InPhO for All: Why APIs Matter

Jaimie Murdock, Colin Allen
Indiana University

What is the InPhO?

The Indiana Philosophy Ontology Project (InPhO — `http://inpho.cogs.indiana.edu`) is a dynamic ontology which models philosophy by using the editors and contributors to its leading digital reference work, the Stanford Encyclopedia of Philosophy (SEP — `http://plato.stanford.edu`) as a source of expertise (Buckner et al. 2010). The SEP contains nearly 1,200 entries and over 15 million words, maintained by more than 1,600 volunteer authors and editors, and is accessed through more than 1,000,000 entry downloads per week.

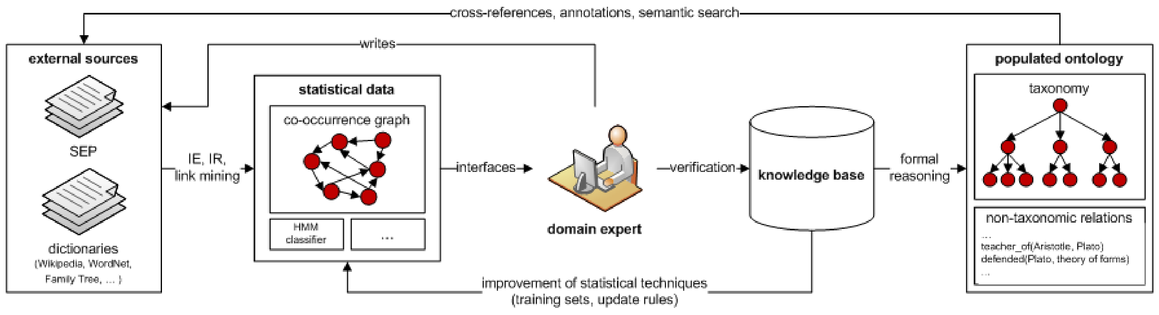

Figure: The InPhO Workflow

Our approach utilizes a three-step process to create a taxonomic representation of philosophy (see Figure). First, data mining of the SEP and statistical inference are used to generate hypotheses about the relations among various topics (Niepert et al. 2007). These hypotheses are then evaluated by domain experts (Niepert et al. 2009). This verification is combined with the statistical measures as a knowledge base for a machine reasoning program, which uses answer set programming to output a taxonomic view of the discipline (Niepert et al. 2008).

