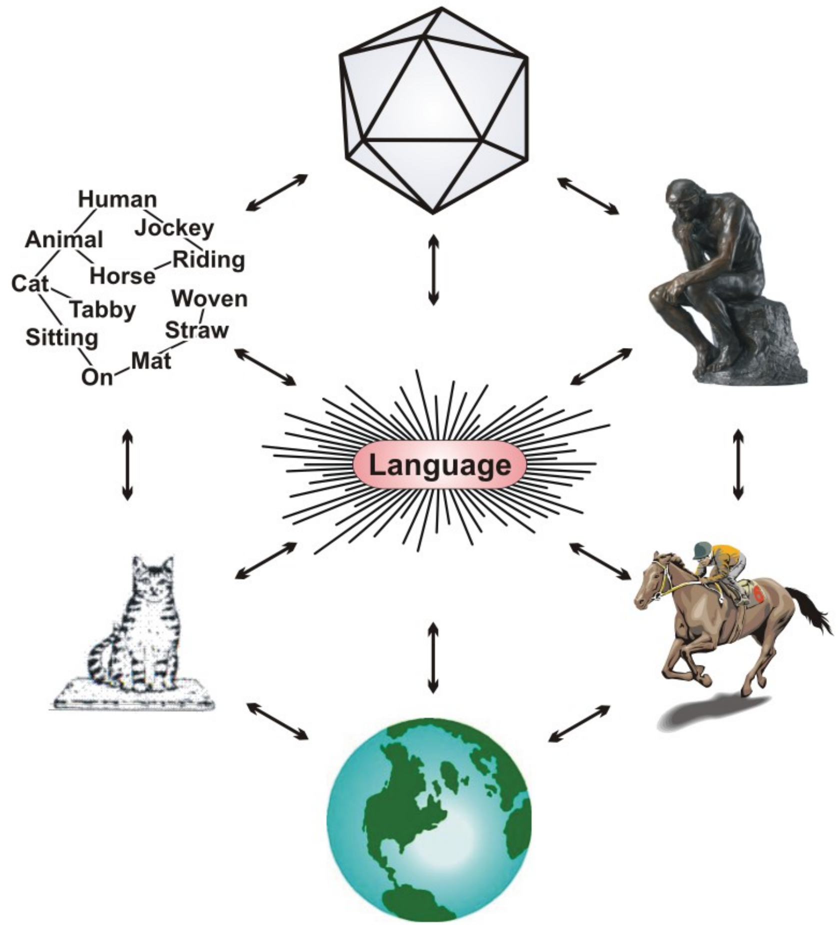
Knowledge Graphs and Logic



John F. Sowa

13 May 2020

1. What is a Knowledge Graph?

Knowledge graphs are simplified semantic networks.

- **Google introduced KGs in 2012 for question answering.**
- **DBpedia and other freely available resources provide the data.**
- **The KGs are stored as triples, and Google added various AI methods for learning and reasoning.**

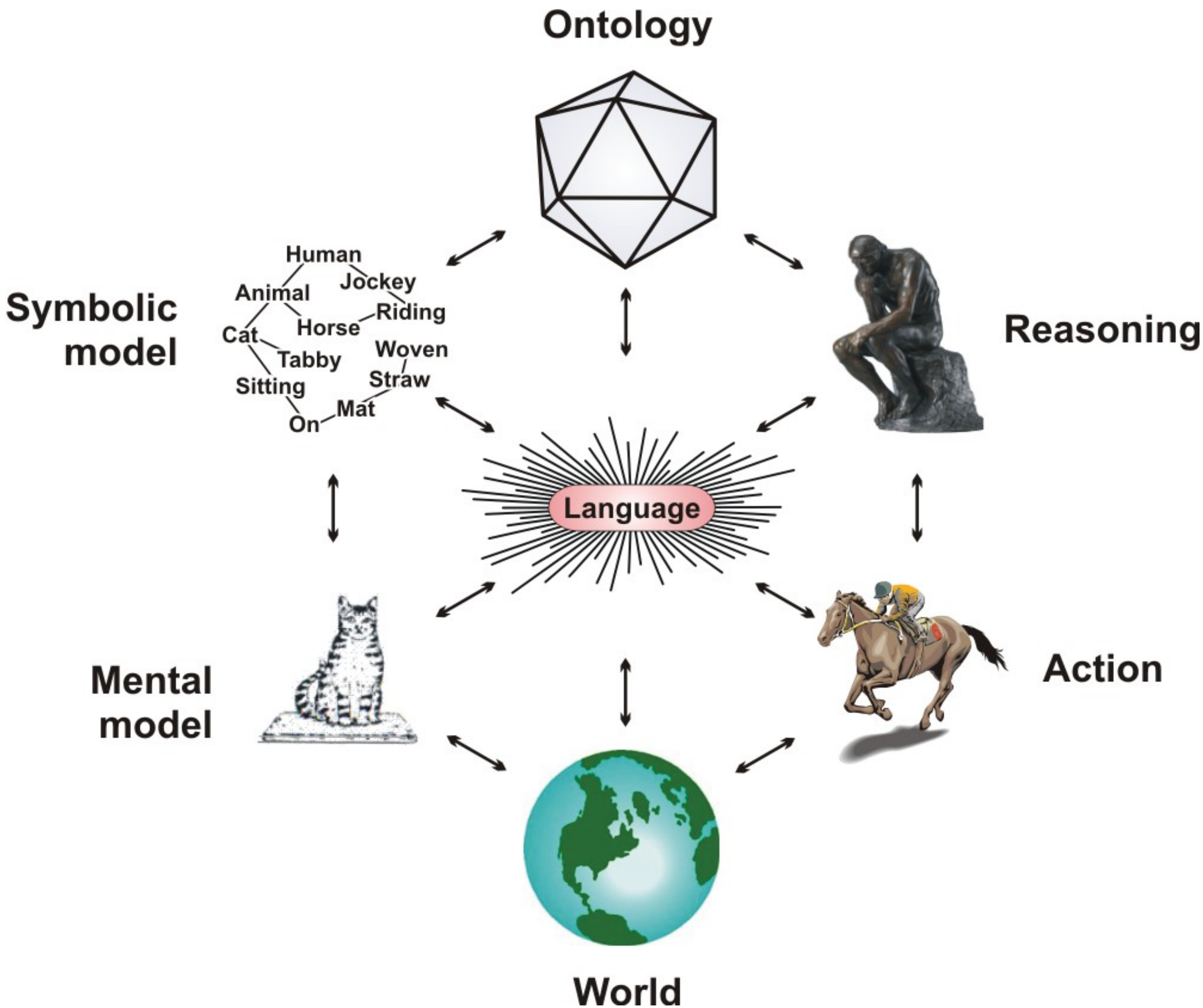
For the Jeopardy challenge, IBM Watson also used DBpedia.*

