A Framework for Constructing Web Ontologies using Concept Maps

Pat Hayes
Tom Eskridge
Thomas Reichherzer
Raul Saavedra

This work is continuing with additional DoD funding, in collaboration with Pragati, Inc (Mala Mehrotra)

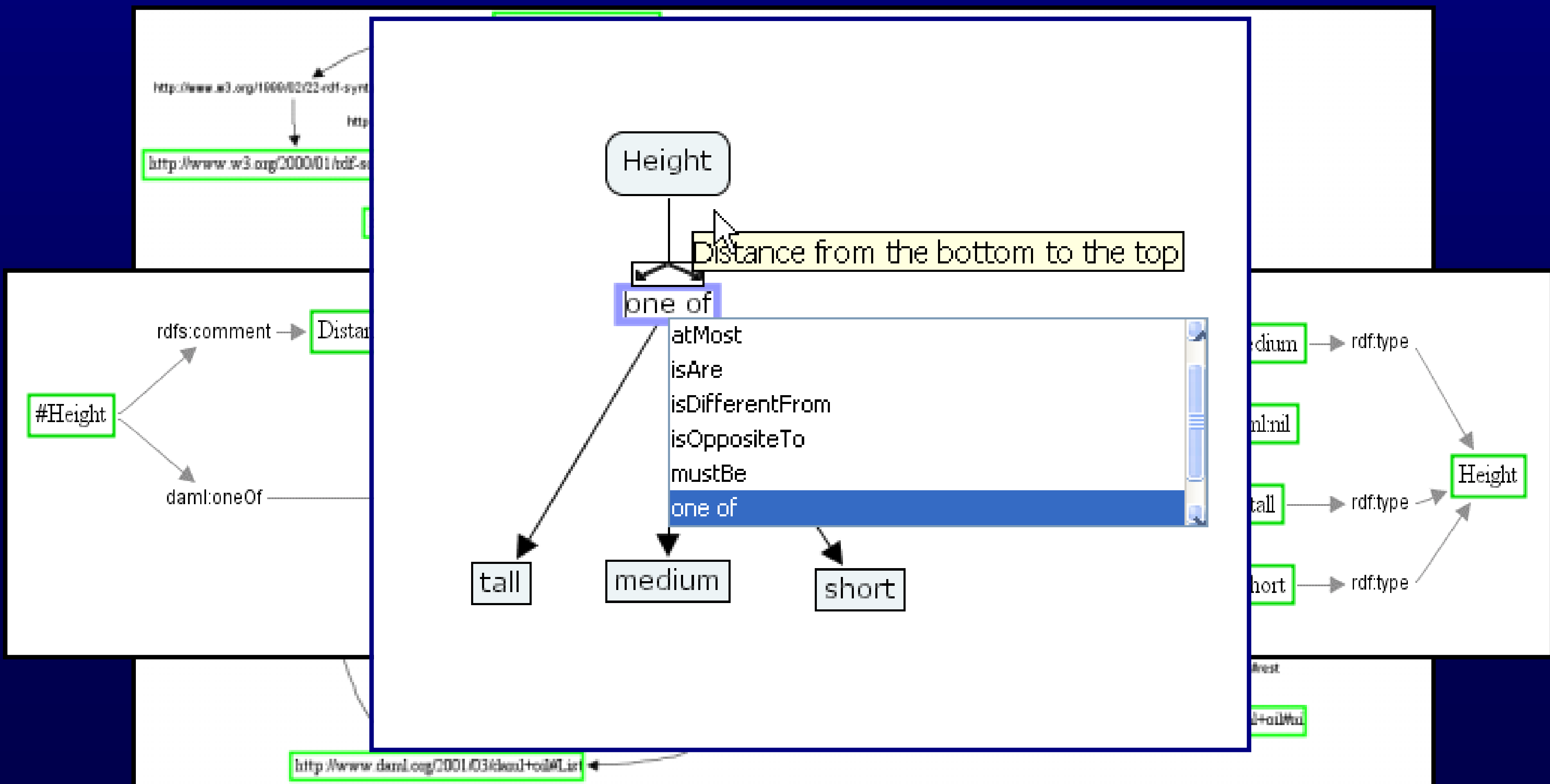
Project Goals

- A tool to build, edit and display OWL ontologies.
  - A graphical approach based on concept maps.
  - Concept maps as an ontology building tool
  - Representing existing ontologies as concept maps.

