ERCIM
European Research Consortium for Informatics and Mathematics

Emergence of the Sentient Web and the revolutionary impact of Cognitive AI

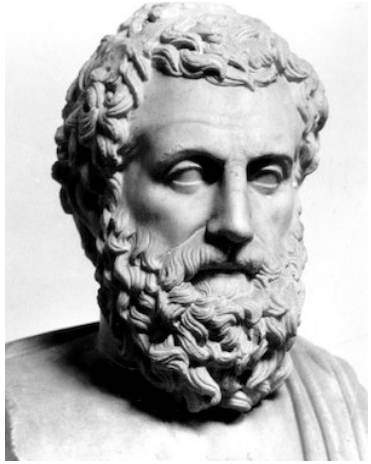
Dave Raggett, W3C/ERCIM

31 July 2020

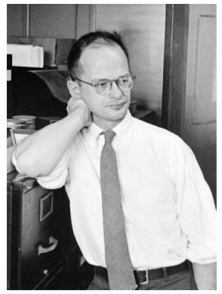
AI Technologies for Trust, Interoperability and Autonomy in Industry 4.0

Philosophical Divide

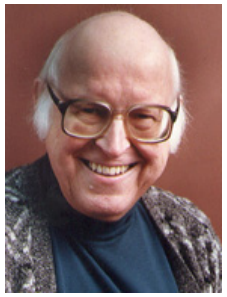
Whither semantic technologies?



Warren Sturgis McCulloch
(1898 – 1969)



Walter Harry Pitts, Jr.
(1923 – 1969)



Allen Newell
(1927 - 1992

John Anderson
(1947 -)

- Aristotelian tradition of logic and the role of formal semantics and model theory
- Predicate calculus and first order logic
- The Semantic Web and RDF
- *What is provably true given assumptions and inference rules; intolerant of inconsistencies*

- Cognitive Science as the experimental study of the organizing principles of the mind
- Cognitive AI – mimicking human reasoning with combination of graphs, statistics, rules and graph algorithms, inspired by evolution
- *What is useful based on prior knowledge and past experience in the face of uncertainty and inconsistency*

Two radically different mindsets with strong implications for the future of computing


Change is Coming

Sentient Web

- **Sentient Web = sensing + actuation + cognition** federated across the Web to enable markets of services based upon open standards
  - *Sentient* used in the sense of systems that are aware of their environment
  - *Cognition* is the process of reasoning, learning and control, or in other words, the means for systems to represent, process, and transform information
  - *Federated* in the sense of a Web of cognitive agents and cognitive databases
- The Sentient Web will subsume both the IoT and AI
- A uniform approach to cognition will replace today's fragmented IoT
- Today's relational and graph databases will give way to cognitive databases that mimic the cortex
- AI and machine learning will be ubiquitous
- Change is coming, what role will *you* play?

Deep Learning Limitations

- Deep Learning has been very successful but its limitations are increasingly obvious – Cognitive AI can complement Deep Learning as well as reducing the carbon footprint for machine learning!
  - MIT study on The Computational Limits of Deep Learning
  - Yuille & Liu: Limitations of Deep Learning for Vision, and How We Might Fix Them
  - Gary Marcus: Deep Learning a Critical Appraisal
    - "Deep learning must be supplemented by other techniques if we are to reach artificial general intelligence"
- Yoshua Bengio, a Deep Learning pioneer, acknowledges this:

  "We have machines that learn in a very narrow way," Bengio said in his keynote at NeurIPS in December 2019. "They need much more data to learn a task than human examples of intelligence, and they still make stupid mistakes."

  Without question, deep learning is an imperfect model of intelligence. It cannot reason abstractly, does not understand causation and struggles with out-of-distribution generalization.

From Forbes article on 9 February 2020

The Cambrian Explosion

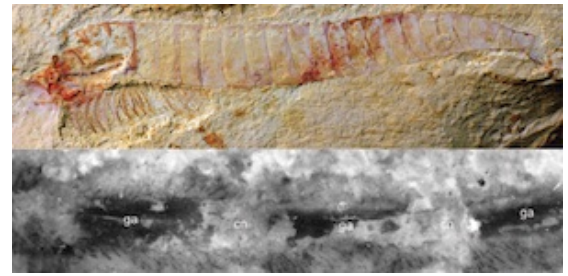
Left credit: Smithsonian Institution

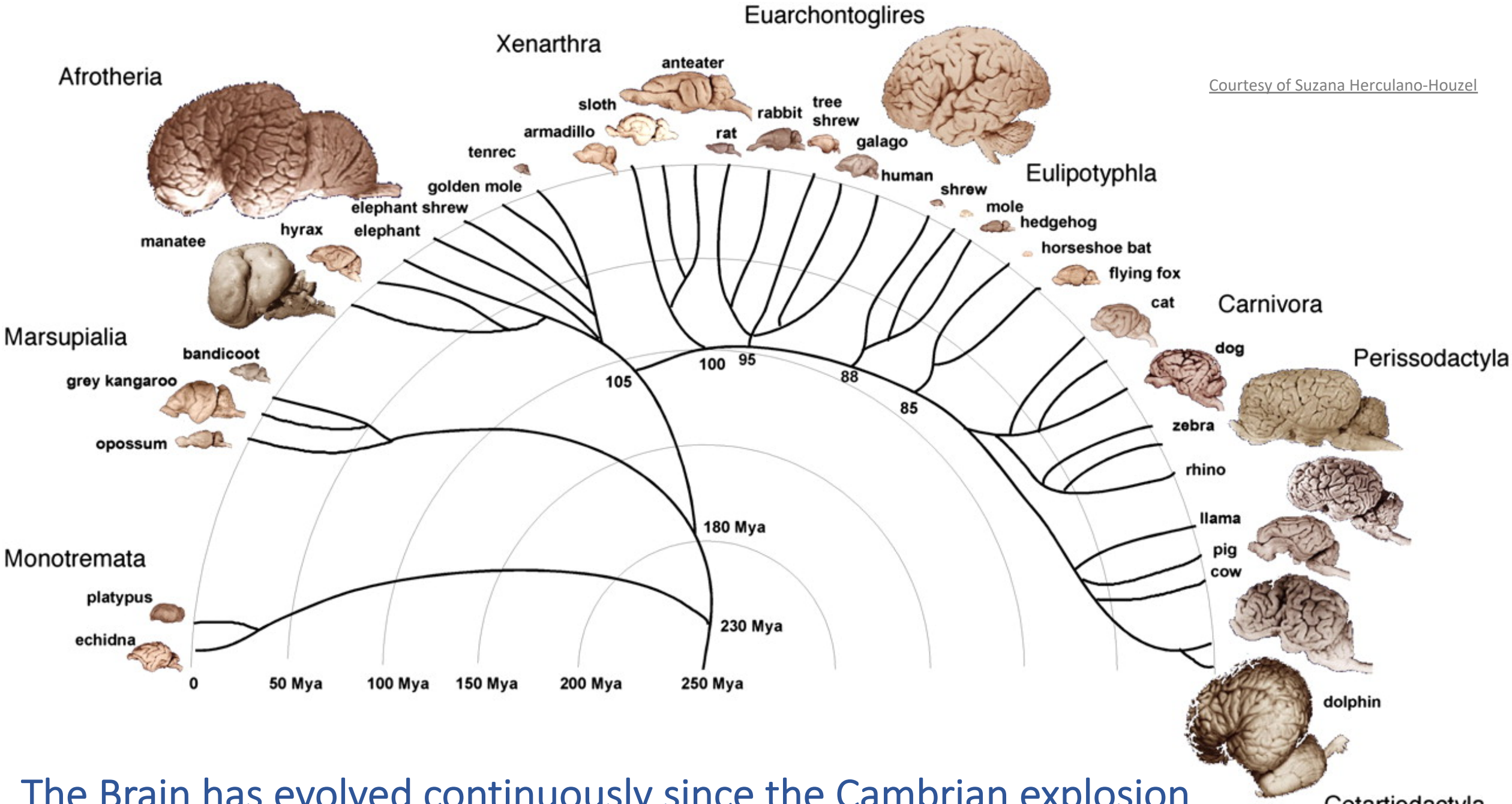
The Cambrian period occurred over 500 million years ago and included the biggest evolutionary explosion in Earth's history linked to higher concentrations of oxygen and warmer seas.

