

Sztuczna Inteligencja

GOFAI

Architektury kognitywne-modele umysłu

Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Włodzisław Duch

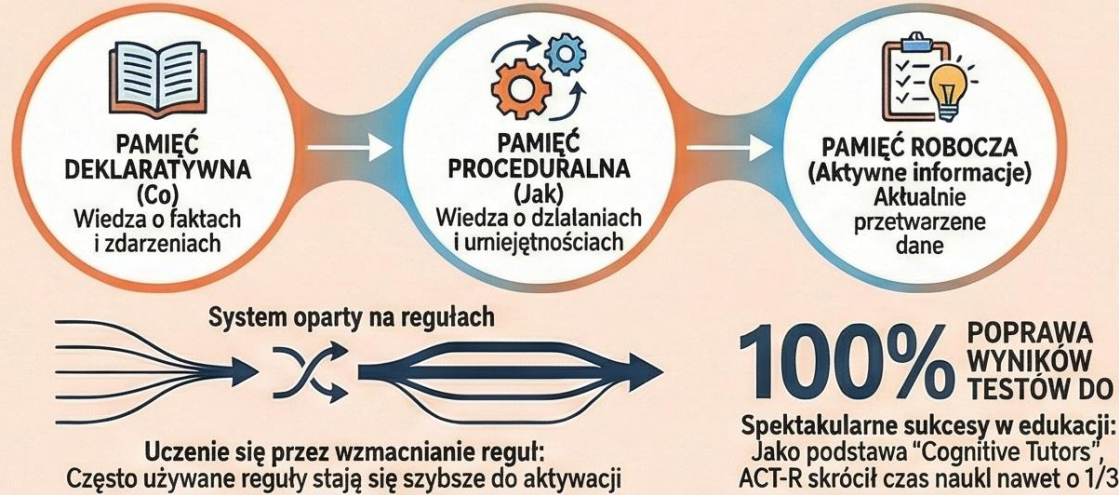
[Strona wykładów](#)

Co będzie

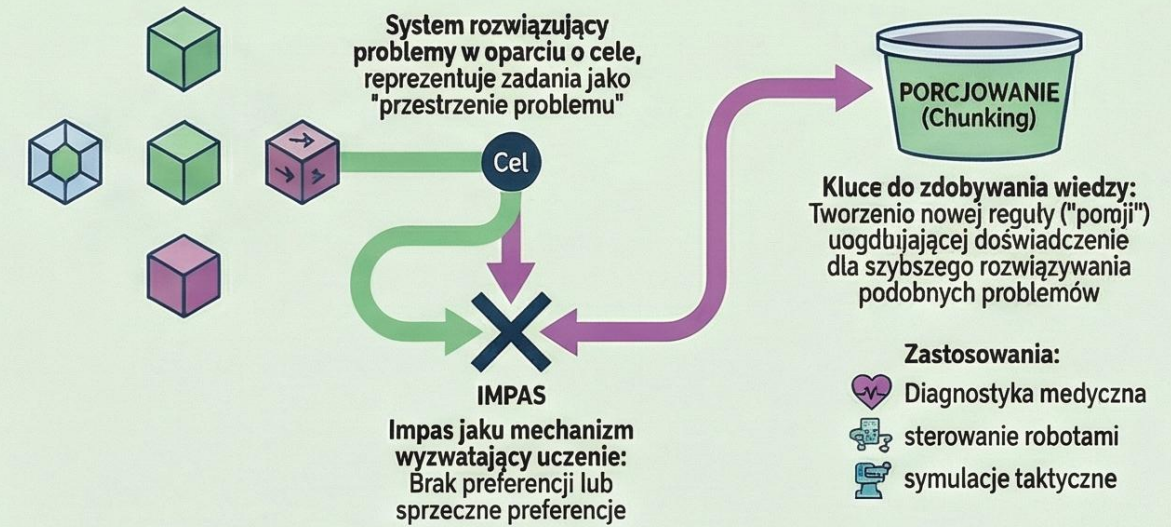
- Teorie poznania
- Systemy oparte na wiedzy
- Modele kognitywne
- ACT
- SOAR
- Cog

ARCHITEKTURY KOGNITYWNE: PRZEGLĄD MODELI UMYSŁU W SZTUCZNEJ INTELIGENCJI

MODEL ACT-R: ZINTEGROWANA TEORIA POZNANIA



ARCHITEKTURA SOAR: UCZENIE SIĘ Z DOŚWIADCZENIA



PROJEKT COG: INTELIGENCJA UCIELEŚNIONA I ROZWOJOWA

Inteligencja jako wynik rozwoju i interakcji z otoczeniem:
Wymaga fizycznego ciała i interakcji społecznych

Osiągnięcia:
Ruchy sakadyczne oczu, koordynacja wzrokowo-ruchowa, rozpoznawanie twarzy, kontakt wzrokowy



Brak centralnego procesora (CPU): Mózg nie ma jednego, centralnego kontrolera; składa się z wielu niespójnych podsystemów

Ograniczenie:
Brak wyższych funkcji poznawczych (zdolności językowe, abstrakcyjne myślenie)

PRZYSZŁOŚĆ: KOMPUTERY NEUROMORFICZNE



Sprzęt inspirowany budową mózgu:
Naśladowanie gęstości i energooszczędności neuronów i synaps w biologicznym mózgu



IBM TrueNorth
1 milion neuronów i 256 milionów synaps przy minimalnym zużyciu energii
Rozwijany system o coraz większej skali

Potencjał do rewolucji w AI:
Nowe architektury (w tym obliczenie optyczne i kwantowe) obiecują ogromny wzrost szybkości i efektywności

Model Act-R

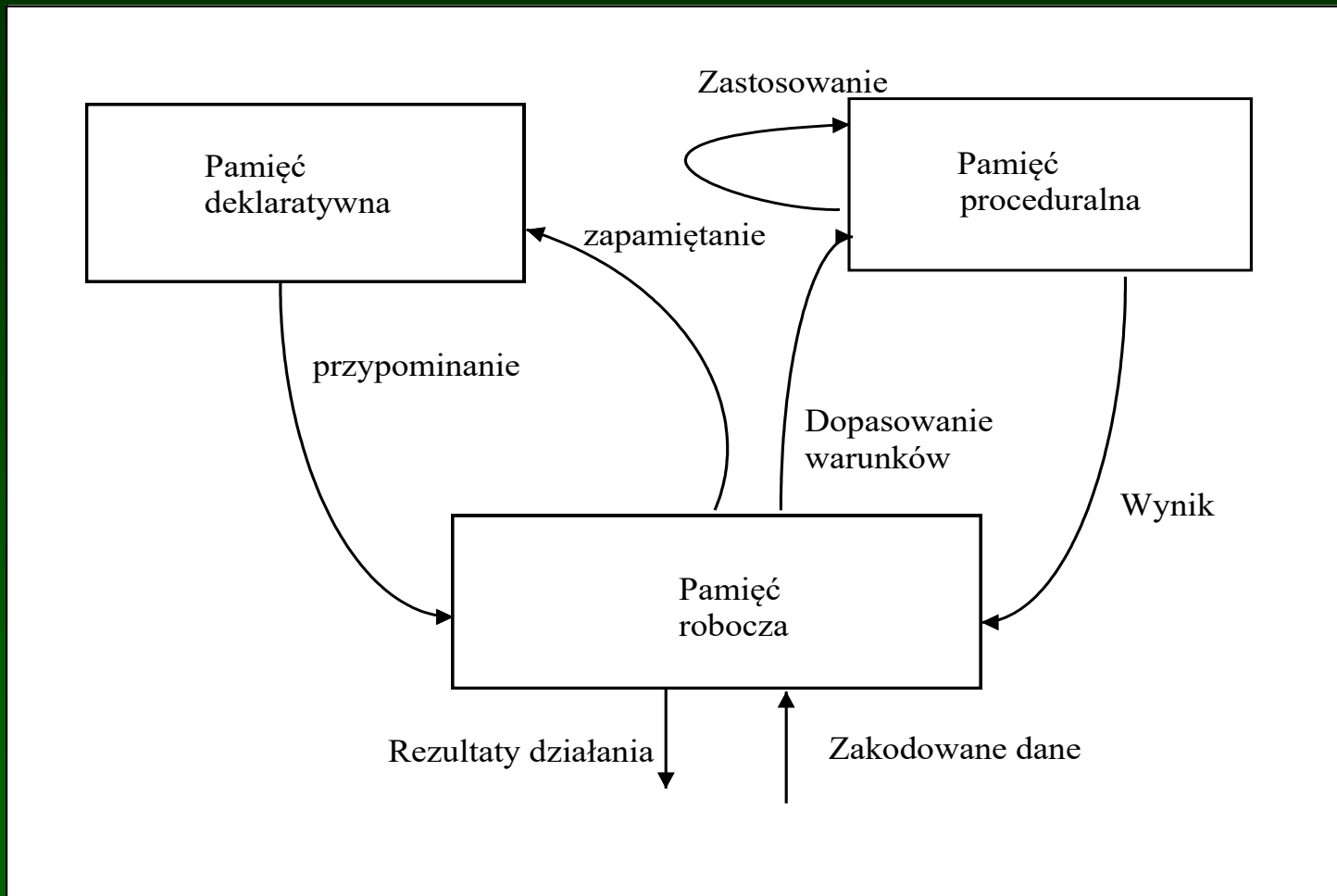
John Anderson, CMU, modele Act* i ACT-R oparte są na teorii poznania, definiując prostą architekturę kognitywną. Są w nim 3 rodzaje pamięci + uczenie.

- **Deklaratywna pamięć długotrwała:** sieć semantyczna + mechanizm asocjacji. W pamięci tej przechowywane są pojęcia, reguły, obrazy.
- **Pamięć proceduralna:** reguły produkcji. Reguły mają warunki związane z informacją przechowywaną w pamięci deklaratywnej a wynikiem ich działania mogą być nowe obiekty lub asocjacje, które mogą zostać zapisane w pamięci deklaratywnej.
- **Pamięć robocza:** aktywna część pamięci deklaratywnej.
- **Uczenie:** przypisanie wag regułom produkcji.

