3D Visualization of Application Ontology Class Hierarchies

Damion DOOLEY[a,1] and William HSIAO[a,b,c]
[a] *Department of Pathology and Laboratory Medicine, UBC, Vancouver, Canada*
[b] *Department of Molecular Biology and Biochemistry, SFU, Burnaby, Canada*
[c] *BCCDC Public Health Laboratory, Vancouver, Canada*

Abstract. An application ontology often reuses terms from other related, compatible, upper-level or domain-specific ontologies. The extent of this interconnectedness is not readily apparent when browsing through larger textual presentations of term class hierarchies, be it Manchester text format OWL files or as presented in an ontology editor like Stanford Protégé, where one either mentally notes the location or frequency of ontology prefixes in term identifiers as the encompassing ontology is browsed, or one selects an ontology import file to view individually, out of context of the whole. Interconnectedness may be easier to perceive in two-dimensional hierarchical graphs that visually code ontology term origins, but canvass size and multiple inheritance links that break tree layouts become challenging at scale. We present OntoTrek, a visualization tool that explores the benefits of interactive three-dimensional class hierarchy presentation. Our aim is to develop features, such as a consistent visual shape for ontologies based on the upper level ontology they subscribe to, that enable data project stakeholders to more quickly learn and appreciate the content domains of imported terms, ultimately illustrating how projects can describe knowledge through a vocabulary of interwoven community-supported ontology resources.

Keywords. visualization, ontology dependency, hierarchic navigation, education

1. Introduction

Data project stakeholders –software developers, curation teams, and funders – have been attracted to aspects of ontology-driven data modelling that would appear to solve data interoperability issues. The reuse of expert-curated domain-specific vocabularies, the validation of term use within an upper level framework, and global access to ontology terms via lookup services all encourage data harmonization. Visualization tools that enable exploring domains of interest along these lines should help in investigating the structure of current upper-level-ontology-compatible application and reference ontologies. There are a vast number of visualization approaches for hierarchic information as indicated in the https://treevis.net/ catalogue, and a lesser number of efforts dedicated to ontology visualization[1][2], of which the authors state only two projects, OntoSphere[3], and OntoSELF[4] are 3D visualizers, and which are earlier

[1] Damion Dooley, Department of Pathology and Laboratory Medicine, University of British Columbia, BCCDC Site, 655 West 12th Avenue, Vancouver, BC, Canada; E-mail: damion.dooley@bccdc.ca. Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

efforts that appear not to be supported or in development. OntoSphere provides network, tree, neighborhood, and hyperbolic views, each of which places more or less emphasis on highlighting class-subclass hierarchy, or object and data property axiomatization within a local or broader context. OntoSELF provides a desktop application relying on the Visualization Toolkit (https://vtk.org/) conical graph layout display to render a three dimensional graph with automatically placed nodes, and subsequent OntoSELF+TQ[5] which uses the same framework to display a custom TQ query language's results. Two and three-dimensional efforts (for example, WebVOWL[6] and OntoSphere) use links, symbols and colour codes to represent nodes, object properties, data properties and axiomatic expressions in order to capture the full semantics of interrelated ontology terms. Such presentations could potentially be visually filtered and augmented to highlight ontology term origin by colour coding.

Our main criticism, also noted by Dudáš et al [2], is that existing approaches lack a consistent layout, either relying on each user to place nodes, or in the case of force-directed-graph node placement, yielding shifting layouts over time or on each view. A more resilient approach is called the 3D "Botanical Tree" algorithm[7], illustrated in Figure 1. It builds branch width according to a size metric of branch content, and concludes with "phiballs", spheres that are decorated with either a conical cap, representing a single graph leaf element, or polka dots representing several leaf elements emanating from a final branch juncture. Such trees, guided by the class structure of an upper-level ontology, could provide a consistent layout steered by branch bifurcation and leaf volume. Our work is a variation on this approach, in which the leaf and node structure is exposed in order to provide labelling, and other functionality described below.