This included predators with the emergence of nervous systems for complex behaviour.

Bottom: 520 million year old fossilised ventral nerve cord of Chengjiangocaris kunmingensis.

Credit: Top: Jie Yang, Bottom: Yu Liu





Courtesy of Suzana Herculano-Houzel

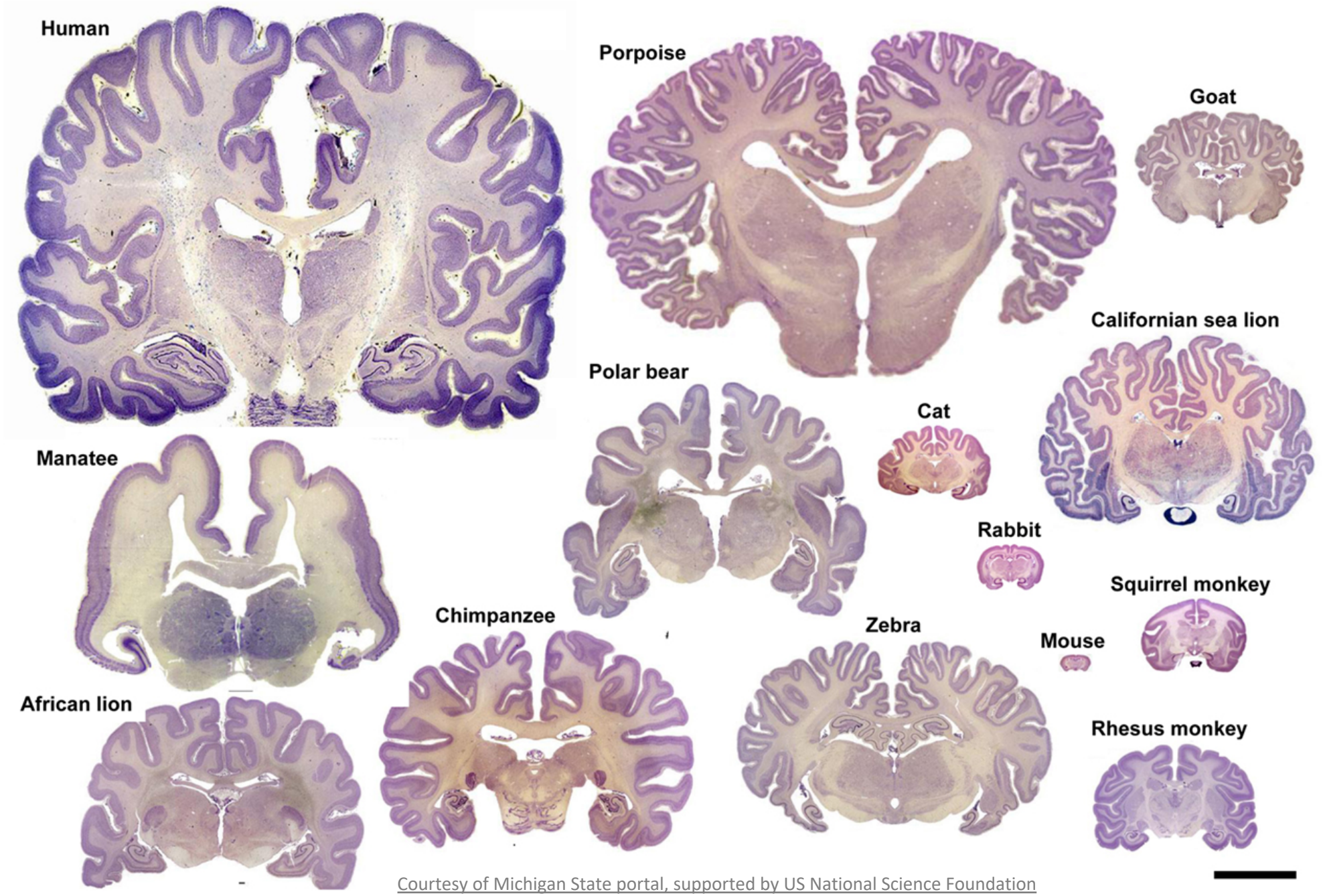
The Brain has evolved continuously since the Cambrian explosion

BRAIN SIZE AND NEURON COUNT

Cerebral cortex mass and neuron count for various mammals.

