Chunks & Rules*

Dave Raggett <dsr@w3.org>

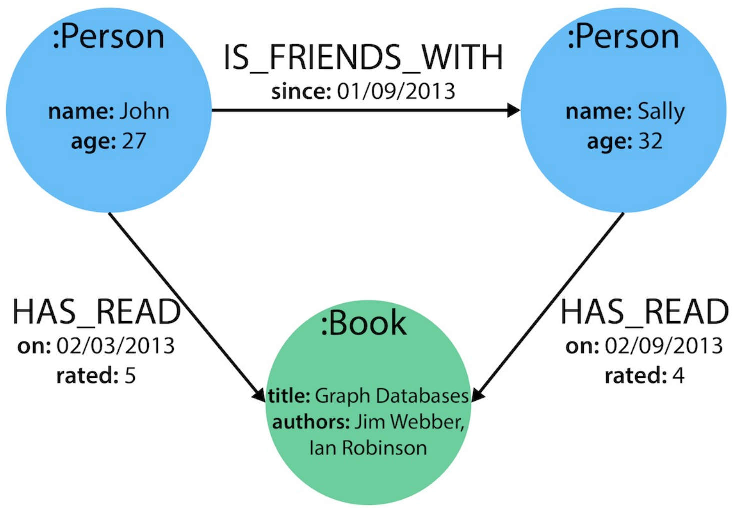
5 December 2019

Based upon the Cognitive Sciences
(the scientific study of human mind and behaviour)

* see: https://www.w3.org/Data/demos/chunks/chunks.html

Resource Description Framework *compared to* Labelled Property Graphs

- Resource Description Framework (RDF)
  - Popularised in 2001 Scientific American article by Tim Berners-Lee, Jim Hendler & Ora Lassila
  - RDF reduces semantic networks to constituent triples
  - Ontologies describe how triples form larger structures
  - URIs for dereferenceable globally unique names
  - Queries using SPARQL analogous to SQL for RDBMS
  - Reification as means for annotating triples
  - Semantic Web – formal semantics and logical deduction
  - Allegedly difficult for average developers
- Labelled Property Graphs (LPG)
  - Nodes & Relationships can both have a set of key-value pairs
  - Queries using Cypher, GQL, Gremlin and other languages
  - Informal semantics in terms of graph traversal
  - Rapidly growing commercial popularity compared to RDF

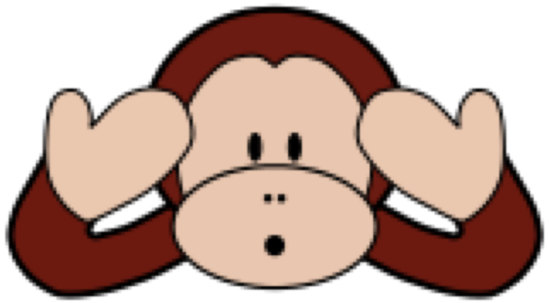


Example courtesy of Neo4J

Are we stuck within walls of our own devising?

Don’t speak about my discipline’s taboos

Don’t look at work outside of my discipline



Don’t listen to people from other disciplines

Has science descended into isolated ghettos?

Time to embrace a new paradigm

- The real world is frustratingly uncertain, incomplete and inconsistent
- Cold logic needs to be complemented with emotion and intuition
- More explicitly – we need to blend symbolic and statistical approaches
- This will allow us to create agents that learn and reason based upon prior knowledge and past experience
- This has been studied for many decades in Cognitive Science!

Captain James Kirk and Science Officer Spock in StarTrek

Chunks

- Chunks are a common concept in the Cognitive Sciences
  - First introduced by Miller in 1956 to account for work on human memory

"A chunk is a collection of basic familiar units that have been grouped together and stored in a person's memory. These chunks are able to be retrieved more easily due to their coherent familiarity" (Wikipedia)

"Researchers in cognitive science have established chunking as one of the key mechanisms of human cognition, and have shown how chunks link the external environment and internal cognitive processes" (Chunking mechanisms in human learning, Gobet et al.)

