Historical Perspectives
On Problems of Knowledge Sharing

John F. Sowa

6 March 2014

Latest version: http://www.jfsowa.com/talks/history.pdf

Prospects for a Universal Ontology

Many projects, many useful theories, but no consensus.

- 4th century BC: Aristotle’s categories and syllogisms.
- 12th to 16th c AD: Scholastic logic, ontology, and semiotics.
- 17th c: Universal language schemes by Descartes, Mersenne, Pascal, Leibniz, Newton, Wilkins. L’Académie française.
- 18th c: More schemes. Satire of the Grand Academy of Lagado by Jonathan Swift. Kant’s categories.
- 19th c: Ontology by Hegel, Bolzano. Roget’s Thesaurus. Boolean algebra. Modern science, philosophy of science, early computers.
- Late 19th and early 20th c: FOL. Set theory. Ontology by Peirce, Brentano, Meinong, Husserl, Leśniewski, Russell, Whitehead.
- 1970s: Databases, knowledge bases, and terminologies.
- 1980s: Cyc, WordNet, Japanese Electronic Dictionary Research.
- 1990s: Many research projects. Shared Reusable Knowledge Base (SRKB), ISO Conceptual Schema, Semantic Web.
- 21st c: Many useful terminologies, but no universal ontology.

The Challenge

World’s Largest Formal Ontology

Cyc project founded by Doug Lenat in 1984:

- Name comes from the stressed syllable of encyclopedia.
- Starting goal: Implement the background knowledge of a typical high-school graduate.
- Ultimate goal: Learn new knowledge by reading textbooks.

After the first 25 years,

- 100 million dollars and 1000 person-years of work,
- 600,000 concepts,
- Defined by 5,000,000 axioms,
- Organized in 6,000 microtheories.

Some good applications, but more work is needed:

- Cyc cannot yet learn by reading a textbook.
- Cyc software is not well integrated with mainstream IT.