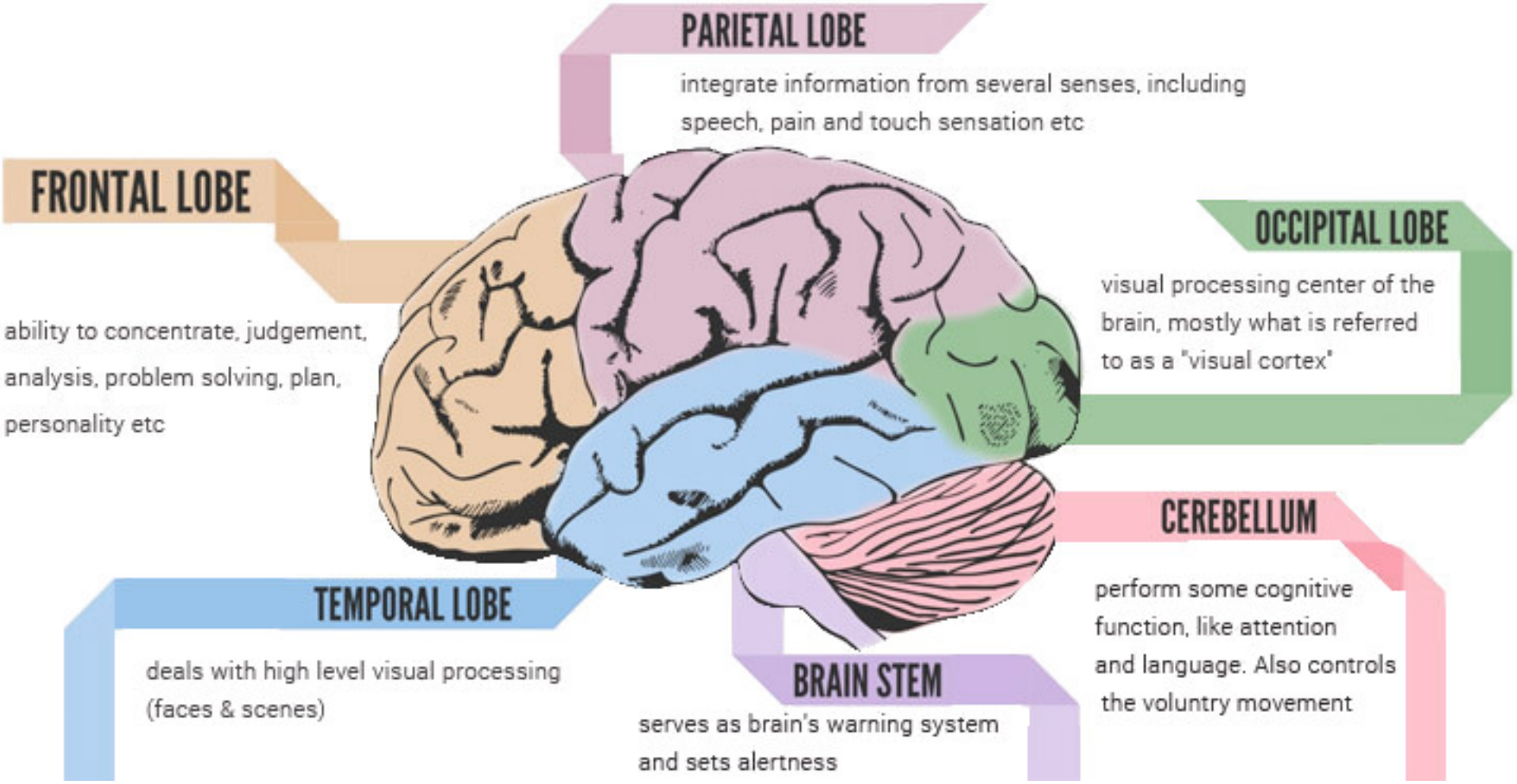
Courtesy of Quanta magazine

Capybara	Rhesus Macaque	Western Gorilla	Human	African Bush Elephant
non-primate	primate	primate	primate	non-primate
48.2 g	69.8 g	377 g	1232 g	2848 g
0.3 billion neurons	1.71 billion neurons	9.1 billion neurons	16.3 billion neurons	5.59 billion neurons



Courtesy of Michigan State portal, supported by US National Science Foundation

Brain function – many specialized areas



Co r e of H man brain fac

Cognitive AI

- In short, Cognitive AI is Artificial Intelligence inspired by advances in the cognitive sciences
- In other words, we would do well to borrow from nature when it comes to building AI systems
- We can mimic nature at a functional level using conventional computer technology without having to implement cognitive agents in terms of artificial neurons
- There are many potential applications of cognitive agents for human-machine collaboration

Long Term Aims

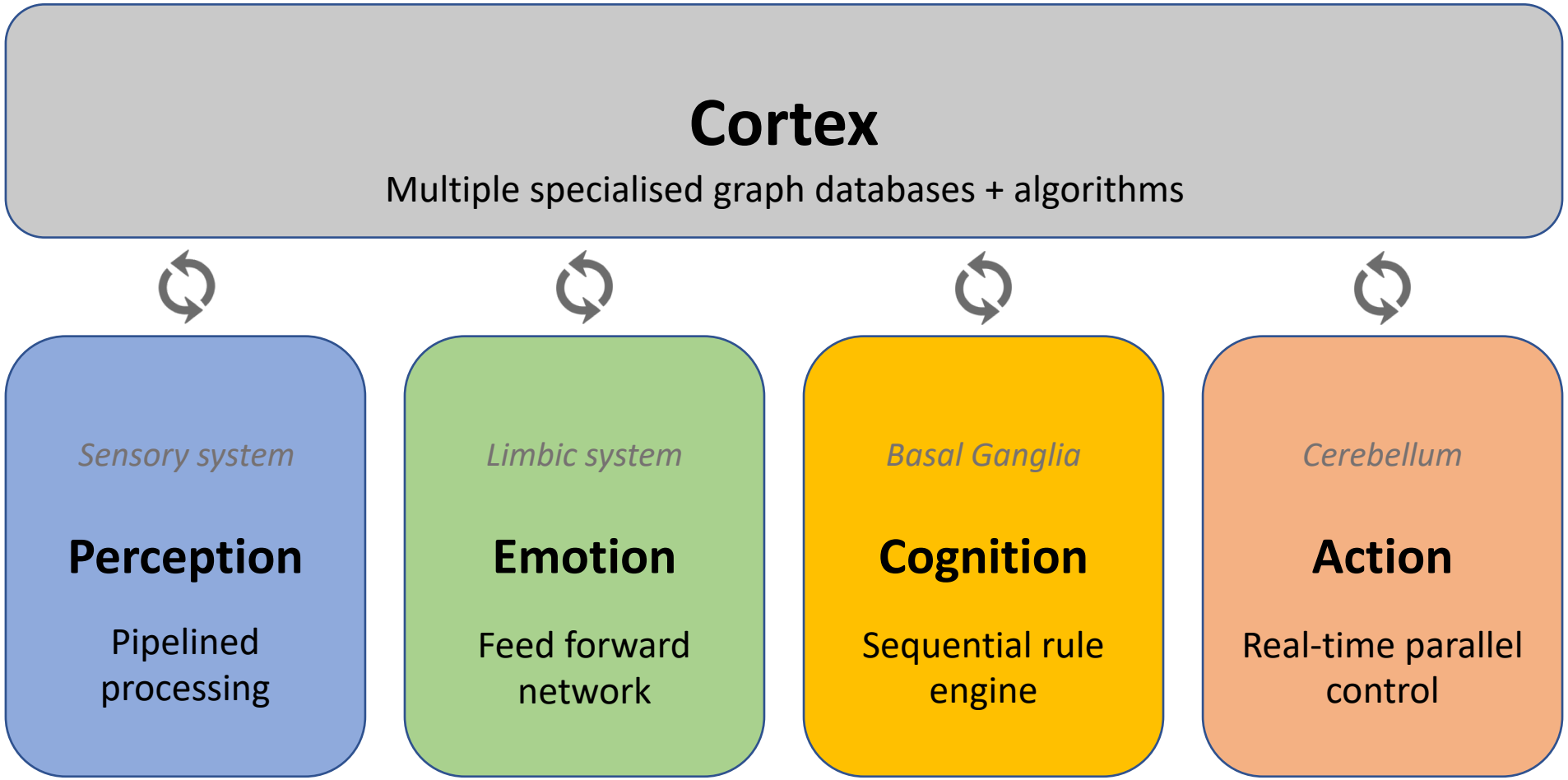
ERCIM
European Research Consortium for Informatics and Mathematics