"Chunks correspond to concurrent stochastic spiking patterns across bundles of nerve fibres. You can think of this in terms of vectors in noisy spaces with a large number of dimensions. The set of name/value pairs in a chunk is represented by the projection of the vector onto orthogonal axes" (Concepts as semantic pointers, Eliasmith)

Chunks

For this work, a chunk is modelled as a concept with a set of properties. Each chunk has a type and an identifier. Chunk property values are either booleans, numbers, names, string literals enclosed in double quote marks, or a comma separated list thereof. Here is an example:

```
friend f34 {
  name Joan
}
friend {
  name Jenny
  likes f34
}
```

- Where *friend* is a chunk type, *f34* a chunk identifier, *name* and *likes* are property names, *Joan* and *Jenny* are also names.
- *likes f34* signifies that Jenny likes Joan via the link to the chunk for Joan.
- Missing chunk identifiers are automatically assigned when inserting a chunk into a graph
- Uses line breaks as punctuation

Links are a subclass of Chunk

- The chunk type is the predicate, and the link is described by chunk properties *subject* and *object*

```
dog kindof  mammal
cat  kindof  mammal
```

Is equivalent to

```
kindof {
  subject dog
  object mammal
}
kindof {
  subject cat
  object mammal
}
```

It is trivial to annotate links by adding further properties

Integration with RDF

- Use @rdfmap to map names to RDF URIs

```
@rdfmap {
  dog http://example.com/ns/dog
  cat http://example.com/ns/cat
}
```

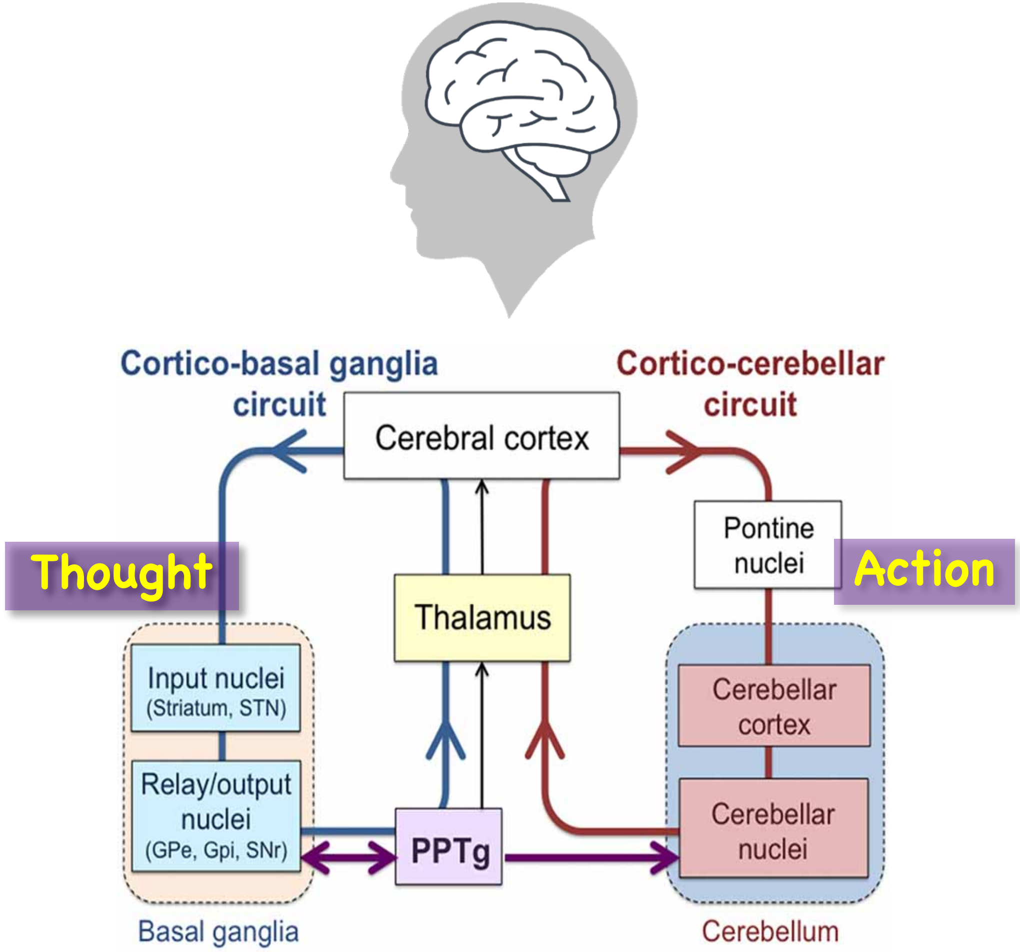
- Use @prefix to declare URI prefixes

```
@prefix p1 {
  ex: http://example.com/ns/
}
@rdfmap {
  @prefix p1
  dog ex:dog
  cat ex:cat
}
```