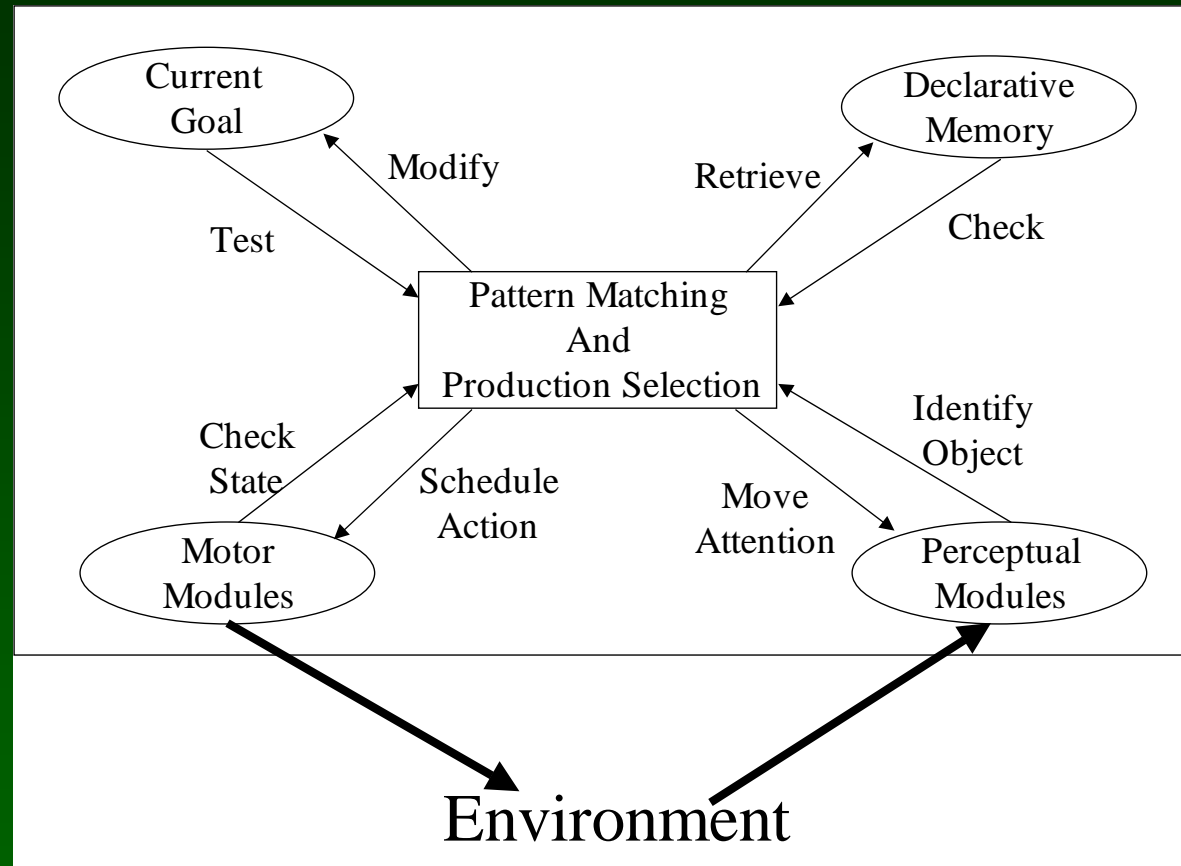
Często używane reguły stają się coraz ważniejsze.

Model Act* zrealizowano w oparciu o język symulacyjny Grapes.

Act* - architektura



Act-R 5 - architektura



Cele: integracja percepcji, działania i poznawania, działanie w czasie rzeczywistym, uczenie się, odporność na niespodzianki, inspiracje neurobiologiczne, brak parametrów typowych dla systemów uczących się.

ACT - zastosowania

ACT zastosowano do:

wyjaśnienia własności pamięci, kolejność odpowiedzi i przypominania, uczenie się nowych słów, uczenie się elementów programowania i rozumowanie geometryczne w czasie dowodzenia twierdzeń.

System jest zbyt skomplikowany by można było przeanalizować teoretycznie jego zachowanie, pozostają jedynie symulacje.

Cognitive Tutor™ for Writers

Act* jako baza do inteligentnych programów wspomagających nauczanie (tutoriali): algebra, geometria, pisanie, quantitative literacy, programowanie, Lisp.

Testy w szkołach dały bardzo dobre wyniki: wyniki testów do 100% lepiej, czas nauki do 1/3 krótszy; komercyjna firma Carnegie Learning:

<http://www.carnegielearning.com/>

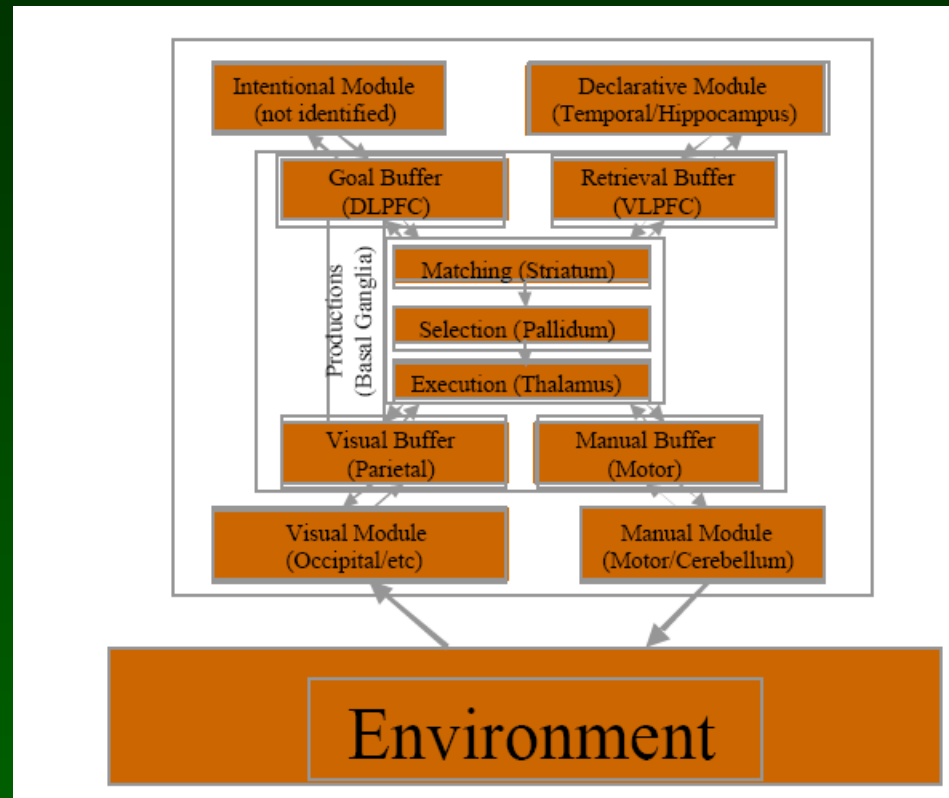
Pomaga ponad 475.000 uczniów (2007)!

ACT – R PM (Perceptual Motor)

Wersja do modelowania interakcji człowieka z maszynami lub programami.

- Działa w złożonym środowisku, ma moduł sterujący ruchem, wzrok, słuch i mowę.
- Równoległe operacje: jednoczesna obserwacja wzrokowa, słuchanie, poruszanie kończynami.
- Synchronizacja czasowa: ACT-R/PM modeluje czasy reakcji dla procesów percepcji, działania i poznawczych człowieka.
- Użyty do modelowania mechanizmów uwagi, jednoczesnego wykonywania dwóch zadań, ruchu oczu, zachowań kierowców, złudzeń wzrokowych (np. efektu Stroop'a)

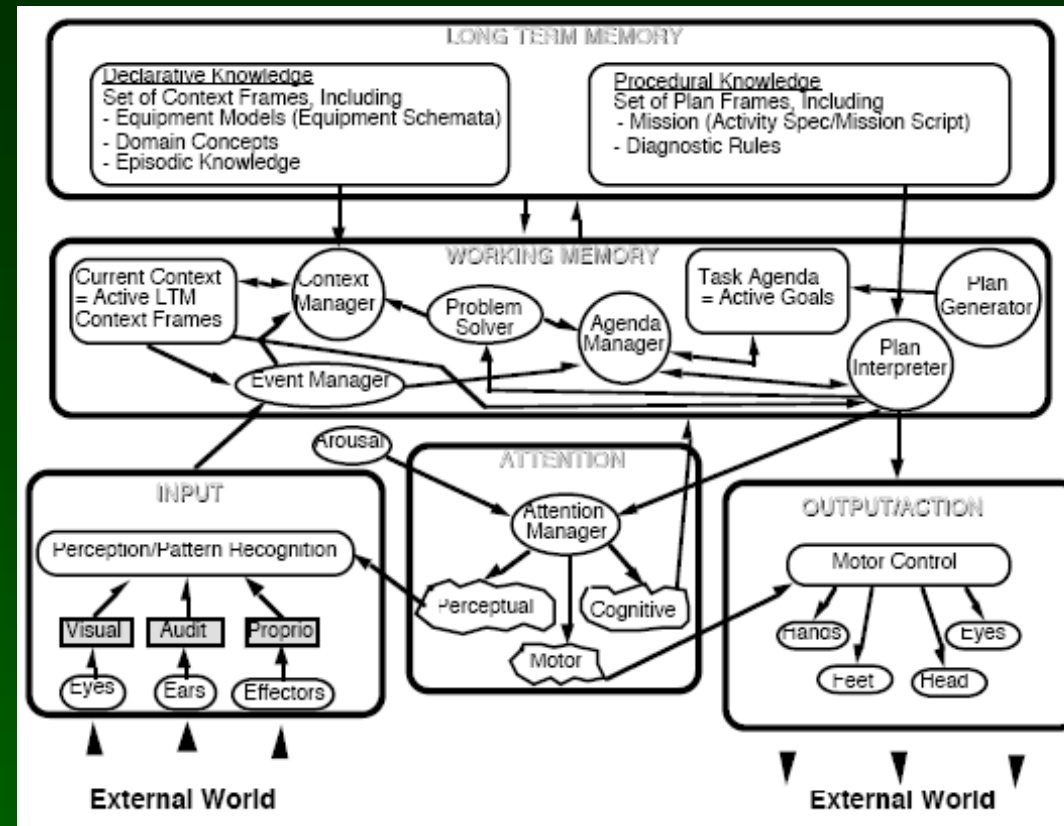
Act-R-PM - architektura



Wersja PM integruje percepcję (P) i działania motoryczne (M).

Jej moduły można z grubsza powiązać z funkcjami różnych części mózgu, pamięcią deklaratywną (płaty skroniowe, hipokamp), roboczą w korze przedczołowej (PFC), jądrami podstawy mózgu.