To enable cognitive agents that:

1. **Are knowledgeable, general purpose, collaborative, empathic and trustworthy**
2. **Can apply metacognition and past experience to reason about new situations**
3. **Support continuous learning based upon curiosity about the unexpected**
4. **Have a level of self awareness in respect to current state, goals and actions**
5. **Have an awareness of others in respect to their beliefs, desires and intents**
6. **Are multilingual and can interact with people using their own language**

Cognitive AI Architecture

with multiple cognitive circuits



Cognitive Functions

- **Perception** involves interpreting sensor data in the current context, focusing attention on things of interest, and placing short lived representations in the cortex
- **Emotion** is about fast, intuitive assessments of the current situation and potential courses of action
- **Cognition** is slower and more deliberate thought, involving sequential execution of rules to carry out particular tasks
  - Thought can be expressed at many different levels of abstraction
- **Action** is about carrying out actions initiated under conscious control, leaving the mind free to work on other things
  - An example is playing a musical instrument where muscle memory is needed to control your finger placements as thinking explicitly about each finger would be far too slow

Modelling the Cortex with Cognitive Databases

ERCIM
European Research Consortium for Informatics and Mathematics

- The human cortex is functionally equivalent to a set of specialised cognitive databases and associated algorithms
- A cognitive database holds chunks: collections of properties that include references to other chunks
- Chunks are associated with statistical information reflecting prior knowledge and past experience
- Cognitive databases have the potential to store vast amounts of information similar to the human cortex
- **Cognitive databases** can be **local** or **remote**, and **shared** with multiple cognitive agents, subject to **access control** policies

- 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
- Cognitive databases support a variety of algorithms that are executed local to the data
  - Scalable to handling Big Data
  - Distributed queries across multiple cognitive databases to mimic operation of the anterior temporal lobe*
- The algorithms depend on the intended function of the database, e.g.
  - Basic storage and recall
  - Specialised algorithms for natural language, spatial and temporal reasoning
  - Algorithms for data analytics

* Sharon Thompson-Schill on a hub and spoke model for how the anterior temporal lobe integrates unimodal information from different cortical regions.

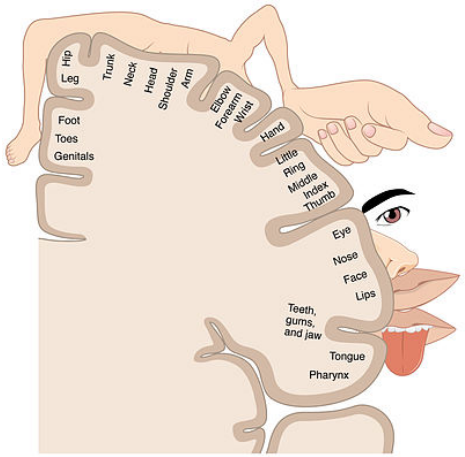
Sensory Perception

- Our senses
  - Smell, taste, touch, pain, heat, sound, vision, …
  - **Perception creates short lived representations in the cortex**
  - The cortex can likewise direct sensory processing as needed
- Touch and pain are mapped to a homuncular model of our bodies
- Proprioception – sense of self-movement and body position
  - Limbs, joints, muscle load
  - Vestibular system (inner ear)
- Sound is fleeting
  - Processing word by word
  - Emotional cues

- Vision is much more complex
  - Two eyes for stereo depth perception
  - Each eye: high resolution narrow angle + low resolution wide angle
  - Saccades as eyes swivel to scan areas of interest
  - Good at recognizing many different kinds of things, including their structures & behaviours
  - Context determines what is interesting and relevant
  - Alerts signal relevant things in field of view
  - Focus directs attention to specific things
  - Reinforcement learning from experience

Visual system

Hearing



Cortical homunculus

Implementation as pipelined neural networks

Emotions, Feelings and Moods

Towards strong empathic AI*

- **Cortico-Limbic system**
- Important from an evolutionary perspective
  - Avoidance of harm, fear of predators, interest in prey, courtship, care of young
- Enhanced for living in social groups
  - Emotional intelligence – awareness of what others are feeling, and signalling your own feelings
- Emotions are associated with a feeling and something they apply to
  - Valence describes whether feeling is positive, neutral or negative
  - Arousal describes whether feeling is calming or exciting
  - Moods are long lasting emotions that lack the cognitive element
- Triggered by
  - Perception (e.g. seeing a predator), reasoning about situations, recall of emotive memories
- Effects
  - Instinctive behaviours and how these are regulated by cognitive control
  - Prioritising what you are thinking about and what feels important
  - Influences on recall, new memories, reinforcement of existing memories and reinforcement learning of behaviours
- Fast and instinctive vs slow and deliberate
  - Rapid instinctive appraisal and response, avoiding the delay incurred with conscious thought, but subject to errors of judgement due to lack of considered thought
  - Functional implementation as a feed-forward classification network

* **empathic***: /ɛmˈpaʊɪk/ adjective* – showing an ability to understand and share the feelings of another

Cognition and Conscious Thought

ERCIM
European Research Consortium for Informatics and Mathematics

- **Cortico basal-ganglia circuit**
  - The centre of conscious thought
- Symbolic (graphs) + sub-symbolic (statistics)
  - Chunk based symbolic representation of concepts and relationships
  - Statistical weights reflecting prior knowledge and past experience
- Rule engine connected to many parts of the cortex
  - Connections via buffers that hold single chunks
  - Rules represent reasoning & procedural knowledge
  - Learned from experience (hierarchical reinforcement learning)
- Sequential application of rules to cognitive buffers
  - Approximately every 50 mS or longer
- Parallel processes for graph algorithms
  - Recall of memories
  - Selection of rules
- Autobiographical and episodic memories
- Reasoning at multiple levels of abstraction