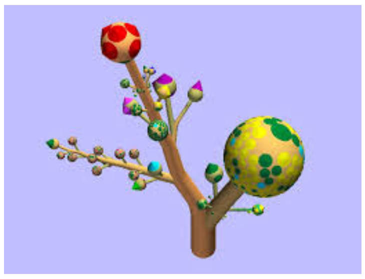


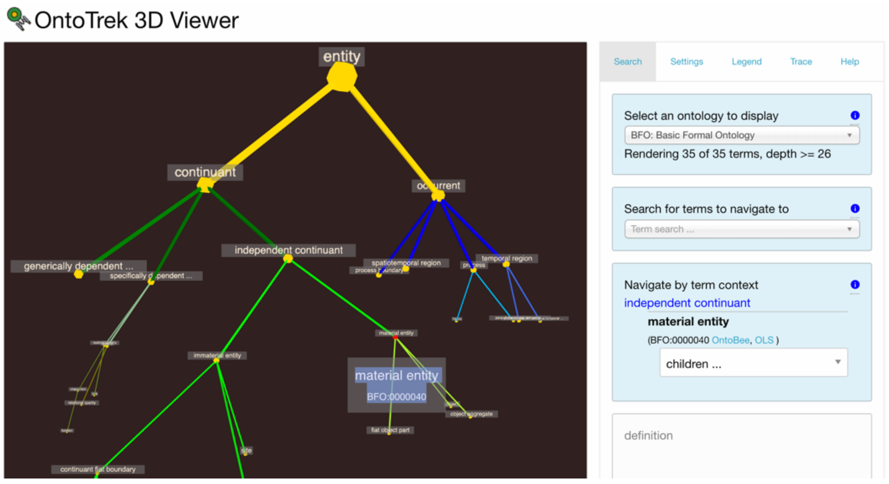
Figure 1: Botanical Tree algorithm display of hierarchic graph structure

2. Approach

Hsiao Lab, associated with the University of British Columbia Department of Pathology and Laboratory Medicine, and with the British Columbia Centre for Disease Control Public Health Laboratory, has created OntoTrek (https://genepio.org/ontotrek), a lightweight javascript-based open source web browser application that visualizes the scope of ontology integration to stakeholders. It explores the idea that humans benefit from data representations that maintain spatial consistency across successive presentations and incremental content changes. This translates to enabling users to virtually fly around and through the OntoTrek three-dimensional viewport's representation of a given ontology, which often resembles a jellyfish with tentacle structures dangling downward.

OntoTrek takes input from a user-selected JSON-LD file containing a given ontology's term class hierarchy, description and synonyms which are encoded using the Gene Ontology hasSynonym, hasExactSynonym, hasBroadSynonym and hasNarrowSynonym annotations. The ontofetch.py script generates the JSON-LD file by fetching an ontology from a local file or via URL. OntoTrek uses WebGL 3D graph rendering software (https://github.com/vasturiano/3d-force-graph/), which provides a suite of graph node and edge rendering features, along with user interface interactivity, to enable a 100% browser driven display of this content.

OntoTrek displays all class-subclass relations starting from user-defined root entities such as owl:Thing or the Basic Formal Ontology[**8**] (BFO) root "entity". Upper-level BFO terms are assigned fixed (pinned) locations, and underlying nodes are iteratively positioned by depth, leading the force directed graph to consistently place them in the same region relative to each other on each fresh generation of the visualization. The algorithm builds mountains of terms from the top-down, with lower-tier nodes pulling away from each other to help reduce density.



Select an ontology to display

BFO: Basic Formal Ontology

Rendering 35 of 35 terms, depth >= 26

Search for terms to navigate to

Term search ...

Navigate by term context

independent continuant

material entity

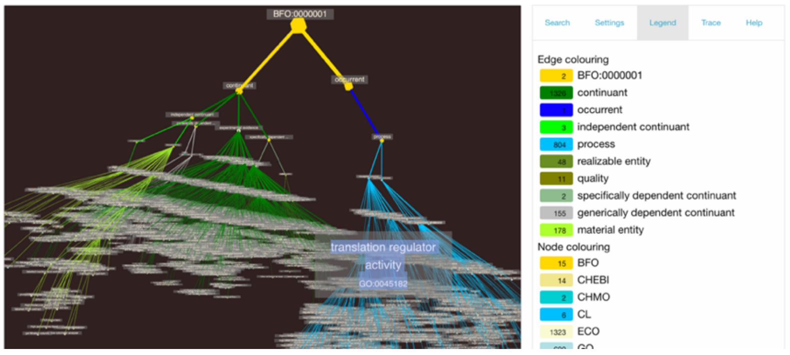
(BFO:0000040 OntoBee, OLS)

children ...

definition

Figure 2: OntoTrek's display of the upper level Basic Formal Ontology in which all 34 term nodes are fixed in position such that force directed graph algorithm doesn't affect them, but only influences the positioning of underlying ontology terms.

