Zachowania

* Opisz trzy poziomy teorii obliczeniowej.

* Podaj matematyczną definicję zachowania.

* Wyjaśnij w jednym lub dwóch zdaniach każdą z trzech kategorii zachowań (odruchowe, reaktywne, świadome).

*Bądź w stanie sklasyfikować zachowanie jako odruch, taksówki lub wzorzec działania stałego.

* Jeśli otrzymasz graficzną reprezentację zachowania, zidentyfikuj schematy behawioralne, percepcyjne i motoryczne i przetłumacz je na notację schematu S-R.

* Wyraź zestaw zachowań w notacji schematu S-R.

Przegląd

W Części 4 przedstawiono kanoniczną architekturę operacyjną z trzema warstwami, reprezentującymi różne style inteligencji. Najbardziej podstawową warstwą była warstwa reaktywna, złożona z zachowań zwierzęcych. Analogia reaktywnej inteligencji robota i inteligencji zwierząt rodzi dwa pytania, które zostaną omówione w tej części. Po pierwsze, czy warstwa reaktywna naprawdę przypomina etologię? To pytanie można również sformułować w następujący sposób: jak robot może być jak zwierzę i czy jest jakaś realna, konkretna wartość w myśleniu o robocie jako o zwierzęciu? Drugie, bardziej dosadne pytanie brzmi: Jeśli tak, to w jaki sposób przekuć ten chrupiący materiał dokumentalny o zwierzętach w formacie obliczeniowym? Analogie mogą być interesujące z teoretycznego punktu widzenia, ale programowanie robota wymaga konkretnych mechanizmów. Pomocne byłyby specyficzne mechanizmy powielania inteligencji zwierząt za pomocą konstruktów obliczeniowych. Rozdział odpowie na te pytania, przedstawiając najpierw historyczną motywację badaczy sztucznej inteligencji do badania zachowań zwierząt. Następnie w tej części przedstawimy teorię obliczeniową Marra, która została zaprojektowana w celu zapewnienia pomostu między inteligencją biologiczną a inteligencją opartą na krzemie oraz przyspieszenia procesu przenoszenia zasad biologicznych do konstrukcji programistycznych. Teoria obliczeniowa Marra ma trzy poziomy luźno odpowiadające architekturom operacyjnym (duży obraz), systemowym (potrzebne przekształcenia) i technicznym (rzeczywista implementacja). Po wprowadzeniu ram myślenia obliczeniowego o zachowaniach zwierząt, następny rozdział przedstawia rodzaje zachowań zwierząt, kładąc nacisk na aspekty teorii obliczeniowej poziomu 1. Zachowania zwierząt można modelować jako wysoce modułowe, ogólne, niezależne procesy, które są wyzwalane przez zdarzenia percepcyjne lub wewnętrzne lub stany obliczeniowe. Zachowania można wyrazić za pomocą teorii schematów, która przekształca dane wejściowe, wyjściowe i transformacje w bodźce, reakcje i zachowania. Schematy stają się namacalnymi modułami programistycznymi z notacją S-R, gdzie schematy są identyczne z obiektami w programowaniu obiektowym. Ten rozdział jest pierwszym z trzech rozdziałów o zachowaniach. Zachowania odruchowe, omówione w następnym rozdziale, nie wymagają modeli świata ani pamięci długotrwałej, co czyni je niezwykle wydajnymi obliczeniowo. Jednak ceną wysokiej modułowości i wydajności jest niedeterminizm algorytmiczny; ogólne zachowanie robota wyłania się, a nie jest programowane liniowo. Część 8 obejmie różne algorytmy koordynacji, za pomocą których indywidualne, niezależne zachowania tworzą ogólne zachowanie.

Motywacja do badania zachowań zwierząt

Postęp w robotyce w latach 70. był powolny. Najbardziej wpływowym robotem był Stanford Cart, opracowany przez Hansa Moraveca, który używał widoku stereo, aby widzieć i omijać przeszkody

wystające z ziemi. Pod koniec lat 70. i na początku 80. Michael Arbib zaczął badać modele inteligencji zwierząt z punktu widzenia nauk biologicznych i kognitywnych w nadziei uzyskania wglądu w to, czego brakowało w robotyce. Podczas gdy wielu robotyków było zafascynowanych zwierzętami, a wielu, w tym Moravec, wykorzystywało jakiś artefakt zwierzęcego zachowania do motywacji, nikt nie podszedł do tego pola z powagą i oddaniem Arbiba. Niemal w tym samym czasie ukazał się smukły tom Valentino Braitenberga zatytułowany Vehicles: Experiments in Synthetic Psychology. Była to seria gedanken lub koncepcyjnych eksperymentów myślowych, spekulujących, w jaki sposób inteligencja maszynowa może ewoluować. Braitenberg zaczął od Pojazdu 1, który miał prostą parę czujnik termiczny-siłownik silnika. Pojazd mógł poruszać się szybciej w ciepłych obszarach i wolniej w zimnych obszarach. Kolejny, bardziej złożony pojazd miał dwie pary czujnik-silnik, po jednej z każdej strony pojazdu, imitujące mutację. W wyniku efektu napędu różnicowego, w którym ciepło po jednej stronie powodowałoby, że silnik obracałby się szybciej niż silnik po chłodniejszej stronie, Pojazd 2 może zawrócić, aby wrócić do zimnych obszarów. W całej książce Braitenberga każdy pojazd zwiększał złożoność. To nawarstwianie było intuicyjne i wydawało się naśladować zasady ewolucji u naczelnych. Pojazdy stały się kultowym klasykiem wśród robotyków, zwłaszcza w Europie. Wkrótce nowe pokolenie badaczy sztucznej inteligencji odpowiedziało na wezwanie syreny, aby zbadać lekcje, które można wyciągnąć z inteligencji biologicznej. Zaczęli badać nauki biologiczne w poszukiwaniu nowych zasad organizowania i metod wytwarzania inteligencji. Jak zobaczymy później, wysiłki te doprowadziłyby do paradygmatu reaktywnego. Ta Część próbuje przygotować grunt pod zrozumienie paradygmatu reaktywności poprzez przypomnienie wpływowych badań i odkryć oraz próbę przedstawienia ich w świetle tego, jak mogą przyczynić się do inteligencji robotów. Dlaczego robotycy powinni zgłębiać biologię, etologię, psychologię poznawczą i inne nauki biologiczne? Ludzie mają tendencję do sprzeciwiania się uwzględnianiu inteligencji biologicznej podczas projektowania robotów, posługując się analogią, że samoloty nie trzepoczą skrzydłami. Kontrargumentem jest to, że prawie wszystko inne w aerodynamice samolotu powiela skrzydło ptaka. Prawie wszystkie ruchome powierzchnie na skrzydle samolotu pełnią te same funkcje, co części ptasiego skrzydła. Postępy w aeronautyce nastąpiły, gdy bracia Wright i inni przechwycili zasady aerodynamiczne z obserwacji lotu ptaków. Po ustaleniu zasad lotu można było zaprojektować systemy mechaniczne, które byłyby zgodne z tymi zasadami i spełniały te same funkcje, ale niekoniecznie w taki sam sposób, jak systemy biologiczne. Argument „samoloty nie trzepoczą skrzydłami" okazuje się jeszcze mniej przekonujący dla systemów komputerowych. Zwierzęta wykorzystują wrodzone zdolności; roboty polegają na skompilowanych programach. Robotycy AI często zwracają się do nauk biologicznych z różnych powodów. Po pierwsze, zwierzęta i ludzie dostarczają dowodów na istnienie różnych aspektów inteligencji. Często pomaga naukowcowi wiedzieć, że zwierzę może wykonać określone zadanie, nawet jeśli nie wiadomo, jak zwierzę to robi, ponieważ naukowiec przynajmniej wie, że jest to możliwe. Na przykład kwestia łączenia informacji z wielu czujników (fuzja czujników) jest od lat otwartą kwestią. W pewnym momencie opublikowano artykuły zniechęcające robotyków do badania fuzji czujników, ponieważ łączenie czujników było zjawiskiem, które brzmiało rozsądnie, ale w rzeczywistości nie miało podstaw. Dodatkowe badania wykazały, że zwierzęta (w tym ludzie) dokonują fuzji czujników, chociaż za pomocą zaskakująco innych mechanizmów niż większość badaczy. Drugim powodem zgłębiania nauk biologicznych jest to, że zwierzęta żyją w otwartym świecie, a robotycy chcieli przezwyciężyć założenie o zamkniętym świecie, które przysparzało Shakeyowi tak wielu problemów. Wiele „prostych" zwierząt, takich jak owady, ryby i żaby, wykazuje inteligentne zachowanie, ale ma niewiele mózgu lub nie ma go wcale. Dlatego muszą robić coś, co pozwala uniknąć problemu z ramkami.