Watson added a wide range of AI technology: English parsers, question classification, question decomposition, automatic source acquisition and evaluation, entity and relation detection, logical form generation, statistics, machine learning, knowledge representation, and several methods of reasoning.

These are two of the many kinds of applications for KGs.

- **What kinds of standards could support all of them?**

*** See https://www.aaai.org/Magazine/Watson/watson.php**



Standards for knowledge graphs must support direct and indirect mappings to and from the many kinds of software for processing natural languages at every corner of the hexagon. How?

Implementing the Hexagon

The corners of the hexagon represent aspects of knowledge.

1. The world is everything we encounter in space and time.
2. Mental models represent everything we experience or imagine.
3. Symbolic models consist of words related by words to other words.
4. Ontology is a catalog of all the words and whatever they refer to.
5. Reasoning includes all the ways of thinking about anything.
6. Action is what thinking leads us to do in and on the world.

Language can represent knowledge about all those aspects. *

- No computer system can understand language at a human level.
- But many systems can process more data faster than humans.
- AI tools can process some aspects of what humans do
- But standards that are too restrictive may stifle progress.

* For more detailed slides, see http://jfsowa.com/talks/kg2ll.pdf

2. DOL for Knowledge Graphs

DOL is a standard for integration and interoperation among distributed ontologies, models, and specifications (OMS). *

- Knowledge graphs (KGs) are as diverse as the many kinds of OMS, and they can benefit from the OMS tools.
- UML and the Semantic Web logics are supported by DOL.
- Tools based on DOL can map KGs to and from those logics.

DOL is formally defined by logic and mathematics.

- Logic is essential for guaranteeing precision.
- DOL uses Common Logic (CL) to define, relate, and integrate heterogeneous OMS.
- But people can still use their familiar diagrams and notations.

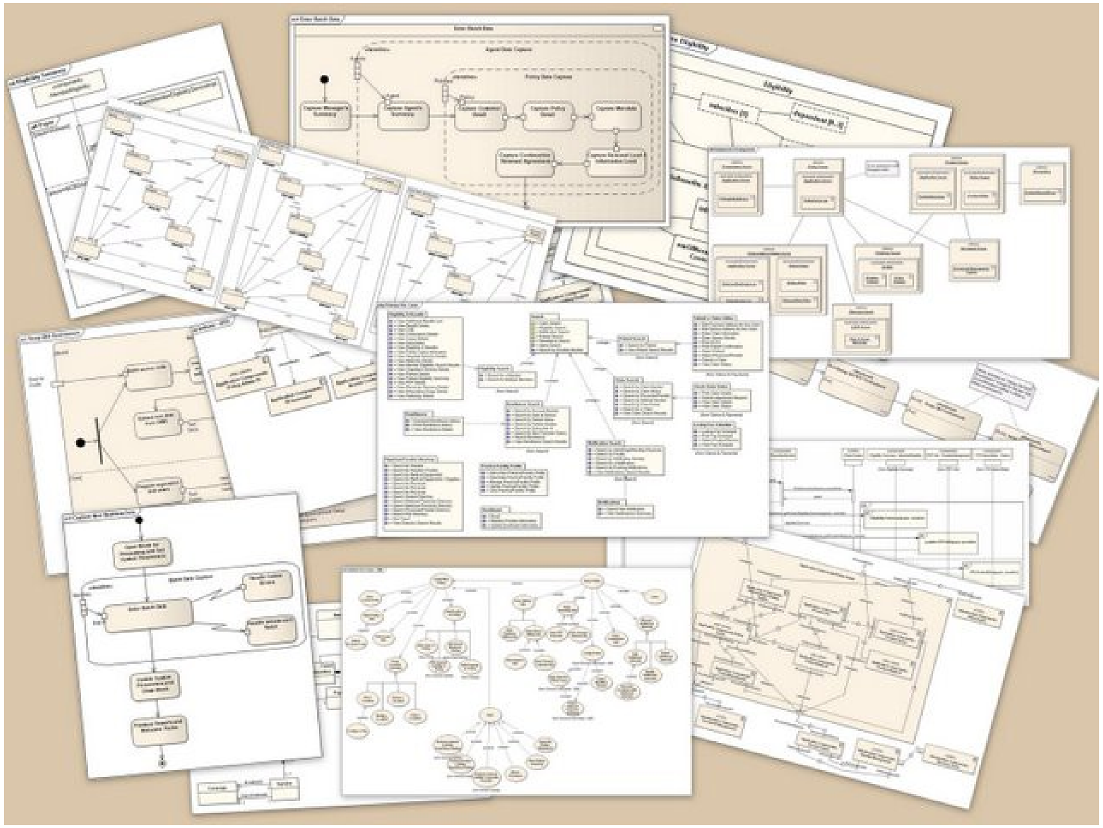
* OMG Standard for DOL: Distributed Ontology, Modeling, and Specification Language: https://www.omg.org/spec/DOL/1.0

