like *BitBucket, CloudForge, GitLab, Launchpad* can have a consistent representation, using the same parental classes.
The SemanGit ontology can then be used to collect data about developers from different domains, interlinking information about publicly available software, its developers and interactions between them with social media and research platforms. While we propose not the first ontology related to data from `git` [6], our approach is a novelty, focusing on the possibilities of automated data extraction and analysis. Indeed *De Nies et al.* proposed the `git2PROV` system, tailored to the task of extracting provenance information [6] as a feature for developers publishing their work.

More generally, we fall within the domain of structuring public data (see e.g. [7] focusing on describing projects). So far, numerous projects are already providing such datasets with each one tackling a distinct domain. Among this list, we can mention DBpedia [3] which proposes a linked version of Wikipedia, or also LinkedGeoData [4] which deals with geographical data. Both are excellent targets for a future interlinking of data extracted from *git* hosters, to enrich the information about users origin.

## 2 A `git` Ontology and its `GitHub` Extension

The `git` protocol relies on so-called repositories for storing files and tracking data modifications. A large share of online `git` repository providers add features of their own that are not part of the `git` protocol, such as social features. To create an extensible ontology, we need to implement a strict distinction between what is part of the `git` protocol and what is provider specific. As an example, according to the `git` protocol, the author of a commit is a pair "Name <email>" whereas on *GitHub* an author, i.e. a user, is much more complex. It has additional attributes such as a creation date, an avatar, a location and even social-featured ones such as e.g. an associated website. The part of the SemanGit ontology covering the `git` protocol features only the data that strictly belongs to the protocol. The classes in this section mostly form the basis from which platform-specific classes inherit, see Figure 1 for an example. This protocol-related part is rather small and comprises of merely four classes: users, projects (i.e. repositories), commits and pull requests, the user class storing only an email address. The projects refer to a URL, a time stamp of its creation and the commits that were submitted to it. The other two classes are slightly more complex as commits have a hierarchical structure in themselves and pull requests are requests to accept a cross-branch commit, possibly coming from another project. Seeing that all extensions of the `git` protocol are still required to provide the base functionality, we have chosen a hierarchical approach for the SemanGit ontology, letting extensions inherit from protocol-conform classes all properties they are expected

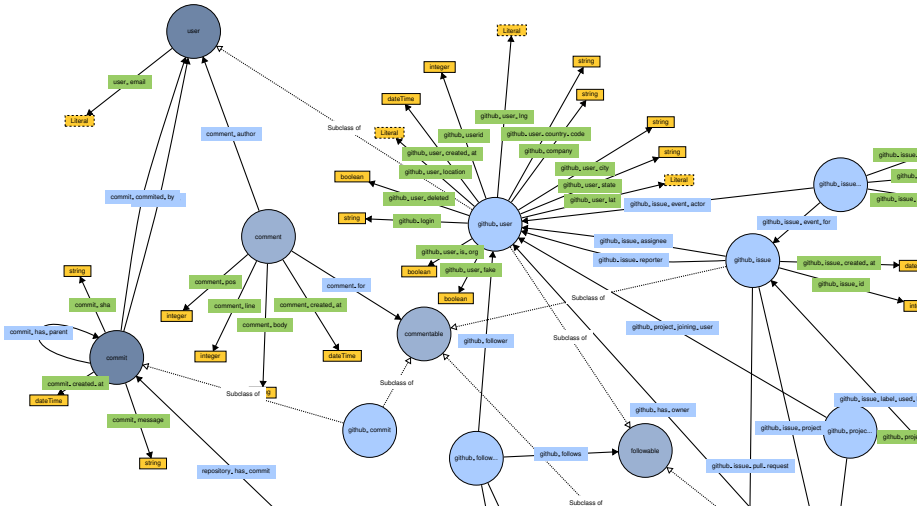


**Fig. 1.** An excerpt of the SemanGit ontology. In this example the layers of the ontology are represented in three shades of blue, representing from dark to light, the `git`, the abstraction and the *GitHub* specific classes.