Agenci i Teoria obliczeniowa Marra

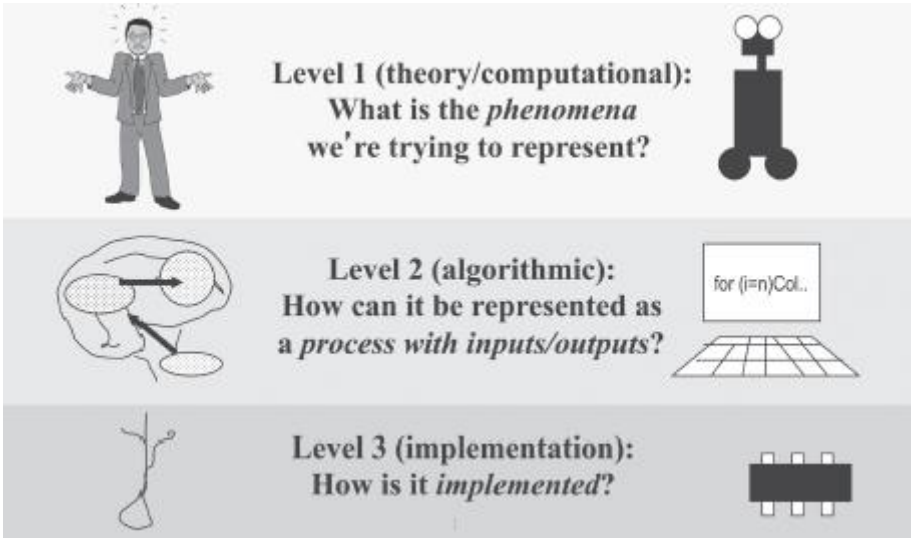
TEORIA OBLICZANIA

Nawet jeśli rozsądne wydaje się badanie nauk biologicznych i kognitywnych w celu uzyskania wglądu w inteligencję, jak możemy porównać tak różne systemy: formy „życia" węgla i krzemu? Jednym z potężnych sposobów konceptualizacji różnych systemów jest myślenie o abstrakcyjnym inteligentnym systemie jako o agencie. W terminologii programowania obiektowego „agent" jest nadklasą, a od niej wywodzą się klasy agentów „osoby", „zwierzę" i „robota". Oczywiście samo nazywanie zwierząt, robotów i inteligentnych pakietów oprogramowania „agentami" nie sprawia, że podobieństwa i powiązania między inteligencjami stają się jaśniejsze. Jednym z przydatnych sposobów przeglądania korespondencji jest decydowanie, na jakim poziomie te podmioty mają coś wspólnego. Zestaw poziomów wspólności prowadzi do tego, co często nazywa się teorią obliczeniową120, zgodnie z definicją Davida Marra. Marr był neurofizjologiem, który próbował przekształcić biologiczne procesy widzenia w nowe techniki widzenia komputerowego. Poziomy w teorii obliczeniowej można znacznie uprościć jako:

Poziom 1: Dowód istnienia tego, co można/należy zrobić. Załóżmy na przykład, że robotnik jest zainteresowany zbudowaniem robota do poszukiwania ocalałych uwięzionych w budynku po trzęsieniu ziemi. Robotyk może rozważyć zwierzęta, które szukają ludzi. Jak wie każdy, kto był na kempingu, komary są bardzo dobre w znajdowaniu ludzi. Komary dostarczają dowodu istnienia na to, że prosty obliczeniowo agent jest w stanie znaleźć człowieka za pomocą ciepła i dwutlenku węgla. Na poziomie 1 agenci mogą mieć wspólny cel lub funkcjonalność.

Poziom 2: Dekompozycja „co" na dane wejściowe, wyjściowe i przekształcenia. Ten poziom można traktować jako tworzenie schematu blokowego „czarnych skrzynek", aby faktycznie osiągnąć „co" z poziomu 1. Każda skrzynka reprezentuje przekształcenie wejścia w wyjście. Wracając do przykładu komara, robotnik może zdać sobie sprawę z biologii, że komar znajduje ludzi, kierując się w ciepło człowieka (lub innego stałocieplnego zwierzęcia). Jeśli komar wyczuje gorący obszar, leci w jego kierunku. Robotyk może zamodelować ten proces jako: wejście=obraz termiczny, wyjście=polecenie sterujące. „Czarna skrzynka" to symbol zastępczy mechanizmu, którego komar używa do przekształcania danych wejściowych w dane wyjściowe. Komar może pobierać środek ciężkości obrazu termicznego (centroid ważony ciepłem w każdym obszarze obrazu) i kierować się do tego obszaru. Jeśli gorący obszar się poruszy, obraz termiczny zmieni się przy następnej aktualizacji sensorycznej i zostanie wygenerowane nowe polecenie sterowania. Mechanizm ten może nie odpowiadać dokładnie temu, w jaki komar faktycznie generuje polecenia sterujące, ale sugeruje, w jaki sposób robot mógłby powielić tę funkcjonalność. Zauważ też, że skupiając się na procesie, a nie na wdrożeniu, robotnik nie musi brać pod uwagę, że komary latają, ponieważ robot poszukiwawczo-ratowniczy może mieć koła zamiast skrzydeł. Na poziomie 2 agenci mogą wykazywać wspólne procesy.