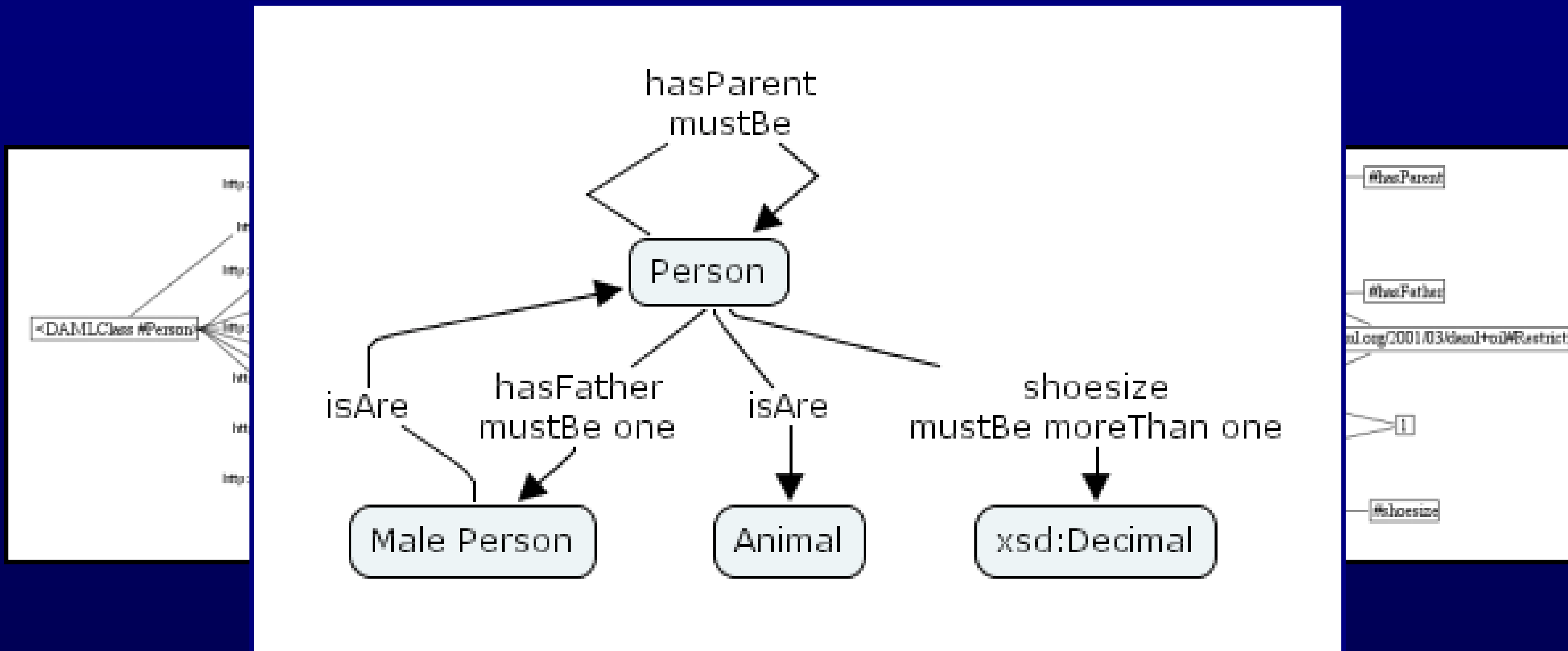
- Support tools for building OWL ontologies.
  - Search in ontologies for suitable concepts.
  - Cluster analysis

Simplifications of Maps after Import

- RDF lists, OWL restrictions, and other constructs are hidden from explicit view

