Distributed Semantic Web Knowledge Representation and Inferencing

Harold Boley

UNB, Faculty of Computer Science

Keynote at ICDIM 2010

6 July 2010

Update: 3 October 2016

Introduction

- Interdisciplinary approach to the (Social) Semantic Web
  - Computer, Information, and Data Science, AI, Logic, Graph Theory, Linguistics, ...
- Representation & Inferencing Techniques for Distributed (Internet/Web-networked) **Knowledge** Management, Visualization, Interoperation (e.g., Object-Relational), and Access to (Big) Data
- Applications in eHealth, eLearning, eBusiness, Ecosystem Research, ...

Three Levels of Knowledge: Visual and Symbolic Representations

Knowledge elicitation as gradual formalization

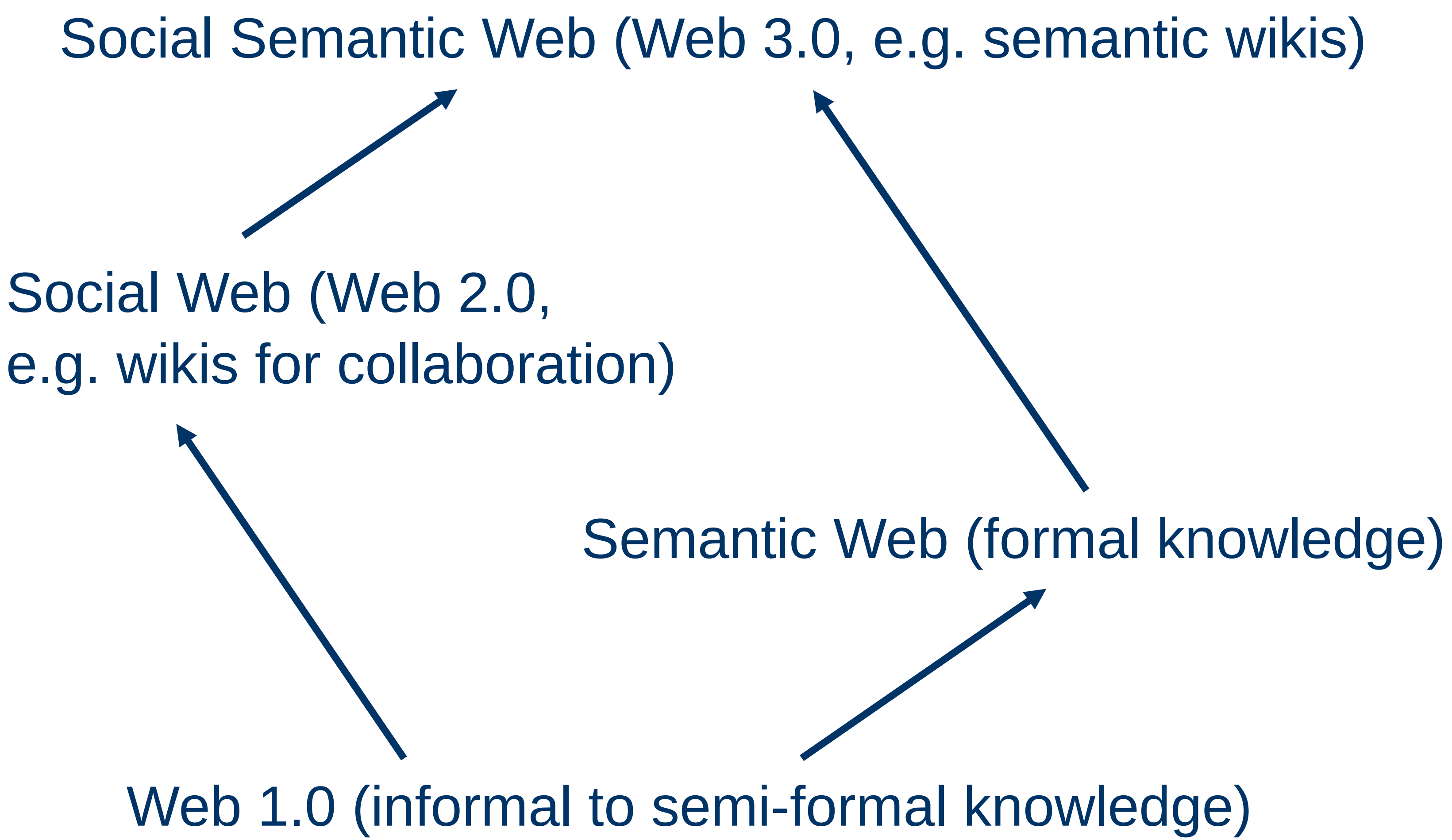
	visual	*symbolic*
formal	graph theory	predicate logic
semi-formal	standardized graphics	controlled natural language
informal	hand drawing	natural language

