Universal Query Language

Declarative or procedural, linear or graphic

John F. Sowa

11 August 2021

1. Questions and Answers

A question or query is an incomplete statement.

- **It has two parts: a request and a constraint.**
- **The request names or describes some unknown things.**
- **The constraint specifies relations among those things.**
- **An answer is information that completes the statement.**

Examples:

- **In English, requests are marked by wh- words; in Latin. by qu- words; in Polish, Russian, and Japanese, by k- words.**
- **In SQL, a request begins with the keyword *SELECT*, and a constraint begins with the keyword *WHERE*.**

The question words *why* and *how* ask for explanations. *

- **An answer would be a proposition or a procedure.**

*** In English, the initial w is silent in *who*; it is silent and invisible in *how*.**

Defining a Query Language

The definition must specify three aspects of the language:

- Syntax: The patterns of symbols that people read and write.
- Semantics: Formal structure, inference rules, and truth values.
- Pragmatics: Support for the intended applications.

Semantics is fundamental.

- External syntax is designed for human factors: readability and ease of use with maximal speed and minimal error rate.
- But the external form is irrelevant for the computer because a translation to the internal form is rapid and error-free.
- Pragmatics is determined by the applications, and different applications may require different versions of semantics.
- For maximum generality, a general semantics must support all the subsets that any applications may require.

The DOL standard specifies a general semantics.

2. DOL Standard

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

- DOL tools can relate anything specified in an open-ended variety of notations, linear or graphical.
- UML and the Semantic Web are supported by DOL.
- DOL can also represent the notations for legacy software and the latest technologies of the 21st century.

DOL is specified in formal logic.

- Logic is essential for guaranteeing precision.
- DOL supports the very general Common Logic (CL) and all the logics that can be mapped to CL.
- But people may use any notations and logics they prefer.

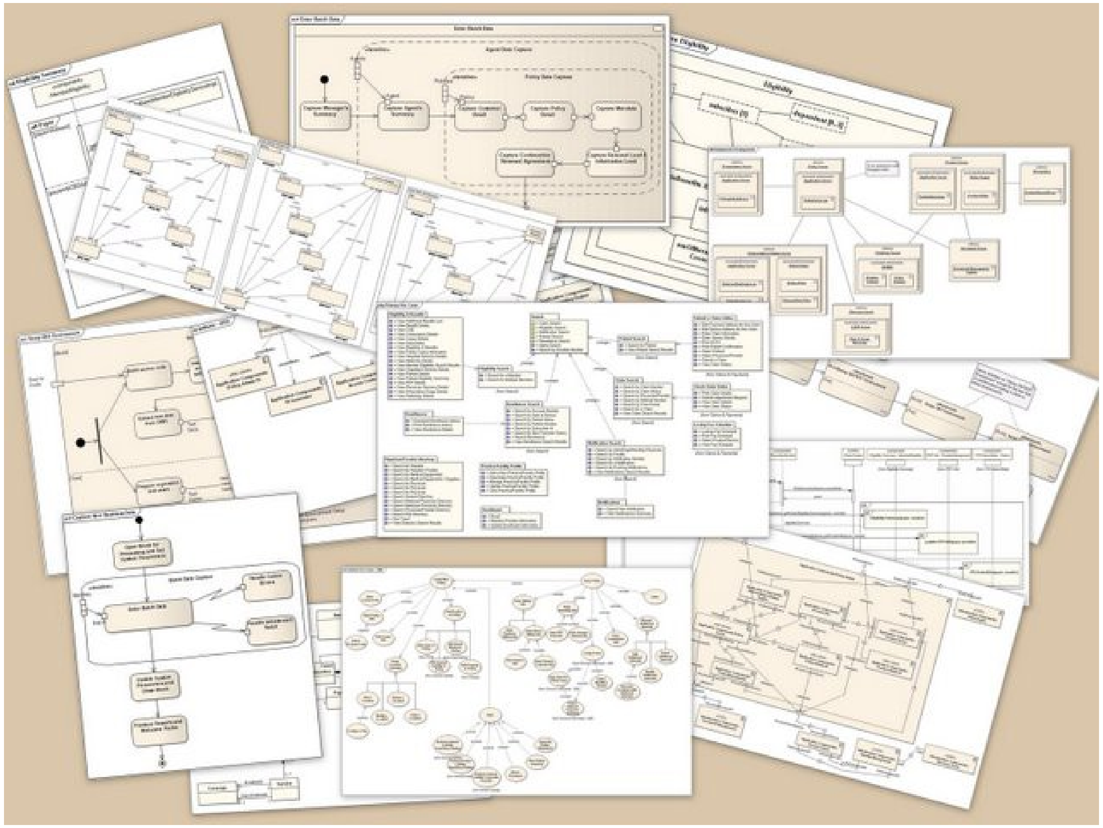
* DOL is standardized by the Object Management Group: Distributed Ontology, Modeling, and Specification Language: https://www.omg.org/spec/DOL/1.0

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 in any field.