Poziom 3: Jak wdrożyć proces. Ten poziom teorii obliczeniowej koncentruje się na opisaniu sposobu implementacji każdej transformacji lub czarnej skrzynki. Na przykład u komara polecenia sterujące mogą być realizowane za pomocą specjalnego rodzaju sieci neuronowej, podczas gdy w robocie implementacja może obejmować algorytm obliczający kąt między środkiem ciepła a miejscem, w którym aktualnie wskazuje robot. Podobnie badacz zainteresowany czujnikami termicznymi może zbadać komara, aby zobaczyć, w jaki sposób jest on w stanie wykryć różnice temperatur u tak małego owada; elektromechaniczne czujniki termiczne ważą blisko funta! Na poziomie 3 agenci mogą mieć niewielką lub żadną cechę wspólną w rzeczywistej fizycznej implementacji funkcjonalności określonej na poziomie 1. Rysunek



ilustruje ogólną ideę teorii obliczeniowej. Najwyższy, najbardziej abstrakcyjny poziom może odpowiadać zjawisku, które ludzie potrafią zrobić, takiemu jak rozpoznanie domu na zdjęciu lub domu na podstawie kreskówek. Drugi poziom doprecyzowuje opis zjawiska w serię transformacji, które opisują, jakie dane z czujników wchodzą do mózgu, które obszary mózgu są aktywowane i wytwarzają dane wyjściowe prowadzące do ostatecznego wyniku rozpoznania domu. Obszary mózgu odpowiadają modułom lub algorytmom w programie komputerowym. Najniższy poziom opisuje aktywność każdego obszaru mózgu pod względem określonych typów neuronów i sposobu ich połączenia. Ten poziom aktywności w robocie można wdrożyć na dedykowanym komputerze

Powinno być jasne, że poziomy 1 i 2 są wystarczająco abstrakcyjne, aby zastosować je do dowolnego agenta. Dopiero na Poziomie 3 naprawdę pojawiają się różnice między agentem zrobotyzowanym a biologicznym. Niektórzy robotycy aktywnie próbują naśladować biologię, odtwarzając fizjologię i mechanizmy neuronalne. (Większość robotyków jest zaznajomiona z biologią i etologią, ale nie próbuje tworzyć dokładnych duplikatów natury.) Praca Roberta Fulla nad analizą, jak gekony przylegają do ścian podczas wspinania się i jak działają nogi karalucha, dokonały wielkich przełomów w poruszaniu się robotów. Ogólnie rzecz biorąc, powielanie sposobu, w jaki czynnik biologiczny coś robi, może nie być możliwe, a nawet pożądane. Większość robotyków nie dąży do dokładnego odtworzenia inteligencji zwierząt, mimo że wielu buduje stworzenia przypominające zwierzęta, jak to widzą przypominające owady Czyngis lub roboty humanoidalne. Ale skupiając się na poziomie 2 obliczeniowej teorii inteligencji, robotycy mogą uzyskać wgląd w to, jak organizować inteligencję

Przykład teorii obliczeniowej: Rana Computatrix

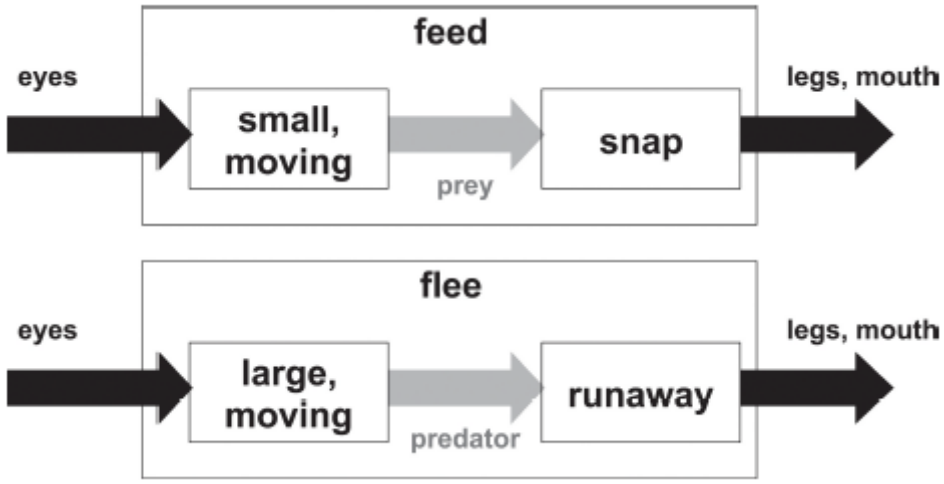
Rana computatrix jest najwcześniejszym przykładem teorii obliczeniowej zastosowanej w robotyce AI. Pokazuje wartość teorii obliczeniowej. Jest to konkretny przykład tego, jak pewien badacz przełożył organizację biologiczną na obiekty obliczeniowe. Posiadanie konkretnego przykładu jest pomocne w zrozumieniu procesu obliczeniowego, ponieważ różni badacze mogą różnie interpretować biologiczną organizację zjawiska, a tym samym proponować różne teorie. Komputer obliczeniowy Rana jest również interesujący, ponieważ Arbib użył szczególnej struktury obliczeniowej, zwanej schematami, do matematycznego wyrażenia relacji między tymi obiektami. Schematy będą podstawową strukturą programowania warstwy reaktywnej w inteligentnych robotach i zostaną formalnie przedstawione w dalszej części.

