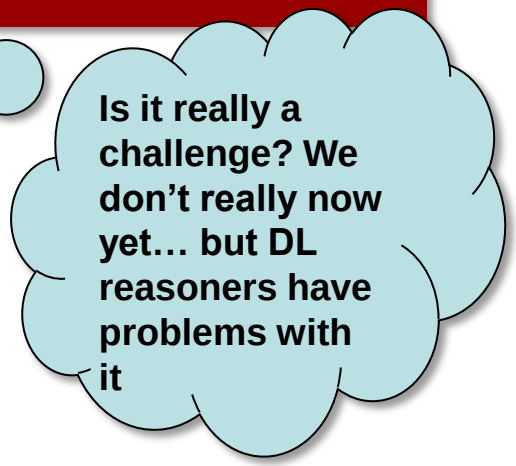
AURA

Bio_KB_101: A Challenge for TPTP First-Order Reasoners (?)

Vinay K. Chaudhri
Michael A. Wessel
Stijn Heymans

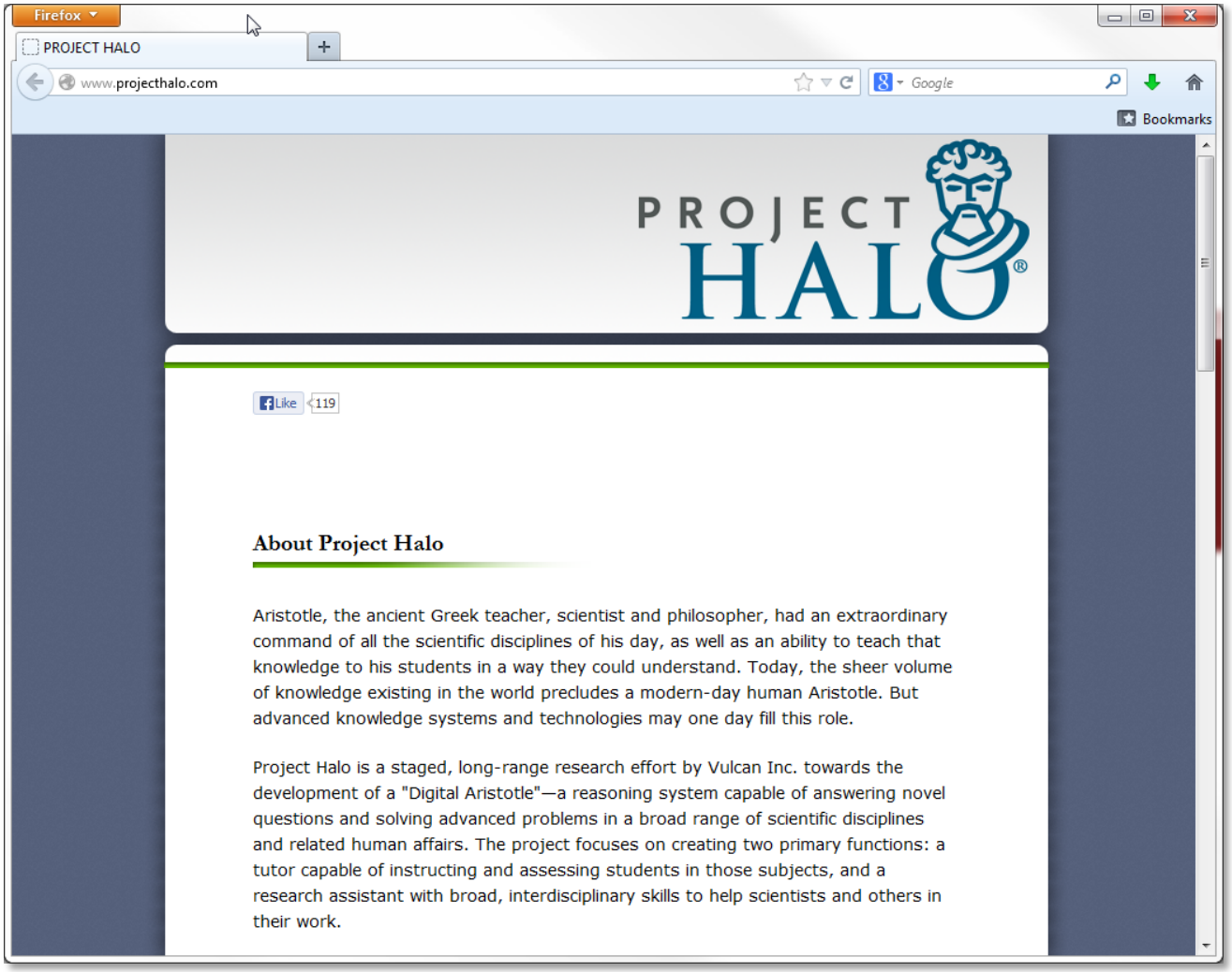


SRI International

PROJECT HALO

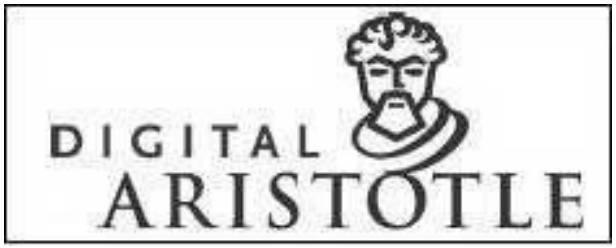
Acknowledgment

- This work has been funded by Paul Allens' Vulcan Inc.
  http://www.vulcan.com
  http://www.projecthalo.com



