Introduction to BFO 2 OWL Implementation

Choices, Issues, Resolutions

Alan Ruttenberg, Graz, July 2012

Quick start

- O Get Protégé 4.2 latest version
- Open http://purl.obolibrary.org/obo/bfo.owl
- Follow along

Goals of this section

- Say a few words about OWL 2
- Say a few words about BFO
- O Describe the BFO 2 OWL design principles
- Make clear the status of BFO 2 OWL
- O Discuss contents of BFO 2 OWL
- Review changes relative to BFO 1
 - O Give guidelines for migrating from BFO 1 to BFO 2
- O Discuss selected aspects of the implementation
 - Strategy for time-indexed properties
- Review selected open issues
- Answer your questions

How is BFO made

- O Historically, Barry Smith developed BFO
- Because BFO will be part of the OBO Foundry, it will now be developed collaboratively
- There is a group of about 10 individuals who have actively worked on the current version
- O It should be understood that BFO is not yet finalized
- You should understand phrases such as "BFO has X, or BFO doesn't X", until this development cycle of BFO is finished, as "in my opinion BFO has X" or "the current BFO document has X"
- In particular, there are disagreements between, e.g. Barry and myself and among other BFO working group members.
- We want the participants of this and future workshops to also participate in the development of BFO. That means you should question if you don't understand and complain if you too disagree with what you hear.

Status of BFO 2 2012-07-20 "Graz" Release

- This version of BFO represents a major update to BFO and is not backwards compatible with BFO 1.1. The previous version of BFO, version 1.1.1 will remain available at http://ifomis.org/bfo/1.1 and will no longer be updated. We expect to provide automated support for migrating from BFO 1.1 to BFO 2 at some point in the future.
- This version is a draft release for public comments. As such we expect it to generate a lively discussion rather than be a stable ground for building application ontologies. The release is the product of intensive discussions and a series of prototypes which kicked off with a November 2011 workshop held in Buffalo.
- While we have attempted to produce a file reflecting the current BFO2 specification document, there remain issues and bugs that will be fixed in subsequent releases. Our expectation is that IDs in this version, going forward, will not disappear and we will follow the GO/OBI deprecation policy. In addition, not all elements of the reference itself are stable and there may be (even substantial) changes to the reference before we have the first release candidate.

OWL 2

- We make use of the full range of OWL 2 using the direct semantics.
- We may release weakened versions (e.g. dropping axioms that are outside the EL profile) if there is demand
- Features of OWL 2 that are useful for us
 - Property chains
 - Ability to annotate anything (for documentation purposes)
 - Availability of free computational tools for reasoning
 - O Built to live on the web, which makes for easy uptake
 - ObjectHasSelf, which allows for local reflexivity

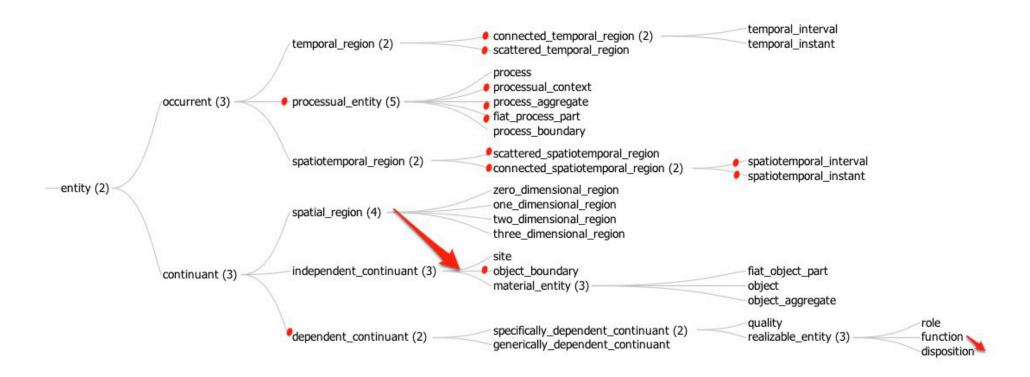
Problems

Running into issues related to the OWL 2 global constraints, for example wanting to use transitive properties in role chains, or make properties disjoint with self-linking properties