RANA COMPUTATRIX

W latach 80. Arbib i współpracownicy skonstruowali komputerowe modele wizualnie sterowanych zachowań żab i ropuch, aby lepiej zrozumieć inteligencję. Nazwali swój model rana computatrix (Rana to rodzaj żab). Począwszy od pierwszego poziomu teorii obliczeniowej, ropuchy i żaby mają dwa wizualne zachowania: żerują i uciekają. Poziom 2 wymaga wyrażenia zachowań paszy i ucieczki jako danych wejściowych, wyjściowych i transformacji. Na podstawie badań biologicznych żaby można scharakteryzować jako zdolne do wykrywania tylko dwóch wizualnych sygnałów wejściowych: małych poruszających się plam w polu widzenia lub dużych ruchomych plam. Małe ruchome plamy są danymi wejściowymi do zachowania karmienia, które generują dane wyjściowe. Wynik jest taki, że żaba orientuje się w kierunku plamki (co nazywa się zachowaniem taksówki), a następnie rzuca się na niego. Mała, poruszająca się kropla może być gumką ołówka machanego przed żabą, ale ponieważ pasuje do bodźca małej ruchomej kropli, żaba rzuci się na nią, jakby była jadalną muchą. Jeśli przedmiot powiązany z wizualną plamą okaże się nie muchą, żaba może go wypluć. Duże poruszające się plamy są sygnałem wejściowym do zachowania ucieczki, powodując, że żaba odskakuje.

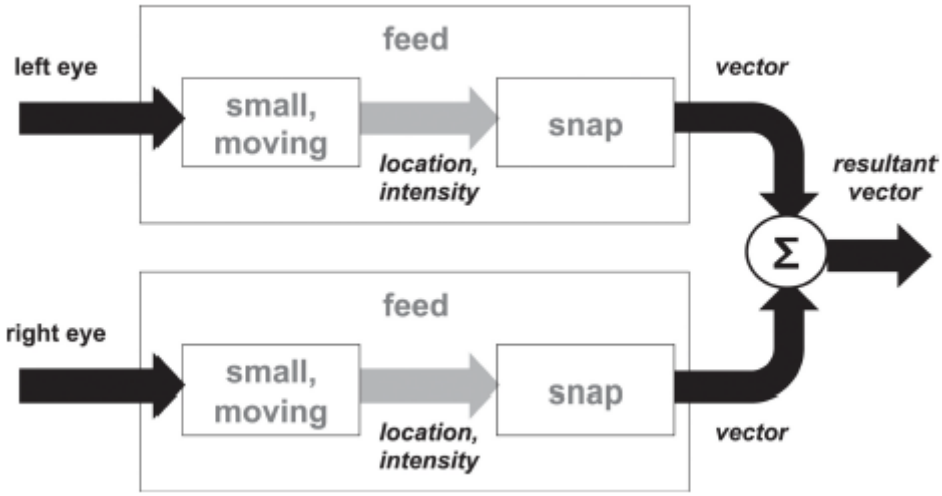
PERCEPT

Rysunek



pokazuje graficzną reprezentację tej konceptualizacji poziomu 2, w której żaba dostrzega małą poruszającą się plamę w swoim polu widzenia, tym samym dając przykład zachowania podczas karmienia, które powoduje, że żaba zwraca się w kierunku tego bodźca i rzuca się na niego. Zauważ, że bodziec zapewnia lokalizację w polu widzenia. Wyjście bodźca nazywa się percepcją; przypisujemy do niego etykietę „mucha" lub „ofiara", ale nie zachodzi przetwarzanie uziemiające symbole. Jest tylko bodziec połączony z odpowiedzią. Kiedy żaba dostrzega dużą poruszającą się plamę, pojawia się zachowanie ucieczki i żaba odskakuje. Ten bodziec można nazwać „drapieżnikiem", ale etykieta jest dowolna, jak wybór nazwy zmiennej w programie. W robotyce zmienna percepcyjna jest ogólnie określana jako percept. Grupa Arbib badała również poziom 3 w teorii obliczeniowej.9 Zaimplementowali zachowanie taksówek jako pole wektorowe: rana computatrix dosłownie odczuwałaby siłę przyciągania wzdłuż kierunku lotu. Względny kierunek i intensywność bodźca przedstawiono jako wektor. Kierunek wskazywał, gdzie rana musi się skręcić, a wielkość wskazywała siłę, którą należy zastosować do trzaskania. Słabszy bodziec małej poruszającej się plamki może wskazywać na muchę w dalszej odległości, a zatem żaba musiałaby mocniej trzasnąć. Poziom 3 jest pokazany na rysunku

, gdzie percept (położenie środka ciężkości plamki i intensywność bodźca) i wyjście (wektor) są dokładniej zdefiniowane, choć nadal nie są wyrażone w prawdziwym formacie inżynierii oprogramowania. Chodzi o to, że łatwo sobie wyobrazić, że istnieje wywołanie funkcji do wykrywania małych i poruszających się na obrazie, odpowiadającej prymitywowi SENSE, oraz inna funkcja przekształcająca kierunek i intensywność w wektor, odpowiadająca prymitywowi ACT . Szczególnie interesujące jest to, że model obliczeniowy może pomóc biologom. Model wyjaśnił obserwacje Ingle'a dotyczące tego, co czasami się dzieje, gdy ropucha widzi dwie muchy naraz



Ropuchy mają oczy skierowane w różne strony. Ropucha może jednocześnie widzieć muchę lewym okiem i inną muchę prawym okiem. W tej sytuacji każda mucha wyzwala osobną instancję zachowania żywieniowego. Każde zachowanie tworzy wektor, do którego ropucha musi się zwrócić, aby złapać muchę, nie zdając sobie sprawy, że istnieje inne zachowanie. Zgodnie z implementacją pola wektorowego modelu schematu, ropucha otrzymuje teraz dwa wejścia wektorowe zamiast jednego. Co robić? Ponieważ implementacja obliczeniowa Rana wykorzystuje metodologię wektorową, oba wektory są sumowane, co powoduje wygenerowanie trzeciego wektora pomiędzy dwoma oryginalnymi! Model przewidywał, że ropucha nie będzie trzaskać ani mucha, tylko trzaśnie pomiędzy, co zaobserwowano w przypadku prawdziwych ropuch. Nieoczekiwana interakcja dwóch niezależnych instancji prawdopodobnie nie jest istotną wadą ropuchy, ponieważ jeśli w tak bliskiej odległości znajdują się dwie muchy, w końcu jedna z nich wróci w zasięg.

ZACHOWANIE WYJAŚNIAJĄCE

Ten przykład ilustruje trzy ważne lekcje dla robotyki. Po pierwsze, potwierdza ideę teorii obliczeniowej, w której funkcjonalność komputera i zwierzęcia może być równoważna i wykorzystywana do inspirowania odkryć. Podsumowując, równoważność między komputerem a zwierzęciem jest następująca: pojęcie zachowań jest na poziomie 1 teorii obliczeniowej, teoria schematów wyraża poziom 2, a implementacja pola wektorowego działania motorycznego jest ujęta na poziomie 3. Po

drugie, przykład pokazuje właściwość zachowań emergentnych, gdzie agent wydaje się robić coś dość złożonego, ale obserwowane zachowanie jest tak naprawdę tylko wynikiem interakcji między prostymi modułami. Ilustruje również jeden z powodów, dla których symulowanie wyłaniającego się zachowania może być trudne; projektantowi często trudno jest wyobrazić sobie wyjątkowe przypadki lub „narożne przypadki", takie jak to, co się dzieje, gdy ropucha widzi jednocześnie dwie muchy. Po trzecie, przykład pokazuje również, w jaki sposób zachowania odpowiadają zasadom programowania obiektowego.