Background: The Digital Aristotle, Project Halo, and AI^2

AURA



Digital Aristotle – a tutoring and reasoning sstem capable of teaching, answering novel questions and solving advanced problems in a broad range of scientific disciplines

Project Halo – Vulcan's phased, long-range past research effort to build the Digital Aristotle, with 3 areas of concentration:

- *AURA / Inquire:* A question-answering **biology** text (SRI)
- *SMW*: Low-cost knowledge from the public
- *SILK*: **S**emantic **I**nferencing on **L**arge **K**nowledge - a new semantic web rule language

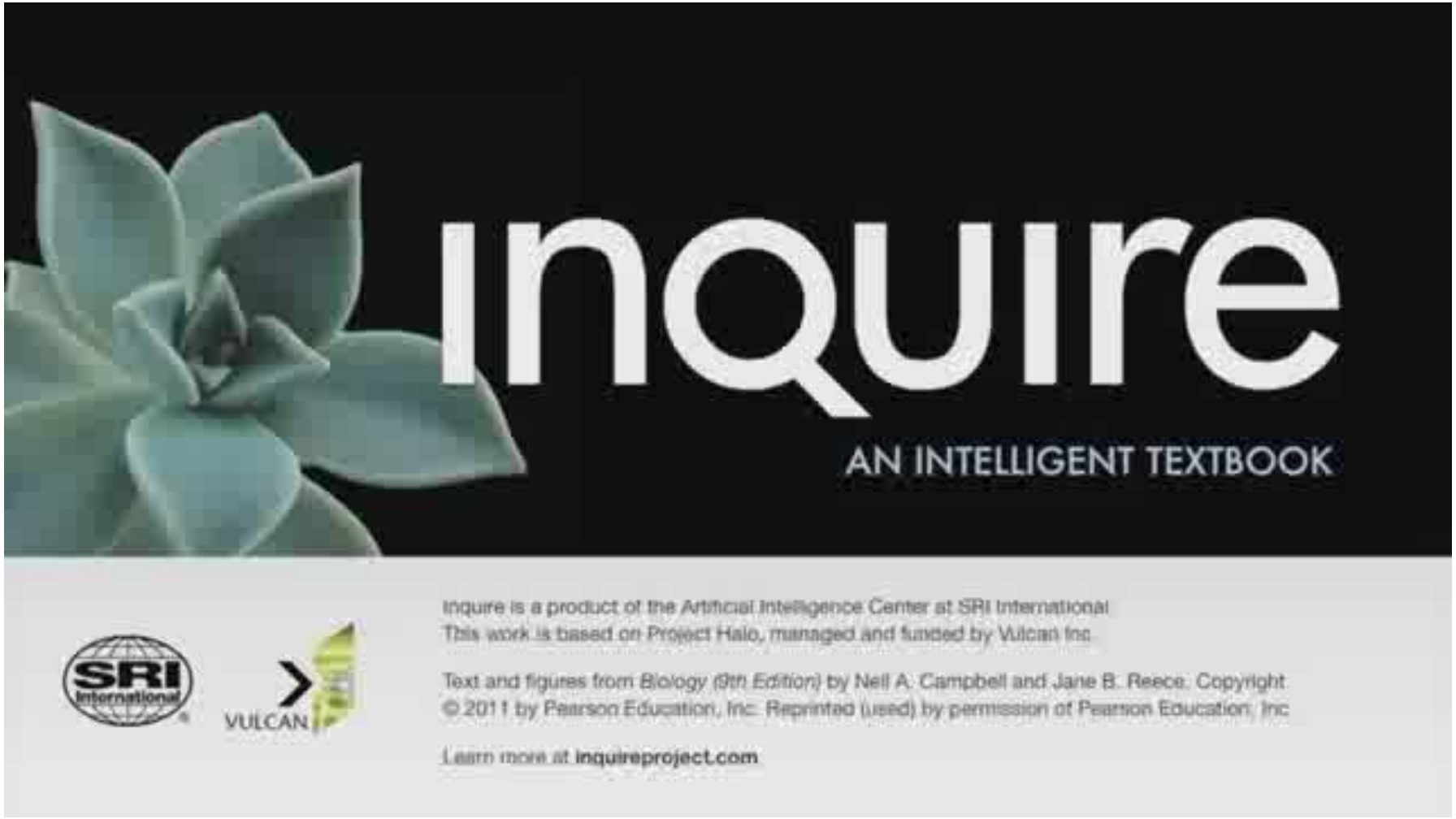
Currently, Vulcan is in the process of defining its future direction for AI research (AI^2). SRI is looking at marketing opportunities for the developed technology.

