Two Paradigms are Better than One, And Multiple Paradigms are Even Better

John F. Sowa and Arun K. Majumdar
VivoMind Intelligence

6 August 2009

How to Design More Intelligent Systems

The Cyc approach to "classical" artificial intelligence:

- Design a very large ontology with millions of axioms.
- Build a few dozen inference engines to process the axioms.
- Use deductive logic as the primary paradigm for reasoning.

Marvin Minsky's Society of Mind:

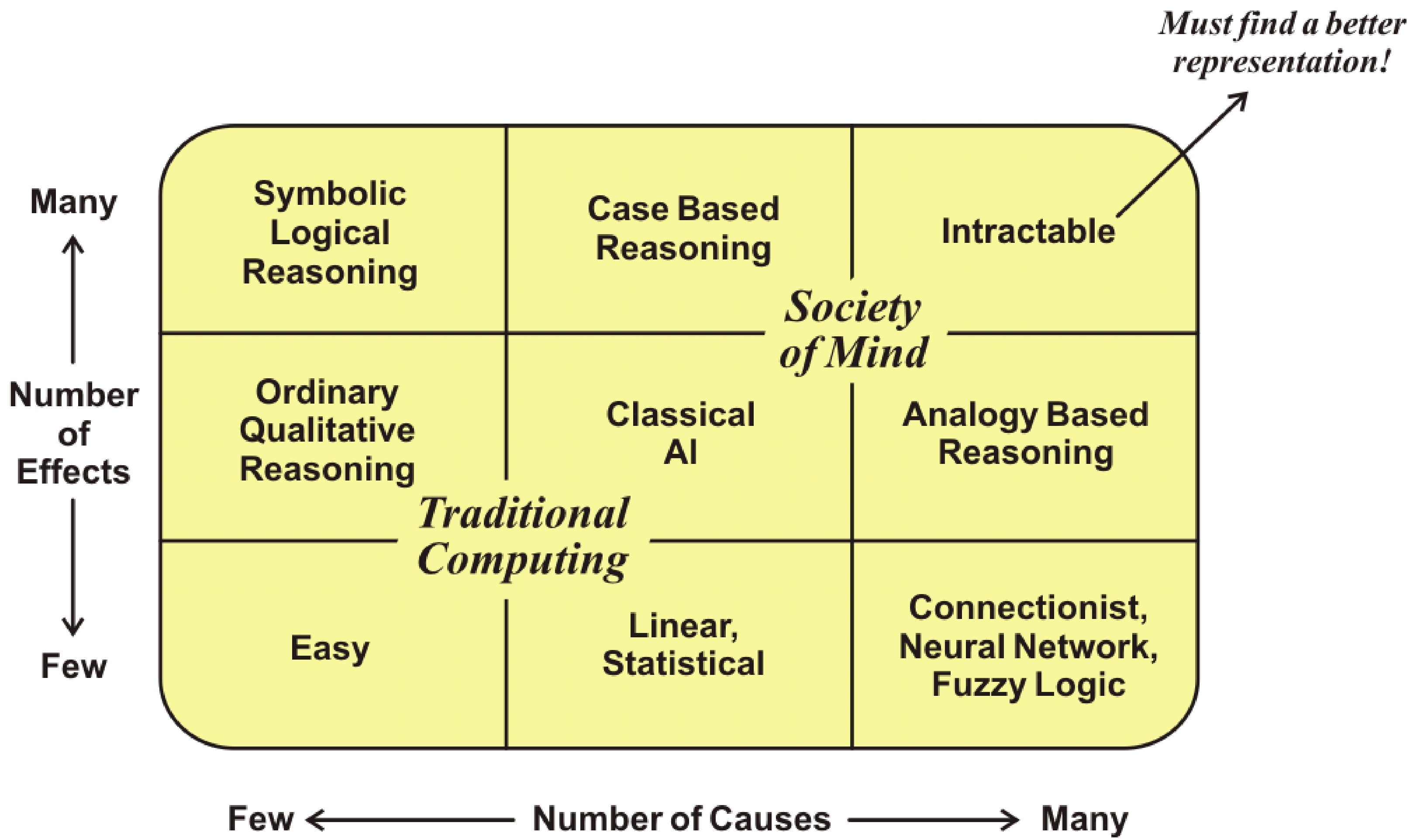
- Support reasoning by an open-ended number of agents.
- Allow different agents to use different reasoning paradigms.
- Use learning mechanisms to adapt the society to diverse problems.
- A powerful set of ideas that haven't been fully exploited.

Society of Mind

Quotations by Marvin Minsky:

- What magical trick makes us intelligent? The trick is that there is no trick.
- The power of intelligence stems from our vast diversity, not from any single, perfect principle.
- Our species has evolved many effective although imperfect methods, and each of us individually develops more on our own.
- Eventually, very few of our actions and decisions come to depend on any single mechanism. Instead, they emerge from conflicts and negotiations among societies of processes that constantly challenge one another.
- To develop this idea, we will imagine first that this Mental Society works much like any human administrative organization.

The Causal Diversity Matrix



- State of the art in 2004.
- Adapted from a paper by Minsky, Singh, and Sloman.
- See http://www.cs.bham.ac.uk/research/projects/cogaff/AIMag/singh-minsky-sloman-aimag-04.pdf

Scalability

Traditional computing is good for regular problems:

- A small number of causes (types of interactions).
- Each cause has a small number of possible effects.
- The combinations can be processed with a nest of for-loops.

Classical artificial intelligence:

- Axioms represent complex, highly irregular problems.
- Can support moderate numbers of causes and effects.

Minsky's Society of Mind:

- Multiple paradigms can support 8 of the 9 boxes of the matrix
- None of them scale to problems in the 9th box: large numbers of causes and effects.
- But people can and do reason about such problems. How?

Organizing a Society of Agents

1. Pandemonium. Selfridge (1959) designed a system of agents called demons. Each demon could observe aspects of the current situation or workspace, perform some computation, and put its results back into the workspace.

2. Rational agents. At the opposite extreme from simple demons are rational agents that simulate a human-like level of beliefs, desires, intentions, and the ability to reason about them.

3. Reactive agents. For designing robots, Brooks (1991) noted that the major challenge was not in deliberative planning and reasoning, but in the seemingly simpler insect-like functions of perception, locomotion, and goal seeking.

Instead of these three kinds of organizations, Minsky recommended a hierarchy of managers and employees.

A business organization enables tighter control over resources while accommodating agents with a wide variety of talents.

Flexible Modular Framework (FMF)

A platform for agents that communicate by passing messages.

Each message has 6 fields:

1. Language. Identifier of the language used in the message.
2. Source. Identifier of the agent that sent the message.
3. Message ID. Identifier generated by the sender.
4. Destination. Identifier of the intended receiver, if known.
5. Pragmatics. The speech act or other purpose for the message.
6. Message. Any sentence or list of sentences in language #1.

If the destination is 0, the message is sent to an associative blackboard, where suitable agents can find it and answer it.

Agents can be anywhere on the Internet.

Implementing a Society of Agents

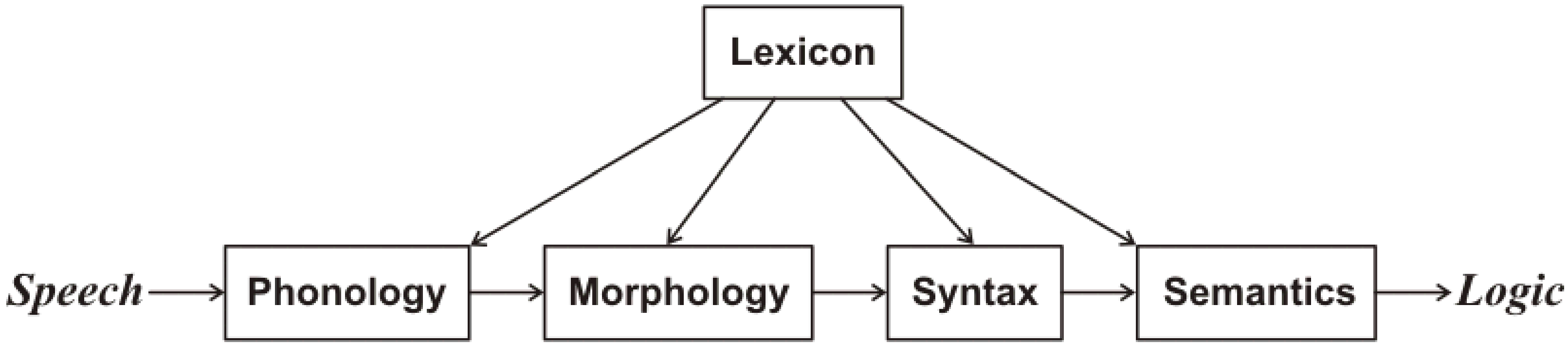
FMF protocols are simple and general:

- Based on message passing among agents of any kind.
- No restrictions on the message types or the language they use.
- No built-in or preferred organization among agents.
- Minimum space for a simple agent is 8K bytes.
- But agents can be arbitrarily large and complex.

Using the FMF to implement Minsky's hierarchy of agents:

- One agent is the Chief Executive Officer (CEO).
- Every employee except the CEO reports to exactly one manager.
- All employees report directly or indirectly to the CEO.
- All employees perform tasks assigned by their managers.
- Managers reward employees with space and time resources.

