

Politechnika Poznańska
Instytut Informatyki

Adam Meissner

Adam.Meissner@put.poznan.pl

<http://www.man.poznan.pl/~ameis>

SZTUCZNA INTELIGENCJA

Architektury systemów eksperckich, systemy regułowe

Literatura

- [1] Cichosz P., *Systemy uczące się*, WNT, Warszawa, 2000.
- [2] Flasiński M., *Wstęp do sztucznej inteligencji*, PWN, 2011.
- [3] Giarratano J.C., Riley G.D., *Expert Systems: Principles and Programming, Fourth Edition*, Course Technology, 2004.
- [4] Hopgood A.A., *Intelligent Systems for Engineers and Scientists, Third Edition*, CRC Press, 2011.
- [5] Lebowitz J., *The Handbook of Applied Expert Systems*, CRC Press, 1998.
- [6] Puppe F., *Systematic Introduction to Expert Systems*, Springer-Verlag, 1993.
- [7] Russell S.J., Norvig P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey, 2009.

Plan wykładu

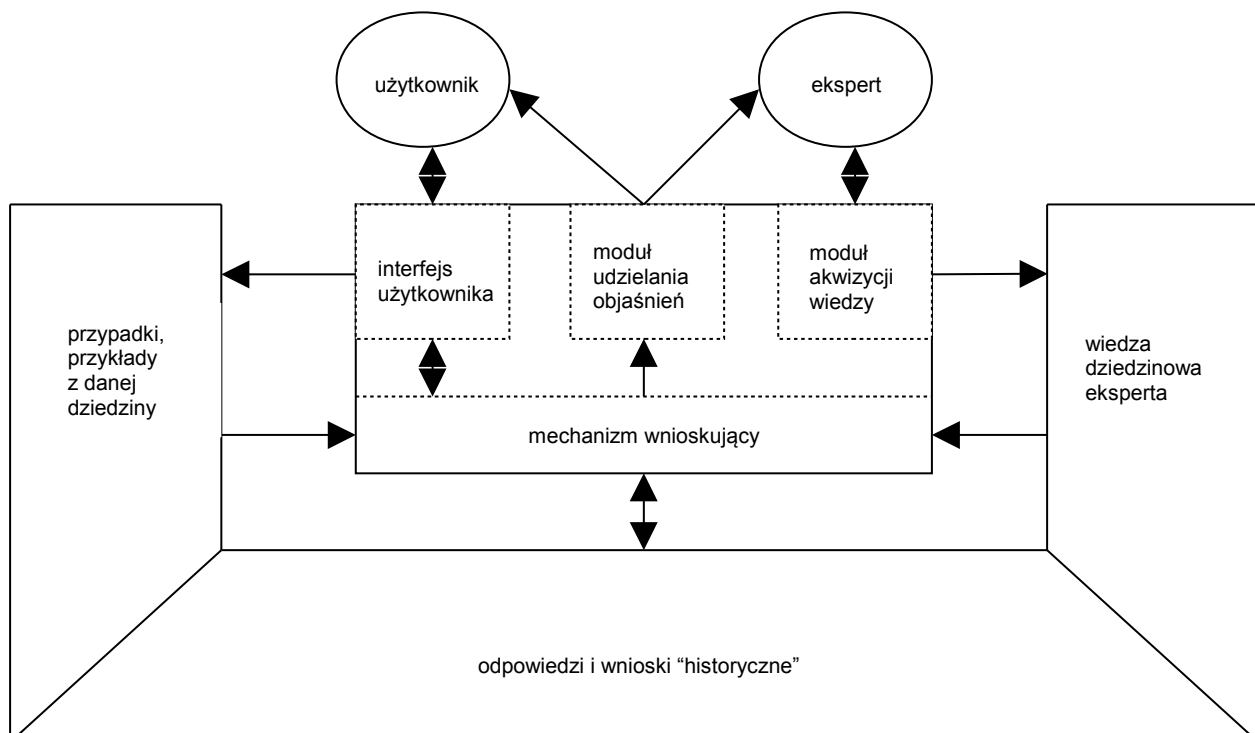
1. Wprowadzenie.
2. Metody pozyskiwania (akwizycji) wiedzy od ekspertów.
3. Metody reprezentowania i przetwarzania wiedzy.
4. Zagadnienia komunikacji z użytkownikiem i udzielania objaśnień.
5. Konstruowanie systemów eksperckich.
6. Podsumowanie.