Three Levels of Knowledge: Described by Formal Metadata

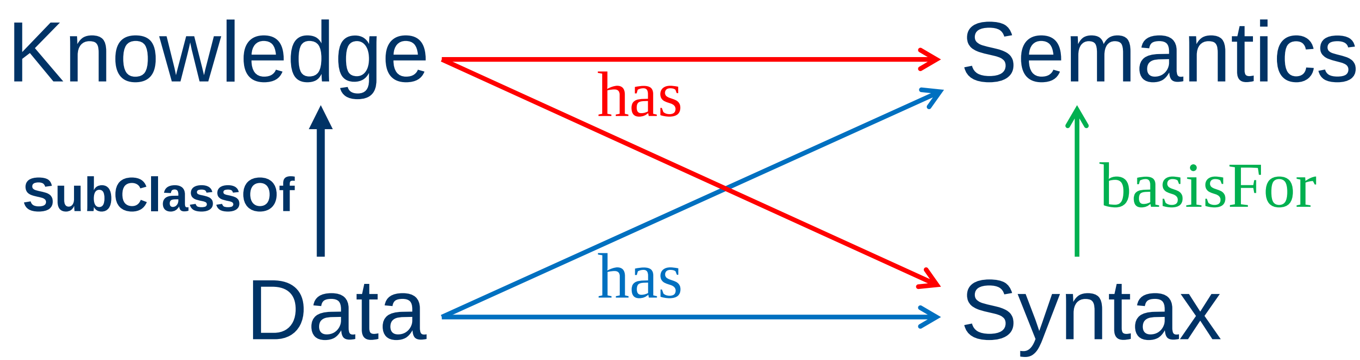
Formal knowledge can act as metadata to describe knowledge of all three levels for retrieval and inferencing with high accuracy

	visual	*symbolic*
formal	graph theory	predicate logic
semi-formal	standardized graphics	controlled natural language
informal	hand drawing	natural language

Web as Standard Distributed Knowledge Medium for Collaboration



Knowledge and, Specifically, Data have Semantics, Based on Syntax



Knowledge subsumes Data by inferring Knowledge (e.g. Data) from other Knowledge (e.g. Data)

Semantics based on Syntax by distinguishing subsets of (‘true’) formulas from the set of all formulas

Engines compute Inferences

Inferences preserve truth distinction

Via ‘Meaning Function’ (part of Interpretation)

Example: Data (Ground Facts)

Croco(c)
Horse(h)
Mule(m)
Pony(p)

Ground: No variables as arguments

Example: Knowledge (Beyond Data: Implication Rules)

$$\text{Mule}(x) \Rightarrow \text{Horse}(x)$$

$$\text{Pony}(x) \Rightarrow \text{Horse}(x)$$

Implies

Example: Inference

$$\begin{array}{l} \text{Pony}(x) \Rightarrow \text{Horse}(x) \\ \text{Pony}(p) \end{array} \quad \vdash \quad \text{Horse}(p)$$

Entails

Example: Syntax

$Mule(x) \Rightarrow Horse(x)$
$Pony(x) \Rightarrow Horse(x)$
Croco(c)
Horse(h)
Mule(m)
Pony(p)
. . .

$pred(var) \Rightarrow pred(var)$

$pred(const)$

Example: Semantics (Truth Directly Distinguished)

$\text{Pony}(x) \Rightarrow \text{Horse}(x)$ ◄ "Each pony is a horse"

Mule(m) ◄ "m is a mule"

Pony(p) ◄ "p is a pony"

Asserted by an authority
or
Found by a sensor-based IoT system
or
. . .

Example: Semantics
(Directly Distinguished, Fully Interpreted)

$Pony \subseteq Horse$ ◄ "Each pony is a horse"

$m \in Mule$ ◄ "m is a mule"

$p \in Pony$ ◄ "p is a pony"

Italics font indicates individuals and their (*extensional*) sets

Asserted by an authority
or
Found by a sensor-based IoT system
or
. . .

Example: Semantics (Truth Including Inferred)

$$\text{Pony}(x) \Rightarrow \text{Horse}(x)$$

Mule(m)

Pony(p)

Horse(p)

Example: Semantics (Including Inferred, Fully Interpreted)