- Use @rdfmap to link to externally defined mappings

```
@rdfmap from http://example.org/mappings
```

Rules as basis for human consciousness



With thanks to [Fumika Mori et al.](#)

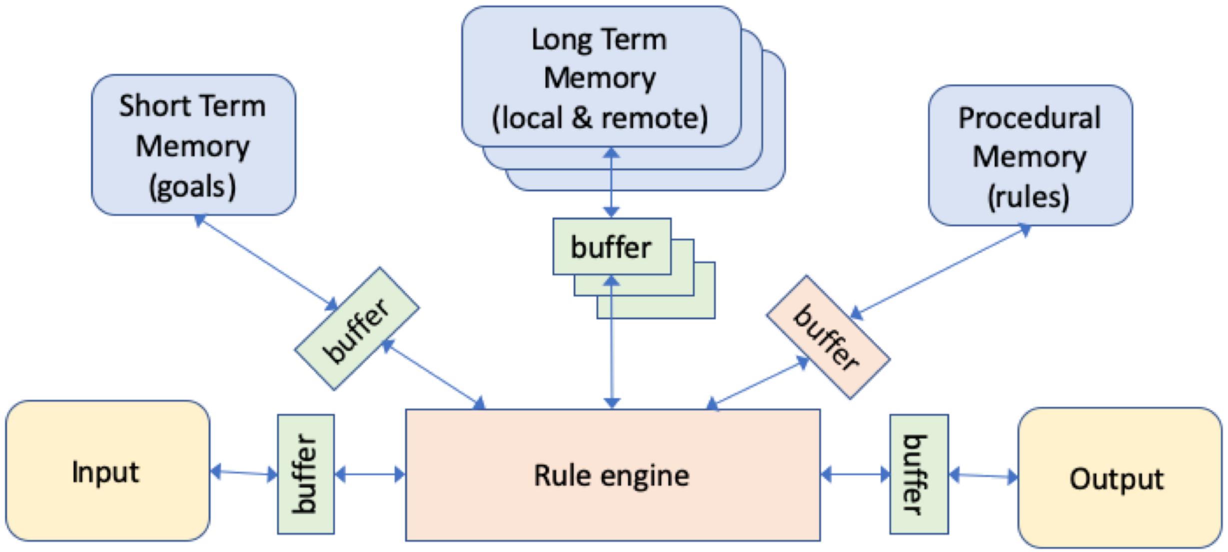
- Cortico-basal ganglia circuit
  - Cerebral cortex functionally equivalent to a set of graph databases serving different purposes
  - Basic ganglia and thalamus functionally equivalent to a rule engine
  - Rules are executed sequentially approximately every 50mS
  - The seat of human consciousness!
- Cortico-cerebellar circuit supports actions initiated by conscious control
  - Cerebellum analogous to flight controller coordinating muscle activation according to sensory data
  - Talking, walking, playing the piano, …

