Sztuczna Inteligencja

Systemy ekspertowe - teoria

Włodzisław Duch
Katedra Informatyki Stosowanej, UMK
Google: Włodzisław Duch

Co było:

- NLP - czym się zajmuje
- Języki formalne i ich gramatyki
- Generacja tekstu
- Tłumaczenie maszynowe
- Przykłady programów
- Nagroda Loebnera

Co będzie

- ES – co to
- Etapy tworzenia
- Akwizycja wiedzy
- Architektury ES
- Języki programowania ES

System ekspertowy - definicja

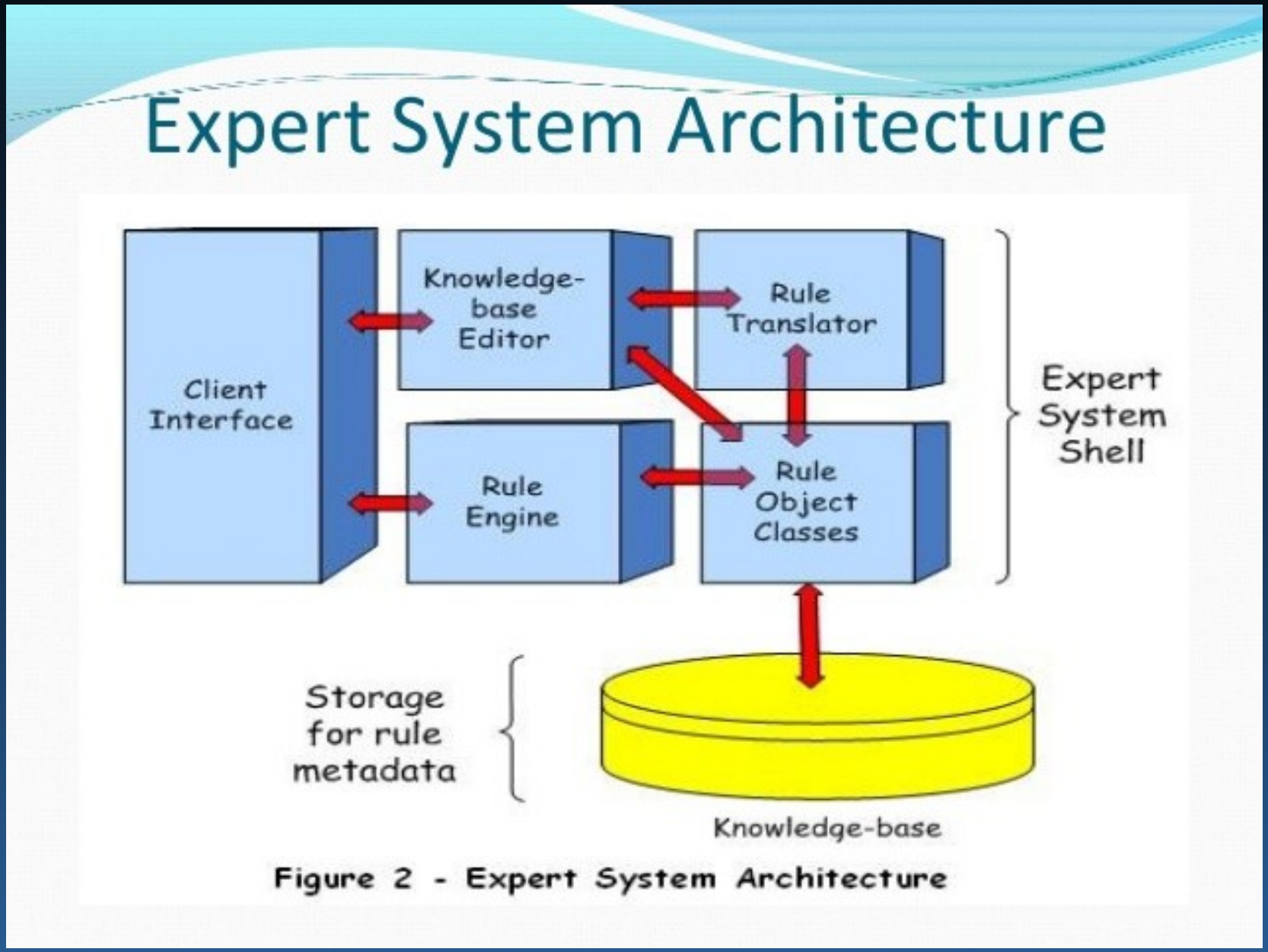
Tradycyjna sztuczna inteligencja to inżynieria wiedzy i systemy ekspertowe, lub systemy oparte na wiedzy (KBS). Wikibooks: Expert Systems
Uczenie maszynowe to głównie akwizycja wiedzy dla SE.

- System ekspertowy (doradczy, ekspercki):
  program komputerowy wykorzystujący wiedzę i procedury wnioskowania do rozwiązywania problemów, które są na tyle trudne, że wymagają znaczącej ekspertyzy specjalistów.
- Wiedza (niezbędna, by zapewnić odpowiedni poziom ekspertyzy), wraz z procedurami wnioskowania stanowi model ekspertyzy, posiadanej przez najlepszych specjalistów w danej dziedzinie.
- Zrobotyzowana Automatyzacja Procesów (RPA) często nie wymaga AI,
  ale można tu wyróżnić robotyzację kognitywną, np. UIPath.

Program = algorytm + struktury danych.

ES = interfejs użytkownika + wiedza + system wnioskujący.

System ekspertowy - schemat



System ekspertowy - intro

Wiedza systemu ekspertowego składa się z faktów i heurystyk.

Fakty: powszechnie akceptowane przez specjalistów.

Heurystyki: informacja subiektywna, która charakteryzuje proces oceny przez określonego specjalistę.

Mogą to być intuicyjne domysły, przypuszczenia, zdroworozsądkowe zasady postępowania.

Poziom ekspertyzy to funkcja rozmiaru i jakości bazy wiedzy danego systemu.

Po co SE?

Dlaczego buduje się SE?

1. Koszty: w dłuższym okresie czasu są znacznie tańsze, pomagają w rozwiązywaniu problemów wymagających najbardziej specjalistycznej (najdroższej) wiedzy.
2. Brak ekspertów w wielu dziedzinach.
3. ES pracują szybciej, nie męczą się, są bardziej niezawodne niż ludzie.
4. Konsekwentne, konsystentne, obiektywne, dokładne.
5. Zawsze do dyspozycji (nie strajkują!).
6. Analiza dużych ilości danych wymaga komputera.

SE: systemy oparte na wszystkich sposobach reprezentacji wiedzy, najczęściej w postaci reguł produkcji.

Etapy tworzenia SE

1. Analiza problemu - oceny, czy budowa SE dla danego problem ma sens, jakie są potencjalne korzyści.
2. Specyfikacja systemu - szczegółowe określenie funkcji i oczekiwań.
3. Akwizycja wiedzy - zgromadzenie, wydobycie z ekspertów i organizacji potrzebnej wiedzy.
4. Wybór metody reprezentacji wiedzy i narzędzi do budowy systemu.
5. Konstrukcja systemu - utworzenie bazy wiedzy, reguł wnioskowania, systemu wyjaśniającego rozumowanie i prowadzenie dialogu z użytkownikiem.
6. Weryfikacja i testowanie systemu.

Akwizycja wiedzy wymaga transferu ekspertyzy + reprezentacji wiedzy.

Postać wiedzy

- **Fakty** z danej dziedziny wiedzy, np:
  „W starych silnikach Diesla przy przegrzaniu dochodzi do gwałtownego podwyższenia obrotów na skutek chwilowego spalania oleju.”
- **Reguły** typu: „Przed zdjęciem obudowy wyciągnąć wtyczkę.”
- **Heurystyki**, czyli co by tu zrobić, np.:
  „Jak nie zaskakuje, a jest iskra, to warto sprawdzić przewód paliwa”.
- **Ogólne strategie** postępowania.
- **Teoria** danej dziedziny, np. teoria działania silników samochodowych.

Akwizycja wiedzy

- prowadzenie wywiadów z ekspertami
- analiza kwestionariuszy wypełnianych przez ekspertów
- analiza raportów pisanych przez ekspertów
- analiza komentarzy ekspertów wykonywanych w czasie pracy
- obserwacja ekspertów przy pracy
- introspekcja + opis działań
- szukanie w Internecie ...
- analiza dużej liczby przykładów ocenionych przez ekspertów za pomocą metod uczenia maszynowego
- upraszczanie wiedzy zawartej w dużych bazach danych przez poszukiwanie struktur za pomocą metod nienadzorowanego uczenia.