Unified Modeling Language (UML)

A family of diagrams for representing database and computer system designs.

Originally specified as informal notations without a precise definition in logic.

The Object Management Group (OMG) standardized formal UML by definitions stated in Common Logic.*



By mapping UML diagrams and SW logics to CL, DOL can facilitate data sharing among applications of any kind – including the trillions of dollars of legacy systems.

* See https://www.omg.org/spec/FUML/1.4

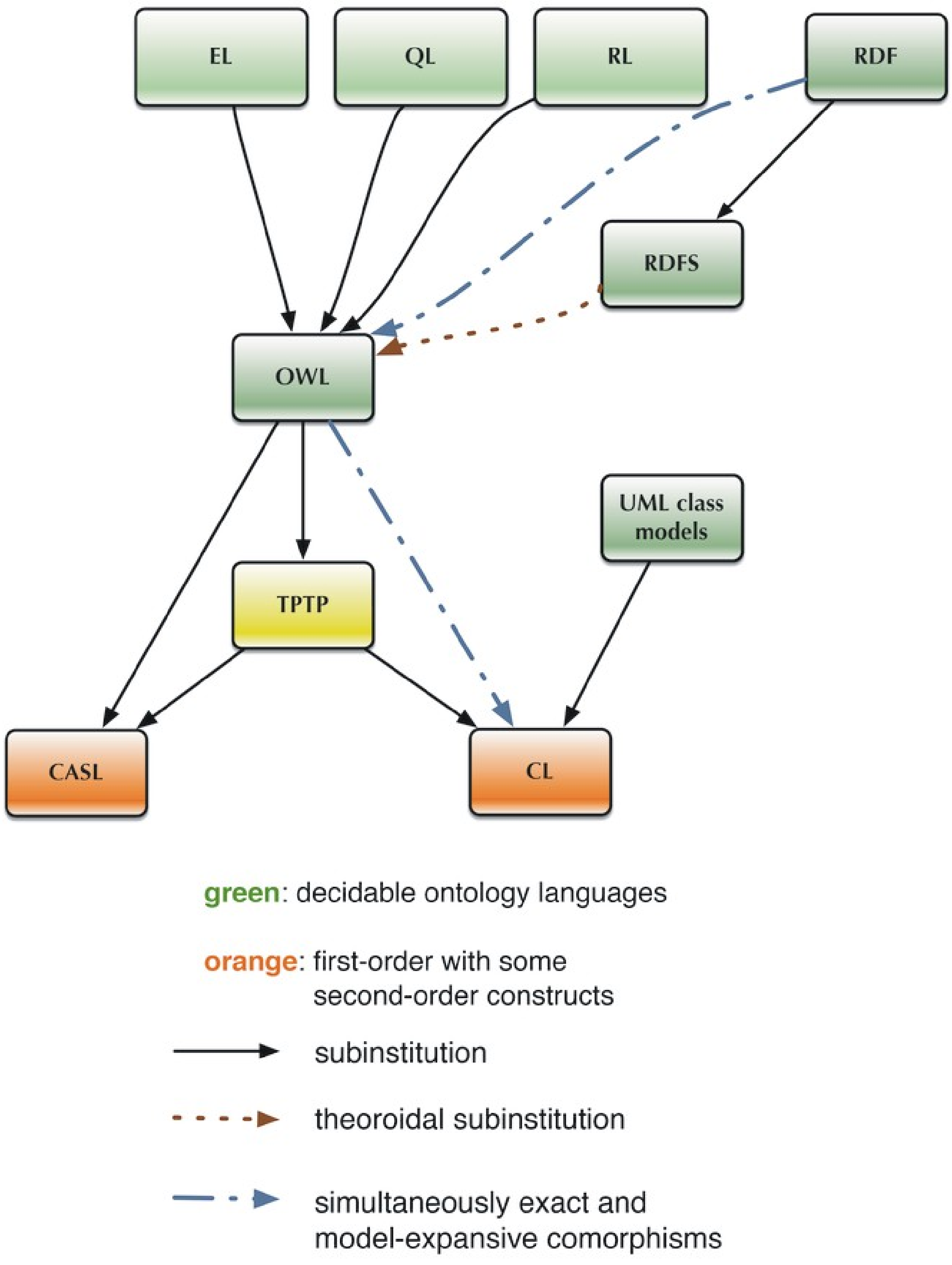
Mapping UML and the Semantic Web to CL

The DOL standard specifies formal mappings from the Semantic Web logics and UML diagrams to TPTP, CASL, and Common Logic.