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

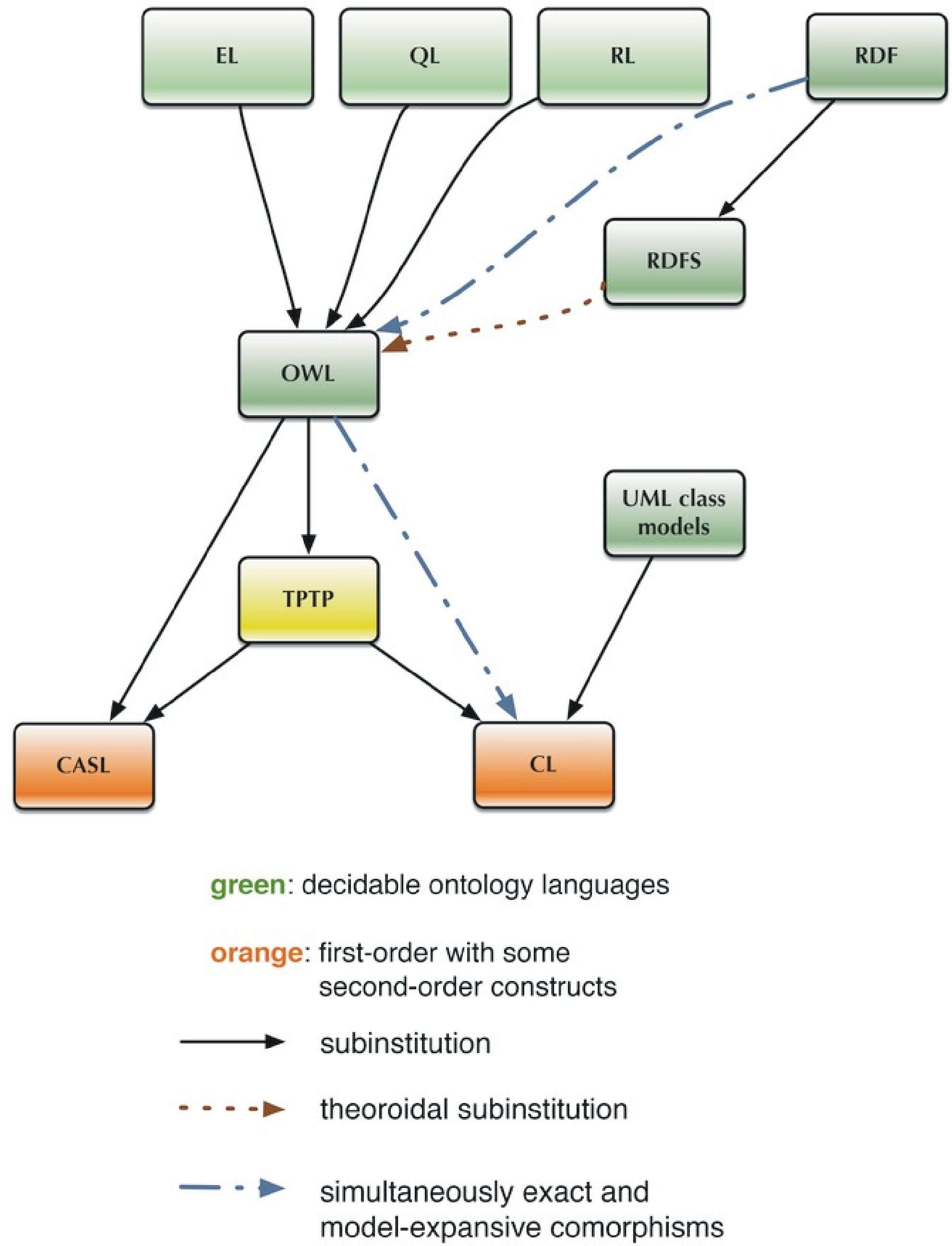
Mapping UML and the Semantic Web to CL

The diagram shows the most widely used logics supported by DOL.

Arrows show the mappings from less expressive logics to more expressive logics. Common Logic is at the lower right.

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.

HeTS (the Heterogeneous Tool Set) uses CASL as the interchange logic for this diagram.   But other tools may use other logics.



Supporting Interoperability

A programmer's lament at a database symposium:

- *Any one of those tools, by itself, is a tremendous aid to productivity, but any two of them together will kill you.* *

Usage scenarios for DOL (Section 7 of the DOL standard):

- **Interoperability between OWL and FOL ontologies**
- **Module extraction from large ontologies**
- **Interoperability between closed-world data and open-world metadata**
- **Verification of rules for translating Dublin Core into PROV**
- **Maintaining different versions of an ontology in languages with different expressivity**
- **Metadata within OMS repositories**
- **Modularity and 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**

*** Comment by Terry Rankin, circa 1980. But it's just as true today.**

Computability and Decidability

The logics for the Semantic Web are decidable.

