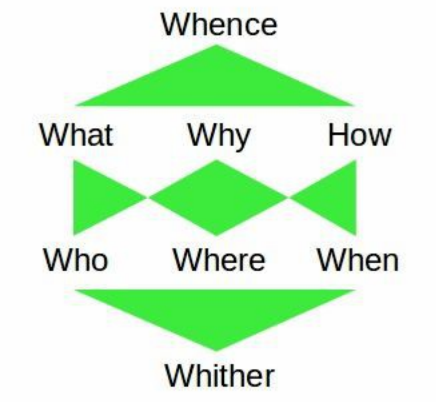


Knowledge Graphs

For Language, Logic, Data, Reasoning

John F. Sowa

12 February 2020

Knowledge Representation

1900 to 1959: Data representation and formatting.

- **From punched cards to FORTRAN, LISP, COBOL, Algol.**

1960s: Multiple concurrent programs that use the same data.

- **Data structures, databases, locking, virtual memory.**
- **Theorem provers, formal semantics, specification languages.**

1970s: DB wars, conceptual schema, expert systems.

1980s: Knowledge bases, object-oriented systems. *

1990s: Ontologies, statistics, machine translation.

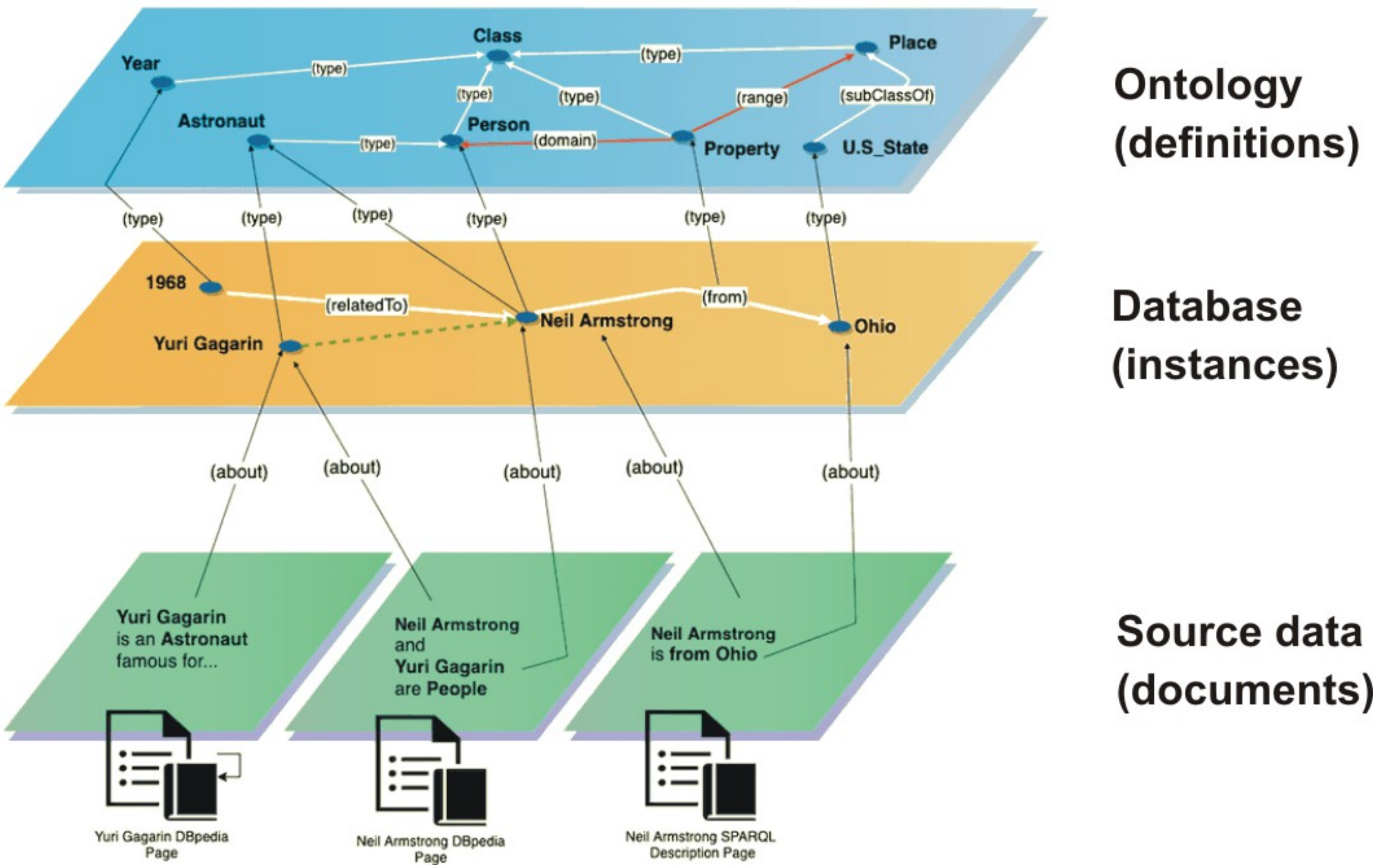
2000s: Semantic Web, large ontologies, need for standards.

2010s: BIG Data, Deep NNs, but no universal ontology.

2020s: What can knowledge graphs do? How? Why?

*** For more detail from 1980 to the present, see http://www.jfsowa.com/ikl**

Knowledge Graphs

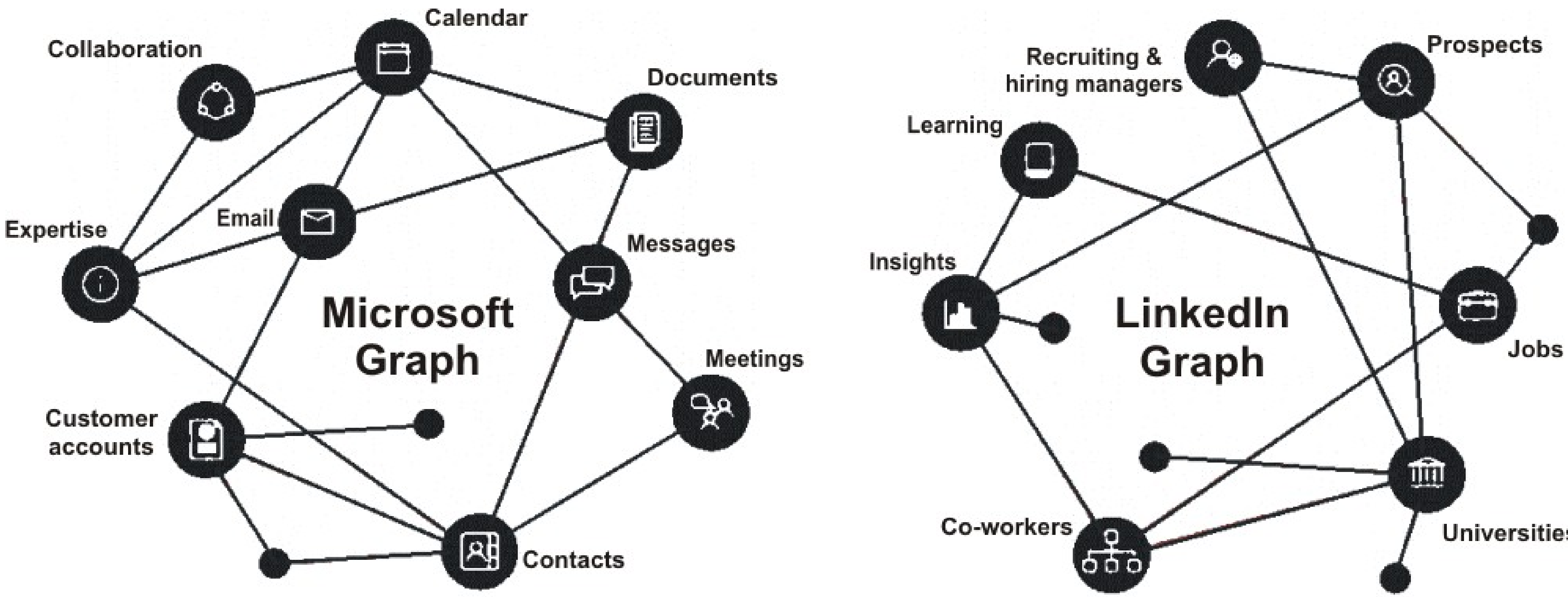


Knowledge graphs may be used in a variety of ways. *

- **Ontology: Specifying classes, properties, and relations.**
- **Database: Storing Big Data about anything and everything.**
- **Source data: Mapping language to a computable notation.**

*** Diagram by Kingsley Idehen,** https://medium.com/@kidehen/heres-why-9d278d5aa725

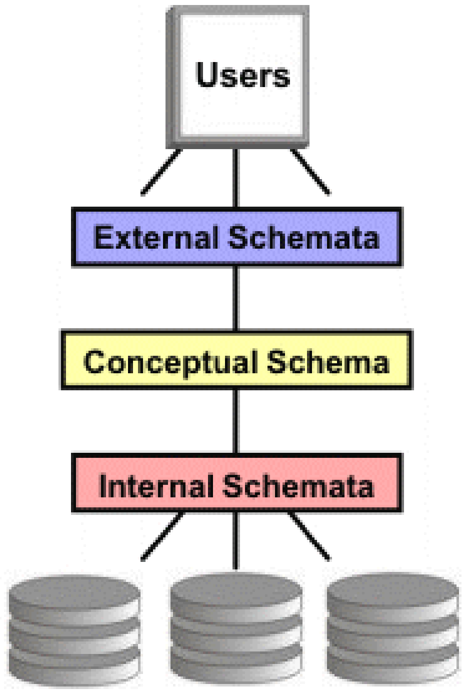
KGs for Microsoft and LinkedIn



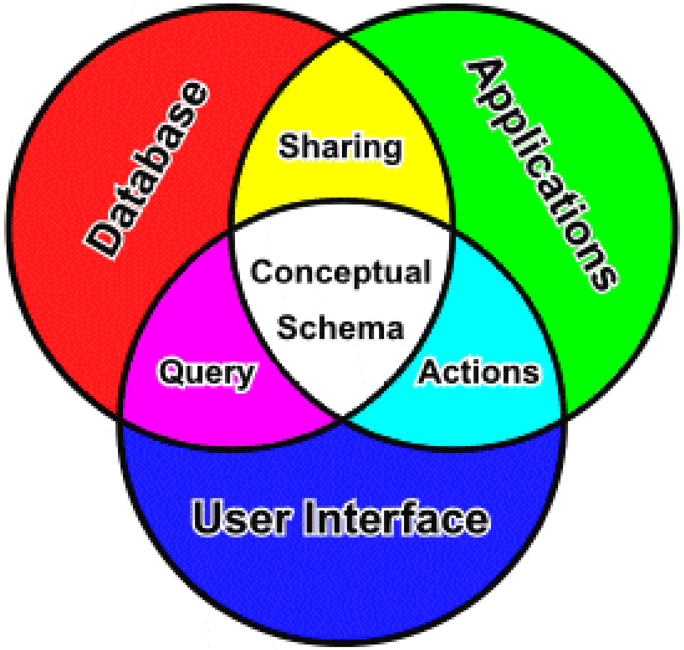
Serve the same purpose as a DB conceptual schema.