Zachowania zwierząt

ZACHOWANIE

Rana computatrix ilustruje, w jaki sposób teoria obliczeniowa obejmuje biologiczną i sztuczną inteligencję. Inteligencja biologiczna jest głównym źródłem inspiracji dla warstwy reaktywnej w kanonicznej architekturze operacyjnej i robotów opartych na zachowaniu. Dlatego warto wprowadzić definicje z etologii, które przydadzą się później. Zachowanie to mapowanie bodźców zmysłowych na wzór działań motorycznych, które są następnie wykorzystywane do wykonania zadania.

ZACHOWANIE REFLEKSYJNE

BODZIEC-ODPOWIEDŹ

ZACHOWANIE REAKTYWNE

ŚWIADOME ZACHOWANIE

Zachowania można podzielić na trzy szerokie kategorie. Zachowania odruchowe to reakcje na bodziec (S-R); na przykład, gdy twoje kolano zostanie uderzone, podskakuje w górę. Zasadniczo zachowania odruchowe są „na stałe"; są to obwody neuronowe, które zapewniają, że bodziec jest bezpośrednio połączony z odpowiedzią, aby wytworzyć najszybszy czas odpowiedzi. Zachowania reaktywne są wyuczane, a następnie konsolidowane do miejsca, w którym można je wykonać bez świadomego myślenia. Każde zachowanie, które wiąże się z tym, co w sporcie określa się mianem „pamięci mięśniowej", jest zwykle zachowaniem reaktywnym (np. jazda na rowerze lub nartach). Zachowania reaktywne można również zmienić poprzez świadome myślenie; rowerzysta jadący po bardzo wąskim moście może „zwrócić uwagę" na równowagę, ułożenie rąk, ułożenie stóp na korbach i gdzie należy zwrócić uwagę na punkt niezamierzonego pogorszenia wydajności. Świadome zachowania są przemyślane (np. składanie zestawu robota, łączenie ze sobą wcześniej wypracowanych zachowań itp.). Zrozumienie różnic między trzema kategoriami zachowań jest warte uwagi z kilku powodów. Po pierwsze, warstwa reaktywna intensywnie wykorzystuje zachowania odruchowe, do tego stopnia, że niektóre architektury nazywają zachowanie robota zachowaniem tylko wtedy, gdy jest to reakcja na bodziec. Po drugie, kategorie mogą pomóc projektantowi określić, jakiego rodzaju zachowania użyć, co prowadzi do wglądu w odpowiednią implementację. Po trzecie, użycie słowa „reaktywny" w etologii jest sprzeczne ze sposobem, w jaki to słowo jest używane w robotyce. W etologii zachowanie reaktywne oznacza wyuczone zachowania lub umiejętność; w robotyce oznacza zachowanie odruchowe. Jeśli robotycy nie są świadomi tych różnic, może być trudno czytać literaturę etologiczną lub sztuczną inteligencję bez pomyłki.

Zachowania odruchowe

Zachowania odruchowe są szczególnie interesujące, ponieważ nie implikują potrzeby poznania: jeśli to wyczuwasz, to robisz. W przypadku robota oznacza to przewodową reakcję, eliminującą obliczenia i gwarantującą szybkość. Rzeczywiście, wiele robotów kitowych lub hobbystycznych działa odruchowo, reprezentowanych przez obwody. Zachowania odruchowe można dalej podzielić na trzy kategorie:

ODRUCHY

1. odruchy: gdzie reakcja trwa tylko tak długo, jak bodziec, a reakcja jest proporcjonalna do natężenia bodźca.

TAKSJA

2. podatki: gdzie odpowiedzią jest przejście do określonej orientacji. Młode żółwie wykazują tropotaksję; wykluwają się w nocy, ukryte przed ptakami przybrzeżnymi, które zwykle je zjadają, i poruszają się w najjaśniejszym świetle. Do niedawna najjaśniejszym światłem byłby ocean odbijający księżyc, ale ingerencja człowieka to zmieniła. Właściciele posiadłości przy plaży na Florydzie muszą teraz wyłączać oświetlenie zewnętrzne podczas sezonu lęgowego, aby uniknąć sytuacji, w której ich światła stają się konkurencyjnym źródłem tropotaksji. Mrówki wykazują szczególną taksówkę znaną jako chemotaksja; podążają śladami feromonów.

WZORY STAŁYCH DZIAŁAŃ

3. wzorce o ustalonym działaniu: gdzie reakcja trwa dłużej niż bodziec. Ten rodzaj zachowania jest pomocny przy odwracaniu się i ucieczce przed drapieżnikami. Należy pamiętać, że taksówka może skutkować dowolnym wzorcem ruchu w stosunku do bodźca, a nie tylko ruchem w jego kierunku. Powyższe kategorie nie wykluczają się wzajemnie. Na przykład zwierzę przechodzące przez skały lub las z drzewami blokującymi jego widok może upierać się (wzory stałych działań) w orientowaniu się na ostatnią wykrytą lokalizację źródła pożywienia (taksówki), zanim straci je z oczu.

IDIOTETYCZNA

ALETOTYCZNY

Ścisłe połączenie działania i percepcji można często określić ilościowo za pomocą wyrażeń matematycznych. Przykładem tego jest orientacja w skalarach. Aby pływać w pozycji pionowej, skalary posługują się wewnętrznym (idiotetycznym) zmysłem grawitacji w połączeniu ze zmysłem wzroku (alotetycznym), aby zobaczyć zewnętrzną percepcję linii horyzontu wody. Jeśli ryba zostanie umieszczona w zbiorniku z pryzmatami, które powodują, że linia horyzontu pojawia się pod kątem, skalary będą pływać z wykrzywionym nosem. Przy bliższym przyjrzeniu się, kąt, pod którym pływa skalary, jest sumą wektorów wektora równoległego do grawitacji i wektora prostopadłego do postrzeganej linii horyzontu! Zdolność do ilościowego określenia zachowania zwierząt sugeruje, że można napisać programy komputerowe, które robią to samo.