Rodzaje systemów ekspertowych

- Systemy **edukacyjne** typu CAI lub ICAI (Intelligent Computer Aided Instruction), a więc inteligentne wspomaganie nauczania, systemy algebry symbolicznej.
- Systemy **interpretujące,** wspomagające analizę i interpretację informacji, wydobywanie informacji z baz danych, interpretujące dane geologiczne.
- Systemy **planistyczne** wspomagające strategiczne działanie i planowanie zadań, np. planowanie syntezy związków chemicznych czy budowy systemów komputerowych.
- Systemy **prognostyczne** wspomagające wyciąganie wniosków i przewidywanie tendencji.
- Systemy **kontrolne** pozwalające na sterowanie skomplikowanymi systemami, takimi jak automatyczne zakłady produkcyjne itp.

Rodzaje cd.

- Systemy **diagnostyczne** to jedno z najbardziej popularnych zastosowań SE, w zagadnieniach technicznych, medycynie, analizie chemicznej i wielu innych problemach.
- Systemy **testujące** pomagają przy znajdywaniu problemów i mogą być częścią systemów kontrolnych lub systemów diagnostycznych.
- Systemy **naprawcze** nie tylko prowadzą testy ale i planują działania korekcyjne. Można do nich zaliczyć również niektóre systemy medyczne, zalecające leczenie.
- Systemy **projektujące** wspomagają prace projektowe, takie jak projektowanie układów elektronicznych, CAD czy CAM.

10 kategorii ES

Klasyfikacja Hayes-Roth, Waterman, Lenat (1983)

1. Interpretacja: sensory => fakty
2. Predykcja: konsekwencje obserwacji
3. Diagnoza: przewidywanie i przyczyny problemów
4. Projektowanie: konstrukcje z ograniczeniami
5. Planowanie: sekwencje działań
6. Monitorowanie: porównywanie obserwacji, alarmy
7. Debugowanie: poprawki w złożonych systemach
8. Naprawa: plany naprawcze i ich monitoring
9. Nauczanie: ocena postępów i planowanie materiału
10. Sterowanie: interpretacja i poprawa zachowania systemu

5 typów ES

Z punktu widzenia konstrukcji wyróżnia się też:

1) Systemy regułowe, oparte na regułach produkcji i logice klasycznej.
2) Systemy oparte na ramach i logice klasycznej.
3) Systemy wykorzystujące reguły i logikę rozmytą.
4) Systemy neuronowe, wnioskujące bezpośrednio z danych.
5) Systemy neuro-rozmyte, odkrywające cechy rozmyte i wnioskujące na ich podstawie.

Konstrukcja systemów eksperckich

ES ma odpowiadać na pytania na poziomie eksperta.
W wielu zastosowaniach próbuje się oddzielić bazy wiedzy od samych mechanizmów wnioskowania, czyli unikać reprezentacji proceduralnych.

Reguły produkcji:

<obiekt,atrybut,wartość>, np. <samochód,kolor,czerwony>

Stosowane są też ramy, sieci semantyczne, sieci Bayesowskie, reprezentacje bezpośrednie i proceduralne;

rzadziej reprezentacje logiczne.

Alternatywy dla ES prowadzących wnioskowanie:
modele statystyczne, symulacje procesów w różnych warunkach, np. rozchodzenia się epidemii.

Rodzaje rozumowania

DSS (Decision Support Systems), Inteligentne DSS?

Dialog z użytkownikiem + wyjaśnienia sposobów wnioskowania.

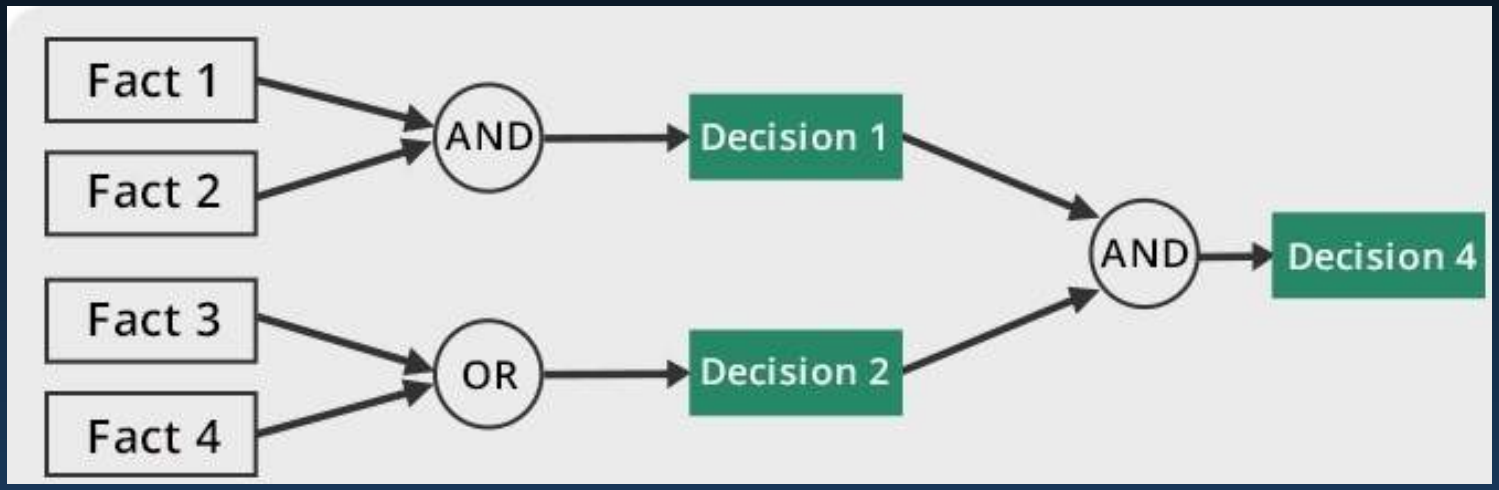
ES nie zawsze system rozumuje w sposób podobny do człowieka – ma inne ograniczenia „sprzętowe”, ale powinno wyjaśniać decyzje – *explainable* AI stało się jednym z wymogów wraz z akwizycją wiedzy z dużych danych.

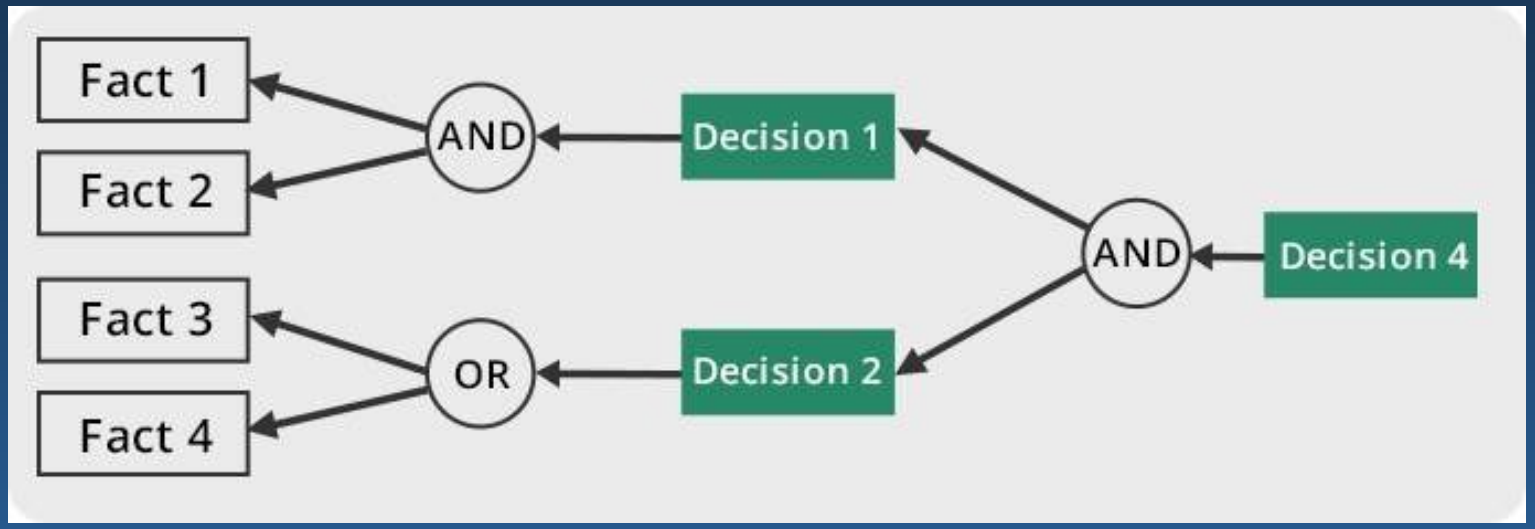
Jak wyjaśniać?