- **But decidabiity is a property of the problem, not the notation.**
- **The best TPTP systems use syntactic checks to determine the methods to use for any particular problem.**
- **For the same problems, TPTP systems are as fast or faster than the tools designed for the Semantic Web.**

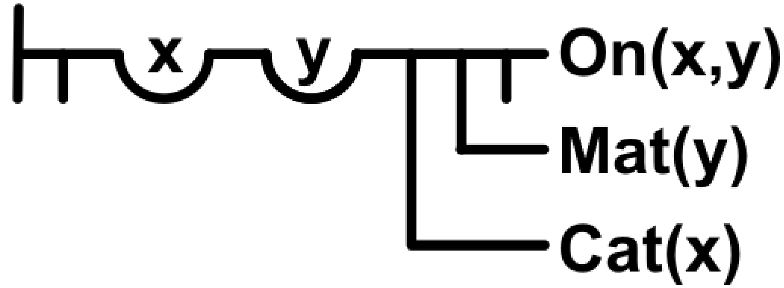
Restricting expressive power cannot improve performance. *

- **It just makes certain problems impossible to state.**
- **Natural languages are more expressive than any version of logic.**
- **But the only people who can state an undecidable sentence are those who have studied advanced logic and mathematics.**
- **Users always ask for more expressive power. They never ask for decidability.**

*** See "Fads and fallacies about logic," http://jfsowa.com/pubs/fflogic.pdf**

How to say “A cat is on a mat.”

Gottlob Frege (1879):

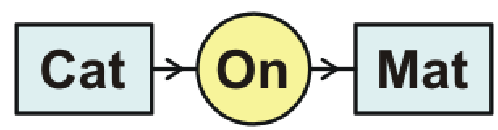


Charles Sanders Peirce (1885): $\Sigma_x\ \Sigma_y\ \mathrm{Cat}_x \bullet \mathrm{On}_{x,y} \bullet \mathrm{Mat}_y$

Giuseppe Peano (1895): $\exists x\ \exists y\ \mathrm{Cat}(x) \wedge \mathrm{On}(x, y) \wedge \mathrm{Mat}(y)$

Existential graph by Peirce (1897): Cat —— On —— Mat

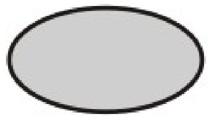
Conceptual graph (1976):



CLIP dialect of Common Logic: (∃ x y) (Cat x) (On x y) (Mat y).

Existential Graphs (EGs)

Existence: —

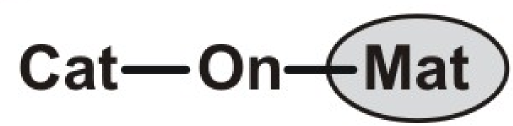
Negation: 

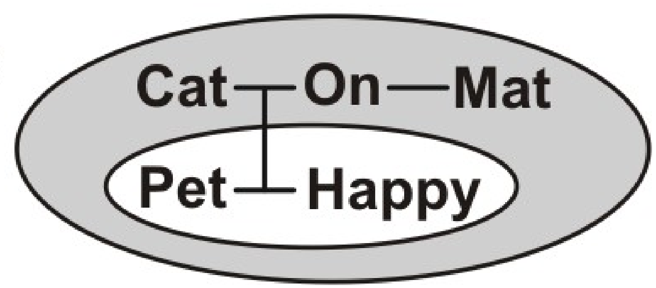
Relations: Cat• Mat• Happy• •On• •Under• •Give•

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: 

If a cat is on a mat, then it is a happy pet: 

The Core CLIP Notation

Existence: (∃ x) or (Exists x)

Negation: ~[ ] but ~[~[ ]] may be written [If [Then]]

Relations: (Cat x), (Mat x), (Pet x), (Happy x), (On x y), (Under x 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)].

If a cat is on a mat, then it is a happy pet:
[If (∃ x y) (Cat x) (On x y) (Mat y)
[Then (Pet x) (Happy x)]].

3. Universal Query Language (UQL)

The DOL framework can support any computable logic.

- DOL + Common Logic can support the major logics in use today.
- DOL can be extended to support any new logics of any kind.
- The external notations may be graphic or linear. But to be computable, the graphs must have a linearized internal form.

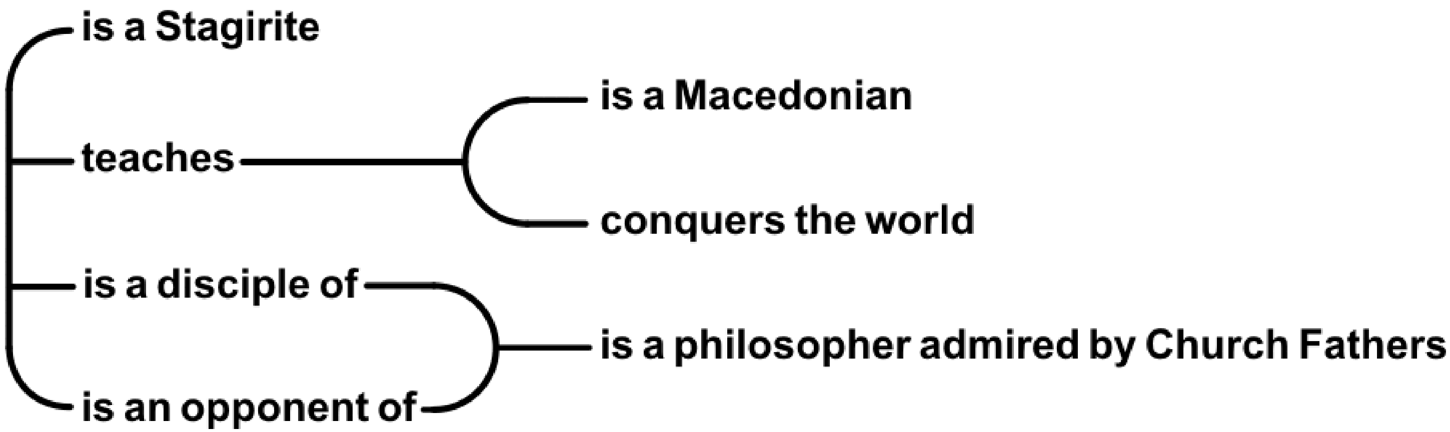
Notations for the request and constraint clauses:

- In SQL, the keywords SELECT and WHERE mark the clauses.
- In CLIP, the symbol '?' may be used as a kind of quantifier.
- Other conventions can be mapped to these two forms.