- **The labels on the nodes may be used to describe instances in a DB.**
- **But KGs are more informal – fewer constraints on the sources of data, the uses of the graphs, and the relationships among the nodes.**
- **Diagram adapted from "Building The LinkedIn Knowledge Graph."**

The Conceptual Schema



ANSI-SPARC Architecture

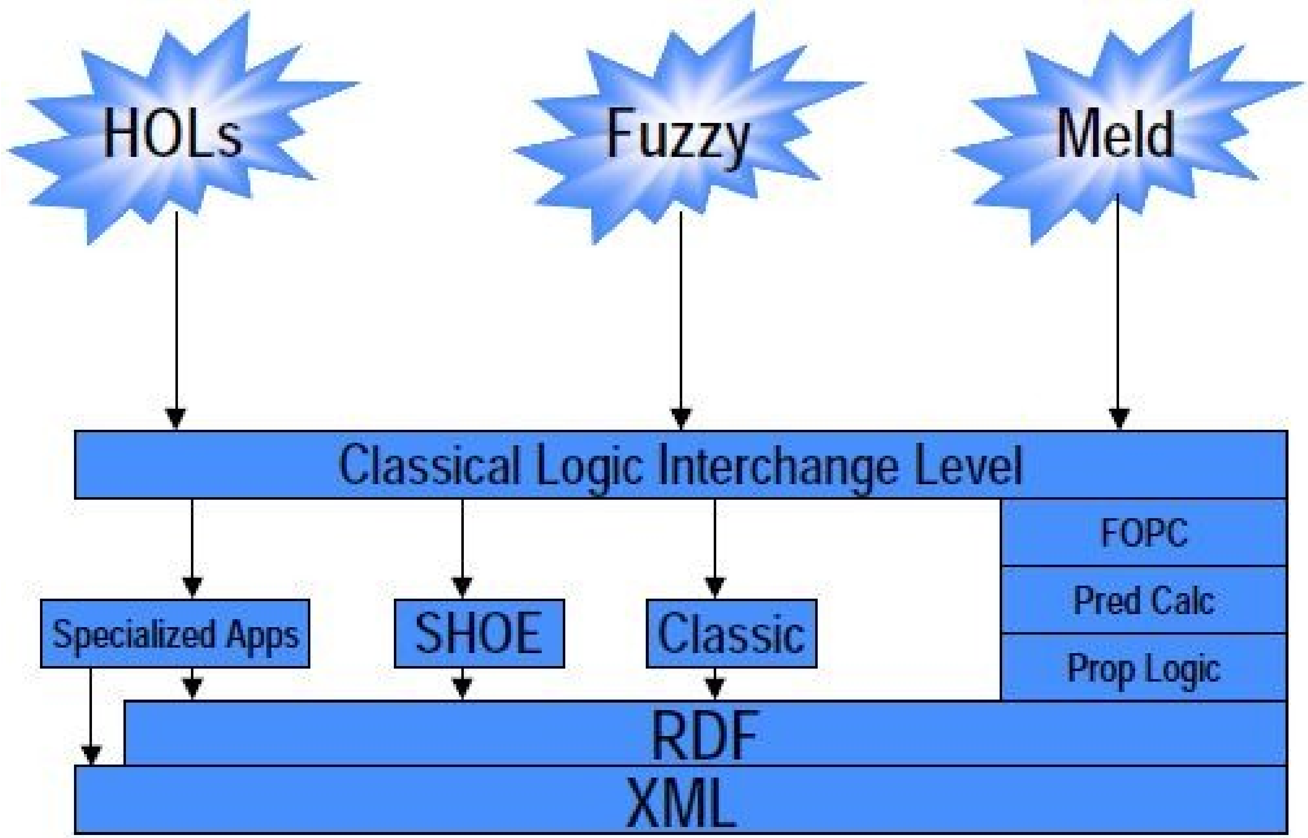


An Application System

Shared ontology to resolve the database wars of the 1970s.

- **Early debates led to an ANSI technical report in 1979.**
- **Further discussions led to an ISO technical report in 1987.**
- **Standards projects ended in an ISO technical report in 1999.**
- **The Semantic Web became the next great hope.**

DARPA Agent Markup Language

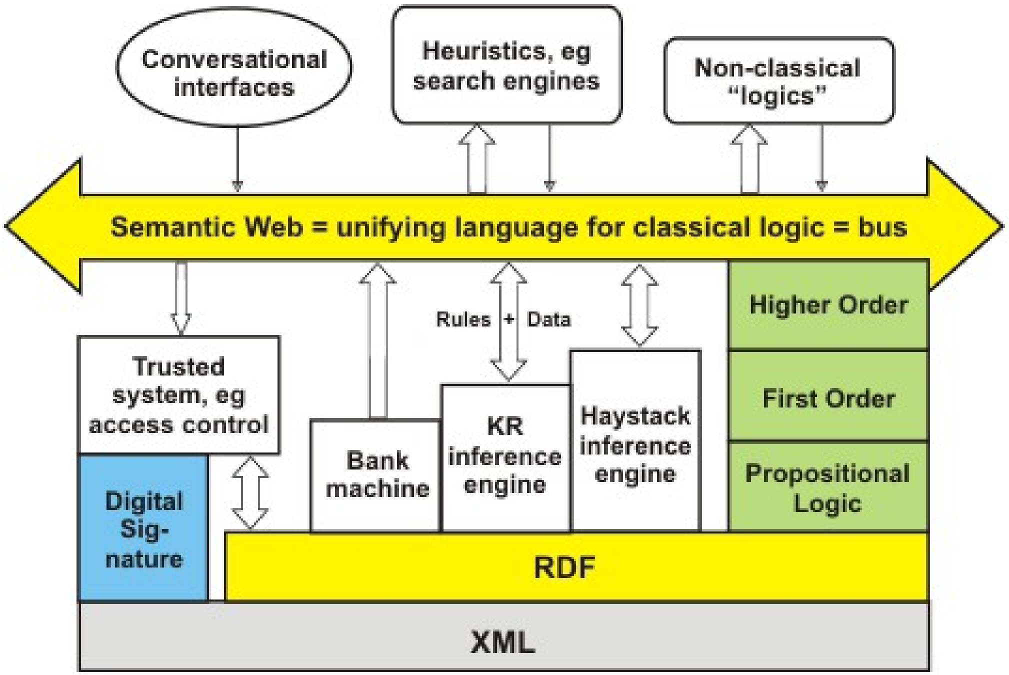


The diagram summarizes the requirements for the DAML project.

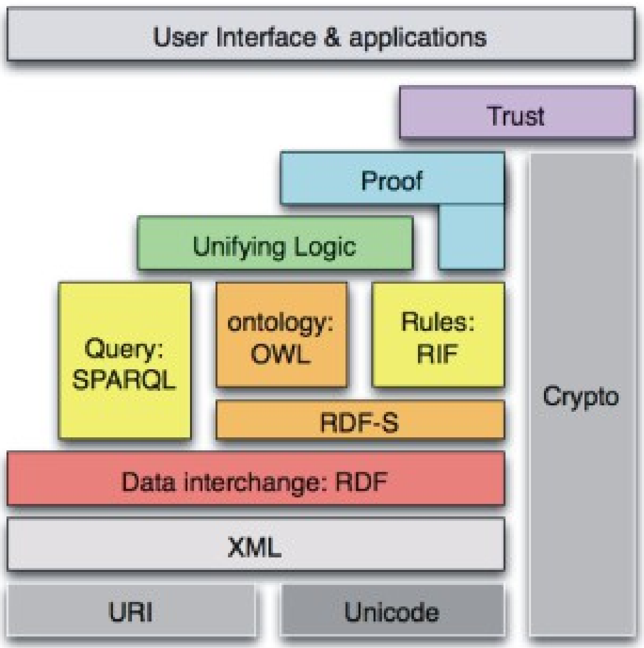
- **From a presentation by Jim Hendler, the DARPA project manager. ***
- **The PI of the winning proposal was Tim Berners-Lee.**

*** See http://www.jfsowa.com/ikl/Hendler00.pdf**

Layer Cakes for the Semantic Web



As proposed (2000)



As delivered (2005)

The original diagram embodied many good ideas.

But building semantics on top of syntax was not one of them.