- Rozumowanie **retrospektywne** (które reguły i dlaczego).
- Rozumowanie **hipotetyczne** (co by było gdyby ...).
- Rozumowanie „**alternatywne**” (counterfactual reasoning): alternatywne możliwości: gdyby było P byłoby inaczej, a tak jest S.
  Jeśli P powoduje S, to gdyby nie było P nie byłoby S.
  Szukamy więc czemu nie ma P.
  Stanford Encyclopedia, Conterfactual Theories of Causation.

Rodzaje wnioskowania

Wnioskowanie **w przód** (forward chain) i **wstecz** (backward chain)





Rozstrzyganie konfliktów

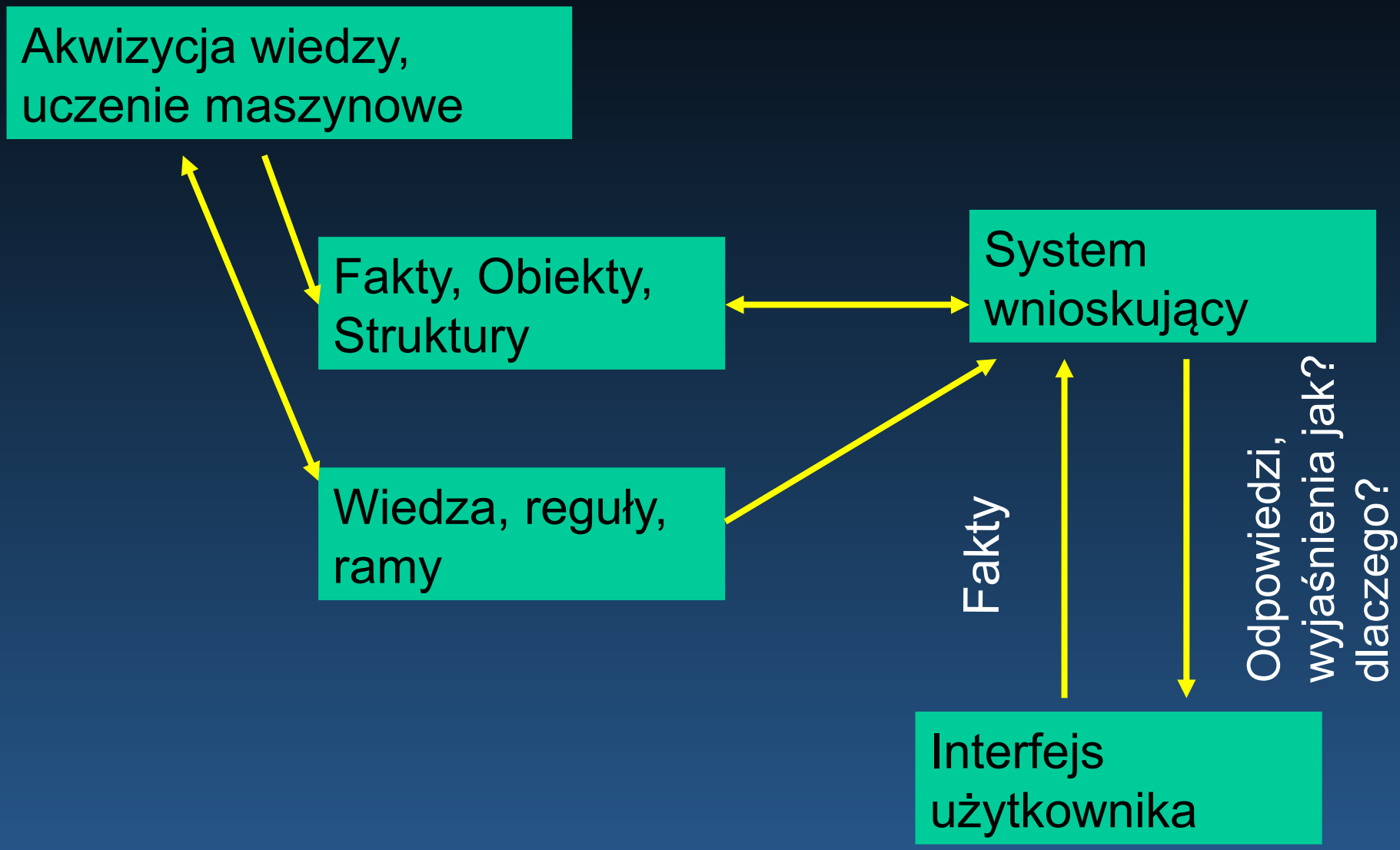
Jeśli kilka reguł daje się zastosować do tej samej sytuacji:

- użyj reguły o najwyższym priorytecie
- użyj reguły która ma najwięcej szczegółowych warunków
- użyj ostatnio wykorzystywaną regułę
- użyj regułę, która została dodana najpóźniej
- użyj regułę zawierającą zmienne, które były ostatnio używane.

Jeśli mamy wagi przesłanek (stopień prawdziwości) to

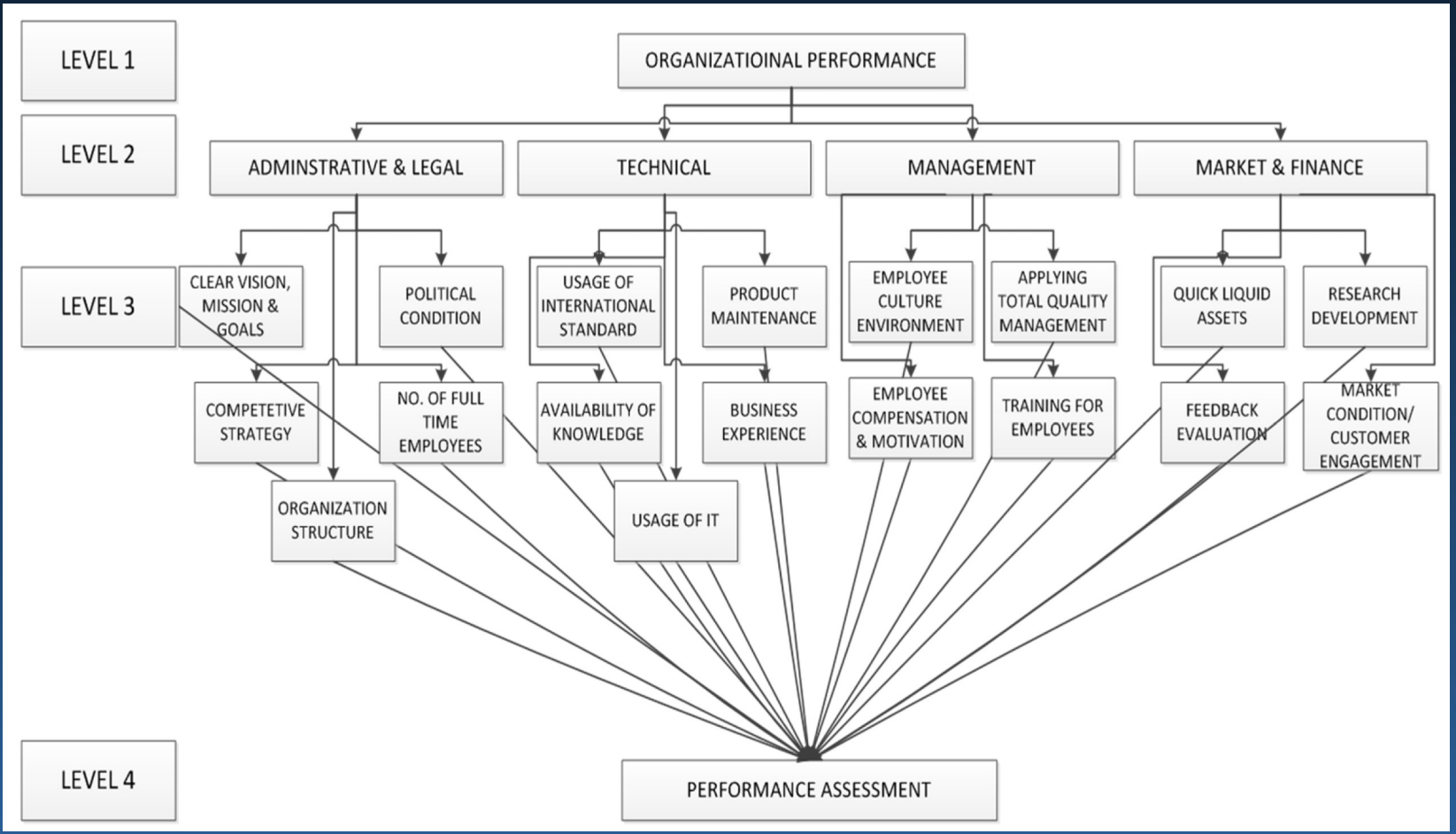
- użyj regułę do której konkluzji masz największe zaufanie.

