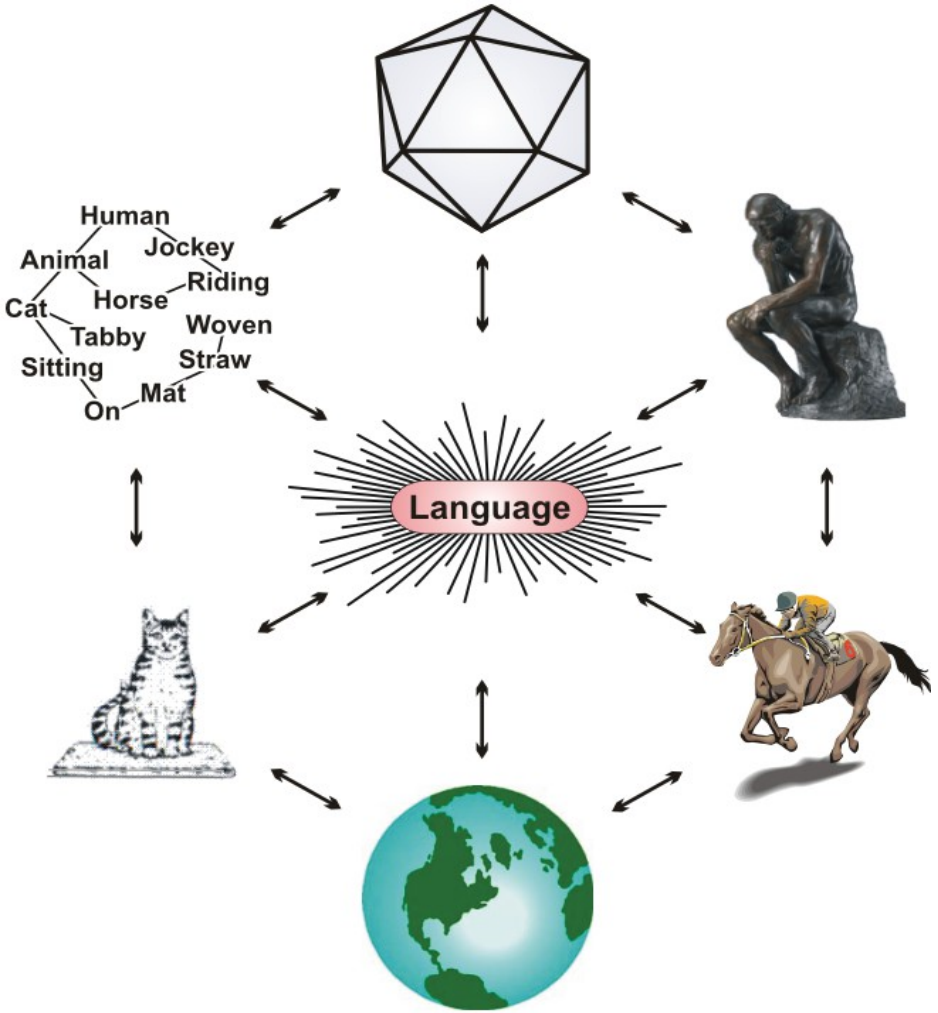
Automated and Semi-automated Methods for Deriving Ontologies



John F. Sowa

24 March 2021

Automated and Semi-automated Methods

Abstract: *In 2000, Tim Berners-Lee proposed a vision for the Semantic Web that was more ambitious than the results delivered in 2005. Research in the past 15 years produced advanced technology in artificial intelligence, language processing, and reasoning methods, both formal and informal. But many systems are proprietary, incompatible with one another, and too complex for widespread adoption. Among the most important requirements, trusted systems were never adequately implemented. This talk surveys promising developments and suggests ways of adapting them to the Semantic Web.*

Contents:

1. The Semantic Web from 2000 to 2005
2. Interoperability among heterogeneous systems
3. Common Logic as the Semantic Web Logic Language
4. Mapping logic to and from natural languages
5. Cognitive memory
6. Automated and semi-automated tools

1. Semantic Web From 2000 to 2005

In 2000, Tim Berners-Lee wrote an ambitious proposal.*

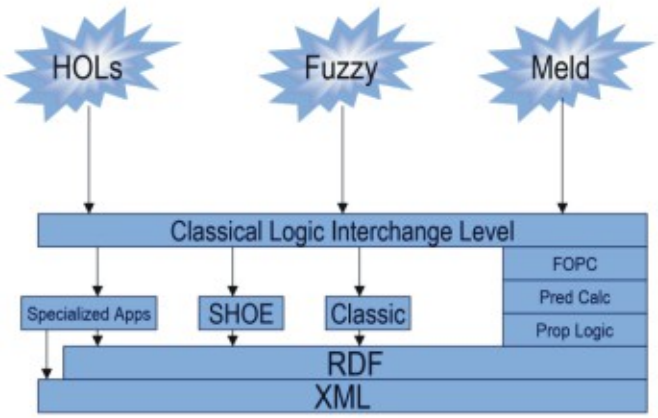
- **The Semantic Web "as an interchange bus for on-line data."**
- **RDF as a simple language for exchanging "raw data" among "heterogeneous systems."**
- **SWeLL (Semantic Web Logic Language) "extends RDF by including negation and explicit quantification."**
- **SWeLL should represent first-order and higher-order logic and pair "simple, predictable, reliable systems with complex, unpredictable, heuristic systems."**
- **But the tools delivered in 2005 were more limited.**

Goal: Implement Tim's vision with a new generation of tools.

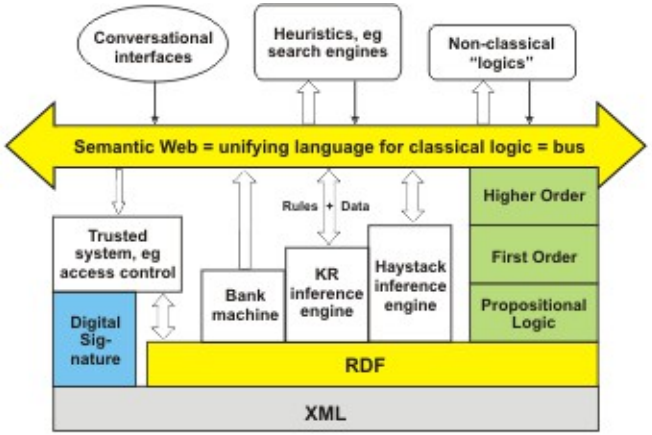
- **More advanced methods for language, learning, and reasoning.**
- **Better methods for building trust and ensuring security.**

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

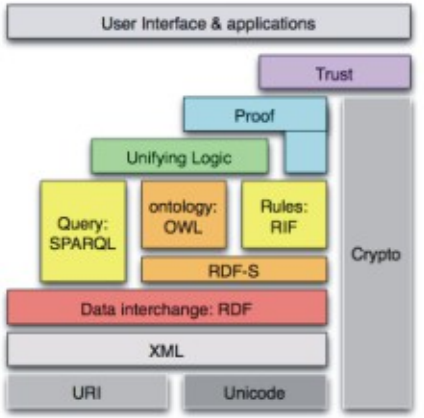
Semantic Web “Layer Cakes”



DAML Requirements (2000)



Winning Proposal (2000)



Final Report (2005)

The proposal was more ambitious than what was delivered.

- **New AI technology has been developed in the past 15 years.**
- **Advanced applications have gone far beyond the tools of 2005.**
- **The new technology should be more widely available.**
- **It should be easy to learn, easy to use, and upward compatible.**

In the diagram, the large yellow arrow is the SWeLL bus:

- **Semantic Web = unifying language for classical logic = bus.**

Two Examples of Advanced AI

Google's knowledge graphs (KGs) are represented in RDF.

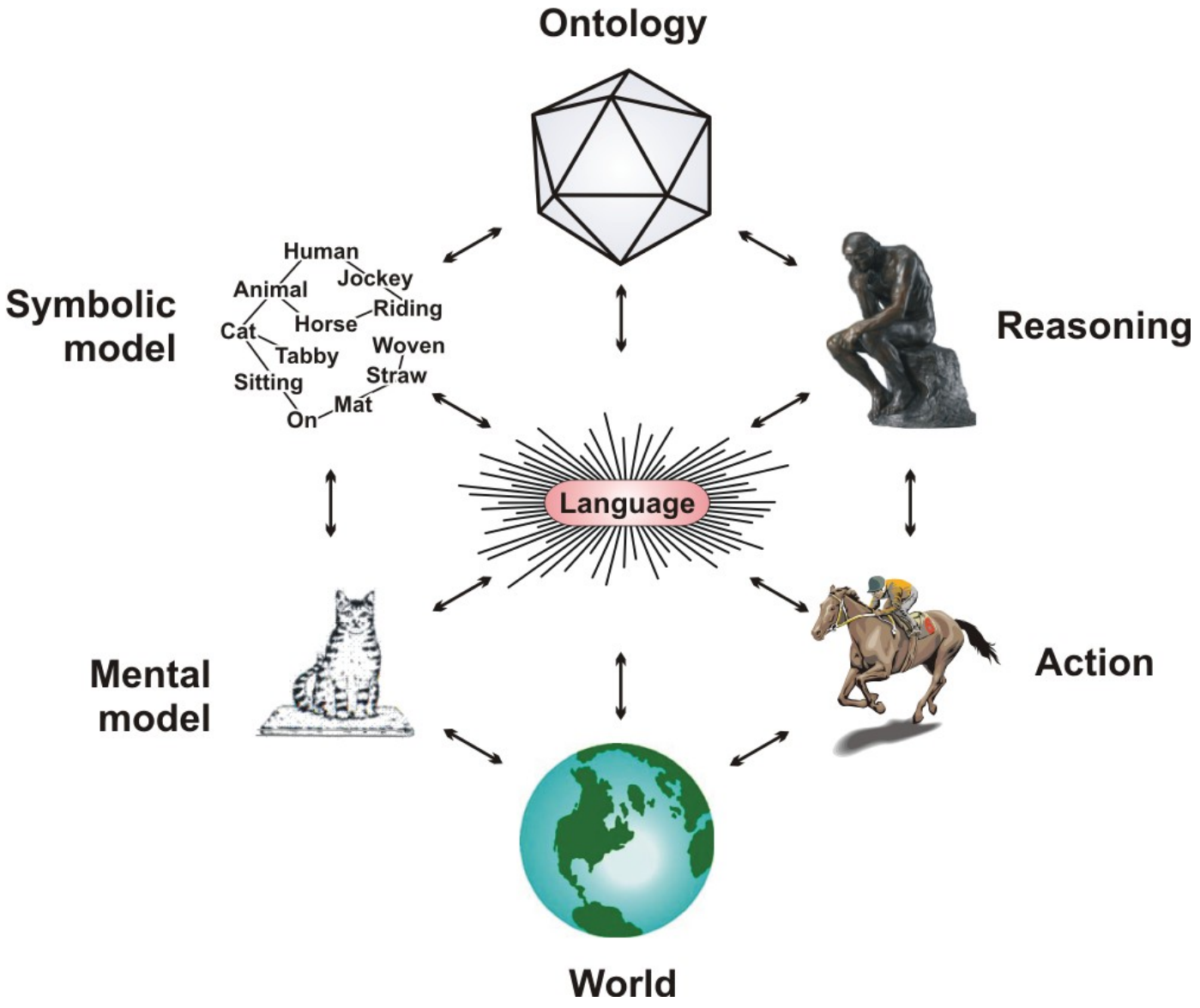
- **DBpedia and other freely available resources provide the data.**
- **Google added AI methods for learning and reasoning with KGs and deriving new KGs from documents.**

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.

Goals for the future: Automated and semi-automated tools to make such systems easier to design, cheaper to build, and more reliable.

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



RDF can represent symbolic models that are directly related to ontology, language, and mental models. Tools for logic and language can relate them to reasoning, action, and the world.

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 words and the kinds of things they refer to.**
5. **Reasoning includes all our ways of thinking about anything.**
6. **Action is what our thinking leads us to do in and on the world.**

Natural languages represent conscious knowledge.

