The Whyis Knowledge Graph Framework in Action

James P. McCusker, Sabbir M. Rashid,
Nkechinyere Agu, Kristin P. Bennett, and Deborah L. McGuinness

Rensselaer Polytechnic Institute, Troy, NY 12180, USA

1 Introduction

We will demonstrate a reusable framework for developing knowledge graphs that supports general, open-ended development of knowledge curation, interaction, and inference. Knowledge graphs need to be easily maintainable and usable in sometimes complex application settings. Often, scaling knowledge graph updates can require developing a knowledge curation pipeline that either replaces the graph wholesale whenever updates are made, or requires detailed tracking of knowledge provenance across multiple data sources.

Fig. 1 shows how Whyis provides a semantic analysis ecosystem: an environment that supports research and development of semantic analytics for which we previously had to build custom applications [3,4]. Users interact through a suite of knowledge graph views driven by the node type and view requested in the URL. Knowledge curation methods include Semantic ETL, external linked data mapping,and Natural Language Processing (NLP). Autonomous inference agents expand the available knowledge using traditional deductive reasoning as well as inductive methods that can include predictive models, statistical reasoners, and machine learning. Whyis is used in a number of areas today, including nanopolymers, spectrum policy, and health informatics. We demonstrate Whyis by creating and deploying an example Biological Knowledge Graph (BioKG), using data from DrugBank and Uniprot[1], and briefly discuss benefits of using our approach over a conventional knowledge graph pipeline.

2 Architecture

Whyis uses *nanopublications* to encapsulate every piece of knowledge introduced into the knowledge graphs it manages. A nanopublication is composed of three named RDF graphs: *Assertion*, *Provenance*, and *Publication Info* [2]. We see knowledge graphs with the level of granularity supported by nanopublications as essential to fine-grained management of knowledge graphs that are curated and inferred from diverse sources and can change on an ongoing basis. The use of nanopublications as a fundamental unit of knowledge in Whyis has enabled the systematic inclusion of provenance in ways that support knowledge revision

[1] http://drugbank.ca, http://uniprot.org, respectively

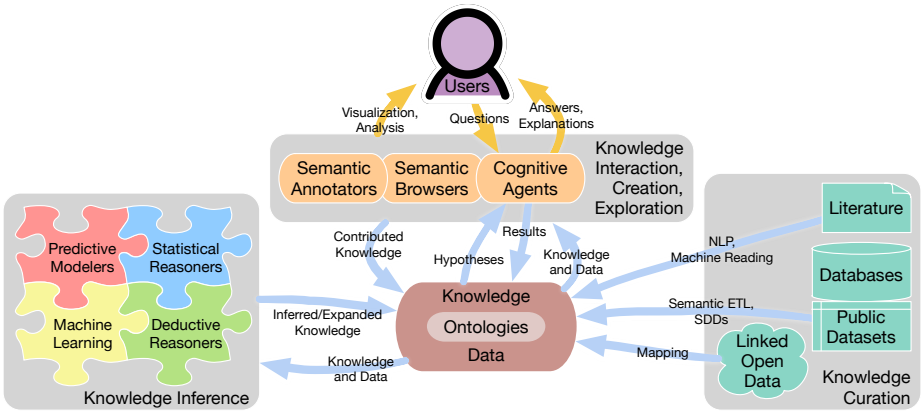

Fig. 1. The semantic ecosystem enabled by the Whyis framework for knowledge curation, interaction, and inference.

and truth maintenance of inferred knowledge as underlying knowledge changes. Whyis is written in Python using the Flask framework, and uses a number of existing infrastructure tools to work, as shown in Fig. 2.

Whyis inference is handled by a suite of "Agents", each performing as the analogue to a single rule in traditional deductive inferencing. An agent is composed of a SPARQL query that serves as a "body" and a python function that serves has the "head". The agent is invoked when new nanopublications are added to the knowledge graph that match the SPARQL query defined by the agent. The agent superclass assigns some basic provenance related to the given inference activity, which developers can customize in their implementations. Included inference agent types include entity extraction and resolution against existing knowledge graph nodes, deductive reasoning agents that can be configured with custom rules, as well as many available pre-configured OWL 2 rules.

3 Related Work

Some existing frameworks support some of Whyis' capabilities. Stardog[2] includes OWL reasoning, mapping of data silos into RDF, and custom rules. Ontowiki provides a user interface on top of an RDF database that tracks history, allows users to browse and edit knowledge, and supports user interface extensions [3]. Callimachus, a "Semantic Content Manager," lets developers create UIs by object type using RDFa [1]. Virtuoso Openlink Data Spaces is a linked data publishing tool that provides a set of pre-defined data import tools and a fixed set of views

[2] A case study: https://www.stardog.com/blog/nasas-knowledge-graph/
[3] http://ontowiki.net