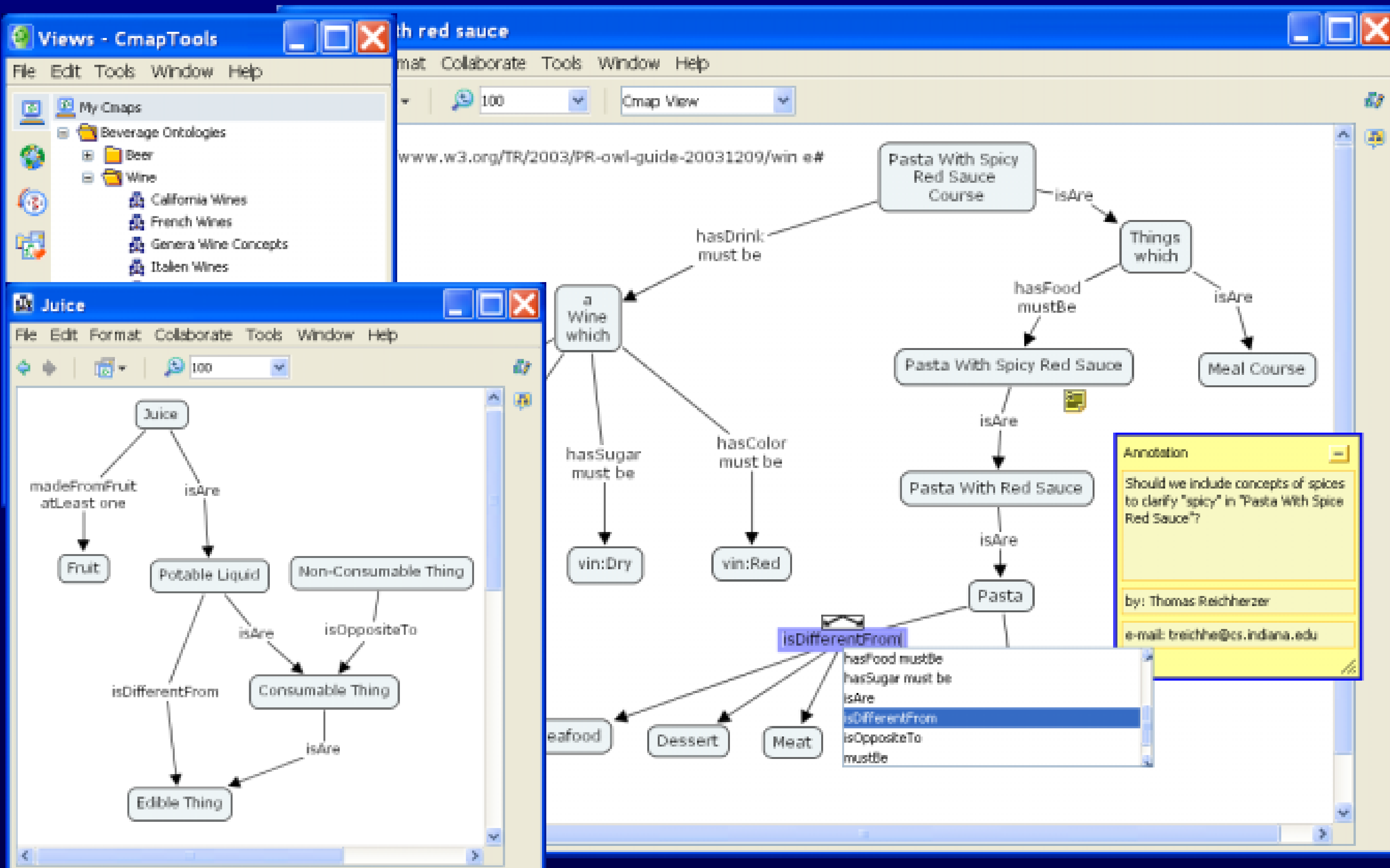


Simplifications of Maps after Import (Cont.)