Cyc at the Cleveland Clinic

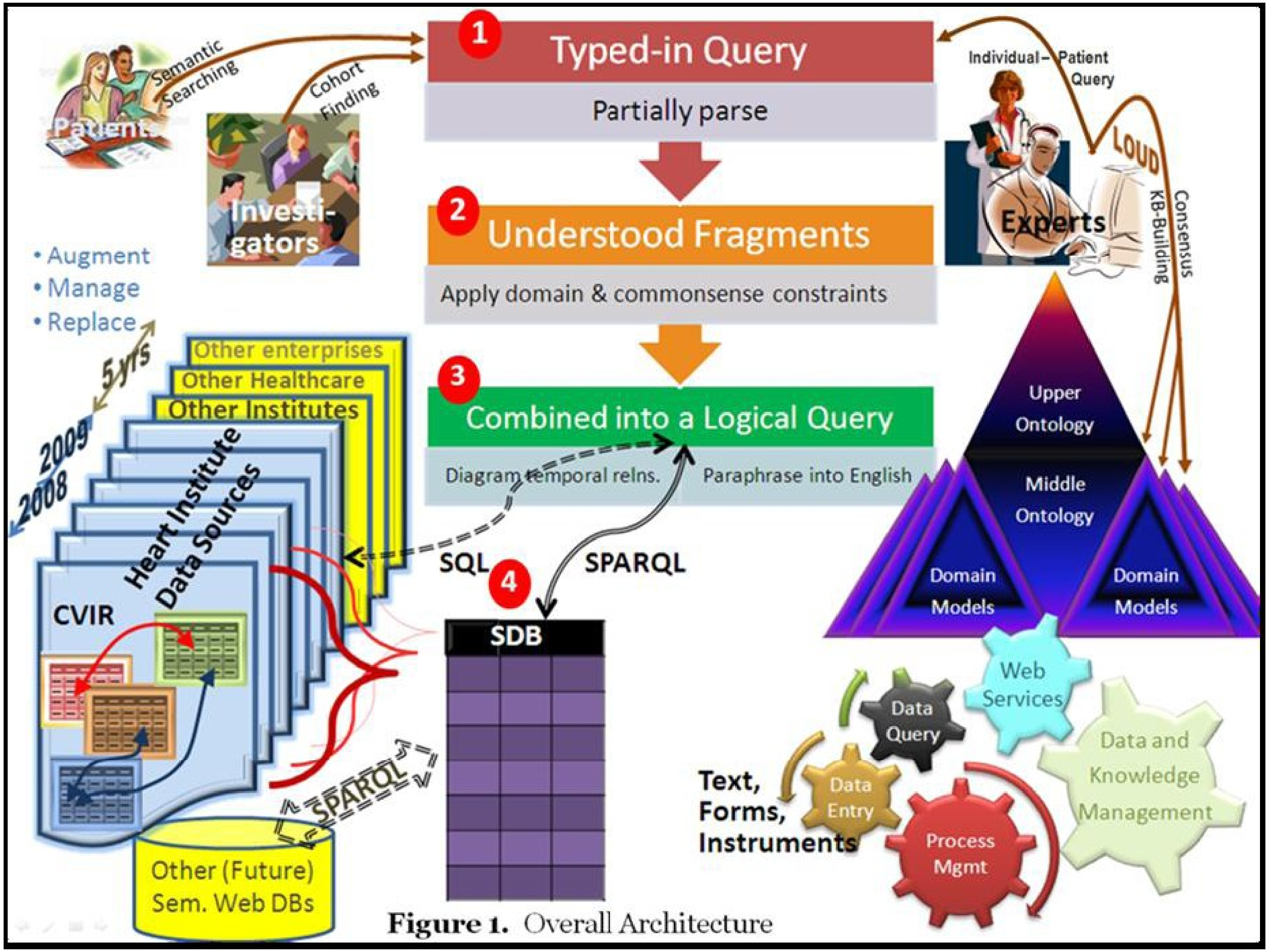


Figure 1. Overall Architecture

Source: http://www.cyc.com/technology/whitepapers_dir/Harnessing_Cyc_to_Answer_Clincal_Researchers_ad_hoc_Queries.pdf

Mismatched Design Patterns

Problems at the Cleveland Clinic:

All the tools process the same semantics.

But major differences in notations and methodologies.

Steep learning curve for IT personnel who try to use Cyc.

Complaint by Terry Longstreth at a DB symposium in 1980:

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

Over thirty years later, that statement is just as true.

We need better tools, interfaces, and methodologies:

Experts in any field spend years to become experts.

They don't have time to learn complex tools and notations.

The ideal amount of training time is ZERO.

Subject-matter experts should do productive work on day 1.

Supporting Interoperability

For over 50 years, computer systems have been interoperating and sharing data without using any formal semantics.

Every branch of science and engineering uses multiple, often inconsistent approximations for different kinds of problems.

There is no possibility of a single, unified, consistent, detailed ontology that can support multiple inconsistent applications.

Cyc, for example, has an underspecified upper-level ontology and 6000 detailed microtheories with complex interrelationships.

Questions:

- How can applications with inconsistent semantics share data?
- What are the criteria for sharing data among inconsistent systems?
- Would semantic definitions with formal logic and ontology help?
- Or would they create more problems than they could solve?
- How can we detect, avoid, or work around the inconsistencies?

Aristotle’s Categories

Ten ways of describing anything that exists or can exist.

Each category has a corresponding question:

Substance – What is it?

Relation – Toward what?

Quantity – How much?

Quality – What kind?

Activity – Doing what?

Passivity – Undergoing what?

Condition – Having what?

Position – How situated?

Place – Where?

Time – When?

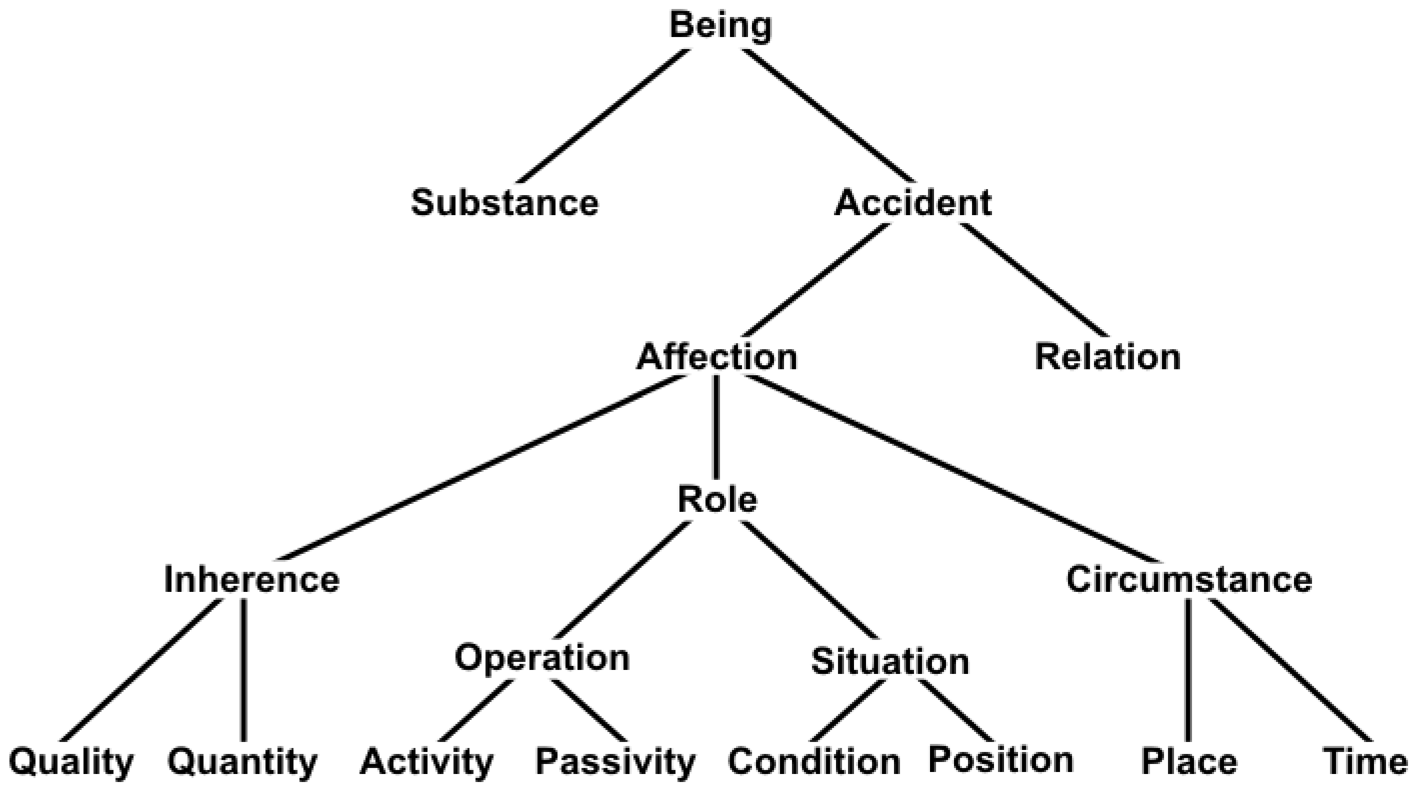
Substance is the unchanging form that determines what something is.