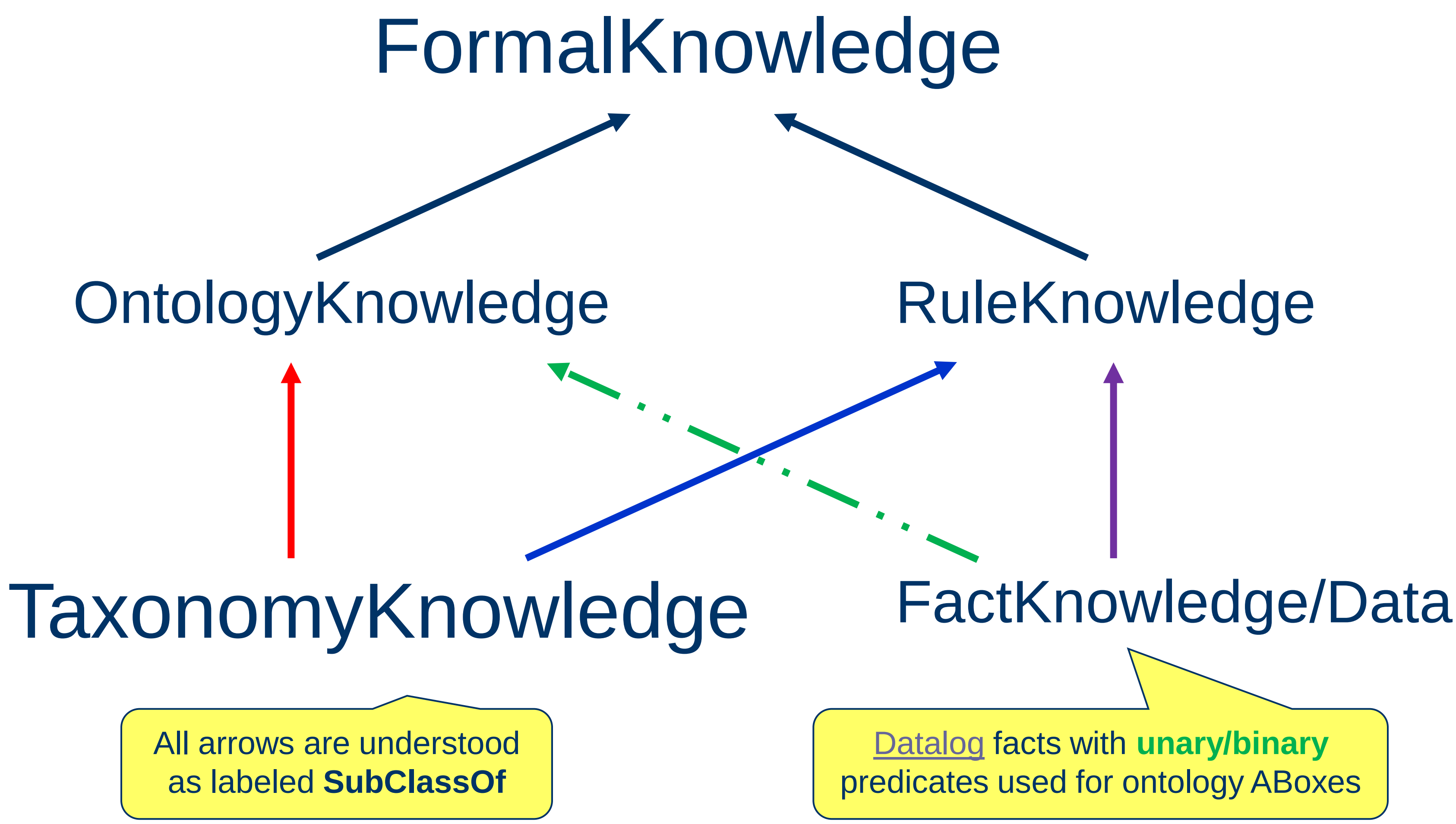
$$Pony \subseteq Horse$$

$$m \in Mule$$
$$p \in Pony$$
$$p \in Horse$$

Species of Formal Knowledge on the Web

Making distributed formal knowledge a universal commodity on the Web

Formal Knowledge as Ontologies or Rules



Taxonomy Knowledge: TBox (1)

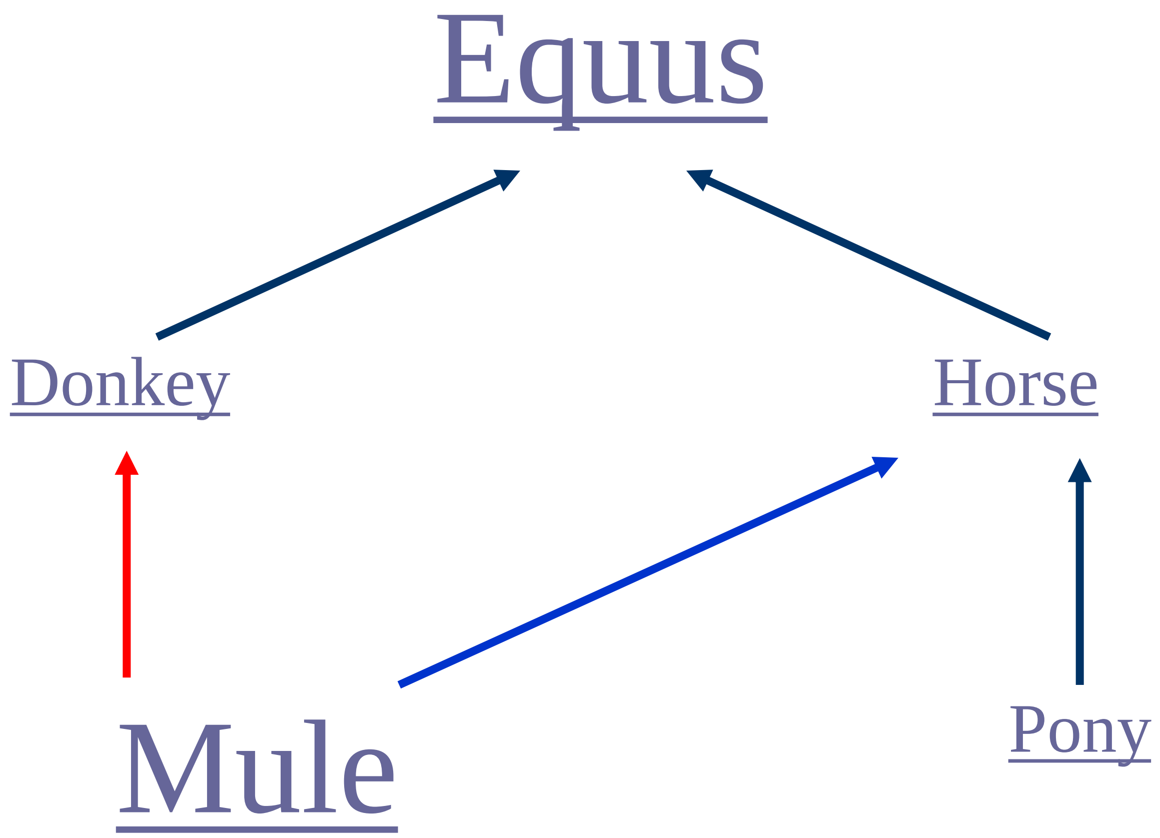
- Class hierarchies for conceptual classification
- Example: Above classification of FormalKnowledge
- Discover subsumptions/implications for inference;
  e.g., $\mathit{TaxonomyKnowledge} \subseteq \mathit{RuleKnowledge}$
  i.e., $\mathrm{TaxonomyKnowledge}(x) \Rightarrow \mathrm{RuleKnowledge}(x)$
- Thus allowing multiple parents (shown above):
  From trees to Directed Acyclic Graphs (DAGs)
  - Here, taxonomies as 'intersection' of ontologies and rules
- Realized several taxonomies in projects, including 'Computing' classification in FindXpRT and 'Tourism' classification in eTourPlan