AI^2 – Sponsors conferences, prizes, competitions, and the construction of large public knowledge bases

PROJECT HALO

Winner of the 2012 AAAI Video Award

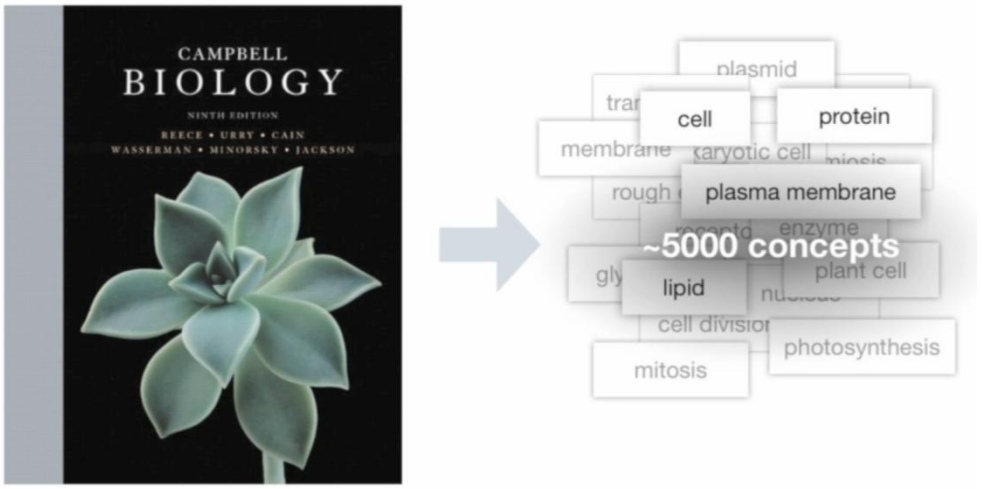
AURA



PROJECT HALO

The Underlying Knowledge Base

AURA



- A team of **biologists** is using graphical editors to curate the KB from the textbook, using a **sophisticated knowledge authoring process** (see below) http://dl.acm.org/citation.cfm?id=1999714

- The KB is a valuable asset: it contains 11.5 man years of biologists, and estimated 5 (2 Univ. Texas + 3 SRI) years for the upper ontology (CLib)

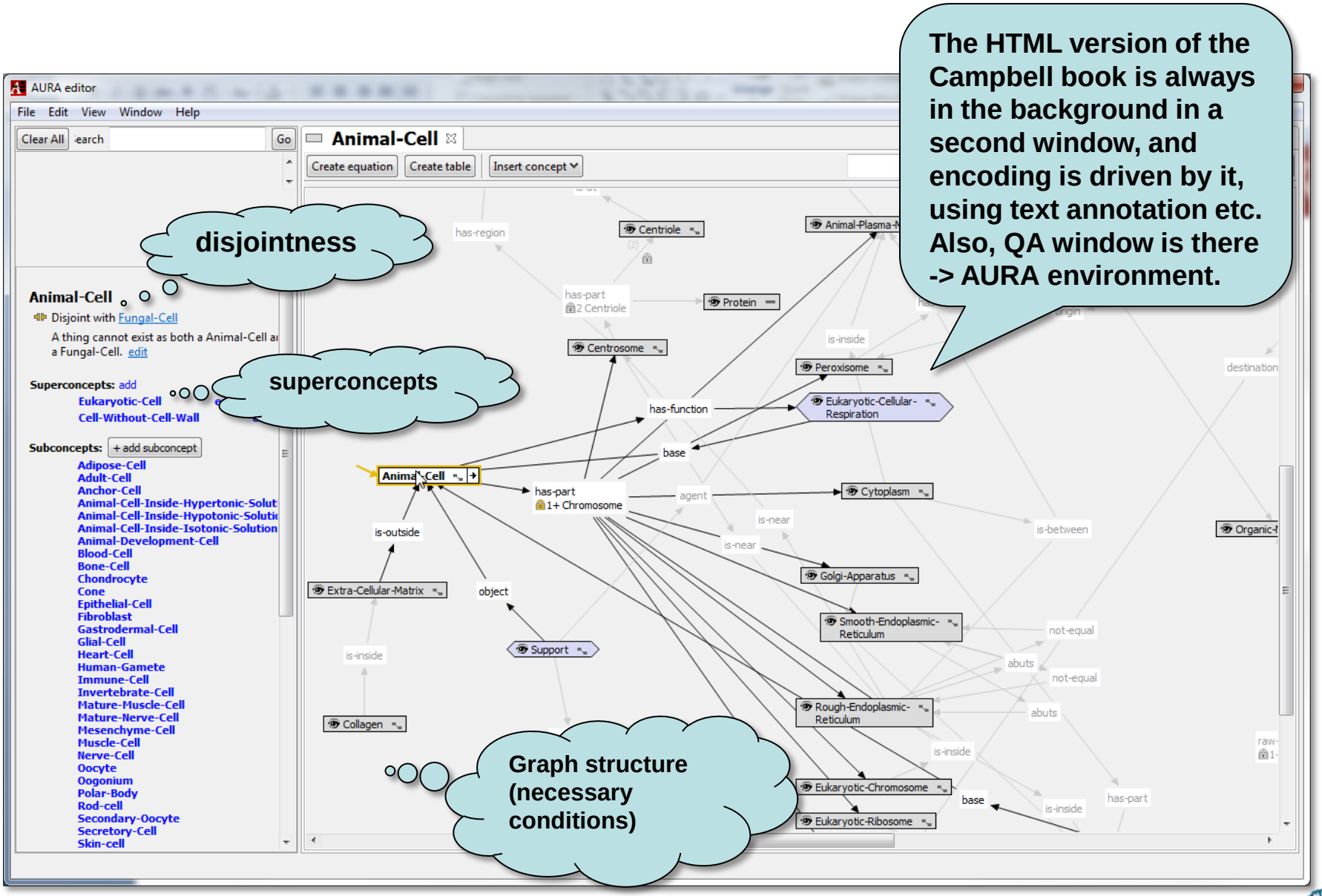
- Vulcan and SRI are giving this asset free of charge to the research community (subject to a research license agreement): http://www.ai.sri.com/halo/halobook2010/exported-kb/biokb.html