The other nine categories describe accidents that can change.

Describing George Washington

Category	Question	Answer
Substance	What is it?	A man
Relation	Toward what?	A general of the US army
Quality	What kind?	Handsome
Quantity	How much?	Tall
Activity	Doing what?	Talking
Passivity	Undergoing what?	Listening
Condition	Having what?	Victory in battle
Position	How situated?	Mounted on a horse
Place	Where?	Yorktown, Virginia
Time	When?	19 October 1781, 2 pm

Tree of Aristotle's Categories



Aristotle's categories, as arranged by Franz Brentano (1862).

The ten categories are the endpoints (leaves) of the tree.

The branch points are based on writings by Aristotle.

Patterns that Relate Categories

Four sentence patterns used in categorial syllogisms:

A	**Universal affirmative:**	**Every S is P.**
I	**Particular affirmative:**	**Some S is P.**
E	**Universal negative:**	**No S is P.**
O	**Particular negative:**	**Some S is not P.**

Boethius (6th century AD) introduced the letters A, I, E, and O as mnemonics for naming and remembering the combinations:

- **A and I are the first two vowels in the Latin *affirmo* (I affirm).**
- **E and O are the first two vowels in *nego* (I deny).**

The letters S and P represent *terms* in the subject and predicate.

Each term is a word that names a category or a phrase that states some *differentia* for defining a category.

Categorical Syllogisms

Barbara, the name of the first pattern, has three As, which indicate three type A sentences:

A Every quadruped has four legs.
A Every elephant is a quadruped.
A Therefore, every elephant has four legs.

The pattern named Darii has sentence types A, I, I:

A Every mammal breathes oxygen.
I Some sea creature is a mammal.
I Therefore, some sea creature breathes oxygen.

The pattern Barbara supports the inheritance of properties from a supertype to every individual of a subtype.

The pattern Darii supports the inheritance of properties from a type to particular individuals of a different type.

Negative Syllogisms

The first negative syllogism is named Celarent:

E No animal with lungs is an oyster.
A Every mammal is an animal with lungs.
E Therefore, no mammal is an oyster.

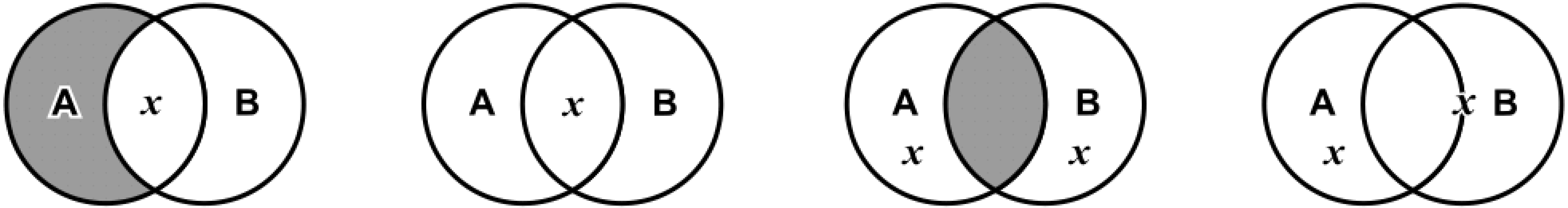
The negative syllogism Ferio applies to particular individuals:

E No herbivore eats meat.
I Some mammal is a herbivore.
O Therefore, some mammal does not eat meat.