to have. The SemanGit ontology comprises three different and distinct layers of abstraction:

1. A set of classes corresponding to entities and relations directly specified by the `git` protocol, containing the semantic representation of all information delivered by the execution of this protocol.
2. An intermediate layer abstracting any common functionality provided by different implementations of the protocol, forming the link between the `git` standards and the providers' systems. Some of these functionalities are concerning purely social relations, such as one user following another, or multiple users forming an organization.
3. Provider specific classes that usually derive from the upper layers as subclasses and contain unique implementation features for the respective `git` implementations. We set apart these classes corresponding to provider-specific extensions of the protocol from the original one by adding a prefix e.g. "github_" to the class name and their properties.

While for now being tailored to model information provided by *GitHub*, this structure leaves space for any extension by other providers' specific implementations and even encourages representations of project migration from one provider to other alternatives.

## 3 Conclusion

In this study, we presented the SemanGit ontology, an ontology dedicated to the `git` protocol. In addition, we also described how it can be extended to en-

compass additional features from public open-source platforms by considering *GitHub*. The full SemanGit ontology is publicly available for further community driven development on *GitHub*[1] and on our website[2]. An interactive visualization can be found on *VisualDataWeb*[3]. While focusing on the structure and features of *GitHub* for the moment, we designed our ontology to be extensible by the information generated by any other host of `git` based version-control systems. We created this ontology as a starting point to build semantic datasets from various collaborative-platforms. Moreover, we already built one from *GitHub*: the SemanGit dataset [8]. Such datasets could then allow innovative perspectives for data analysis if one considers an enrichment of data from other sources like `DBpedia` [3], `LinkedIn` [9] or linking researchers code development with `ScienceGRAPH` [5]. While the SemanGit ontology is currently published at our homepage, we strive for the integration on external hubs such as LOV [11] in the near future.

## References

1. Comparison of source code hosting facilities. `https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities`, accessed: August 28, 2019
2. Gitlab gains developers after microsoft buys rival github. `https://www.reuters.com/article/us-github-microsoft-gitlab/gitlab-gains-developers-after-microsoft-buys-rival-github-idUSKCN1J12BR`, accessed: August 28, 2019
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: Semantic Web, pp. 722–735. Springer (2007)
4. Auer, S., Lehmann, J., Hellmann, S.: Linkedgeodata: Adding a spatial dimension to the web of data. In: ISWC. pp. 731–746. Springer (2009)
5. Auer, S.: Towards an open research knowledge graph (Jan 2018), `https://doi.org/10.5281/zenodo.1157185`
6. De Nies, T., Magliacane, S., Verborgh, R., Coppens, S., Groth, P.T., Mannens, E., Van De Walle, R.: Git2prov: Exposing version control system content as w3c prov. In: ISWC (Posters & Demos). pp. 125–128 (2013)
7. Dumbill, E.: Doap: Description of a project. http://trac.usefulinc.com/doap (2010)
8. Kubitza, D.O., Böckmann, M., Graux, D.: SemanGit: A linked dataset from git. In: Proceedings of 18th International Semantic Web Conference (2019)
9. Li, J., Wade, V., Sah, M.: Developing knowledge models of social media: A case study on linkedin. Open Journal of Semantic Web (OJSW) **1**(2), 1–24 (2014)
10. Torvalds, L., Hamano, J.: Git: Fast version control system. http://git-scm.com (2010)
11. Vandenbussche, P.Y., Atemezing, G.A., Poveda-Villalón, M., Vatant, B.: Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web. Semantic Web **8**(3), 437–452 (2017)

[1] `https://github.com/SemanGit/SemanGit/blob/master/Documentation/ontology/`

[2] `http://www.semangit.de/`

[3] `http://visualdataweb.de/webvowl/#opts=doc=0;editorMode=true;#iri=https://raw.githubusercontent.com/SemanGit/SemanGit/master/Documentation/ontology/semangitontology.ttl`