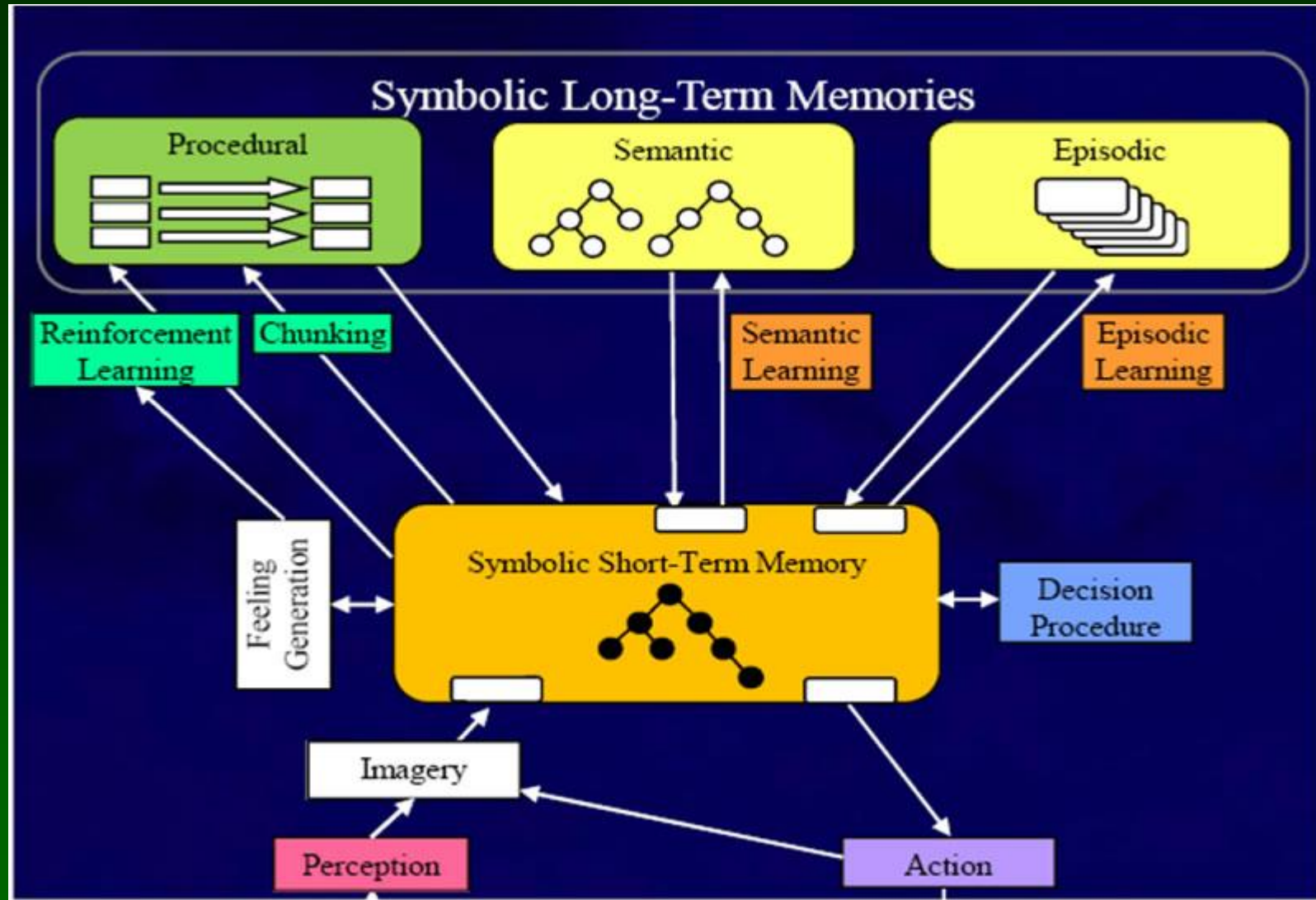
MIDAS



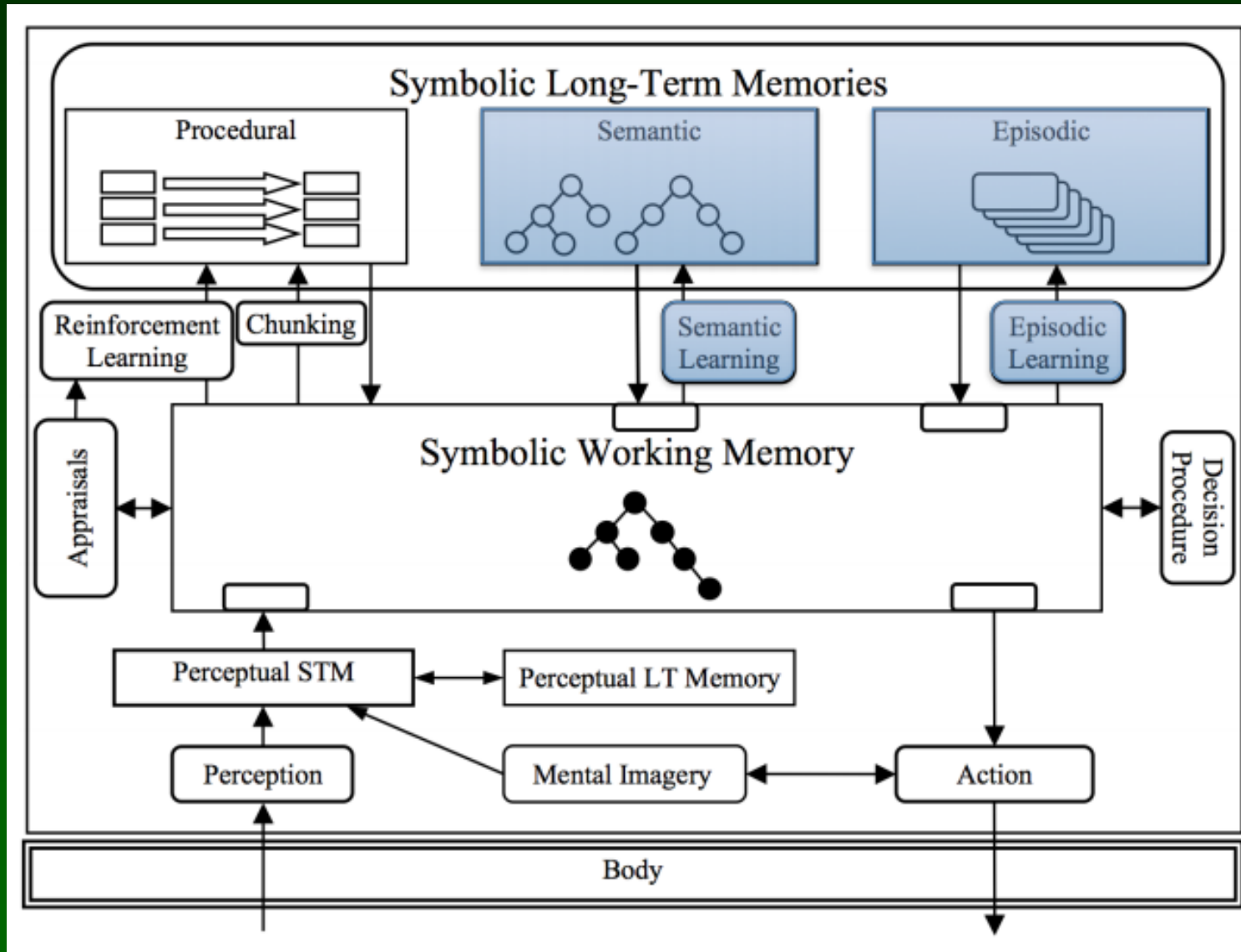
Przykład architektury typu BICA, czyli Brain Inspired Cognitive Architecture: projekt MIDAS opracowany przez NASA.

Problemy związane z projektowaniem, skomplikowana architektura, <http://www-midas.arc.nasa.gov/>

Elementy architektury SOAR



Architektura SOAR 9



Architektura SOAR

SOAR, architektura kognitywna (Laird, Newell, Rosenbloom, od 1982).

Algorytm i informacja + odtwarzacz => konkretne działanie.

SOAR miał początkowo tylko mechanizmy poznawcze.

Percepcja, motoryka, kognicja zostały zintegrowane na dalszym etapie.

- ✓ Zadania reprezentowane są przez podanie przestrzeni problemu.
- ✓ Pamięć i symbole określone są przy pomocy reguł produkcji.
- ✓ Zapis własności polega na przypisaniu atrybutom wartości.
- ✓ Decyzje podejmowane są w oparciu o preferencje: akceptuj/odrzuć, lepszy/obojętny/gorszy.
- ✓ Zachowanie sterowane jest przez cele; podcele generowane są automatycznie w odpowiedzi na impas w działaniu systemu.
- ✓ Mechanizm tworzenia porcji wiedzy działa w ciągły sposób na wynikach pośredniej realizacji celów.

Porcjowanie (chunking)

Teoria Newella i Rosenbloom (1981).

Pamięć działa hierarchicznie, grupuje dane w “porcje” (chunks).

STM, pamięć krótkotrwała: kilka sekund, 7 grup danych.

Konieczne rekursywne budowanie „bloczków”.

1. Mózg tworzy hierarchiczne porcje informacji w ciągły sposób.
2. Im więcej porcji tym szybciej można wykonać zadanie.
3. Porcje na wysokim poziomie hierarchii pojawiają się rzadziej; mają skomplikowaną strukturę, odwołują się do mniejszych porcji.

Te założenia wystarczą, by zbudować system o czasach reakcji nieodróżnialnych od prawa potęgowego.

Konieczna teoria + architektura systemu.

Inteligencja: nie algorytmu ale cała architektura systemu zdolnego do poznawania (kognitywnego).



Szukanie w przestrzeni wiedzy

Reprezentacja problemu = przestrzeń możliwych rozwiązań.

Stany układu i operatory zmieniające te stany.

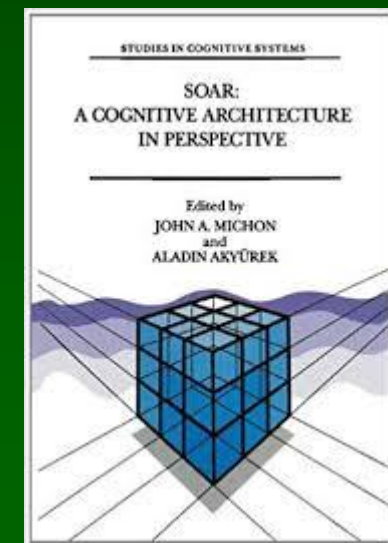
Zadania reprezentowane są przez podanie przestrzeni problemu.