All ontology term nodes – imported or native to a given ontology - are colour-coded using a lookup table so their colouration is constant regardless of the umbrella ontology they are being included in. On the Settings tab, a "Colour edges by" setting allows a user to select from two edge color schemes. Figure 2 shows the "BFO branch" scheme at work, generating edge colours representative of their parent nodes, obtained from an



Edge colouring

🟨	2	BFO:0000001
🟩	1326	continuant
🟦	1	occurrent
🟩	3	independent continuant
🟦	804	process
🟩	48	realizable entity
🟫	11	quality
🟩	2	specifically dependent continuant
⬜	155	generically dependent continuant
🟩	178	material entity

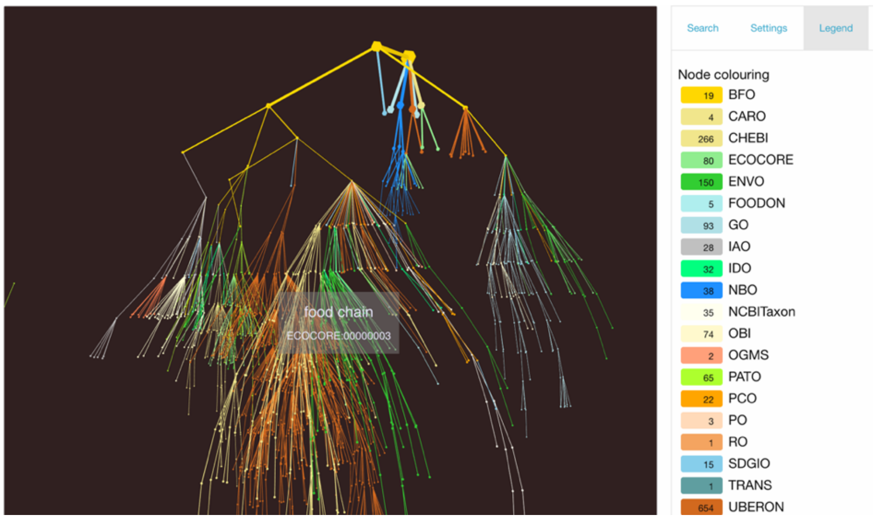
Node colouring

🟨	15	BFO
🟨	14	CHEBI
🟦	2	CHMO
🟦	6	CL
🟨	1323	ECO
🟦	[illegible]	GO

Figure 3: Legend of edge colouring by upper level node colour.

upper level ontology lookup table, and which all descendent edges inherit. Figure 3 shows this scheme applied to the Evidence and Conclusion Ontology (ECO). The alternative "source term" scheme shown in Figure 4 colours each edge according to its source node, thus re-enforcing the visual presence of that ontology. Both colour schemes are detailed on the application's Legend tab.

The OntoTrek viewport enables full mouse/trackpad pan/zoom/roll fly-through navigation of this hierarchy. Nodes of a given depth are provided in a correspondingly deep horizontal plane. Upper level ontology terms are given a larger size, enabling them



Node colouring

19	BFO
4	CARO
266	CHEBI
80	ECOCORE
150	ENVO
5	FOODON
93	GO
28	IAO
32	IDO
38	NBO
35	NCBITaxon
74	OBI
2	OGMS
65	PATO
22	PCO
3	PO
1	RO
15	SDGIO
1	TRANS
654	UBERON

Figure 4: Distant OntoTrek 3D view of the Ontology of Core Ecological Entities (ECOCORE). Coloring nodes according to "Term source" setting to emphasize imported ontologies, and with label display turned off.

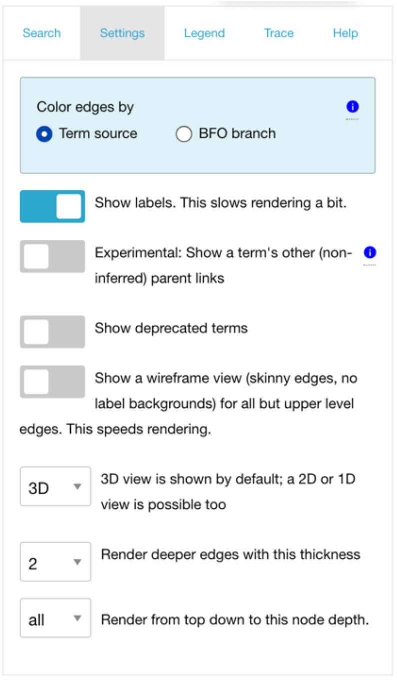
to be discerned while substantially zoomed-out from the graph. The 3d landscape also allows structure to be "stored" in plain view at scale, which is a kind of data compression as long as content areas are semantically summarized, which currently is achieved by mouseover identification of distant nodes/terms within those areas.

