Overcoming the Ontology Enrichment Bottleneck with Quick Term Templates

ICBO: International Conference on Biomedical Ontology
University at Buffalo, NY July 24-26, 2009

Session 6: Computing with Ontologies

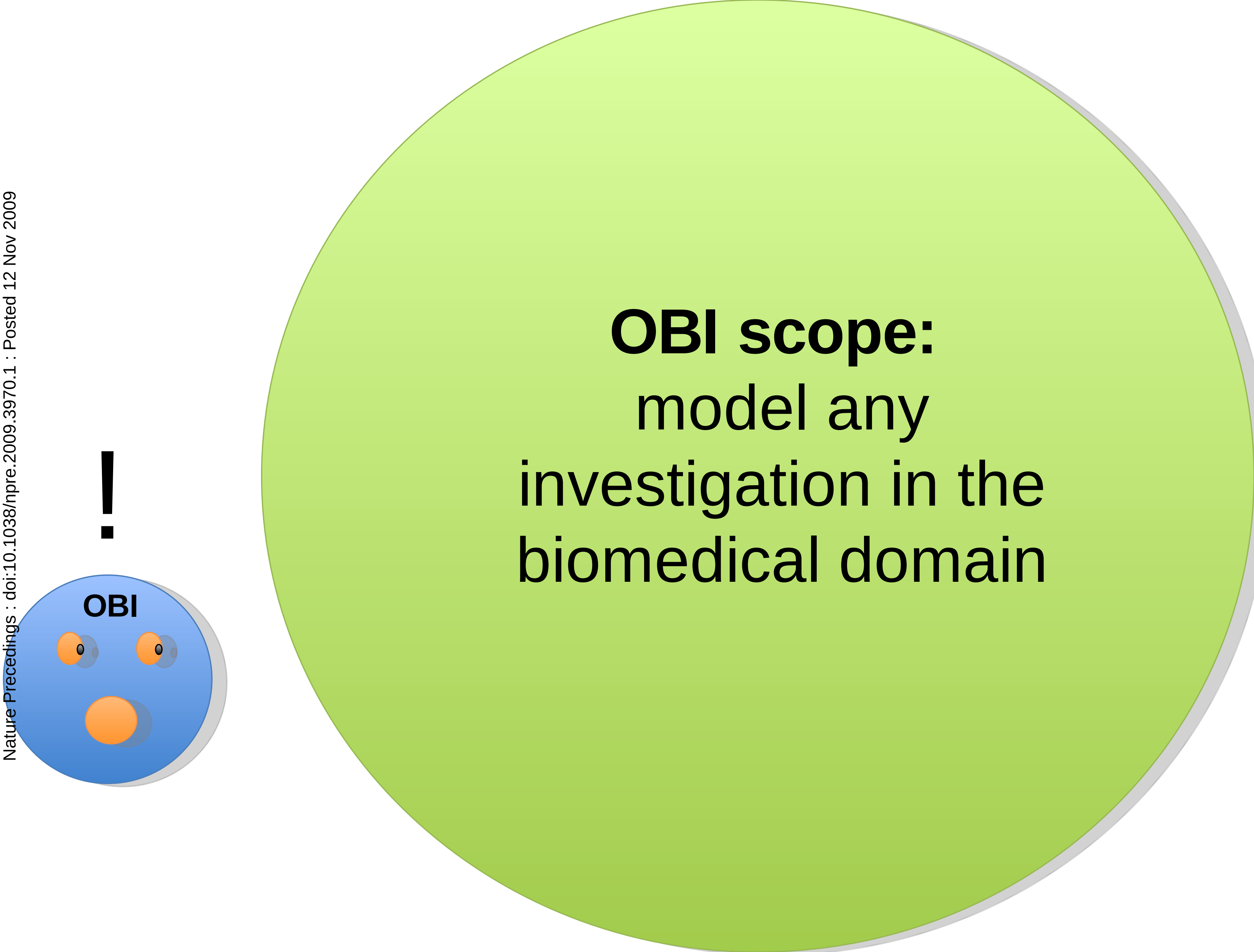
Bjoern Peters, Alan Ruttenberg, Jason Greenbaum, Melanie Courtot, Ryan Brinkman, Patricia Whetzel, Daniel Schober, Susanna Assunta Sansone, Richard Scheuermann, the OBI consortium and Philippe Rocca-Serra.