Ogólna konstrukcja ES



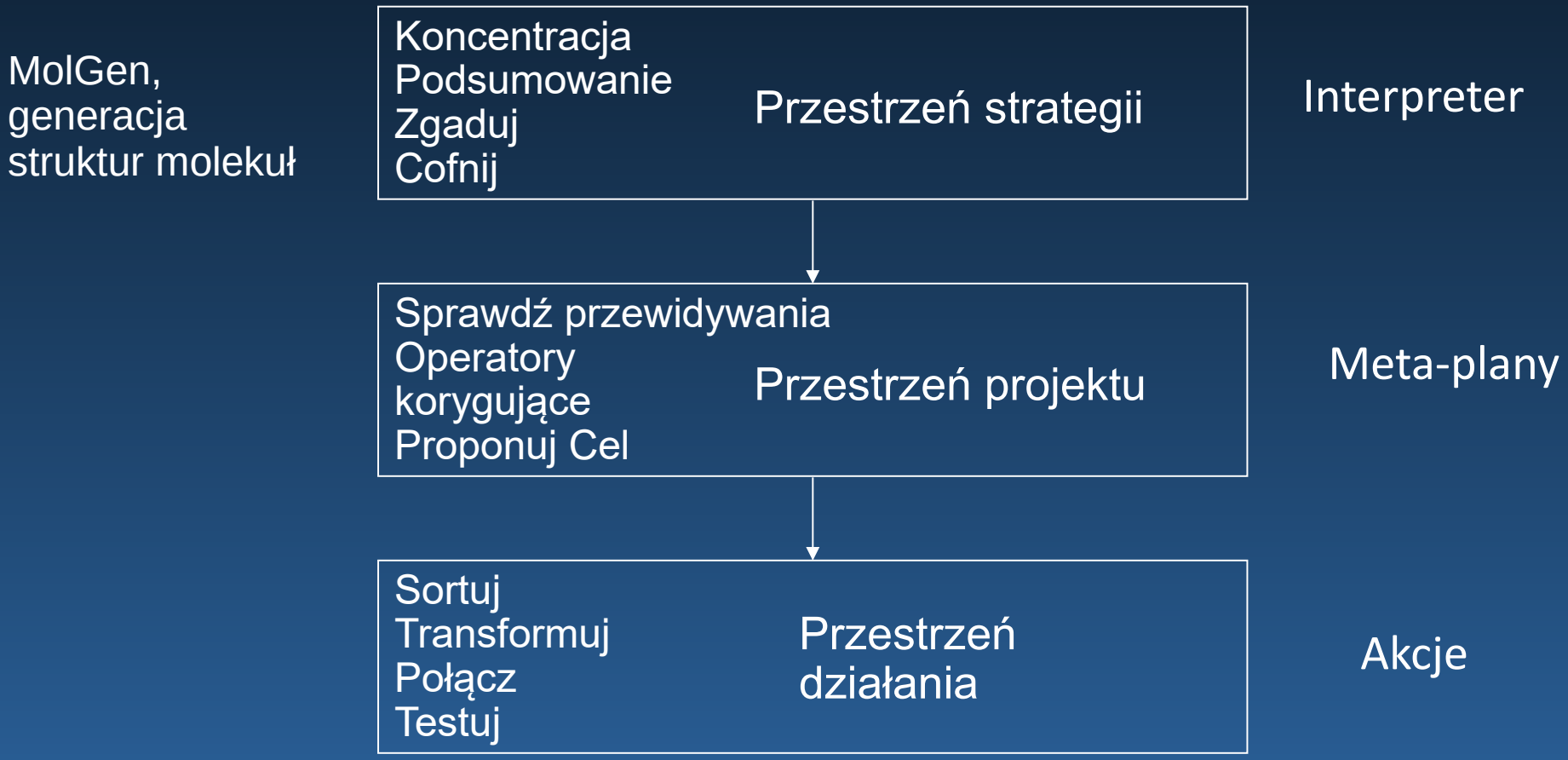
Architektura hierarchiczna

- **Architektura hierarchiczna**:
  poziom faktów z danej dziedziny + przynajmniej jeden meta-poziom, wiedza strategiczna, fakty dotyczące reguł niższego poziomu.
  Systemy hierarchiczne działające w szerszych domenach wiedzy tworzą drzewa taksonomiczne usiłując podzielić całą wiedzę na rozłączne specjalistyczne dziedziny.



Architektura wielowarstwowa

- **Architektura wielowarstwowa**: kilka warstw, leżące wyżej kontrolują działanie na niższym poziomie; meta-wiedza i kryteria strategicznego planowania i działania.
- Czasami przestrzeń zamiast warstwy, np. przestrzeń działania (konkretne akcje), przestrzeń planowania (określanie celów bieżących), przestrzeń strategii (koncentracja uwagi na jakimś obszarze, cofanie działań).



Architektura tablicowa

- Architektura tablicowa (blackboard): łączenie wiedzy z kilku źródeł w „pamięci roboczej”, z której korzystają moduły wnioskujące.
  Jedna lub kilka tablic, informacje mają hierarchiczną strukturę o wzrastającym stopniu szczegółowości.

Zastosowana po raz pierwszy w systemie HEARSAY, jednym z pierwszych działających systemów do rozpoznawania mowy.
Popularna w systemach inspirowanych biologicznie, jako model pamięci roboczej (Global Workspace Theory).