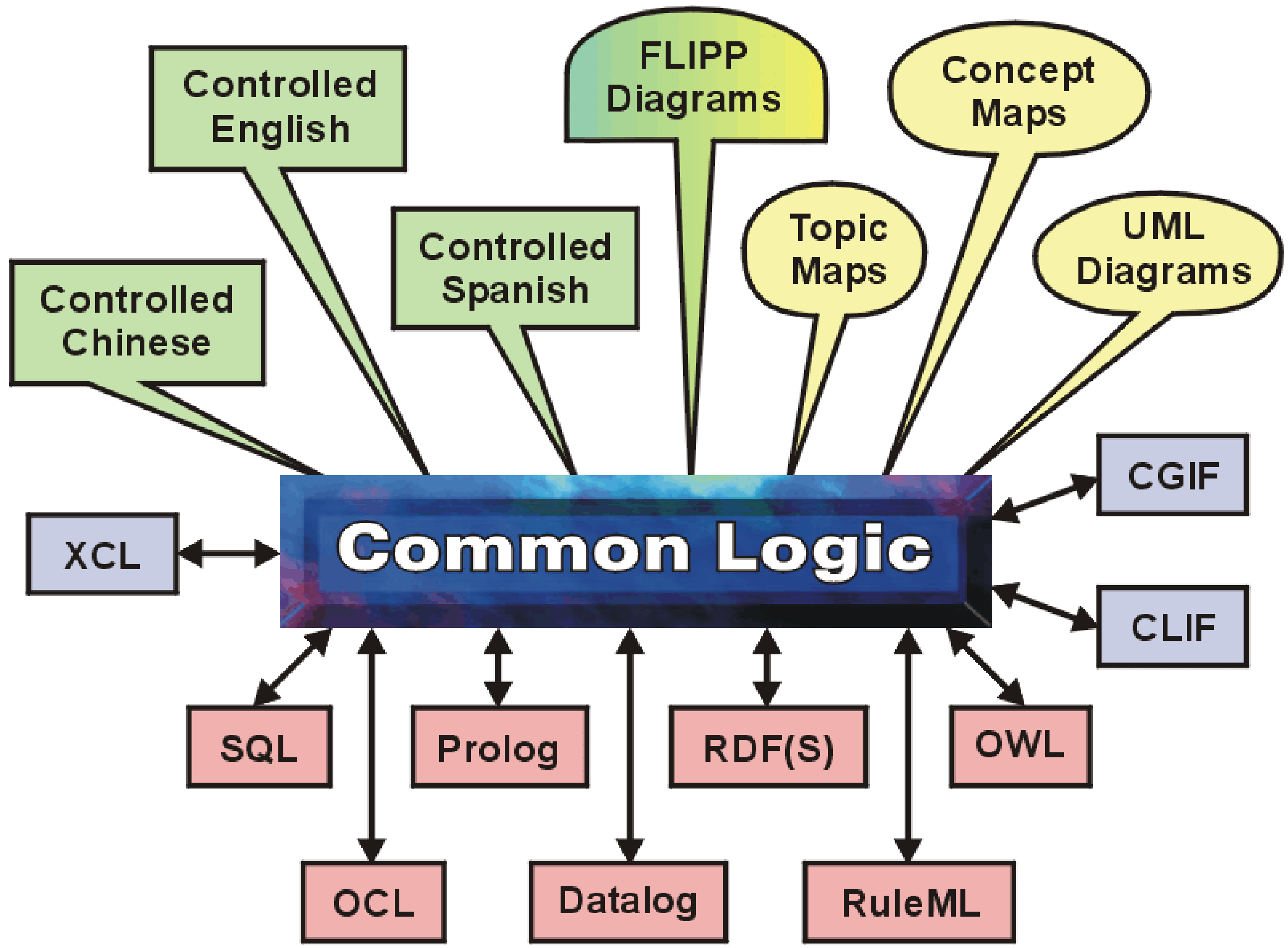
Result: Miscellaneous notations with unifying logic as a future hope.

Future Challenges and Possibilities

Knowledge graphs are readable and flexible.

- **They explore new technology developed in the past 15 years.**
- **Important goals: A humanly readable notation for anything derived from the WWW by new technology, such as DNNs.**
- **Flexability is essential, decidability is meaningless.**
- **Some versions can be mapped to and from RDF.**
- **Others are closer to natural languages.**
- **And some are used to represent a kind of conceptual schema.**
- **It's premature to specify a standard, since the range of possible technology and applications is still in flux.**

Human Interfaces

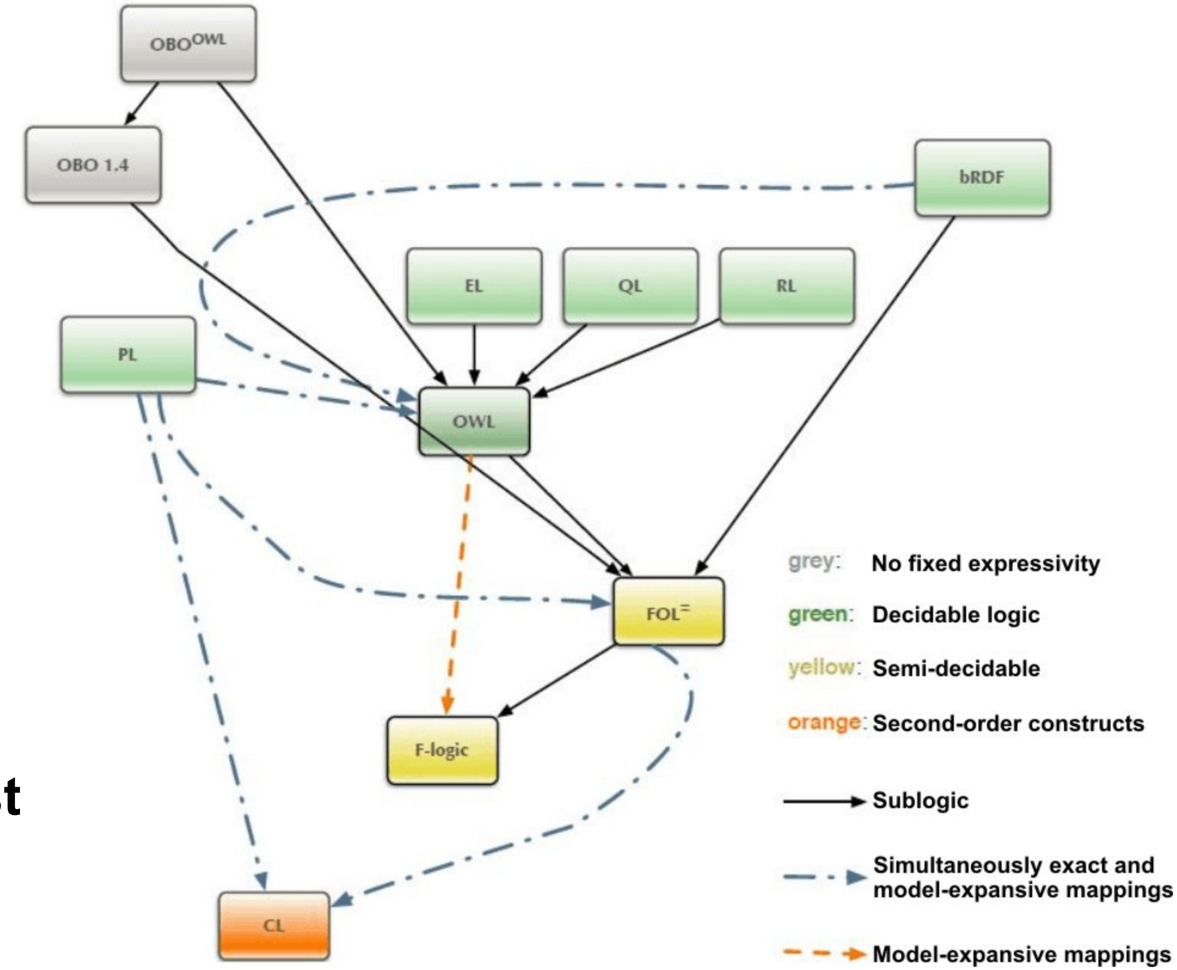


Machine Interfaces

Distributed OMS Language (DOL)

Mapping ontologies: *

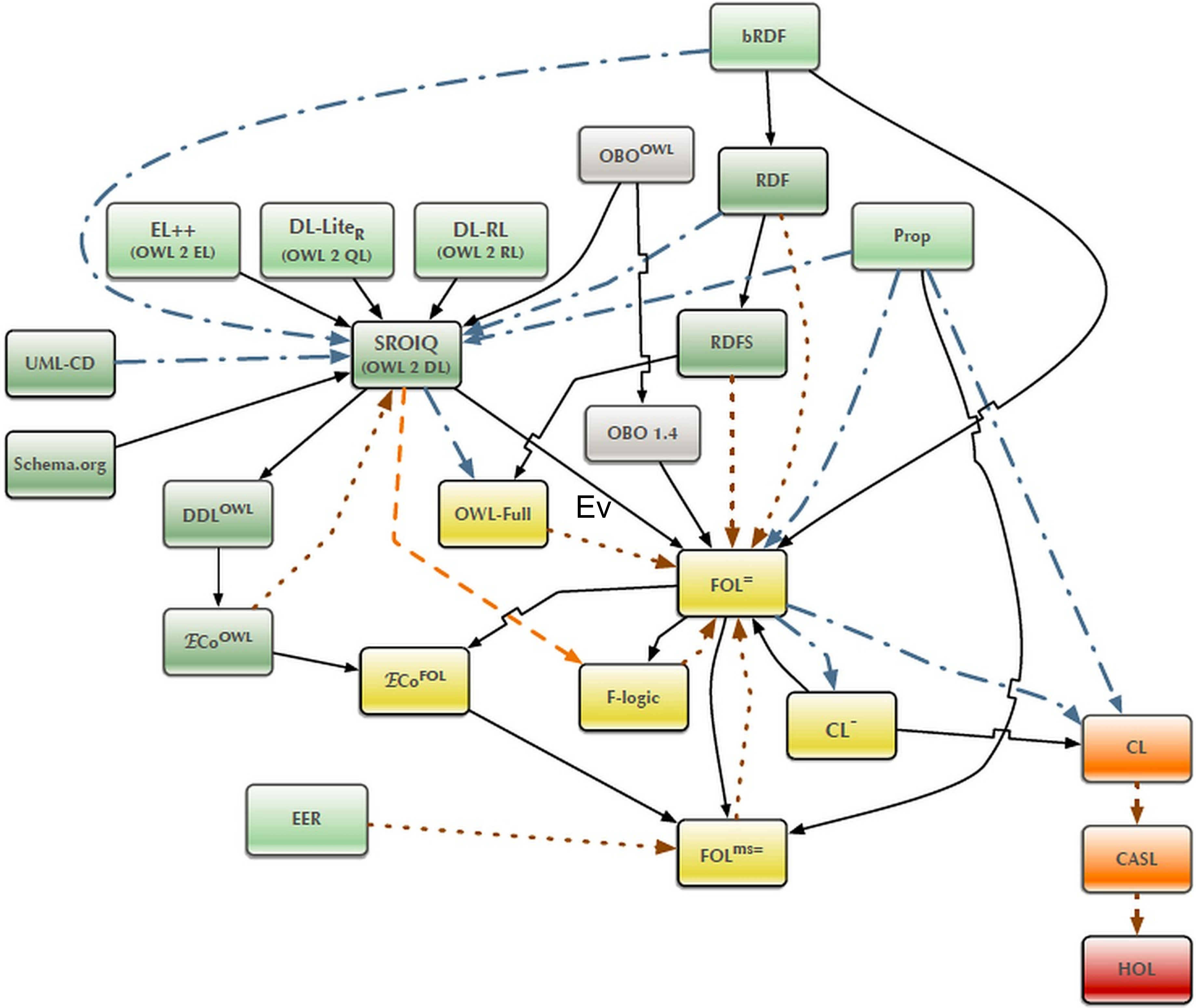
- OMS: Ontology, Model, and Specification.
- Goal: Map an OMS expressed in one logic to equivalent versions in other logics.
- The diagram at right shows possible mappings.
- The target logic must have the same or greater expressivity.
- Common Logic (CL) is the most expressive logic shown.