For representing the examples in these slides,

- The CLIP dialect of Common Logic is used as the linear notation.
- Existential graphs (EGs) are used as the graphic notation.
- Other notations, linear or graphic, can be mapped to CLIP or EGs.

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:

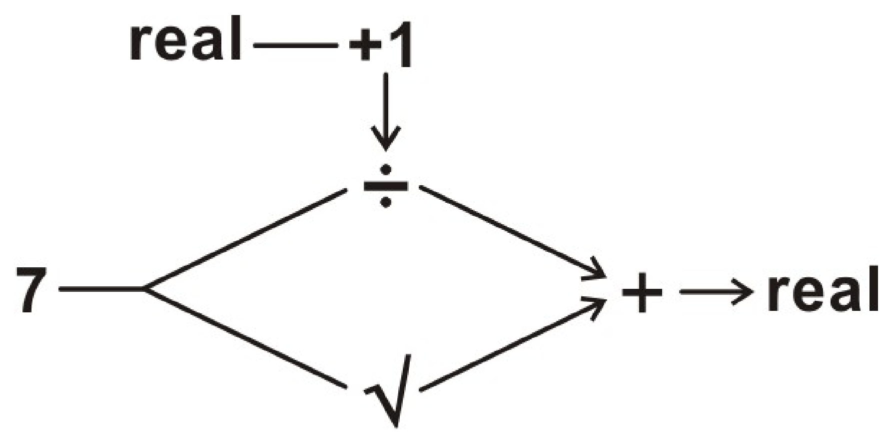
> **(∃ x y z) ("is a Stagirite" x) (teaches x y) ("is a Macedonian" y) ("conquers the world" y) ("is a disciple of" x z) ("is an opponent of" x z) ("is a philosopher admired by church fathers" z).**

Without negation, CLIP and EGs can represent the content of a relational DB, a graph DB, or anything in RDF or RDFS.

Representing Functions

An example in mathematical notation: $y = 7 \div (x + 1) + \surd\, 7.$

In EGs, a function may be represented as a relation with an arrow for its last line of identity. The four functions may be named +1, ÷, √, +.



A direct mapping of the EG to CLIP:

(∃ (x y u v w):real) (+1 x → u) (÷ 7 u → v) (√ 7 → w) (+ v w → y).

Another option of CLIP avoids the need for the names u, v, w:

(∃ (x y):real) (= y (+ (÷ 7 (+1 x)) (√ 7)))).

Second-Order Logic

Quantifiers range over functions, relations, and propositions.

English: *Bob and Sue are related*.
CLIP: (related Bob Sue).

English: *There is a familial relation between Bob and Sue*.
CLIP: (∃ r) (familial r) (relation r) (r Bob Sue).

English: *Every numeric function maps numbers to numbers*.
CLIP: (∀ f) (numeric f) (function f)
(∀ x y) [If (f x → y) [Then (number x) (number y)]].

Literal translation of CLIP to English: *For any numeric function f and any x and y, if f maps x to y, then x is a number and y is a number*.

Note: Higher-order logic with a hierarchy of infinite sets is extremely inefficient for computer processing. But Common Logic is a version of second-order logic that is as efficient as FOL.

Relating Logic to Natural Languages

For computers, informal mappings must be formalized.

- **Informal mappings to natural languages (NLs) are OK for humans.**
- **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.**
- **The DRS logic has a precise mapping to EG and to CLIP.**

Semi-automated translation of natural languages:

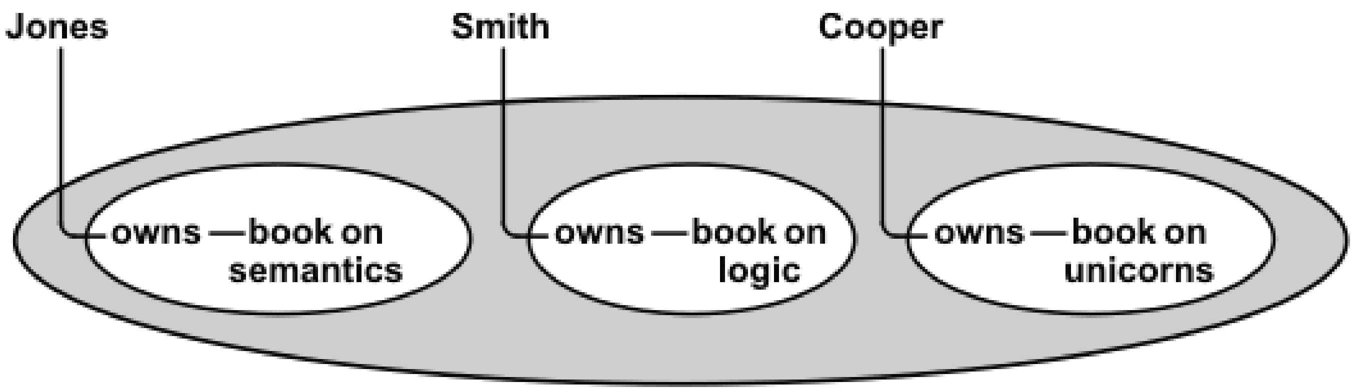
- **Computer translation of NL → CLIP is error prone.**
- **Computer translation of CLIP → NL is precise, but verbose.**
- **Human translation is as reliable as the human.**
- **Simpler and more reliable: Human-aided computer translation.**