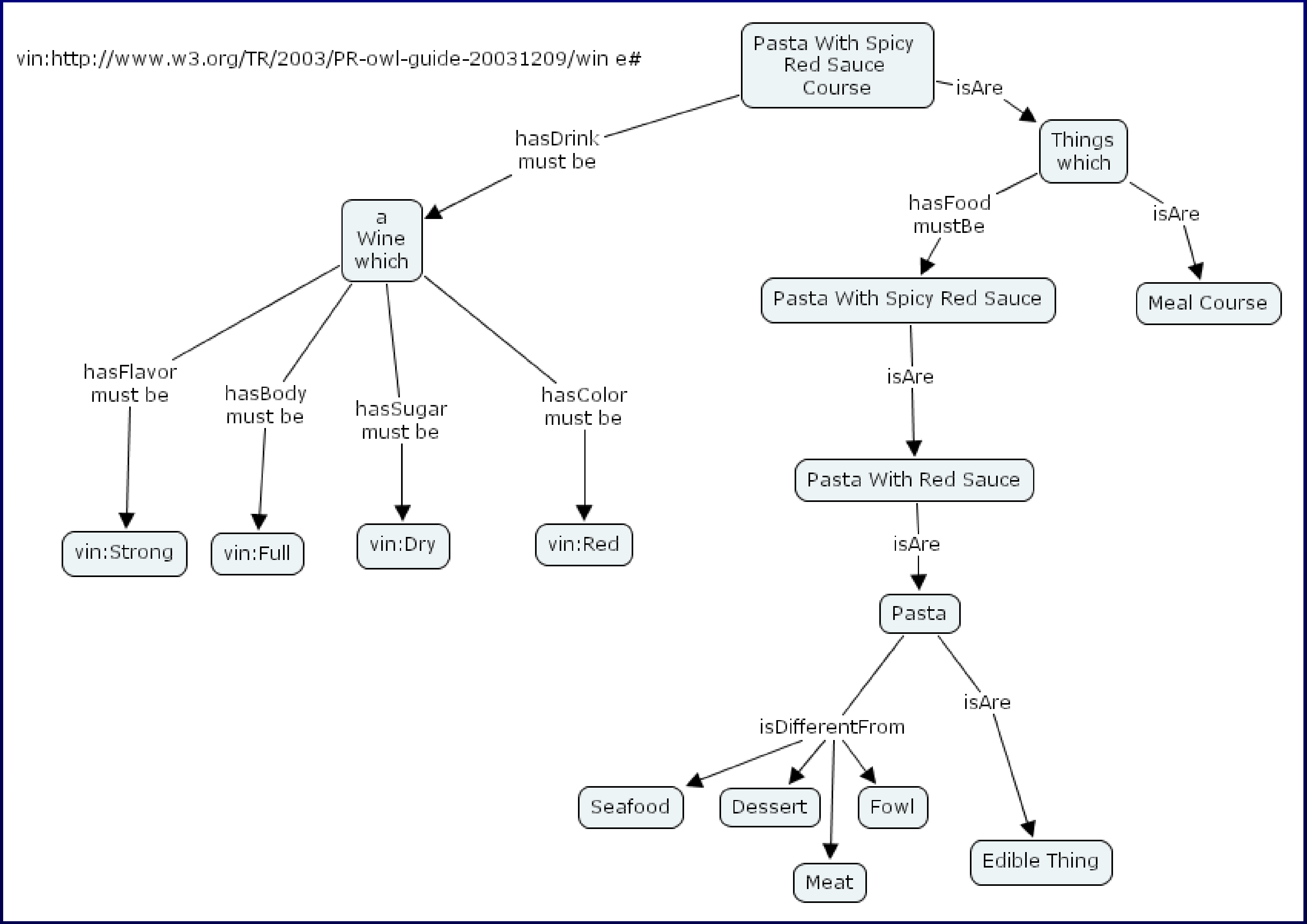


- Reduction of complex Cmap Ontology graphs into easier readable concept maps.

Cmap OWL Editor/Browser

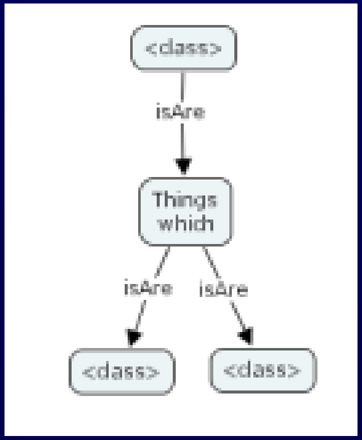


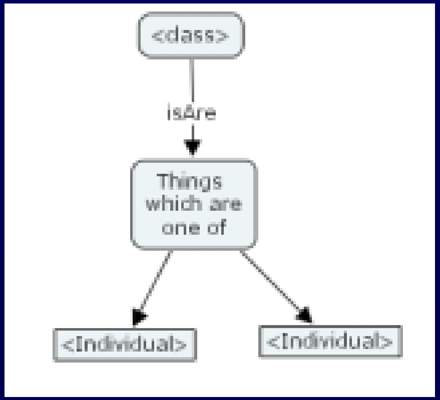
OWL Concept Map Examples (cont.)