Taxonomy Knowledge: TBox (2)

- With the metamodel about FormalKnowledge defined, it is instructive to separate the representation method (a taxonomy) from what is represented:
  - Earlier: FormalKnowledge, containing TaxonomyKnowledge
  - Now: A ‘folksonomy’ of Equus, containing Mule
- Structurally a subDAG of the FormalKnowledge taxonomy, but completely different content
- Again discover subsumptions/implications which enable inferences, e.g. about mules as horses;
  e.g., $Mule \subseteq Horse$
  i.e., $\mathrm{Mule}(x) \Rightarrow \mathrm{Horse}(x)$

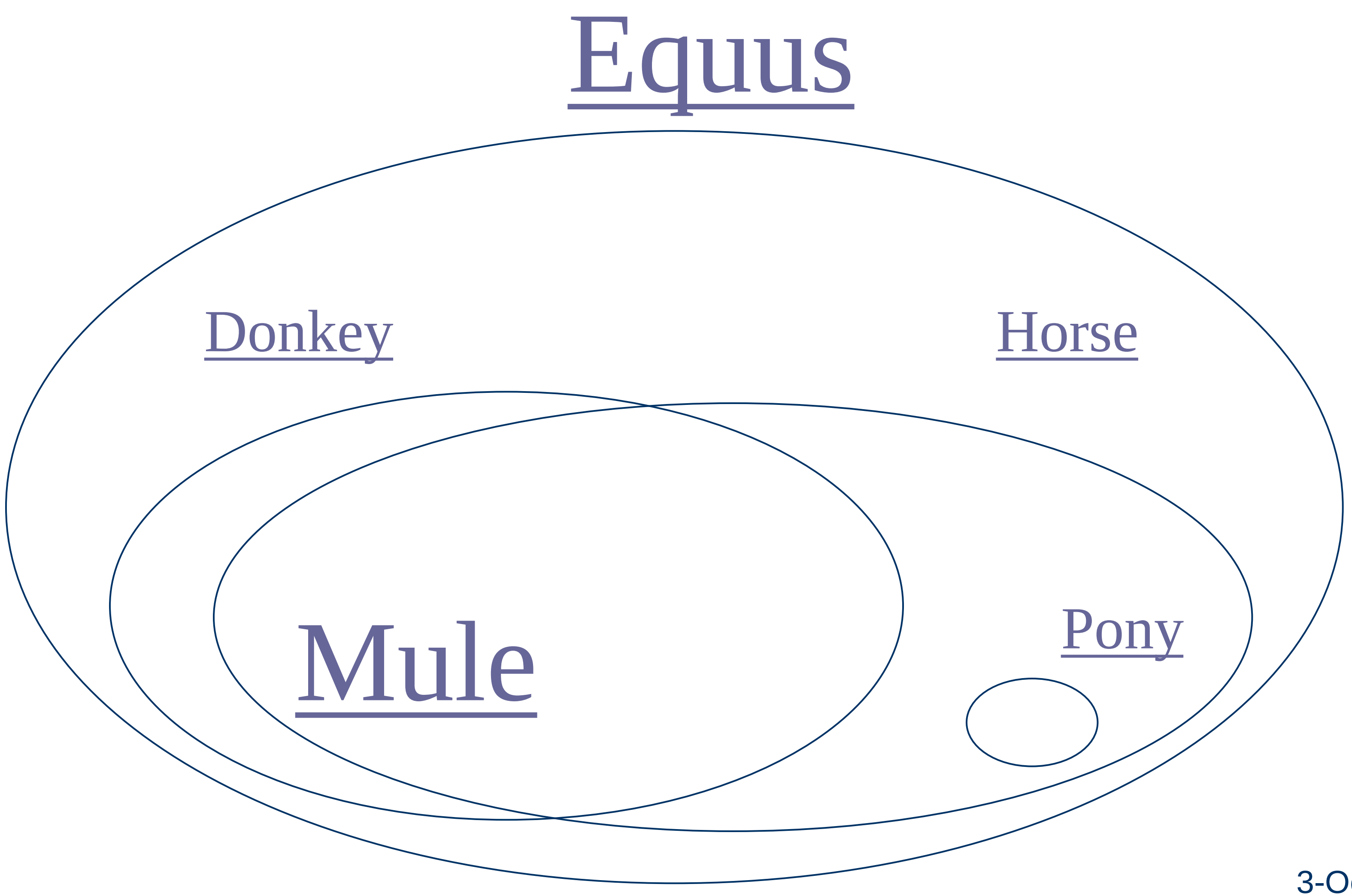
  Single-premise rules whose predicates have one and the same variable argument
- Thus also allowing multiple parents (shown below)
- But ‘commonsense’: Much simplified biologically!