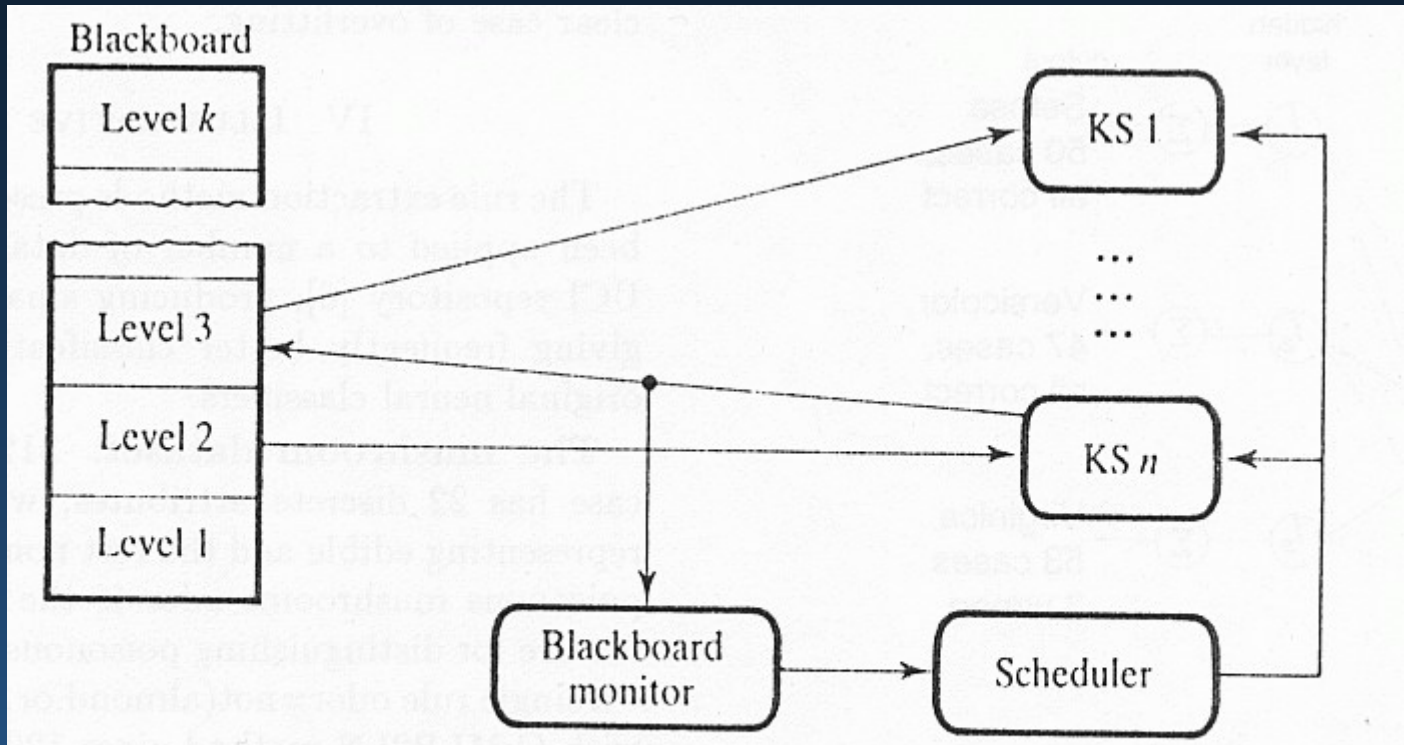
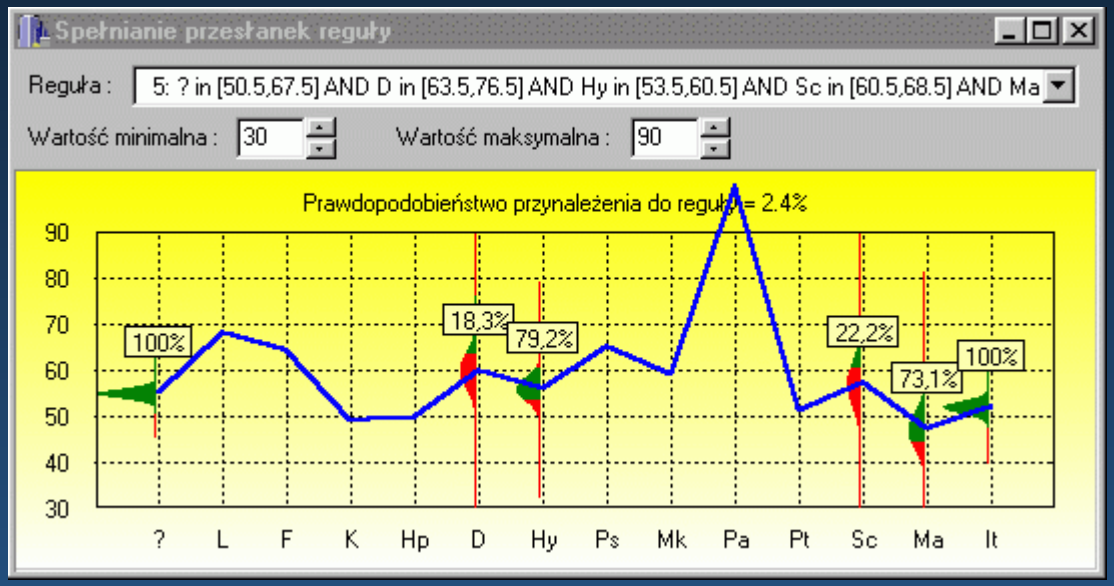


Figure 22.1 Simplified schematic of the HEARSAY-II architecture.

Architektury hybrydowe

- Architektura hybrydowa: regułowo-koneksjonistyczna
  Umożliwia automatyczne tworzenie skojarzeń.
  Systemy koneksjonistyczne mogą służyć odkrywaniu wiedzy na podstawie analizy danych; wiedza dodawana jest do systemu.
- Przykład: MMPI-IDSS, nasz system analizy psychometrycznej.

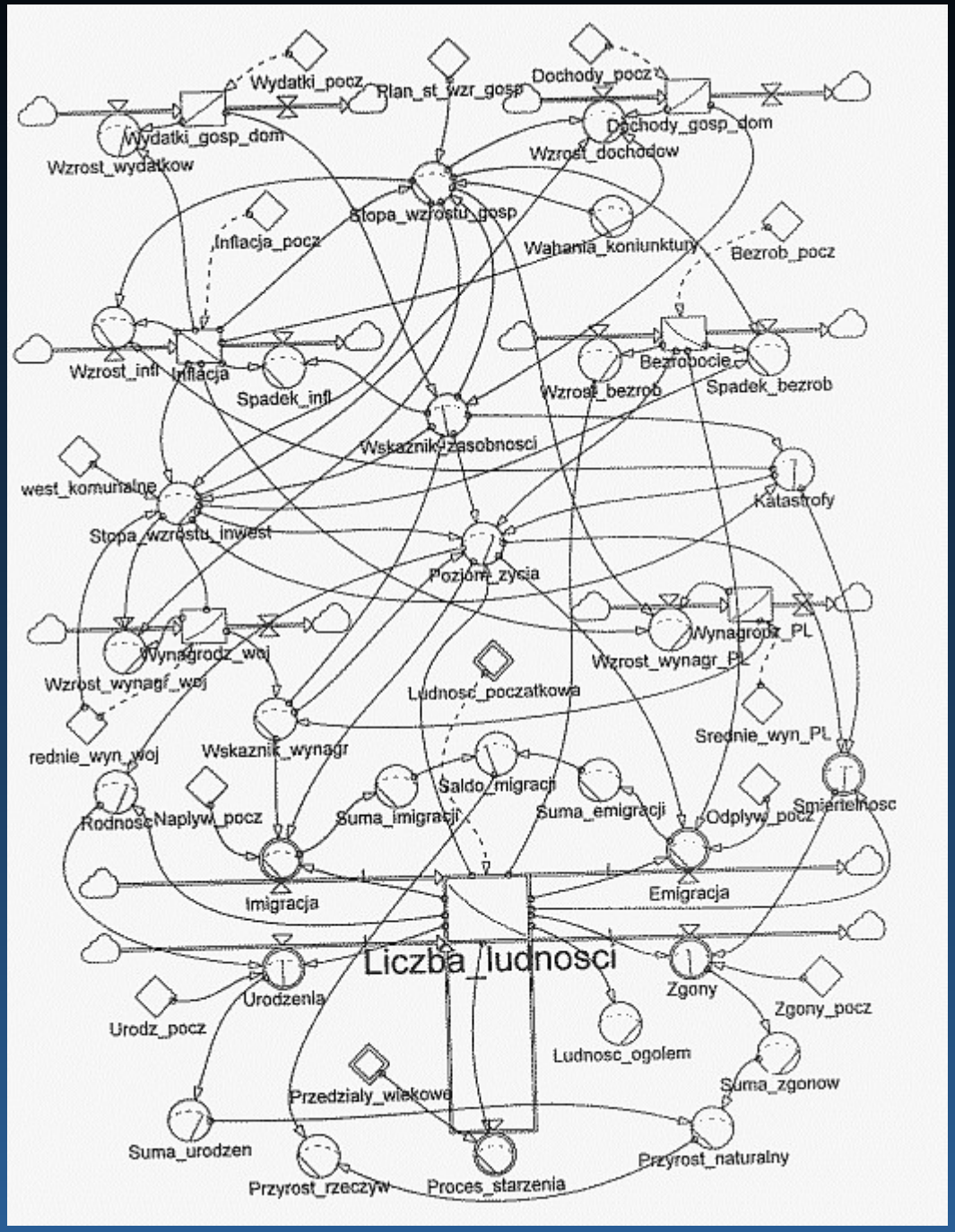


Architektury symulacyjne

Architektura symulacyjna: modele numeryczne danej dziedziny, zależności funkcyjne nie dające się uchwycić w postaci reguł. System do symulacji ustala funkcje wpływu poszczególnych węzłów na inne i rozwiązuje równania różniczkowe pozwalające na śledzenie zmian.