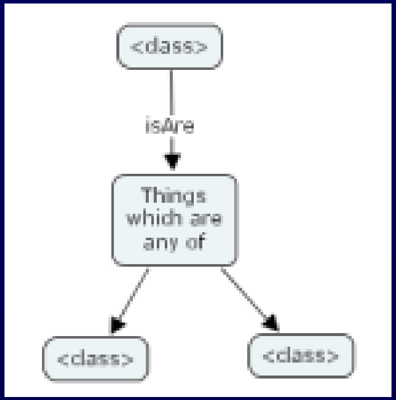


Constructing ontologies as concept maps

- Exporting concept maps to OWL format.
- Concept map conventions for defining restrictions.
  - atleast, atMost, must be, Things which
- Templates aid in forming most repetitive, complicated restrictions that can be exported to OWL.



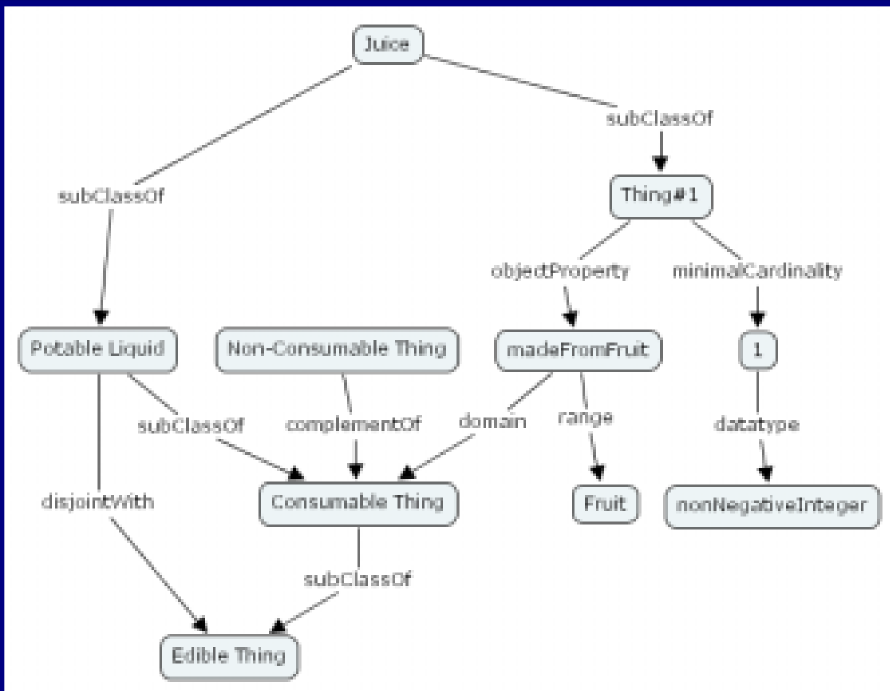


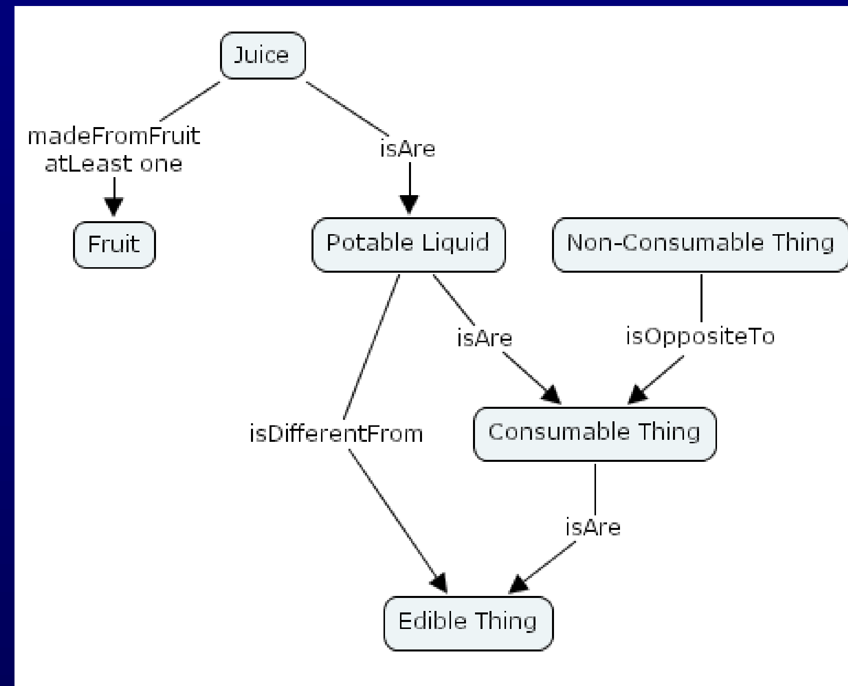


Rendering XML ontologies as concept maps

- Segmenting large ontologies into smaller, more user-manageable concept maps.
  - Topological approach to segmentation.
  - Consider particular restrictions or subclass hierarchies.
- Simplifying large ontologies.
  - Hiding technical details.
  - Collapsing large constructs into small set of nodes with special notations.
- Navigational support for large ontologies.
  - navigating by RDF statements (table of RDF triples)
  - navigating by concepts (rapid ordered search)

OWL Concept Map Examples





OWL ontology <u>represented</u> as a concept map.

OWL ontology <u>constructed</u> as a concept map using conventions.

Selecting concepts based on clustering

- User composing an ontology requires a concept. Invent one or re-use one?
  - Keyword-based matching to concept names in OWL directories is feasible …
  - …but may produce many results. Selection requires analysis of clustering and connectivity between and among existing concepts
- CODE tool will interface with Pragati's MVP-CA system to provide user analysis of concept reusability in context.
  - Preliminary studies using Cmap contexts for Web search are promising.

End

Some Principles of Web Logic

Pat Hayes
work in progress

The logic of the Sem Web

- All Sweb ontology languages are (subsets of) FOL.
  - Possible exceptions all seem to be concerned with ‘local’ information, eg non-mon rule applications to elements of a class described by Sweb notations: the global SW description sanctions the local use of non-mon rules. I will ignore these local issues.

- Abstraction: *FOL reasoners*/communicating agents in an *open world* linked by *Web transfer protocols*

- What *general conditions* constrain this picture?

Sweb: logic meets architecture

FOL reasoners/communicating agents in an *open world* linked by *Web transfer protocols*

1. Inference and communication should commute.