- **They can represent and relate all six corners of the hexagon.**
- **Every artificial language, notation, or diagram is a simplified or stylized version of something that could be said in a natural language.**
- **But the nervous system contains an enormous amount of unconscious knowledge that supports the basic operations of the human body.**

Challenge: Implement AI tools to support all the above.

2. Interoperability

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

- **UML and the Semantic Web logics are supported by DOL.**
- **DOL tools can relate anything specified by those logics or an open-ended variety of others.**
- **That includes the notations for representing legacy software and the latest technologies of the 21st century.**

DOL is formally defined by logic and mathematics.

- **Logic is essential for guaranteeing precision.**
- **DOL can integrate heterogeneous OMS by relating the logics that specify them. Common Logic (CL) is one of the most general.**
- **But people may continue to use any notations they prefer.**

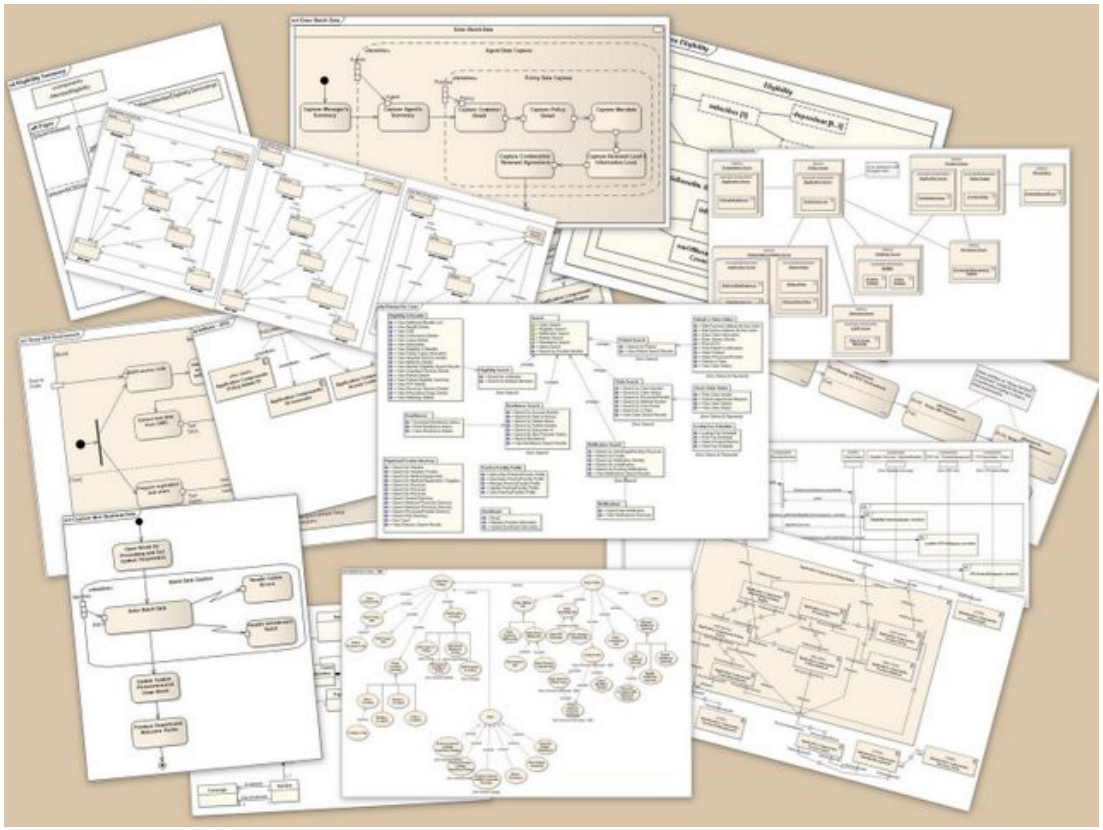
*** OMG Standard for DOL: Distributed Ontology, Modeling, and Specification Language: https://www.omg.org/spec/DOL/1.0**

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 in any field.

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

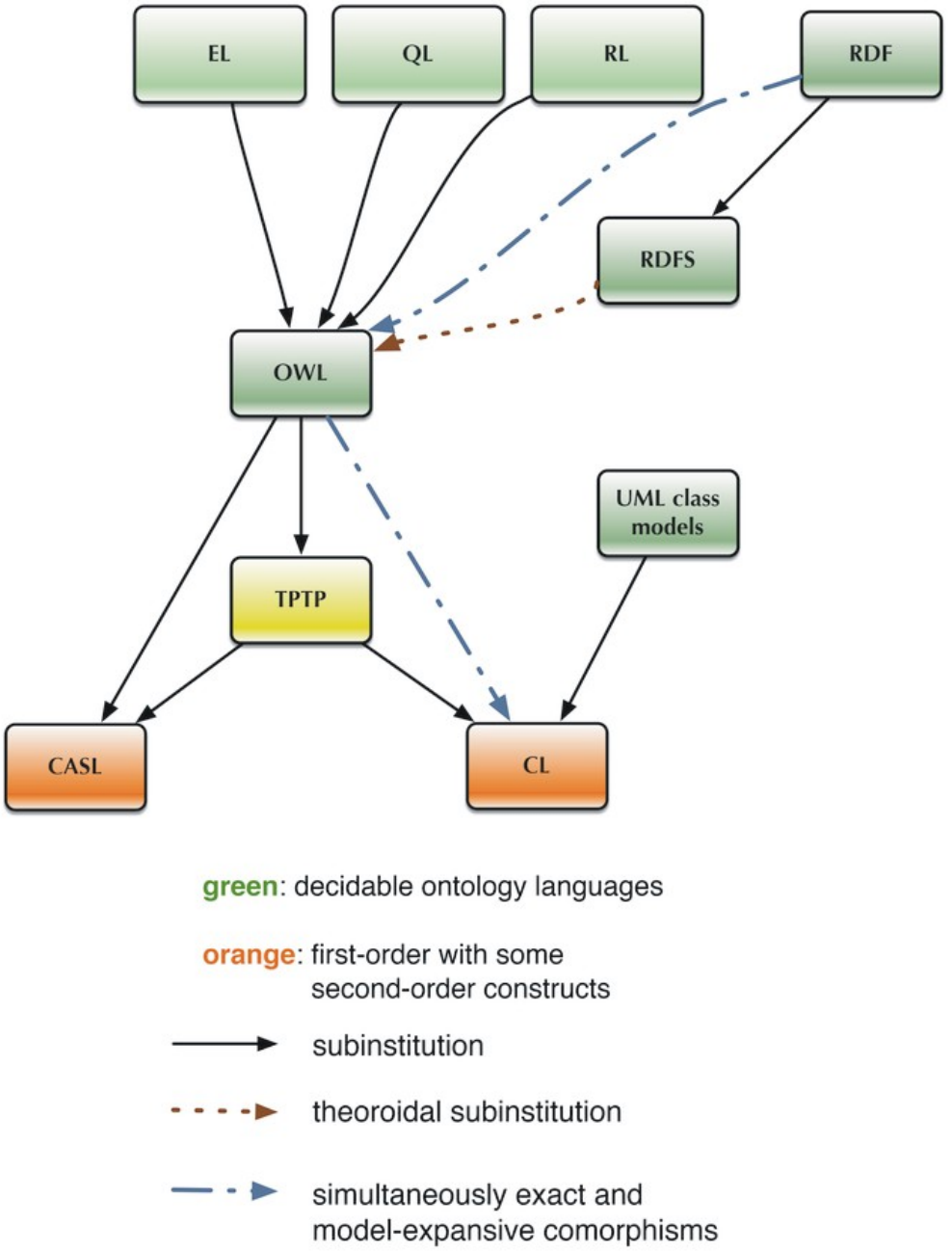
Mapping UML and the Semantic Web to CL

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 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.

3. Common Logic is SWeLL

A proposal for the Semantic Web Logic Language (SWeLL) evolved into the ISO/IEC standard for Common Logic (CL).*

CLIP (CL Interface to Predicate Calculus) is a linear dialect of Common Logic that has a simple mapping ↔ graph logics .

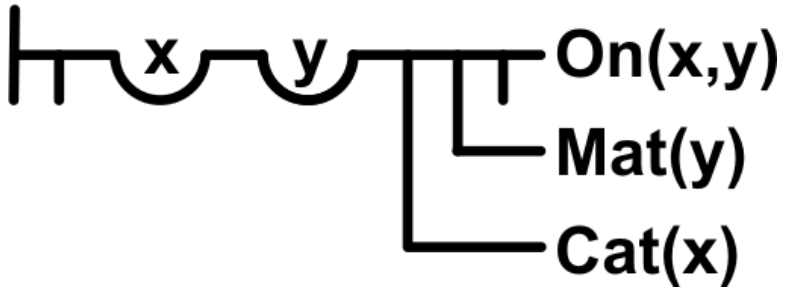
Design goals for CLIP:

- **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 a wide range of graph logics, including CGs, EGs, KGs, RDF, OWL, and UML diagrams.**
- **Query option: Select (*list of names*) where (*any CLIP sentence*).**
- **Support mappings of logics ↔ natural languages (NLs).**

*** The LBASE proposal: http://www.w3.org/TR/2003/NOTE-lbase-20031010**

How to say "A cat is on a mat."

Gottlob Frege (1879):



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

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

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)

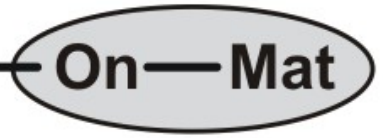
Existence: —

Negation: 

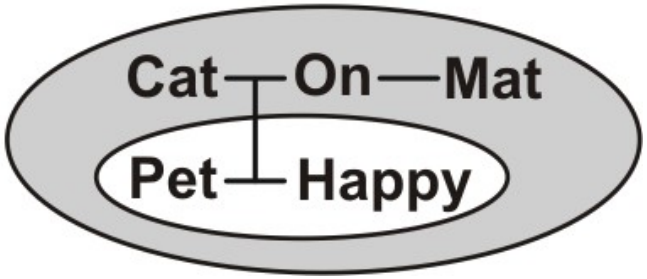
Relations: Cat• Mat• Happy• •On• •Under• •Give•

A cat is on a mat: Cat—On—Mat

Something is under a mat: —Under—Mat

Some cat is not on a mat: Cat—

Some cat is on something that is not a mat: Cat—On—

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)]].

Relating and Integrating Everything

CLIP can relate legacy systems to the latest AI tools.

- Freely mixing and matching any notations supported by DOL.
- Anyone may continue to use their favorite notations indefinitely.

Semantic Web annotations may be replaced by CLIP:

- Any URI, enclosed in quotes, is a valid CLIP name.
- An annotation that uses the full expressive power of CLIP is written <clip> (*one or more CLIP sentences*) </clip>
- Any annotation written in a Semantic Web logic x may be rewritten <clip logic=x> (*one or more CLIP sentences*) </clip>
- For tools that do not support CLIP, a preprocesor may translate CLIP annotations to the corresponding SW logic.

For integrating legacy systems with AI technology,

- Any software that is described or specified in any UML or SW notation can take advantage of tools that process CLIP.

4. 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 NLs to and from CLIP:

- **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.**

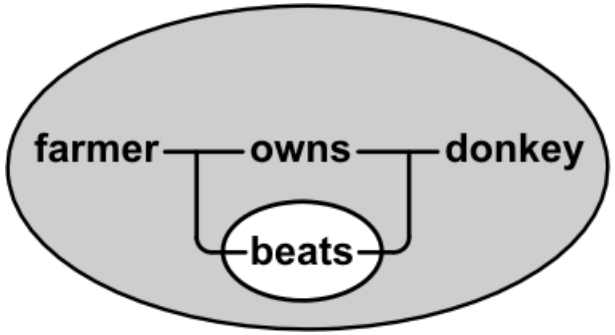
*** Hans Kamp & Uwe Reyle (1993) *From Discourse to Logic*, Dordrecht: Kluwer.**

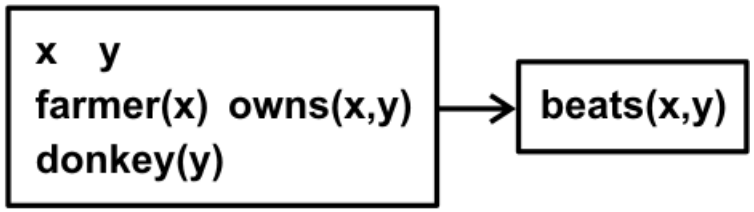
Mapping EG and DRS to CLIP

Peirce and Kamp independently chose equivalent structures.