Teoria schematów

Teoria schematów zapewnia pomocny sposób tłumaczenia spostrzeżeń z zachowań zwierząt na format programowania obiektowego. Jak widać w przypadku rana computatrix, komponenty SENSE i ACT zachowania paszy i ucieczki są modułami wielokrotnego użytku. Rzeczywiście, moduły SENSE i ACT to szablony, które można tworzyć z różnymi warunkami (np. Predator, Prey), aby tworzyć różne akcje. Psychologowie od początku XX wieku stosowali teorię schematów do formalnego modelowania zachowań i ich komponentów.

Schematy jako obiekty

SCHEMAT

SCHEMAT KLASA

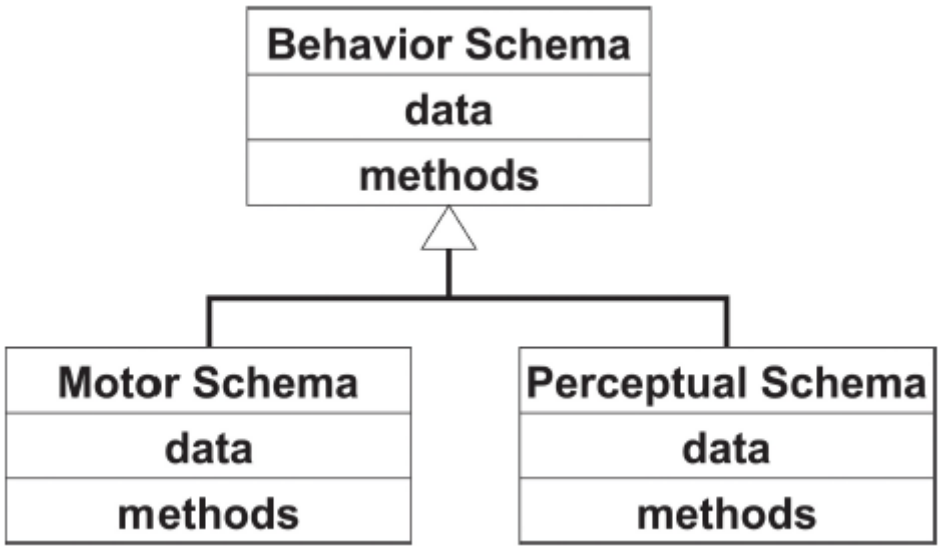
Schemat jest używany w kognitywistyce i etologii w odniesieniu do określonego zorganizowanego sposobu postrzegania poznawczego i reagowania na złożoną sytuację lub zestaw bodźców. Schemat składa się z dwóch części: wiedzy o tym, jak działać i/lub postrzegać (wiedza, struktury danych, modele) oraz procesu obliczeniowego (algorytmu lub procedury), którego używa do wykonania działania. Idea schematu ładnie odwzorowuje się na klasę w programowaniu obiektowym (OOP). Klasa schematu w C++ lub Javie zawiera zarówno dane (wiedzę, modele, wydania) jak i metody (algorytmy postrzegania i działania), jak pokazano poniżej.

Behavior::Schema

Data	
Methods	perceptual_schema() motor_schema()

INSTANCJA SCHEMATU (SI)

Schemat to ogólny szablon do wykonywania pewnych czynności, takich jak jazda na rowerze. Jest to szablon, ponieważ osoba może jeździć na różnych rowerach bez rozpoczynania całego procesu uczenia się. Ponieważ schemat jest sparametryzowany jak klasa, parametry (typ roweru, wysokość siodełka roweru, pozycja kierownicy) mogą być dostarczone do obiektu w momencie tworzenia instancji (gdy obiekt jest tworzony z klasy). Podobnie jak w przypadku programowania obiektowego, tworzenie określonego schematu jest instancją, która jest konkretnie nazywana instancją schematu (SI). Instancja schematu to obiekt, który jest konstruowany z dowolnymi parametrami potrzebnymi do dostosowania go do sytuacji. Na przykład może istnieć schemat move_to_food, w którym agent zawsze nawiguje w linii prostej do jedzenia. Zauważ, że „zawsze głowa w linii prostej" to szablon aktywności, a jako szablon jest to algorytm wielokrotnego użytku do sterowania ruchem. Jednak „zawsze głowa w linii prostej" to tylko metoda. Dopóki schemat move_to_food nie zostanie utworzony, nie ma określonego celu (np. batonika na stole). Ten sam schemat można utworzyć w przypadku przejścia na kanapkę. Schemat można utworzyć z parametrami, takimi jak stopień głodu agenta lub strach, że ktoś inny weźmie najpierw batonika. Te parametry mogą spowodować, że agent wykona ten sam szablon działania, ale szybciej. Rysunek ilustruje, w jaki sposób podstawowe zachowanie, motorykę i schemat percepcyjny można wyrazić w zunifikowanym języku modeli.

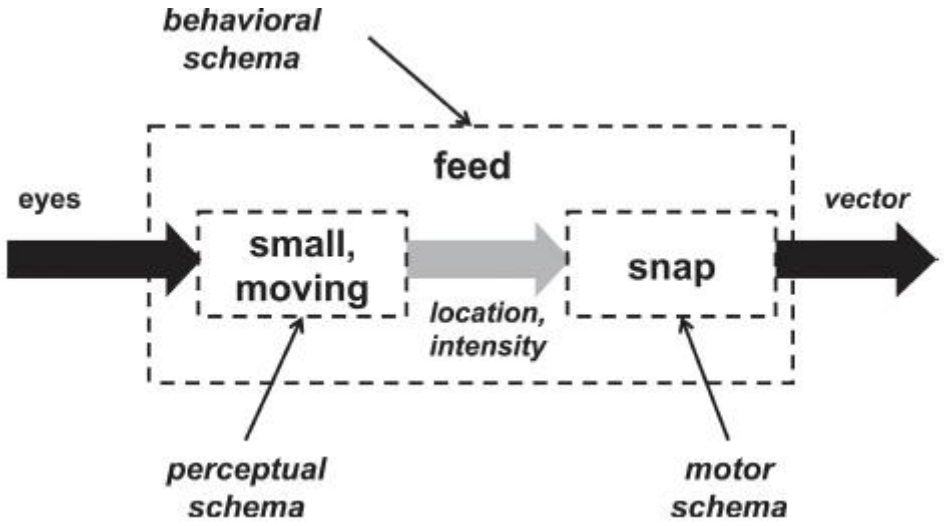


Zachowania i teoria schematów

SCHEMAT SILNIKA

SCHEMAT PERCEPTUALNY

W arbibskim zastosowaniu teorii schematów w obliczeniowej teorii inteligencji zachowanie jest schematem, który składa się ze schematu motorycznego i schematu percepcyjnego .