Wprowadzenie (1)

Znaczenie terminu „system ekspercki”

„System ekspercki to program komputerowy wyposażony w wiedzę i umiejętności wnioskowania właściwe dla specjalistów z pewnej dziedziny” [6]

Typowa architektura systemu eksperckiego (wg. [6])



Wprowadzenie (2)

Rys historyczny (lata ok. 1950 – 1975)

- rozwój teorii gier (np. algorytm cięć alfa-beta), badania nad przeszukiwaniem heurystycznym (np. algorytm A* jako uogólnienie algorytm Dijkstry)
- skonstruowanie pierwszych systemów rozwiązujących proste problemy, np. GPS (Newell A, Simon H.) oraz SHRDLU (Winograd T.), badania nad automatycznym wnioskowaniem – zdefiniowanie reguły rezolucji (Robinson J.R.)
- ukształtowanie się koncepcji systemu eksperckiego, powstanie pierwszych systemów eksperckich, takich jak:
 - MYCIN (Feigenbaum E., 1976) – diagnozowanie bakteryjnych zakażeń krwi, regułowa reprezentacja wiedzy, wnioskowanie na podstawie współczynników pewności
 - CADUCEUS (Uniw. Pittsburgh, 1974 - ok. 1985) – diagnozowanie chorób wewnętrznych, zaimplementowany w języku INTERLISP, rozpoznaje ok. 1000 tj. 75% wszystkich znanych jednostek chorobowych, problem współwystępowania chorób
 - PUFF (1979) – system diagnozowania chorób płuc, zaimplementowany z wykorzystaniem powłoki EMYCIN, duża trafność postawionych diagnoz (ok. 85%)

Wprowadzenie (3)

Rys historyczny (cd.)

- MACSYMA – (MIT, początek lat 70-tych) – wykonywanie obliczeń symbolicznych, zaimplementowany w języku LISP, popularny wśród matematyków i inżynierów
- DENDRAL (Feigenbaum E., Buchanan B., Lederberg J.) – określanie struktury związku chemicznego na podstawie analizy spektralnej, zaimplementowany w INTERLISPie, generowanie i testowanie hipotez
- PROSPECTOR (SRI Int., przełom lat 70/80) – interpretowanie map geologicznych, odkrycie bogatych złóż rudy molibdenu w stanie Washington (wartych ok. 1 mld USD)
- HERSAY I i II (Uniw. Carnegie-Mellon) – rozpoznawanie mowy, pionierskie rozwiązania z zakresu architektur tablicowych.

Wprowadzenie (4)

Kryteria klasyfikowania systemów eksperckich

- dziedzina zastosowań: np. medycyna, inżynieria, matematyka, chemia, fizyka, geologia, meteorologia, rolnictwo, prawo, zarządzanie, doradztwo finansowe, wojskowość, transport, kosmonautyka, sterowanie produkcją
- przeznaczenie – systemy: kontrolne, diagnostyczne, testujące (i naprawcze), projektujące, edukacyjne, interpretujące, planistyczne, prognostyczne.

Specyfika wiedzy w systemie eksperckim

- niepewność
- niepełność
- zmienność w czasie
- potrzeba uzupełniania wiedzą zdroworozsądkową

Metody reprezentowania i przetwarzania wiedzy

- rachunki formalne (np. rachunek klauzul Horna, logiki deskrypcyjne, logiki nieklasyczne – np. modalne, temporalne)
- reguły IF-THEN, wnioskowanie progresywne (ang. *forward chaining*), wnioskowanie regresywne (ang. *backward chaining*)
- ramy i obiekty

Pozyskiwanie wiedzy od ekspertów (1)