- **Peirce chose ovals for EG with lines to show references.**
- **Kamp chose boxes for DRS with variables to show references.**
- **But the boxes and ovals represent the same logic in the same way.**

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





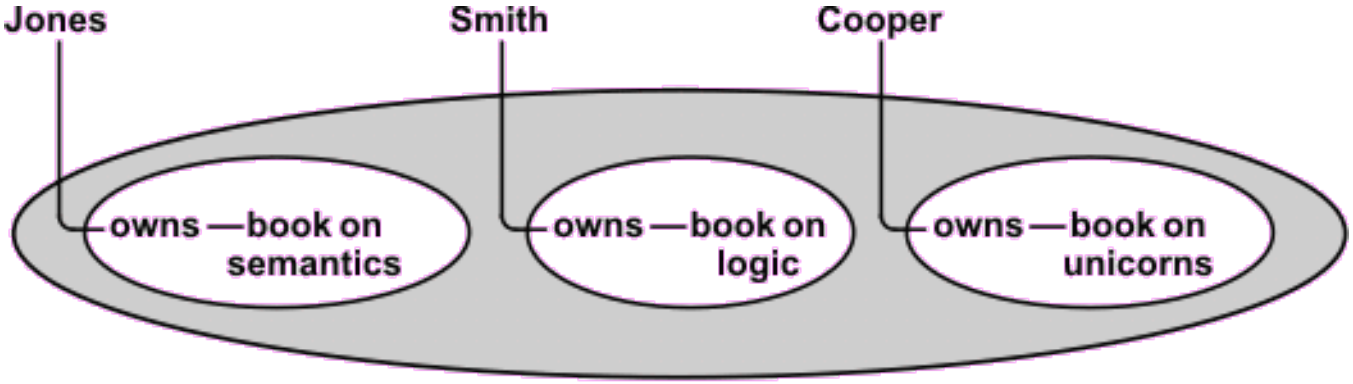
The EG and DRS may be translated to and from exactly the same CLIP: [If (∃ x y) (farmer x) (donkey y) (owns x y) [Then (beats x y)]].

Free software: Attempto Controlled English (ACE) translates a version of controlled English to DRS notation.

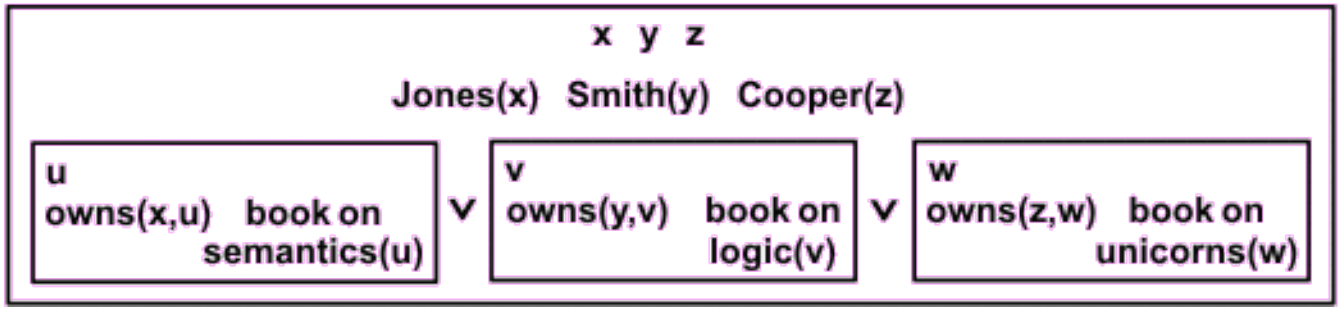
Disjunction in EG, DRS, and CLIP

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:



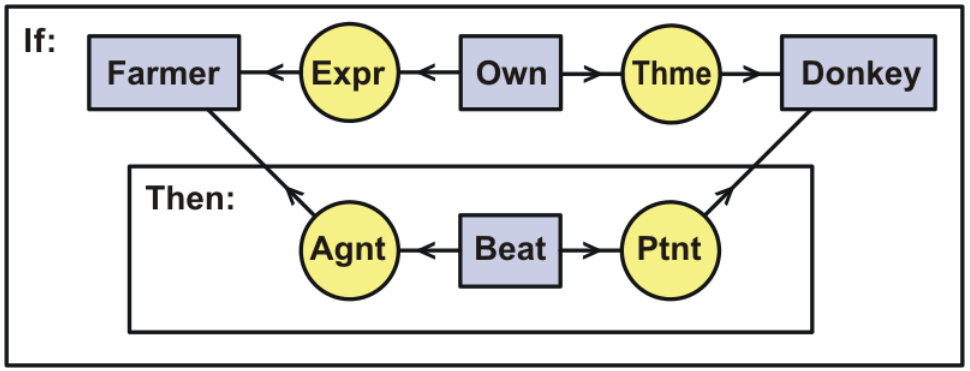
DRS:



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) ] ].
```

Conceptual Graphs



Conceptual graphs (CGs) express the same logic as EGs, but they are designed to represent 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.

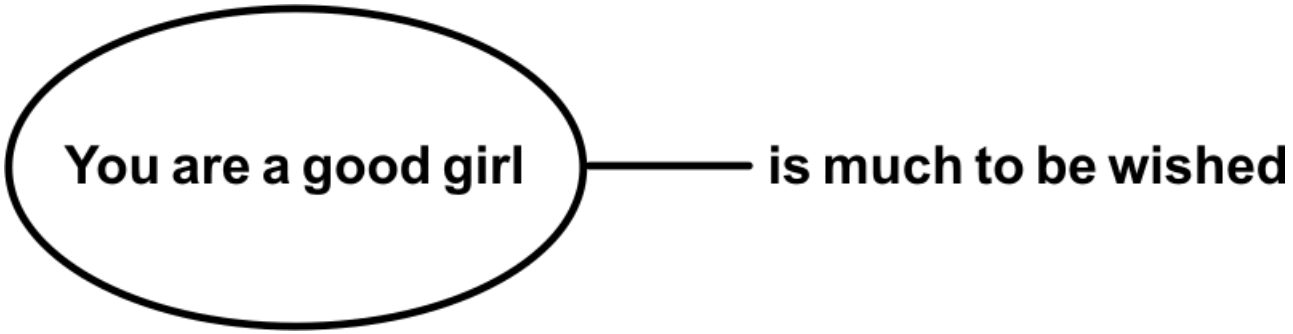
Names may refer to concept boxes that represent verbs.

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**

Metalanguage in Existential Graphs

A metalevel EG by Peirce (1898):



Peirce's English: "*That you are a good girl is much to be wished.*"

A shaded oval negates the nested EG. Without shading, the EG expresses a proposition that is neither asserted nor negated.

The CLIP representation for the EG:

("is much to be wished" ["You are a good girl"]).

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.***

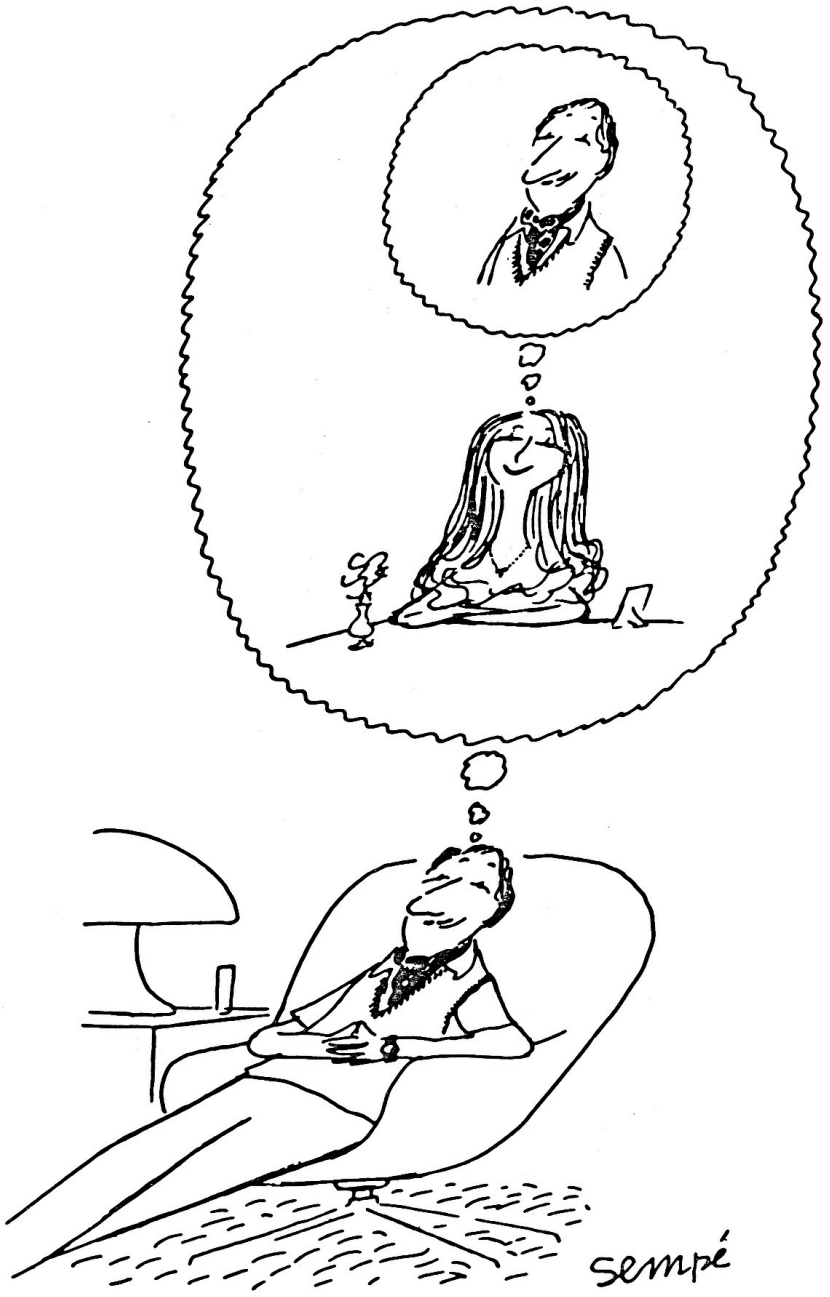
2. Modal: ***Pierre is thinking of Marie, who may be thinking of him.***

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.



Metalanguage in CLIP

Peirce's example of 1898 represents an intended situation.

English: ***That you are a good girl is much to be wished*** **[*by someone*].**

CLIP: ("is much to be wished" [Situation "You are a good girl"]).

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).

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

CLIP: (thinkingOf Pierre Marie) (possible [(thinkingOf Marie Pierre)]).

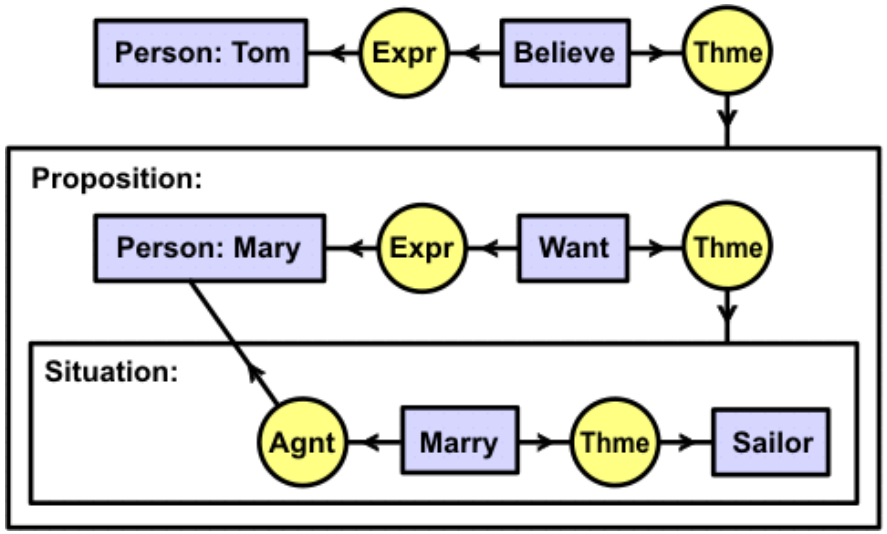
English: ***Pierre hopes that Marie is thinking of him*****.**

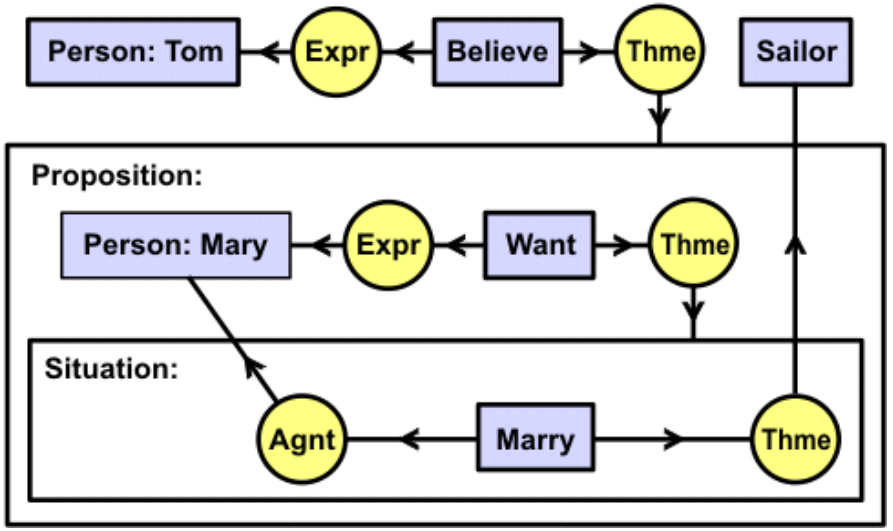
CLIP: (hopesFor Pierre [Situation (thinkingOf Marie Pierre)]).

By itself, IKL does not support modal logic. But IKL at the metalevel can be used to define modal relations in terms of laws and facts.*

*** See http://jfsowa.com/pubs/worlds.pdf**

CGs for Propositions and Situations





The CGs above show two of the three interpretations of the sentence *Tom believes that Mary wants to marry a sailor:*

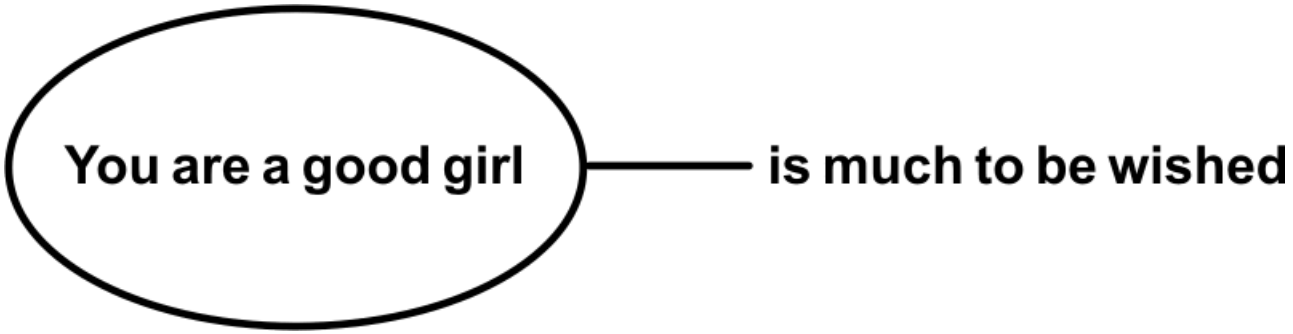
- ***Tom believes a proposition that Mary wants a situation in which there exists a sailor whom she marries.***
- ***There is a sailor, and Tom believes that Mary wants to marry him.***

For the third interpretation, the blue box for Sailor would be moved to the area of the proposition:

- ***Tom believes that there is a sailor whom Mary wants to marry.***

Metalanguage in Existential Graphs

A metalevel EG by Peirce (1898):



Peirce's English: "*That you are a good girl is much to be wished.*"

A shaded oval negates the nested EG. Without shading, the EG expresses a proposition that is neither asserted nor negated.

The CLIP representation for the EG:

("is much to be wished" ["You are a good girl"]).

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.***

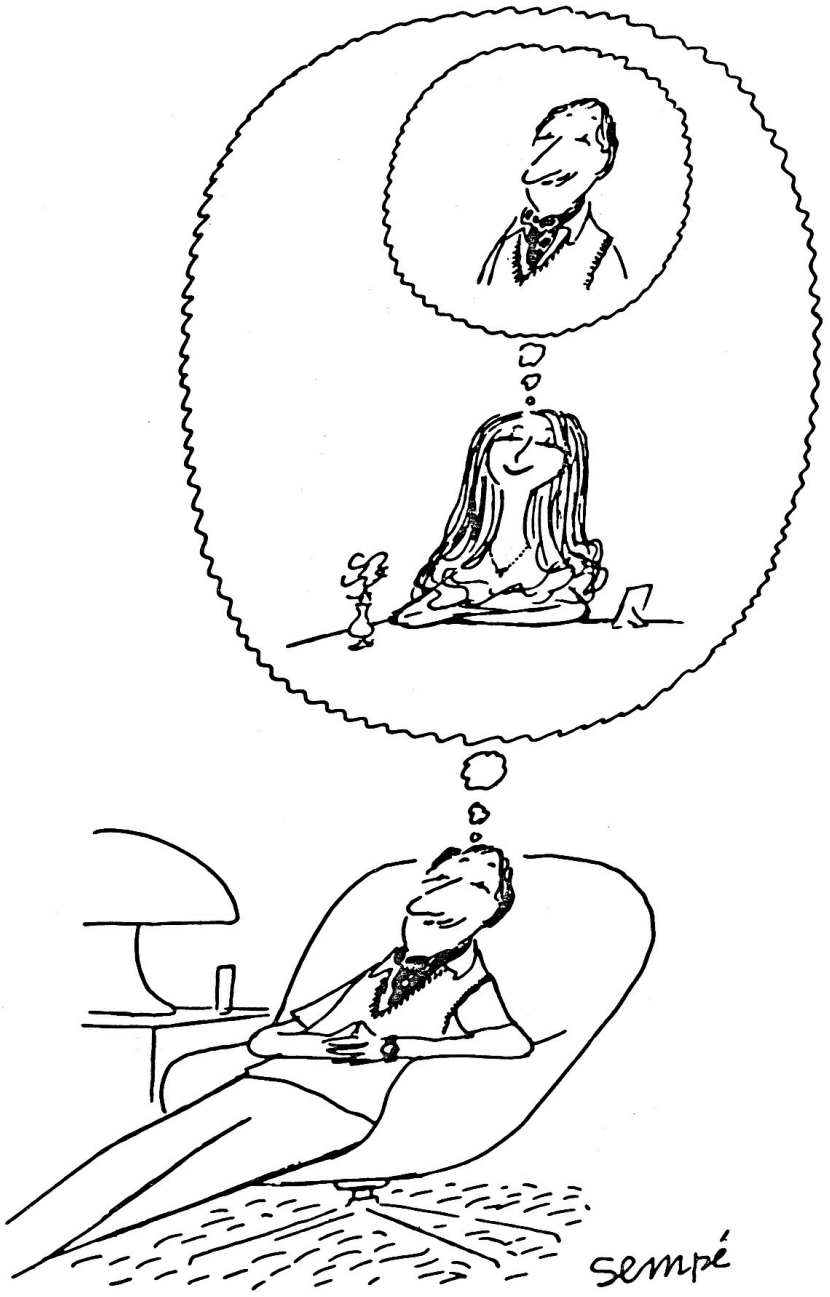
2. Modal: ***Pierre is thinking of Marie, who may be thinking of him.***

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.



Metalanguage in CLIP

Peirce's example of 1898 represents an intended situation.

English: ***That you are a good girl is much to be wished*** **[*by someone*].**

CLIP: ("is much to be wished" [Situation "You are a good girl"]).

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).

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

CLIP: (thinkingOf Pierre Marie) (possible [(thinkingOf Marie Pierre)]).

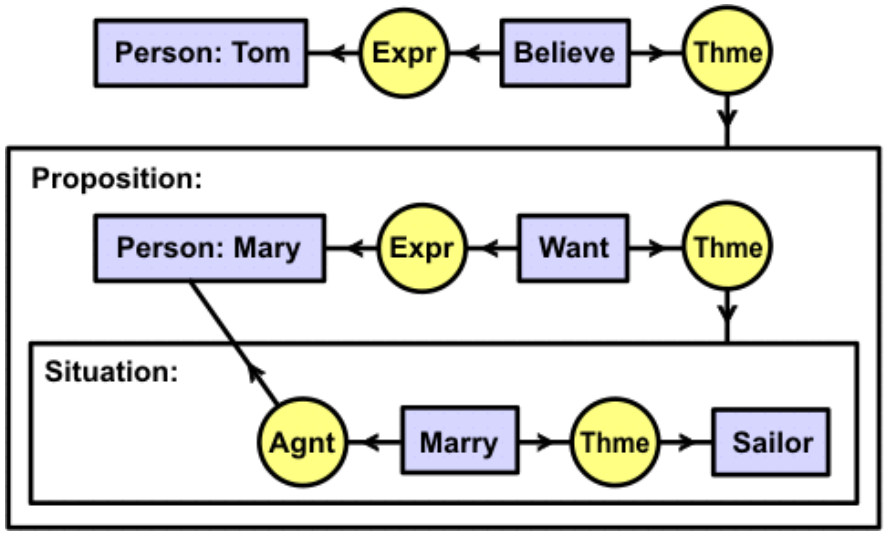
English: ***Pierre hopes that Marie is thinking of him*****.**

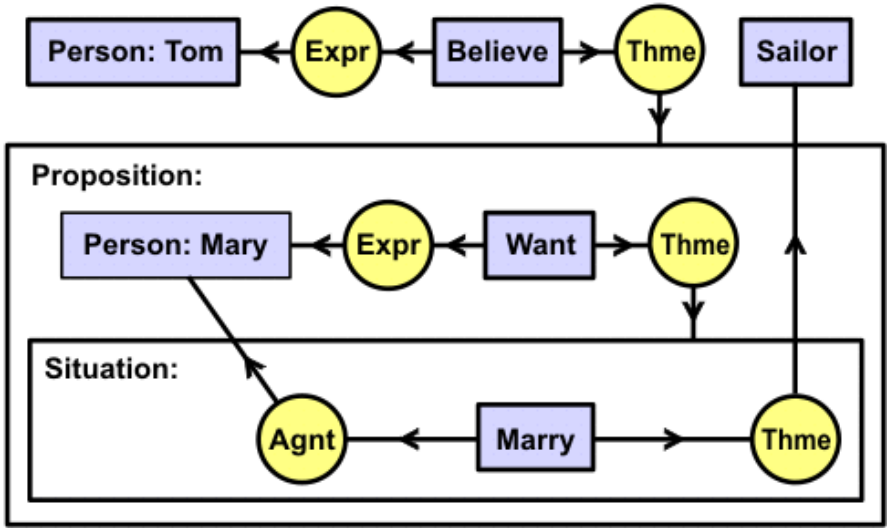
CLIP: (hopesFor Pierre [Situation (thinkingOf Marie Pierre)]).

By itself, IKL does not support modal logic. But IKL at the metalevel can be used to define modal relations in terms of laws and facts.*

*** See http://jfsowa.com/pubs/worlds.pdf**

CGs for Propositions and Situations





The CGs above show two of the three interpretations of the sentence *Tom believes that Mary wants to marry a sailor:*

- ***Tom believes a proposition that Mary wants a situation in which there exists a sailor whom she marries.***
- ***There is a sailor, and Tom believes that Mary wants to marry him.***

For the third interpretation, the blue box for Sailor would be moved to the area of the proposition:

- ***Tom believes that there is a sailor whom Mary wants to marry.***

5. Cognitive Memory (CM)

Cognitive Memory is key to VivoMind technology (2001 to 2014)

- Approximate pattern matching for analogies and metaphors.
- Associative storage and retrieval of graphs in log(N) time.