Equi as Donkies or Horses: Visual (DAG)

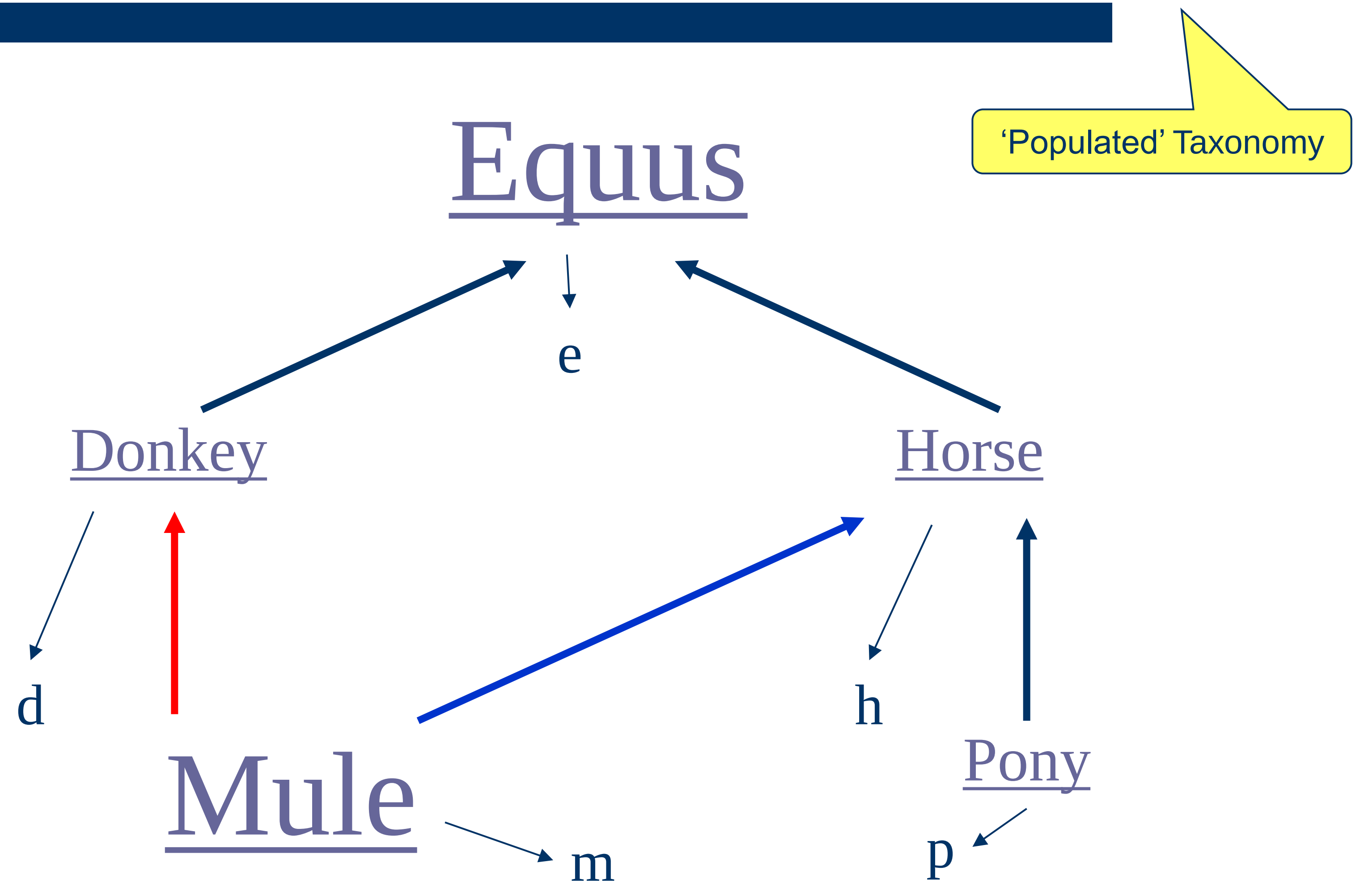


Species

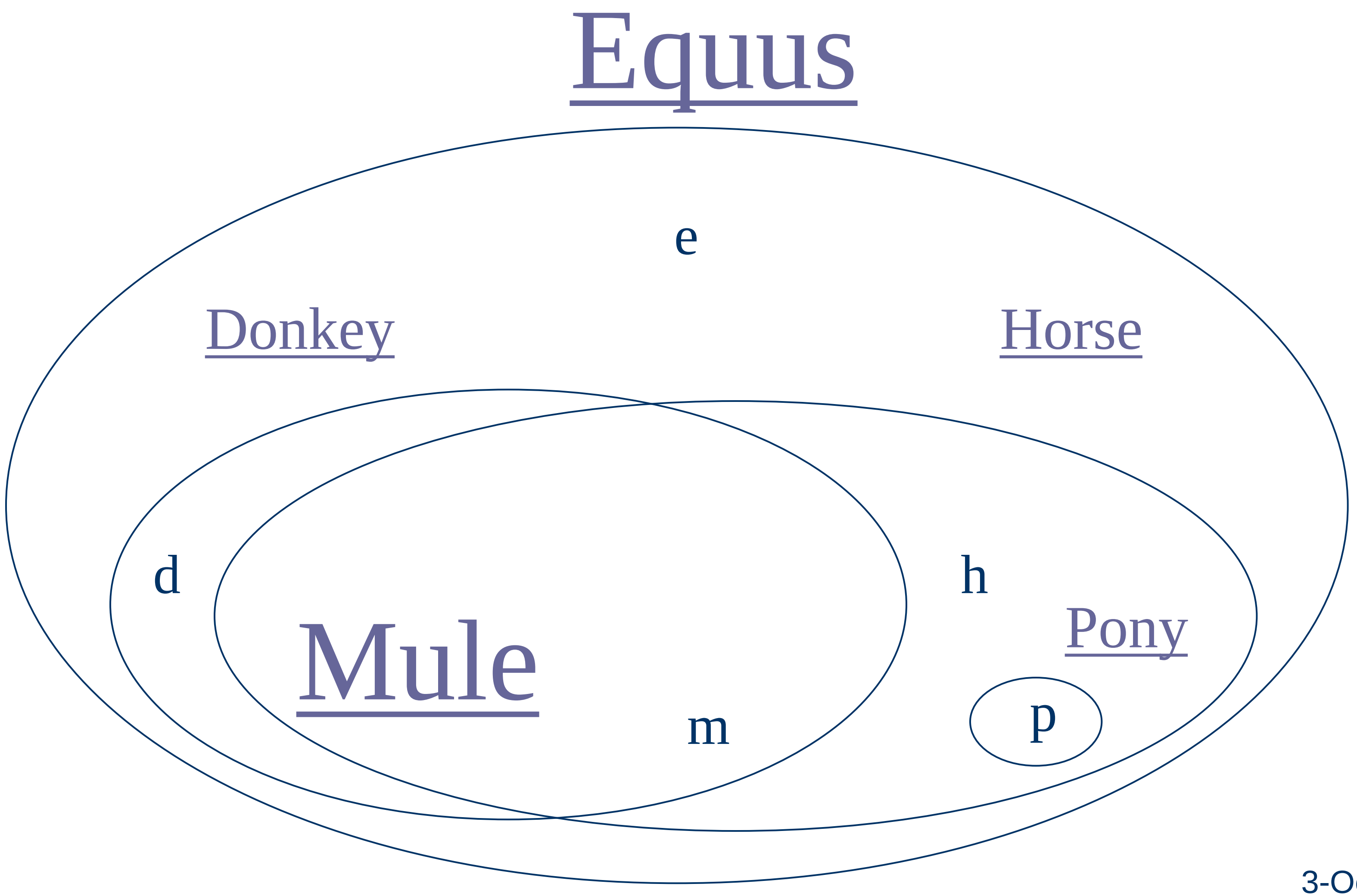
Equi as Donkies or Horses: Visual (Venn Diagram)