Chunks: a collection of properties that include references to other chunks

Modules: specialised graph databases and algorithms, accessed via buffers that hold a single chunk

Rules: conditions ranging over module chunk buffers, and actions that either update the buffers or invoke graph algorithms

Courtesy of Freepik

Actions

- **Cortico cerebellar circuit**
- Handles actions devolved to it by conscious thought
- Real-time control with parallel processing
- Contains more than three times the number of neurons in the cortex*
- Cerebellum acts as flight controller managing activation of myriad sets of muscles in coordination with perceptual input from the cortex
  - Fast, real-time control without the overheads of conscious thought
- Offloads processing from cortico basal-ganglia circuit thereby enabling higher level thought whilst actions are underway
- Performance degrades when conscious thought diverts visual attention, starving cerebellum of visual feedback
  - Analogous to loss of a sensor – where you can fallback on dead reckoning
- Learning through experience, starting with conscious thought
  - The acquisition of so called *muscle memory*
- Implemented as suite of analogue computers computing functions over time
- Examples: talking, walking and playing the piano

* The human cerebellum contains 70 billion nerves vs 20 billion for the cerebral cortex, see Suzana Herculano-Houzel, 2010

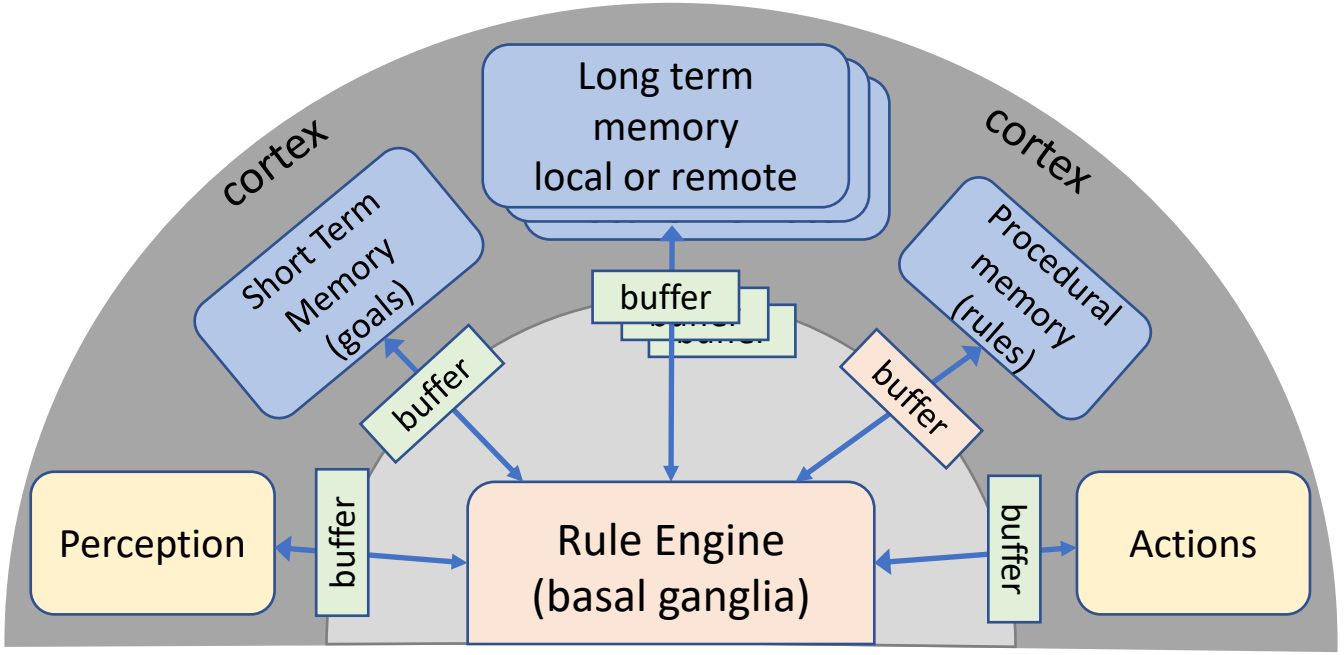
ERCIM
European Research Consortium
for Informatics and Mathematics

Diving Deeper

Cognition

Basal ganglia as a sequential rule engine

and nature's solution to the scaling problem



Each buffer holds a single chunk and represents the current state of a bundle of nerve fibres connecting to a particular cortical region. The rule conditions and actions operate over these buffers. Moreover, the buffers can be likened to HTTP clients, where the cortex is like a set of HTTP servers. This architecture originates in John Anderson's work on ACT-R.

High performance will entail emulating the Striatum, Pallidum and Thalamus, akin to the Rete algorithm

Formal Semantics

ERCIM
European Research Consortium
for Informatics and Mathematics

- Formal semantics seeks to understand meaning by constructing precise mathematical models that can determine which statements are true or false in respect to the assumptions and inference rules given in any particular example.
- The Web Ontology Language (OWL) is formalised in terms of description logic, which is midway between propositional logic and first order logic in its expressive power. OWL can be used to describe a domain in terms of classes, individuals and properties, along with formal entailment, and well understood complexity and decidability.

Cognitive AI and Chunks

ERCIM
European Research Consortium for Informatics and Mathematics

- Chunks, by contrast to mathematical logic, makes no claims to formal semantics, reflecting the lack of certainty and incomplete knowledge found in many everyday situations.
- Instead of logical entailment, the facts are updated by rules in ways that have practical value for specific tasks, e.g. turning the heating on when a room is observed to be too cold*, or directing a robot to pick up an empty bottle, fill it, cap it and place it into a box.
- Rules are designed or learnt to carry out tasks on the basis of what is found to be effective in practice.
- The combination of symbolic and statistical information is critical for machine learning and reasoning about likelihoods

* See smart home demo

Chunks

Chunks is a simple amalgam of RDF and Property Graphs

Chunks correspond to concurrent firing patterns across bundles of nerves to a particular cortical region, see Chris Eliasmith's work on Semantic Pointers

Each chunk is a typed named collection of properties whose values are names, numbers, booleans (i.e. true or false), dates, string literals of comma separated lists thereof*

Here is an example of a chunk – you can use newline or semicolon as punctuation

```
dog dog1 {
   name "fido"
   age 4
}
```

```
dog dog1 {name "fido"; age 4}
```

The chunk ID is optional, and if missing, will be automatically assigned when adding the chunk to a graph. If the graph already has a chunk with the same ID, it will be replaced by this one. You are free to use whitespace as you please, modulo the need for punctuation. String literals apart from URIs must be enclosed in double quote marks.