- The KB has non-trivial graph structure (unlike some medical ontologies)

PROJECT HALO

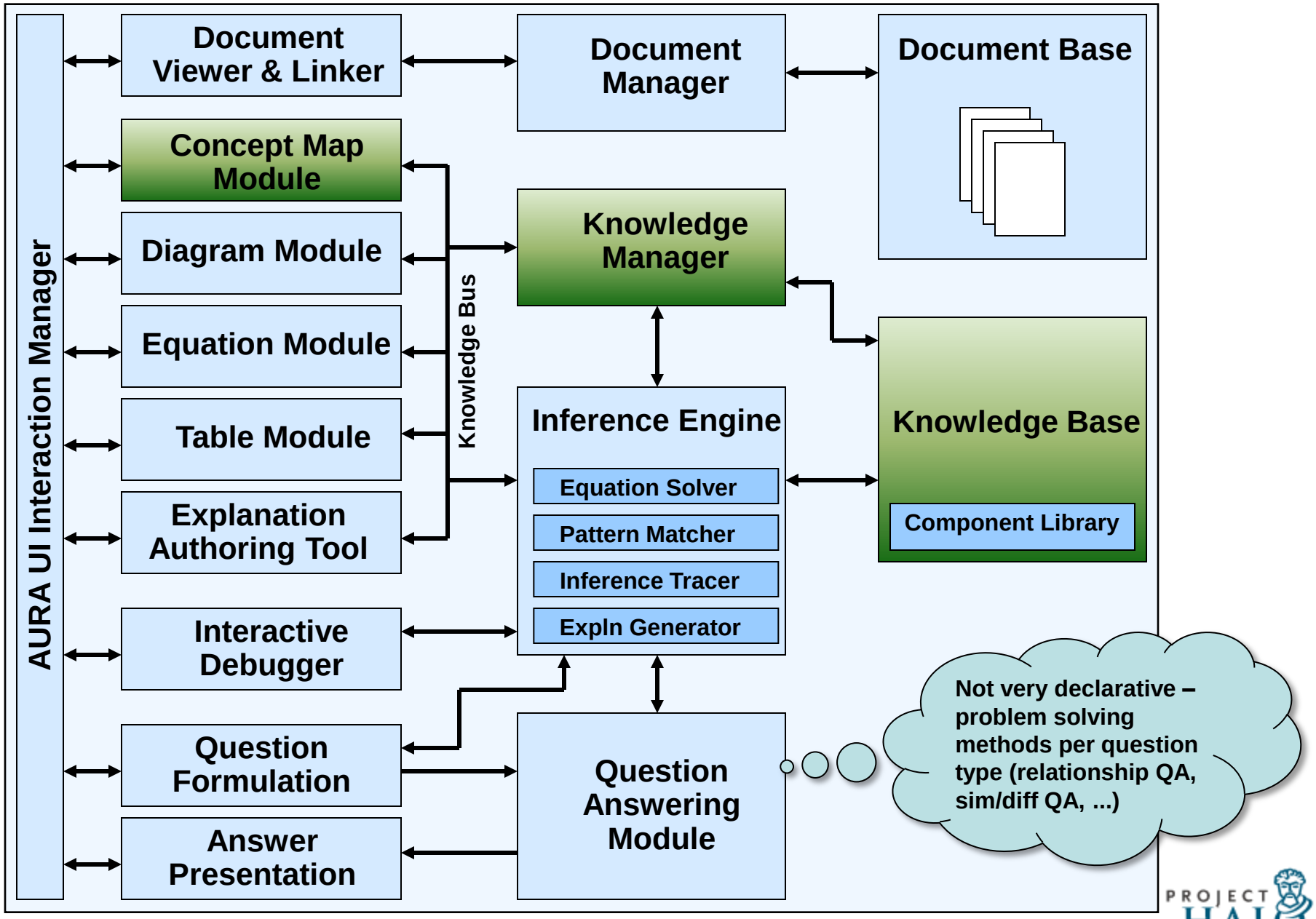
AURA Graphical Knowledge Editor

AURA



PROJECT HALO

AURA Architecture



Knowledge Authoring Process

AURA

1) Determining Relevance and Pre-Planning

Pre-planning	Determining relevance of sentences Status labeling per sentence: relevant, irrelevant