E and O sentences express constraints on the type hierarchy.

Negative syllogisms derive the implications of those constraints.

Venn Diagrams



Every A is B

Some A is B

No A is B

Some A is not B

In the 19th century, John Venn used patterns of overlapping circles to determine the valid patterns of syllogisms.

Each circle contains every instance x for which some A is true.

Four kinds of areas:

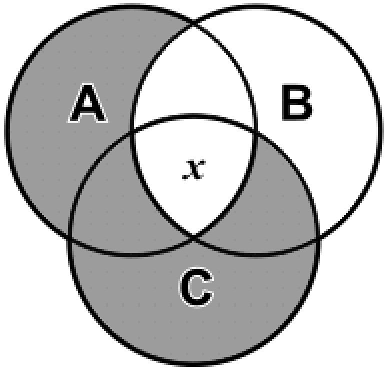
- Shaded area: known to be empty – no instances exist.
- White area: contents unknown – some instance x might exist.
- White area marked with x: contains at least one instance x.
- Border marked with x: some x exists on one side or the other.

Note: Aristotle assumed that every category has at least one x.

Venn Diagrams for Barbara and Darii

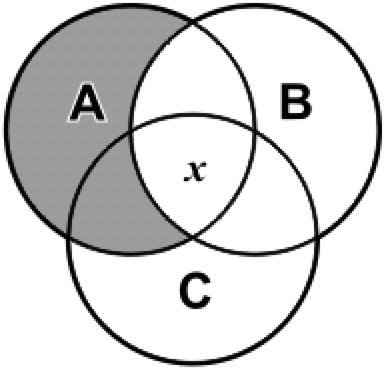
For the pattern Barbara, the area of A outside B is empty, and the area of C outside A is empty.

Every A is B.
Every C is A.
___
∴ Every C is B.



For Darii, the area of A outside B is empty, but the area of C outside A is irrelevant.

Every A is B.
Some C is A.
___
∴ Some C is B.



Therefore, any data that makes Barbara true will also make Darii true.

This observation shows that Aristotle's derivation of the pattern Darii from Barbara is valid: it preserves truth.

John Venn invented the diagrams for analyzing syllogisms.

They're still a good way to show the patterns in an ontology.

Universal Language Schemes

In the 16th century, books in modern languages were rapidly displacing Latin.

Scientists, philosophers, merchants, bankers, and diplomats felt the need for a new universal language.

Francis Bacon claimed that “real characters” similar to Chinese characters could be used for mutually unintelligible languages.

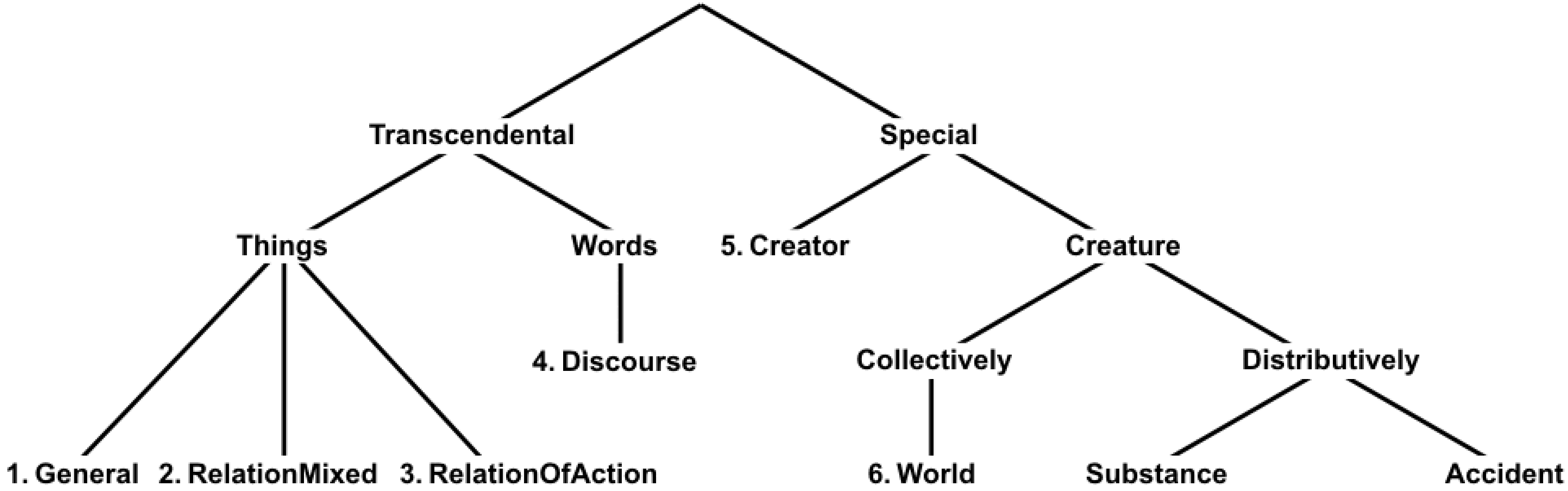
Descartes, Mersenne, Pascal, Newton, and Leibniz proposed mathematical principles as the basis for a universal language.