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

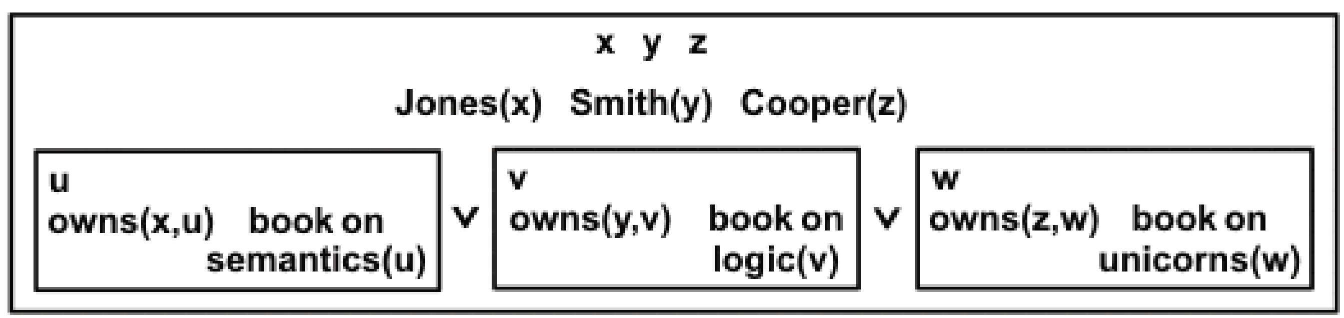
Example of EG, DRS, and CLIP

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:



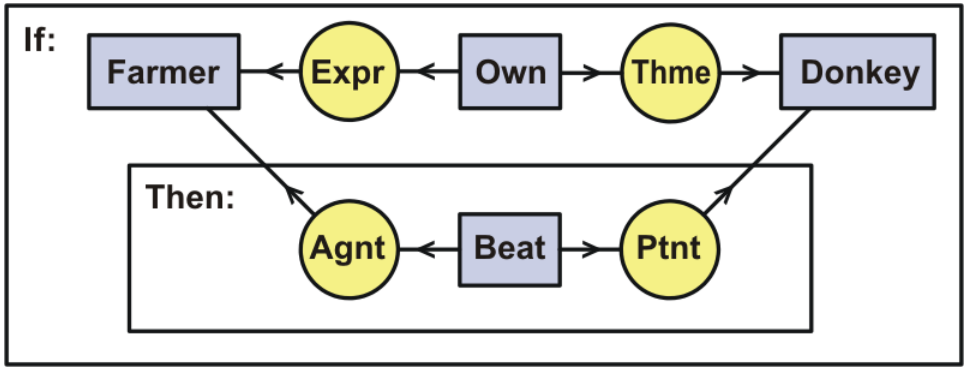
DRS:



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)]].

Conceptual Graphs



Conceptual graphs (CGs) express the same logic as EGs, but they emphasize the details of NL semantics. *

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

CLIP: [If (∃ x:farmer y:own z:donkey) (Expr y x) (Thme y z)
[Then (∃ w:beat) (Agnt w x) (Ptnt w z)]].

Unlike EGs, quantifiers in CGs are represented by boxes, not lines.

Names may refer to concept boxes that represent verbs.

The *semantic* or *thematic roles* used in linguistics relate verbs to nouns: experiencer (Expr), theme (Thme), agent (Agnt), and patient (Ptnt).

*** See "From EGs to CGs", http://jfsowa.com/pubs/eg2cg.pdf**

4. Requests and Constraints

A request is similar to the quantifier prefix of a proposition.

- **It names one or more variables that occur in the constraint.**
- **It may also specify the type or sort of some variables.**
- **For a *what* question, the variable represents a thing; for *who*, a person; for *where*, a place; for *when*, a time; for *how*, a method; for *why*, a proposition or a conjunction of multiple propositions.**

A constraint states relations among variables in the request.