Traditional Language Processing

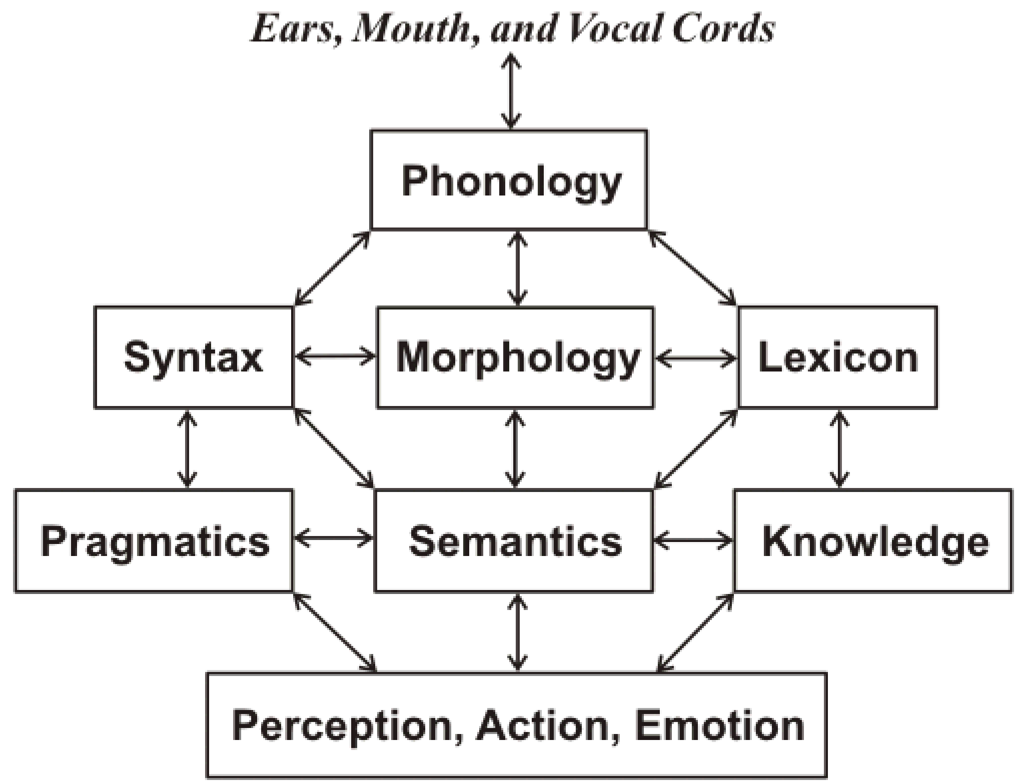


The output of each stage is the input to the next stage.

A later stage, such as semantics, can choose among the options identified by an earlier stage, such as syntax.

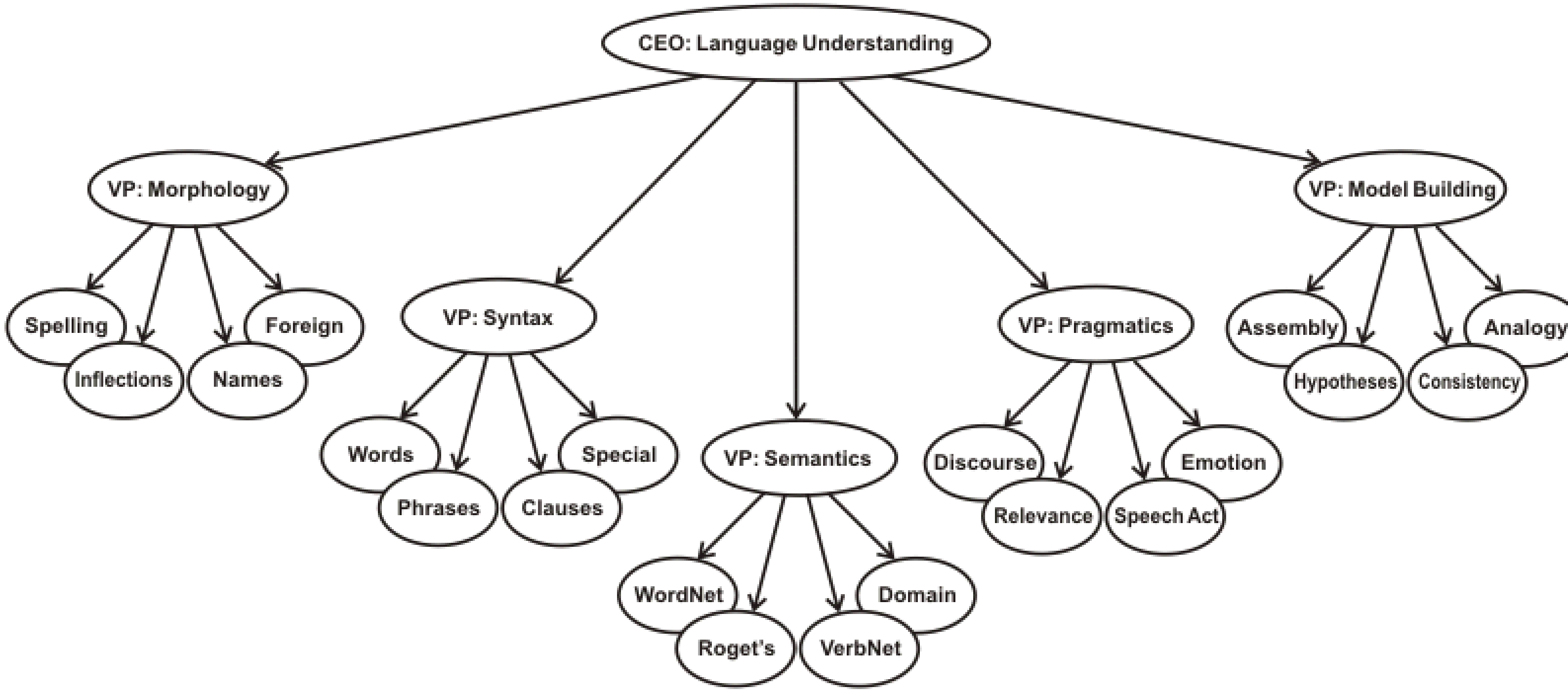
But semantics cannot guide, correct, or modify the options chosen by syntax, morphology, or phonology.

Diversity of Modules



Everything is interconnected with everything else.

A Hierarchy for Language Understanding



The CEO determines the goals and allocates resources.

Managers and employees are rewarded with resources (space and computer time) depending on how well they meet the goals.

Agents on all branches and levels can work in parallel.

Parsing Technology

Ideas that influenced the VivoMind Language Processor:

1. **Generate a dependency graph with a Link Grammar.**

   Sleator, Daniel, & Davy Temperley (1993) Parsing English with a link grammar, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/LG-IWPT93.ps

2. **Enable multiple agents to collaborate during the parsing.**

   Hahn, Udo, Susanne Schacht, & Norbert Bröker (1994) Concurrent natural language parsing, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9457

3. **Let semantic agents contribute knowledge during the parsing.**

   Bröker, Norbert (1999) *Eine Dependenzgrammatik zur Kopplung heterogener Wissensquellen*, http://elib.uni-stuttgart.de/opus/volltexte/1999/398/pdf/398_1.pdf

4. **Use a voting scheme to select among conflicting options.**

   Zeman, Daniel, & Zdenek Žabokrtský (2005) Improving parsing accuracy by combining diverse dependency parsers, http://www.aclweb.org/anthology-new/W/W05/W05-1518.pdf

Sample Sentence

The Diana field is situated in the western Gulf of Mexico
260 km (160 mi) south of Galveston
in approximately 1430 m (4700 ft) of water.

Syntax alone is sufficient to parse the first line correctly.

But the next two lines raise more complex issues:

- **What is south of Galveston? The western Gulf of Mexico? Diana field?**
- **What is in the water? Diana field? The western Gulf of Mexico? Galveston?**
- **How are the parenthesized phrases related to the rest of the sentence?**

How do the agents interact to process this sentence?

Agents for Morphology

The Diana field is situated in the western Gulf of Mexico
260 km (160 mi) south of Galveston
in approximately 1430 m (4700 ft) of water.

Morphology agents

- initialize a workspace for the information needed to build a conceptual graph,
- look up words and phrases in dictionaries and lists of names,
- identify 'Diana' as a proper name, 'Gulf of Mexico' as a body of water, and 'Galveston' as a city,
- identify 'is' as a present tense verb and 'situated' as a past tense or past participle,
- recognize 'km', 'mi', 'm', and 'ft' as abbreviations for 'kilometer', 'mile', 'meter', and 'foot'.

Agents for Syntax

The Diana field is situated in the western Gulf of Mexico
260 km (160 mi) south of Galveston
in approximately 1430 m (4700 ft) of water.

Syntax agents

- start a bottom-up process of linking words,
- link all the words of the first line into a dependency tree,
- form smaller, unlinked trees for '260 kilometers', '(160 miles)', 'south of Galveston', 'in approximately', '1430 meters', '(4700 ft)', 'of water'.
- wait for more information from other agents.