Szukanie w przestrzeni wiedzy i szukanie w przestrzeni problemów lub stanów to różne zagadnienia - wiedza jest w pamięci systemu, kolejne stany nie istnieją przed rozpoczęciem szukania.

Elementarne funkcje systemu:

1. wybór przestrzeni problemu,
2. wybór stanu w tej przestrzeni,
3. wybór operatora stosowanego do wybranego stanu,
4. zastosowanie tego operatora.

Podobna idea: [dataflow architecture](#).



Soar: sterowanie przez zdarzenia

Funkcje elementarne programów komputerowych:

wybierz operatory, wybierz dane do operacji, wykonaj operację, zachowaj rezultaty, wybierz następną instrukcję.

Zachowania Soar: ruch w przestrzeni problemu.

Duża wiedza => Soar zmierza wprost do celu.

Strategia „minimalnego deklarowania się”: działania ustalane w momencie podejmowania decyzji.

W programach komputerowych działania ustalane są wcześniej, twórcy programów przewidują różne możliwości, tu mamy sterowanie przez dane, zmieniające program działania, zamiast przez algorytmy, zawierające takie programy działań.

Własności Soar

Jednorodność reprezentacji wiedzy dzięki regułom produkcji.
Tylko jeden rodzaj pamięci trwałej i pamięć robocza (WM).

Struktury danych przechowywane w pamięci roboczej to warunki występujące w regułach produkcji.

Przybliżenie pamięci adresowalnej kontekstowo: warunki w pamięci roboczej są wskazówkami pozwalającymi na odtworzenie struktur w postaci wyników reguł produkcji.

Reguły ciągle dodają nowe elementy do pamięci, ale nie wymazują ich ani nie modyfikują.
Wymazywanie niepotrzebnych elementów pamięci możliwe jest dzięki temu, że niektóre z nich przestają być dostępne i nie mogą wpływać na proces rozwiązywania problemu.

Nie ma rozstrzygania konfliktów: wszystkie reguły dające się zastosować są wykonywane, wyniki dopisywane są do WM.

Przykład reguł Soar

Przykład reguły: zaproponuj operator 'zrozumiałem'.

*(sp propose-operator*comprehend: nazwa op.*

(goal <g>^problem-space <p>^state <s>)

(problem-space <p>^name base-level-space) nazwa p-ni

(state <s>^object ^input <i>)

(box^on-table^on-top nothing)

-(signal<i>^examined yes) pudełko na stole

→

(operator <o>^name comprehend)

*(preference<o> ^role operator ^value acceptable ^goal <g> ^problem-space
<p> ^state <s>)*

(input<i> examined yes))

Przykład reguł Soar

Zmienne w nawiasach <> atrybuty ^wartość.

Reguła *propose-operator*comprehend* ma 5 warunków, 3 wyniki.

War. 1: w WM musi być cel, zmienna <g>, posiadająca atrybut ^*problem-space*<p> oraz atrybut ^*state*<s>.

War. 2: p-ń problemów *base-level-space*, czyli przestrzeń na poziomie podstawowym.

War. 3: stan <s> = pudełko (box) na stole (^*on-table*) i na wierzchu (^*on-top nothing*).

War. 4: pudełko nie zostało („-”) sprawdzone (^*examined yes*).

Jeśli spełnione to tworzony jest nowy operator przypisany zmiennej <o>, nazwany *comprehend*.

Operator ma preferencje o atrybutach „rola” (^*role operator*) i „wartość” określoną jako akceptowalna (^*value acceptable*) w kontekście celu, przestrzeni problemu i stanu.

Soar - podejmowanie decyzji

Określenie własności obiektów: przypisanie wartości atrybutom.

Atrybuty i własności mogą być innymi obiektami.

Obiekty identyczne: te same atrybuty/wartości.

Podejmowanie decyzji: faza przygotowań i faza decyzji.

Przygotowania: gromadzona jest wiedza o obecnej sytuacji.

Rozważanie sytuacji: Soar uznaje, że należy podjąć jakieś działanie bez jego podjęcia.

Jest to możliwe w oparciu o mechanizm tworzenia preferencji.

Decyzja podejmowana jest w kontekście historii działania systemu, poprzednio podejmowanych decyzji, używanych przestrzeni problemów i operatorów.

Preferencje określają czy akceptować jakiś cel czy go odrzucić, czy dany obiekt jest lepszy czy gorszy dla danego operatora.

Soar - preferencje

Preferencje używają koncepcji: lepszy/gorszy, najlepszy/ najgorszy, akceptuj/odrzuć, wymagany/zabroniony i obojętny.

Faza przygotowania trwa do zakończenia dopisywania nowych elementów do pamięci roboczej.

Jedynie decyzje z preferencjami określonymi jako akceptowalne mogą zostać podjęte.

Faza przygotowania tworzy wszystkie możliwe w danej sytuacji preferencje, spośród których wybierane są najkorzystniejsze. Decyzja oznacza, że dla aktualnego kontekstu wybierany jest operator, cel, przestrzeń problemów lub stan w tej przestrzeni.

Cele i konteksty przechowywane są na stosie, dzięki czemu powstaje hierarchia celów, od supercelu początkowego, jakim jest rozwiązanie danego problemu do cząstkowego celu związanego z rozwiązaniem problemu na danym etapie.

Soar - impasy

Nowe cele tworzone gdy Soar wpada **w impas**.

Impas: brak preferencji, sprzeczne preferencje, jednakowe, lub preferencje odrzucenia już podjętej decyzji.

Brak wiedzy powoduje impasy.

Impas - okazja, by nauczyć się czegoś nowego.

Nowy cel: początkowo bez operatora, przestrzeni i stanu.

Impas jednakowo dobrego wyboru: podcele związane są ze sposobem reprezentacji zadania, wyborem pomiędzy różnymi przestrzeniami problemów lub stanami i operatorami.

Brak preferencji prowadzi do podcelów związanych z określeniem warunków zastosowania operatora. Cele przyjmują postać operatorów, np. cel związany z ułożeniem klocków jest operatorem przesuwania klocków.

Soar - porcjowanie

„Porcjowanie” (**chunking**) - nowe reguły produkcji, dodawanie wiedzy w oparciu o odniesione sukcesy.

Rozstrzygnięcie impasu: obiekty A, B, C w pamięci + nowe obiekty i preferencje D, E jako wyniku rozstrzygnięcia.

Nowa reguła: IF A.and.B.and.C THEN D.and.E.

Porcjowanie to aktywny proces uczenia się, nowa wiedza obejmuje sukcesy i porażki.

Soar zwiększa swoje kompetencje i szybciej rozwiązuje problemy podobnej klasy + coraz bardziej złożone problemy.

Nie ma podziału na etap uczenia się i wykonywania zadania.

Wiedza zapisana w postaci warunków i akcji może być użyta w każdej, w której występują podobne warunki - uogólnianie.

Porcjowanie pozwala odkrywać strategie postępowania, zamieniając mikro na makroposunięcia.

Soar - uczenie

Przykład efektywności porcjowania:

R1-Soar, system ekspercki do konfiguracji systemów komputerowych.

Bez uczenia: 1731 cykli decyzji, 232 reguł.

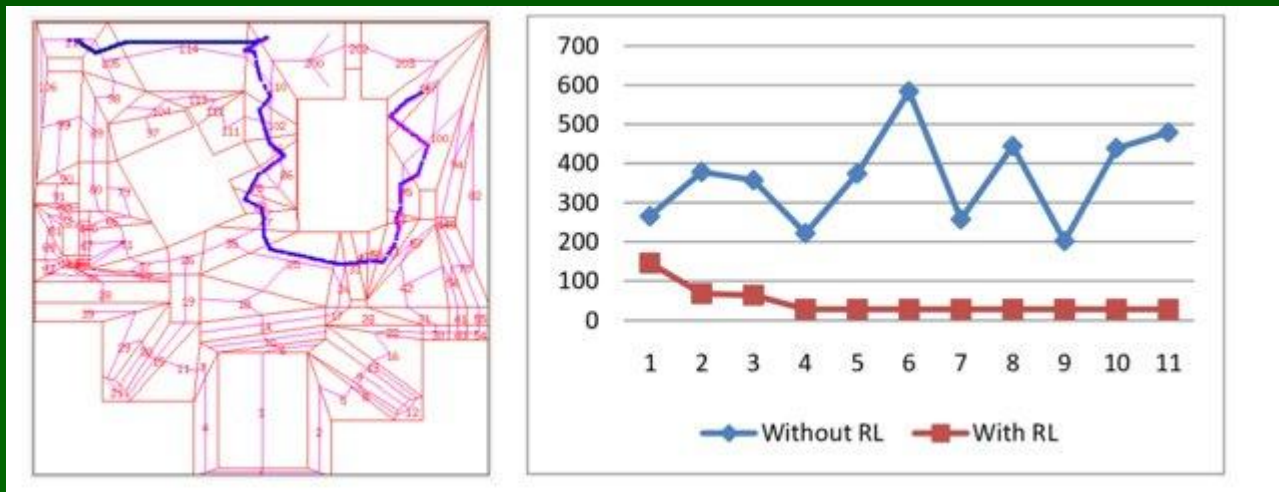
Bez uczenia, dodanie dodatkowych 10 reguł strategicznych skraca szukanie rozwiązania do 150 cykli, 242 reguły.

Po nauczaniu: 7 cykli, 291 reguł (dodanych w wyniku porcjowania).

W nowszych wersjach zastosowano algorytm RL, uczenia z krytykiem.

Np.: poszukiwanie drogi, start=>cel, liczba sprawdzanych węzłów/it.

Robot Splinterbot
szuka drogi.



Soar - zastosowania

Designer-Soar: projektowanie algorytmów komputerowych.
Dane wejściowe to zbiór liczb całkowitych a wyjściowe to największy podzbiór składający się z liczb dodatnich.

Neomycin-Soar, diagnostyka medyczna.

Merl-Soarl, sterowanie produkcją.

Robo-Soar, robot + analiza obrazu.

Tank - Soar;

TacAir-Soar, ekspertyza czasu rzeczywistego

Quake - Soar

Steve VR - tutoriale; [EPIC-Soar](#)

[Emocje Steve'a](#) i Steve w [maszynowni statku](#).