The largest and most impressive system was the *Real Character and Philosophical Language* by John Wilkins.

Wilkins was secretary of the British Royal Society. Several other members collaborated on the project.

For further discussion, see Knowlson (1975), Eco (1995), and Okrent (2009).

Wilkins’ Upper-Level Ontology



Transcendental
Special
Things
Words
5. Creator
Creature
4. Discourse
Collectively
Distributively
1. General
2. RelationMixed
3. RelationOfAction
6. World
Substance
Accident

In a 600-page book, Wilkins (1668) devoted 270 pages to tables that define 40 genera subdivided in 2,030 species.

The categories labeled 1 through 6 are the first of his 40 genera. The other 34 genera are subtypes of Substance or Accident.

Inheritance: Each species is defined by the conjunction of all the differentiae along the path from one of the 40 genera.

Summary of Wilkins’ System

An impressive combination of upper-level ontology, metalevel ontology, mid-level ontology, thesaurus, and notation.

A failure as a replacement for Latin, but an inspiration for Leibniz, Kant, Roget, and many others.

The division of Transcendental vs. Special corresponds roughly to the distinction between signs and their referents.

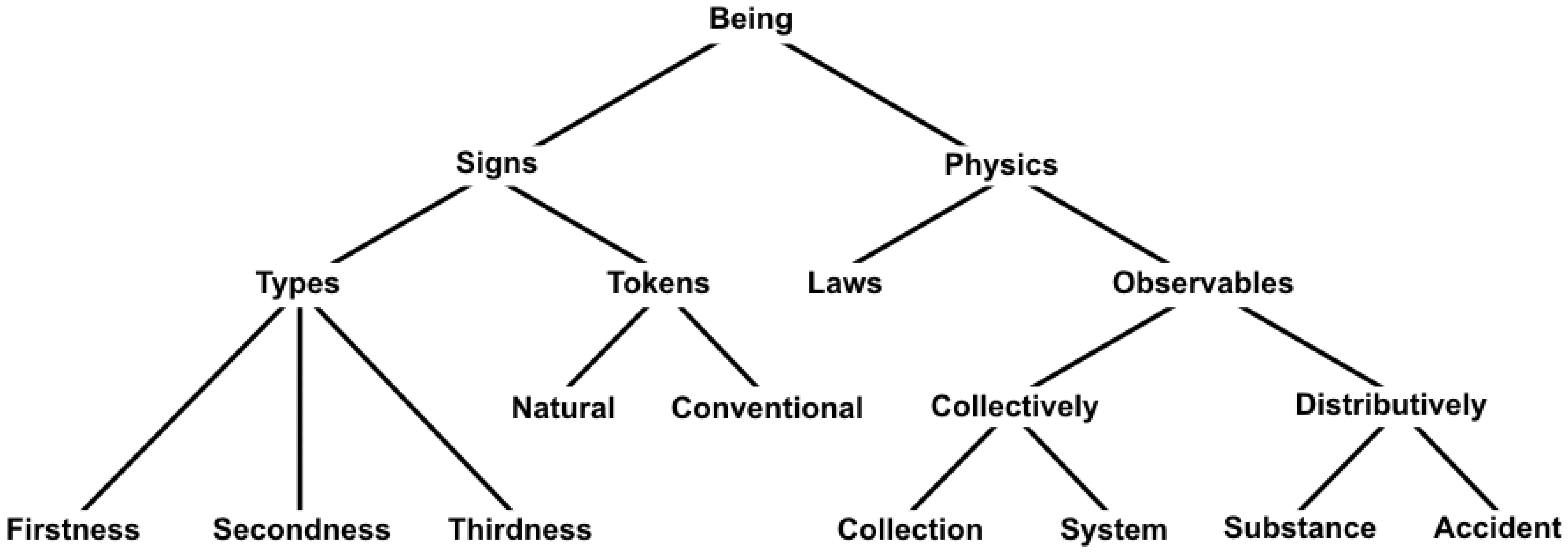
The division of Collectively vs. Distributively is an important distinction that many ontologies ignore.

But 2,030 categories at the endpoints of the tree are inadequate for a general-purpose language.

Other members of the Royal Society added about 15,000 English words as approximate synonyms of those 2,030 categories.

Unfortunately, the system contained many ad hoc features that were ridiculed by Jonathan Swift and Jorge Luis Borges.

Updated Version of Wilkins' Ontology



This diagram preserves the pattern, but it relabels the nodes:

- **The top division distinguishes signs from their physical referents.**
- **The creator is replaced by the laws of physics. Theists can think of the laws as the *logos*, which John the Evangelist said is God.**
- **Sign types are defined by laws, and sign tokens refer to observables.**
- **The types can be organized in triads, as defined by C. S. Peirce**

Can Any Ontology be Complete?