The InPhO Data Model

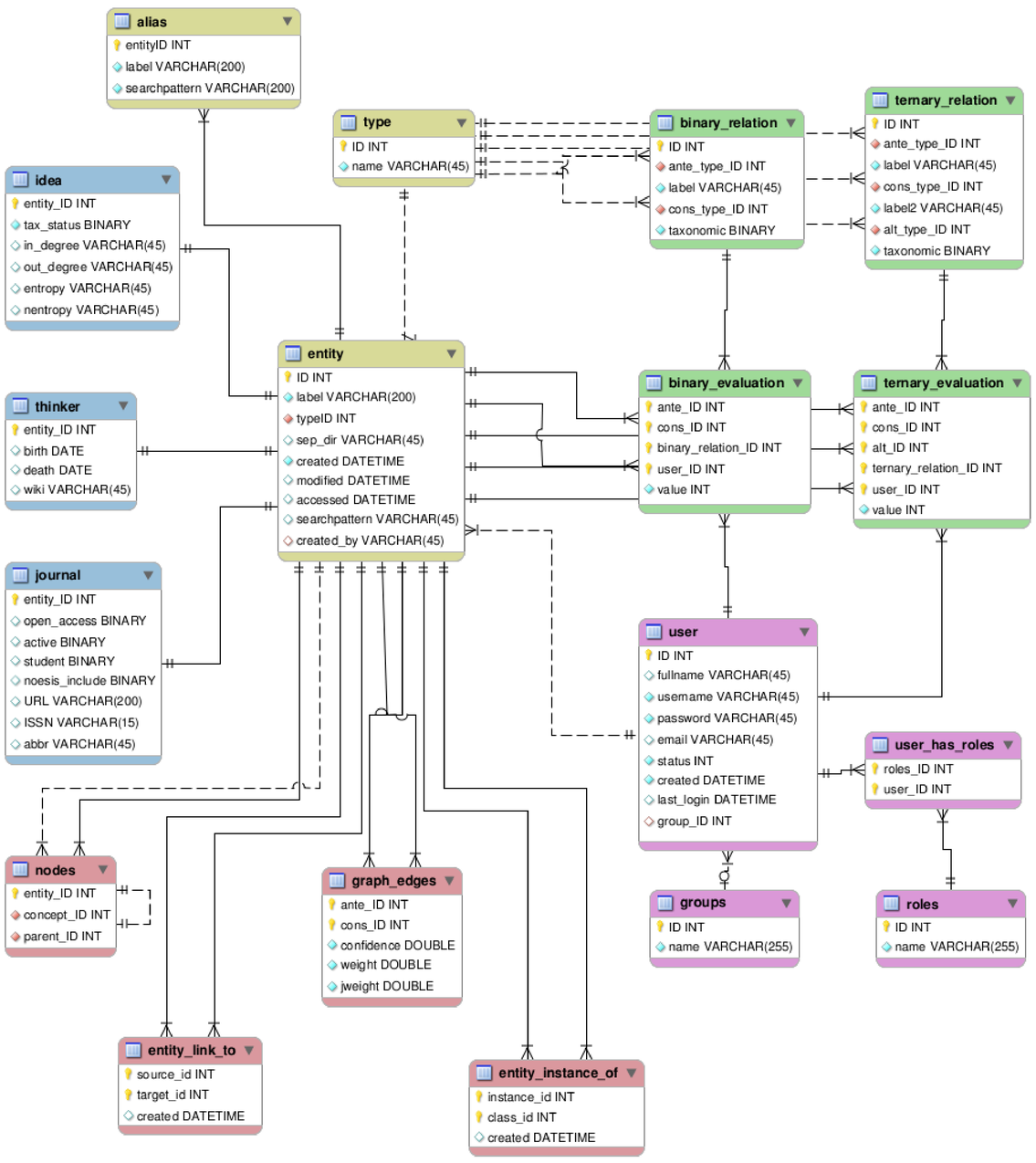

Figure: Database Schema

Facilitating Collaboration

InPhO frequently receives inquiries about collaboration, but our ability to respond to this interest was constrained by the lack of a standard interface to our data. We have exposed some of this data through an industry-standard Web Ontology Language (OWL) file, that is simultaneously providing more and less data than many projects needed. For instance, our data mining algorithms look for nearly 5,000 unique terms. As of May 2010, only 1,000 of these terms have received enough human evaluation to be asserted as instances by the machine reasoning program. Therefore, the OWL file excludes data about these philosophical concepts for which we have statistical information, which may still be useful for other applications, such as latent dirichlet analysis or other clustering techniques.

For example, although we have entered a strategic partnership with the Noesis project (`http://noesis.evansville.edu`) to power their domain-specific search engine, our OWL file did not afford them the ability to leverage and contribute to our journal database. Furthermore, the Noesis project currently has no need to receive our entire ontology file when seeking to query specific pieces of information in our journal database. Other projects may have similar requirements – a project wishing to trace the history of specific philosophers would only need data from our thinker database.

Why APIs Matter

The unique marriage of humanities scholars, computer programmers and information scientists in digital humanities projects highlights the collaborative opportunities such research entails. Digital humanities scholars need access to the data in many representations: from HTML for the ordinary end user, to fully integrated XML specifications and raw data dumps for the information scientist, to lightweight JSON stores for the web programmer. Unfortunately, limited human resources in these projects have led to unwieldy initial implementations and underutilization of semantic web technology which can provide easy access to peferred representations (Crane et al. 2007).

The InPhO aims to overcome this barrier by providing a simple, lightweight API (application programming interface) capable of serving a wide variety of data formats. APIs allow programmers to focus on the what of computing rather than the how. So, for instance, it is an API that allows programmers to tell your computer's operating system to respond to a mouse click by opening a "window" on the screen, without the programmer having to worry about the graphics needed to produce a rectange of a certain size, border, color, etc. Similarly, programmers can exploit databases on another server through an API without having to know what the underlying database model is on the remote server. APIs give power to programmers by allowing them to stand on the shoulders of others.

Humanities scholars who aren't programmers should care about the ad hoc nature of application integration, because so much of our time is used manually transferring what we learn in one context to what we do in another. For example, in the realm of digital philsophy, a bibliographic citation in PhilPapers doesn't tell you which entries in the Stanford Encyclopedia of Philosophy (SEP) refer to it, and the search process requires manual cut and paste from one to another.

The InPhO API

To address these concerns in a consistent, easy-to-use interface for both humanities scholars and web developers, we adopted teh REpresentational State Transfer (REST) web servcies paradigm (Fielding 2000, Richardson and Rubt 2007). REST capitalizes on one of the most ubiquitous standards – the HTTP protocol – to provide several benefits for end users and developers. As an ontology captures representations of various entities, the translation to a resource-oriented architecture (ROA) should be seamless.

URI	GET	PUT	POST	DELETE
/idea	list	–	create	–
/idea/{id}	view	update	–	delete
/idea/{id}/{idea-read-relation}	view	–	–	–
/idea/{id}/{idea-write-relation}/{id2}	–	update	create	delete
/thinker	list	–	create	–
/thinker/{id}	view	update	–	delete
/thinker/{id}/{thinker-relation}/{id2}	–	update	create	–
/journal	list	–	create	–
/journal/{id}	view	update	–	delete
/taxonomy	list	–	–	–
/taxonomy/{id}	view	–	–	–

Table: API Structure. {idea-read-relation} may be any of: related, hyponyms, evaluated, first_order. {idea-write-relation} may be any of: relatedness, generality. {thinker-relation} may be any of: student_of, teacher_of, influenced, influenced_by.

Implementation Details

The service is implemented through the Pylons web framework utilizing the SQLAlchemy object relational mapper (ORM). The selection of a Python-based infrastructure was driven by a desire for future integration with the Natural Language Toolkit (NLTK) to improve our data minining techniques. The growing popularity of Python as an instructional language was another factor, as we plan for new teams of student developers.

Usage

The API has already seen adoption by sister projects in the six months of public release. The SEP has ported their InPhO-driven cross-reference engine to utilize API calls rather than direct SQL connections. This engine executes with equivalent speeds. Additionally, we have begun work on integrating the SEP authentication system with the InPhO system to enable author feedback from within the editorial process. This benefits both projects – by providing the InPhO with high quality data, the SEP receives improved cross-reference suggestions. Additionally, the Noesis philosophy search engine is utilizing our journals database to assist in targeted query building.

Acknowledgements

Research presented in this poster was funded through the National Endowment for the Humanities award HD-50203, principal investigator Colin Allen.