Arrows show the mappings from less expressive logics to more expressive logics.

TPTP notation (for Thousands of Problems for Theorem Provers) is a version of many-sorted logic, of which classical first-order logic is a single-sorted subset.

CASL is the logic for HeTS (the Heterogeneous Tool Set), which implements the mappings.



Using DOL With KGs

Usage scenarios for DOL may be adapted to data from KGs. *

- Interoperability between OWL and FOL ontologies
- Module extraction from large ontologies
- Interoperability between closed-world data and open-world metadata
- Verification of rules translating Dublin Core into PROV
- Maintaining different versions of an ontology in languages with different expressivity
- Metadata within OMS repositories
- Modularity of specifications
- Refinement of specifications
- Consistency among UML models of different types
- Refinements between UML models of different types, and their reuse
- Coherent semantics for multi-language models

*** See Section 7 of the DOL standard (pp. 33 to 48).**

3. CLIP for Linear and Graphic Logics

CLIP is a dialect of Common Logic that has a direct mapping to and from predicate calculus and the graph notations for logic.

The existential graphs (EGs) by C. S. Peirce are the simplest graph logic with the full expressive power of Common Logic.*

The EG rules of inference and the methods for linearizing graphs may be adapted to KGs, UML, and SW notations.

Design goals for CLIP:

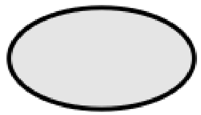
- **Immediately readable by anyone who knows predicate calculus.**
- **As readable as Turtle for the RDF and OWL subsets.**
- **As readable as any notation for if-then rules.**
- **Serve as a linearization for UML diagrams.**
- **Query option: Select (*list of names*) where (*any CLIP sentence*).**
- **Support mappings of KGs ↔ logics and languages.**

*** An introduction to existential graphs: http://jfsowa.com/talks/egintro.pdf**

Existential Graphs

A line for existence. An oval for negation. Conjunction is implicit.

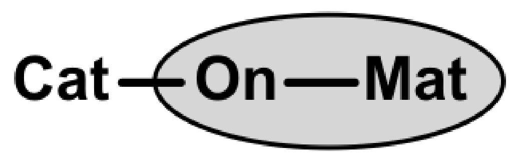
Existence: —

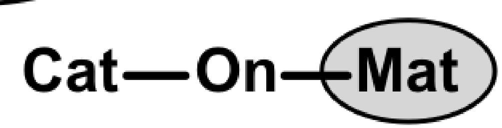
Negation: 

Relations: **Cat-** **-On-** **-Under-** **-With-** **-Mat**

A cat is on a mat: **Cat—On—Mat**

Something is under a mat: **—Under—Mat**

Some cat is not on a mat: 

Some cat is on something that is not a mat: 

The Core CLIP Notation

Core CLIP has a one-to-one mapping to and from EGs.

Existence: (∃ x) or (Exists x)

Negation: ~[ ]

Relations: (Cat x), (On x y), (Under x y), (Mat y)

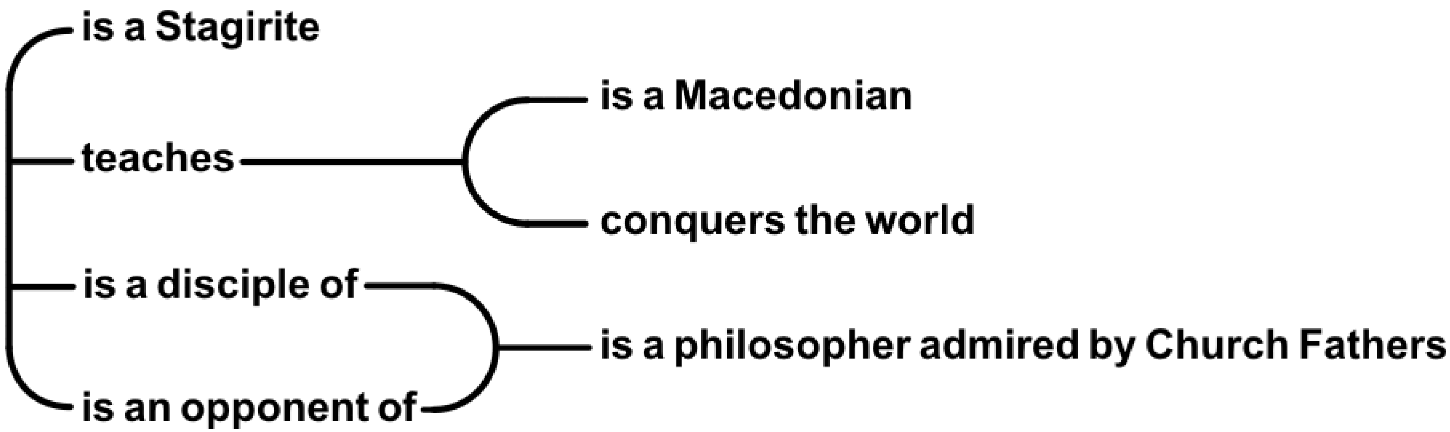
A cat is on a mat: (∃ x y) (Cat x) (On x y) (Mat y)