Selected design principles

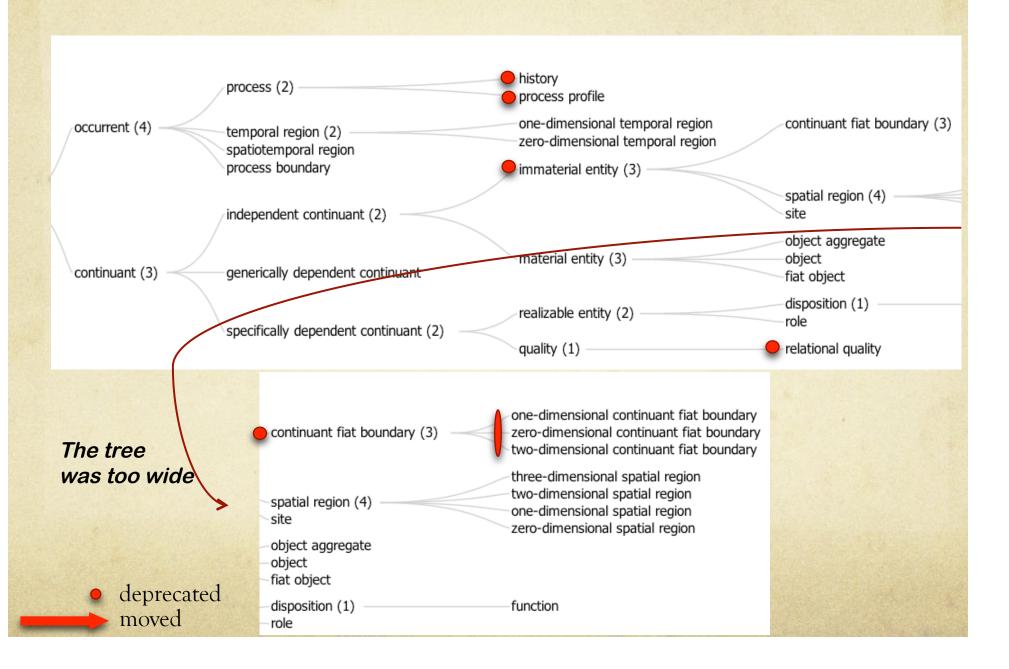
- 1. A clear reading of the OWL version in terms of BFO reference. A translation is a mapping from the OWL model axioms, entities, relations, etc. To the formal language used in the reference, currently FOL. A reading can be considered a data transformation that takes asserted and inferred axioms and results in FOL using types defined in BFO 2 reference. This translation should be complete no assertions in the OWL file can be left untranslated.
- 5. Make maximal use of reasoning to ensure quality/correctness. It is easy to make mistakes when ontologies have no automated procedure to test them. By adding as many constraining axioms which correspond to textual descriptions in BFO 2 Reference, we have he best chance of finding conceptual errors. We will aim to make clear the relation between the FOL and the OWL, independently of the reference. We will examine the reference with a mind to ensuring axioms that are not so labeled are marked explicitly as axioms.

BFO 1.1 Classes





BFO 2 Classes



Relations in BFO 2

- has material basis (disposition -> material entity)
- has history (material entity >> process, 1:1)
- o continuant part of (continuant -> continuant)
 - member part of (object <-> object aggregate)
 - o continuant proper part of
- occurrent part of (occurrent -> occurrent)
- o concretizes (sdc -> gdc)
- o exists at (entity -> temporal region)
- o specifically/generically depends on (sdc, gdc -> site or material entity)
- o specifically depends on (many possibilities)
- occupies spatial region (continuant -> spatial region)
- occupies spatiotemporal, temporal region
 - (occurrent -> spatiotemporal, temporal region)
- occurs in (process -> material entity or site)
- O located in (material entity or site -> material entity or site)
- has participant (process -> continuant)_{sdc} = specifically dependent continuant gdc = generically dependent continuant