Schemat motoryczny reprezentuje szablon dla aktywności fizycznej, a schemat percepcyjny obejmuje bodziec i wszelkie opóźnienia w działaniu stałym. Zasadniczo koncepcje schematów percepcyjnych i motorycznych pasują do pojęć w etologii i psychologii poznawczej w następujący sposób:

* Zachowanie pobiera bodźce sensoryczne i jako wynik wytwarza działania motoryczne.

* Przekształcenie bodźców czuciowych w wyniki działania motorycznego można podzielić na dwa podprocesy: schemat percepcyjny i schemat motoryczny.

* Zachowanie może być reprezentowane jako schemat, który jest zasadniczo konstrukcją programistyczną zorientowaną obiektowo.

* Schemat jest wielokrotnego użytku.

W terminologii OOP, schemat motoryczny i klasy schematu percepcyjnego wywodzą się z klasy schematu. Zachowanie prymitywne ma tylko jeden motor i jeden schemat percepcyjny. Jako obiekty, schematy mogą zawierać dwa rodzaje wiedzy:

1. Wiedza o schemacie, czyli dane lokalne w żargonie OOP. Opisane dotychczas schematy nie wykorzystywały żadnych danych lokalnych, ale jest to możliwe.

2. Wiedza proceduralna, czyli metody w żargonie OOP. Każdy ze schematów posiada funkcję, która wykonuje niezbędne obliczenia.

Zgodnie z OOP, instancja schematu jest specyficzna dla sytuacji i równoważna instancji w OOP. Podobnie schemat może składać się z innych schematów, na przykład schemat behawioralny składa się ze schematu percepcyjnego i motorycznego. Zauważ, że koncepcja schematu wymusza istnienie bibliotek schematów behawioralnych, schematów percepcyjnych i schematów motorycznych. Z punktu widzenia architektury systemów, moduły te mogą być wyrażane jako schematy i umieszczane w odpowiednich podsystemach, takich jak Percepcja i Silnik, z których każdy ma bibliotekę schematów

do tworzenia w razie potrzeby. Z punktu widzenia implementacji (lub perspektywy architektury technicznej) schematy są obiektami.

S-R: notacja schematu

MATEMATYCZNA DEFINICJA ZACHOWANIA

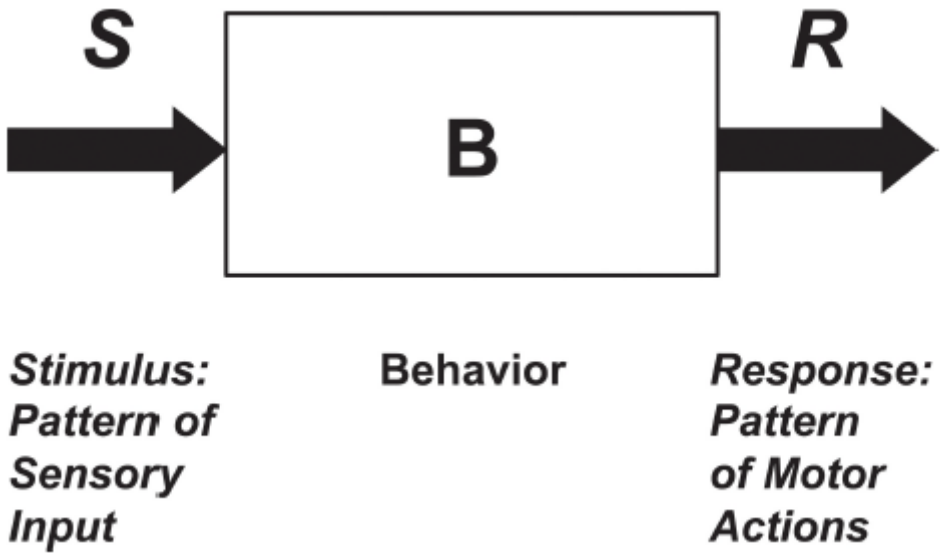
Matematyczna definicja zachowania polega na odwzorowaniu bodźców zmysłowych na wzorzec działań motorycznych, które są następnie wykorzystywane do wykonania zadania:

$$\{B : S \rightarrow R\}$$

gdzie B reprezentuje zachowanie, które pobiera dane wejściowe S („S" dla wyczuwania lub bodźca) i generuje wynik R („R" dla reakcji lub odpowiedzi, co jest tym samym co działanie). Definicja zachowania w teorii schematów jest napisana jako:

$$B[S] = R$$

gdzie B, S, R są funkcjami. S to funkcja, która pobiera dane z czujnika jako dane wejściowe i zwraca percept jako dane wyjściowe. R to funkcja, która przyjmuje percept jako dane wejściowe i zwraca akcję jako dane wyjściowe. Równanie powyższe wyraża sprzężenie SENSE-ACT prymitywów robotów. Konotuje samoistną pętlę, która jest niezależna od innych zachowań; to jest kompletne. Graficzna reprezentacja zachowania jest pokazana na rysunku 6.9.



Zachowania związane z karmieniem i ucieczką w obliczeniach rany są wyrażone w notacji schematu S-R jako:

$$\begin{aligned} \text{Feed[prey]} &= \text{Snap} \\ \text{Flee[predator]} &= \text{Runaway} \end{aligned}$$

Zwróć uwagę, że feed i flee to nazwy funkcji (schematów zachowań) dla zachowania. zdobycz i drapieżnik to nazwy funkcji (schematów percepcyjnych), które wywołuje zachowanie. Każde zachowanie aktywuje funkcję (schemat silnika), przyciąganą lub uciekającą. Jak omówiono wcześniej w tym rozdziale, nazwy takie jak pasza, ucieczka, zdobycz, drapieżnik, przyłapanie i ucieczka są

arbitralnymi etykietami; nazwy istnieją, ponieważ programy wymagają etykiet dla funkcji i zmiennych. Nie ma wyraźnego modelu drapieżnika lub ofiary, są to po prostu wygodna abstrakcja „dużej ruchomej kropli" i „małej ruchomej kropli". Inny programista może zadeklarować postrzeganie zdobyczy z nazwą latać w zależności od preferencji. Zauważ również, że funkcja zachowania używa nawiasów [ ] zamiast (). Jest to celowe, ponieważ ogólne zachowanie agenta może być wynikiem wielu współbieżnych zachowań, które są reprezentowane jako macierz schematów behawioralnych.

$$B\begin{bmatrix} prey \\ predator \end{bmatrix} = \begin{bmatrix} \beta_{feed} \\ \beta_{flee} \end{bmatrix}$$