Something is under a mat: (∃ x y) (Under x y) (Mat y)

Some cat is not on a mat: (∃ x) (Cat x) ~[(∃ y) (On x y) (Mat y)]

Some cat is on something that is not a mat:
(∃ x y) (Cat x) (On x y) ~[(Mat y)]

One of Peirce's Examples



Peirce's translation to English: ***"There is a Stagirite who teaches a Macedonian conqueror of the world and who is at once a disciple and an opponent of a philosopher admired by Fathers of the Church."***

A translation to CLIP:

(∃ x y z) ("is a Stagirite" x) (teaches x y) ("is a Macedonian" y) ("conquers the world" y) ("is a disciple of" x z) ("is an opponent of" x z) ("is a philosopher admired by church fathers" z).

Without negation, CLIP can represent the content of a relational database or the graph databases of the Semantic Web.

4. Relating KGs to Natural Languages

For computers, informal mappings must be formalized.

- **Informal mappings to natural languages (NLs) are OK for humans.**
- **Informal mappings to RDF and OWL are OK for simple KGs.**
- **But anything a computer does is formal.**

Discourse Representation Theory specifies a subset of NLs.*

- **DRT is widely used for natural language processing (NLP).**
- **Discourse representation structures (DRSs) support full FOL.**
- **Every DRS has a precise mapping to and from CLIP.**

Semi-automated translation of NLs and KGs to and from CLIP.

- **Computer translation of NL → CLIP is error prone.**
- **Computer translation of CLIP → NL is precise, but verbose.**
- **Human translation depends on the human.**
- **Simpler and more reliable: Human-aided computer translation.**

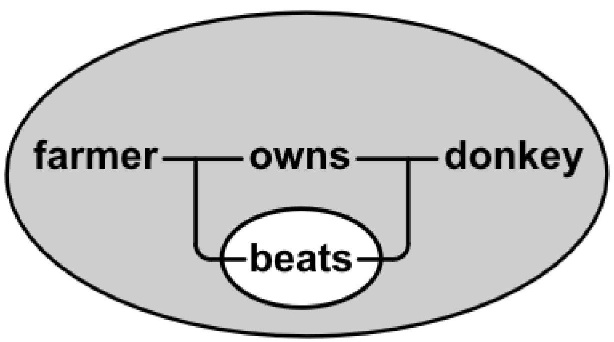
*** Hans Kamp & Uwe Reyle (1993) *From Discourse to Logic*, Dordrecht: Kluwer.**

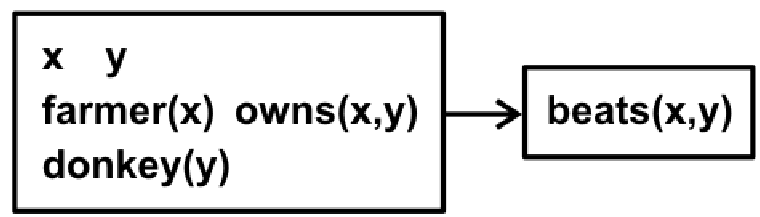
Mapping Language to Logic

C. S. Peirce and Hans Kamp designed graph logics.

- **Peirce chose nested ovals for EGs with lines to show references.**
- **Kamp chose boxes for DRS with variables to show references.**
- **But they chose equivalent operators and equivalent conventions for mapping languages to diagrams to linear logics.**

Example: ***If a farmer owns a donkey, then he beats it.***



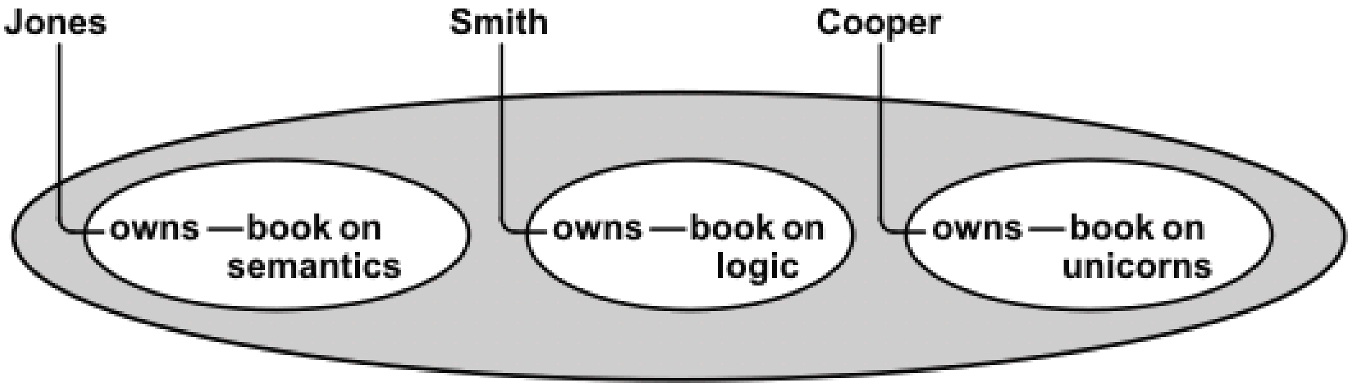


EG (left) and DRS (right) are mapped to exactly the same CLIP:
[If (∃ x y) (farmer x) (donkey y) (owns x y) [Then (beats x y)]].