Two related strategies:

- Dialects: Adopt a highly expressive logic such as CL as the base, and define all other logics as dialects of the base logic.
- Mappings: Use DOL to specify mappings (morphisms) among logics, but no logic is treated as a dialect of any other.

* Distributed Ontology, Model, and Specification Language (DOL), http://www.omg.org/spec/DOL/



bRDF
OBO^OWL
RDF
EL++
(OWL 2 EL)
DL-Lite_R
(OWL 2 QL)
DL-RL
(OWL 2 RL)
Prop
SROIQ
(OWL 2 DL)
RDFS
UML-CD
Schema.org
OBO 1.4
DDL^OWL
OWL-Full
Ev
FOL^=
ℰCo^OWL
ℰCo^FOL
F-logic
CL^-
CL
EER
FOL^ms=
CASL
HOL

Formal Concept Analysis (FCA)

A theory with supporting algorithms and methodology:

- Theory. Define a minimal lattice that shows all inheritance paths among a set of concepts, each defined by a list of attributes.
- Algorithms. Efficient methods for computing a minimal lattice from a set of concepts and attributes.
- Methodology. Techniques for describing concepts by attributes and generating lattices for ontologies and inheritance.

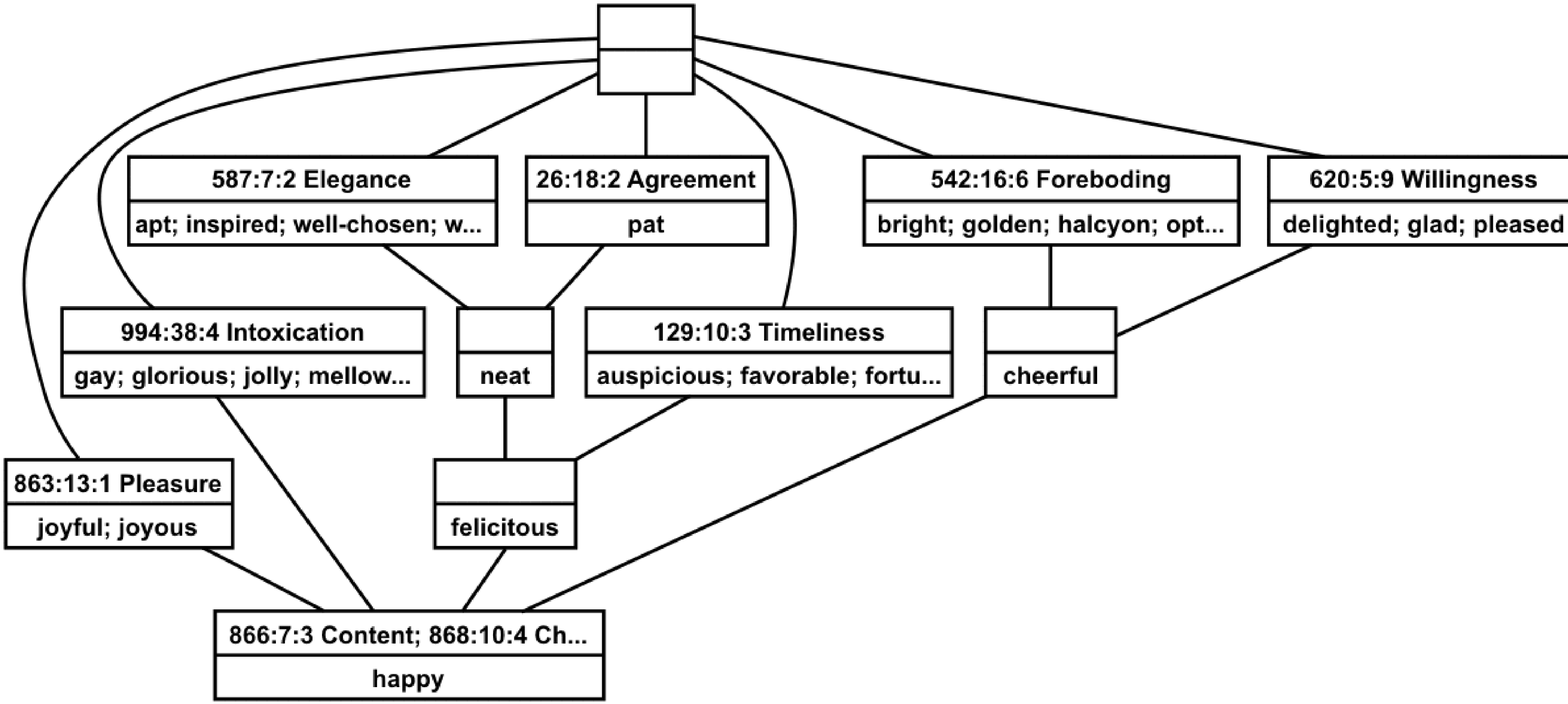
Applications:

- Ontology development and alignment; classification methods; machine learning; defining concepts used in other logics.
- FCA tools are commonly used to show that ontologies specified in OWL and other notations are consistent.

The FCA Homepage: http://www.upriss.org.uk/fca/fca.html

For deriving lattices from lexical resources: http://www.upriss.org.uk/papers/jucs04.pdf

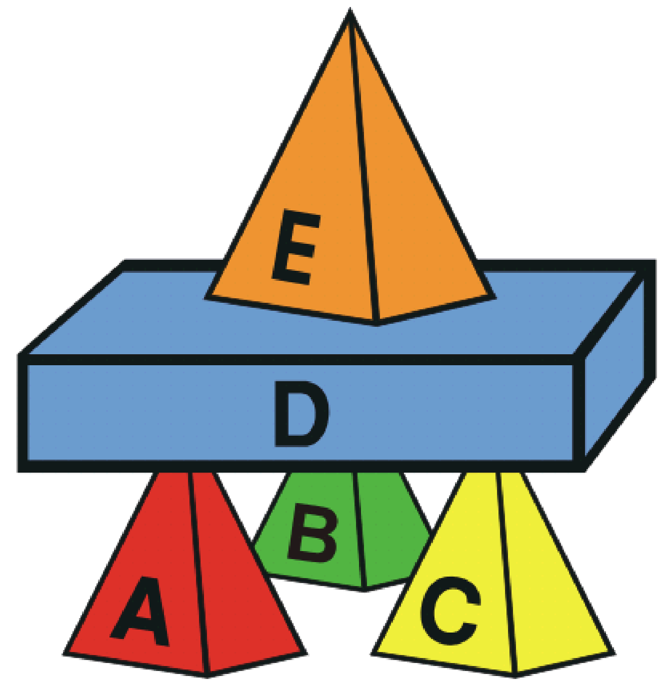
Generating Lattices Automatically



FCA algorithms used the data in Roget's Thesaurus to generate this lattice for the word 'happy' and its hypernyms (supertypes).

To generate this or similar lattices, enter 'happy' or any other word at the web site http://www.ketlab.org.uk/roget.html

Describing Things in Different Ways



How can we describe what we see?

In ordinary language?

In some version of logic?

In a relational database?

In the Semantic Web?

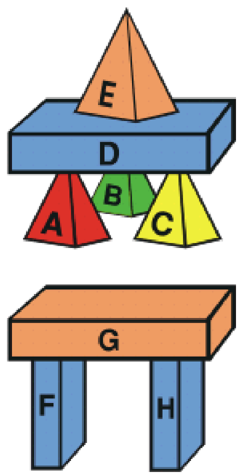
In a programming language?

Even when people use the same language,
they may use different words and expressions.

How could humans or computers relate different descriptions to one another?

Structured and Unstructured Representations

A description in tables of a relational database:



Objects

Entity	Shape	Color
A	pyramid	red
B	pyramid	green
C	pyramid	yellow
D	block	blue
E	pyramid	orange
F	block	blue
G	block	orange
H	block	blue