- Precise pattern matching for logic and mathematics.

Analogies can support informal, case-based reasoning:

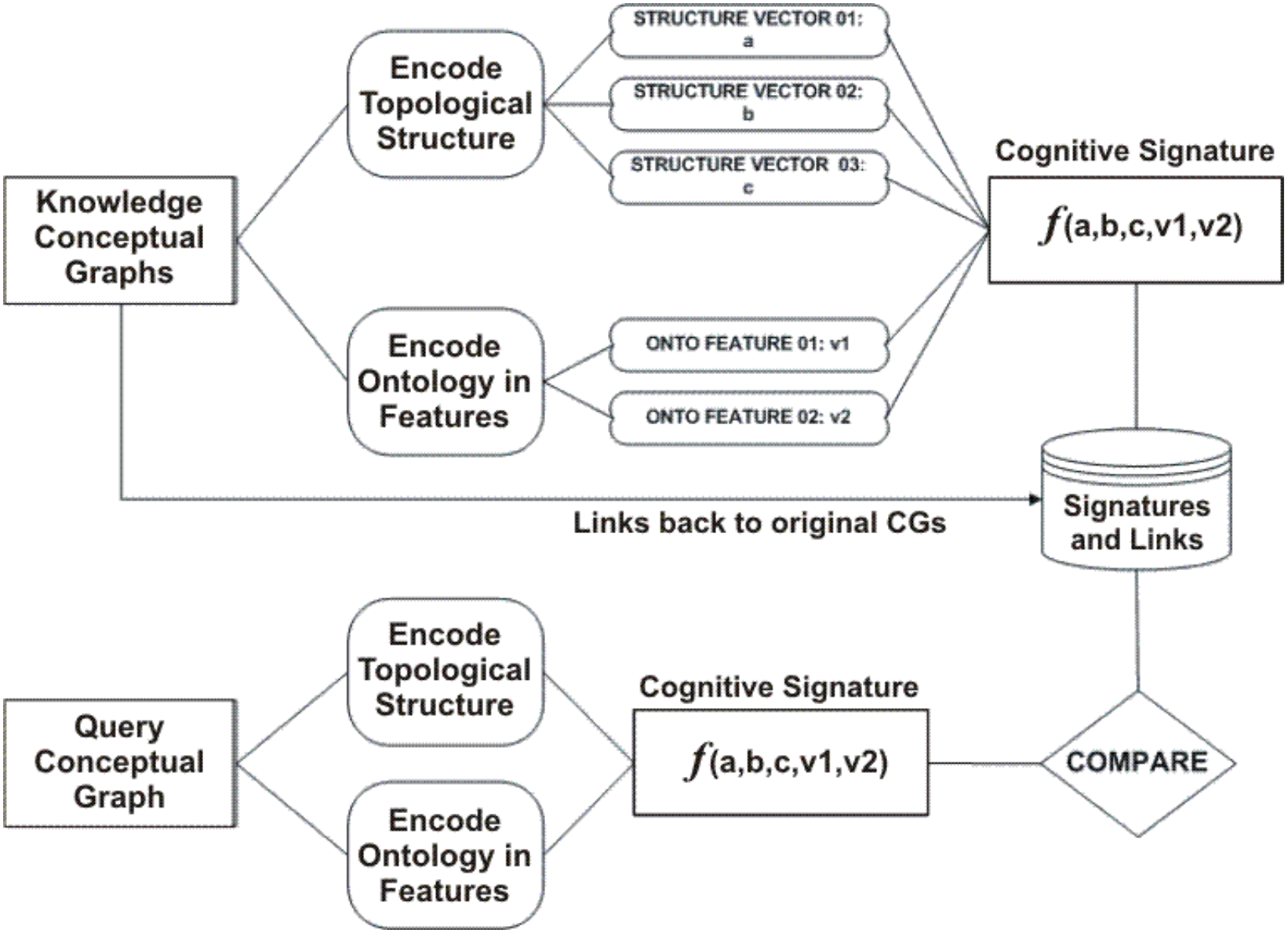
- CM can store large volumes of previous knowledge and experience.
- Any new case can be matched to similar cases in long-term memory.
- Close matches are ranked by a measure of semantic distance.

Formal reasoning is based on a disciplined use of analogy:

- Induction: Generalize multiple cases to create rules or axioms.
- Deduction: Match (unify) a new case with part of some rule or axiom.
- Abduction: Form a hypothesis based on aspects of similar cases.

Although 20 years old, CM is beyond the state of the art today.

Cognitive Memory

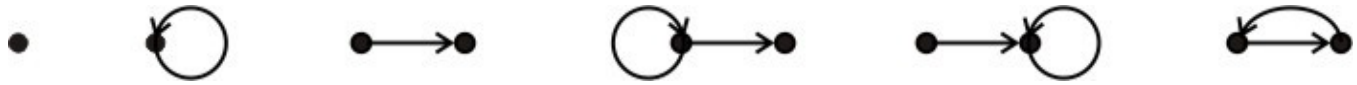


High-speed associative memory for graphs of any kind.

Structure Preserving

A cognitive signature™ encodes the exact structure of a graph.

- It is a lossless encoding, similar to a Gödel numbering.
- For unlabeled graphs, integers are sufficient for a cognitive signature.
- For example, 0 maps to and from an empty graph with no nodes or arcs.
- 1, 2, 3, 4, 5, and 6 can be mapped to and from the following graphs:



- To encode the structure of conceptual graphs in Cognitive Memory, the cognitive signatures are based on generalized combinatorial maps. **

By contrast, a word vector encodes labels, but not structure.

- A word vector is bsed on a “bag of words” that ignores the connectivity.
- Word vectors are useful for measuring the similarity of texts.
- But they discard the structural information necessary for reasoning, question answering, and language understanding.

From Perception to Cognition

Deep neural nets (DNNs) were designed for perceptual learning.

- **Require huge amounts of data to learn patterns.**
- **Learn complex patterns as well or better than humans.**
- **But they can't explain or reason about the patterns they learn.**

Traditional AI tools were designed for cognition:

- **Parsers and translators for natural and artificial languages.**
- **Grammars, lexicons, ontologies, terminologies, corpora, Wikipedia, DBpedia, Linked Open Data, and the Semantic Web.**
- **Inference engines and rule-based systems for formal logics and many kinds of informal or fuzzy reasoning.**

Cognitive Memory does learning and reasoning for cognition.

- **Analogies for induction, abduction, and case-based reasoning.**
- **Learn cognitive structures from small amounts of data — even "one-shot learning" from a single example.**
- **Can be used with NNs in hybrid systems.**

Exact and Approximate Matching

For logic, CM can find an exact match that unifies two graphs:

- **For example, match a graph from English to a graph from SQL.**
- **These matches are essential for rule-based inference engines.**
- **They are also important for comparing programming statements.**

But CM also finds approximate matches for analogies:

- **Given a graph g and a small semantic distance ε, CM can find all graphs within the distance ε from g — in log(N) time.**
- **This option is essential for natural languages, which rarely have exact matches, even for sentences with the "same" meaning.**

Information Extraction Project

The next slide shows a table derived from research reports.

To extend the semantics, an ontology for chemistry was added to the basic VivoMind ontology.

Then for each report,

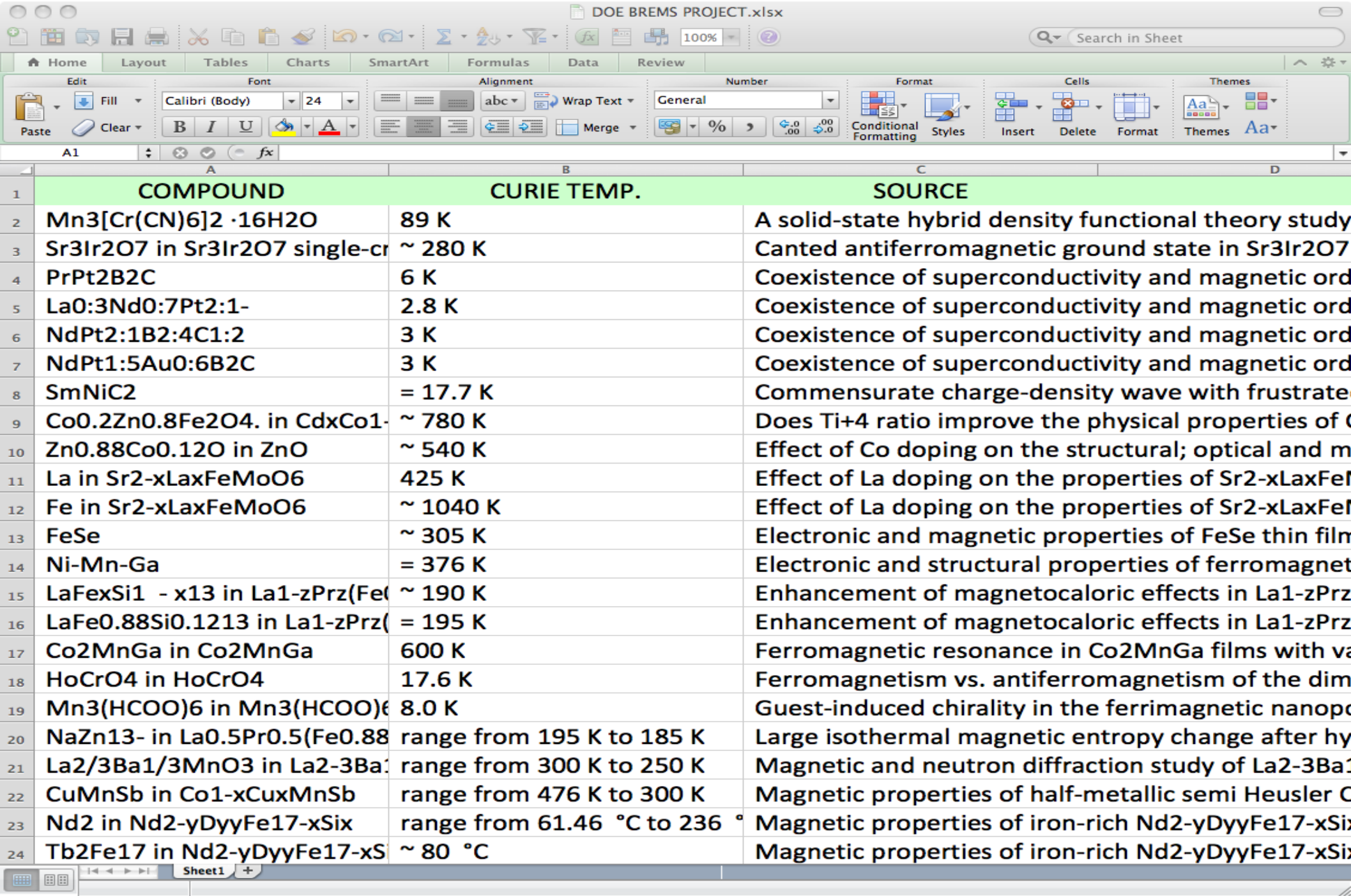
- **Map each sentence to a conceptual graph (CG). ***
- **Analyze anaphoric references to link pronouns to named entities.**
- **The result is a large CG that represents every sentence in the document.**
- **Store that graph (including subgraphs) in Cognitive Memory.**
- **Query Cognitive Memory for the data in each row of the table.**
- **Store the answers in the table.**

In a competition among twelve NLP systems,

- **The VivoMind system got 96% of the entries correct.**
- **The second best score was 73%. Most scores were below 50%.**

*** For an overview of CG methods, see http://www.jfsowa.com/pubs/template.pdf**

Information Extracted from Documents



	A	B	C	D
1	COMPOUND	CURIE TEMP.	SOURCE	
2	Mn3[Cr(CN)6]2 ·16H2O	89 K	A solid-state hybrid density functional theory study	
3	Sr3Ir2O7 in Sr3Ir2O7 single-cr	~ 280 K	Canted antiferromagnetic ground state in Sr3Ir2O7	
4	PrPt2B2C	6 K	Coexistence of superconductivity and magnetic ord	
5	La0:3Nd0:7Pt2:1-	2.8 K	Coexistence of superconductivity and magnetic ord	
6	NdPt2:1B2:4C1:2	3 K	Coexistence of superconductivity and magnetic ord	