Equi as Donkies or Horses (DAG): ABox Asserting Instances d, e, h, m, p



Equi as Donkies or Horses (Venn): ABox Asserting Instances d, e, h, m, p



Equi as Donkies or Horses: Symbolic (1)

Semantics:			Rule syntax:		
Subsumptions			Implications		
Donkey	$\subseteq$	*Equus*	Donkey(x)	$\Rightarrow$	Equus(x)
Horse	$\subseteq$	*Equus*	Horse(x)	$\Rightarrow$	Equus(x)
Mule	$\subseteq$	*Donkey*	Mule(x)	$\Rightarrow$	Donkey(x)
Mule	$\subseteq$	*Horse*	Mule(x)	$\Rightarrow$	Horse(x)
Pony	$\subseteq$	*Horse*	Pony(x)	$\Rightarrow$	Horse(x)

Italics font indicates (*extensional*) sets

Normal font indicates (intensional) predicates

Equi as Donkies or Horses: Symbolic (2)

Ontology syntax: Classifications	Rule syntax: Implications
Donkey $\sqsubseteq$ Equus	Donkey(x) $\Rightarrow$ Equus(x)
Horse $\sqsubseteq$ Equus	Horse(x) $\Rightarrow$ Equus(x)
Mule $\sqsubseteq$ Donkey	Mule(x) $\Rightarrow$ Donkey(x)
Mule $\sqsubseteq$ Horse	Mule(x) $\Rightarrow$ Horse(x)
Pony $\sqsubseteq$ Horse	Pony(x) $\Rightarrow$ Horse(x)

Normal font indicates (intensional) classes

Normal font indicates (intensional) predicates

Equi as Donkies or Horses: Symbolic (3)

Logic rule syntax: Backward implications	Logic rule syntax: Forward implications
Equus(x) ⇐ Donkey(x)	Donkey(x) ⇒ Equus(x)
Equus(x) ⇐ Horse(x)	Horse(x) ⇒ Equus(x)
Donkey(x) ⇐ Mule(x)	Mule(x) ⇒ Donkey(x)
Horse(x) ⇐ Mule(x)	Mule(x) ⇒ Horse(x)
Horse(x) ⇐ Pony(x)	Pony(x) ⇒ Horse(x)

Species

Equi as Donkies or Horses: Symbolic (4)

Prolog rule syntax: Backward implications	Logic rule syntax: Forward implications
equus(X) :- donkey(X).	Donkey(x) $\Rightarrow$ Equus(x)
equus(X) :- horse(X).	Horse(x) $\Rightarrow$ Equus(x)
donkey(X) :- mule(X).	Mule(x) $\Rightarrow$ Donkey(x)
horse(X) :- mule(X).	Mule(x) $\Rightarrow$ Horse(x)
horse(X) :- pony(X).	Pony(x) $\Rightarrow$ Horse(x)

Upper-case (first) letter indicates (∀) variables;
so predicates are lower-cased

Letters *x*, *y*, and *z* often used as (∀) variables

Inference: Modus Ponens, Bottom-Up (Two 'Sequential' Applications)

TBox rules

equus(X) :- horse(X).

horse(X) :- pony(X).

ABox instance/fact (datum)

pony(p).

Bottom-up (➦) derivation, i.e. forward-chaining, realizes inheritance (via ':-' transitivity)

pony(p) ➦ horse(p) ➦ equus(p)

fact-to-fact

Inference: Modus Ponens, Top-Down (Two 'Sequential' Applications)

TBox rules

equus(X) :- horse(X).

horse(X) :- pony(X).

ABox instance/fact (datum)

pony(p).