Supports

Supporter	Supportee
A	D
B	D
C	D
D	E
F	G
H	G

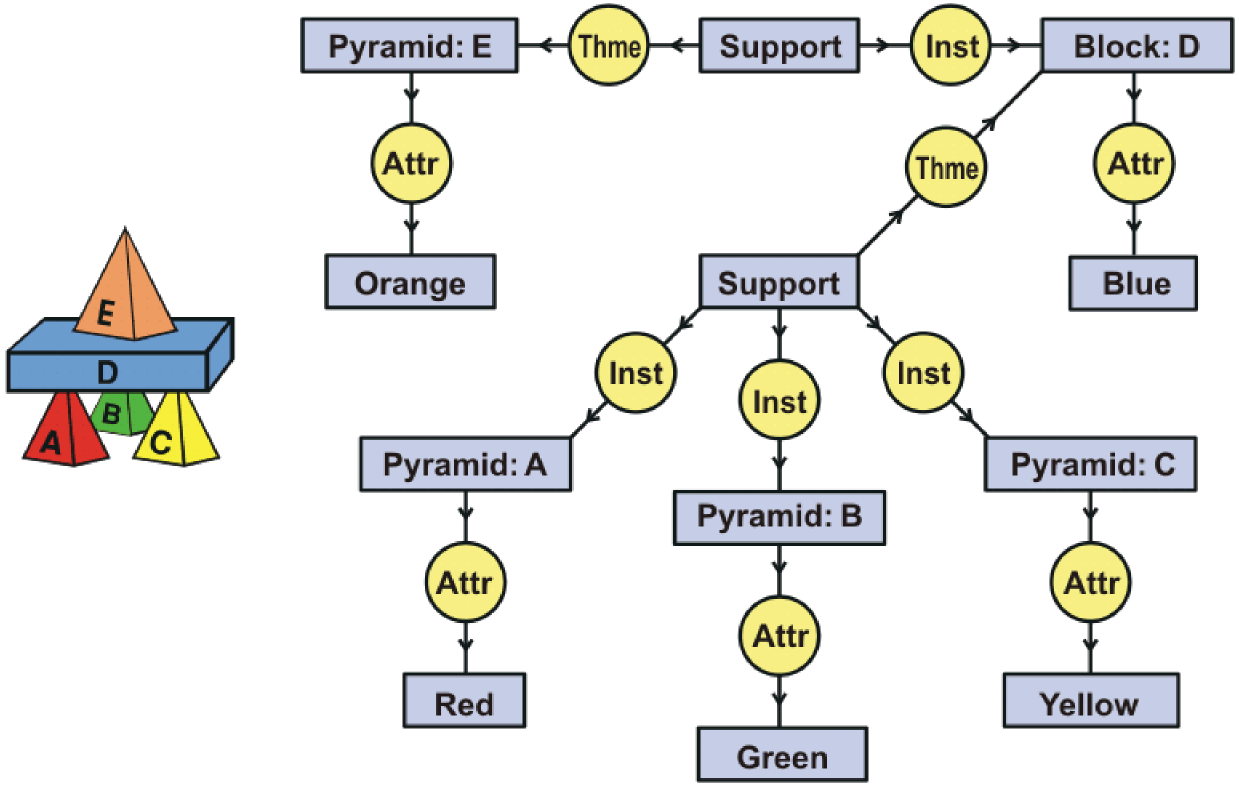
A description in English:

> “A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E.”

The database is called structured, and English is called unstructured.

Yet English has even more structure, but of a very different kind.

Mapping English to a Conceptual Graph



"A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E."

The concepts (blue) are derived from English words, and the conceptual relations (yellow) from the case relations or thematic roles of linguistics.

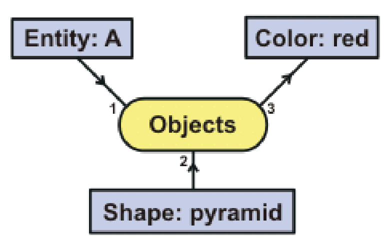
Mapping Database Relations to Conceptual Relations

Objects

Entity	Shape	Color
A	pyramid	red
B	pyramid	green
C	pyramid	yellow
D	block	blue
E	pyramid	orange
F	block	blue
G	block	orange
H	block	blue

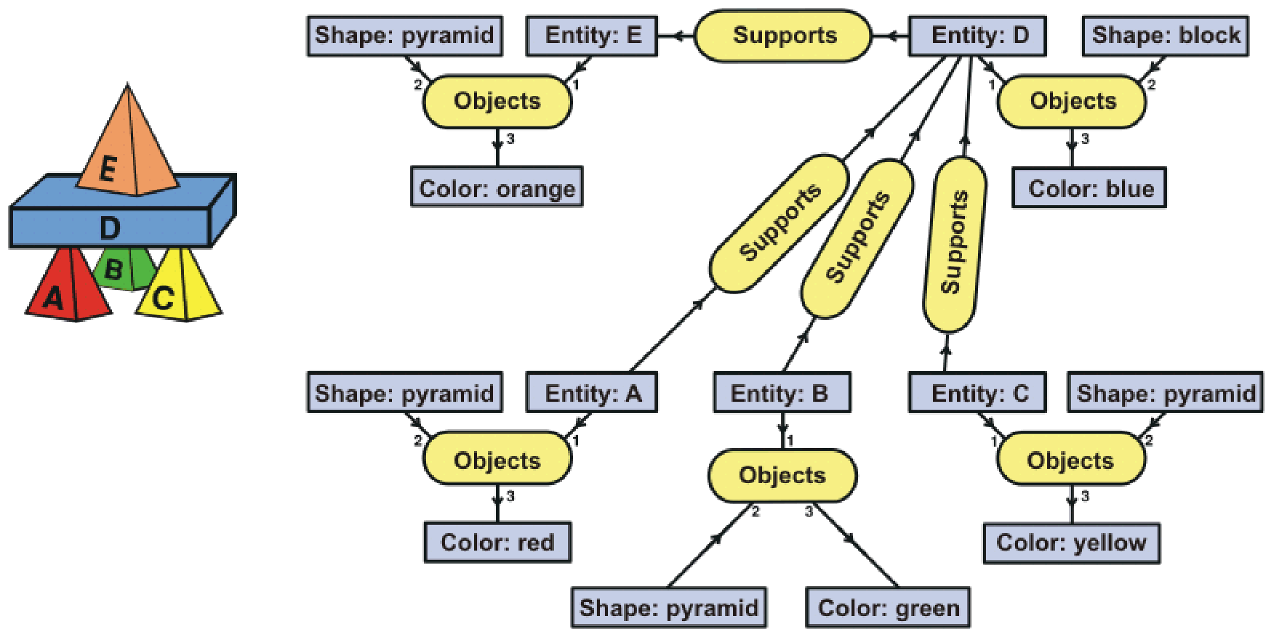
Supports

Supporter	Supportee
A	D
B	D
C	D
D	E
F	G
H	G



Each row of each table maps to one conceptual relation, which is linked to as many concepts as there are columns in the table.

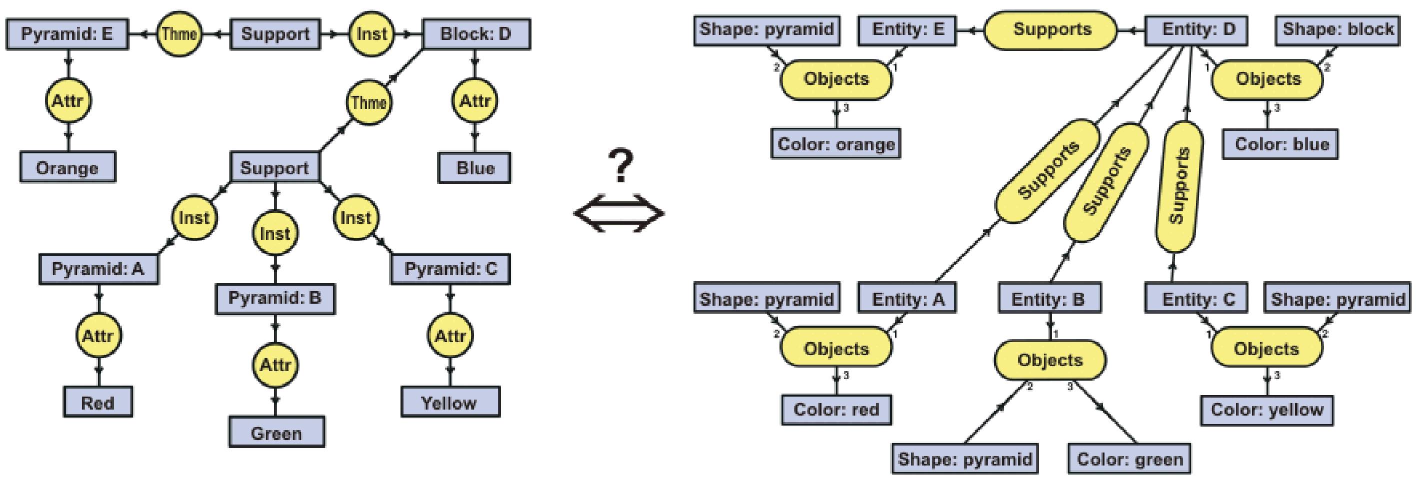
Mapping an Entire Database to Conceptual Graphs



Join concept nodes that refer to the same entities.

Closely related entities are described by connected graphs.

Mapping the Two Graphs to One Another



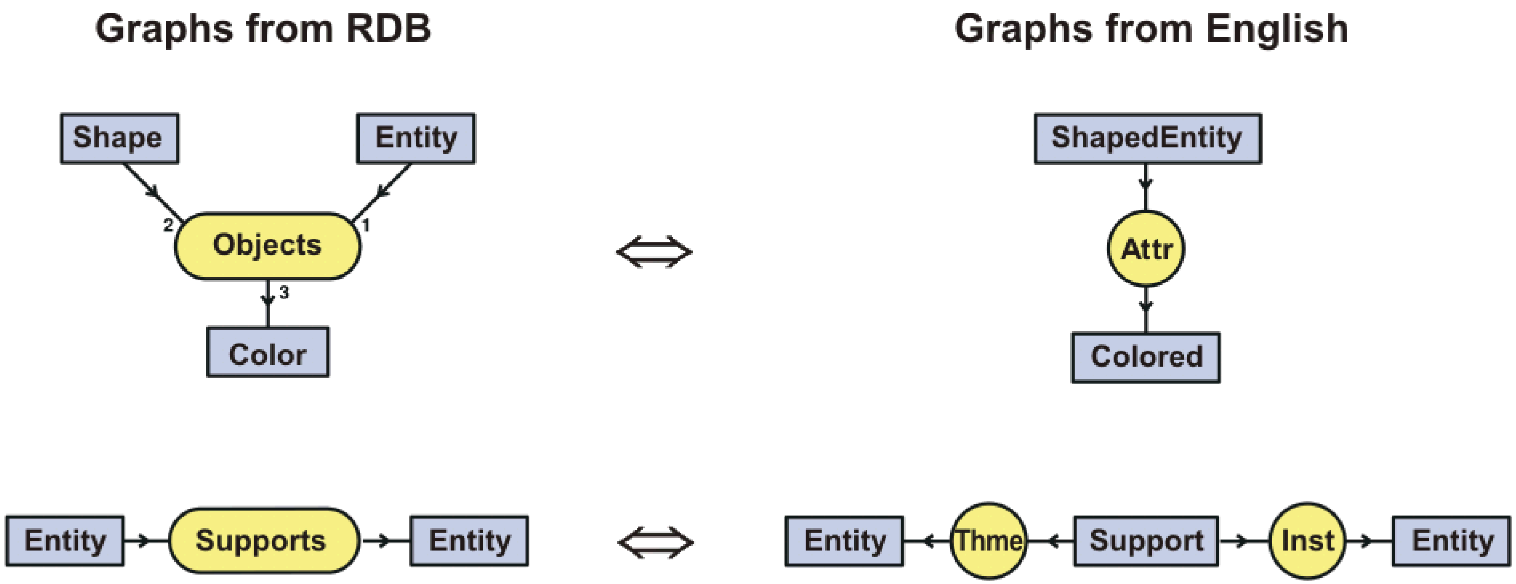
Very different ontologies: 12 concept nodes vs. 15 concept nodes, 11 relation nodes vs. 9 relation nodes, no similarity in type labels.