Multiple Semantic Resources

Many ontologies and lexical resources are available, but they are very different in format and coverage:

WordNet, Roget's Thesaurus, CoreLex, Framenet, VerbNet, etc.

Some of them can be merged.

But others are difficult or impossible to merge.

Solution:

- **Don't attempt to merge all resources in advance.**
- **Assign them to different agents, which can contribute relevant information during the parsing process.**
- **Use a voting mechanism to resolve conflicts as they occur.**

Agents for Semantics

The Diana field is situated in the western Gulf of Mexico
260 km (160 mi) south of Galveston
in approximately 1430 m (4700 ft) of water.

Semantic agents

- translate the dependency graph to a conceptual graph,
- treat 'the Diana field' as a single name and use the ontology for oil exploration to determine that it is a hydrocarbon reservoir,
- wait for more information from other agents before linking the trees for the last two lines of the sentence,
- finish connecting the nodes of the CG when more information becomes available.

Agents for Pragmatics

The Diana field is situated in the western Gulf of Mexico
260 km (160 mi) south of Galveston
in approximately 1430 m (4700 ft) of water.

Pragmatic agents

- perform context-dependent reasoning to complete and correct the conceptual graph generated by the semantic agents,
- note that '260 kilometers' is approximately equal to '160 miles'; assume that they are noun phrases in apposition and link the two concepts by a coreference link; do the same for '1430 meters' and '(4700 feet)',
- wait for semantic agents to link those concepts into subgraphs for lines two and three of the sentence,
- use background knowledge to prune incorrect options for attaching those subgraphs: cities should not be in water, but hydrocarbon reservoirs are under ground, which may be in water.

Concurrency

The previous description was largely sequential, but the agents at various levels can run concurrently.

The morphology agents normally begin the processing.

But as soon as a word is recognized by a morphology agent, other agents at later stages can contribute any information about that word they might find.

Information from later stages can often reduce the search by syntactic agents.

Information from semantic agents can often fill in gaps (ellipses) or correct errors in the syntax.

How can a computer understand language?

According to Alan Turing,

If people can’t tell the difference between
what a computer does and and what a person does,
then the computer is thinking the way people do.

A more implementable idea:

Human thinking is based on analogies.
People understand language by finding analogies between
patterns of words and patterns in what they see and do.

Research project:

How can an analogy engine make computers more human-like?

Computational Complexity

Research by Falkenhainer, Forbus, & Gentner:

Pioneers in finding analogies with their Structure Mapping Engine.

Demonstrated that the SME algorithms take time proportional to N-cubed, where N is the number of graphs in the knowledge base.

MAC/FAC approach: Use a search engine to narrow down the number of likely candidates before using SME.

VivoMind approach:

Encode graph structure and ontology in a Cognitive Signature™.

Find the closest matching signatures in logarithmic time.

Use structure mapping only on a very small number of graphs.

Chemical Graph Search Engine

Techniques for searching chemical graphs:

- Represent each graph by its International Chemical Identifier (InChI).
- Map the InChI codes to numeric vectors that encode both the graph structure and the labels of the atoms and bonds.
- Index the vectors by a locality-sensitive hashing (LSH) algorithm.
- Estimate the semantic distance between graphs by a measure of intermolecular similarity.
- Use the semantic distance measure to find the most similar graphs.

For a description of these algorithms,

Mining Patents Using Molecular Similarity Search
By James Rhodes, Stephen Boyer, Jeffrey Kreulen, Ying Chen, & Patricia Ordonez
http://psb.stanford.edu/psb-online/proceedings/psb07/rhodes.pdf

For indexing and searching over 4 million chemical graphs,

https://chemsearch.almaden.ibm.com/chemsearch/SearchServlet

Three Applications for an Analogy Engine

1. Aligning independently developed ontologies.

2. Educational Software:

   Evaluating student answers written in free-form English.

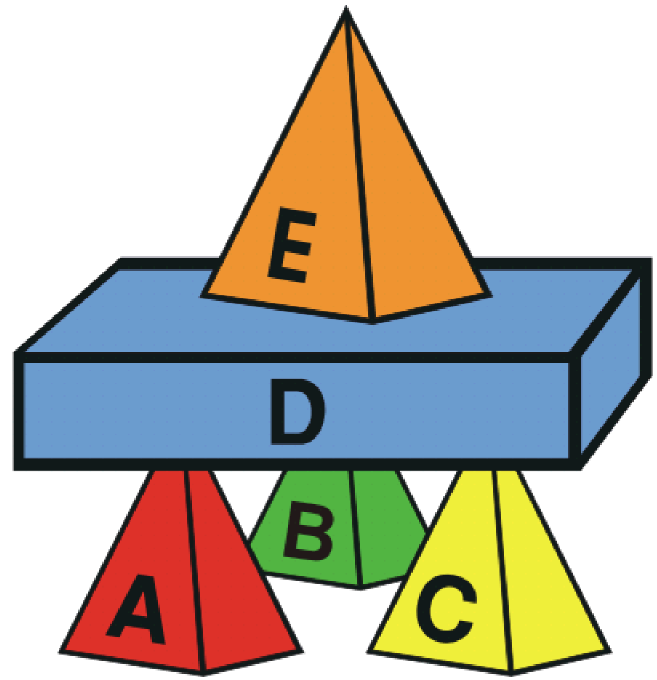
3. Oil and gas exploration:

   Reading English documents about oil and gas exploration and answering English questions written by a geologist.

Note: Applications #1 and #2 used an older version of the VivoMind software. Application #3 uses a completely rewritten system called the VivoMind Language Processor (VLP).

Describing Things in Different Ways

How can we describe what we see?



In ordinary language?

In some version of logic?

In a relational database?

In the Semantic Web?

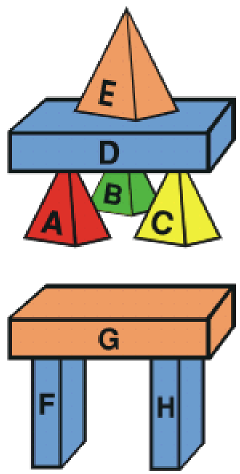
In a programming language?

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

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

Structured and Unstructured Representations

A description in tables of a relational database:



Objects

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

Supports

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

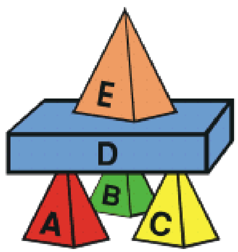
A description in English:

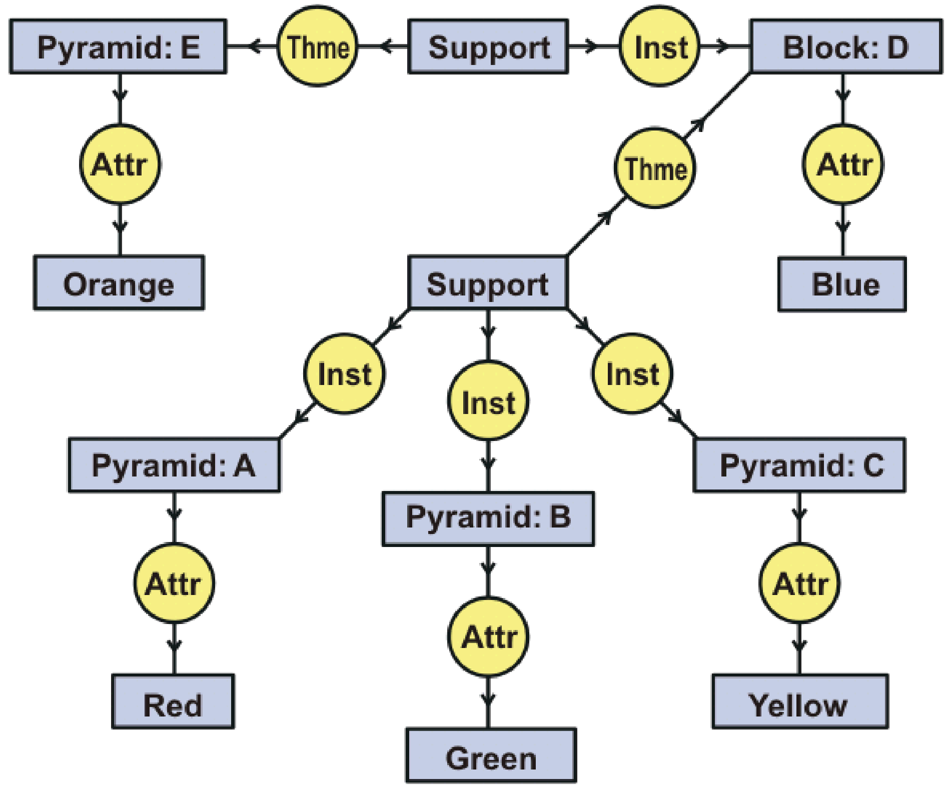
> **"A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E."**

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

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

Mapping English to a Conceptual Graph





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

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

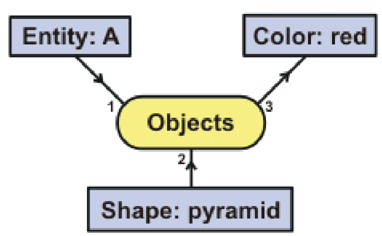
Mapping Database Relations to Conceptual Relations