Top-down (↪) reduction, i.e. backward-chaining, realizes inheritance (via ':-' transitivity)

equus(p) ↪ horse(p) ↪ pony(p) ↪ true

query-to-query

equus(W) ↪ horse(W) ↪ pony(W) ↪ true, W=p

Inference: Modus Ponens, Bottom-Up (Two 'Parallel' Applications)

TBox rules

equus(X) :- donkey(X).

equus(X) :- horse(X).

ABox instances/facts (data)

donkey(d).

horse(h).

Bottom-up (➦) derivation/inheritance

donkey(d) ➦ equus(d)

horse(h) ➦ equus(h)

Inference: Modus Ponens, Top-Down (Two 'Parallel' Applications)

TBox rules

equus(X) :- donkey(X).
equus(X) :- horse(X).

ABox instances/facts (data)

donkey(d).
horse(h).

Top-down (↪) reduction/inheritance

equus(d) ↪ donkey(d) ↪ true
equus(h) ↪ horse(h) ↪ true
equus(W) ↪ donkey(W) ↪ true, W=d
↪ horse(W) ↪ true, W=h

Ontology Knowledge

- Ontologies extend taxonomies by property hierarchies, ∀/∃-restricted properties, etc. of description logics
- Int'l standards:
  - ISO: Common Logic (CL 2, incl. CGs: Conceptual Graphs)
  - OMG: Ontology Definition Metamodel (ODM 1.1)
  - W3C: Web Ontology Language (OWL 2)
- $\text{Datalog}^{+/-}$ and Deliberation RuleML 1.02 allow to represent ontologies as (existential) rules, e.g. for Rule-Based Data Access and ΔForest (RBDA)
- Target representation for knowledge discovery (e.g. business intelligence/analytics) from instances
  - Background knowledge for further discovery

Fact Knowledge

- Facts (data) can be asserted in two paradigms:

Positional	**Slotted**
Relational-table (SQL) rows (column headers = signatures)	Object-centered instances (o-c directed labeled graphs)
XML elements	RDF triples / XML attributes
n-ary predicates (Prolog)	AI frames (F-logic)

- POSL and OO RuleML combine these paradigms; cross-paradigm translators enable interoperation
  - Used in projects SymposiumPlanner, WellnessRules2, PatientSupporter, and EnviroPlanner
- The paradigms and translators have been lifted to object-relational rules, as in PSOA RuleML

Rule Knowledge

- Rules generalize facts by making them conditional on other facts (often via chaining through further rules)
- Rules generalize taxonomies via multiple premises, n-ary predicates, structured arguments, etc.
- Two uses of rules – *top-down* (backward-chaining) and *bottom-up* (forward-chaining) – represented only once
- To avoid n^2–n *pairwise* translators:
  Int'l standards with 2n–2 *in-and-out* translators:
  - RuleML: Rule Markup Language (work with ISO, OMG, W3C, OASIS)
    - Deliberation RuleML 1.02 / Reaction RuleML 1.0 released as de facto standards
  - ISO: Common Logic (incl. CGs & KIF: Knowledge Interchange Format)
    - Collaboration on Relax NG schemas for XCL 2 / CL RuleML
  - OMG: Production Rules Representation (PRR), SBVR, and API4KP
  - W3C: Rule Interchange Format (RIF)
    - Gave rise to open-source and commercial RIF implementations
  - OASIS: LegalRuleML
- Target representation and background knowledge for discovery from facts: Inductive Programming

Ontology-Rule Synthesis: Hybrid and Homogeneous

- Hybrid combinations
  - Reuse existing ontology and rule standards
  - Allow rule conditions to refer to ontologies
  - Explored in projects:
    - Object Oriented RuleML: RDF Schema taxonomies
    - Datalog^{DL}: Datalog with Description Logics
- Homogeneous integrations
  - Merge ontologies and rules into a single representation
  - Explored in projects:
    - ALC$^{u}_{P}$: ALC/Datalog merger with safeness condition
    - Semantic Web Rule Language: OWL/RuleML merger as W3C Member Submission (http://scholar.google.ca/scholar?q=SWRL)
    - PSOA (Positional-Slotted, Object-Applicative) RuleML semantics allows taxonomic subclass relationships

RuleML Tools from the Semantic Technology Stack

Foundational and extended RuleML technology available online

Rule Responder: Reference Architecture for Distributed Query Engines

- Enables expert finding and query-based knowledge discovery in distributed virtual organizations
- Queries and answers exchanged in RuleML/XML
- Supported rule engines (int'l collaboration): Prova, OO jDREW, Euler, and DR-Device
- Based on the Mule Enterprise Service Bus
- Instantiated, e.g., in deployed SymposiumPlanner and prototyped WellnessRules2 / PatientSupporter
- Foundation for Master's projects on EnviroPlanner and SP-2012 at UNB. Also used in PhD projects in Fredericton, Berlin, Vienna, and Thessaloniki

TOOLS

Example of Semantic Wiki Page: Markup

(http://semanticweb.org/index.php?title=Rule_Responder&action=edit)

```
{{Tool
 | Name=Rule Responder
 | Homepage=http://responder.ruleml.org/
 | Affiliation=RuleML
 | Status=beta
 | Version=894
 | Release=May 13 2012
 | License=LGPL
 | Download=http://mandarax.svn.sourceforge.net/viewvc/mandarax/RuleResponder3/
}}
```

Metadata fact as object-centered instance of semantic template for Tool:
http://semanticweb.org/wiki/Template:Tool

Rule Responder is a tool for creating virtual organizations as multi-agent systems that support collaborative teams on the Semantic Web. It provides the infrastructure for rule-based collaboration between the distributed members of such a virtual organization. Human members of an organization are assisted by semi-autonomous rule-based agents, which use Semantic Web rules to describe aspects of their owners' derivation and reaction logic.