2) Reaching Consensus

Universal Truth authoring, Concept chosen	QA check

3) Encoding Planning

Group common UTs, Identify KR/KE issues, Identify already encoded, Write how to encode	Planning, QA check Status Labeling: Encoding Complete, KR Issue (closed)

4) Encoding

Encode, File KR JIRA issues	QA check Status Labeling: Encoding Complete, KE Issue (closed)

5) Key Term Review

KR evaluated by modeling expert and SME, Encoder makes changes	KR evaluated by modeling expert and SME QA check

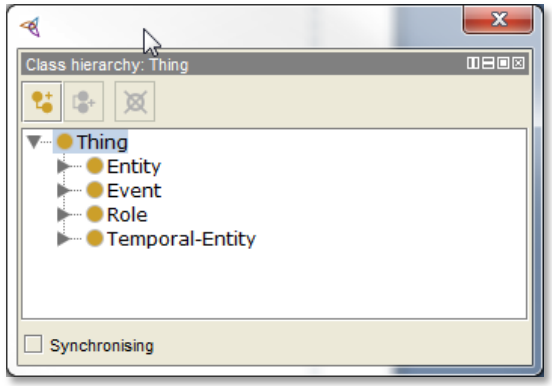
6) Question-Based Testing

Use Minimal Test Suite, File reasoning JIRA issues, Encoder fills KB gaps	QA check with screenshots of 'Passing' comparison and relationship questions

Knowledge Authoring Process

AURA

Expressive Means Used in AURA



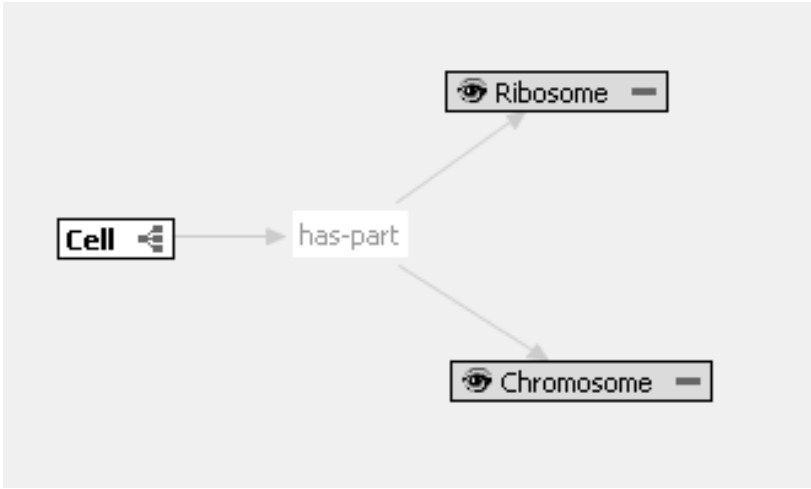
- Classes (concepts) in a class hierarchy
  - multiple inheritance
  - top classes below Thing:
    Entity (Cell), Event (Diffusion), Role (Nutrient)
  - disjointness
  - necessary and sufficient conditions (“triggers”)
    GRAPH STRUCTURED DESCRIPTIONS (NOT TREES)
  - (tables, equations, descriptions / annotations, …)
- Relations and attributes (properties)
  - domain, range and (inverse) functionality
  - transitivity
  - converse
  - hierarchy
  - composition and qualified composition
  - qualified number restrictions (a là OWL2) in classes
- Upper Ontology Clib: arbitrary “First-Order Axioms” in KM
- Biologists can only model CMaps, superclasses, disjointness axioms, but cannot change CLib, nor define new relations

Illustration of Bio Concept and Clib Axiom in KM

AURA

```
(_Cell1172 has
  (has-part (_Ribosome1180
             _Chromosome1179))
  (instance-of (Cell))
  (prototype-participants (_Ribosome1180
                           _Chromosome1179
                           _Cell1172))
  (prototype-participant-of (_Cell1172))
  (prototype-of (Cell))
  (prototype-scope (Cell)))

(_Ribosome1180 has
  (instance-of (Ribosome))
  (is-part-of (_Cell1172))
  (prototype-participant-of (_Cell1172)))

(_Chromosome1179 has
  (instance-of (Chromosome))
  (is-part-of (_Cell1172))
  (node-coordinate ((:pair 165 660)))
  (prototype-participant-of (_Cell1172)))
```