Building good ontologies is hard

- domain knowledge is required to know **what** terms to add
- technical skills are required to know **how** to add terms and define them logically
- both are required to ensure that term definition matches intended meaning

Each term can take significant time and effort

many terms are needed to cover a domain



OBI scope:
model any investigation in the biomedical domain

Approach 1

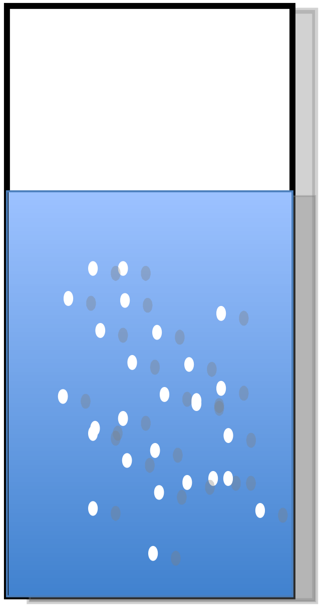
- ‘Quick ID’
  - Provide term submission interface for domain experts through spreadsheet / website
  - ask for textual definition, examples, ...
  - assign an ID immediately (or after cursory review); guarantee stability of ID.
  - Review later where to place the term in the ontology, rework definition, and how to define it logically
- Problem:
  - Later modeling may not reflect what user intended.
  - Danger of building ‘trash heap’ of poorly defined

Approach 2: Quick Term Template

- Identify set of classes that are specializations of one parent
- Generate logical definition of parent class, including restrictions that serve as a design pattern for specializations
- Create spread sheet (quick term template) for use by domain experts to create child terms
- Parse spreadsheet and add fully logically defined child classes into ontology

Example: analyte assay

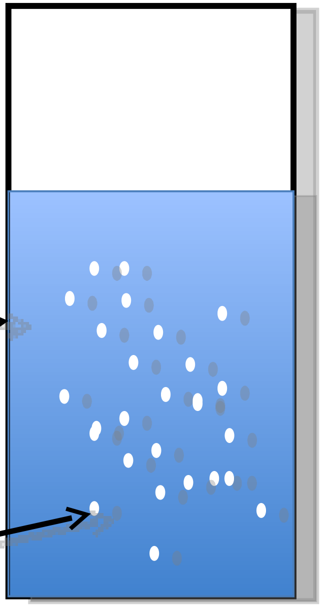
- Many assays determine the concentration of an entity of interest, e.g.
  - glucose concentration in blood
  - antibody titer in serum
- Definition of analyte assay: An assay with the objective to capture information about the concentration of an analyte in an evaluant.



Logical definition of ‘analyte assay’

an assay that

- ‘achieves planned objective’ some ‘analyte measurement objective’
- and realizes some ('evaluant role' that ‘inheres in’ some ‘material entity’)
- and realizes some ('analyte role' that ‘inheres in’ some ('scattered molecular aggregate‘ and has_grain only ‘molecular entity’))
- and has_specified_output_information some ('scalar measurement datum' and ('is quality measurement of' some 'molecular concentration') and ('has measurement unit label' some



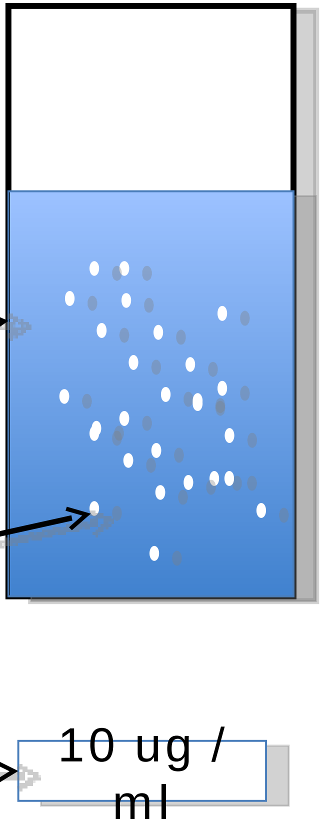
10 ug / ml

This took ~2 years

Logical definition of ‘analyte assay’

an assay that

- ‘achieves planned objective’ some ‘analyte measurement objective’
- and realizes some ('evaluant role' that ‘inheres in’ some ‘**material entity**’)
- and realizes some ('analyte role' that ‘inheres in’ some ('scattered molecular aggregate‘ and has_grain only ‘**molecular entity**’))
- and has_specified_output_information some ('scalar measurement datum' and ('is quality measurement of' some 'molecular concentration') and ('has measurement unit label' some concentration unit label'))