No closure axioms in BFO 2

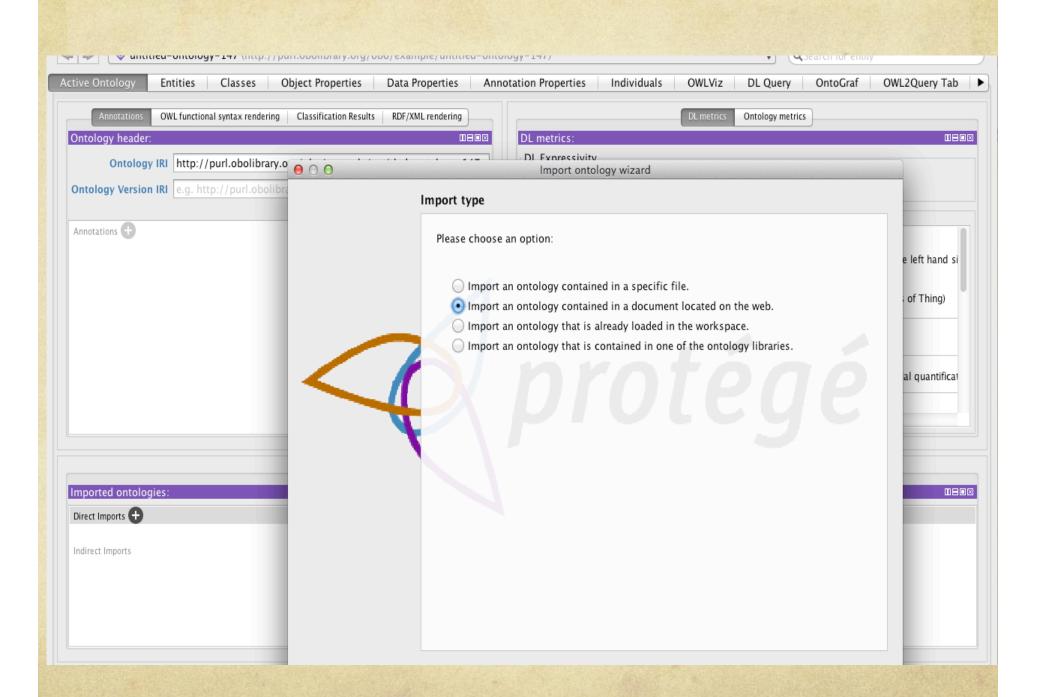
- OWL 1 had closure axioms axioms that said that at each level the subclasses of a class were the only subclasses.
- O These are no longer present in BFO 2
- Example: We don't close specifically dependent continuants because we're pretty sure there are more
- Example: We don't close continuant fiat boundary because the subclasses don't include the case of a sum of non-intersecting zero-dimensional (point) and one-dimensional (line) continuant fiat boundary.

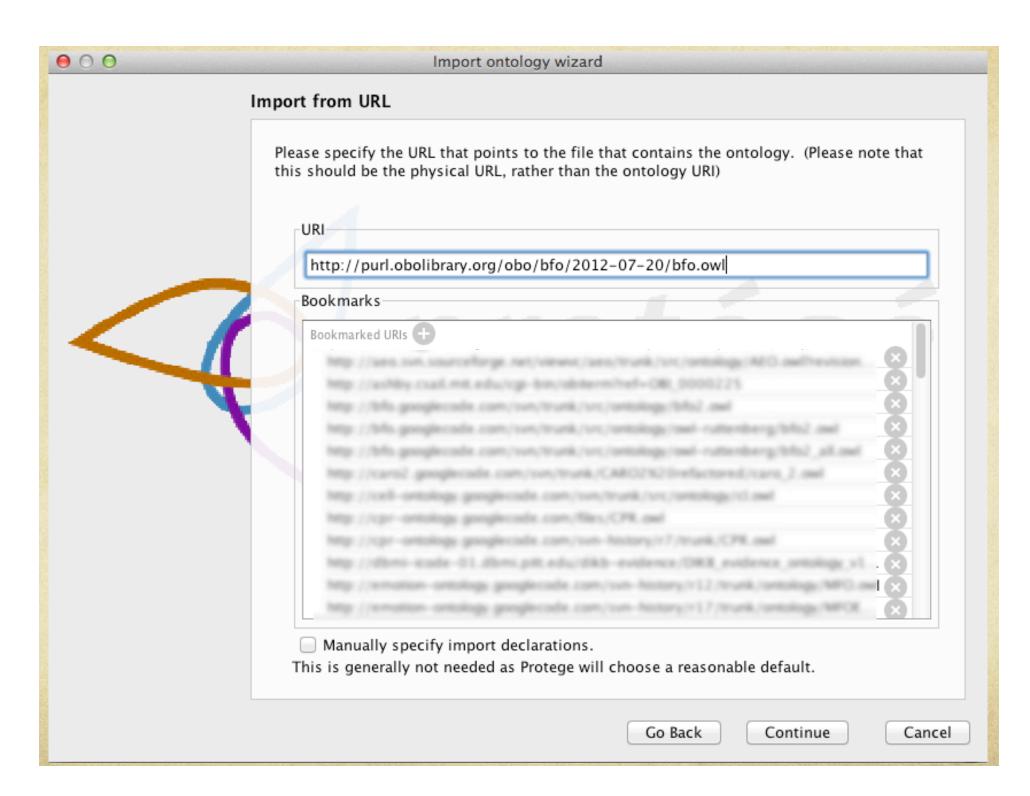
Siblings disjoint except between material entity children