KM Prototype



KM First-Order Axiom

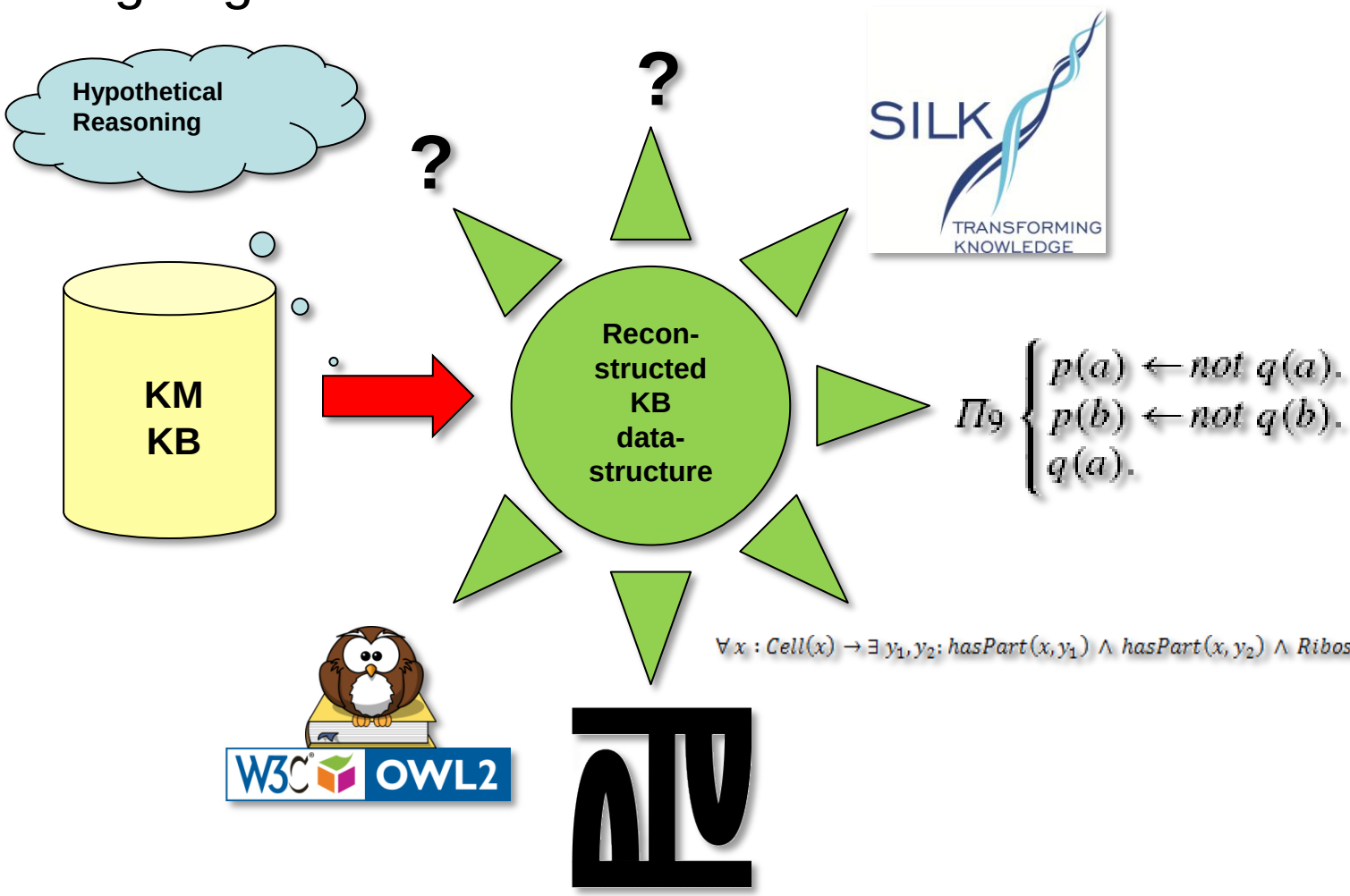
```
(Move has
       (superclasses    (Action)))

(every Move has
       (object ((a Spatial-Entity)
                (excluded-values
                       (the origin of Self)
                       (the destination of Self)
                       (the away-from of Self)
                       (the toward of Self)
                       (the path of Self)
                       (the site of Self)))))
```

PROJECT HALO

From KM to FOPL to <name your logic>

AURA

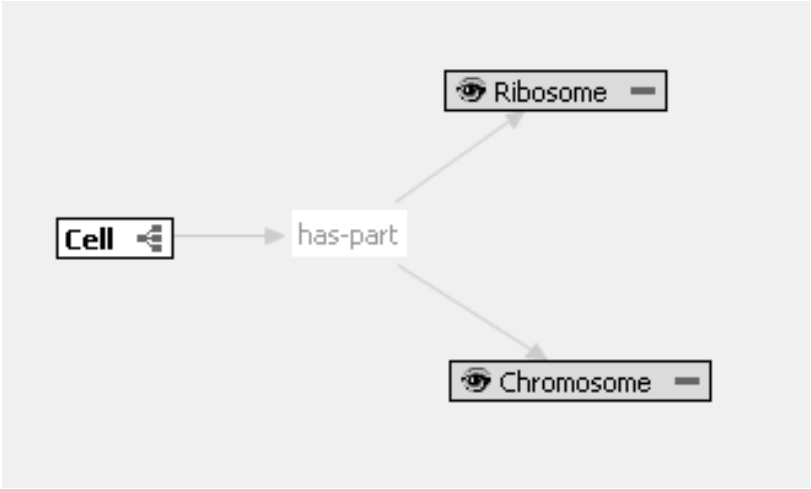
- The **logical reconstruction** of the KM KB turns out to be challenging, due to some unsound default reasoning going on there