Macierz schematów behawioralnych ma dwie odpowiadające sobie macierze reprezentujące powiązany bodziec, czyli schemat percepcyjny S, oraz odpowiedź, czyli schemat motoryczny R:

$$S = \begin{bmatrix} S_{prey} \\ S_{predator} \end{bmatrix}$$

$$R = \begin{bmatrix} R_{snap} \\ R_{runaway} \end{bmatrix}$$

Indeksy dolne zapewniają wewnętrzną etykietę działania schematu. Sprey to schemat percepcyjny, który pobiera dane sensoryczne i znajduje ofiarę. Jednak te wewnętrzne etykiety są nadal niejasne z punktu widzenia programowania. Dlatego rzeczywiste wywołania funkcji można zastąpić elementami w macierzach. Rozważmy

$$S = \begin{bmatrix} prey\_dir = blob\_analysis(vision, SMALL, MOVING) \\ predator\_dir = blob\_analysis(vision, LARGE, MOVING) \end{bmatrix}$$

Ta macierz może być interpretowana w następujący sposób: Agent ma dwa schematy percepcyjne, po jednym dla każdego zachowania. Schemat percepcyjny powiązany z zachowaniem kanału, β_{feed}, to funkcja blob_analysis. blob_analysis przyjmuje wizję jako określone dane wejściowe i małe, poruszając się jako parametry i tworzy percepcję względnego kierunku obiektu blob w polu widzenia agenta, co nazywa się prey_dir. β_{flee} korzysta również z funkcji blob_analysis, która przyjmuje wizję jako określone dane wejściowe, ale instancja wywoływana przez β_{flee} ma duże rozmiary, poruszając się jako parametry i generuje percepcję względnego kierunku obiektu blob w polu widzenia agenta, co nazywa się predator_dir. Dobra praktyka programistyczna polegająca na posiadaniu jednego obiektu ogólnego przeznaczenia blob_analysis oznacza, że technicznie rzecz biorąc, tak naprawdę istnieje tylko jeden schemat percepcyjny z dwoma instancjami. Instancje blob_analysis są niezależne i nie są ze sobą powiązane. Zwierzę może tylko szukać pożywienia, tylko szukać drapieżników lub jedno i drugie, ale indywidualne zachowania nie są świadome niczego poza zakresem ich wystąpienia. Podobnie dla schematu silnika:

$$R = \begin{bmatrix} snap(turn = prey\_dir), hop = NULL) \\ runaway(turn = (predator\_dir + 180), hop = FORWARD) \end{bmatrix}$$

Macierz reakcji behawioralnej lub działania może być interpretowana w następujący sposób: Obliczenia Rana mają dwa schematy motoryczne, po jednym dla każdego zachowania. Schemat silnika

powiązany z zachowaniem kanału, β_{feed}, to funkcja snap, która pobiera percept prey_dir ze skojarzonego schematu percepcyjnego, aby określić, gdzie wykonać snap. β_{feed} ma inny parametr, hop, ponieważ akcja jest kierunkiem i ruchem wzdłuż tego kierunku. Podczas karmienia rana computatrix powinna po prostu zwrócić się w kierunku bodźca i trzasnąć. Jednak w ucieczce powinien zarówno odwrócić się o 180 stopni od bodźca drapieżnika, jak i podskoczyć. Nazywanie funkcji jest dowolne, ale ogólna konwencja jest taka, że ogólny opis behawioralny, na przykład kanał lub ucieczka, jest powiązany z β, podczas gdy nazwa schematu silnika odzwierciedla konkretną akcję, na przykład przyciąganie lub uciekanie.

Podsumowanie

W tym rozdziale omówiono dwa pytania, jakie czytelnik może mieć na temat inteligencji inspirowanej biologicznie. Pierwsze postawione pytanie brzmiało: Czy warstwa reaktywna naprawdę przypomina etologię? Odpowiedź brzmi „tak", zarówno z perspektywy projektowania, jak i sposobu programowania za pomocą programowania obiektowego. Warstwa reaktywna opiera się na zachowaniach, które są podstawowym elementem inteligencji biologicznej. Zachowanie definiuje się jako mapowanie bodźców zmysłowych do wzorca działań motorycznych, które następnie są wykorzystywane do wykonania zadania. Drugie postawione pytanie brzmiało: Jeśli tak, to w jaki sposób przekuć ten chrupiący materiał dokumentalny o zwierzętach w formacie obliczeniowym? Odpowiedzią jest wykorzystanie teorii obliczeniowej Marra jako ogólnych ram do przenoszenia spostrzeżeń biologicznych do rzeczywistego kodu, a następnie wykorzystanie teorii schematów jako konkretnego, zorientowanego obiektowo sposobu przedstawiania i myślenia o zachowaniach.

Ważnymi atrybutami teorii schematów dla zachowań są:

* Teoria schematów służy do reprezentowania zachowań zwierząt i komputerów i wystarcza do opisania inteligencji na pierwszych dwóch poziomach teorii obliczeniowej. Należy zauważyć, że wybór pól wektorowych do implementacji zachowań związanych z karmieniem i ucieczką był na trzecim poziomie teorii obliczeniowej i można było wybrać inny mechanizm obliczeniowy.

* Schemat behawioralny składa się z co najmniej jednego schematu motorycznego i co najmniej jednego schematu percepcyjnego oraz lokalnej, specyficznej dla zachowania wiedzy o tym, jak koordynować wiele schematów składowych. Chociaż zostanie to omówione w dalszych rozdziałach, dla kompletności warto zauważyć, że jeden ze schematów motorycznych lub percepcyjnych w zachowaniu może mieć wartość NULL. Schemat NULL umożliwia agentowi obserwację bez działania (schemat silnika NULL) lub może działać w oparciu o wewnętrzne popędy, takie jak głód lub potrzeba reprodukcji (schemat percepcyjny NULL).

* Jednocześnie można utworzyć więcej niż jeden schemat zachowania, ale każdy schemat działa niezależnie. W rezultacie ogólne zachowanie agenta wyłania się jako interakcja tych niezależnych zachowań. Rodzaje interakcji niezależnych zachowań i sposób ich koordynacji znajduje się w części 8.

Chociaż rozdział odpowiedział na dwa pytania postawione we wstępie, treść rozdziału mogła wzbudzić więcej pytań niż odpowiedzi. Wprowadził teorię schematów i jej ekspresyjną moc tłumaczenia spostrzeżeń etologicznych na obiekty obliczeniowe, ale rozdział nie zagłębiał się w treść, w szczególności: Co dzieje się w schematach, zwłaszcza w schemacie percepcyjnym? Prawdziwym problemem jest to, czy potrzebna jest jakaś koordynacja, aby uzyskać przewidywalne, złożone, wyłaniające się zachowanie. Odpowiedź brzmi „tak", koordynacja jest konieczna.