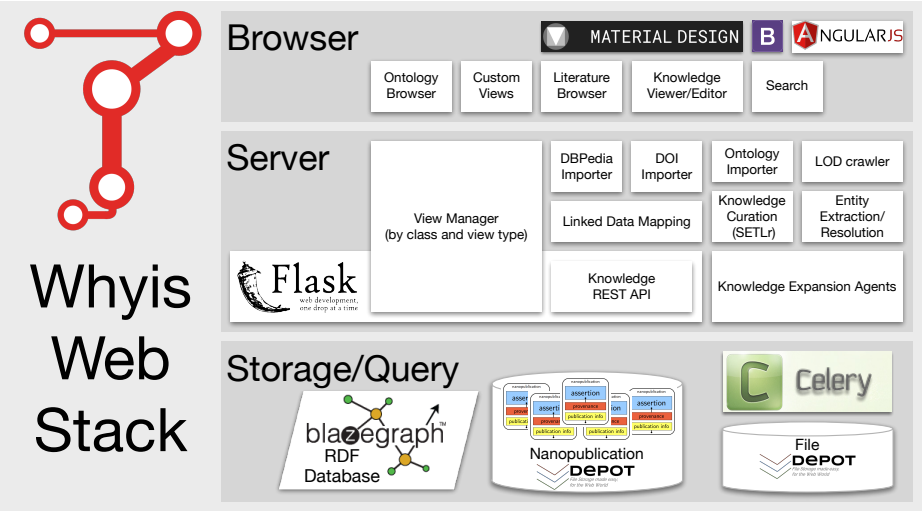

Fig. 2. The Whyis technology stack. Nanopublications are stored in the RDF database, while the entire history is stored in the nanopublication file archive using File Depot. Celery invokes and manages autonomic inference agents by listening for graph changes.

on the linked data it creates.[4] Vitro[5] supports the creation of new ontology classes and instances, but does not allow users to create custom interfaces.

4 Demonstration

We demonstrate Whyis using our Biology knowledge graph at http://bit.ly/whyis-demo. All user views are built-in views in Whyis. Nothing has been customized for the biology domain except for queries to find biological interactions. The BioKG main page allows users to view knowledge graph along with the most recent changes and the graph neighborhood of the most recently changed entity. Users can search for entities and either view search results or select from one of the resolved entities. Every entity in the knowledge graph gets its own page, which can be customized by knowledge graph developers by the entity type. Users can also explore the knowledge graph beyond the current node using the knowledge explorer (Figure 3), a refinement of the user interface developed in [3].

5 Conclusions

We believe Whyis is the first provenance-aware framework for knowledge graph development that enables curation, interaction, and inference within a unified

[4] https://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/Ods
[5] Available: https://github.com/vivo-project/Vitro

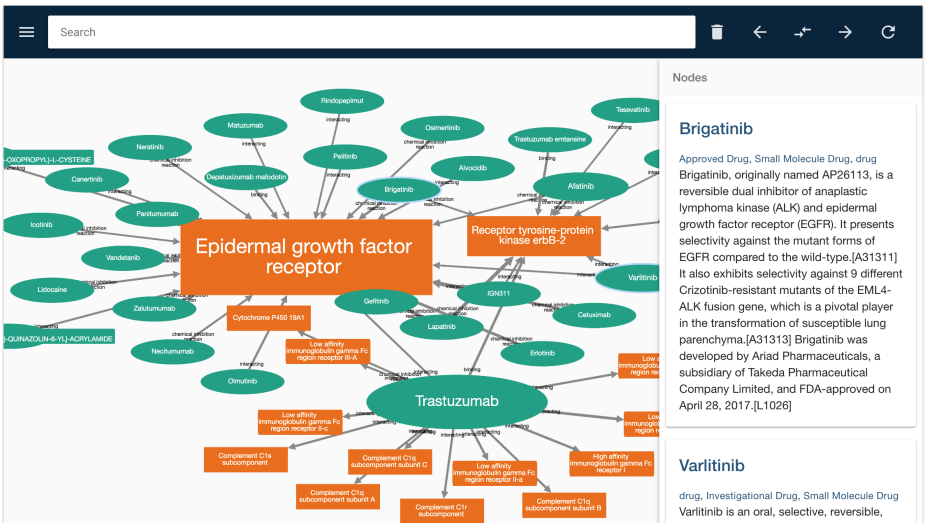

Fig. 3. The knowledge explorer lets users expand the current view by expanding node connections (upper right) or searching for new entities (search box in upper left).

ecosystem. We demonstrate these features in a BioKG setting, exploring drug-protein-disease interactions, and providing semi-automated support for semantic queries previously custom developed [3]. Whyis is published under the Apache 2.0 License on Github[6] with documentation on how to develop custom knowledge graphs.

Acknowledgements: This work was funded by NIEHS Award 0255-0236-4609 / 1U2CES026555-01, NSF Award OAC-1640840 IBM Research AI Horizons Network, and by the Gates Foundation through HBGDki.

References

1. Battle, S., Wood, D., Leigh, J., Ruth, L.: The callimachus project: Rdfa as a web template language. In: Proceedings of the Third International Conference on Consuming Linked Data-Volume 905. pp. 1–14. CEUR-WS. org (2012)
2. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Information Services and Use 30(1), 51–56 (2010), http://dx.doi.org/10.3233/ISU-2010-0613
3. McCusker, J.P., Dumontier, M., Yan, R., He, S., Dordick, J.S., McGuinness, D.L.: Finding melanoma drugs through a probabilistic knowledge graph. PeerJ Computer Science 3, e106 (Feb 2017), https://doi.org/10.7717/peerj-cs.106
4. McGuinness, D.L., Bennett, K.: Integrating semantics and numerics: Case study on enhancing genomic and disease data using linked data technologies. Proceedings of SmartData pp. 18–20 (2015)

[6] https://tetherless-world.github.io/whyis