Cognitive Agents

- Chunks manipulation via graph API
  - Implemented as JavaScript library
  - Used to implement graph algorithms
- Chunks can also be used with goal-directed rules, inspired by ACT-R
  - popular cognitive science architecture
- Rule conditions match module buffers which hold a single chunk
- Rule actions update the buffers, either directly, or indirectly via invoking module graph algorithms
- Chunk retrieval is stochastic reflecting prior knowledge and past experience

Cognitive Agent Architecture

Modelled on what we know about the brain

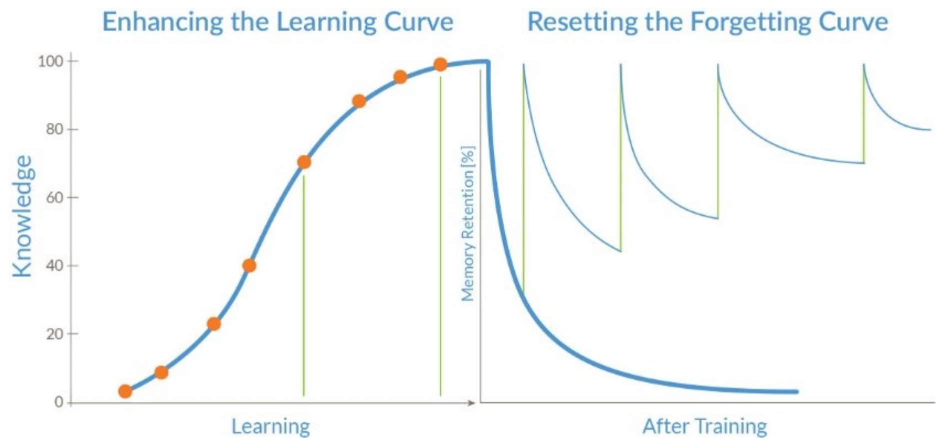


Module buffers hold a single chunk, Rule buffer holds a single rule

ACT-R's buffers are related to Baars' global workspace theory (GWT) where attention acts as as a spotlight of awareness moving across a vast space of unconscious (thus hidden) processes. The brain is richly connected locally, and weakly remotely. The buffers correspond to the constrained communication capacity for such long range communication.

Stochastic Chunk Retrieval

- Probability of chunk retrieval depends on expected utility based upon prior knowledge and past experience
- Ebbinghaus on memory (1885)
  - Learning: successive repetitions have progressively less effect*
  - Forgetting: successful recall drops off exponentially over time
- Each chunk is associated with an activation level that exponentially decays over time, and is boosted on every recall or update
  - This is computed using log values for greater efficiency
  - Retrieval is subject to a minimum activation level threshold
- Spreading activation models how related memories boost each other
  - Activation is spread evenly through links between chunks
  - The more links from a given chunk, the weaker its effect on linked chunks
  - This process continues recursively until some cut off threshold
- This is a simplification of the more complex model in ACT-R, see Said et al.



- Underwood (1957) showed that memory loss is mostly due to interference with other memories
- Refined model of chunk retrieval
  - Persistent chunk strengths as accretion from history of changing activation levels
  - Probability of chunk retrieval depends on combination of persistent strength and activation level
  - Should persistence be associated with chunks or links between chunks?