- metody pozyskiwania (akwizycji) wiedzy
 - rola inżyniera wiedzy
 - metody „aktywne” – przeprowadzanie rozmów z ekspertem ewentualnie wspomagane użyciem ankiet, kwestionariuszy, diagramów, itp.
 - metody „pasywne” – obserwowanie pracy eksperta i wyników tej pracy, np. analiza raportów sporządzanych przez eksperta
 - porównywanie działań różnych ekspertów
- automatyzacja pozyskiwania wiedzy
 - niektóre elementy procesu pozyskiwania wiedzy, mające charakter dobrze zdefiniowanych procedur, mogą być realizowane przez środowiska narzędziowe
 - środowiska do „pasywnej” akwizycji wiedzy wykorzystują np. techniki analizy NLP, algorytmy uczenia maszynowego i inne
 - korzyści z automatyzacji pozyskiwania wiedzy polegają m.in. na zwiększeniu efektywności tego procesu (przez „dyscyplinowanie” eksperta) oraz na unikaniu przekłamań wynikających z udziału inżyniera wiedzy
 - wiedza pozyskana w sposób automatyczny może jednak pomijać specyficzne elementy rozpatrywanej rzeczywistości, które mogłyby być uwzględnione podczas bezpośredniej współpracy eksperta z inżynierem wiedzy

Pozyskiwanie wiedzy od ekspertów (2)

Drzewa decyzyjne

- **założenia:** dany jest zbiór D decyzji eksperta, klasyfikujących podany zbiór przykładów S na podstawie wartości wyróżnionych cech
- **zadanie:** skonstruować drzewo, którego wierzchołki wiszące reprezentują elementy zbioru D , a wierzchołki wewnętrzne odpowiadają cechom; każda krawędź wychodząca z wierzchołka wewnętrznego reprezentuje jedną z możliwych wartości cechy przypisanej temu wierzchołkowi
- ww. zadanie jest przykładem problemu **indukowania pojęć**, który wchodzi w zakres **uczenia maszynowego** [1]
- do najpopularniejszych metod konstruowania drzew decyzyjnych należy algorytm **ID3** Rossa Quinlanna (1979) oraz jego warianty i udoskonalenia (**C4.5**, 1993).

Algorytm ID3

Dane: zbiór przykładów S .

Wynik: korzeń w drzewa decyzyjnego dla S .

Metoda:

1. Utworzyć wierzchołek w .
2. Jeżeli zbiór S zawiera wyłącznie *przykłady pozytywne*, to w jest liściem o etykiecie 1; **stop**.
3. Jeżeli zbiór S zawiera wyłącznie *przykłady negatywne*, to w jest liściem o etykiecie 0; **stop**.
4. Wśród wszystkich cech występujących w zbiorze S znaleźć cechę c o maksymalnym *zysku informacyjnym*.
5. Podzielić zbiór S na podzbiory S_1, \dots, S_n w których cecha c ma odpowiednio tę samą wartość; n jest liczbą wartości cechy c .
6. Nadać wierzchołkowi w etykietę c .
7. Dla każdego zbioru S_i ($i = 1, \dots, n$) skonstruować drzewo decyzyjne $ID3(S_i)$ i połączyć jego korzeń z wierzchołkiem w krawędzią o etykiecie reprezentującej wartość cechy c w zbiorze S_i .

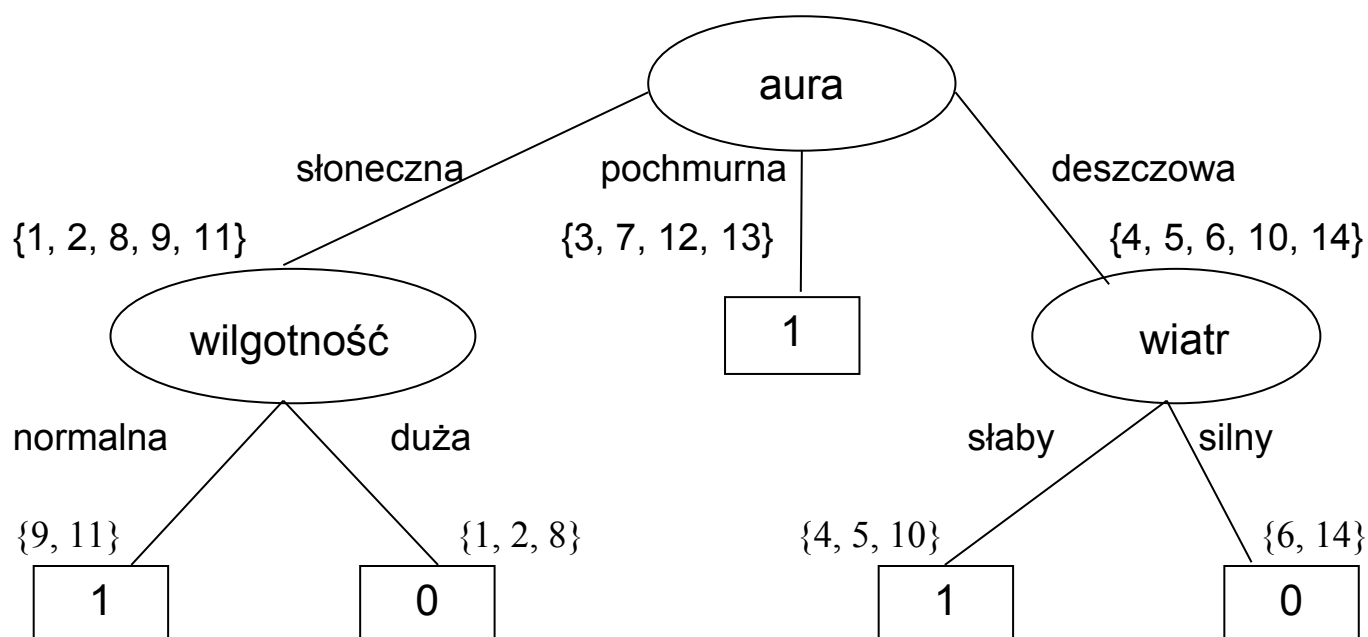
Pozyskiwanie wiedzy od ekspertów (3)