Objects

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

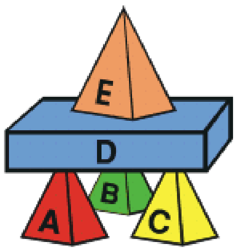
Supports

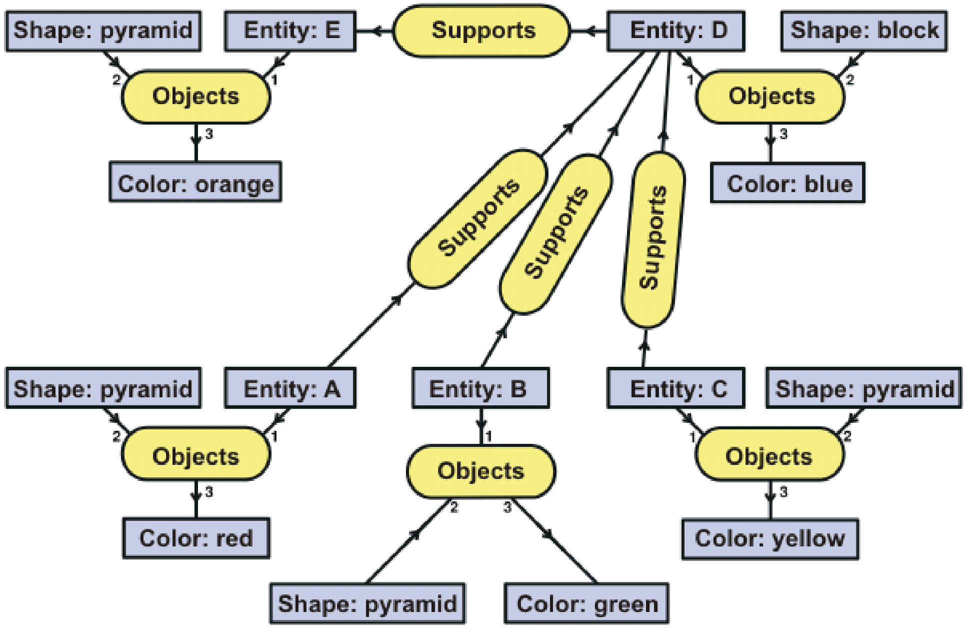
Supporter	Supportee
A	D
B	D
C	D
D	E
F	G
H	G



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

Mapping an Entire Database to Conceptual Graphs

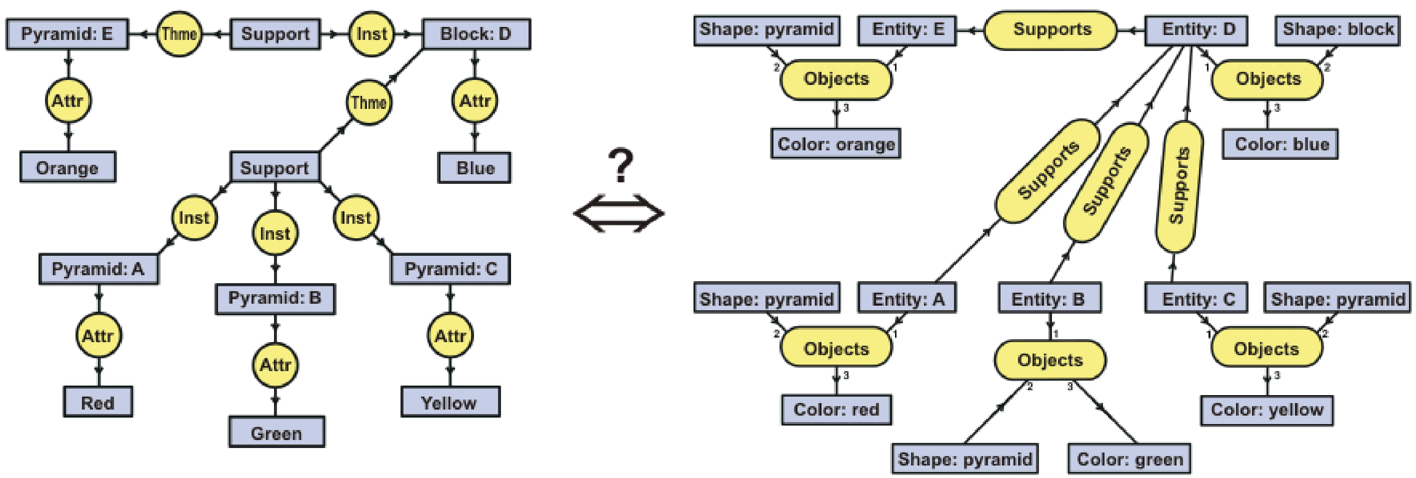




Join concept nodes that refer to the same entities.

Closely related entities are described by connected graphs.

Mapping the Two Graphs to One Another



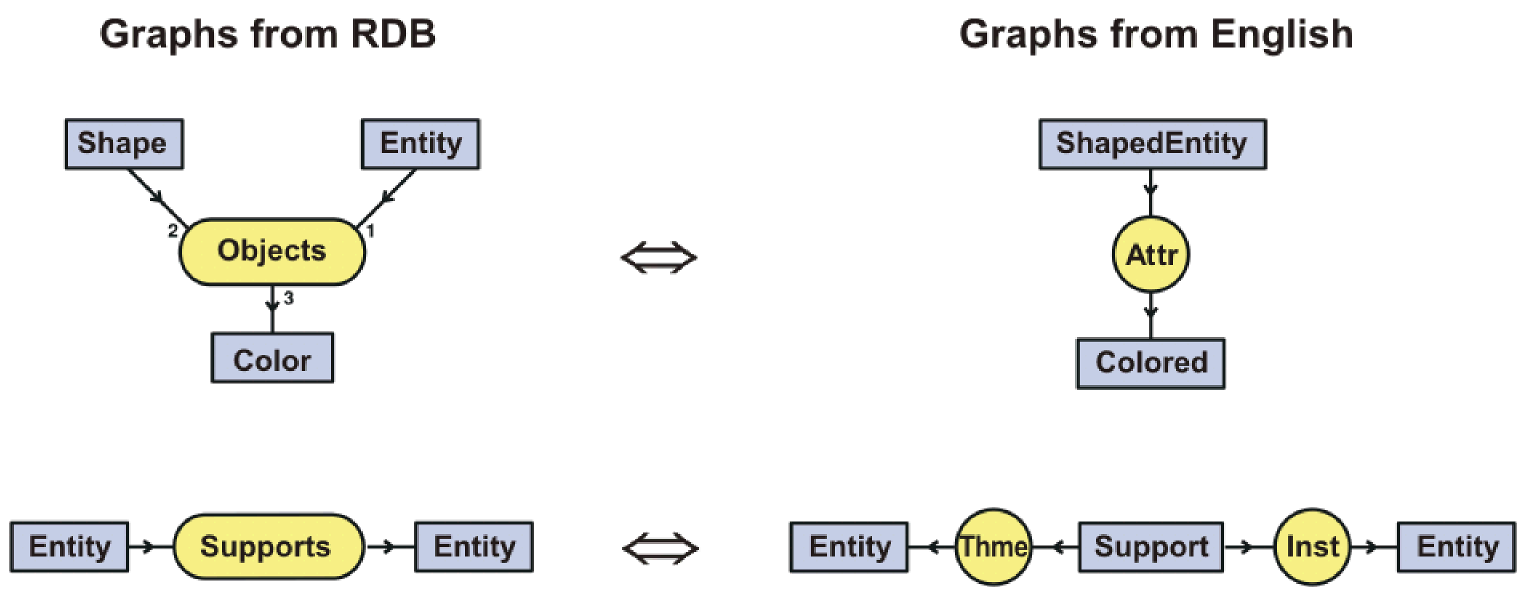
Very different ontologies: 12 concept nodes vs. 15 concept nodes, 11 relation nodes vs. 9 relation nodes, no similarity in type labels.

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

People can recognize the underlying similarities.

How is it possible for a computer to discover them?

Mapping the Graphs by Aligning the Ontologies



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

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

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

Application to Educational Software

A textbook publisher wanted to grade student answers to math problems.

But the answers were written in free-form English sentences.

Typical problem:

> ***The following numbers are 1 more than a square: 10, 37, 65, 82.***
> ***If you are given an integer N that is less than 200,***
> ***how would you determine whether N is 1 more than a square?***
>
> ***Explain your method in three or four sentences.***

How could a computer system evaluate answers to such problems?

Determine whether they are correct, incorrect, or partially correct?

And make helpful suggestions about the incorrect answers?

Publisher’s Current Procedure

To evaluate new exam questions, the publisher normally gives the exam to two or three classes of students.

For each problem, they would get about 50 different answers:

- Some are completely correct
  — but stated in different ways.