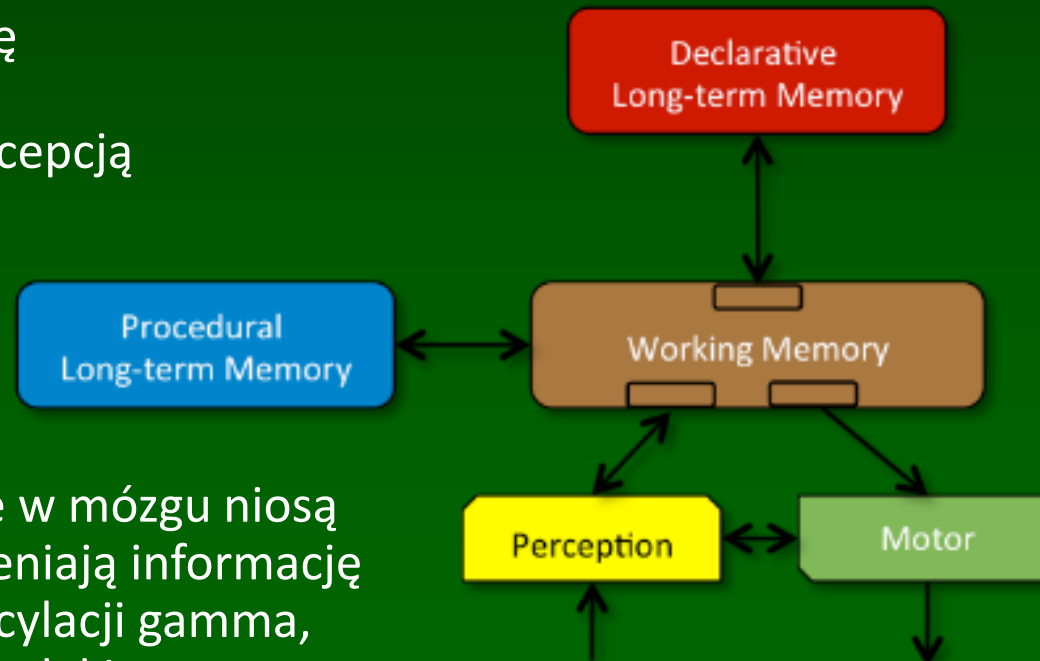
[Soartech](#): komercyjna firma wykorzystująca technologię SOAR,
automatyzacja złożonych czynności, rozszerzone możliwości ludzi.

Standardowy model umysłu

Standardowe architektury symboliczne ACT-R, SOAR, MIDCA, LIDA były omawiane na [Symposium on Cognitive Architecture](#) (VISCA 2021), zorganizowanym przez grupę SOAR.

Tendencje 2021: próba zdefiniowania „standardowego modelu umysłu”; robotyka i ucieleśnieni kognitywni agenci, ale również wiele inspiracji neurokognitywnych, uczenia zachowań metodą reinforcement learning.

Model standardowy, na który się większość zgadza, to 3 rodzaje pamięci, moduły związane z percepcją i efektorami.



Czego nie rozumiemy? Oscylacje w mózgu niosą informację, różne regiony wymieniają informację za pomocą krótkich, szybkich oscylacji gamma, wolniejsze oscylacje przecierają szlaki komunikacji. Jak to działa?

Cog

Projekt realizowany od 1994-2003 roku na MIT, w grupie R. Brooksa: **inteligencja behawioralna**.

Założenie: inteligencja ludzka jest wynikiem procesów rozwojowych, oddziaływań społecznych, ucieleśnienia umysłu oraz integracji wielomodalnej informacji zmysłowej.

- Nowe umiejętności powstają w oparciu o wcześniej nabyte.
- Ludzie nie tworzą pełnych, wiernych reprezentacji.
- Wiele niespójnych reprezentacji używanych jest w różnych kontekstach.
- Mózg nie ma CPU – nie ma centralnego kontrolera (np. rozczepione mózgi).
- Mózg ma specyficzne ograniczenia, nawet w zakresie logiki.
- Irracjonalność zachowań człowieka, rola emocji.

Cog - rozwój

Rozwój: złożone zachowanie w oparciu o prostsze,
analiza strategii chwytania zabawek przez niemowlęta;
uczenie w etapach prostsze niż końcowe;
uczenie wskazywania celu wzrokowego,
mapy wizualno-motoryczne,
stopniowe zwiększanie złożoności – niemowlęta robią ruchy synchronicznie, mają początkowo słaba ostrość wzroku.

Kontakty społeczne i opiekuńcze środowisko: konieczne do prawidłowego rozwoju => potrzebna humanoidalna postać robota.

Zaburzenia, np. autyzm pokazują, jakie interakcje są ważne.

Ostatnie prace zmierzają między innymi do modelowania i leczenia autyzmu przy wykorzystaniu robotów.

Dzielenie uwagi, kontakt wzrokowy, gesty i wyrazy twarzy, kiwanie głową i wiele innych odruchów stwarza realistyczne wrażenie.

Cog - cielesność

Cielesność: bezpośrednie sprzężenie percepcji i działania motorycznego pozwala na ugruntowanie sensu wewnętrznych rep.

Ludzka inteligencja wymaga ciała; łatwiej zrobić robota niż symulować grawitację, sprężystość, koordynacje kończyn itp.

Cog ma tułów o 21 stopniach swobody (od 1998).

Dynamika systemu wynika z oddziaływania ze środowiskiem, informacji priopriorecepcyjnej, zmysłu równowagi.

Jedynie reprezentacje to wewnętrzne sekwencje motoryczne.

Integracja wielomodalna: zmysły wzajemnie sobie pomagają, iluzje akustyczne mogą wywołać iluzje wzrokowe i odwrotnie!

System stabilizacji obrazu wykorzystuje system równowagi i odruch przedsionkowy (VOR).

Orientacja: sygnały akustyczne i wizualne.

Cog - zrobiono:

1. Ruchy sakadyczne oczu,
2. koordynację słuchu i oczu,
3. lokalizację dźwięku,
4. układ równowagi,
5. kooordynację ruchu głowy i oczu, rąk i oczu,
6. sprężystość kończyn,
7. kontrolę ruchów,
8. rozpoznawanie twarzy i kontakt wzrokowy,
9. skórę wrażliwą na dotyk,
10. kategoryzację dotykanych obiektów.

Na poziomie działań sensomotorycznych, lub niższych czynności poznawczych, da się osiągnąć wiele.

Co z wyższymi czynnościami poznawczymi? Językiem?

Nie udało się tego osiągnąć.

Cog - pytania

Jak rozwijać taki system? Jakie etapy i strategie uczenia?

Jakie zachowanie wywołuje zachowania opiekuńcze u ludzi?

Jakie motywacje, emocje, popędy są konieczne do komunikacji?

Jak ma wyglądać pamięć zdolności motorycznych i wzrokowo-przestrzennych?

Jak wykorzystać wyuczone zdolności w nowych kontekstach?

Jak osiągnąć spójne zachowanie zbioru różnych podsystemów o różnych (czasem sprzecznych) celach, zachowaniu, działaniu, stabilności?

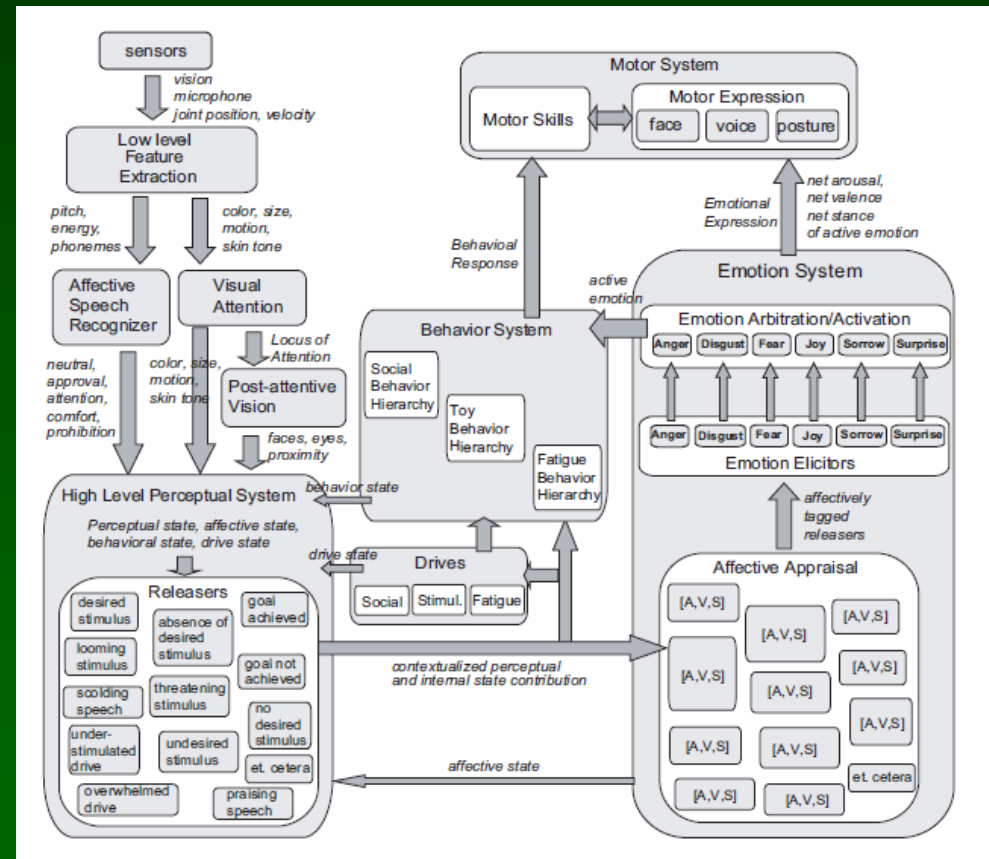
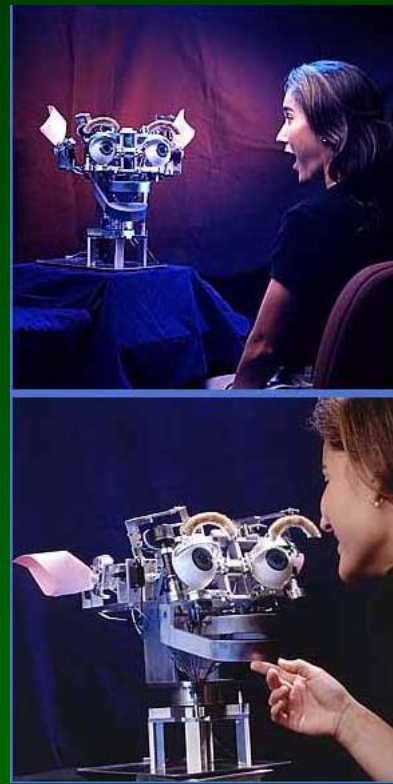
Jak szczegółowo należy modelować biologiczne cechy organizmu?