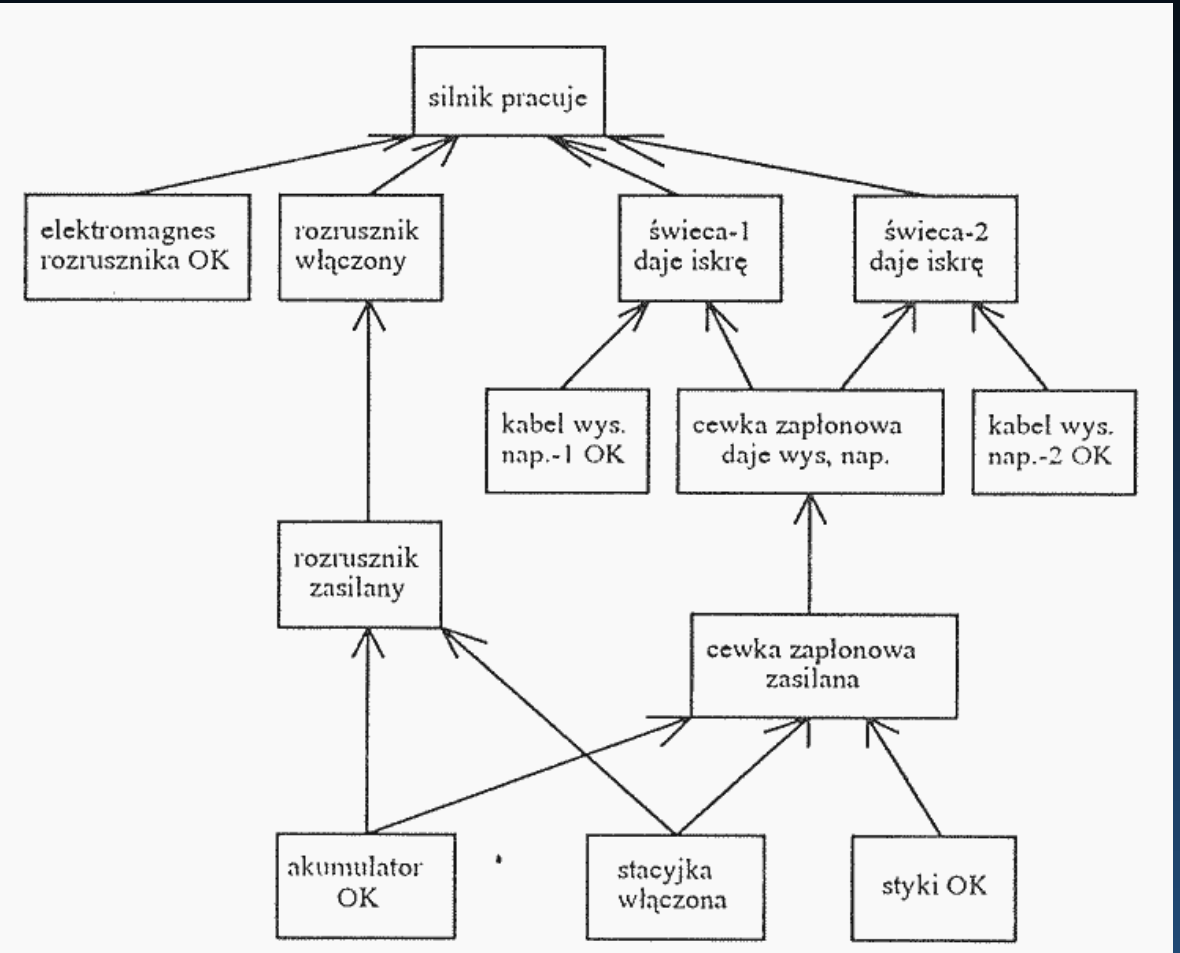
Współczyniki wpływu i funkcje trzba tak dopasować by z danych z przeszłości odtworzyć obecnie obserwowane.

Przykład: prognozowanie liczby ludności.



Architektury powiązań przyczynowych

Systemy tworzące sieci przyczynowych powiązań (causal networks) lub probabilistyczne sieci Bayesowskie.

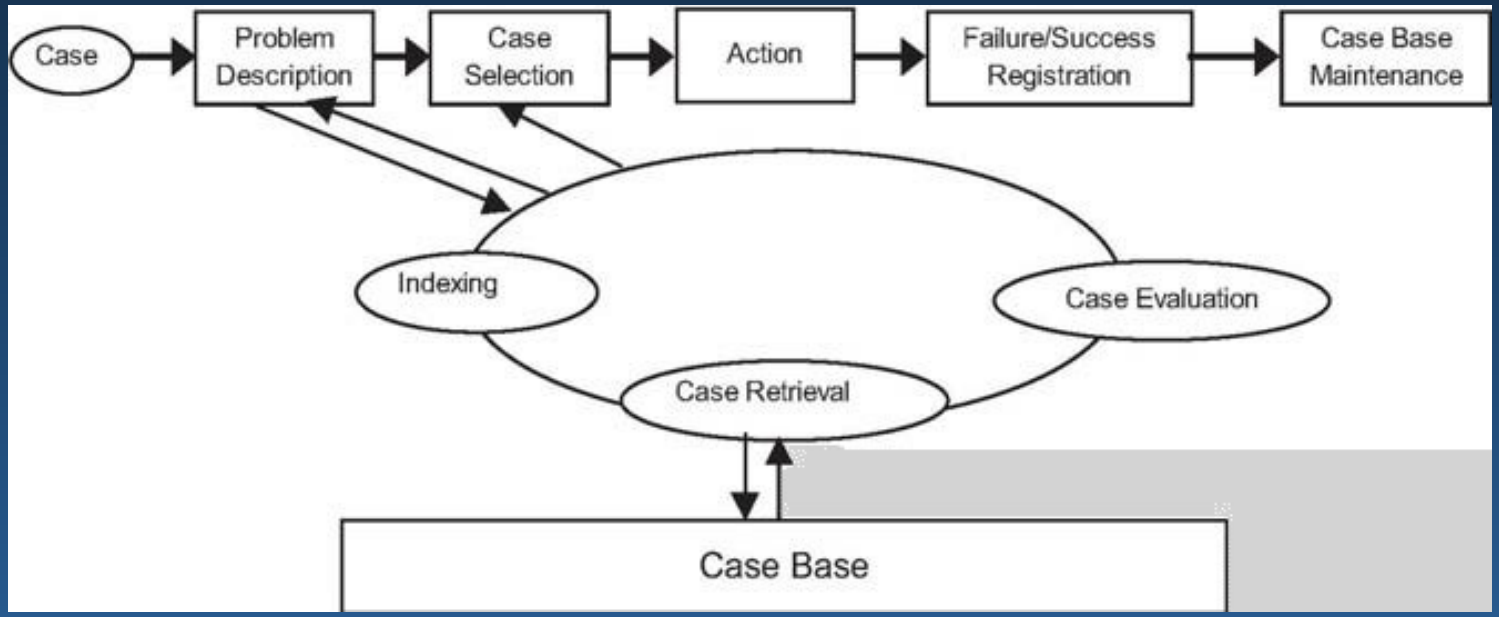


Rys. 1. Sieć przyczynowa.

Architektury wykorzystujące analogie

- Architektura **korzystająca z analogii** (case-based reasoning).
  W wielu dziedzinach (prawo, medycyna) podstawą rozumowania są analogie, precedensy, do których można się odwołać.
- Korzystają z bazy danych opisujących znane przypadki, ocen podobieństwa, reguł szukania i używania analogii.

Zawierają opis klas problemów, jakie potrafią rozwiązać, wraz ze schematami rozwiązań i sposobami określania podobieństwa do znanych przypadków z danej klasy.



Konstrukcja ES

Systemy klasyfikujące: wybór rozwiązania z ustalonej grupy.

Systemy konstruujące: składanie rozwiązania z elementów.

Problem: brak wiarygodnej wiedzy, różne rodzaje niepewności.

Rozwiązanie: prawdopodobieństwa warunkowe,
współczynniki ufności lub pewności (confidence factors),
teoria wiarygodności, teoria zbiorów rozmytych.

Metodologia konstrukcji dużych systemów podobna jest do narzędzi CASE (Computer Aided Software Engineering), np:

KADS, Common KADS, Pragmatic KADS.
Książka o Common KADS.

Przykład diagramów UML (Universal Modelling Language) stosowanych w KADS.