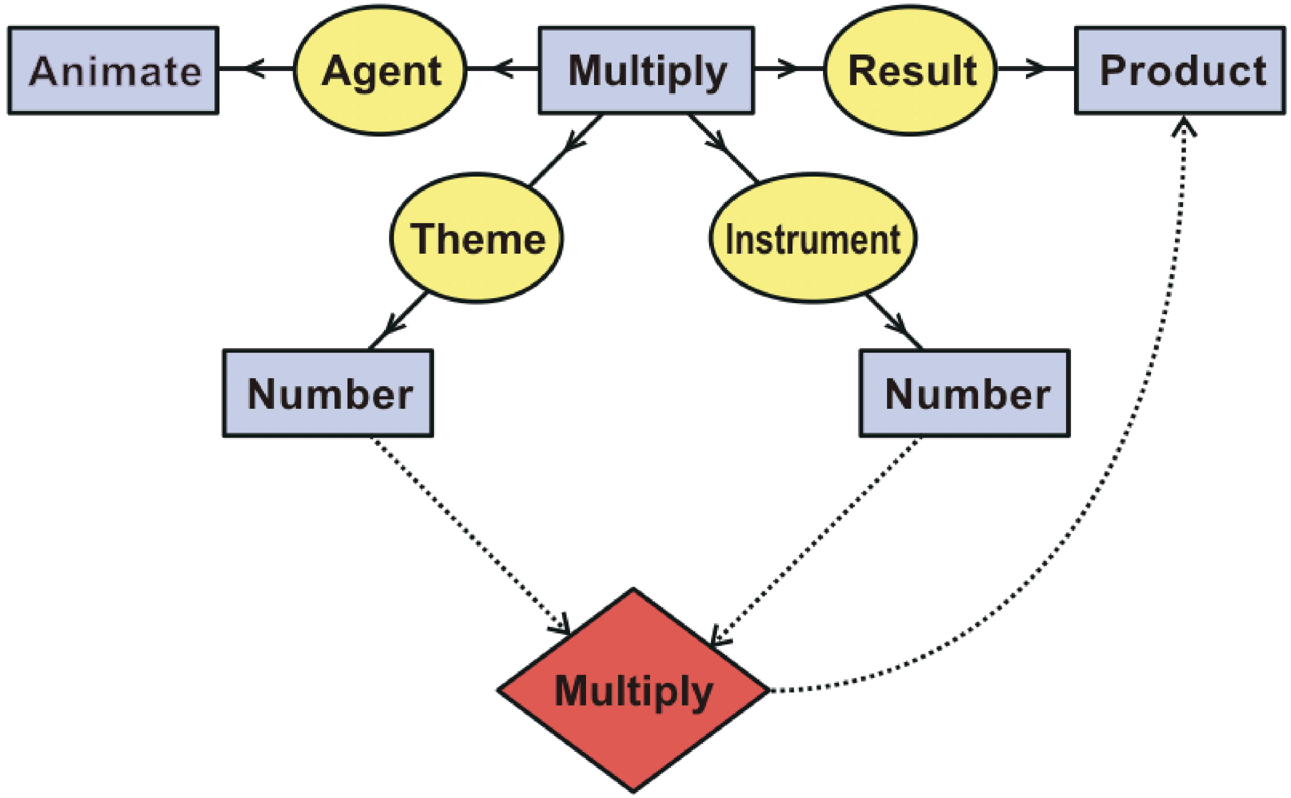
- Some are partially correct
  — and the teacher says what is missing.

- Others are wrong
  — in many different ways.

Result: 50 pairs of student answer and teacher’s response.

Each answer-response pair is a case for case-based reasoning.

Conceptual Graphs Relate English to Math

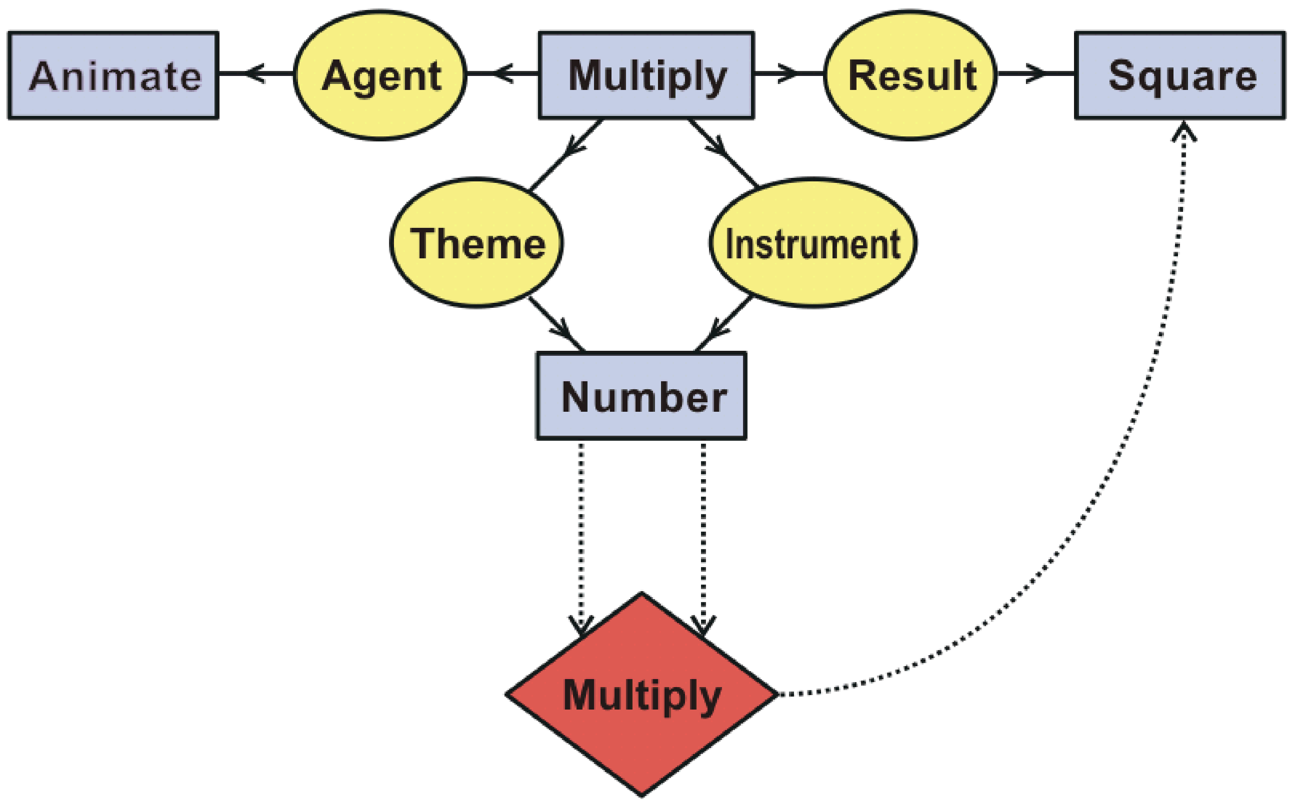


The boxes and circles represent an English sentence pattern:

[Someone] multiplies a number by a number to get a product.

The diamond node, called an actor, represents a function that computes the result of multiplying values inside the concept boxes.

Definition of the Word *square*



This conceptual graph is a specialization of the previous CG for the case when both inputs to the multiply actor are the same number.

This graph expresses the context that distinguishes the numerical square from the geometrical square.

Using VAE to Evaluate Student Answers

Use VAE to compare each new answer to the 50 previous cases:

1. For all 50 cases, translate student answer to conceptual graphs.
2. Translate each new answer to a CG and compute its Cognitive Signature™.
3. Compare the signature to the signatures of the 50 previous answers.
4. Use a semantic distance measure to determine the best match.
5. If there is a good match, print out the corresponding response.
6. Otherwise, send the new student answer to a teacher to evaluate.

Results:

- VAE found a good match for nearly all of the student answers.
- For each good match, the previous teacher's response was appropriate.
- When VAE failed to find a good match, the new case could be added to the list of cases in order to improve its coverage.
- No need for teachers to use any language other than English.

Application to Oil and Gas Exploration

The VivoMind Language Processor learns by reading:

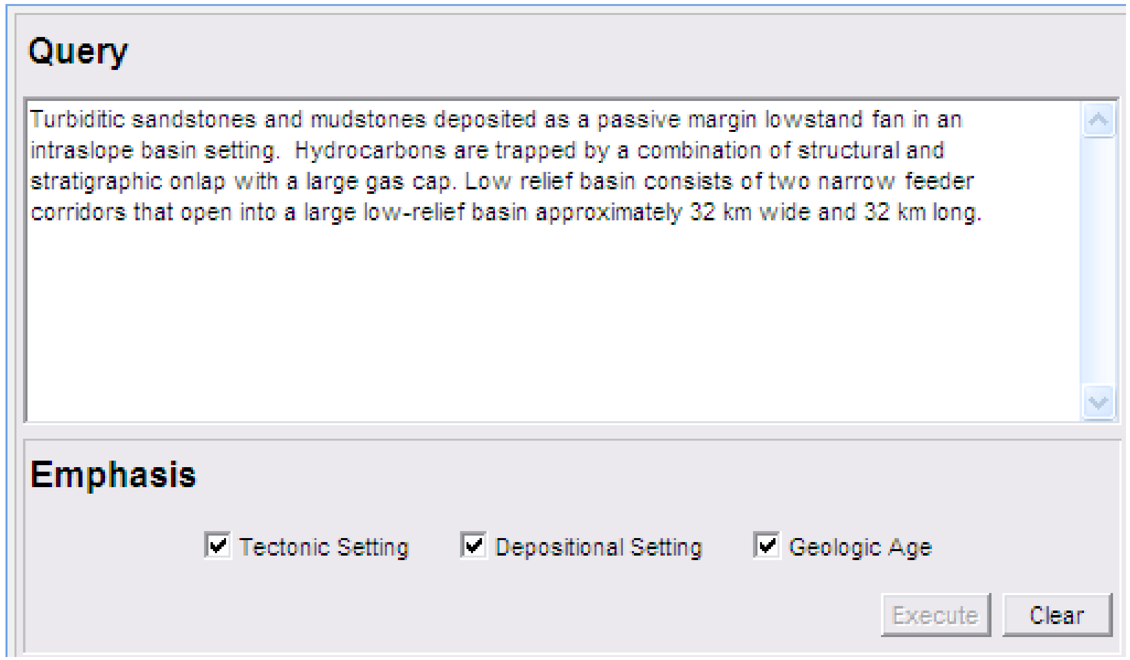
- 79 documents about the geology of oil and gas fields.
- English, as written for human readers (no semantic tagging).
- Additional data from relational DBs and other structured sources.
- Basic VivoMind ontology plus a domain-dependent ontology written in controlled English by geologists at the University of Utah.
- Very few detailed axioms in the ontology.