PROJECT HALO

Reconstructed KB in FOPL

Every cell has a ribosome part and a chromosome part



$$\forall x : Cell(x) \rightarrow \exists y_1, y_2 : hasPart(x, y_1) \wedge hasPart(x, y_2) \wedge Ribosome(y_1) \wedge Chromosome(y_2)$$

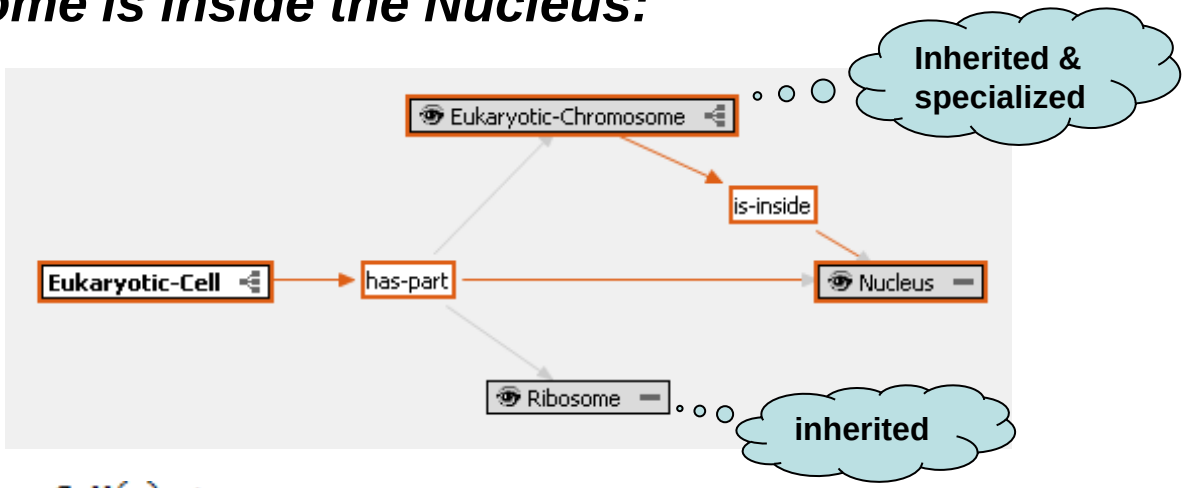
- However, what we really need is this skolemized version, so that classes that refer to Cell can refer to its Ribosome and Chromosome by means of the Skolem functions:

$$\forall x : Cell(x) \rightarrow hasPart\left(x, f_{Cell}\#1(x)\right) \wedge hasPart\left(x, f_{Cell}\#2(x)\right) \wedge Ribosome\left(f_{Cell}\#1(x)\right) \wedge Chromosome\left(f_{Cell}\#2(x)\right)$$

Skolem Function Inheritance and Equality

AURA

- ***Every Eukaryotic-Cell is a Cell***
- ***Every Eukaryotic-Cell has part a Eukaryotic-Chromosome, a Ribosome, and a Nucleus, such that the Eukaryotic-Chromosome is inside the Nucleus:***



$$\forall x : EukaryoticCell(x) \rightarrow Cell(x) \ \wedge$$
$$hasPart\left(x, f_{ECell}\#1(x)\right) \wedge hasPart\left(x, f_{ECell}\#2(x)\right) \wedge hasPart\left(x, f_{ECell}\#3(x)\right) \wedge$$
$$EukaryoticChromsome\left(f_{ECell}\#3(x)\right) \wedge Nucleus\left(f_{ECell}\#1(x)\right) \wedge Ribosome\left(f_{ECell}\#2(x)\right) \wedge$$
$$isInside\left(f_{ECell}\#3(x), f_{ECell}\#1(x)\right)$$
$$f_{ECell}\#3(x) = f_{Cell}\#2(x), \qquad f_{ECell}\#2(x) = f_{Cell}\#1(x)$$

Often, those equalities are NOT explicit in the KM KB,
but they need to be ***reconstructed*** by a special algorithm.

Also, the equalities can describe “node unifications”.