Konstrukcja ES

Systemy klasyfikujące: wybór rozwiązania z ustalonej grupy.

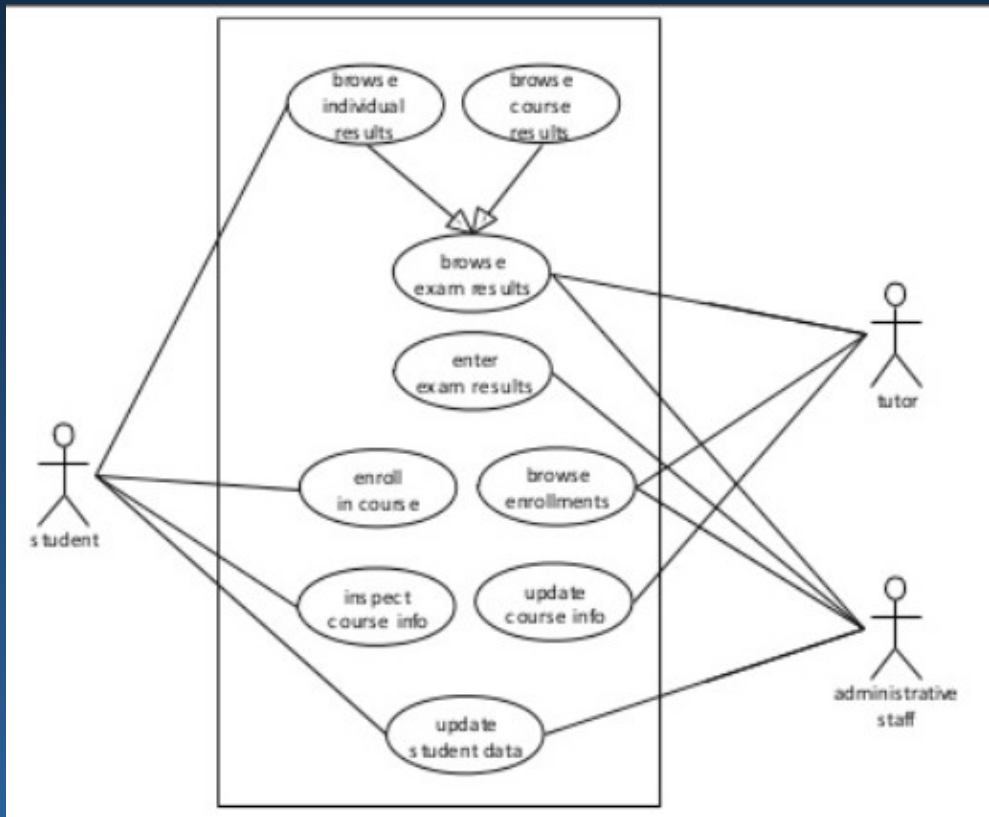
Systemy konstruujące: składanie rozwiązania z elementów.

Problem: brak wiarygodnej wiedzy, różne rodzaje niepewności.

Rozwiązanie: prawdopodobieństwa warunkowe,
współczynniki ufności lub pewności (confidence factors),
teoria wiarygodności, teoria zbiorów rozmytych.

Metodologia konstrukcji dużych systemów
podobna jest do narzędzi
CASE (Computer Aided Software Engineering).

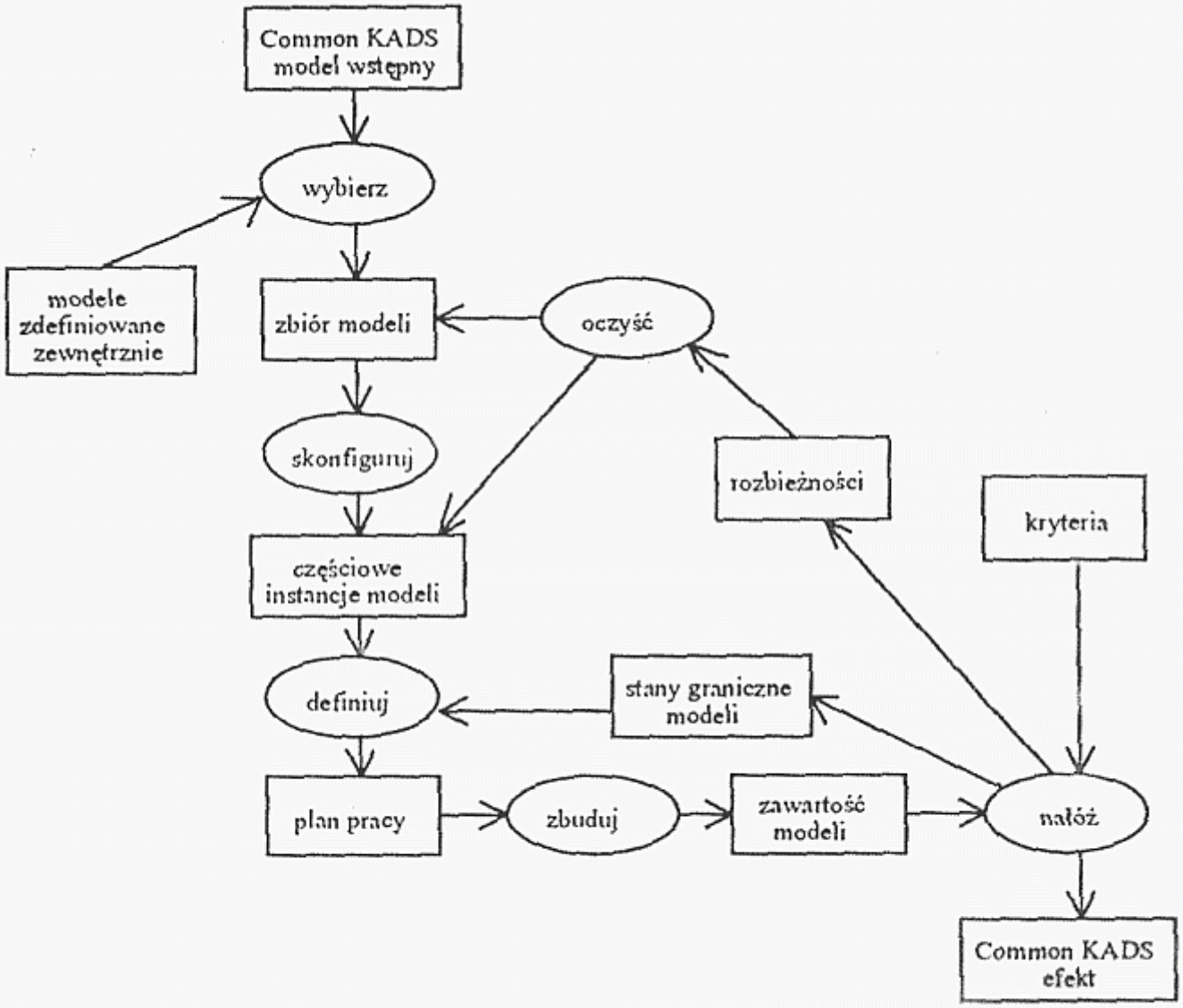
Przykład diagramów UML
(Universal Modelling Language)
stosowanych do konstrukcji dużych
systemów ekspertowych.



Common KADS

Diagram ilustrujący schemat budowy modelu w narzędziach Common KADS.

Książka o Common KADS.