After reading, VLP answers questions by geologists:

- Input: Description of a geological site in unrestricted English,
- Query: Find, compare, and rank all similar sites in the documents.

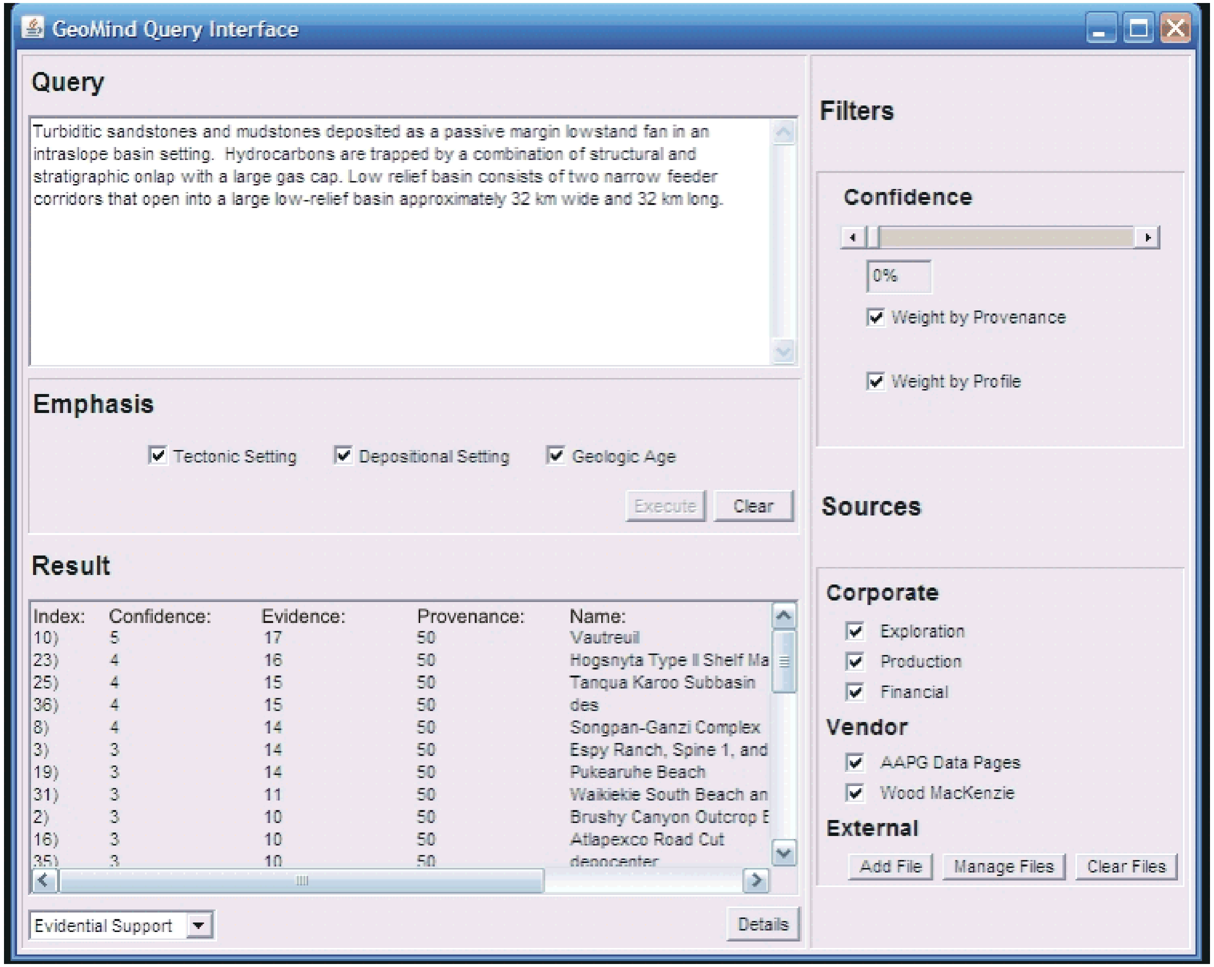
Note: Controlled English is used to define the ontology, but unrestricted English is used for the documents and queries.

A Query Written by a Geologist

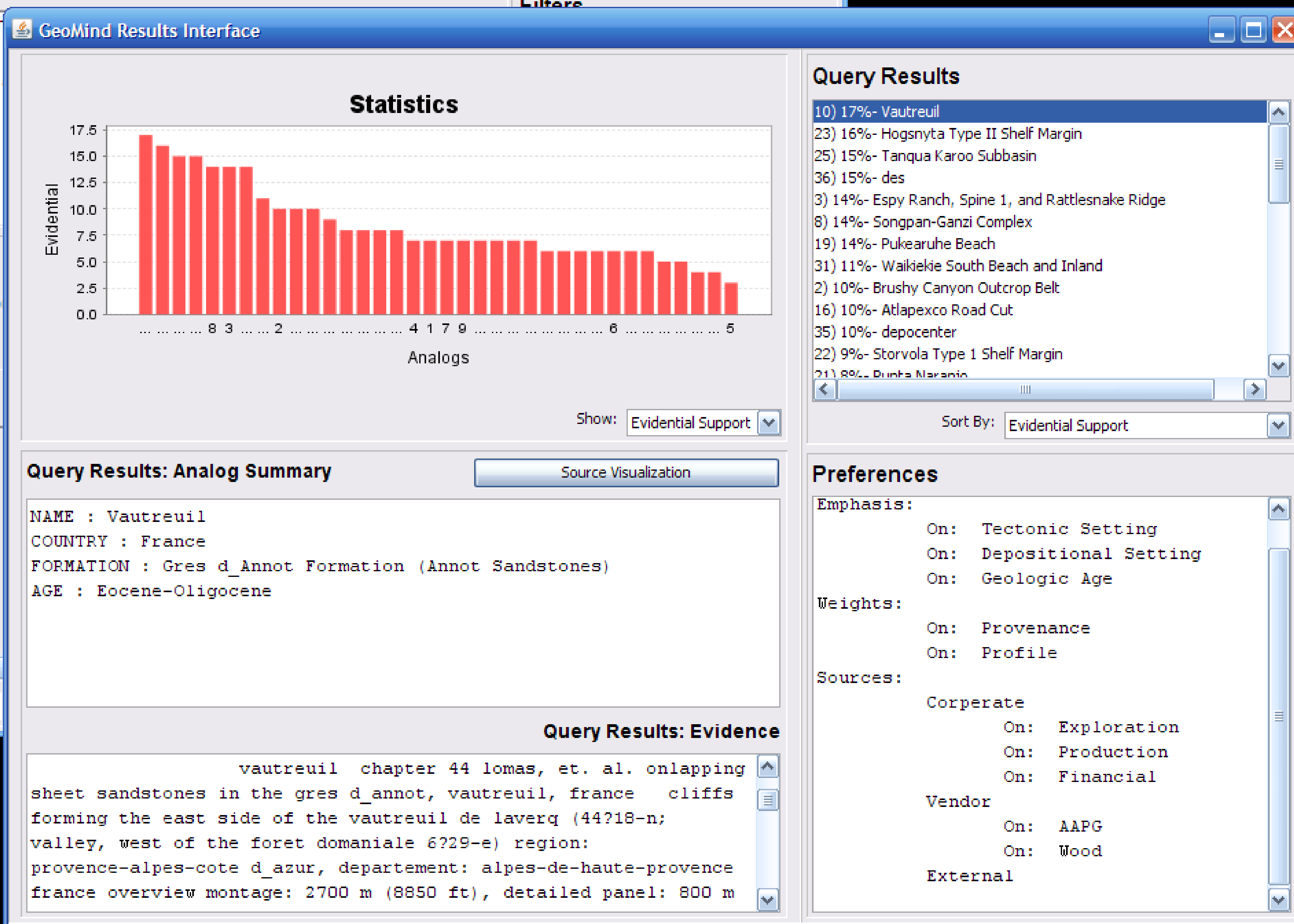


Turbiditic sandstones and mudstones deposited as a passive margin lowstand fan in an intraslope basin setting. Hydrocarbons are trapped by a combination of structural and stratigraphic onlap with a large gas cap. Low relief basin consists of two narrow feeder corridors that open into a large low-relief basin approximately 32 km wide and 32 km long.