* A minimalist version of chunks limits properties to just the names of other chunks

Chunks

Numbers are the same as for JSON, i.e. integers or floating point numbers. Dates can be given using a common subset of ISO8601 and are treated as identifiers for read-only chunks of type *iso8601* with properties for the year, month, day etc., e.g.

```
# Albert Einstein's birth date
iso8601 1879-03-14 {
   year 1879
   month 3
   day 14
}
```

Names are used for chunk types, chunk IDs, chunk property names and for chunk property values. Names can include the following character classes: letters, digits, period, and hyphen. Names starting with @ are reserved. A special case is the name formed by a single asterisk which is used to match any chunk type.

Chunks & Links

Sometimes you just want to indicate that a named relationship applies between two concepts. This can expressed conveniently as follows:

John likes Mary

which is equivalent to

```
likes {
   @subject John
   @object Mary
}
```

Mapping names to RDF

For integration with systems using RDF

To relate names used in chunks to RDF, you should use @rdfmap along with @base. For instance:

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

which will map *mouse* to http://example.org/ns/mouse

It may be more convenient to refer to a collection of @rdfmap and @prefix mappings rather than in-lining them, e.g.

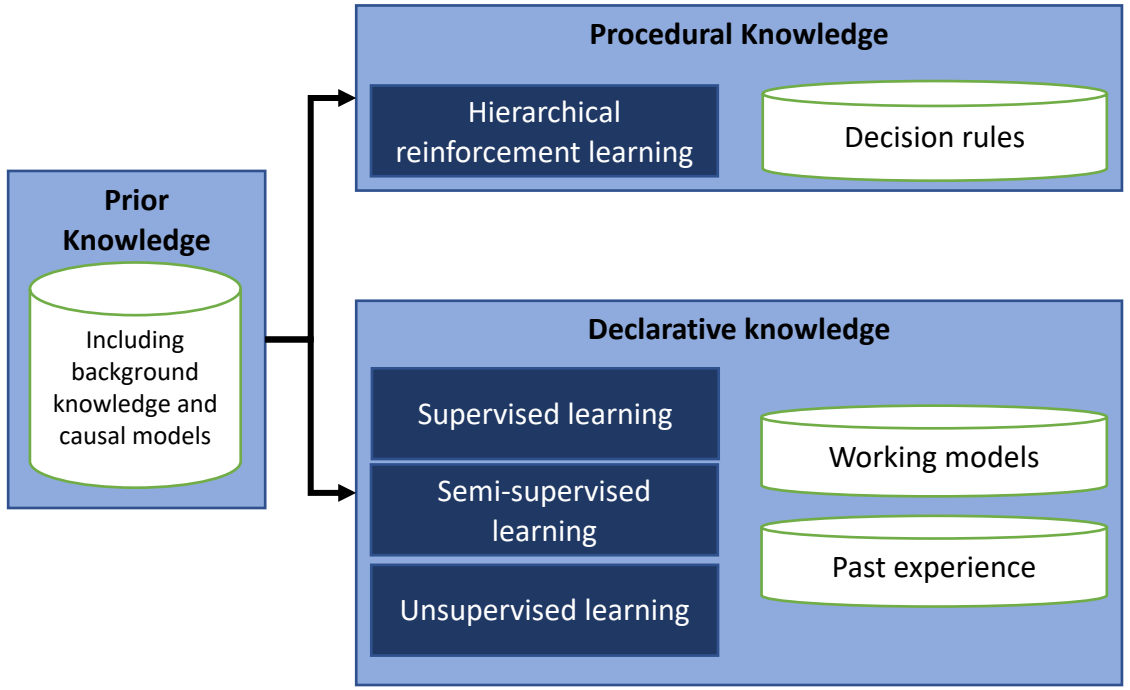
```
@rdfmap from http://example.org/mappings
```

You can likewise use @prefix for defining URI prefixes, e.g.

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

Machine Learning

- Manual development of knowledge won't scale cost effectively
- We therefore need to rely on machine learning for declarative and procedural knowledge
  - Many algorithms to take advantage of
- Prior knowledge enables learning from small datasets
- Semi-supervised learning as human guided exploration with attention to salience
- Learning at multiple levels of abstraction
  - Case based reasoning to speed learning
- Active learning – continuous, surprise driven
  - Mimicking humans as prediction machines – we attend to novelty to improve our predictions
- Learning from experience with real or simulated environments and multiple cognitive agents
- Use with natural language for teaching skills to cognitive agents



Chunk Rules

- Condition-action rules expressed in chunks with a convenient syntax
- Each chunk names the module it applies to (default is goal module)
- Inject goals to trigger rules
- Conditions match module buffers
- Actions update buffers directly or invoke module operations, e.g. to recall a fact from memory, to assert a fact, or to invoke an external operation, e.g. to move a robot's arm, or to say hello
- Variables pass information from conditions to actions

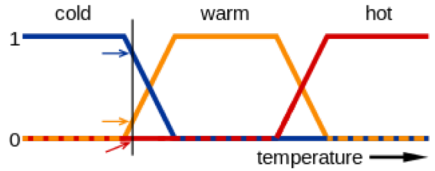
```
# Given a goal like
#     count {state start; start 2; end 5}
# prepare to start counting using facts like
#     increment {number 1; successor 2}
count {state start; start ?num}
=>
    count {state counting},
    increment {@module facts; @do get; number ?num},
    console {@do show; value ?num}

# count up one at a time
count {state counting; start ?num1; end ~?num1},
   increment {@module facts; number ?num1; successor ?num3}
=>
   count {start ?num3},
   increment {@module facts; @do get; number ?num3},
   console {@do show; value ?num3}

# stop after last one
count {start ?num; end ?num; state counting}
=>
   count {@do update; state stop}
```