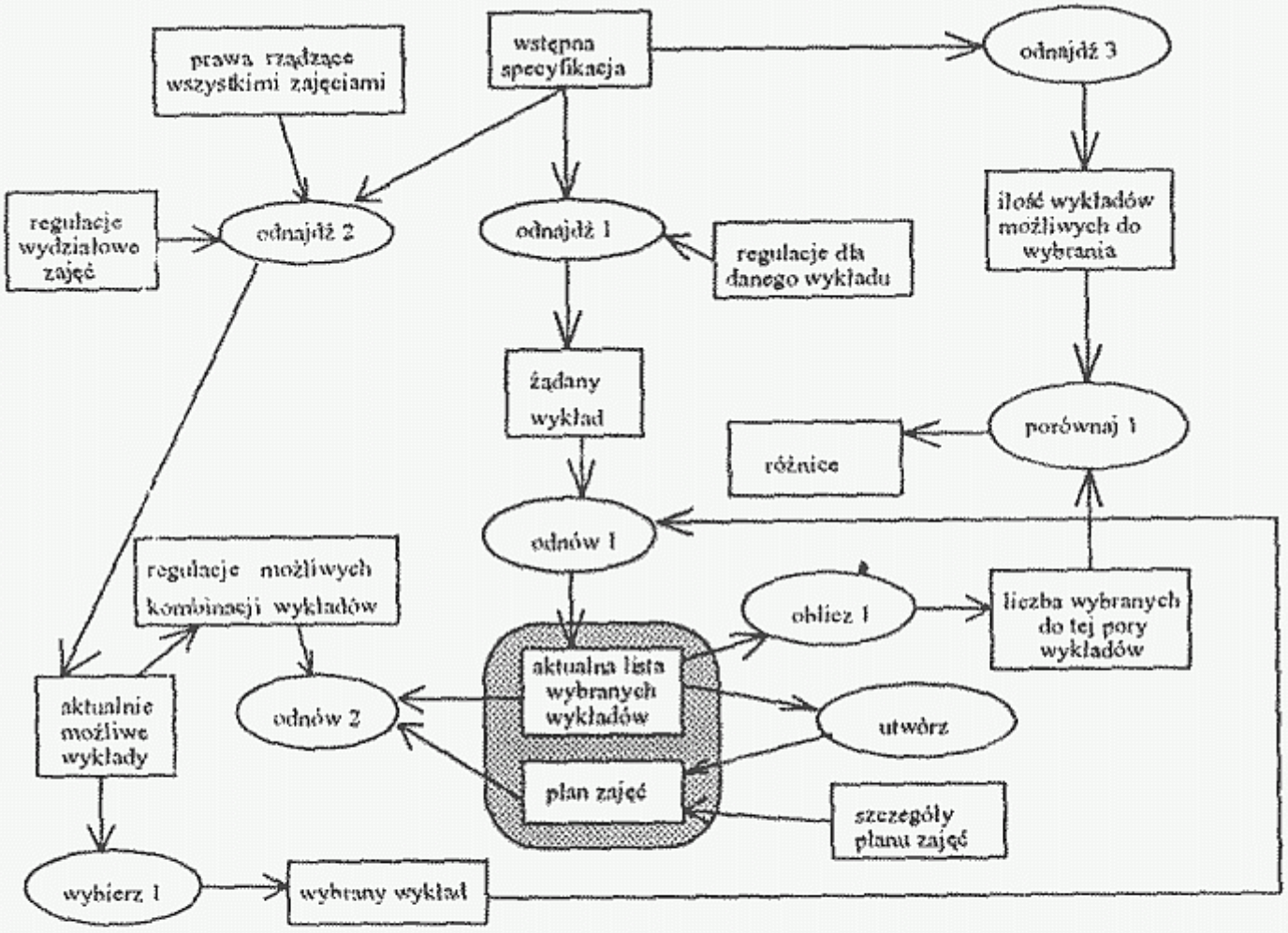


Rys. 3. Common KADS - budowa modelu.

Pragmatic KADS

Diagram ilustrujący schemat budowy modelu w narzędziach Pragmatic KADS, uzywanych zwykle do mniejszych projektów.

Inne narzedzia do tworzenia systemów ES.



Rys. 5. Diagram dla problemu wyboru zajęć stworzony przez Pragmatic KADS.

Języki programowania do tworzenia ES

LISP (List PROcessing, przetwarzanie list), 1958, J. McCarthy

Common Lisp 1984 rok, wiele dialektów, np. Scheme

CLOS (Common Lisp Object System)

Pakiety graficzne (np. AUTOCAD), interfejsy użytkownika

Specjalne komputery dla Lispu: stacje SYMBOLICS i inne, wyszły z użycia po 2015 r.

Język funkcyjny: listy i funkcje

(minimalnie 7 funkcji pozwala zrealizować model maszyny Turinga)

FACTORIAL(N):

```
(COND ( ( EQUAL N 1)  1 )
(TRUE (TIMES N (FACTORIAL (DIFFERENCE N 1))) ))
```

Języki ES cd

Prolog (Programming in Logic), Marsylia i Edynburg.

Realizacja rachunku predykatów pierwszego rzędu, do prototypów, Prolog w projekcie V generacji; raczej mniejsze systemy lub prototypy.

Inne: POP-2 do POP-5, FUZZY

Expert System Shells (ESS):

EMYCIN, KAS (Knowledge Aquisition System), OPS5,

KEE, Knowledge Engineering Environment, KES

ESS: czas opracowania systemu 10-20 razy krótszy

Ostatnio również języki zorientowane obiektowo: C++, Smalltalk, Dylan.

CLIPS

C- Language Integrated Production System, CLIPS, najczęściej używany do budowy SE.

Projekt NASA, połowa lat 1980, oparty na regułach produkcji.

Wykorzystuje proceduralną reprezentację wiedzy.

- Ma bazę faktów i bazę reguł.

  * (defrule Zadania
  „Do zrobienia w niedzielę”
  (salience 10)
  (dzisiaj is Sobota) (pogoda is ładna) =>
  (assert(jedź-do lasu)) (assert(zrób zakupy))

- Zmienne: ?dzień, ?zakupy
- Działa w cyklu: rozpoznaj warunki, działaj.

90% czasu zajmuje rozpoznawanie warunków i dopasowanie reguł do zaistniałej sytuacji.

JESS, rozszerzenie CLIPS w Java, to Expert System Shell.

CLIPS cd

- **Polecenia**:
  (run), (refresh), (watch rules), (agenda), (list-defrules) ...
- **Wzorce**: pozwalają zdefiniować rekordy
  *(deftemplate student „informacja o studencie”
  (slot nazwisko (type STRING))
  (slot miasto (type NUMBER) (default Torun)) ...
- **Funkcje** - notacja z Lispu:
  * (deffunction przeciwp(?a ?b)
  (sqrt(+ (* ?a ?a) (* ?b ?b))))
- COOL – CLIPS Object Oriented Language, czyli obiektowo zorientowana wersja CLIPS
- **Rozumowanie**: w przód z rozstrzyganiem konfliktów, rozumowanie zorientowane na cel, wykorzystujące „podpowiedzi” (task tokens), możliwe definiowanie wielu kontekstów, wspomaganie rozrastania się systemu.

Niepewność wiedzy: przyczyny

Przyczyny niepewności wiedzy:

- Niewiarygodne źródła informacji.
- Zbyt wiele informacji nie mającej znaczenia.
- Brak precyzji w obserwacjach i opisie.
- Błędy aparatury.
- Brak zrozumienia sytuacji.
- Sprzeczne informacje.
- Nieznane czynniki wpływające na sytuację.
- Zmiana sytuacji w czasie, starzenie się wiedzy.
- Wysokie koszty pozyskiwania nowych informacji.

Niepewność w ES

Logika rozmyta, teoria prawdopodobieństwa i inne sposoby.

Najprostsze: czynniki „zaufania”, CF (confidence factor)

CF: {Stwierdzenia X} $\in [-1,+1]$

- CF = +1 na pewno prawdziwe
- CF = -1 na pewno fałszywe
- CF = 0 nic nie wiadomo.

CF(wyniku akcji) = CF(warunków) x CF(reguł)

Jest to aproksymacja wnioskowania probabilistycznego.
W praktyce stosuje się heurystyczne formuły do obliczania CF.

Logika Rozmyta i teoria Dampstera-Shafera są również często stosowane.

Zalety i wady ES

- Przydatne do rozwiązywania złożonych problemów, w dziedzinach, w których zgromadzono wiedzę empiryczną
- Potrafią odpowiadać na pytania prezentując swoje konkluzje w intuicyjne zrozumiały sposób, nie potrzeba programistów by zrozumieć ich działanie, to „wyjaśnialne AI”.
- Zwykle oparte są na jednolitym sposobie reprezentacji wiedzy, np. regułach/ramach, dzięki czemu łatwo jest modyfikować wiedzę.

Wady:

- Trudno przewidzieć skutki dodania nowej wiedzy, rozumowanie zdroworozsądkowe jest trudne, wymaga obszernej wiedzy i wyobraźni - brak możliwości analizy obrazów i sygnałów.
- Trudno jest pozyskiwać wiedzę – ale uczenie maszynowe i odkrywanie wiedzy powoli staje się częścią ES.
- Uwzględnianie niepewności jest rzadko spotykane w klasycznych systemach ES.