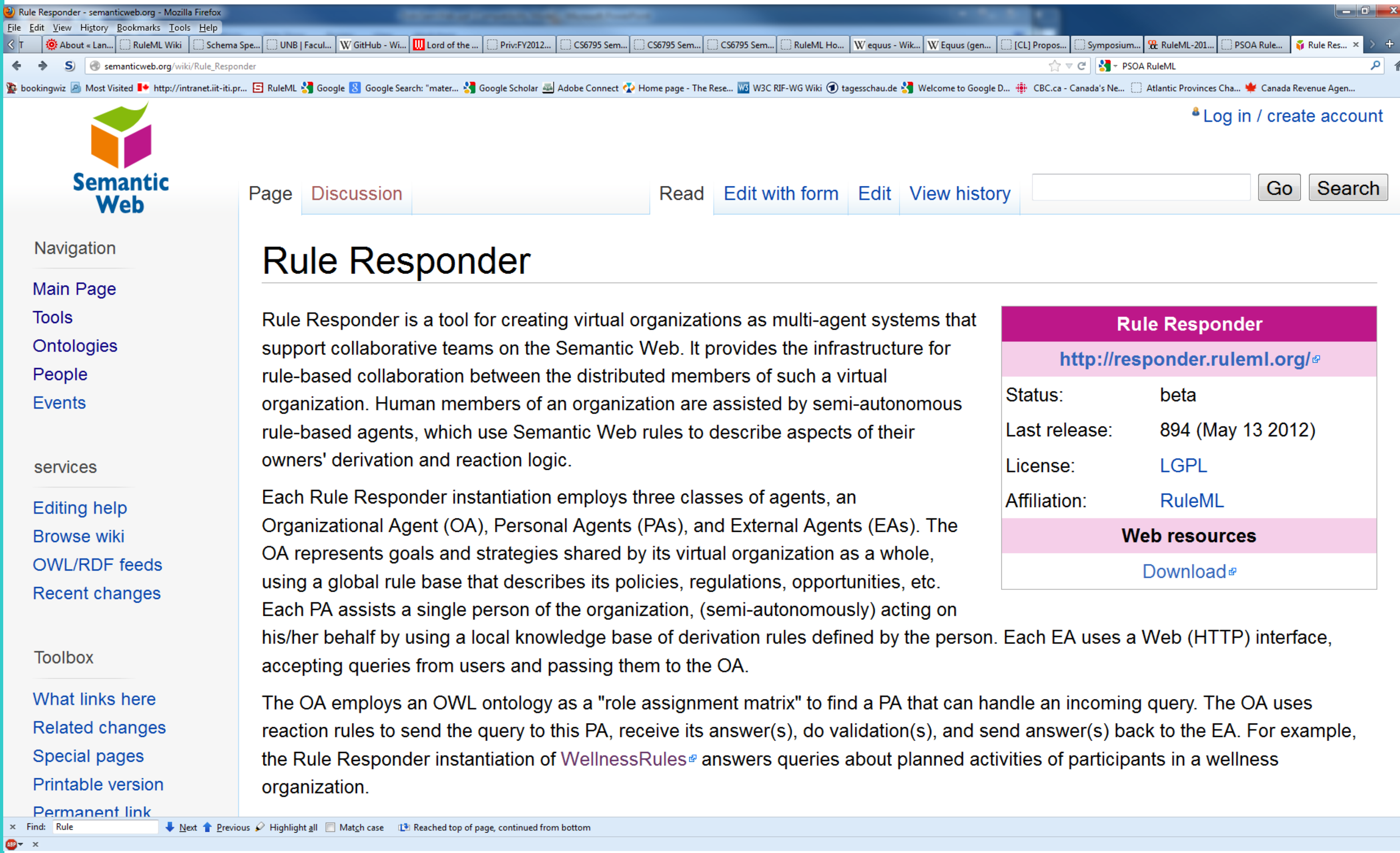
Each Rule Responder instantiation employs three classes of agents, an Organizational Agent (OA), Personal Agents (PAs), and External Agents (EAs). The OA represents goals and strategies shared by its virtual organization as a whole, using a global rule base that describes its policies, regulations, opportunities, etc. Each PA assists a single person of the organization, (semi-autonomously) acting on his/her behalf by using a local knowledge base of derivation rules defined by the person. Each EA uses a Web (HTTP) interface, accepting queries from users and passing them to the OA.

The OA employs an OWL ontology as a "role assignment matrix" to find a PA that can handle an incoming query. The OA uses reaction rules to send the query to this PA, receive its answer(s), do validation(s), and send answer(s) back to the EA. For example, the Rule Responder instantiation of [http://ruleml.org/WellnessRules/RuleResponder.html WellnessRules] answers queries about planned activities of participants in a wellness organization.

[[Category:Semantic agent system]]
[[Category:Reasoner]]

Member of two Tool subclasses:
http://semanticweb.org/wiki/Category:Semantic_Web_tool

Example of Semantic Wiki Page: Rendered (http://semanticweb.org/wiki/Rule_Responder)



Conclusion

- Conceived a joint semantics of objects+relations and ontologies+rules for distributed knowledge querying
  - Developed standard languages, compatible engines, and reference architectures (visualized with Grailog)
- Used to study expert knowledge and communication topologies of virtual organizations
  - Gradual formalization as distributed knowledge and agent-mediated communication (cf. Rule Responder)
- Applied to knowledge representation and inferencing on the Social Semantic Web
  - Use cases in symposium organization, wellness groups, patient support, and environmental querying