Drzewa decyzyjne (cd.) – przykład 1 (wg. [1])

Dany jest zbiór przykładowych decyzji eksperta w kwestii, czy przy danym stanie pogody można ($d(x) = 1$) albo nie można ($d(x) = 0$) grać w tenisa.

x	aura	temperatura	wilgotność	wiatr	d(x)
1	słoneczna	wysoka	duża	słaby	0
2	słoneczna	wysoka	duża	silny	0
3	pochmurna	wysoka	duża	słaby	1
4	deszczowa	umiarkowana	duża	słaby	1
5	deszczowa	niska	normalna	słaby	1
6	deszczowa	niska	normalna	silny	0
7	pochmurna	niska	normalna	silny	1
8	słoneczna	umiarkowana	duża	słaby	0
9	słoneczna	niska	normalna	słaby	1
10	deszczowa	umiarkowana	normalna	słaby	1
11	słoneczna	umiarkowana	normalna	silny	1
12	pochmurna	umiarkowana	duża	silny	1
13	pochmurna	wysoka	normalna	słaby	1
14	deszczowa	umiarkowana	duża	silny	0

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}



Regułowa reprezentacja wiedzy (1)

Wprowadzenie

- reprezentacja wiedzy dziedzinowej jest zbiorem **faktów** o postaci literałów oraz **reguł**
- ogólna postać reguły: IF $L_1 \otimes \dots \otimes L_n$ THEN W
gdzie wyrażenie L_i dla $i = 1, n$ jest literałem, a symbol \otimes oznacza koniunkcję lub alternatywę; całe wyrażenie $L_1 \otimes \dots \otimes L_n$ to tzw. **warunek** reguły
- w zależności od postaci wyrażenia W wyróżnia się reguły deklaratywne oraz proceduralne
- reguła deklaratywna: wyrażenie W jest **wnioskiem**, czyli koniunkcją atomów lub pojedynczym atomem, np.
IF *pacjenta_boli(gardło) \wedge Temp > 38,0* THEN *pacjent_choruje(angina)*
- reguła proceduralna: wyrażenie W jest **akcją**, czyli opisem jakiegoś działania, np.
IF *pacjent_choruje(angina)* THEN *podać_lek(antybiotyk)*
- wśród reguł deklaratywnych można wyróżnić podtypy, np. reguły kompozycji, reguły dziedziczenia i inne
- w skład bazy wiedzy mogą także wchodzić **metareguły**, czyli reguły wyrażające zasady stosowania innych reguł

Przykład 2 (wg. [1])

Drzewo decyzyjne z przykładu 1 można reprezentować w postaci następującego zbioru reguł.