The updated version of Wilkins' ontology is more complete than the great majority of published ontologies:

- **Signs would include all languages, natural or artificial, and any kind of data or metadata on the WWW.**
- **Laws would include theories about any natural, artificial, planned, hypothetical, or fictional phenomena in the universe.**
- **Collections and Systems include all structures and organizations.**

But science, technology, and the world are constantly changing.

- **A general framework can remain useful for centuries.**
- **But nobody can anticipate the innovations in the next 20 years.**

Observation by Alfred North Whitehead:

"Systems, scientific and philosophic, come and go. Each method of limited understanding is at length exhausted. In its prime, each system is a triumphant success: in its decay it is an obstructive nuisance."

Thesaurus vs. Ontology

Peter Roget was a secretary of the Royal Society who developed a thesaurus of words instead of an ontology of things.

- A much simpler system of classification than Wilkins'.
- A top level with just six categories: Abstract relations, Space, Matter, Intellect, Volition, Affections.
- A bushy hierarchy with just three layers beneath the top level.
- No definitions, differentiae, inheritance, or logic.

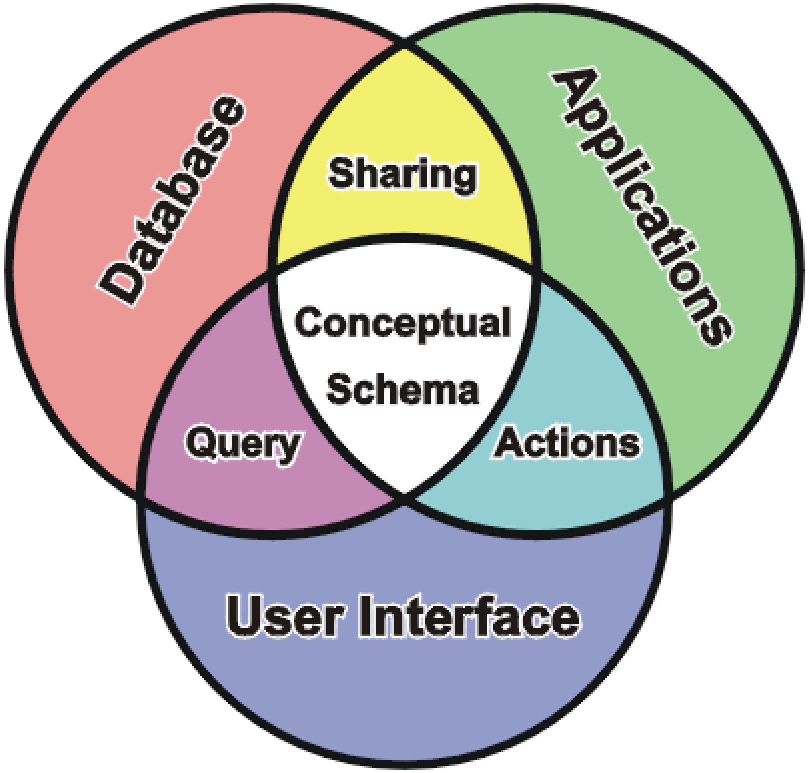
Roget's first edition (1852) was an instant success:

- By 1869, he had produced 28 editions; his son continued the work.
- Computer versions are used in natural language processing (NLP).

The modern WordNet is closer to a thesaurus than an ontology:

- WordNet has a simple top level.
- It has no formal definitions, differentiae, inheritance, or logic.
- But it is a widely used resource for NLP in several languages.

Conceptual Schema



Three-schema architecture by ANSI SPARC in 1978:

- Conceptual schema defines the semantics of a database.
- Physical schema defines the storage and access methods.
- Application schema defines the APIs for programming languages.

Standardizing the Conceptual Schema

Many database experts agreed:

- Logic is neutral on the issue of data storage and access.
- Issues of storage and access are not relevant to semantics.
- Therefore, logic is the only logical choice for the conceptual schema.

But other DB experts disagreed:

- They claimed that programmers should specify the data formats.
- Some claimed that logic was hard for people to understand.
- Commercial DB vendors did not want to disrupt their implementations.

Over thirty years of R & D and ISO proposals:

- Better notations for logic than the SQL where-clause.
- Methodologies, tools, and technical reports, but no standards.