Jak oceniać postępy w tworzeniu takiego systemu?

Kismet

Naturalne interakcje z ludźmi wymagają analizy i ekspresji emocji: projekt Kismet doprowadził do zrobienia głowy „uczuciowego” robota.

[The Rise of Personal Robots](#) | Cynthia Breazeal | TED Talks



- [Microsoft HUE](#): Human Understanding and Empathy, projekty Microsoft.

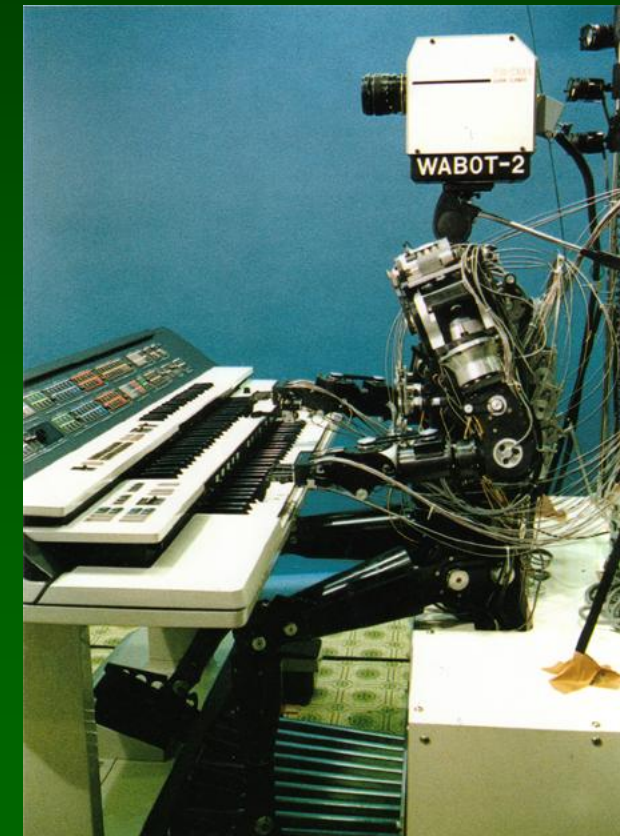
Wabot

Tworzenie wyspecjalizowanych robotów nie jest takie trudne.

Wabot-1 (Waseda Univ, Tokyo, 1970-73) był pierwszym humanoidalnym robotem na świecie; poruszał się na dwóch nogach i mówił syntetycznym głosem.

Wabot-2 (Waseda Univ, Tokyo, 1980-84) był wyspecjalizowanym robotem do grania na klawiaturze, czytał nuty „średniej trudności”, mógł akompaniować śpiewającej osobie grając ze słuchu, mógł też porozumiewać się używając mowy.

Stworzenie inteligencji ogólnej (AGI) jest znacznie trudniejsze.



Sophia

Hanson Robotics, HK.



Prezentacja: IBC 2017-Path to the Human Like Robot: Meet Sophia.
Prosty chatbot do dialogu, dość realistyczny model.

Most advanced robots video

10 Amazing Robots That Really Exist

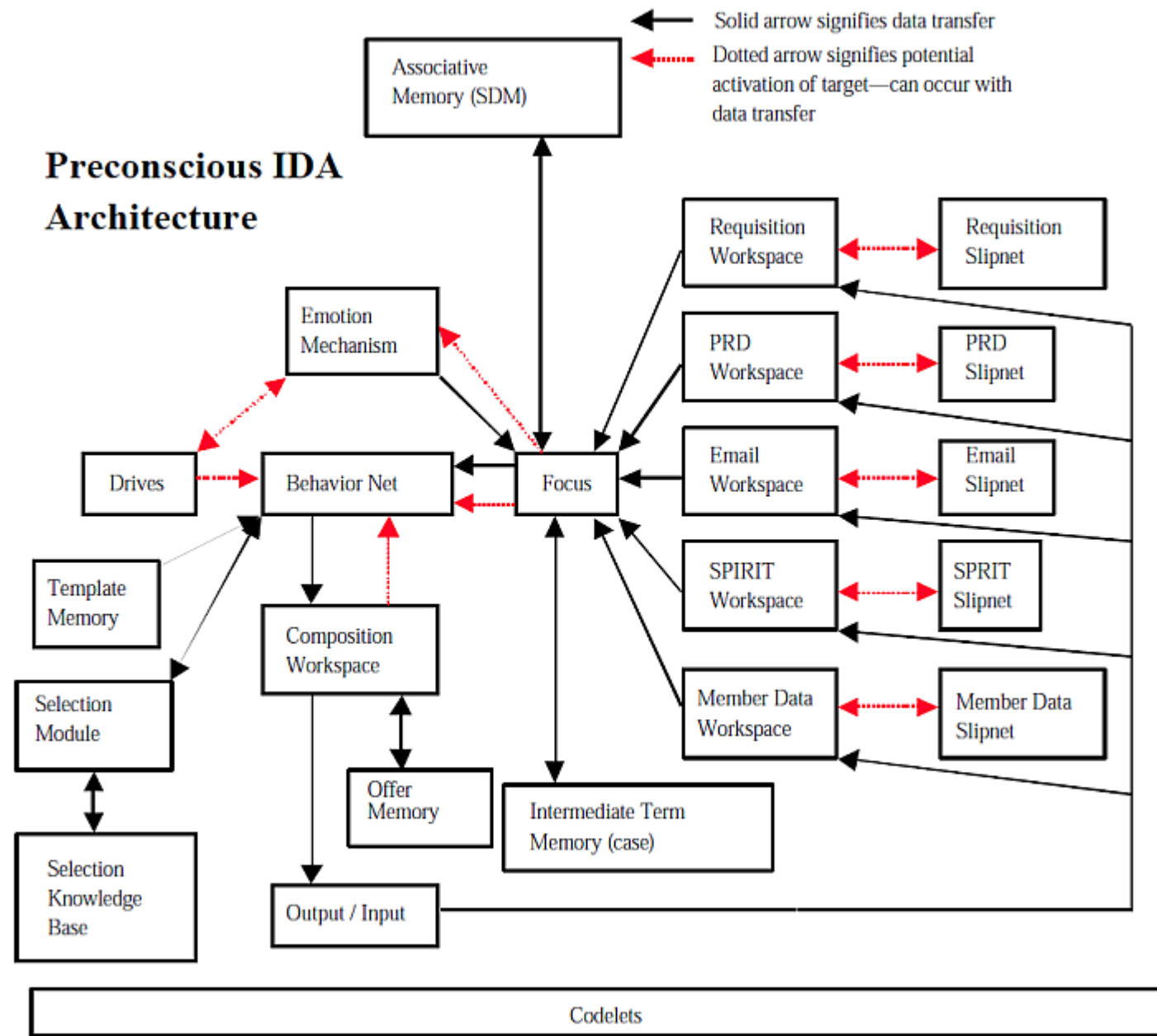
Robot animals

Wielki postęp w możliwości kontroli ruchu robotów osiągnięto dzięki uczeniu maszynowemu.

Boston Dynamics robi najbardziej zaawansowane roboty.

M

Preconscious IDA Architecture



Schemat IDA analizującego zawartość emaili do marynarki wojennej.

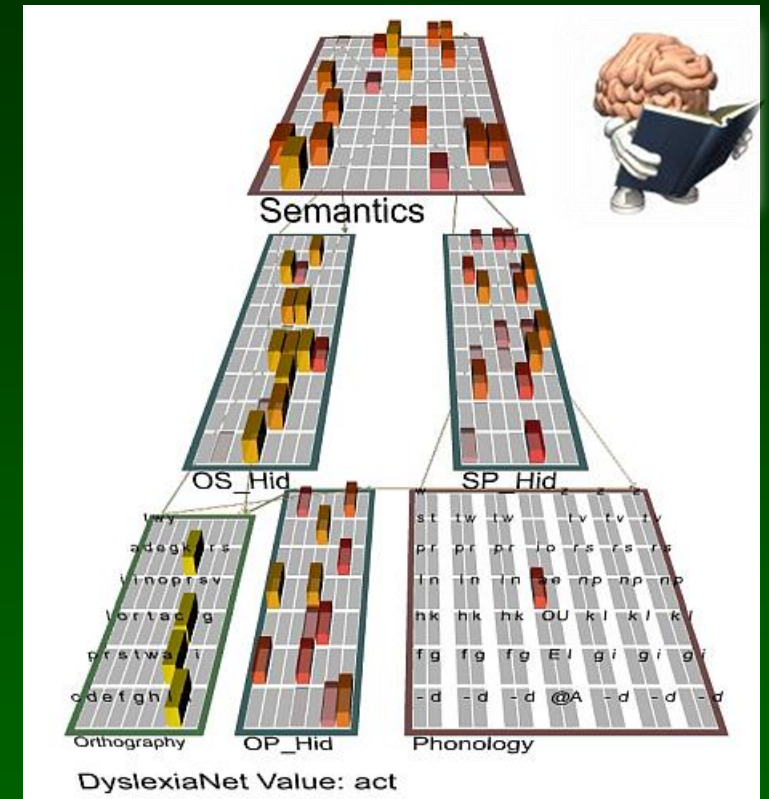
Modele neuronowe

Wiele zjawisk da się modelować na poziomie neuronowym, ale robiono to zwykle tylko na małą skalę.

Przykład: [Emergent](#) (osobny wykład)

Modele i eksperymenty psychologiczne dotyczące percepcji, uwagi, pamięci, oddziaływania różnych rodzajów pamięci ze sobą, języka, podejmowania decyzji i wielu innych zjawisk.

Trudno na razie takie modele przeskalować do rozwiązywania realnych problemów, ale możliwe są symulacje procesów całego mózgu pomagające w zrozumieniu działania mózgu, np. [The Virtual Brain](#).



Model czytania w Emergent: warstwy ortografii, fonologii i semantyki i warstwy ukryte mapujące między nimi.