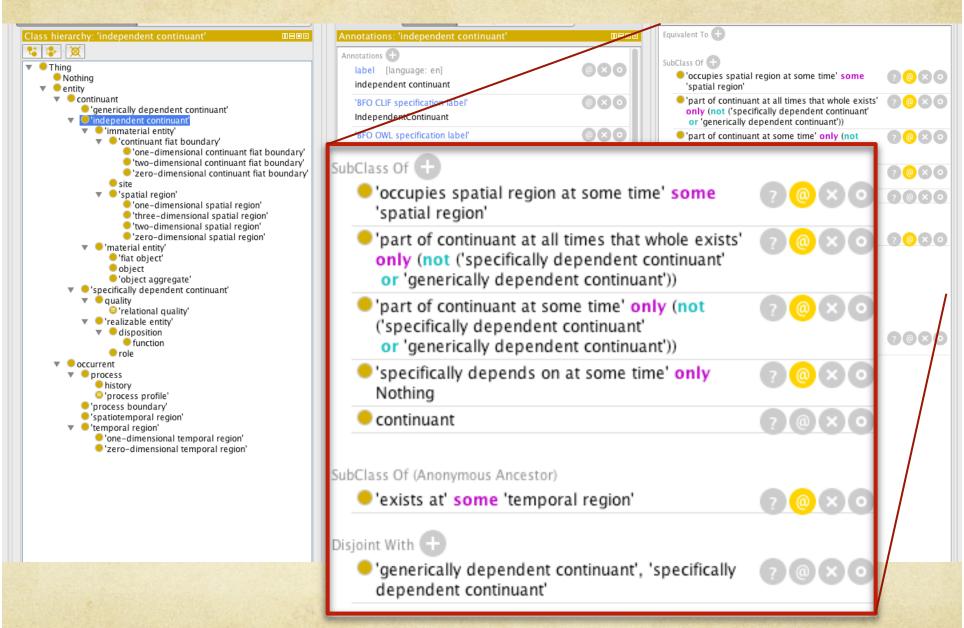
- A paradigm case: not ruling out granularity
- A molecule is, at the same time, an object aggregate (of atoms) and an object (for example as constituent of a crystal)
- In different granular contexts the same physically identifiable thing can justifiably classified as different types
- If we made the children of material entity be disjoint, we would make it hard to extend BFO to deal with granularity effectively.
- O So we don't

Importing BFO into your Ontology

- Use the OWL imports functionality that your editing tool provides
- O Give the import URI as
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/bfo.owl if you want to make sure you get today's version
 - http://purl.obolibrary.org/obo/bfo.owl if you want to import whatever the most recent version of BFO is.
- O Both URIs currently retrieve the same document, however that will change as soon as there is a new version released







http://protege.stanford.edu/download/download.html

Get the latest version of Protégé 4.2

Migrating classes from BFO 1.1 to BFO 2

- Note deprecated classes -
 - Specifically dependent continuant
 - Process X classes
 - Scattered/Connected X
- Attend to classes that are similar to earlier classes
 - object boundary, continuant fiat boundary
 - Function now subclass of disposition
- Look for applicability of new classes
 - o history, relational quality
- Learn the relations that are now in BFO and apply them.
 - Migrate use of RO to BFO, new RO

Migrating relations to BFO 2

- O Properties redesigned to not overload names as in BFO 1.
 - o part_of (class, class)
 - o part_of (continuant, continuant, t)
 - o part_of (occurrent, occurrent)