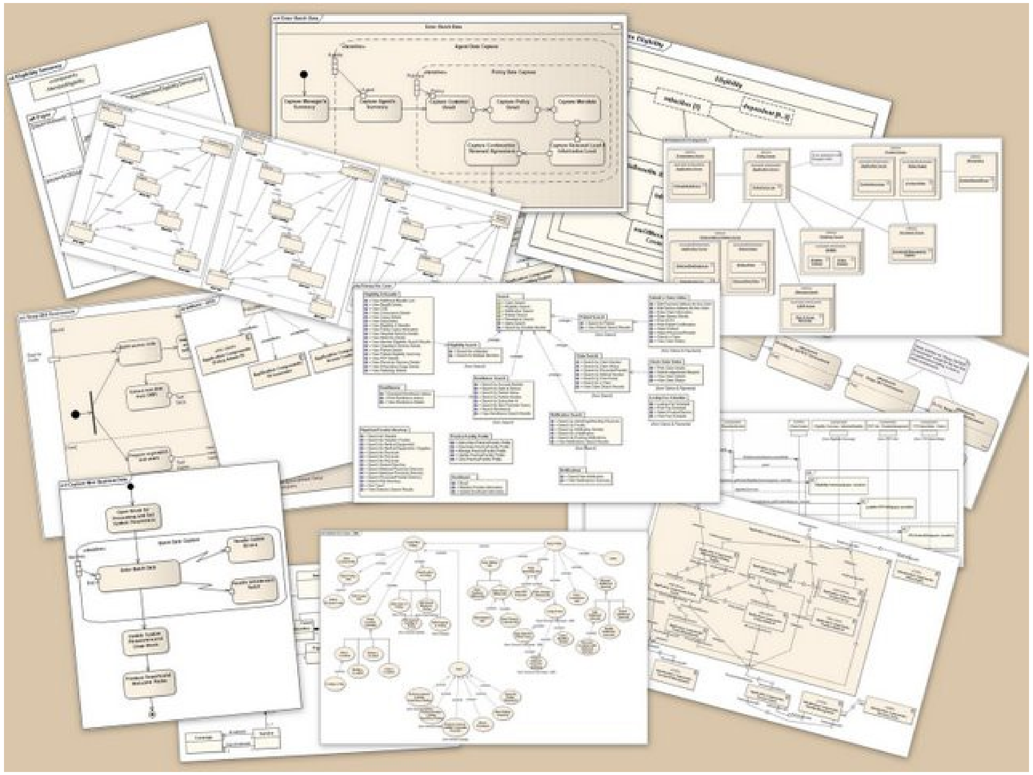
See The Orange Report ISO TR9007 (1982 – 1987): Grandparent of the Business Rules Approach and SBVR, History of the ISO TC97/SC5/WG3 Working Group, *Business Rules Journal,* by J. J. van Griethuysen, Vol. 10, No. 4 (April 2009), http://www.BRCommunity.com/a2009/b474.html

Unified Modeling Language (UML)

A family of readable diagrams that express various subsets of logic and ontology.

Adopted as a standard by the Object Management Group (OMG).

Originally specified as informal notations without a foundation in logic.



The current standard specifies the base semantics in Common Logic.

See http://www.omg.org/spec/FUML/1.1

Relating Language to Logic

Peirce wrote a succinct, but accurate summary of the issues:

> ***"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."*** **(CP 4.237)**

Implications:

- **A precise formal ontology of everything can be stated in logic, but it's almost certainly false in many important respects.**
- **A looser classification, such as WordNet or Roget's *Thesaurus*, can be more flexible for representing patterns of words.**
- **A specification in logic can be "pretty precise and fairly certain" only for a very narrow subject.**

Logic is an abstraction from language that emphasizes patterns of reasoning, but the patterns of words are also important.

Organizing a Large Ontology

No single ontology can ever be complete, consistent, and useful.