   A requests information P from **B** and performs valid FO inferences on it; or, **A** requests **B** to perform valid FO inferences on it and communicate the results. *Web logic should sanction the same inferences in both cases.*

2. Inference should be stable in the light of new information

   Weaker than strict monotonicity since individual-level data can be non-mon; knowledge may change, but *logical inference* (Web entailment) should not.

3. Content transfer on-demand (pull transfer) should be sufficient to perform Web inference

   Reasoners should not be required to *negotiate* before performing valid inferences

4. Lack of information should not restrict Web inference

   If A entails B when C is present, then A should entail B when C is not present.

Some consequences (1)

- 1. EITHER syntax must encode fixed global conventions OR logic must apply independently of syntactic constraints.

- Example: **A** uses a symbol as a relation name, **B** uses it as an individual name. Both uses are logically correct. *Commutation* requires (either that this is rendered impossible, or) that Web inference applies uniformly to both:
  - p=q from B
  - p(a b) from A
  - q(a b) odd conclusion using a valid FO logical principle
  - exists (x) x(a b) even odder conclusion using a valid FO logical principle

- This is all valid FO reasoning but without a fixed signature. SCL provides a framework and a semantics.

Some consequences (2)

- 2. The Horatio Principle: unrestricted universal quantification is *incoherent* in Web logic.
  - Example: **A** writes an ontology about fish which presumes that everything in the universe of discourse is a fish. B writes an ontology about vertebrates using a term from A's ontology. C writes an ontology about living things using a term from B's ontology. D writes an ontology about cellular chemistry using a term…The only way for D to safely use A's ontology is by restricting the quantifiers to the intended domain.
  - Any universal quantifier in an ontology must be understood as ranging over a subset of the global universe. *Nobody owns the universe.*
- Transmission of any ontology implicitly ***guards*** all its quantifiers with the name of the ontology's universe.
  - OWL currently violates this with *owl:complementOf* and *owl:Thing*. Better to use relative complement and a 'local' universe name. Quick fix would be to re-define *owl:imports.*
- Useful side-effect: *Web logic is locally full first-order but globally decideable*.