Strategies to remove overloading

- O Drop class/class relations
- Name properties according to type
- Use "at-some-time", "at-all-times" relations to acknowledge temporal indexing. (maybe –during-process relations coming)

O How to migrate

- Attend to part relations which have split continuant/occurrent/at some times/at all times If you use part relations a lot this will be the biggest work
- Attend to other temporally indexed relations
- Learn and use new relations
 - o material basis
 - o history of
 - o specifically/generically depends on
 - o concretizes
 - o realizes

Why time indexing is important (even in BFO OWL)

- Documentation of BFO has always had time indexed relations, for example parthood between continuant instances is written as part_of(c1,c2,t)
- Our first design principle is that the the OWL file have a BFO reading
- In the prior Relation Ontology OWL implementation the part of relations were binary, and were not given any BFO reading.
- Since BFO defines the meaning of the relations, we have no basis for formally understanding relations if there is not a BFO reading
- We (OBO community) is moving more towards representation of instances. Prior to this only "class level" relations were used, which were not time dependent. However OWL only provides for making quantified sentences over individuals rather than reifying classes.

Understanding the timeindexing approach

- The problem: OWL provides only binary relations
- Time indexed relations are ternary
- O Solution: Move quantification over time into the instance-level relation
- For part_of(c1,c2,t) define:

 part-of-at-some-time(c1,c2) = def

 forsome(t) exists_at(c1,t) & exists_at(c2,t) & part_of(c1,c2,t)

 part-of-at-all-times(c1,c2) = def

 forall(t) exists_at(c1,t) -> exists_at(c2,t) & part_of(c1,c2,t)

Note: only part-of-at-all-times is transitive

Note: part-of-at-all-times is not the inverse of has-part-at-all-times

Note: part-of-at-all-times-that-whole-exists can be defined so as to be the inverse of has-part-at-all-times

part-of-at-all-times-that-whole-exists(c1,c2) = def
forall(t) exists_at(c2,t) -> exists_at(c1,t) & part_of(c1,c2,t)

Known issues in BFO 2

- No proper way to represent non-rigid universals (professor, fetus) yet
- o process profile not well liked by part working group
- Acceptability of time-indexing? How to extend it.
- Material basis only for dispositions (what about qualities, and what about qualitative basis?)
- Alan doesn't like specific dependence of processes on material entities
- Additional requested relations begins/ceases to exist during (continuant -> process)
- O BFO 2, OBO Format, and bfo-ruttenberg.owl (vintage 2010-05)
- Are some definitions too formal? (see def. of process, process profile)
- Change in definition of process boundary (now only temporal)

Known issues in BFO 2

- Permanent generic parthood, and similar. Can't say, currently, that at all times every independent continuant is located in some spatial region (different ones at different times)
- O Support for granularity (what to do about has grain relation in use in some ontologies)
- Stages (temporal parts of history). Need to axiomitizing that they map uniquely to the material entity the history is of.
- Review of reference and completion of adding axioms. Feedback from OWL implementation back into reference.
- Only relation relating site to host is continuant part of
- O How to relate process profile to, e.g. qualities they profile

Resources

- Release notes (start here)
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/ReleaseNotes
- Reference document
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/Reference
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/Reference.htm
- O OWL Document
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/bfo.owl
- First order logic formulation
 - http://purl.obolibrary.org/obo/bfo/2012-07-20/fol.pdf
- Discussion list
 - http://groups.google.com/d/forum/bfo-owl-devel
- O Tracker/Issues list
 - http://code.google.com/p/bfo/issues/list

Acknowledgements

- Recent contributors: Mauricio Almeida, Thomas Bittner, Jonathan Bona, Mathias Brochhausen, Werner Ceusters, Mélanie Courtot, Randall Dipert, Bill Duncan, Janna Hastings, Albert Goldfain, Leonard Jacuzzo, Ludger Jansen, Pierre Grenon, Larry Hunter, Chris Mungall, Fabian Neuhaus, David Osumi-Sutherland, Bjoern Peters, Mark Ressler, Robert Rovetto, Ron Rudnicki, Alan Ruttenberg, Stefan Schulz, Barry Smith.
- O Special thanks to Stefan Schulz, Janna Hastings, Melanie Courtot for review of the OWL document, and Mathias Brochhausen for ideas for this presentation.