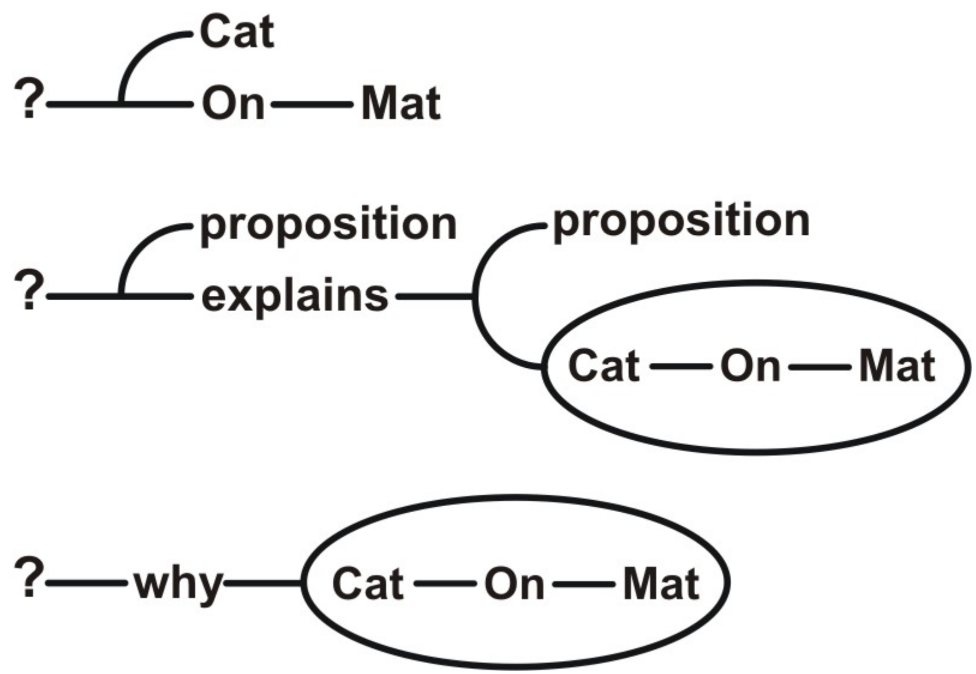
- **Variables in the request are free variables in the constraint.**

In existential graphs, a request consists of one or more lines of identity, each with the symbol ? attached to one end.

In CLIP, a request is represented in the same pattern as a quantifier, but with the symbol ? Instead of ∃ or ∀.

Queries in other notations may be mapped to EGs or CLIP.

Asking Questions in EGs and CLIP



Three questions:

- Which cat is on a mat?
- Which proposition explains the proposition that a cat is on a mat?
- Why is a cat on a mat? /* After defining a relation named 'why' */

Translations of the EGs to CLIP:

- (? x:Cat) (∃ y:Mat) (On x y).
- (? p:proposition) (∃ q:proposition) (explains p q)
  (that (∃ x:Cat y:Mat) (On x y) → q).
- (? p) (∃ q) (why p q) (that (∃ x:Cat y:Mat) (On x y) → q).

5. The Context of a Query

The same question, when asked by different people in different contexts, may require different answers.

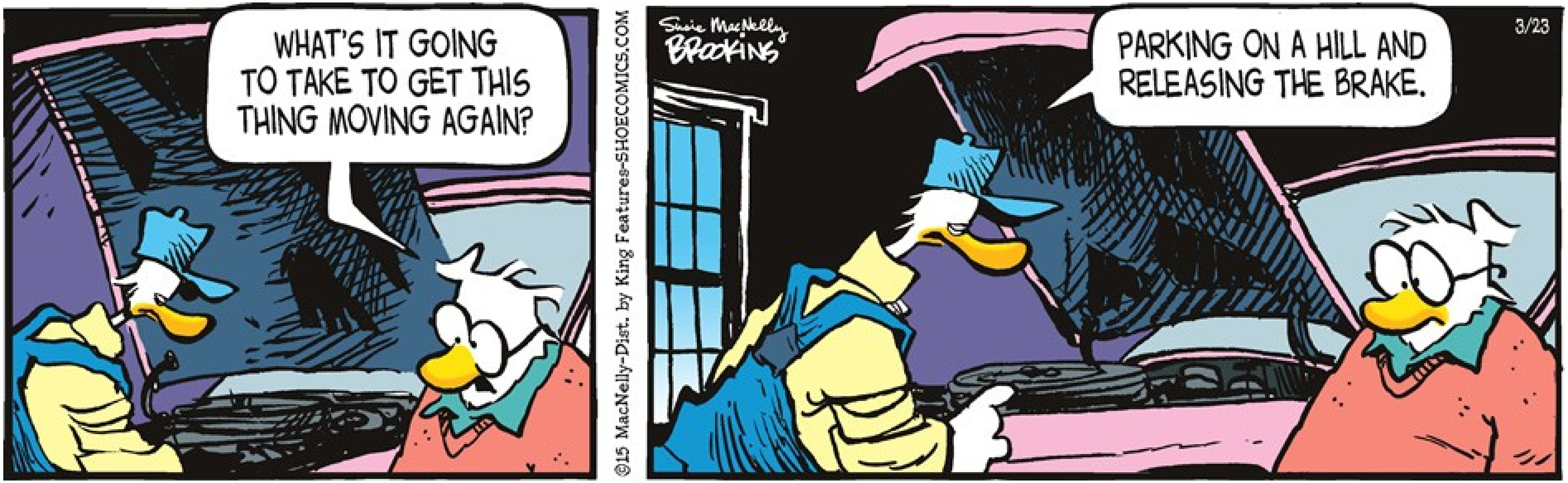
Any contextual information that people can state in ordinary language can be translated to logic.