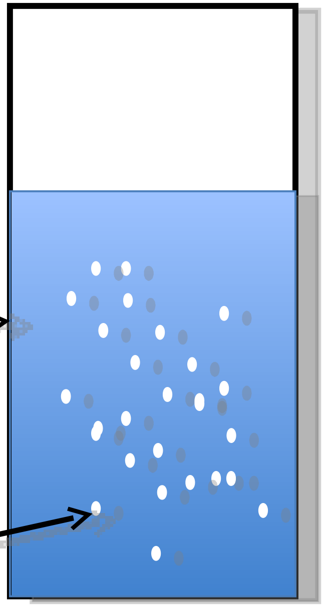


Logical definition of ‘analyte assay’

an assay that

- ‘achieves planned objective’ some ‘analyte measurement objective’
- and realizes some ('evaluant role' that ‘inheres in’ some ‘**portion of blood**’)
- and realizes some ('analyte role' that ‘inheres in’ some ('scattered molecular aggregate‘ and has_grain only ‘**glucose**’))
- and h ('scal and (' 'molecular concentration') and ('has measurement unit label' some concentration unit label'))

QTT: Create a template to define specializations of analyte assay based on the evaluant and analyte used



10 ug / ml

Example QTT template

Analyte	Evaluant
CHEBI:17234 (glucose)	FMA:12274 (urine)
PRO:000000017 (interferon gamma)	OBI:1000023 (cell culture supernatant)
CHEBI:17234 (glucose)	

Automatically created labels

2)glucose concentration measurement in urine

3)interferon gamma concentration measurement in cell culture supernatant

4)glucose concentration measurement

Processing template:

- Parse template for syntactic accuracy
- Import terms as necessary via MIREOT procedure
- Generate a new unique identifier for each class
- Create class metadata (label, textual definition) based on restrictions
- Set N&S class restrictions based on OWL template
- Use reasoner to check for consistency and duplicity
- Return list of identifiers to user

Constraints on classes inserted into template

General:

- Class must already be in OBI
  OR
- Class must belong to OBO Foundry candidate ontology, and have a parent class presently imported in OBI

Specific for example template:

- ‘analyte class’ must be a molecular entity
  (this includes ‘molecular entities’ in ChEBI and ‘protein’ from PRO)
- evaluant class can be any material entity
  (should be further restricted)

Current implementation status

- A perl script has been developed for testing the overall approach. Not for production
- Some java code available (James Malone)
- interest in Protégé community to integrate into plugin (Martin JO'Connor)
- interest to integrate with OntoFox (He group)

Future software development: Xmas list

- A protégé plugin would be nice (our current working environment)
- Ease creation of new templates
  - Point to a parent class
  - Identify which parts of the logical definition should be in template
  - Set constraints on what classes are allowed to fill template
  - Provide text to be completed for metadata
- Ease completing a template
  - directly use OLW / BioPortal to browse for applicable filler classes
- Fully automate class generation from completed template as outlined before

Summary: Benefits of the QTT approach

- Provides a mechanism to add exactly defined classes to the ontology without the need for manual review
- Shields community domain experts (e.g. wet lab scientists) from knowledge engineering jargon / specifics
- Allows to create specific templates to map existing well structured artifact into a new framework
  (e.g. for OBI: IUPAC clinical chemistry, nomenclature, LOINC)

Potential Limitations

- Using tables will not work for very complex class definition
- Fundamental limitation through MIREOT: The imported 3rd party classes do not keep their restrictions
- Relies on imported ontologies to have a clean is-a structure

Acknowledgments

- Philippe Rocca-Serra, Alan Ruttenberg, Jason Greenbaum, Melanie Courtot, Ryan Brinkman, Patricia Whetzel, Daniel Schober, Susanna Assunta Sansone, Richard Scheuermann
- OBI consortium
- OBO Foundry