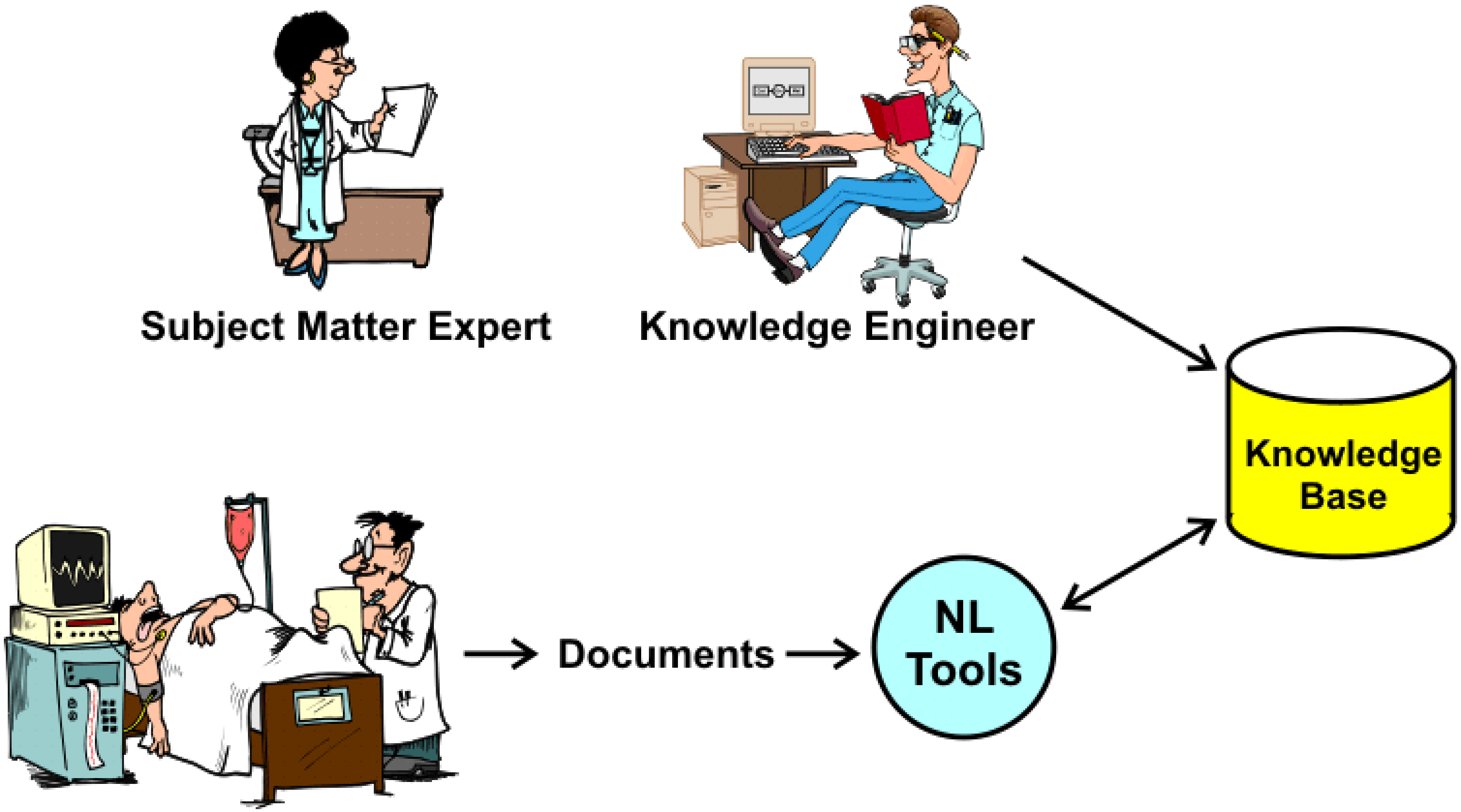
An underspecifed framework for everything with an open-ended collection of microtheories is more useful and practical:

- **An upper-level ontology, such as Aristotle’s, Wilkins’, or Kant’s, can show the broad patterns of how everything fits together.**
- **But different problems for different purposes require different representations and algorithms for processing the details.**
- **For interoperability, upper level definitions must be underspecified with the barest minimum of axioms and differentiae.**
- **For precise reasoning and problem solving, the details must be pushed down to the highly specialized, low-level microtheories.**

To be humanly intelligible, ontology must be related to language.

Lexical resources that show the patterns of words are valuable, but they should never be confused with ontologies.

Old Fashioned Knowledge Acquisition



Knowledge engineers acquire knowledge from a SME and translate it to some knowledge representation language.

But a KE is also a highly trained professional.

Hiring a SME to train a KE doubles the cost.

A Better Division of Labor

Knowledge system:

- Acquire knowledge from structured data, training examples, unstructured natural language, and occasional questions.
- Communicate with people in any notation they prefer.

Subject matter experts:

- Answer questions and correct errors by the system.
- Communicate in their preferred notations and diagrams.
- Contact a KE only to report problems or to request new features.

Knowledge engineers:

- Work with application programmers on the interfaces between the programs and the knowledge system.
- Respond to requests by the SMEs.

A Migration Path to the Future

Any declarative notation, graphic or linear, can be mapped to some version of logic

Good development methodologies supported by controlled natural languages can be used effectively by nonprogrammers.

Recommendation for a new generation of development tools:

- Integrate all systems, including legacy systems, with logic-based methodologies.
- Enable subject-matter experts to review, update, and extend their knowledge bases with little or no assistance from IT specialists.
- Provide tools that support collaboration, review, and testing by people with different levels and kinds of expertise.

Knowledge Discovery

Observation by the philosopher 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***

We need tools that can play the role of Socrates.

They should help us discover the implicit knowledge and use it to process the huge volumes of digital data.

Related Readings

Future directions in semantic systems,
http://www.jfsowa.com/pubs/futures.pdf

Fads and fallacies about logic,
http://www.jfsowa.com/pubs/fflogic.pdf

The role of logic and language in ontology,
http://www.jfsowa.com/pubs/rolelog.pdf

Conceptual graphs,
http://www.jfsowa.com/cg/cg_hbook.pdf

Slides for a tutorial on the goal of language understanding,
http://www.jfsowa.com/talks/goal.pdf

Web site for controlled natural languages,
http://sites.google.com/site/controllednaturallanguage/

For other references, see the bibliography: http://www.jfsowa.com/bib.htm