But some information cannot be described in any language:

1. Mathematics can define continuous spaces and fields, but it cannot precisely describe an arbitrary instance of such a space.
2. The discrete words of any language have the same limitation.
3. The continuous volume and intonation of speech can supplement the discrete words, but not with any degree of precision.
4. What people cannot say in language cannot be said in logic.
5. But sometimes, people can answer a question by an action.

For computer queries, any context that can be stated in logic can be added to the constraint.

Context and Purpose



Syntax is easy: Parse the question and the answer.

Semantics depends on context and background knowledge:

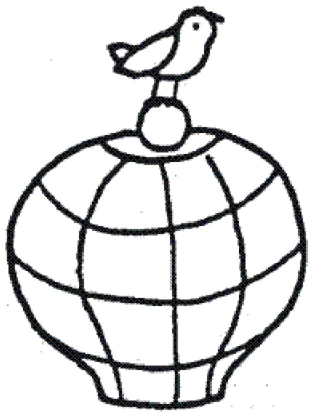
- **Interpret the meaning of *thing*, *take*, and *move* In this situation.**
- **Apply the laws of physics to determine what would happen.**

Pragmatics depends on the intentions of the participants.

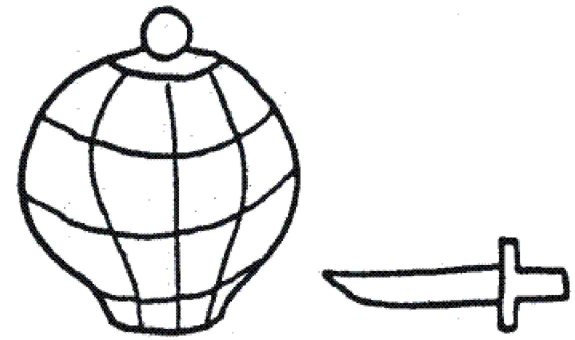
- **No computer system today could understand that cartoon.**
- **Computers should ask people about purpose or intentions.**

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

Effect of Background Knowledge



l'oiseau est à l'extérieur de la cage
the bird is outside the cage



? le couteau est à l'extérieur de la cage
the knife is outside the cage

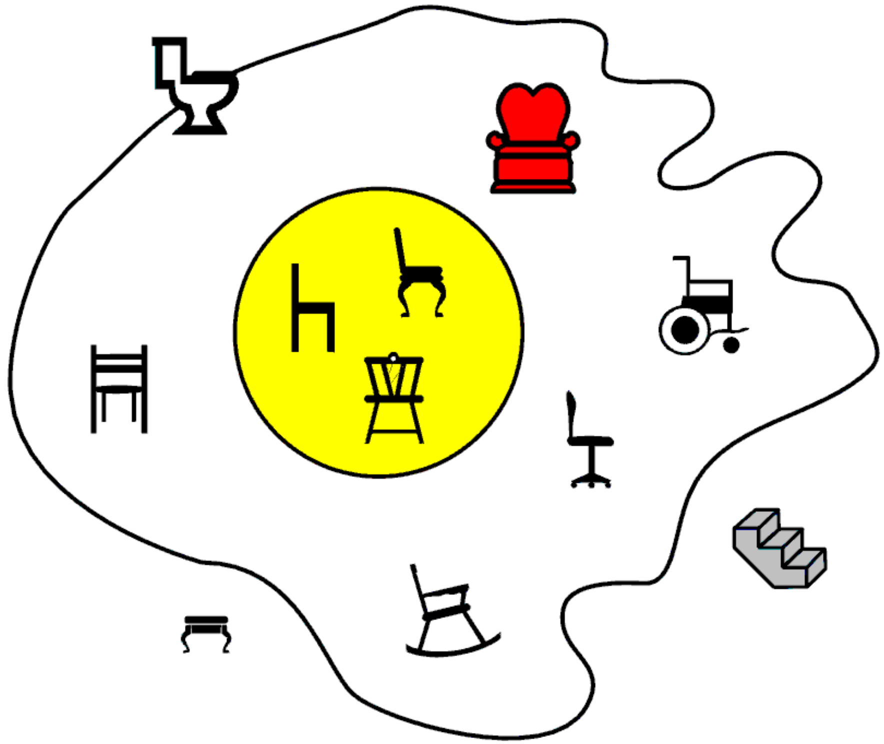
A cage is sometimes used to enclose a bird.

But a cage is an unlikely container for a knife.

Normal comment: "The knife is to the right of the cage."

To say "The knife is outside the cage" implies that there is some reason why it might have been in the cage.

What is a Chair?



An egg-yolk diagram puts typical examples in the yolk and less common variants in the egg white. (Lehmann & Cohn 1994)

Boundaries resemble the *level cuts* of fuzzy set theory: the fuzzy value 0.9 could be the boundary for the yolk, and 0.7 for the egg white.

But the reasons for the variations are more significant than the numbers.

Is it a Chair? Art? Humor? Fantasy?

In a museum, it's funny. But suppose you saw it at night in an old castle. Meaning is always context dependent.

Claw and Ball Chair by Jake Cress. At the Smithsonian Renwick Gallery.

Zachman ISA Framework

The Information System Architecture by John Zachman.