7	NdPt1:5Au0:6B2C	3 K	Coexistence of superconductivity and magnetic ord	
8	SmNiC2	= 17.7 K	Commensurate charge-density wave with frustrate	
9	Co0.2Zn0.8Fe2O4. in CdxCo1-	~ 780 K	Does Ti+4 ratio improve the physical properties of	
10	Zn0.88Co0.12O in ZnO	~ 540 K	Effect of Co doping on the structural; optical and m	
11	La in Sr2-xLaxFeMoO6	425 K	Effect of La doping on the properties of Sr2-xLaxFe	
12	Fe in Sr2-xLaxFeMoO6	~ 1040 K	Effect of La doping on the properties of Sr2-xLaxFe	
13	FeSe	~ 305 K	Electronic and magnetic properties of FeSe thin film	
14	Ni-Mn-Ga	= 376 K	Electronic and structural properties of ferromagnet	
15	LaFexSi1 - x13 in La1-zPrz(Fe	~ 190 K	Enhancement of magnetocaloric effects in La1-zPrz	
16	LaFe0.88Si0.1213 in La1-zPrz(	= 195 K	Enhancement of magnetocaloric effects in La1-zPrz	
17	Co2MnGa in Co2MnGa	600 K	Ferromagnetic resonance in Co2MnGa films with va	
18	HoCrO4 in HoCrO4	17.6 K	Ferromagnetism vs. antiferromagnetism of the dim	
19	Mn3(HCOO)6 in Mn3(HCOO)6	8.0 K	Guest-induced chirality in the ferrimagnetic nanopo	
20	NaZn13- in La0.5Pr0.5(Fe0.88	range from 195 K to 185 K	Large isothermal magnetic entropy change after hy	
21	La2/3Ba1/3MnO3 in La2-3Ba1	range from 300 K to 250 K	Magnetic and neutron diffraction study of La2-3Ba1	
22	CuMnSb in Co1-xCuxMnSb	range from 476 K to 300 K	Magnetic properties of half-metallic semi Heusler C	
23	Nd2 in Nd2-yDyyFe17-xSix	range from 61.46 °C to 236 °	Magnetic properties of iron-rich Nd2-yDyyFe17-xSix	
24	Tb2Fe17 in Nd2-yDyyFe17-xSi	~ 80 °C	Magnetic properties of iron-rich Nd2-yDyyFe17-xSix	

Application to Legacy Re-engineering

Analyze the software and documentation of a corporation.

Programs in daily use, some of which were up to 40 years old.

- **1.5 million lines of COBOL programs.**
- **Several hundred scripts in JCL (IBM Job Control Language).**
- **100 megabytes of English documentation — reports, manuals, e-mails, Lotus Notes, HTML, and program comments.**

Goal:

- **Analyze the COBOL programs.**
- **Analyze the English documentation.**
- **Compare the two to determine:**
  - **Data dictionary of all data used by all programs.**
  - **English glossary of all terms with index to the software.**
  - **Evolution of terminology over the years.**
  - **Structure diagrams of the programs, files, and data.**
  - **Discrepancies between programs and documentation.**

An Important Simplification

An extremely difficult and still unsolved problem:

- **Translate English specifications to executable programs.**

Much easier task:

- **Translate the COBOL programs to conceptual graphs (CGs).**
- **Those CGs provide the ontology and background knowledge.**
- **The CGs derived from English may have ambiguous options.**
- **In parsing English, use CGs from COBOL to resolve ambiguities.**
- **The COBOL CGs show the most likely options.**
- **They can also provide missing information or detect errors.**

The CGs derived from COBOL provide a formal semantics for the informal English texts.

Excerpt from the Documentation

The input file that is used to create this piece of the Billing Interface for the General Ledger is an extract from the 61 byte file that is created by the COBOL program BILLCRUA in the Billing History production run. This file is used instead of the history file for time efficiency. This file contains the billing transaction codes (types of records) that are to be interfaced to General Ledger for the given month.

For this process the following transaction codes are used: 32 — loss on unbilled, 72 — gain on uncollected, and 85 — loss on uncollected. Any of these records that are actually taxes are bypassed. Only client types 01 — Mar, 05 — Internal Non/Billable, 06 — Internal Billable, and 08 — BAS are selected. This is determined by a GETBDATA call to the client file.

Note that none of the files or COBOL variables are named.

By matching graphs derived from English to graphs derived from COBOL, all names of files and COBOL variables were determined.

Interpreting Novel Patterns

Many documents contain unusual or ungrammatical patterns.

They may be elliptical forms that could be stored in tables.

But some authors wrote them as phrases:

- ***32 — loss on unbilled***
- ***72 — gain on uncollected***
- ***85 — loss on uncollected***

The dashes were represented by a default relation (Link):

[Number: 32]→(Link)→[Punctuation: "–"]→(Link)→[Loss]→(On)→[Unbilled]

This CG, which was derived from an English document, matched CGs derived from COBOL programs:

- **The value 32 was stored as a constant in a COBOL program.**
- **The phrase "loss on unbilled" was in a comment that followed the value 32 in that program.**

Results

Job finished in 8 weeks by Arun Majumdar and André LeClerc.

- **Four weeks for customization:**

  Design, ontology, and additional programming for I/O formats.

- **Three weeks to adapt the software that used Cognitive Memory:**

  24 hours a day on a 750 MHz Pentium III.

  Matches with strong evidence (close semantic distance) were correct.

  Weak matches were confirmed or corrected by Majumdar and LeClerc.

- **One week to produce a CD-ROM with the desired results:**

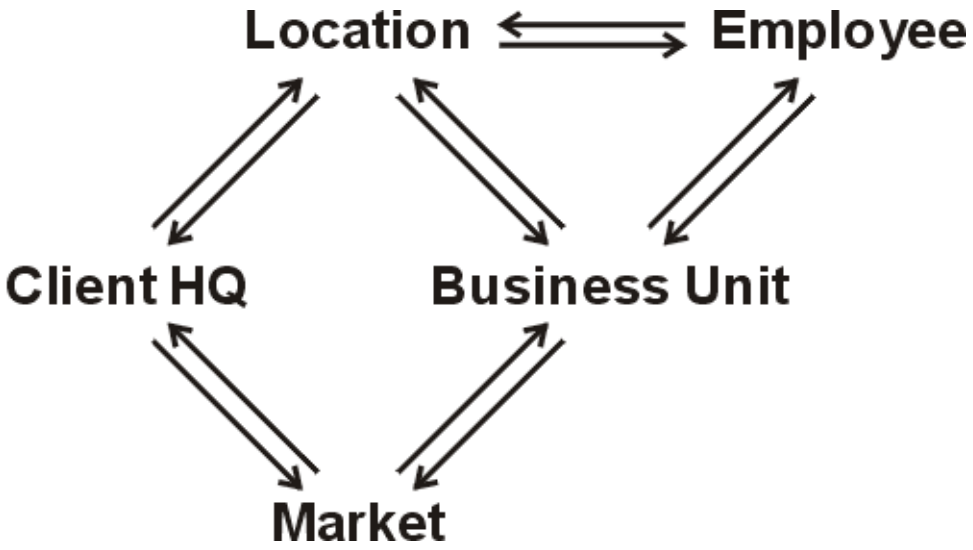
  Glossary, data dictionary, data flow diagrams, process architecture diagrams, system context diagrams, and list of errors detected.

A major consulting firm estimated that the job would take 40 people two years to analyze the documentation and find all cross references.

With Cognitive Memory, the task was completed in 15 person weeks.

Discrepancy Detected

A diagram of relationships among data types in the database:



Question: Which location determines the market?

- **According to the documentation: Business unit.**
- **According to the COBOL programs: Client HQ.**

For many years, management had been making decisions based on incorrect assumptions.

Contradiction Detected

From the ontology used for interpreting English:

- **Every employee is a human being.**
- **No human being is a computer.**

From analyzing COBOL programs:

- **Some employees are computers.**

What is the reason for this contradiction?

Quick Patch in 1979

A COBOL programmer made a quick patch:

- **Two computers were used to assist human consultants.**
- **But there was no provision to bill for computer time.**
- **Therefore, the programmer named the computers Bob and Sally, and assigned them employee ids.**

For more than 20 years:

- **Bob and Sally were issued payroll checks.**
- **But they never cashed them.**

The software discovered two computer "employees."

Relating Formal and Informal CGs

The legacy-reengineering task required two kinds of processing.

Precise reasoning:

- **Analyzing the COBOL programs and translating them to CGs.**
- **Detecting discrepancies between different programs.**
- **Detecting discrepancies between programs and documentation.**

Indexing and cross references:

- **Creating an index of English terms and names of programs.**
- **Mapping English documents to the files and programs they mention.**

Conceptual graphs derived from COBOL are precise.

- **But CGs derived from English are informal and unreliable.**
- **Informal CGs are adequate for cross-references between the English documents and the COBOL programs.**
- **All precise reasoning was performed on CGs from COBOL or on CGs from English that were *corrected* by CGs from COBOL.**

Answering Questions

For the sources, either NL documents or structured data:

- **Translate the text or data to conceptual graphs.**
- **Translate all CGs to Cognitive Signatures™ in time proportional to (N log N), where N is the total number of CGs.**
- **Store each Cognitive Signature in Cognitive Memory with a pointer to the original CG and the source from which that CG was derived.**
- **Use previously translated CGs to help interpret new sentences.**

For a query stated as an English sentence or paragraph,

- **Translate the query to conceptual graphs.**
- **Find matching patterns in the source data and rank them in order of semantic distance. The time is proportional to (log N).**
- **For each match within a given threshold, use structure mapping to verify which parts of the query CG match the source CG.**
- **As answer, return the English sentences or paragraphs in the source document that had the closest match to the query.**

6. Automated and Semi-automated Tools

The tools should support a dialogue.

- **Explanation requires more interaction than question-answering.**
- **Both novices and experts should be able to carry on an open-ended conversation about any subject they choose.**
- **Follow-up questions may drill down to any depth required.**
- **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 they may need human assistance.**

A dialogue should be as precise or vague as the subject matter.

- **Human languages can describe a continuous, dynamically changing world at any level of detail and precision.**
- **A dialogue with computers should be just as flexible.**