Similar Sites Found in the Documents



Sites are ranked by evidence (Dempster-Shafer) and confidence factors.



After clicking the “Details” button on the previous window

Using Multiple Knowledge Sources

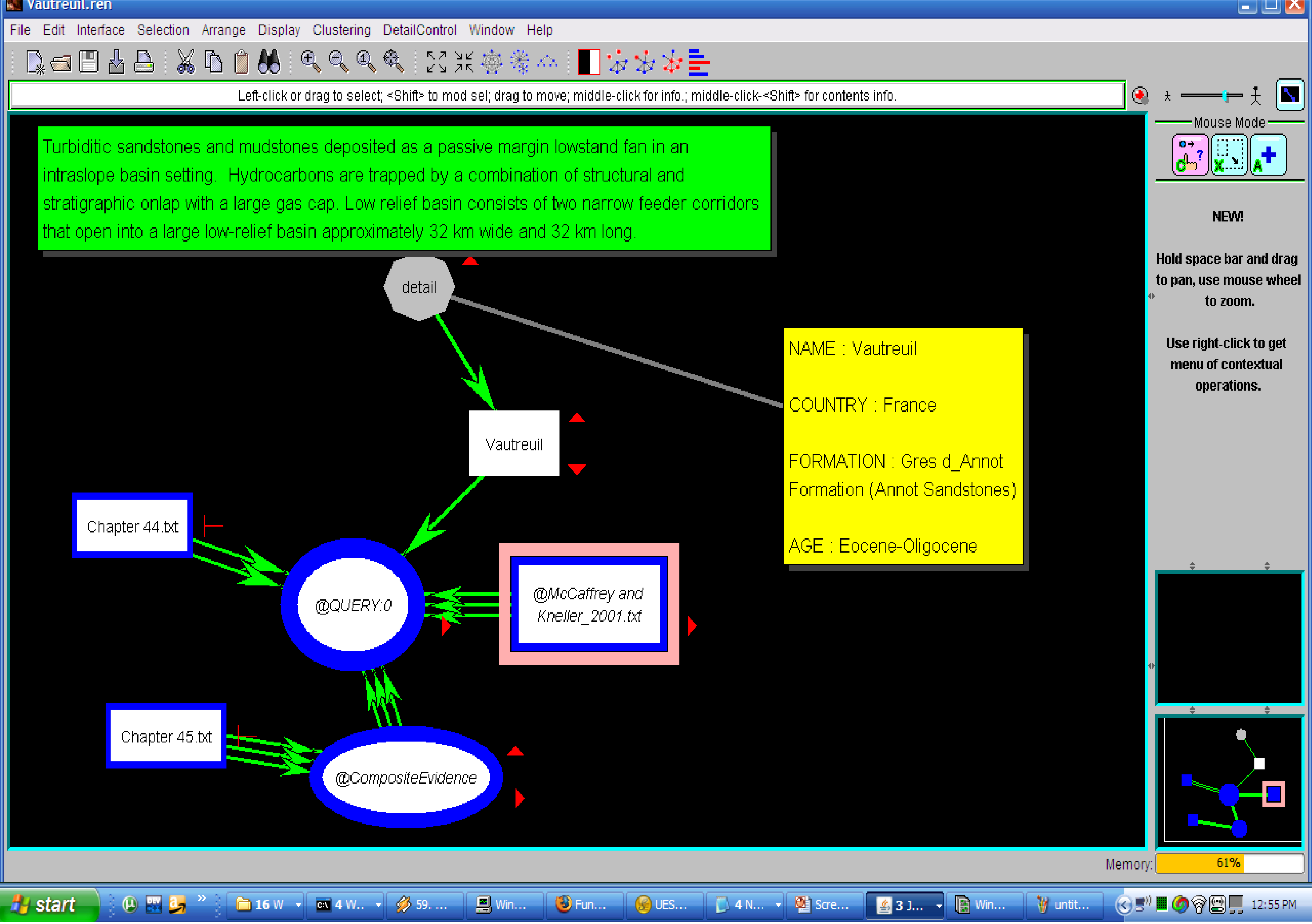
The geologists who wrote an ontology in controlled English included very little domain knowledge.

But that simple ontology was sufficient for the system to acquire more knowledge from a textbook of geology.

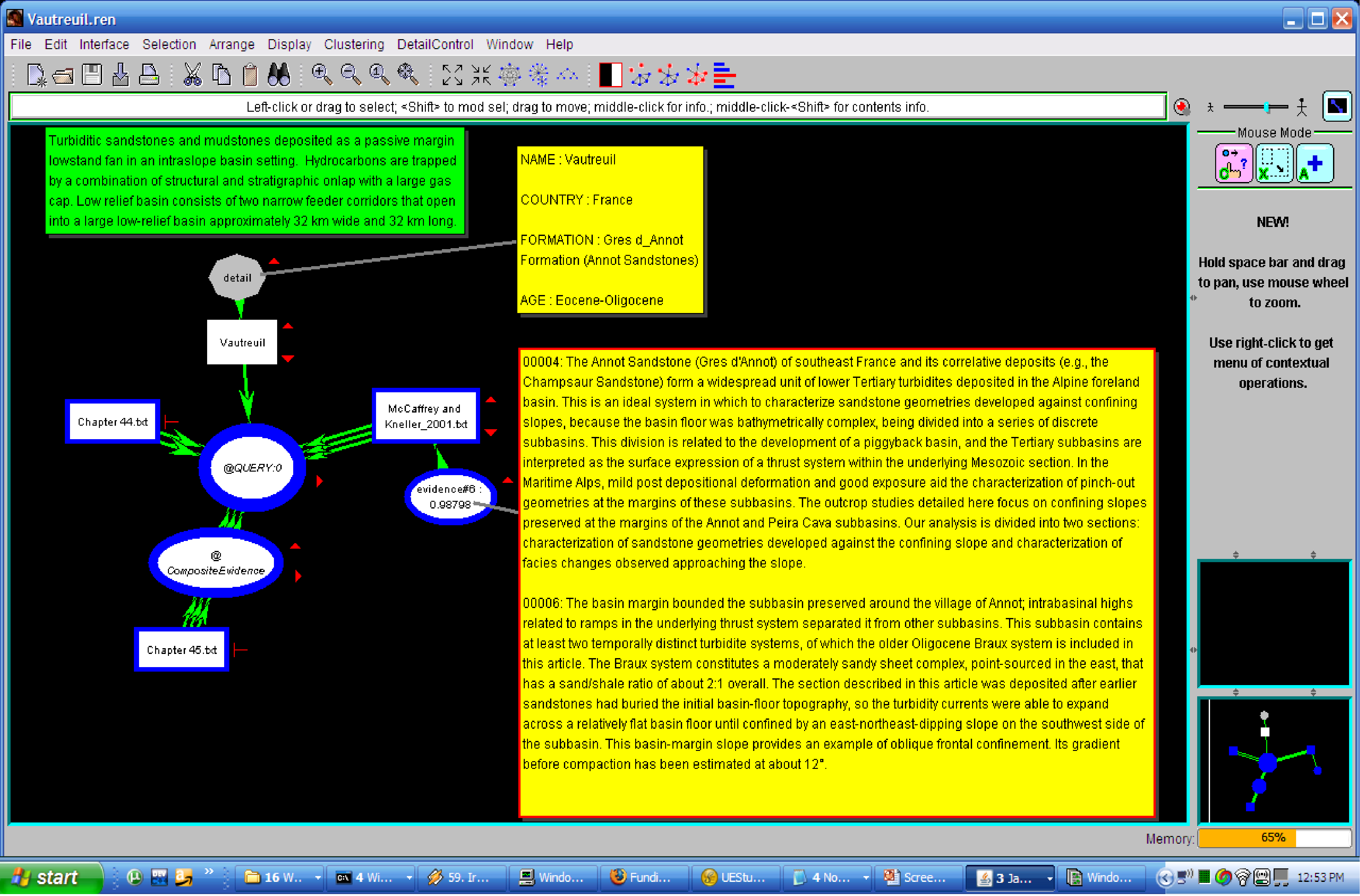
The following screen shot shows that the system accessed parts of four documents to respond to this query:

- A description of an oil field in the Vautreil region of France,
- Chapters 44 and 45 from a textbook on geology,
- A research paper by McCaffrey and Kneller (2001).

But the system also used other documents to evaluate the less highly ranked sites.