IF *aura(słoneczna) \wedge wilgotność(duża)* THEN *nie_grać*

IF *aura(słoneczna) \wedge wilgotność(normalna)* THEN *grać*

IF *aura(pochmurna)* THEN *grać*

IF *aura(deszczowa) \wedge wiatr(silny)* THEN *nie_grać*

IF *aura(deszczowa) \wedge wiatr(słaby)* THEN *grać*

Regułowa reprezentacja wiedzy (2)

Przetwarzanie wiedzy regułowej

- istnieją dwie, podstawowe metody przetwarzania wiedzy regułowej
 - wnioskowanie progresywne (ang. *forward chaining*)
 - wnioskowanie regresywne (ang. *backward chaining*)
- przy doborze faktów oraz reguł w kolejnych krokach wnioskowania wykorzystuje się algorytmy:
 - unifikacji
 - dopasowania wyrażenia do wzorca (ang. *pattern matching*)
- warunki reguł można podzielić na:
 - odpytywalne (ang. *askable conditions*)
 - nieodpytywalne (ang. *non-askable conditions*)
- wiedzę przetwarza się zgodnie z założeniem o zamkniętości świata (ang. *closed-world assumption*); oznacza to, że dowolna formuła jest uważana za prawdziwą wtedy i tylko wtedy, gdy jest dowodliwa - w przeciwnym przypadku formułę tę uważa się za fałszywą
- termin „odpalenie reguły” oznacza dołączenie wniosku W tej reguły do bazy wiedzy, albo wykonanie działania W , jeżeli jest to akcja

Regułowa reprezentacja wiedzy (3)

Wnioskowanie progresywne

Algorytm *FC*

Dane: baza wiedzy *KB* obejmująca zbiór reguł *R* i zbiór faktów *F* oraz hipoteza *H*.

Wynik: sygnał *tak*, jeżeli $KB \vdash H$ albo sygnał *nie* w przeciwnym wypadku; nowy zbiór *F*.

Metoda:

1. Jeżeli $H \in F$ to **stop**(*tak*).
2. Skonstruować zbiór *C* złożony ze wszystkich reguł ze zbioru *R*, których warunki są spełnione, tj. dla dowolnej reguły postaci $L_1 \wedge \dots \wedge L_n \rightarrow W$ ze zbioru *C*, $L_i \in F$ gdzie $i = 1, n$.
3. Jeżeli $C = \emptyset$ to **stop**(*nie*), w przeciwnym wypadku uszeregować zbiór *C* zgodnie z przyjętymi kryteriami.
4. Odpalić pierwszą regułę ze zbioru *C*.
5. Przejść do kroku 1.

Regułowa reprezentacja wiedzy (4)

Wnioskowanie progresywne (cd.)

Popularne strategie szeregowania zbioru **C**:

- przyjąć naturalny porządek w zbiorze
- uszeregować zbiór począwszy od reguł, których warunki odnoszą się do przypadków wprowadzonych do bazy wiedzy jako ostatnie (tj. najpóźniej)
- uszeregować zbiór począwszy od reguł o najbardziej złożonych warunkach
- uszeregować zbiór zgodnie z wiedzą wyrażoną przez metareguły

Udoskonalenia algorytmu ogólnego

- „przyrostowe” konstruowanie zbioru **C**
- klasyfikowanie reguł ze względu na podobieństwo warunków
- grafowa reprezentacja zbioru reguł
- indeksowanie reguł ze względu na zmienne występujące w warunkach.

Powyższe udoskonalenia zrealizowano w algorytmie RETE (C.L. Forgy, 1974) wykorzystywanym w wielu popularnych systemach eksperckich (np. CLIPS) i w środowiskach do ich konstruowania (np. OPS5).

Regułowa reprezentacja wiedzy (5)

Wnioskowanie regresywne

Algorytm *BC*

Dane: baza wiedzy KB obejmująca zbiór reguł R i zbiór faktów F oraz hipoteza H .

Wynik: sygnał *tak*, jeżeli $KB \vdash H$ albo sygnał *nie* w przeciwnym wypadku.

Metoda:

1. Jeżeli $H \in F$ to **stop(tak)**.
2. Skonstruować zbiór C złożony ze wszystkich reguł ze zbioru R postaci $W \rightarrow A$, takich że $A \doteq H$.
3. Jeżeli $C = \emptyset$ to **stop(nie)**, w przeciwnym wypadku ze zbioru C wybrać dowolną regułę $L_1 \wedge \dots \wedge L_n \rightarrow A$.
4. Wykonać algorytm *BC* dla każdego L_i , gdzie $i = 1, n$. Jeżeli każdą z uzyskanych odpowiedzi jest *tak* to **stop(tak)**, w przeciwnym wypadku usunąć wybraną regułę ze zbioru C .
5. Przejść do kroku 3.