- **Both the human and the computer should be able to ask questions.**

Formal Concept Analysis (FCA)

A theory and tools for semi-automated ontology design:

- Theory. Define a minimal lattice that shows all inheritance paths among a set of concept types, each defined by a list of attributes.
- Algorithms. Efficient ways for computing a minimal lattice from a list of terms and defining features.

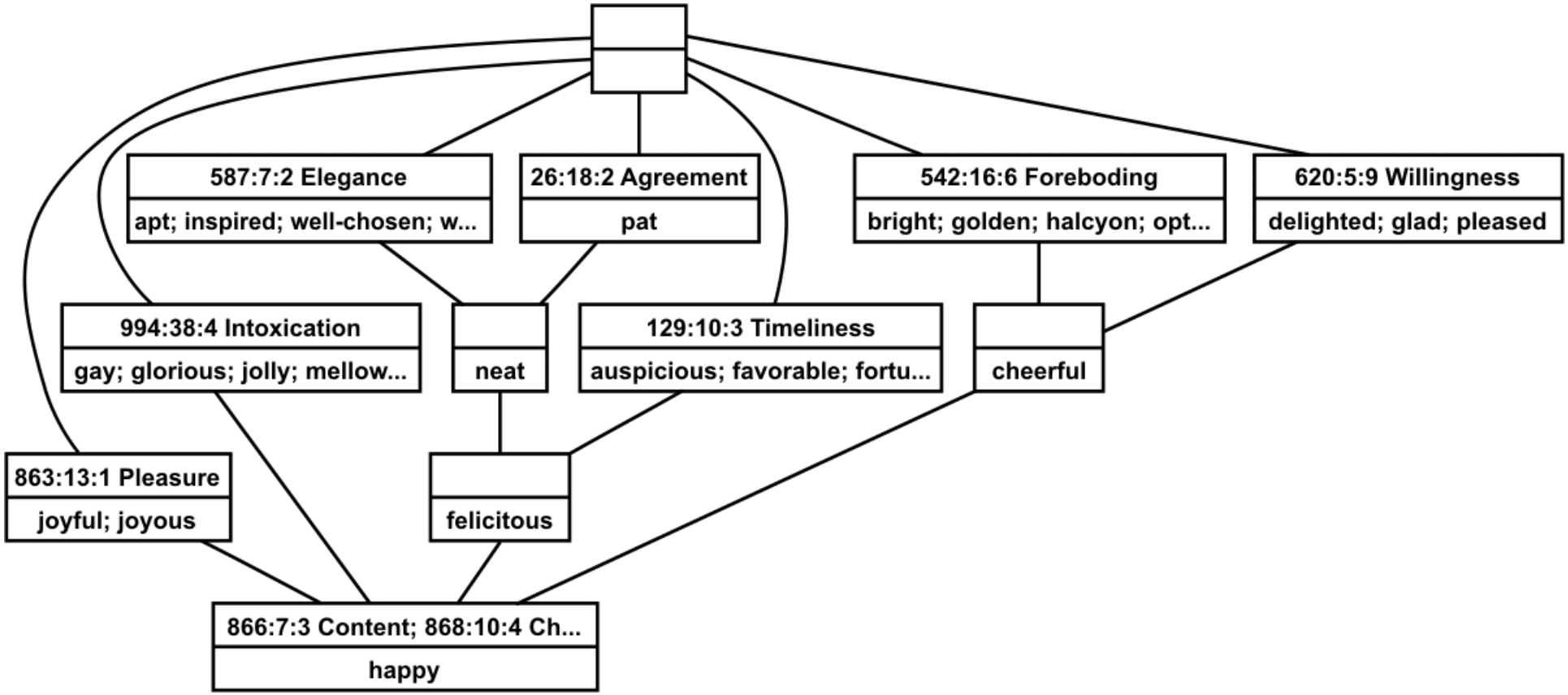
Applications:

- Ontology development and alignment; classification methods; machine learning; defining concepts used in other logics.
- FCA tools are often used to check whether ontologies specified in OWL and other notations are consistent.
- They can also be used to detect inconsistencies among two or more independently developed ontologies.

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

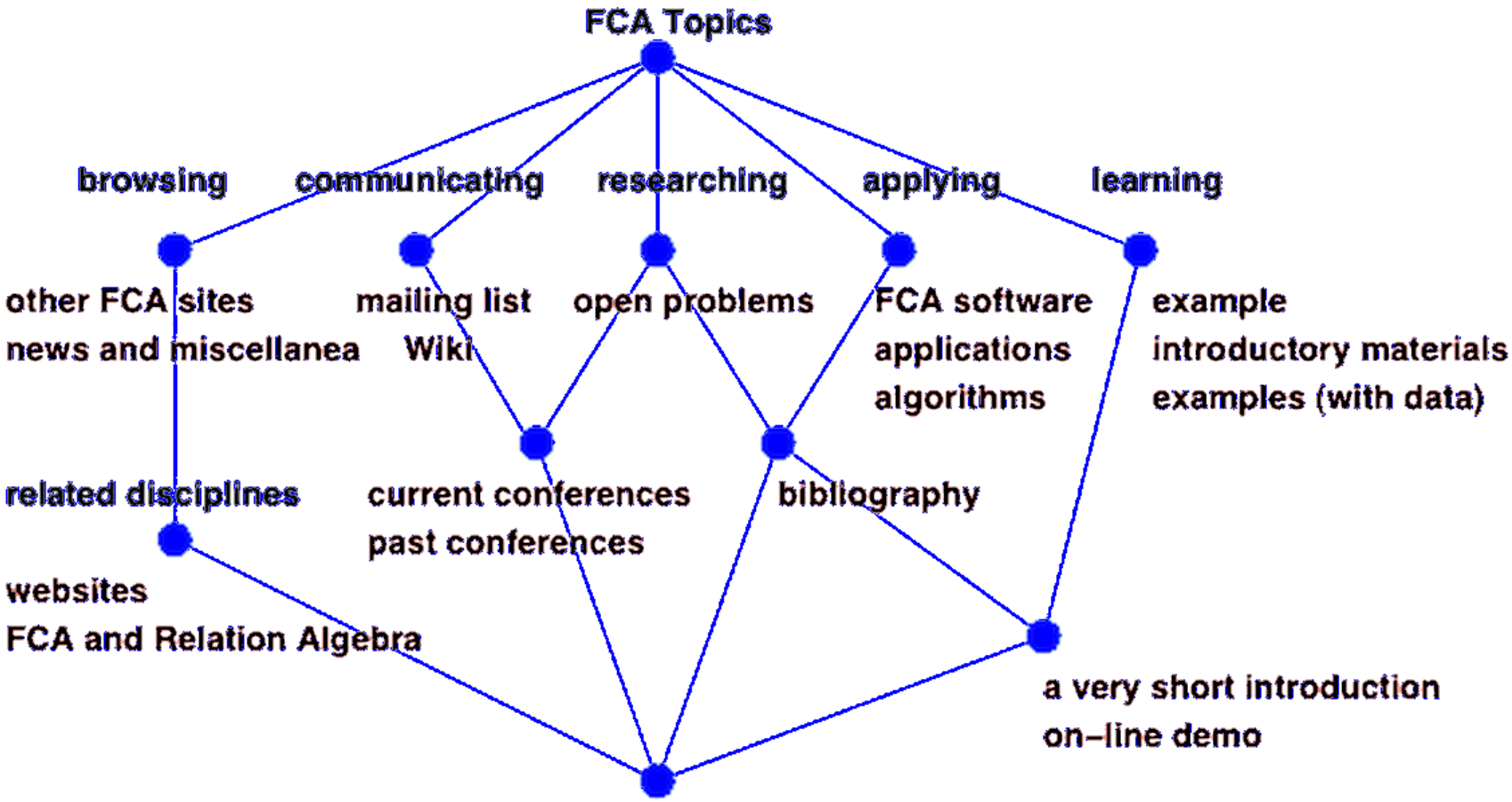
Generating Lattices Automatically



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

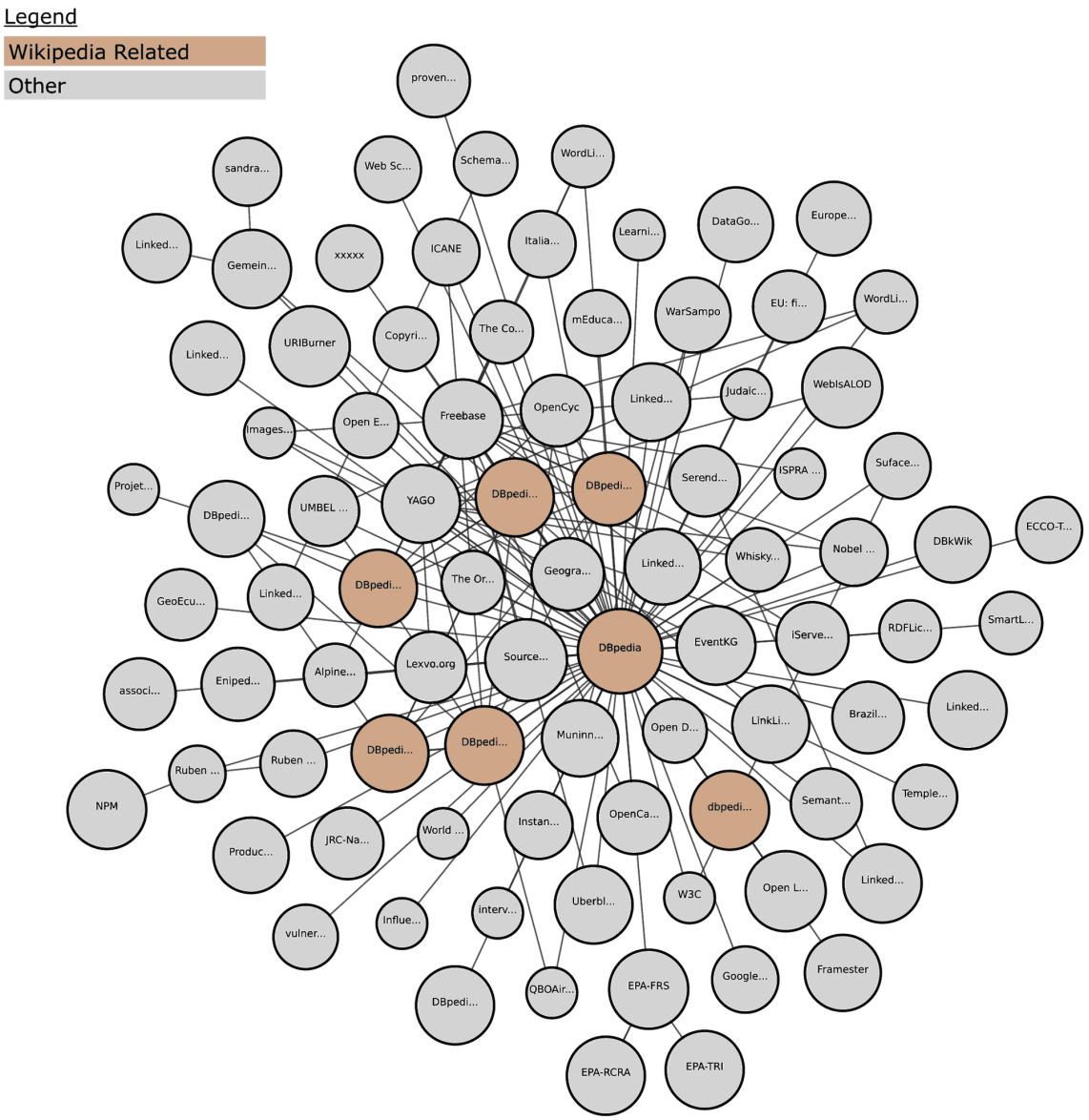
To generate this or similar lattices, enter 'happy' or any other word at the web site http://www.ketlab.org.uk/roget.html

Classifying Resources by Purpose



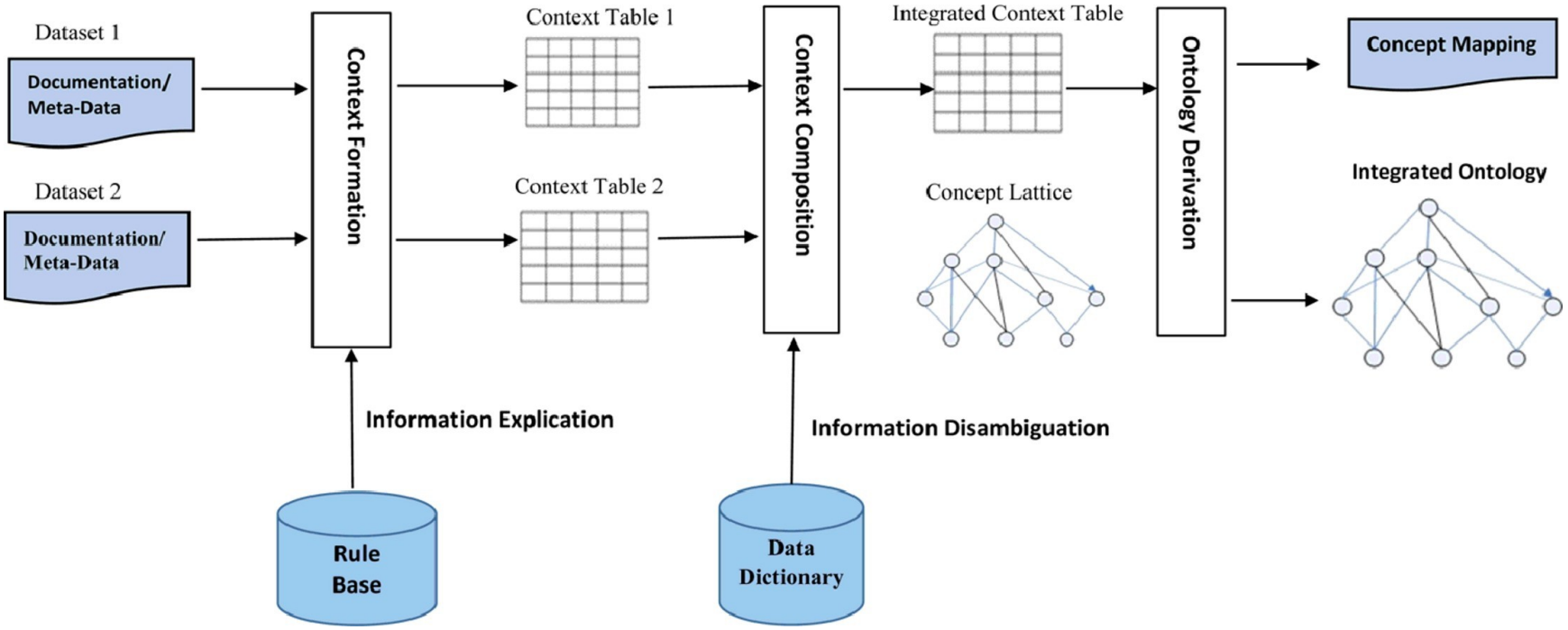
FCA tools may use a variety of criteria for classification.

- **For ontology, the usual criterion is type/subtype.**
- **But a person who asks a question has some purpose in mind.**
- **The lattice above classifies resources by purpose, not type.**



Challenge: Relate diverse ontologies across domains.

Using FCA to Merge Ontologies



Semi-automated method for integrating diverse ontologies.*

- **Independently developed systems are usually incompatible.**
- **FCA tools can detect similarities and conflicts in definitions.**
- **With some human assistance, the tools can derive a merged ontology that can support data sharing among the systems.**

*** Gaihua Fu, FCA based ontology development for data integration, https://www.sciencedirect.com/science/article/pii/S030645731630019X**

Federated Ontologies

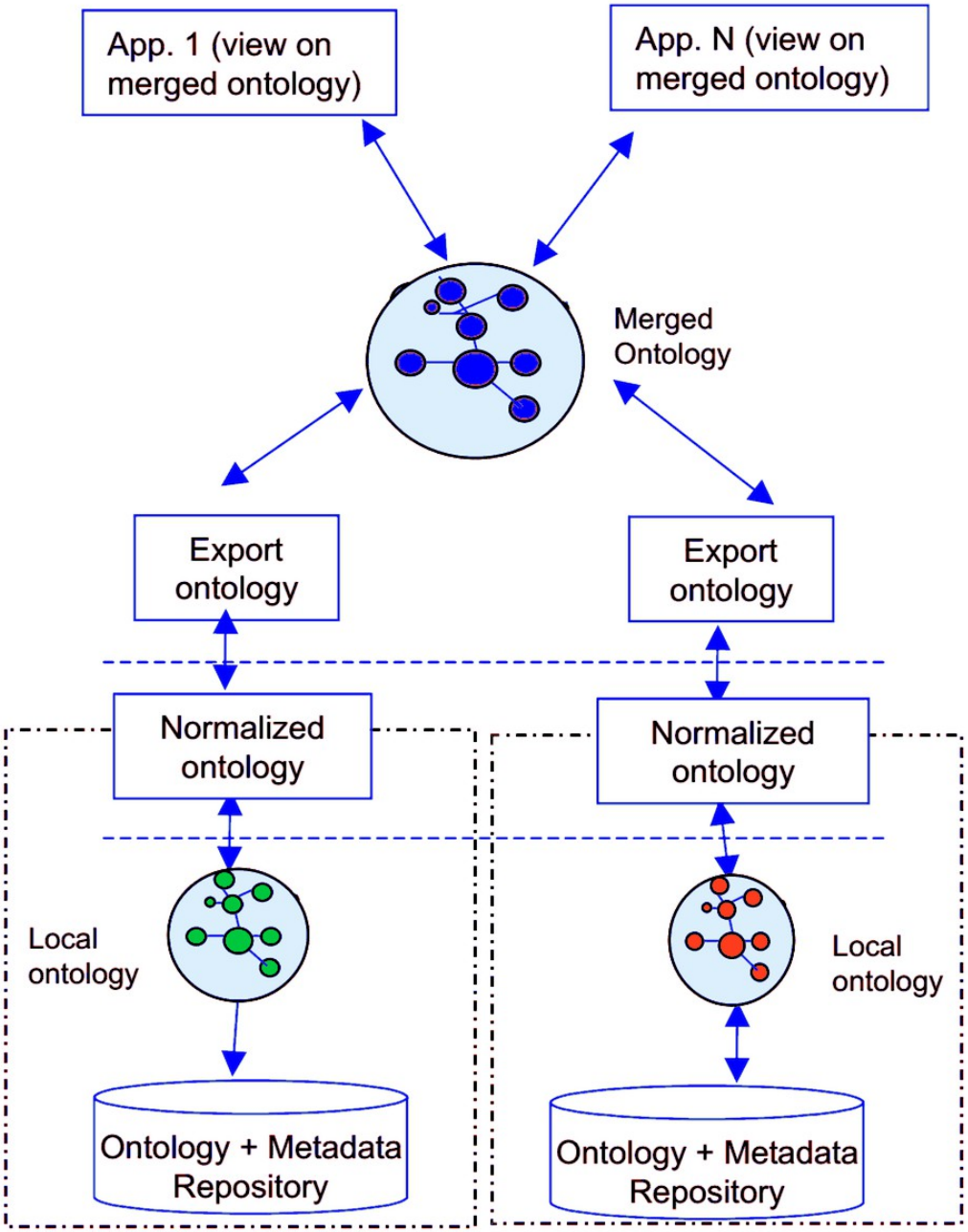
A merger of multiple ontologies may become large and unwieldy.

A federation of ontologies for a large domain would require an underspecified top level. The subdomains would add detail as needed for various applications.

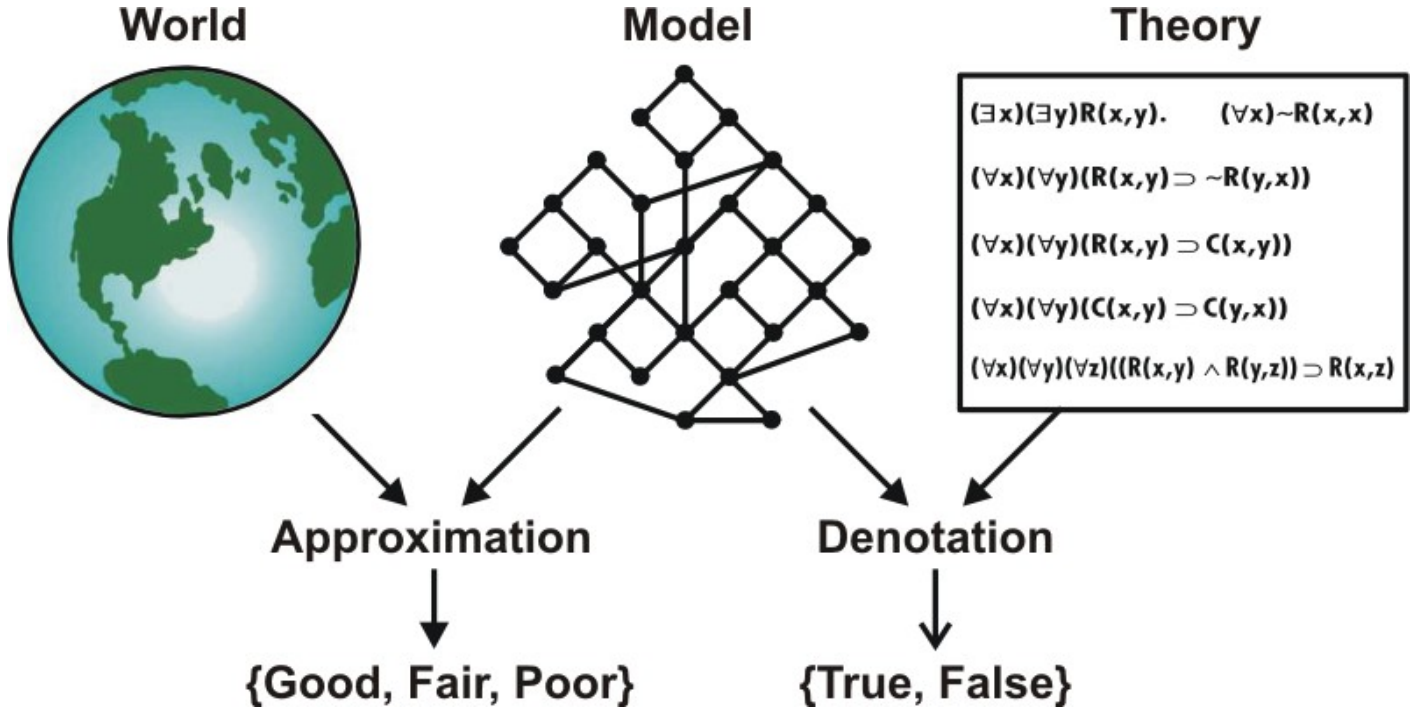
A physician, a pharmacist, a nurse, and a patient, for example, may talk about the same case, but they would discuss different details from different points of view.

The diagram on the right illustrates an application of FCA methods.

See G. Stumme & A. Maedche, Ontology merging for federated ontologies on the Semantic Web.



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.**
- **SW tools should be flexible: Detailed levels must be precise, but the ontology must accommodate anything imaginable.**

Knowledge of Good and Evil

Observation by Immanuel Kant:

> ***Socrates said he was the midwife to his listeners, i.e., he made them reflect better concerning that which they already knew, and become better conscious of it. If we always knew what we know, namely, in the use of certain words and concepts that are so subtle in application, we would be astonished at the treasures contained in our knowledge...***
>
> ***Platonic or Socratic questions drag out of the other person's cognitions what lay within them, in that one brings the other to consciousness of what he actually thought.***
>
> **From his *Vienna Logic***

C. S. Peirce: ***Logic is a sort of tree of knowledge of good and evil which costs the loss of paradise to him who tastes of its fruit.***

But good tools may help us "drag out" the treasures and the treachery hidden in the sources of our knowledge.

Related Readings

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

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

Majumdar, Arun K., & John F. Sowa (2018) Relating language, logic, and imagery, http://jfsowa.com/pubs/relating.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. (2010) Role of logic and ontology in language and reasoning, http://www.jfsowa.com/pubs/rolelog.pdf

Sowa, John F. (2011) Peirce's tutorial on existential graphs, http://www.jfsowa.com/pubs/egtut.pdf

Sowa, John F. (2013) What is the source of fuzziness? http://jfsowa.com/pubs/fuzzy.pdf

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

Sowa, John F. (2018) Reasoning with diagrams and images, http://www.collegepublications.co.uk/downloads/ifcolog00025.pdf