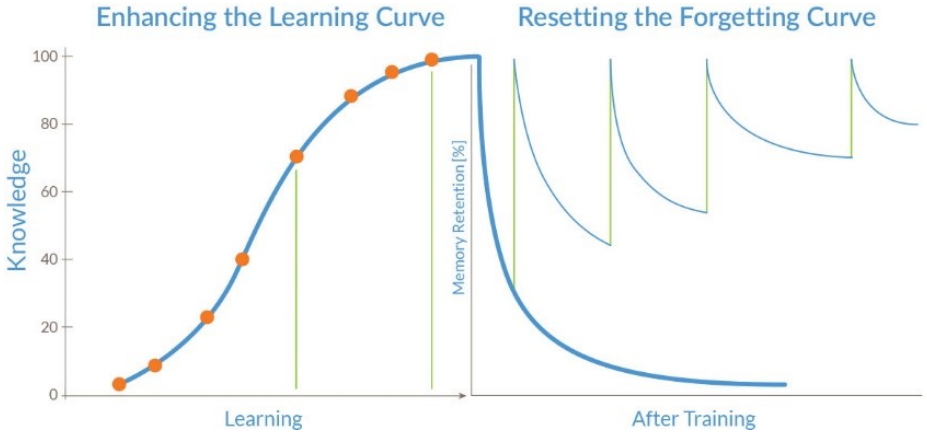
Built-In Module Operations

- Modules must support the following operations
  - **@do clear** to clear the module's buffer and pop the queue
  - **@do update** to directly update the module's buffer
  - **@do queue** to push a chunk to the queue for the module's buffer
  - **@do get** to recall a chunk with matching type and properties
  - **@do put** to save the buffer as a new chunk to the module's graph
  - **@do patch** to use the buffer to patch a chunk in the module's graph
  - **@do delete** to forget chunks with matching type and properties
  - **@do next** to load the next matching chunk in an system dependent order
  - **@do properties** to iterate over the set of properties in a buffer
  - **@for** to iterate over the items in a comma separated list
- Modules may define additional operations, e.g. to control a robot
- More details at:
  - https://github.com/w3c/cogai/blob/master/chunks-and-rules.md
- Formal specification is in preparation with view to standardisation

Many Ways to Reason

- Many forms of reasoning have to deal with uncertainties, e.g.
  - Induction: building models to explain regularities
  - Abduction: determining the most likely explanation of some observations
  - Causal reasoning about plans
  - Fuzzy reasoning involving blends of different states
- Mimicking human memory
  - In any large knowledgebase we only want to recall what is relevant to the current situation based upon past experience
  - Ebbinghaus forgetting curve – our ability to recall information drops off over time unless boosted by repetition
  - Closely spaced repetitions have less effect
  - Longer gaps means less salient features are forgotten
  - Spreading activation – concepts are easier to recall on account of their relationship with other concepts*
  - Stochastic selection of facts and rules, and the role of gaussian noise for recall of memories with similar strengths
  - Multiple implementations possible: chunk graphs (Anderson), holographic memory (Kelly), pulsed neural networks (Eliasmith)





Recommended reading:

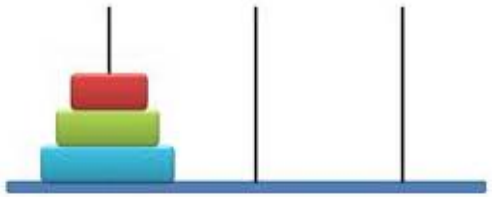
- Large-Scale Model of the Functioning Brain
- Holographic Declarative Memory
- Neural Network Model of Memory Retrieval

** Nature figured out the "Page Rank" algorithm many millions of years ago!*

Natural Language

- Natural language is key to richer human-machine collaboration
  - Syntax, semantics and pragmatics in dialogue
- NLU as combination of pipelined processing + deliberative reasoning
  - Lexicon of words, their parts of speech, word senses and linguistic tags, e.g. person, number, case, gender
  - Word by word incremental concurrent processing of syntax and semantics to resolve ambiguities
  - Spreading activation model for disambiguation based upon the context and statistical likelihood
  - No backtracking!
  - Similar approach in reverse for NLG
- Work now underway on demos for NL understanding and generation
  - Dialogue between customer and waiter at a restaurant as a scenario with well defined language usage and semantics, together with reasoning about shared plans

Towers of Hanoi



```
# move the red disc to the right peg

verb v1 {word move; subject p1; to p2}
phrase p1 {word disc; det the; adj red}
phrase p2 {word peg; det the; adj right}

# after application of ruleset

move m1 {disc disc3; to peg3}
```

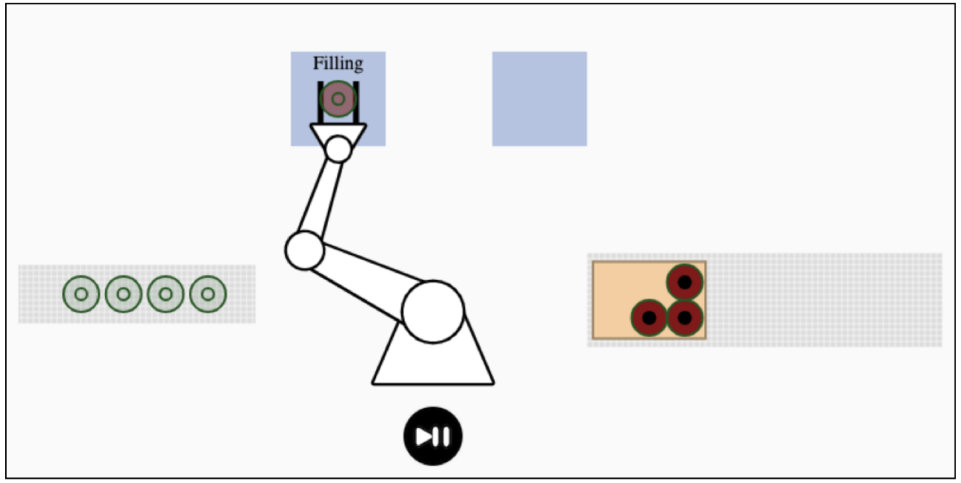
See: https://www.w3.org/Data/demos/chunks/nlp/toh/

and: https://www.w3.org/Data/demos/chunks/nlp/dinner/

Application to Smart Factories

https://www.w3.org/Data/demos/chunks/robot/

- **Cognitive AI demo that runs in a web page**
- Live simulation of bottling plant with robot, conveyor belts, filling and capping stations
- Real-time control by a cognitive agent




```
# add bottle when belt1 has space and wait afresh
space {thing belt1} =>
action {@do addBottle; thing belt1},
space {@do wait; thing belt1; space 30}

# add box when belt2 has space and wait afresh
space {thing belt2} =>
action {@do addBox; thing belt2},
action {@do stop; thing belt2},
space {@do wait; thing belt2; space 95}

# stop belt when it is full and move arm
full {thing belt1} =>
action {@do stop; thing belt1},
action {@do move; x -120; y -75; angle -180; gap 40; step 1}
```

Log:

```
set goal to: after _:54 {step 1}
executed rule _:27 move
set goal to: after _:55 {step 2}
executed rule _:30 grasp
set goal to: after _:56 {step 3}
starting belt1
wait on filled
executed rule _:34 start
```

How it works

- The demo features real-time concurrent control over multiple factory machines
  - uses a mix of events and time-based functions
- Start rule initializes conveyor belts and gets the robot arm ready
  - @do start and @do stop as actions for moving conveyor belts
- Goal queue ensures that closely spaced events are handled without loss
- @do wait action queues goal when something has either happened already or when it happens at some point in the future
  - Used to control when to add an empty bottle at the start of belt1
  - Used to signal when to grab a bottle when it reaches the end of belt1
- The robot arm is controlled with @do move, @do grasp and @do release
  - You tell the arm where you want the gripper and in what orientation
  - The arm works out how to do this in terms of real-time control over the 4 joints, smoothly accelerating to each joint's maximum speed and then back to a stop – lots of math!
- Animation is performed using HTML's requestAnimationFrame method
  - Used to simulate the brain's cortico-cerebellar circuit
- Similar actions for the bottle filling and capping station
- Each bottle's current position is determined by the thing that supports it, e.g. the belt, robot arm or the packing box.