- **An ontology and framework for system design and development.**
- **Different people for different reasons may have different views: Planner, director, designer, builder, subcontractor, and user.**

The next slide shows an example: *

- **The ISO report on Conceptual Schemas (1987) developed an example about the Oz Car Registration Authority (OCRA).**
- **The 36 boxes show 36 perspectives on OCRA: 6 kinds of questions by 6 kinds of people.**
- **Noun phrases are the answers to What? Where? Who? When?**
- **Actions, procedures, or functions are answers to How?**
- **Sentences or paragraphs are answers to Why?**

But there is no limit to the number or kinds of perspectives.

*** Extending and formalizing the ISA framework, by John Sowa & John Zachman.**

Oz Car Registration Authority

	What?	How?	Where?	Who?	When?	Why?
Scope Planner	Oz, OCRA, cars, fees, licenses, car histories.	Register, transfer, collect, enforce.	Emerald City, Munchkin Land, Kansas, Hollywood.	Director, managers, clerks, car owners.	Time of sale, transfer, registration destruction.	Regulate sales, raise money, trace cars.
Enterprise Director	Each car is of a particular model.	Ownership is transferred by reistration of the transfer.	Registrations are recorded at offices of OCRA.	An OCRA clerk must record each registration.	When a car is constructed, transferred, or destroyed.	Keep accurate records and collect fees.
System Designer	Functional dependency from car to model.	Car history updated by transfer module.	Each office must have a connection to OCRA HQ.	A clerk must enter information at a terminal.	DB updates occur at irregular intervals.	Old batch system does not respond fast enough.
Technology Builder	Car relation has a column for model identifier.	Transfer done by COBOL program XFTR397A.	Branch office records are backed up at OCRA HQ.	Clerk completes form REG972 to initiate registration.	Each module is invoked by a menu selection.	Efficient reliable service within budget.
Components Sub-contractor	Model ID PIC X(15).	SELECT SNO FROM HIST WHERE...	Install TCP/IP link to OZNET.	Install cordons to guide queue for clerks.	Use pop-up windows selected by mouse.	Meet specifications for each module.
Working system.	Data about people, places, things, events.	Functions performed.	Places on the network.	Organization.	History, plans, and schedule.	Motivation and strategy.

The Context of a Conversation

A transcript, with background knowledge in italics: *

Husband: **Dana succeeded in putting a penny in a parking meter today without being picked up.**

> ***This afternoon as I was bringing Dana, our four-year-old son, home from the nursery school, he succeeded in reaching high enough to put a penny in a parking meter when we parked in a meter zone, whereas before he has always had to be picked up to reach that high.***

Wife: **Did you take him to the record store?**

> ***Since he put a penny in a meter that means that you stopped while he was with you. I know that you stopped at the record store either on the way to get him or on the way back. Was it on the way back, so that he was with you or did you stop there on the way to get him and somewhere else on the way back?***

Husband: **No, to the shoe repair shop.**

> ***No, I stopped at the record store on the way to get him and stopped at the shoe repair shop on the way home when he was with me.***

*** From page 9 of Confronting context effects, by Keith Devlin (2005).**

Continuation

Wife: **What for?**

> ***I know of one reason why you might have stopped at the shoe repair shop. Why did you in fact?***

Husband: **I got some new shoe laces for my shoes.**

> ***As you will remember I broke a shoe lace on one of my brown Oxfords the other day so I stopped to get some new laces.***

Wife: **Your loafers need new heels badly.**

> ***Something else you could have gotten that I was thinking of. You could have taken in your black loafers which need heels badly. You'd better get them taken care of pretty soon.***

Observation by Devlin: ***"The task*** **[*of specifying all the relevant context*]** ***was endless. At every stage, what has been specified is dependent on further contextual factors."***

These issues are outside the scope of a query language.

Related Readings

ISO/IEC standard 24707 for Common Logic, http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007(E).zip

Object Management Group, Distributed Ontology, Modeling, and Specification Language (DOL), https://www.omg.org/spec/DOL/1.0

Majumdar, Arun K., John F. Sowa, & John Stewart (2008) Pursuing the goal of language understanding, http://www.jfsowa.com/pubs/pursuing.pdf

Majumdar, Arun K., & John F. Sowa (2009) Two paradigms are better than one and multiple paradigms are even better, http://www.jfsowa.com/pubs/paradigm.pdf

Petarinen, Ahti-Veikko (2011) Moving pictures of thought II, *Semiotica* 186:1-4, 315–331, http://www.helsinki.fi/~pietarin/publications/Semiotica-Diagrams-Pietarinen.pdf

Putnam, Hilary (1982) Peirce the logician, http://jfsowa.com/peirce/putnam.htm

Sowa, John F. (2006) Peirce's contributions to the 21st Century, http://www.jfsowa.com/pubs/csp21st.pdf

Sowa, John F. (2006) Worlds, models, and descriptions, http://www.jfsowa.com/pubs/worlds.pdf

Sowa, John F. (2007) Fads and fallacies about logic, http://jfsowa.com/pubs/fflogic.pdf

Sowa, John F. (2010) Role of logic and ontology in language and reasoning, http://www.jfsowa.com/pubs/rolelog.pdf

Sowa, John F. (2013) From existential graphs to conceptual graphs, http://www.jfsowa.com/pubs/eg2cg.pdf

Sowa, John F. (2020) Language, Ontology, and the Semantic Web, http://jfsowa.com/talks/eswc.pdf