The only commonality is in the five names: A, B, C, D, E.

People can recognize the underlying similarities.

How is it possible for a computer to discover them?

Aligning Ontologies by Mapping Graphs



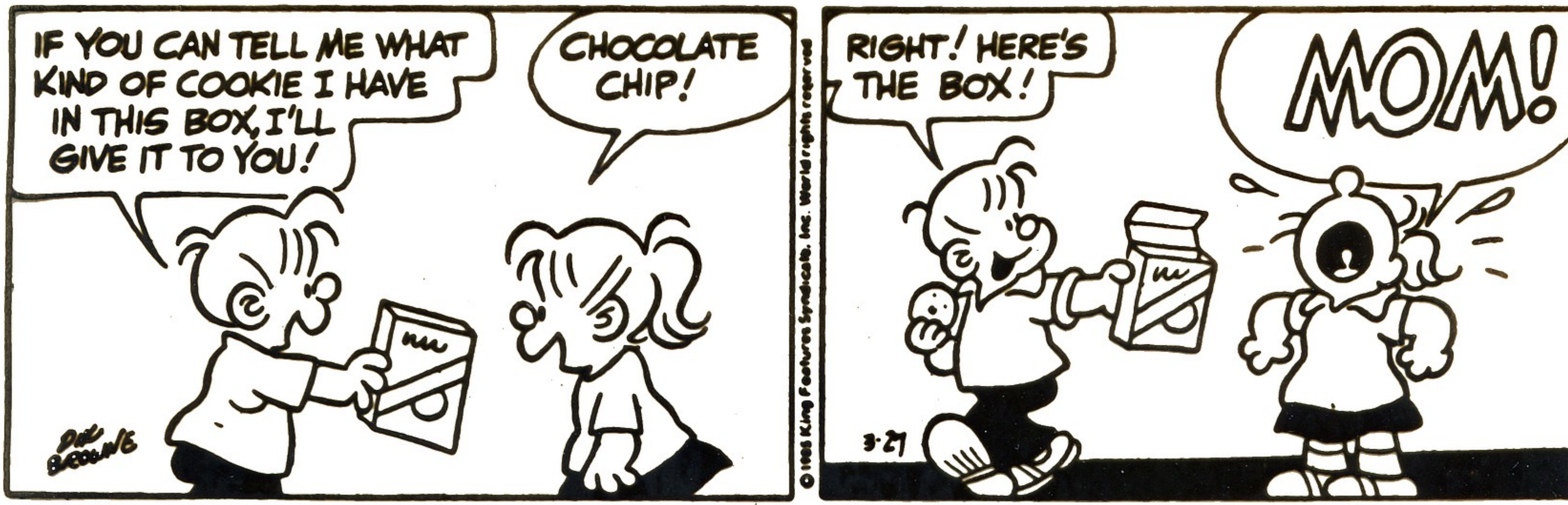
Repeated application of these two transformations completely map all nodes and arcs of each graph to the other.

This mapping, done by hand, is from an example by Sowa (2000), Ch 7.

The VivoMind Analogy Engine (VAE) found the mapping automatically.

Context in Language

Hi & Lois

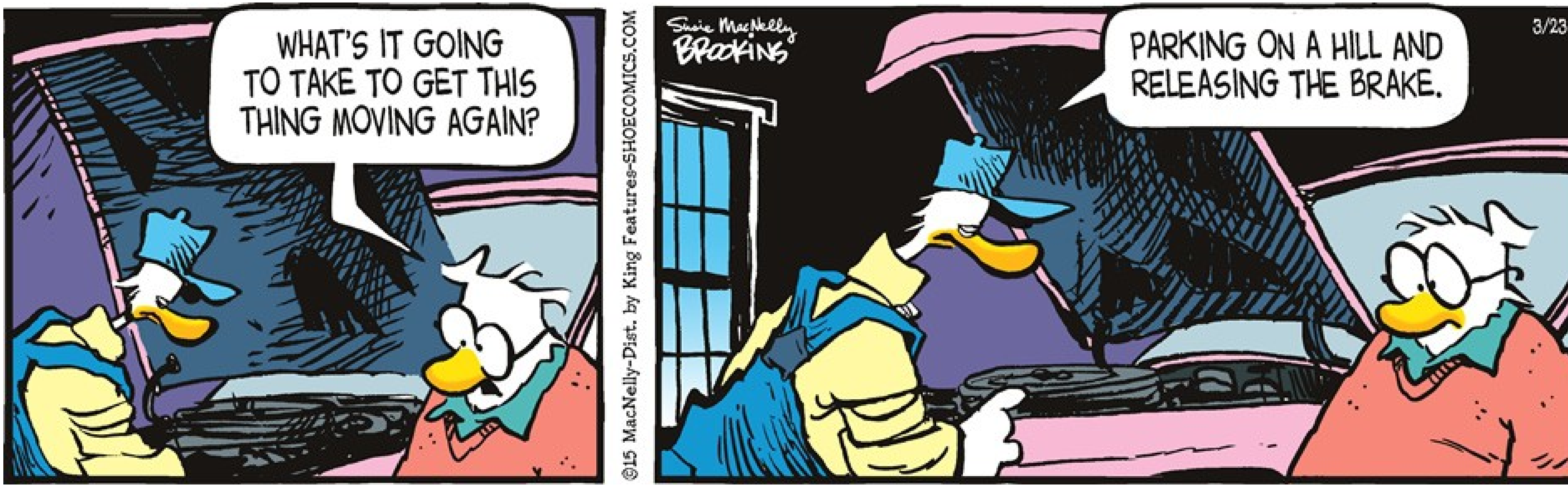


Four kinds of context: The text or discourse; the situation; common background knowledge; and the intentions of the participants.

Linguistics: Parse the sentences, resolve the referents of noun phrases, and determine the literal meaning of the text.

Pragmatics: Determine the implications by relating the meaning to the situation, the background knowledge, and the intentions.

Using the Context in NLP



Syntax is easy: Parse the question and the answer.

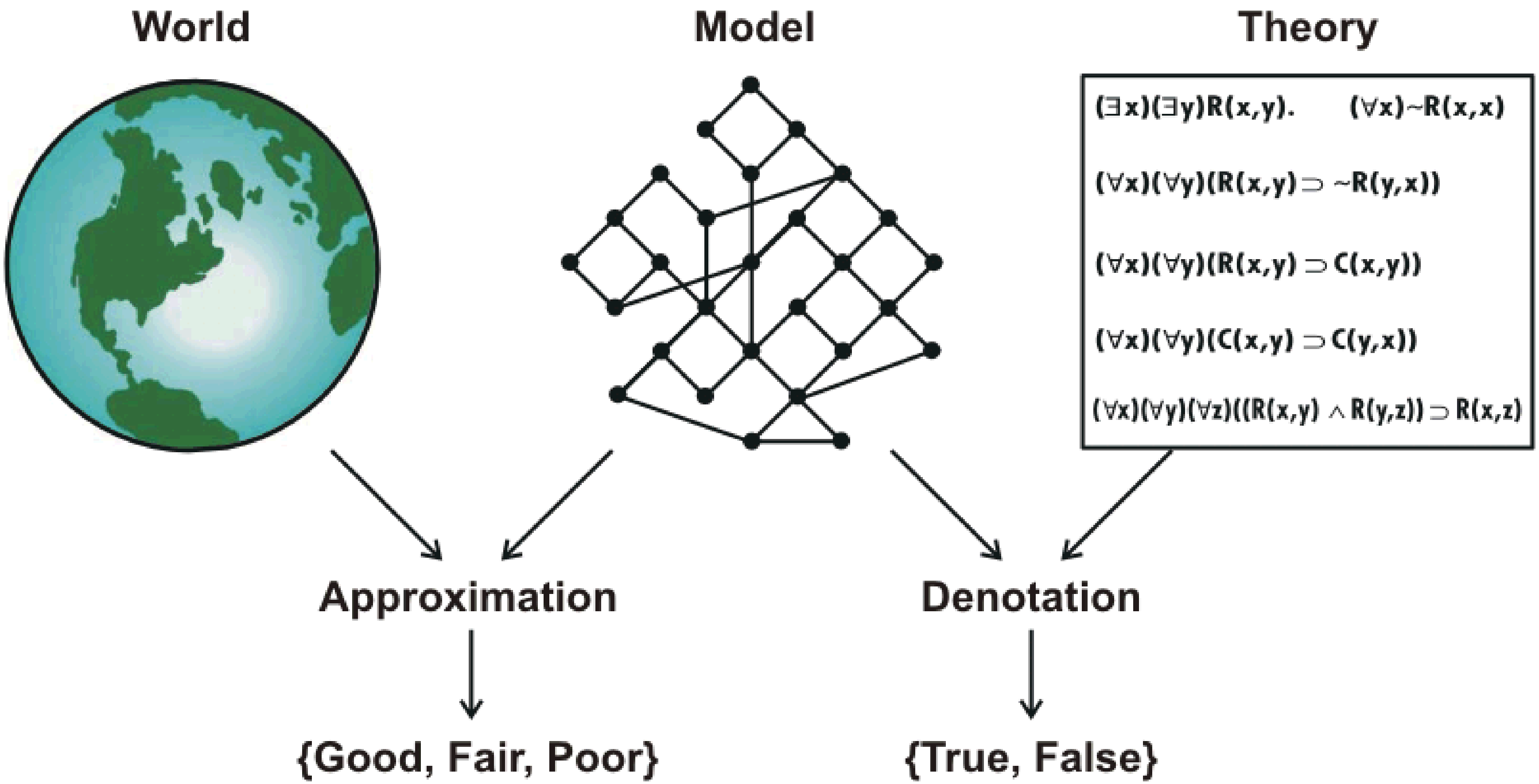
Semantics is harder: Use the context to

- **Recognize the situation type and the roles of the two participants,**
- **Relate the word 'thing' to the car that is in a garage,**
- **Relate the verbs 'take' and 'move' to the situation,**
- **Apply the laws of physics to understand the answer.**

Pragmatics is the hardest: Determine the intentions of the participants and their implications for the irony and humor.