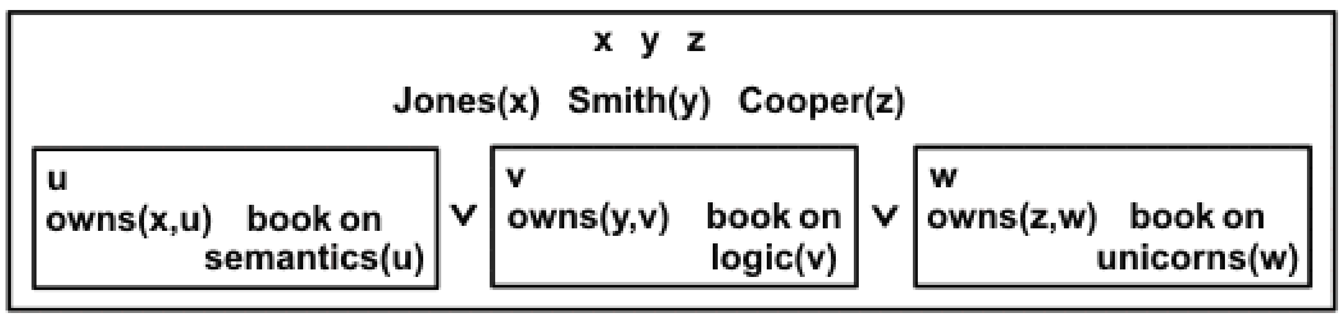
Disjunction in EG, DRS, and CLIP

Kamp and Reyle (1993): *“Either Jones owns a book on semantics, or Smith owns a book on logic, or Cooper owns a book on unicorns.”*

EG:



DRS:



CLIP:

(∃ x y z) (Jones x) (Smith y) (Cooper z)
[Or [(∃ u) (owns x u) ("book on semantics" u)]
[(∃ v) (owns y v) ("book on logic" v)]
[(∃ w) (owns z w) ("book on unicorns" w)]].

5. Metalanguage and Metadata

Metalanguage is language about language, natural or artificial.

- **To define semantics, Tarski (1933) used logic as a metalanguage for defining the truth value of any statement in logic.**
- **Like logic, KGs can state information or talk about information.**
- **A logic for KGs should also support metalevel KGs.**

The IKL extension to Common Logic supports metalanguage. *

- **IKL enables CLIP to comment on anything expressed in CLIP.**
- **It can represent metadata about the sources and reliability of data.**
- **It can support reasoning about metaphor, modality, and the issues of vague, fuzzy, missing, erroneous, or fraudulent information.**

Any Unicode strings may be used for names. Metacomments in CLIP may even be expressed by emojis.

* **For the IKL documents, see http://jfsowa.com/ikl/ .**

Metalanguage About Situations

The drawing on the right may be interpreted in three ways.

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

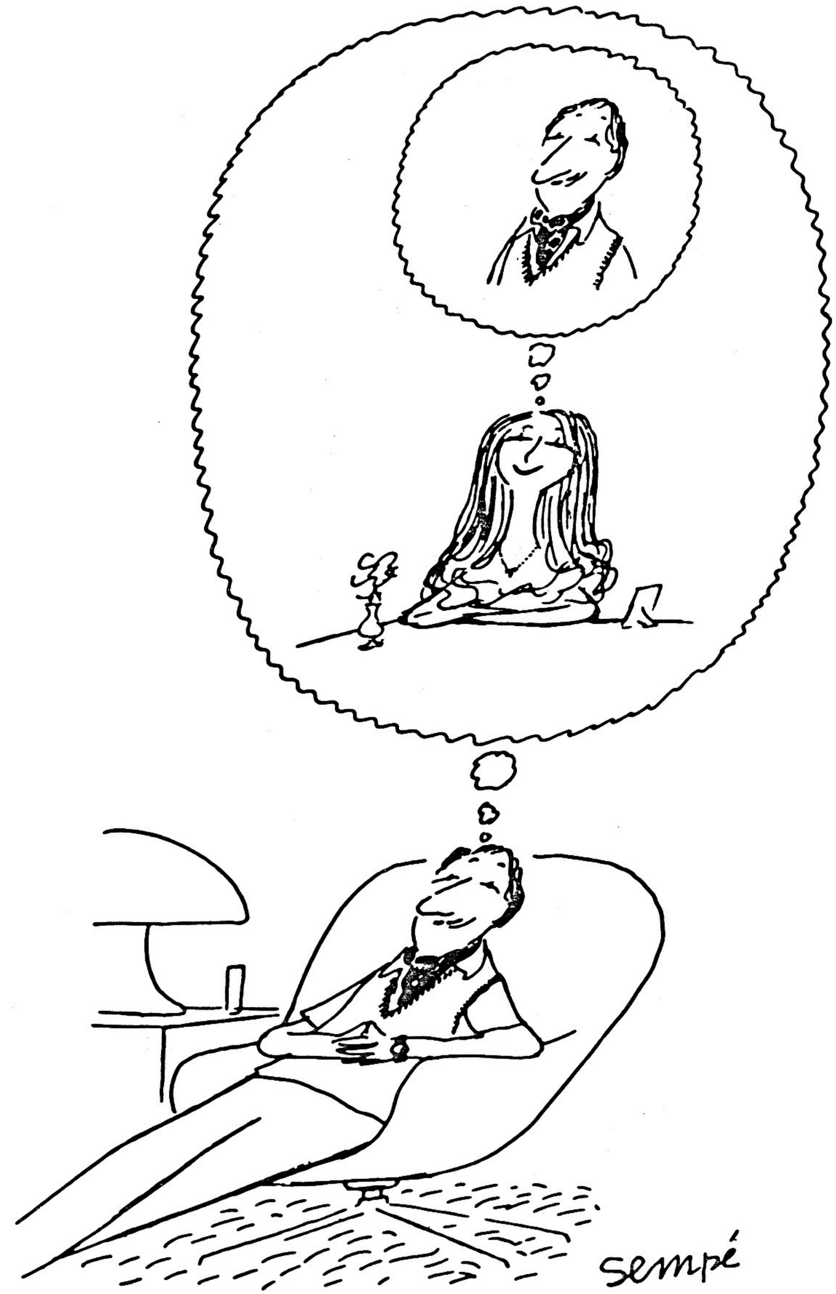
2. Modal: *Pierre is thinking of Marie, who may be thinking of him.*