Clicking on a node will rotate and move the viewport camera towards that item and highlight it in red. As well, on the Search tab shown in Figure 2, a "Term search …" pull-down menu lists all class terms available in the ontology with an added usability feature that as one types, both term label and synonyms can be searched, so that 'dog', though not present in the label, will return term 'Cannis lupus familiaris'. This enables people to use colloquial vocabulary to access information a formal ontology can provide. Selecting a search result locates and travels to the term node of interest. The Search tab also includes the Information Architecture Ontology (IAO) definition, rdfs:label and synonyms of any focused term.

By default each node/term only has one parent link displayed, but the "Experimental" setting shown in Figure 5 triggers the display of multiple parent links in orange. The Trace tab contains a new feature under development for visualizing a disjoint axiom-related unsatisfiability error about a particular term in a given ontology that one has encountered. One must invoke the command line robot tool[9] directly on the unsatisfiable ontology to get the error explanation report (in Markdown format) which can then be entered into OntoTrek to visualize a contradictory path.

3. Discussion

To explore the presence of term reuse, selected OBO Foundry family ontologies[10] are listed in the OntoTrek menu. Some choices such as AGRO, ECO and ECOCORE demonstrate substantial reuse of upper level BFO classes, with remaining items, if any, are located under a top-level owl:Thing node. The Human Disease Ontology illustrates a tighter domain, drawing only on phenotype and anatomy components. Others lack any reference to the BFO context. This exposes the difference between application ontologies involving models that draw on many domains, and reference ontologies that are domain specific or yet to take on upper level schemas.



Color edges by
[✓ Term source] BFO branch

[✓] Show labels. This slows rendering a bit.

Experimental: Show a term's other (non-inferred) parent links

Show deprecated terms

Show a wireframe view (skinny edges, no label backgrounds) for all but upper level edges. This speeds rendering.

3D — 3D view is shown by default; a 2D or 1D view is possible too

2 — Render deeper edges with this thickness

all — Render from top down to this node depth.

Figure 5: Other OntoTrek features that influence rendering speed

OntoTrek enhancements are envisioned both in content display and navigation. A few problems exist in the interface, namely that the animation that takes one to a node of interest involves pitch and roll that can be disorienting. Motion controls should have an option to restrict movement so that top is always top, i.e. pitch and roll never occur, only yawl and z axis elevation. A visual indication of where the centre of rotation is would be useful too. An option to specify an ontology directly by URL would be useful. Adding axiomatic details of nodes - related object properties and data properties that would only be rendered temporarily on demand so as not to visually overwhelm the interface. A step towards making this a tool for building reusable components would involve the ability to select nodes and branches for hiding, deleting, moving, exporting or mapping, and the ability to visualize the results of Sparql queries on a given ontology.

4. Software Availability

OntoTrek can be explored at https://genepio.org/ontotrek . The latest code for OntoTrek is at https://github.com/GenEpiO/ontotrek . The OntoFetch python script for obtaining term hierarchies is available at https://github.com/GenEpiO/ontofetch .

References

[1] Dudáš, M., Lohmann, S., Svátek, V., & Pavlov, D. Ontology visualization methods and tools: A survey of the state of the art. In The Knowledge Engineering Review, 33, e10 (2018).

[2] Web resource: http://voila2018.visualdataweb.org/slides/voila2018_introduction_survey_slides.pdf . Visited August 15, 2019

[3] Bosca, A., Bonino, D. & Pellegrino, P. OntoSphere: more than a 3D ontology visualization tool. Swap (2005).

[4] Somasundaram, Ramanathan. “OntoSelf: A 3D Ontology Visualization Tool.” (2007).

[5] Pei, Zhisong. “OntoSELF+TQ: A Topology Query System for OntoSELF.” (2009).

[6] Wiens, V., Lohmann, S. & Auer, S. WebVOWL Editor: Device-Independent Visual Ontology Modeling.

[7] Kleiberg, E., van De Wetering, H. & Van Wijk, J. J. Botanical visualization of huge hierarchies. in IEEE Symposium on Information Visualization, 2001. INFOVIS 2001. 87–94 (IEEE, 2001).

[8] Arp, R., Smith, B. & Spear, A. D. Building Ontologies with Basic Formal Ontology. (MIT Press, 2015).

[9] R.C. Jackson, J.P. Balhoff, E. Douglass, N.L. Harris, C.J. Mungall, & J.A. Overton. ROBOT: A tool for automating ontology workflows. BMC Bioinformatics, vol. 20, July (2019).

[10] Smith, B. et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. 25, 1251–1255 (2007).