*** Source of cartoon: search for 'moving' at http://www.shoecomics.com/**

Models of Worlds, Real or Possible

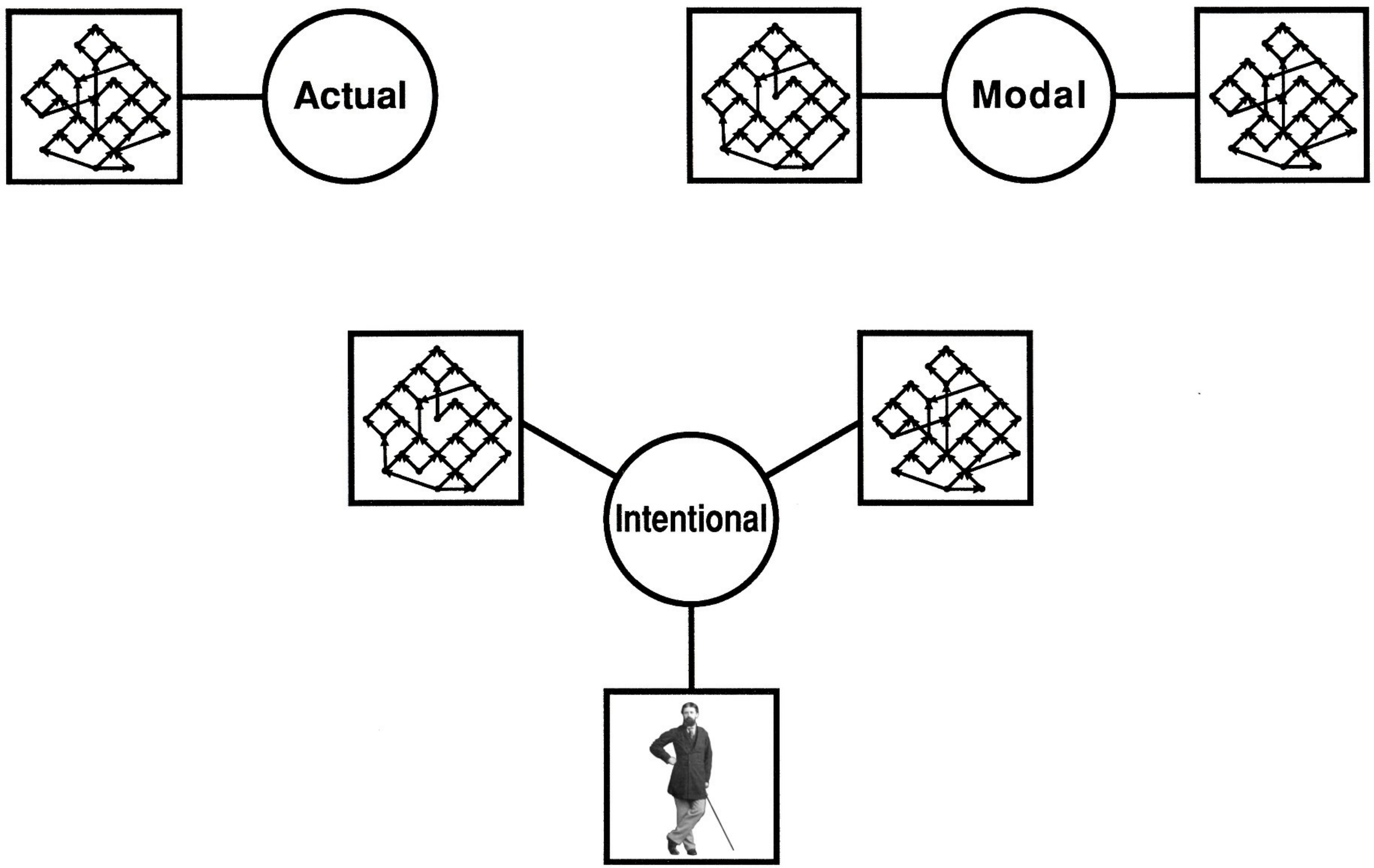


A Tarski-style model evaluates axioms of a theory in terms of a world, which may be described by a set, a network, or a database of facts.

For modal logic, the model may consist of a family of possible worlds.

In computer applications, possible worlds are represented by sets of propositions that are true (facts) or necessarily true (laws).

Actual, Modal, and Intentional Contexts



Three kinds of contexts, according to the source of knowledge:

- **Actual: Something factual about the world.**
- **Modal: Something possible, as determined by some hypothesis.**
- **Intentional: Something an agent believes, desires, or intends.**

Nested Situations

The three situations may be described as actual, modal, or intentional.

1. Actual: ***Pierre is thinking of Marie, who is thinking of him.***

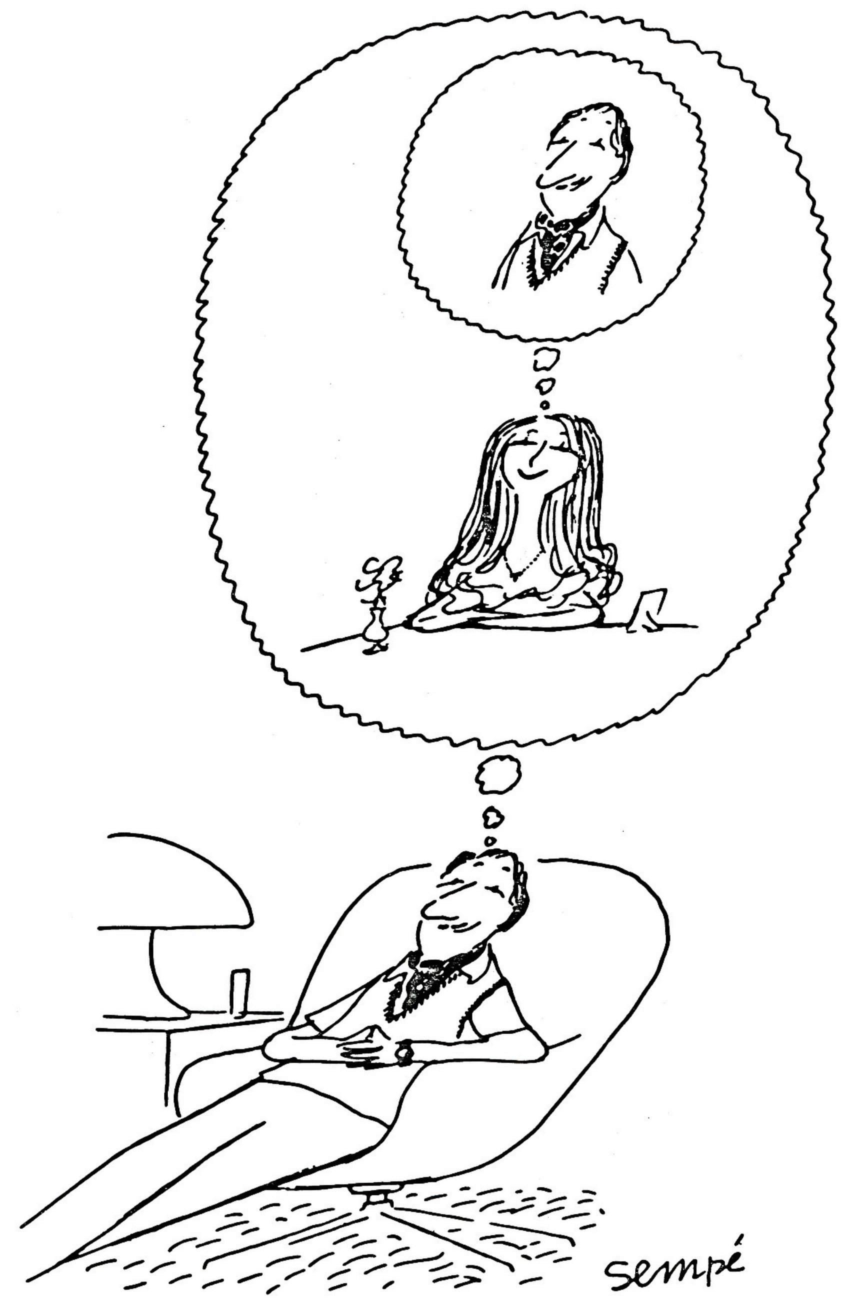
2. Modal: ***Pierre is thinking of Marie, who may be thinking of him.***

3. Intentional: ***Pierre hopes that Marie is thinking of him.***

In #1, both clauses are true, but Pierre may not know what Marie thinks.

In #2, the first clause is true, but the second may be true or false.