Top-level source visualization and the highest-ranked result



Show which paragraphs of a document were used for this query

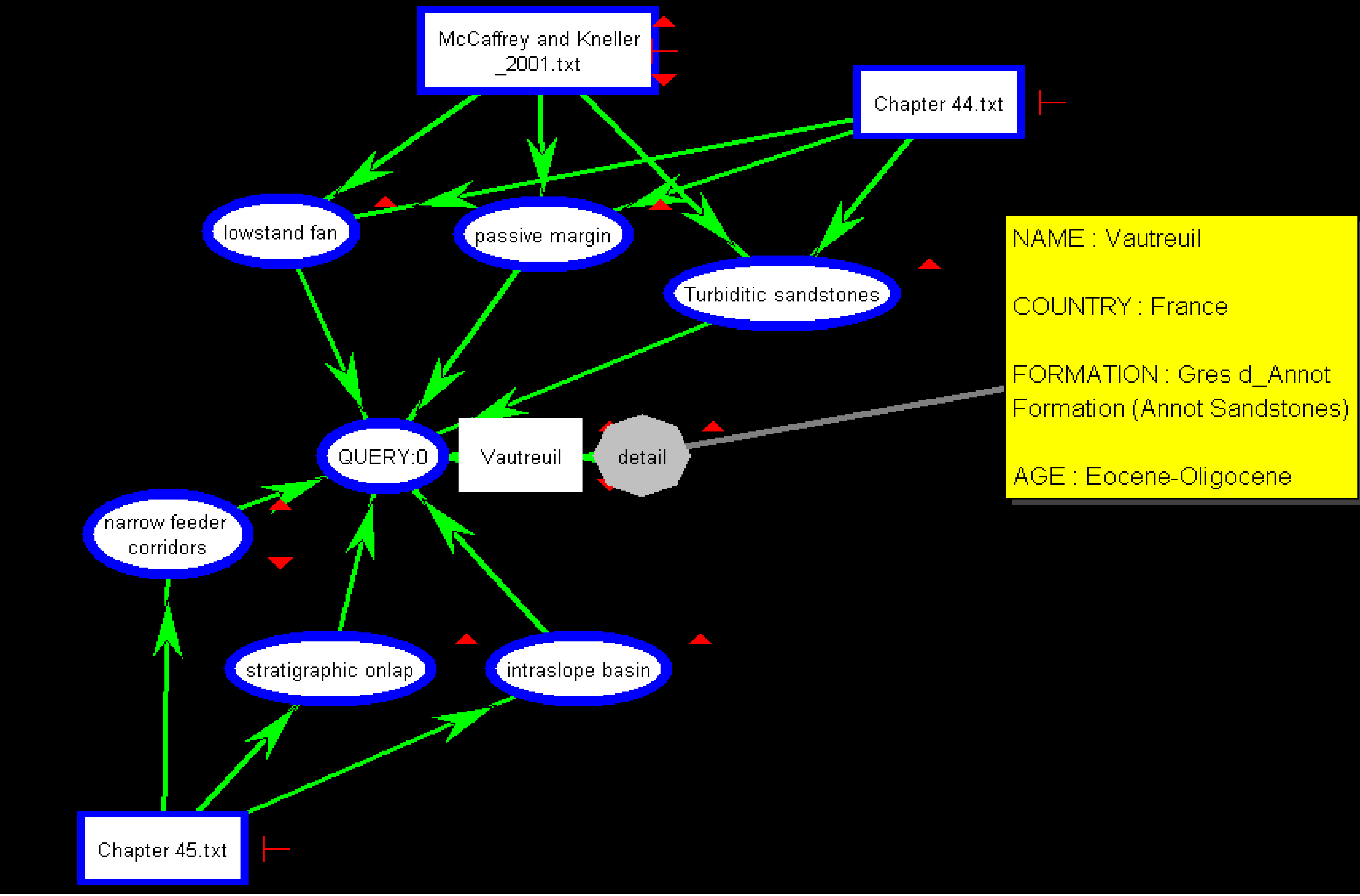
Showing Details of the Analysis

Before answering questions, the system translates all sentences of the source documents to conceptual graphs and indexes them by their Cognitive Signatures™.

Conceptual graphs derived from the query may trigger searches for CGs that provide background knowledge.

The next screen expands the previous screen to show which phrases of the query required additional knowledge.

For each document that was accessed, the geologist can ask for the specific sentences or paragraphs that were analyzed to generate the answer.



Drill down into the query and its relationships to the source documents

Emergent Knowledge

When reading the 79 documents,

- VLP translates the sentences and paragraphs to CGs.
- But it does not do any further analysis of the documents.

When a geologist asks a question,

- The VivoMind system may find related phrases in many sources.
- To relate those phrases, it may need to do further searches.
- The result is a conceptual graph that relates the question to multiple passages in multiple sources.
- Some of those sources might contribute information that does not have any words that came from the original question.
- That new CG can be used to answer further questions.

By a “Socratic” dialog, the geologist can lead the system to explore novel paths and discover unexpected patterns.

Kinds of Reasoning

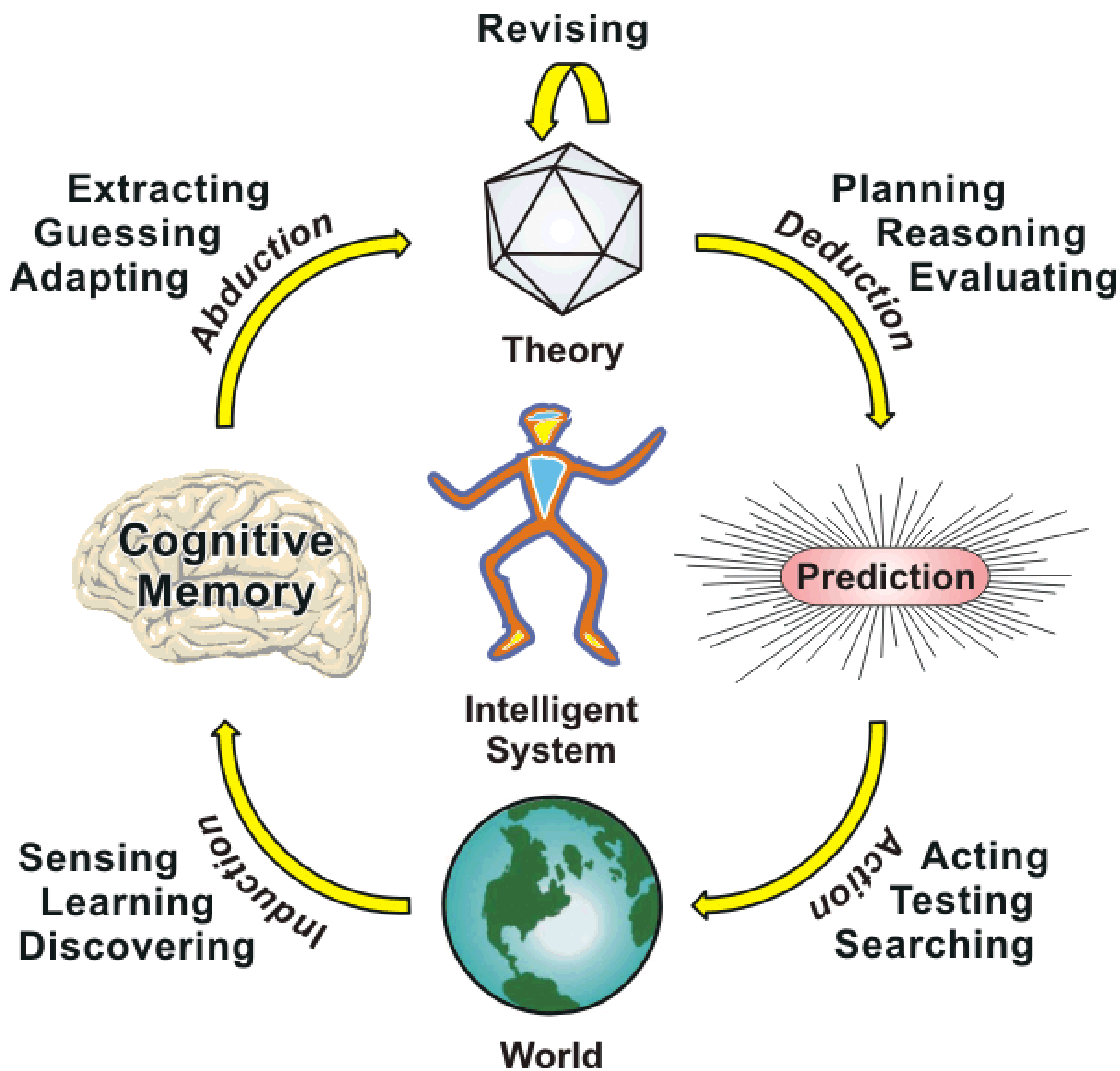
By indexing Cognitive Signatures™, the system can find relevant graphs from any source in logarithmic time.

Those graphs can be processed by multiple paradigms:

- **Analogies for language understanding and case-based reasoning,**
- **Logical reasoning by deduction, induction, or abduction,**
- **Various methods of probability and statistics,**
- **Any method of heuristics or fuzzy reasoning.**

The next slide groups various paradigms according to Peirce’s cycle of pragmatism.

Cycle of Reasoning



An open-ended range of paradigms for every aspect of reasoning.

Suggested Readings

The paper with the same title as this talk:
http://www.jfsowa.com/pubs/paradigm.pdf

A description of the VivoMind Analogy Engine:
http://www.jfsowa.com/pubs/analog.htm

The Flexible Modular Framework used to implement the VivoMind software:
http://www.jfsowa.com/pubs/arch.htm

A discussion of the issues involved in understanding natural languages:
http://www.jfsowa.com/puts/pursuing.pdf

The “Challenge of Knowledge Soup” for any approach to general AI:
http://www.jfsowa.com/pubs/challenge.pdf

A 22-page overview of conceptual graphs and the Common Logic standard:
http://www.jfsowa.com/cg/cg_hbook.pdf

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