Could be combined with a planning system for dynamic control of a production line as a whole

Cognitive Planning

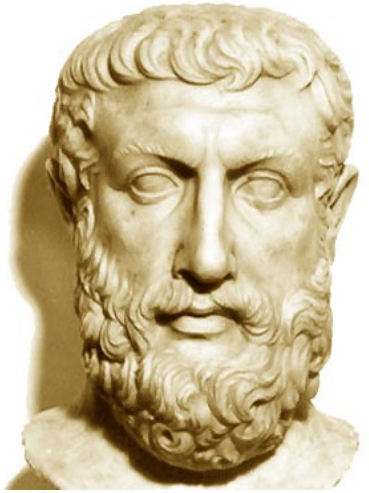
- Cognitive agent for use in reconfiguring production line and reprogramming all of the robots as needed
  - Human-machine collaborative effort
- Declarative knowledge of factory machinery
- Knowledge graph can be used to ensure that
  - Actions only use declared methods
  - Actions are passed valid values
  - Action responses handled correctly
- Hierarchical reinforcement learning with simulated factory to ensure plans are safe and efficient before being deployed in real factory
  - Knowledge based planning and iterative refinement
  - Reasoning at multiple levels of abstraction
- Integration with Manufacturing Execution System (MES)
  - Monitoring and analytics dashboard
  - Materials, orders and shipping
  - Scheduling proactive maintenance

```
# Ontology for things for validating facts and rules

# bottles
thing bottle {properties level, capped}
type level {isa number; min 0; max 100; initial 0}
type capped {isa boolean; initial false}

# robots
thing robot {properties x, y, arm}
thing arm {properties x, y, angle, gap}
type x {isa integer}
type y {isa integer}
type angle {isa number; min -180; max 180}
type gap {isa integer; min 5; max 45}

# belts
thing belt {properties x, y, length, moving, items}
type length {isa integer}
type moving {isa boolean; initial false}

# belts hold a list of items with a position and width
type items {isa list; contains item}
thing item {properties position, p2}
type position {isa number}

# definition of a specific width property
property p2 {name width; isa number; min 0}
```

See robot demo: facts.chk

Ontologies

- An ontology is a model of concepts, their relationships and properties, and usually represented as graph of vertices and edges
- The smart home demo* uses a simple ontology
- If we want to search a marketplace for devices and services then a larger ontology that covers them all would be helpful
- You can think of it as a shared vocabulary for talking about things
- Using a shared vocabulary makes it easier for everyone to understand each other
- That it turn encourages market growth and prosperity



In the **Greek philosophical** tradition, Parmenides was among the first to propose an **ontological** characterization of the fundamental nature of existence. In the prologue (or proem) to "On Nature" he describes two views of existence. ... This posits that existence is what may be conceived of by thought, created, or possessed.

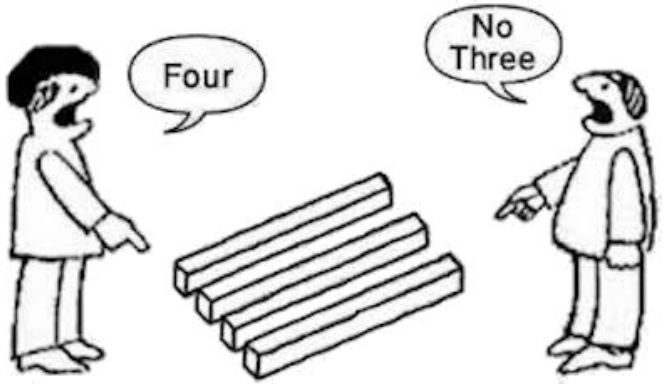
See: Wikipedia article on ontology

* Part of an upcoming AIOTI tutorial on semantic technologies

Different Ontologies for Different Purposes

ERCIM
European Research Consortium for Informatics and Mathematics

- Ontologies are designed with particular purposes in mind
- This means that some things are included and others excluded
- Questions around how much detail to model
  - For instance, public-private partnerships for city services?
  - Different classes of services, e.g. health, education, sanitation, policing, parks, courts, libraries, transport, planning, telecommunications, other utilities, …
- This is challenging for both humans and machines to deal with
  - No one person knows everything, likewise no one ontology will suffice
- We need ways to bridge the comprehension gaps
- This involves bridging both language and knowledge
  - The answers we're seeking will depend on the context
  - However, it only needs to be good enough for the purpose at hand
- In the long term ontologies will be developed in collaboration with cognitive agents
  - Automatic adaptation of applications to cope with evolving requirements



We can't agree on everything!

W3C Cognitive AI Community Group

ERCIM
European Research Consortium for Informatics and Mathematics

See: https://www.w3.org/community/cogai/, https://github.com/w3c/cogai

- Participation is open to all, free of charge
- Focus on demonstrating the potential of Cognitive AI
  - A roadmap for developing AI that is general purpose, collaborative, empathic and trustworthy
- Collaboration on defining use cases, requirements and datasets for use in demonstrators
  - https://github.com/w3c/cogai/tree/master/demos
- Work on open source implementations and scaling experiments
- Work on identifying and analysing application areas, e.g.
  - Helping non-programmers to work with data (worth $21B by 2022 according to Forester)
  - Cognitive agents in support of customer services (worth $5.6B by 2023)
  - Smart chatbots for personal healthcare
  - Assistants for detecting and responding to cyberattacks
  - Teaching assistants for self-paced online learning
  - Autonomous vehicles
  - Smart manufacturing
- Outreach to explain the huge opportunities for Cognitive AI

Acknowledgements

- Chunk rules are a form of production rules as introduced by Alan Newell in 1973 and later featured in his SOAR project
- John Anderson's work on human associative memory in 1973, later combined with production rules for ACT in 1976, maturing as ACT-R in 1993. ACT-R is a theory for simulating and understanding human cognition that has been widely applied to cognitive science experiments
- Marvin Minsky for his work on frames, metacognition, self-awareness and the importance of emotions for cognitive control
- Philip Johnson-Laird for his work on mental models and human reasoning – we don't rely on the laws of logic and probability, but rather by thinking about what is possible
- The European Commission for funding from the Horizon 2020 research and innovation programme under grant agreement No. 780732, Boost 4.0 big data for factories

ERCIM
European Research Consortium
for Informatics and Mathematics

Cognitive AI

giving computing a human touch