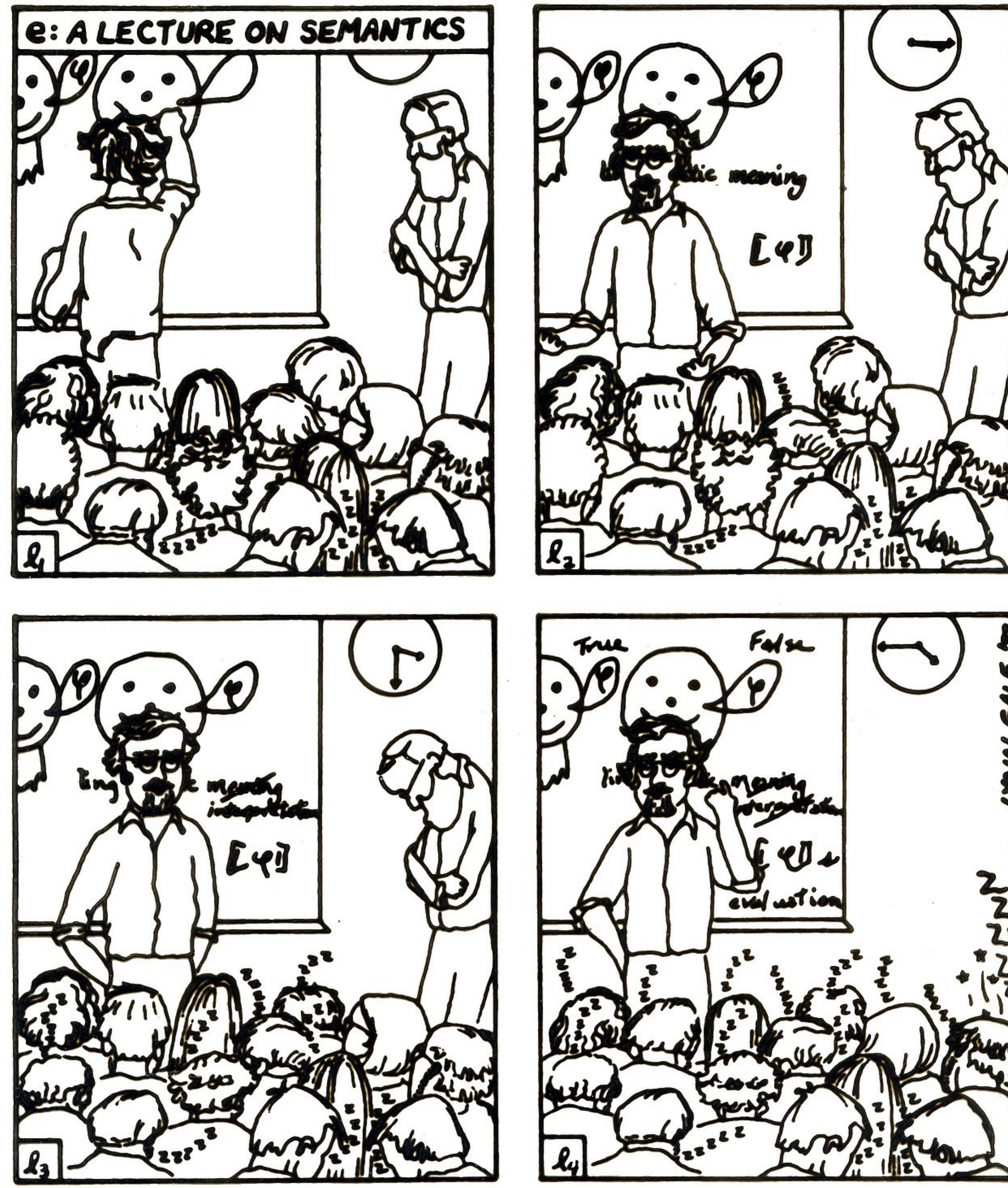
In #3, Pierre assumes or wishes that his thought is true, but it may be false.



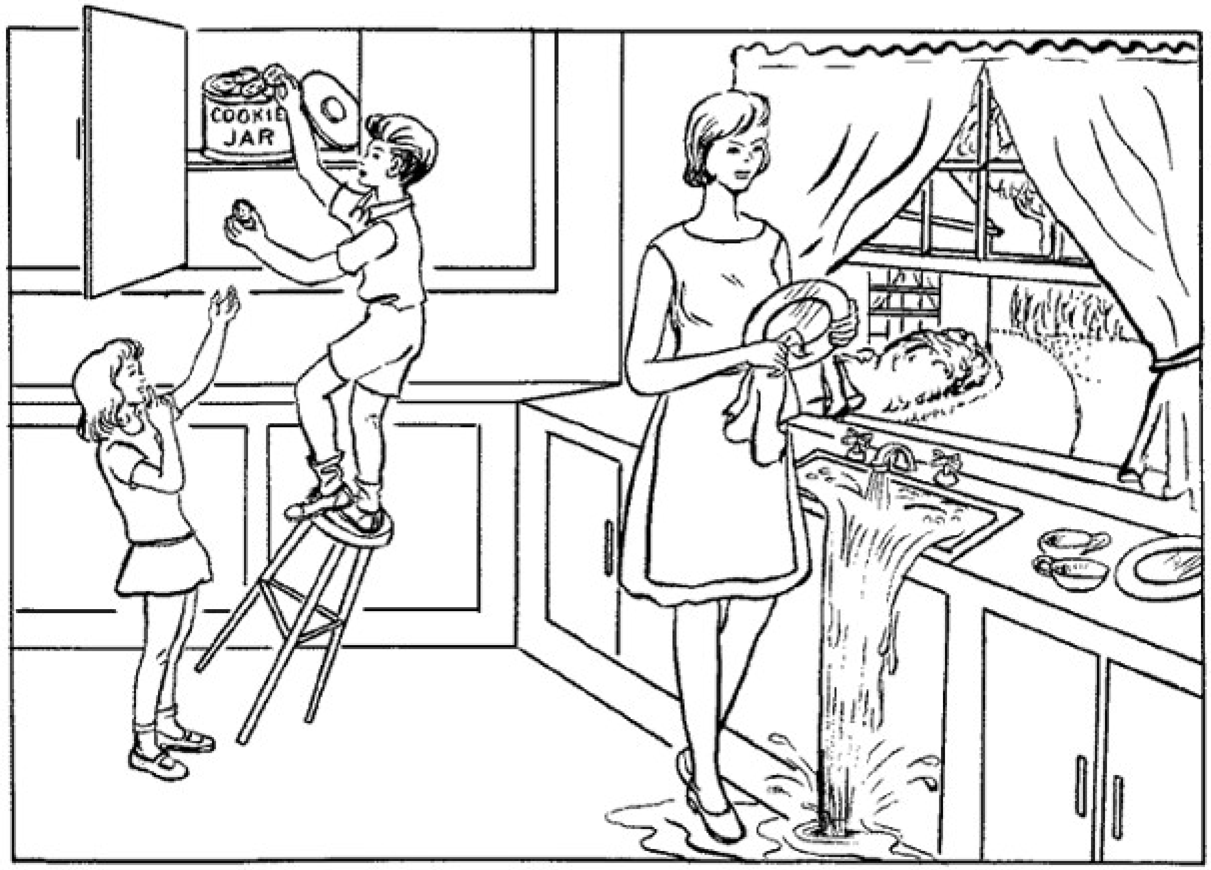
In the situation *e*, John Perry is lecturing while Jon Barwise is standing on the right.

A language expression φ is a relation between a *discourse situation d*, a *speaker connection function c*, and a *described situation e*:
$d, c \,\|\varphi\|\, e$.

If φ is the expression *“the number of sleeping students”*, its value is 3 at 3:01 pm, 5 at 3:15, 9 at 3:30, and 19 at 3:45.



Example of a Situation

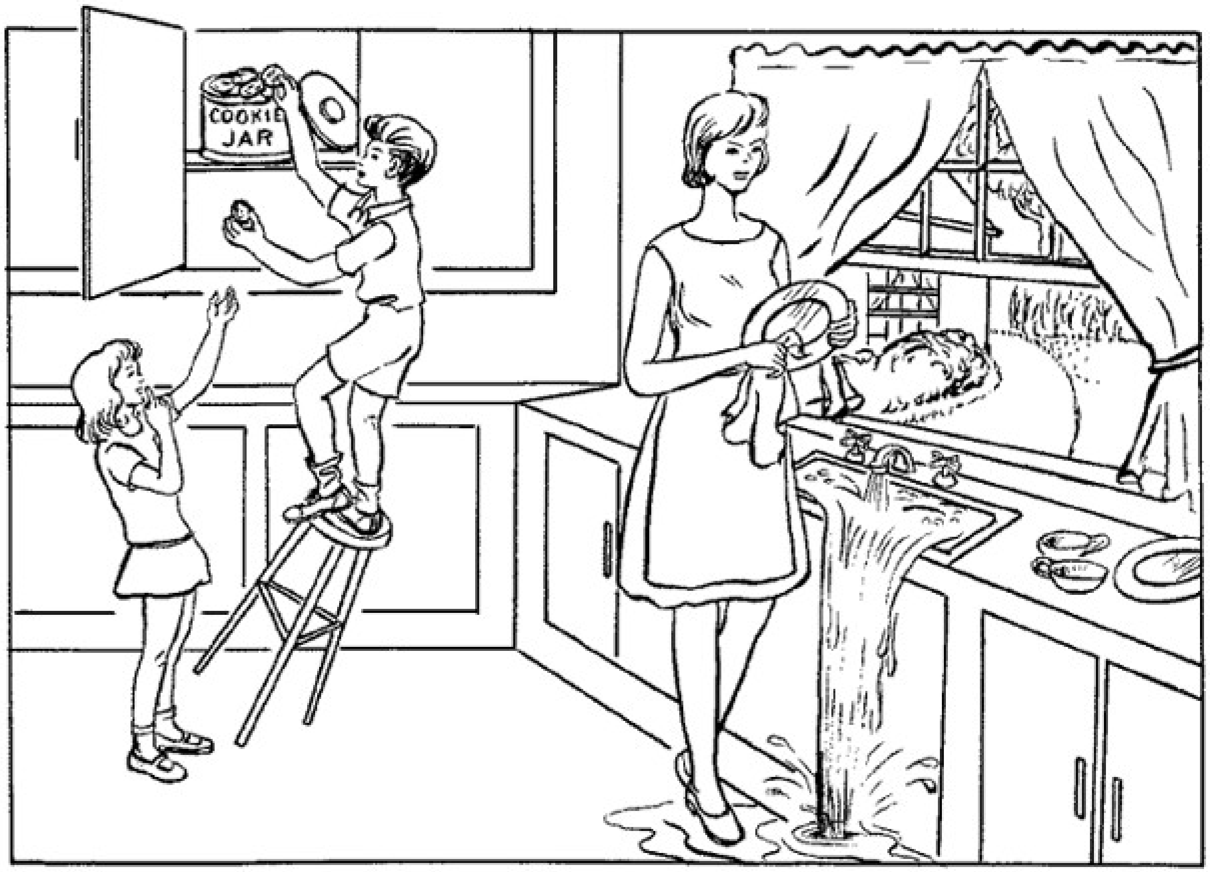


This is a test picture used to diagnose patients with aphasia.

A patient's description of the situation can show the effects of lesions caused by wound, stroke, tumor, or infection.

The "cookie theft" picture was adapted from Goodglass & Kaplan (1972).

A Description in Controlled English



{**Situation:** ***A woman, a girl, and a boy are in a kitchen of a house. The woman wipes a plate with a cloth. Water spills on the floor of the kitchen. The girl reaches for a cookie. The boy holds a cookie in his left hand. The boy grasps a cookie with his right hand. The boy stands on a stool. The stool tips over. The boy falls down.***}