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

In the second clause of #1, the verb *is* implies that Pierre's thought is true.

In #2, the verb *may* implies that his thought is a possible proposition.

In #3, the object of the verb *hopes* is a situation Pierre intends in some way.



Propositions and Situations

Three ways of relating a proposition p to a situation s.

- **Actual: p is true or false about s.**
- **Modal: p is related to s in some manner or mode m.**
- **Intentional: Some agent x relates p to s for some reason r..**

English and CLIP for the sentences about Pierre.

- **English:** ***Pierre is thinking of Marie, who is thinking of him.***
- **CLIP: (thinkingOf Pierre Marie) (thinkingOf Marie Pierre).**

- **Enlish:** ***Pierre is thinking of Marie, who may be thinking of him*.**
- **CLIP: (thinkingOf Pierre Marie) (possible [(thinkingOf Marie Pierre)]).**

- **English:** ***Pierre hopes that Marie is thinking of him*.**
- **CLIP: (hopesFor Pierre [Situation (thinkingOf Marie Pierre)])**

Requirements, planning, analysis, and explanations require modal, temporal, and intentional ways of talking and reasoning.

6. Automated and Semi-automated Tools

Computers should ask humans for help and explanations.

- **The experts in any field rarely have a PhD in computer science. Even computer scientists don't know the jargon of every system.**
- **Computers should accept any language or notation people prefer, and they should read documents without requiring prior annotations.**
- **If a computer can't understand some text, it should ask people for help. People should answer in their own language.**
- **Computers may annotate texts, but human assistance is necessary when a computer is uncertain about the interpretation.**

Analogies can support informal, case-based reasoning:

- **Cognitive Memory can efficiently find analogies in Big Data.**

Formal reasoning is based on a disciplined use of analogy:

- **Theorem provers use a kind of analogy called *unification*.**

Two Logicians and a General

Alfred North Whitehead:

“Human knowledge is a process of approximation. In the focus of experience, there is comparative clarity. But the discrimination of this clarity leads into the penumbral background. There are always questions left over. The problem is to discriminate exactly what we know vaguely.”

Charles Sanders Peirce:

“It is easy to speak with precision upon a general theme. Only, one must commonly surrender all ambition to be certain. It is equally easy to be certain. One has only to be sufficiently vague. It is not so difficult to be pretty precise and fairly certain at once about a very narrow subject.”

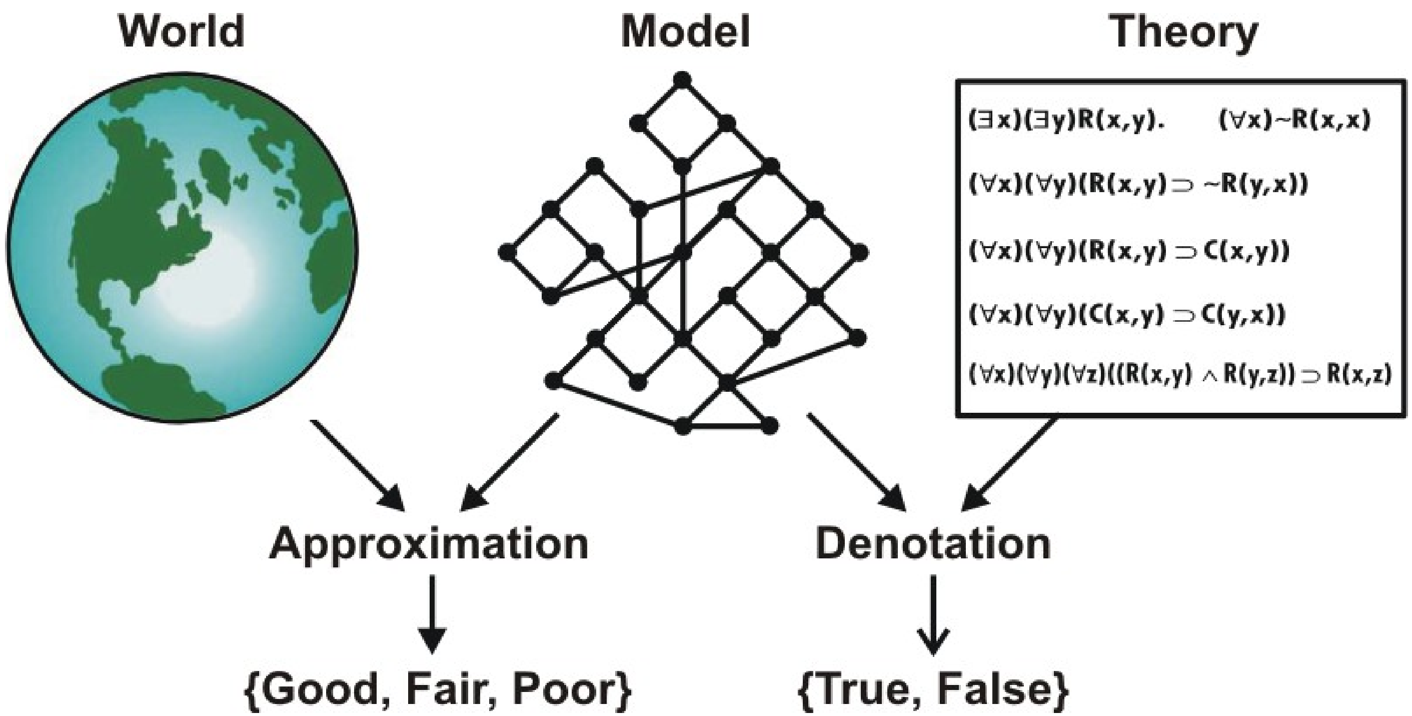
Alfred North Whitehead:

“We must be systematic, but we should keep our systems open.”

Dwight David Eisenhower:

“Plans are worthless, but planning is essential.”

Relating Models to the World



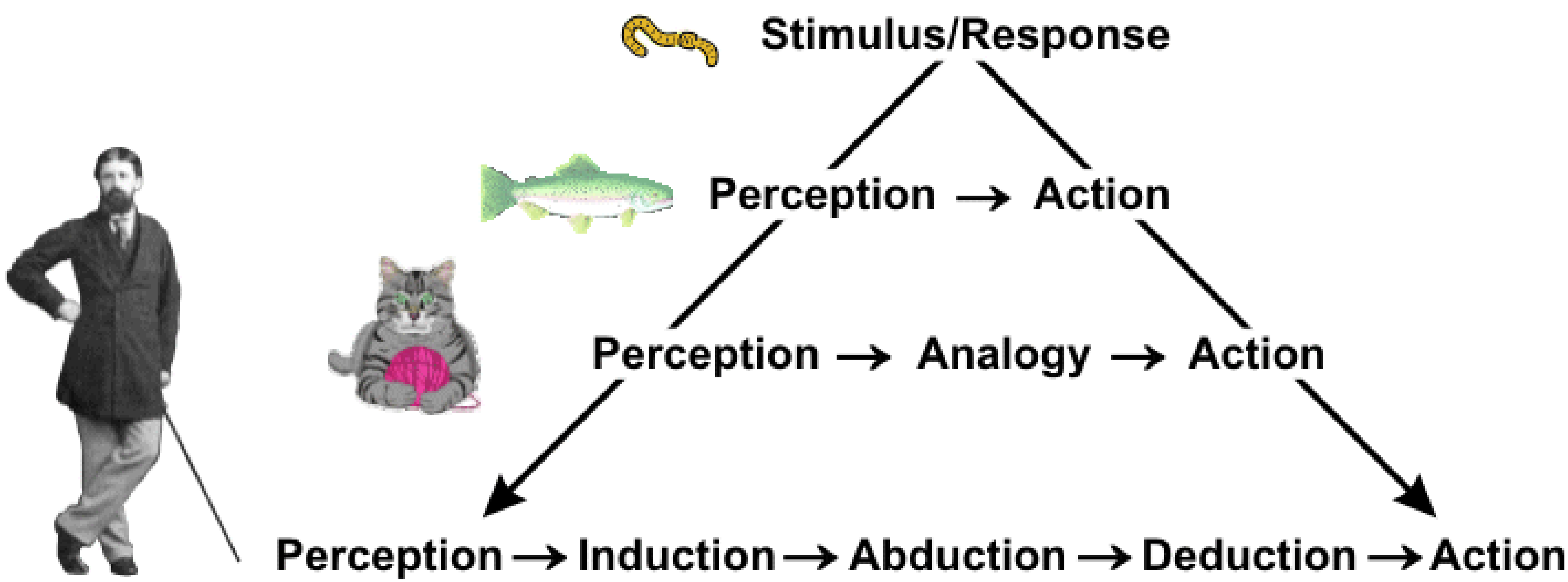
Engineers: "All models are wrong, but some are useful."

- **Discrete symbolic models can be clear, sharp, and precise.**
- **But the world is continuous, disordered, and fuzzy.**

Natural languages are flexible. They can adapt to anything.

- **They can be as vague or precise as the situation requires.**
- **KG tools should be flexible: Detailed levels must be precise, but the top-level ontology must be "sufficiently vague."**

Thinking Beyond the First Second



Perception and classification take one second or less.

- Artificial NNs are very good for learning and recognizing patterns.
- By themselves, NNs support S-R (knee-jerk) reactions.
- Classification by analogy can support a cat-level intelligence.

Analysis, planning, and development take more time.

- They require cycles of induction, abduction, deduction...