

1 Reprezentacja wiedzy

Zachowanie inteligentne zależne jest od posiadanej wiedzy. Parę lat temu demonstrowano program komputerowy służący do tłumaczenia z angielskiego na chiński. Obecny był senator USA - zapytany o podanie jakiegoś zdania do tłumaczenia użył zdania idiomatycznego: „out of sight, out of mind”, co z oczu to z głowy. Program wypuścił parę chińskich znaków. Proszony o przetłumaczenie z powrotem na angielski komputer napisał: „Invisible idiot”, czyli niewidzialny idiota, co jest jednym z możliwych dosłownych tłumaczeń tego zwrotu. Chociaż jest to zdarzenie zabawne nie świadczy to o głupocie programu (akurat zwroty idiomatyczne łatwo jest tłumaczyć) ale o braku znajomości danego idiomu, braku wiedzy.

Reprezentacja wiedzy jest podstawową sprawą dla wszystkich mających się inteligentnie zachowywać systemów. Programy oparte na prostych algorytmach szukania, omówione w poprzednim rozdziale, nie potrafiły rozwiązać wielu zagadnień z braku wiedzy. Specjaliści od reprezentacji wiedzy nazywają się „inżynierami wiedzy” (knowledge engineers). W filozofii nauką o naturze wiedzy jest *epistemologia*. Można więc całą dziedzinę określić mianem eksperymentalnej epistemologii. Jak w komputerze reprezentować wiedzę?

Czym jest wiedza? Nad tym pytaniem debatuje się od bardzo dawna. W AI podchodzi się do tego problemu pragmatycznie: reprezentacja wiedzy to kombinacja struktur danych i procedur interpretacyjnych tak dobranych, że właściwie użyte prowadzić będą do inteligentnego zachowania. Same struktury danych nie są jeszcze wiedzą, tak jak sama encyklopedia nią nie jest, potrzebny jest jeszcze czytelnik, interpretator. Czasami mówi się o źródle wiedzy jako o wiedzy, ale to tylko nieścisłość językowa.

1.1 Rodzaje wiedzy:

Obiekty:

Myśląc o świecie zwykle myślimy o przedmiotach. Ptaki mają skrzydła. Bociany to ptaki. Zimą jest śnieg. Potrzebujemy metody do reprezentacji obiektów i kategorii obiektów.

Zdarzenia:

Oprócz obiektów mamy zdarzenia. Będzie padało. Wieje wiatr. Oprócz informacji o samych zdarzeniach trzeba znać następstwa przyczynowe i sekwencje czasową zdarzeń.

Umiejętności:

to znacznie trudniejszy typ wiedzy. Pewien garncarz odwiedzał dwór chińskiego cesarza i spotkał tam czytającego jakąś księgę urzędnika. Pytając się o drogę, zapytał z ciekawości: o czym, panie, czytasz? Urzędnik na to: to księga mądrości (Tao Te King). Garncarz zaśmiał się na to. Czemu się śmiejesz - pyta rozżłoszczony urzędnik. Czcigodny pan musi żartować. Ja nawet swojego czeladnika nie mogę nauczyć z książek lepienia garnków, jak więc mądrość zawrzeć można w książce? lepienie garnków czy jazda samochodem to wiedza o tym, jak wykonywać pewne zadania.

Meta-wiedza:

to wiedza o tym, co wiemy, o niedoskonałości naszej percepcji, o ograniczeniu naszej wiedzy na dany temat, ocenie jej wiarygodności.

Wierzenia:

prawdziwe czy fałszywe, nastawienia, wpływają na rozumowanie w oczywisty sposób. Mają one różne stopnie pewności.

Sprawy klasyfikacji wiedzy nie są proste i prowadzą do zagadnień psychologii. Pomińmy je na razie koncentrując się na używaniu wiedzy. W grę wchodzi tu 3 aspekty:

- (a) gromadzenie nowej wiedzy
- (b) wydobycie z bazy wiedzy faktów, związanych z danym problemem
- (c) rozumowanie przy użyciu tych faktów w poszukiwaniu rozwiązania.