* The lower the time difference between repetitions, the less the effect

Chunk Rules

- Rules are expressed as a set of chunks
- "@" prefix denotes special names for rule interpreter
- "?" Introduces named variables, scoped to the rule
- Rule chunk
  - *@condition* names the conditions
  - *@action* names the actions
- @module names the module for conditions and actions
  - e.g. *@module goal*
- Action chunks
  - *@invoke recall* initiates chunk query on module's graph
  - *@invoke remember* saves chunk to the module's graph
  - Default action is to directly update the module's buffer
- Additional features, e.g. *@kindof, @isa, @id, @type, @distinct, @lteq*
- Tasks are associated with sets of rules
  - Rules that initiate, progress or complete the task
- Conflicts resolved using expected execution times
  - Estimated via reinforcement learning
  - Back propagation of task reward/penalty
  - Rule sets are abandoned if they take too long
- Rules can be compiled from declarative memory

```
rule r1 {
  @condition g1
  @action a1, a2, a3
}
count g1 {
  @module goal
  start ?num
  state start
}
count a1 {
  @module goal
  state counting
}
increment a2 {
  @module facts
  @invoke recall
  first ?num
}
increment a3 {
  @module output
  value ?num
}
```

Initial rule for counting task

Rule has 1 goal and 3 actions

Condition that matches goal buffer chunk with state = start, and binds ?num variable to the value of start property

Action that initiates counting task by setting the value of state to counting

Action that requests an increment chunk with given value for first property

Action that updates the output module buffer

Cognitive Databases

- Cognitive databases have the potential to store vast amounts of information similar to the human cortex
- Memory retrieval fits Web architecture
  - Remote invocation of graph algorithms in request/response pattern rather like HTTP
  - Analogous to Web search engines where results are computed based upon what is likely to be most relevant to the user – impractical and inappropriate to try to return complete set of matches
- A cognitive agent can access multiple cognitive databases located across the enterprise and the public Internet
- A single cognitive database can be shared with many cognitive agents
  - Some information is available to all agents, other information is restricted to given groups of agents
- Cognitive databases can support a wide variety of graph algorithms, e.g.
  - Retrieval of a single chunk with a given chunk identifier
  - Simple queries for chunks with matching types and properties
  - Queries for sub-graphs matching patterns, similar to SPARQL
  - Queries for sub-graphs based upon graph traversal automata
  - Spreading activation models for word sense disambiguation
  - Queries based upon structural similarities for analogical reasoning
  - Covariance analysis for statistical significance across a dataset
  - Other algorithms for data mining in big data
  - Spatial and temporal operations over indexed graphs, including A* and related search algorithms

Further topics

- Many kinds of reasoning, e.g.
  - Deductive, inductive, abductive, analogical, causal, temporal, spatial, social, emotional, understanding stories
- Learning knowledge graphs from noisy examples
  - Relationship to episodic memories
  - Covariance analysis
  - Strong, weak and unsupervised
- Reinforcement learning of rules for tasks
  - Progressive stages in learning
  - Affordances for heuristics
  - Stochastic cooling
- Task management, attention and reasoning at multiple levels
- Context chains and reasoning from multiple perspectives
  - Planning, imagining, lessons, stories
- Theory of mind and social interaction
  - Autobiographical memory
  - How infants learn from others
- Emotions and system 1 vs system 2
  - Emotion valence and arousal
  - Fast evaluation vs slow deliberation
  - Relationship to expectations of reward/penalty for tasks
- Natural language processing
  - Non-verbal communication
  - Learning new words and meanings

Closing Thoughts

- With a new paradigm, it is important to show benefits over existing paradigms, e.g. the Semantic Web with its focus on logic
- This essentially means creating a series of demos, which will take time
  - Currently working on a task management demo for autonomous vehicles
  - Next demo will address natural language dialogues and situational plans
  - Further demos on machine learning, different kinds of reasoning, emotions & theory of mind
- As this proceeds, I want to launch a W3C Community Group on Cognitive AI to help with use cases, demos, open source software and scaling experiments
- Manual development of applications and vocabularies won't scale, so we need to show how to use cognitive agents as collaborative assistants that we can teach, and which can learn by themselves
- Application to customer service agents embedded in web pages
- Application to enabling non-programmers to work with data using graphical user interfaces together with spoken dialogues
- Application to implementing smart factories with multiple agents
- The role of creativity and consciousness in cognitive agents
- For more details, see: https://www.w3.org/Data/demos/chunks/chunks.html