Synapse i systemy neuromorficzne

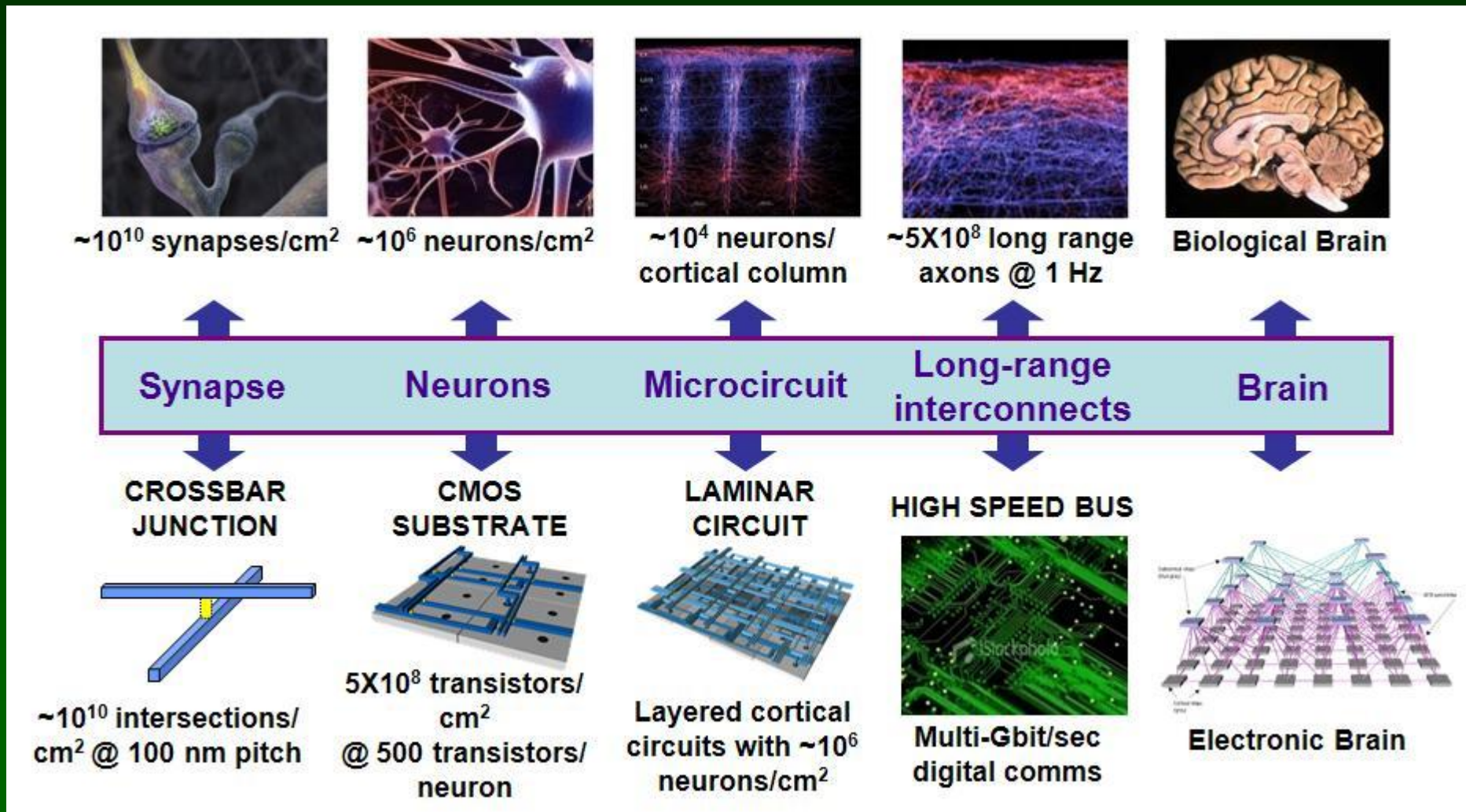
- Projekt DARPA-IBM: cel – budowa sztucznego mózgu.
- Początek 2009, 2 faza 2011, 3 faza koniec 2013 roku.
- Założenie: memristory pozwolą osiągnąć gęstość elementów podobną jak w mózgu.
- W 2012 roku: 16-rack IBM® Blue Gene®/Q (262144 CPUs, 256 TB RAM), 65 mld neuronów, 16 bilionów synaps, tylko ~388x wolniej niż mózgi, średnia częstość impulsacji neuronów >8 Hz.
- <http://www.modha.org/>
- <http://www.neurdon.com/>
- <http://www.artificialbrains.com/darpa-synapse-program>
- <http://www.is.umk.pl/~duch/ref/11/11-Brain-challenge.pptx>

Inne: Samsung Dynamic Vision Sensor (DVS) na czipach TrueNorth, [Neuromorphic Research Community](#) intro (Intel).

Intel Loihi (2017), ~ 68 Gops/W, Pohoiki Springs system.

Gyr Falcon (2017) DSP small chip, 24.3 Tops/W, processing in memory.

Od mózgow do komputerów



DARPA Synapse project

Neuromorficzne komputery/roboty

Projekt SyNAPSE 2015: IBM TrueNorth chip

1 chip ~1 mln neuronów i 1/4 mld synaps (5.4 mld tranzystorów),

1 moduł=16 chipów ~16 mln neuronów, 4 mld synaps, moc 1.1 wata!

Skalowanie: 256 modułów ~4 mld neuronów, 1T = 10^{12} synaps, < 300 W.

IBM Neuromorphic System osiąga
złożoność
≈ ludzkiego mózgu.

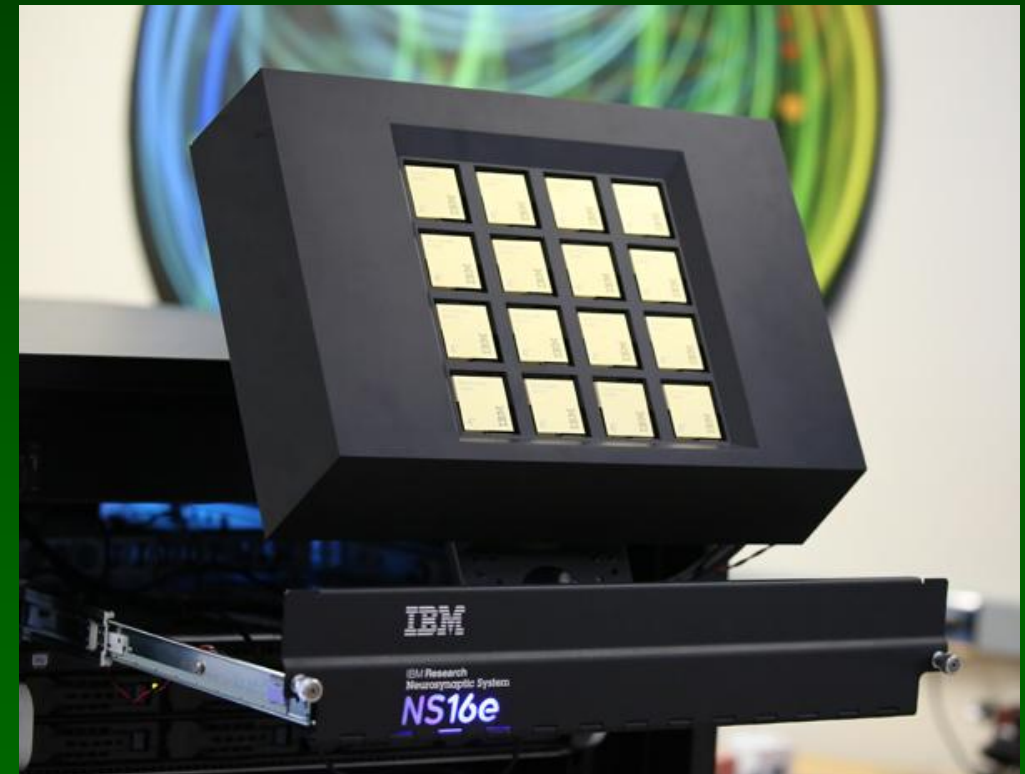
Ale programowanie tych neuronów nie
jest łatwe.

IBM Research założył SyNAPSE
University.

Samsung Dynamic Vision Sensor (DVS)
jest z TN.

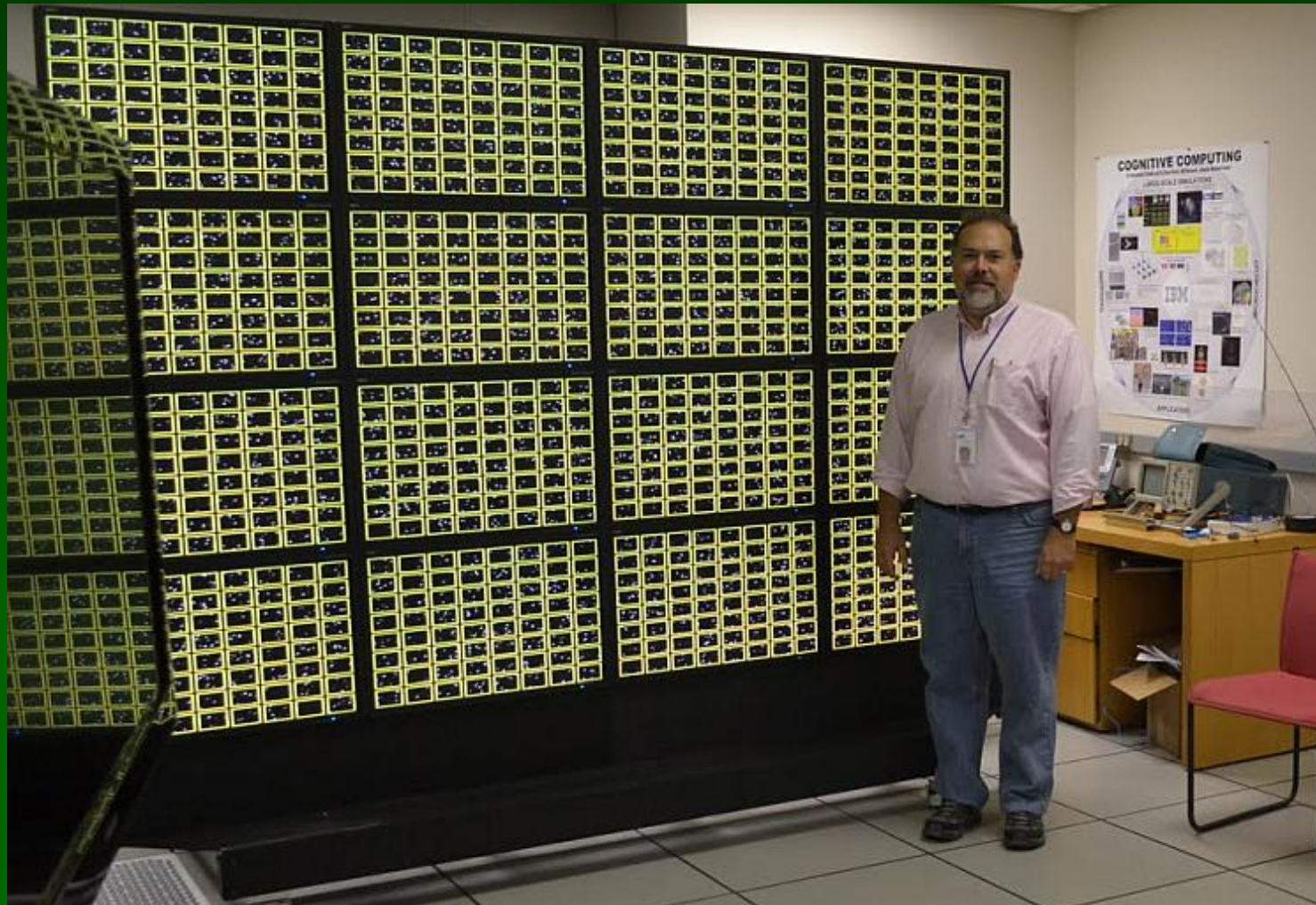
Supersymulator HBP?