PROJECT HALO

TPTP Export Illustration

```
fof(a11860,axiom,(
  ! [X, Y] :
    ( ( has_part(X, Y) )
     =>
      ( tangible_entity(Y) ) ))).

fof(a11861,axiom,(
  ! [X, Y] :
    ( ( has_part(X, Y) )
     =>
      ( tangible_entity(X) ) ))).

fof(a11862,axiom,(
  ( ( has_part(X, Y)
    & has_part(Z, Y) )
   =>
    ( X=Z ) ))).

fof(a11863,axiom,(
  ! [X, Y] :
    ( ( has_part(X, Y) )
     =>
      ( has_structure(X, Y)
        & related_to(X, Y)
        & has_part_or_unit(X, Y)
        & is_part_of(Y, X) ) ))).

fof(a12942,axiom,(
  ! [X, Y, Z] :
    ( ( has_part_or_unit(X, Y)
      & element(Y, Z)
      & tangible_entity(X)
      & aggregate(Y)
      & tangible_entity(Z) )
     =>
      ( has_part_star(X, Z) ) ))).
```

```
fof(a13502,axiom,(
  ! [X] :
    ( ( cell(X) )
     =>
      ( original_name(X, "Cell")
        & description(X, "The basic unit from which living organisms
are made, consisting of an aqueous solution of organic molecules
enclosed by a membrane.  All cells arise from existing cells, usually
by a process of division into two.  (Alberts:ECB:G-3).")
        & class2words(X, "cell")
        & living_entity(X)
        & ribosome(fn_cell_1(X))
        & chromosome(fn_cell_2(X))
        & has_part(X, fn_cell_2(X))
        & has_part(X, fn_cell_1(X)) ) ))).



fof(a13504,axiom,(
  ! [X] :
    ( ( eukaryotic_cell(X) )
     =>
      ( original_name(X, "Eukaryotic-Cell")
        & class2words(X, "eukaryotic cell")
        & class2words(X, "eukaryotic-cell")
        & cell(X)
        & nucleus(fn_eukaryotic_cell_1(X))
        & ribosome(fn_eukaryotic_cell_2(X))
        & eukaryotic_chromosome(fn_eukaryotic_cell_3(X))
        & has_part(X, fn_eukaryotic_cell_1(X))
        & is_inside(fn_eukaryotic_cell_3(X), fn_eukaryotic_cell_1(X))
        & has_part(X, fn_eukaryotic_cell_3(X))
        & has_part(X, fn_eukaryotic_cell_2(X))
        & fn_eukaryotic_cell_3(X)=fn_cell_2(X)
        & fn_eukaryotic_cell_2(X)=fn_cell_1(X) ) ))).
```

KB Stats

AURA

Regarding Class Axioms:

# Classes	# Relations	# Constants	Avg. # Skolems / Class	Avg. # Atoms / Necessary Condition	Avg. # Atoms / Sufficient Condition
6430	455	634	24	64	4

# Constant Typings	# Taxonomical Axioms	# Disjointness Axioms	# Equality Assertions	# Qualified Number Restrictions
714	6993	18616	108755	936

Regarding Relation Axioms:

# DRAs	# RRAs	# RHAs	# QRHAs	# IRAs	# 12NAs / # N21As	# TRANS + # GTRANS
449	447	13	39	212	10 / 132	431

Regarding Other Aspects:

# Cyclical Classes	# Cycles	Avg. Cycle Length	# Skolem Functions
1008	8604	41	73815

PROJECT HALO

Why Might It Be Challenging?

AURA

- KB contains
  - graph structured descriptions
  - sufficient conditions
  - plenty of cycles
  - qualified number restrictions
  - transitive relations
  - almost arbitrary composition axioms of the form
    $\forall x, y, z : R(x,y) \wedge S(y,z) \Rightarrow T(x,z)$

-> neither tree- nor finite model property,
reasoning with the full KB is likely to be undecidable
(KM doesn't really do logical reasoning with it)

- Subsets / fragments of it might be decidable (prefix classes)
- Description logic / OWL reasoners have problems even with small fragments of it
- It may contain yet undiscovered inconsistencies

PROJECT HALO

Why Did We Submit it to KINAR?

AURA

- Among the translation we have, the FOPL translation is the most truthful / complete one (OWL etc. is lossy)
  - we want to apply FOPL reasoners
  - we want to be more declarative
  - we want to engage with the research community on first-order reasoning
  - we want to promote the KB, which is a valuable asset
- What are simple reasoning tasks we care about?
  - check consistency
    - we have successfully used Protégé 4.2 and Fact++ to debug simple inconsistencies resulting from interactions between disjointness, domain and range restrictions, and taxonomic axioms
  - find implicit subclasses
  - computation of (inferred) slot fillers and conjunctive query answering
- More complex reasoning tasks for QA
  - finding relationships
  - sim/diff (sim is similar to computation of a LCS in DLs)
  - …

PROJECT HALO

http://www.ai.sri.com/halo/halobook2010/exported-kb/biokb.html

Thank you!

AURA Team in 2011

Points for the Discussion

AURA

- Which TPTP reasoners should we start with?
- The KB contains logical inconsistencies
  - para consistent reasoning?
- How can we define interesting reasoning problems more declaratively
  - e.g., relationship question answering
  - some require unsound reasoning
    (e.g., going to subclasses and looking up information there)
  - those unsound inferences are desired by the Biologists
- How can we leverage and promote the KB?
  - what other KB applications might be interesting besides reasoner benchmarks

PROJECT HALO

Backup Material – One More Video

AURA

PROJECT HALO

Optional – Simple Demo

AURA

PROJECT HALO