Gromadzenie wiedzy nie jest tylko jej akumulacją, dodawaniem nowych faktów, a przynajmniej tym być nie powinno. Wymagany jest tu proces tworzenia związków z już istniejącą wiedzą. Konieczna jest więc klasyfikacja wiedzy (by wiedzieć „z czym to się je”), interakcja ze starą wiedzą (przy czym możliwe są interferencje z już zintegrowaną wiedzą). Wreszcie, musimy sobie jakoś radzić z formą prezentacji wiedzy naturalną dla ludzi. Niestety, jest to sprawa bardzo subiektywna (czy to numer 12-34-56? Nie, tu 123-456). Brak integracji wiedzy może dać nam w wyniku bazę danych a nie bazę wiedzy.

Wydobywanie wiedzy jest dla człowieka tak naturalne, że tego nie zauważamy. Myślimy poprzez skojarzenia, ale jak kojarzyć informacje w komputerze? Jeśli podejrzewamy, że skojarzenia między parą struktur danych będą przydatne, można je łączyć (linking), jeśli jakaś grupa jest ze sobą związana stosuje się ich grupowanie (lumping). Niestety, wyklucza to większość nieprzewidzianych skojarzeń.

Rozumowanie występuje wtedy, gdy system ma zrobić coś, co nie jest w jawny sposób zaprogramowane. W najprostszym przypadku możemy mieć sylogizmy, np.

- 1) wszyscy ludzie to ssaki;
- 2) Jacek jest człowiekiem.

System bez kłopotu odpowie na pytania: czy ludzie to ssaki itp. ale by odpowiedzieć na pytanie: czy Jacek jest ssakiem? musi przeprowadzić rozumowanie. Możemy sobie wyobrazić rozumowania różnego rodzaju:

Formalne rozumowanie występuje w logice matematycznej i opiera się na ustalonych regułach wnioskowania.

Proceduralne rozumowanie pomaga rozwiązywać problemy i odpowiadać na pytania przeprowadzając symulację danej sytuacji.

Rozumowanie przez analogię jest łatwe dla ludzi ale trudne dla programów. Np.: Jacek był kiedyś mały a niemowlęta ssą więc są ssakami.

Rozumowanie przez uogólnianie, bardzo częste w matematyce ale trudno jest je uchwycić, np. rozumowanie przez indukcję spotyka się w systemach AI bardzo rzadko a dla ludzi jest to jeno z głównych źródeł, czasami skłonni jesteśmy do zbyt szybkich uogólnień.

Meta-rozumowanie związane jest z wiedzą o tym, co wiemy, np.: czy ja to mogę skądś wiedzieć?

Wprowadzając nową wiedzę system AI powinien wiedzieć, w jaki sposób wiedza ta będzie używana, oraz czy da się ją efektywnie wykorzystać. Teoria reprezentacji wiedzy jest w powijakach, nie wiemy czemu jedne schematy działają lepiej niż inne. The proof is in the pudding, jak mawiają Anglicy.

Schematy reprezentacji wiedzy różnią się zakresem i szczegółowością wiedzy („ziarnistość”) reprezentowanej w danym schemacie. Zależy to niestety od kontekstu, np. reprezentacja logiczna będzie dobra w zagadnieniach matematycznych a zła w analizie obrazu. Wszystko co system wie, musi być uchwycone w pewnym formalizmie, ale pewne rzeczy są w nim prostsze niż inne. Kolejnym problemem jest wybór cech prymitywnych w sensie semantyki. Wybór ten nie jest jednoznaczny. Obiekt może mieć części lub też część może być częścią obiektu (np. skrzydła i ptaki). kolejną cechą schematów jest modularność i zrozumiałość. Pomimo powiązań wiedzy chcielibyśmy mieć na tyle zlokalizowane struktury, by usunięcie jednej nie wymagało restrukturyzacji całości. Ten sam problem występuje przy pisaniu wszystkich dużych programów: po pewnym czasie tu się dotknie a tam się wali. Może zmieni się to dzięki programowaniu obiektowemu.

Niestety, całkowita modularność nie wydaje się w ogóle możliwa - słowa zależą od kontekstu. Ogólnie istnieje odwrotna zależność pomiędzy modularnością a łatwością rozumowania: w systemach modularnych trudno jest rozumować ale łatwo dobrać się do faktów, w systemach proceduralnych rozumowanie może być łatwe ale trudno wydobyć fakty. Systemy mogą też w różny sposób traktować wiedzę: mogą ją zapisywać w sposób jawny, jako część struktury programu, a mogą ją mieć gdzieś zawartą implicite w związkach pomiędzy strukturami danych.