Intel Loihi Hala Point
Deep South ludzki mózg.

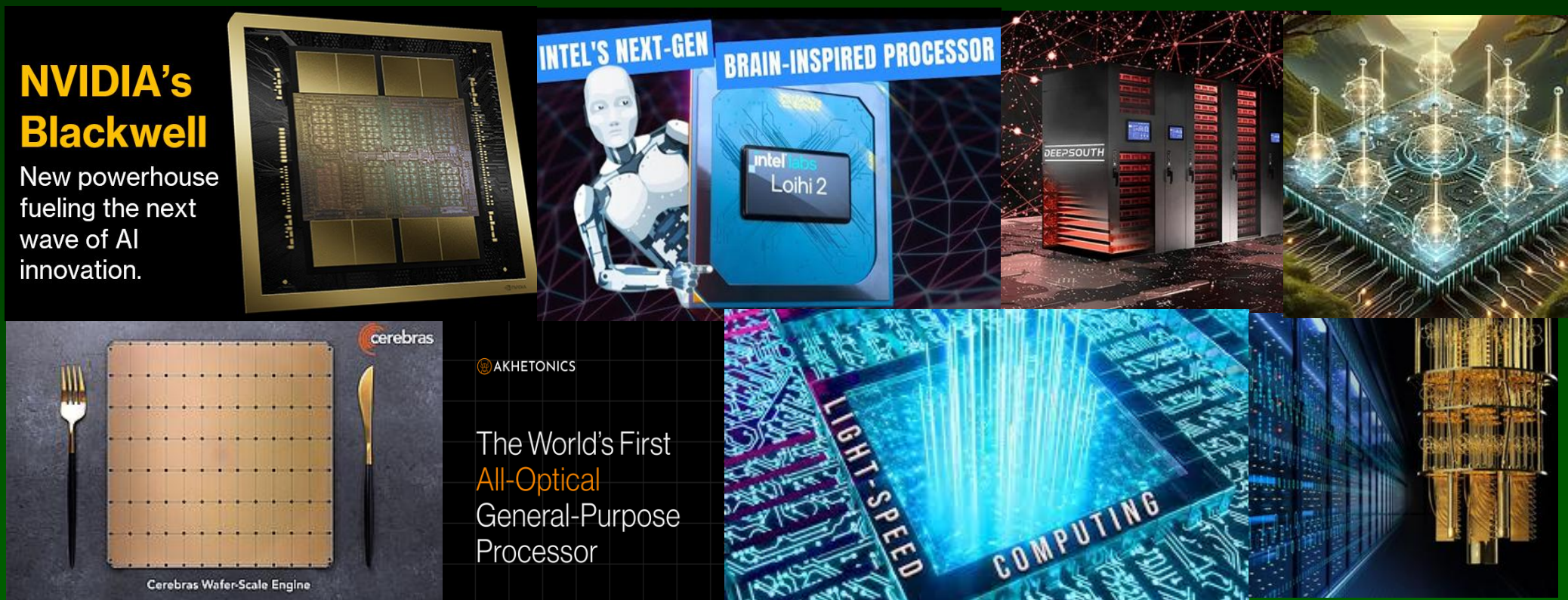


Neuromorficzna przyszłość

- Samsung Dynamic Vision Sensor (DVS) jest w technologii TrueNorth.
- Pozwoli to na automatyzację bardzo wielu zawodów!



Sprzęt: szybkość, energia



Rośnie szybkość, spada zużycie energii, pojawiają się nowe sposoby robienia obliczeń.
Termodynamiczne obliczanie, potrzebna energia mniejsza 100 mln razy!
Większa pamięć, szybsze kojarzenie, brak ograniczeń biologicznych mózgow.
PCSS Poznań + UAM + UMK w projekcie PIAST EuroHPC.

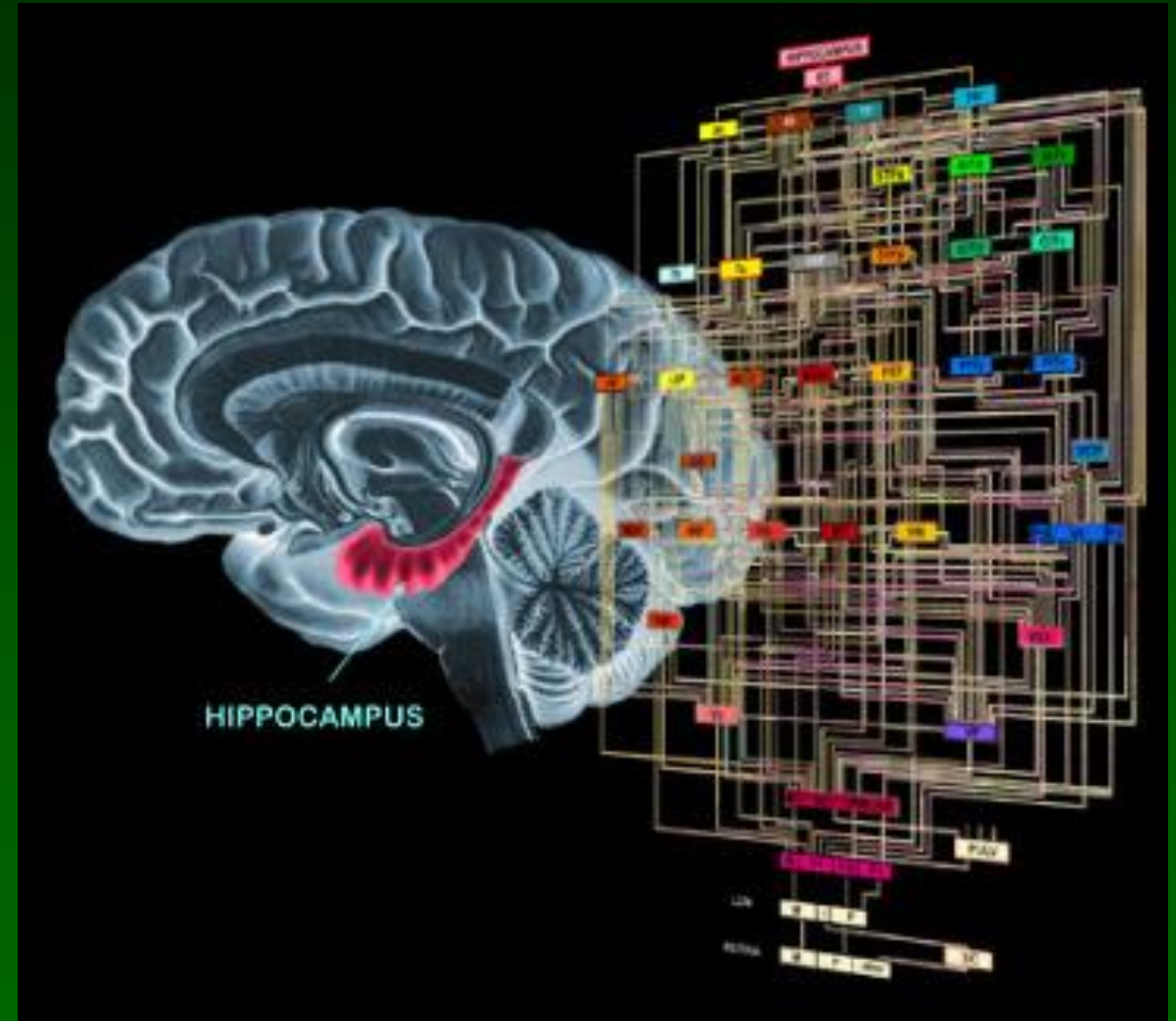
BICA, Brain-Inspired Cognitive Architecture

Mózgo-podobne architektury przetwarzania informacji.

Do zrozumienia potrzebny jest model odtwarzający funkcje, przeniesienie naszej wiedzy do neuronowego symulatora.

Dzięki modelom komputerowym możemy obecnie przewidywać jak będzie się zmieniać aktywność indywidualnego mózgu, przewidywać zachowanie człowieka.

Model TVB jest związany z HBP, ale głównie do zastosowań medycznych.



Co dalej?

- WD, liczne [referaty na temat AI](#), w tym porównania z działaniem ludzkich mózgów.
- NLP i uczenie maszynowe
- Powstało wiele ciekawych architektur kognitywnych, symbolicznych, neuronowych i hybrydowych, krótki opis jest w pracy [Cognitive architectures](#).
- Ciekawostki AI:
[Sztuczna inteligencja.org.pl](#)
[Flipboard AI](#)
- IEEE Spectrum: [AI and Robots](#)

Przykładowe pytania

- Co powinny uwzględniać zunifikowane teorie poznania?
- Jakie wymagania stoją przed teorią umysłu?
- Podać przykład praw, które teorie umysłu powinny wyjaśniać.
- Architektura Act-R; Warianty Act-R; Zastosowania Act-R.
- Definicja umysłu wg. Newella.
- Definicja inteligencji wg. Newella.
- Systemy oparte na wiedzy i systemy intencjonalne.
- Poziomy realizacji modeli: co można ze sobą porównać?
- Symbole, systemy symboliczne i systemy oparte na wiedzy.
- Architektura systemu symbolicznego.
- Wiedza i szukanie; komputery i umysły.
- Architektura SOAR; podstawowe mechanizmy, uczenie w SOAR.
- Idea robota Cog i robotyka rozwojowa.