Objaśnienia:

- symbol W reprezentuje dowolną koniunkcję literałów
- symbol \doteq oznacza równość dwóch wyrażeń z uwzględnieniem unifikacji lub dopasowania do wzorca

Reprezentowanie wiedzy za pomocą ram

- koncepcja reprezentowania wiedzy za pomocą **ram** (ang. *frames*) pochodzi od Marvinina Minskyego (1975)
- baza wiedzy jest zbiorem klas i ich wystąpień (czyli obiektów)
- definicja klasy obejmuje **atrybuty**, ich **wartości domyślne** (opcjonalnie) i **metody**, wśród których można wyróżnić tzw. **demony**
- w zbiorze klas określa się relację dziedziczenia
- atrybuty obiektu mają wartości określone ("konkretne")
- do reprezentowania klas, jak również obiektów, służą ramy
- elementami ramy są **klatki** (ang. *slots*), które wypełnia się **fasetami** (ang. *facets*)
- fasetą jest parą **nazwa:wartość**; za pomocą faset określa się m. in. wartości przechowywane w klatce i ich typy.

Przykład 3 (wg. [6])

Reprezentacja klasy i obiektów za pomocą ram w języku FRL

Frame: expert system

Slots: AKO:

\$value: program

Programming environment:

\$require: (LISP, PROLOG, OPS5, C)

\$default: LISP

\$if-needed: "look up references".

Frame: MYCIN

Slots: AKO:

\$value: expert system.

Komunikacja z użytkownikiem

- ogólne metody komunikowania się systemu z użytkownikiem
 - język stylizowany na naturalny (systemy dialogowe)
 - język symboli graficznych (piktogramy, ikony)
- metody udzielania objaśnień
 - objaśnianie przez retrospekcję (ang. *retrospective explanation*)
 - objaśnianie kontrfaktyczne (ang. *counterfactual explanation*) i odpowiedzi intensjonalne (ang. *intensional answers*).

Konstruowanie systemów eksperckich

- języki programowania
 - języki prototypowania: np. PROLOG, LISP
 - języki docelowe: np. CLIPS, JESS, DROOLS, C++, JAVA
- wykorzystanie gotowych powłok
 - EMYCIN (powłoka systemu MYCIN)
 - KAS (powłoka systemu PROSPECTOR)
 - AGE (powłoka systemu HERSAY)
- środowiska narzędziowe, obejmujące w ogólności:
 - język wyrażania reguł (stylizowany na naturalny)
 - mechanizm indukcji reguł
 - edytor reguł (z mechanizmami weryfikacji poprawności bazy wiedzy)
 - bibliotekę mechanizmów wnioskujących
 - kompilator reguł na język programowania (np. C)
 - mechanizm udzielania objaśnień (w języku quasi-naturalnym)
 - mechanizm śledzenia wywodów
- przykładowe systemy: RULEMASTER (Radian Corp.), FLEX (LPA), Personal Consultant Plus (Texas Instruments), OPS5, KES (Software Architecture & Engineering Inc.), KEE (Intelli Corp.), SRL (Uniwersytet Carnegie-Mellon).

Podsumowanie

- zalety systemów eksperckich (w porównaniu z ekspertami-ludźmi)
 - dostępność
 - szybkość
 - dostępność objaśnień
 - odporność na „ludzkie” błędy
- wady i ograniczenia systemów eksperckich:
 - duże koszty wytworzenia
 - trudności z samodzielnym pozyskiwaniem nowej wiedzy
 - trudności z korzystaniem z wiedzy zdroworozsądkowej (ang. *commonsense knowledge*)
- weryfikowanie i testowanie systemów eksperckich
 - kryteria formalne: np. niesprzeczność, pełność zgromadzonej wiedzy
 - kryteria nieformalne: np. „konsekwencja”, użyteczność
- kierunki rozwoju systemów eksperckich
 - nowe metody skutecznego pozyskiwania wiedzy eksperckiej
 - nowe rozwiązania w zakresie komunikowania się systemu z użytkownikiem (system „przyjazny użytkownikowi”)
 - nowe metody przetwarzania dużych baz wiedzy
 - badania nad metodami reprezentowania wiedzy niepełnej, niepewnej i zmieniającej się w czasie.