

Umelá inteligencia a expertné systémy

HI

3.ročník bakalárské štúdium

RNDr. Eva Rakovská, PhD.



Literatúra:

- *Kelemen, Liday: Expertné systémy pre ekonómov, Skriptum, Ekonóm, 1996*
- *Pavol Návrat a kol.: Umelá inteligencia, STU Bratislava, 2002*
- *Jozef Kelemen: Základy umelej inteligencie, Alfa, Bratislava*
- *moodle. euba.sk, kurz UIES 3HI*
- *Literatúra na moodle.euba.sk*



Čo je to umelá inteligencia?

- *Systémy, ktoré myslia ako l'udia (kognitívna veda)*
- *Systémy, ktoré konajú ako l'udia (Turingov test 1950) – spracovanie prirodzeného jazyka, reprezentácia poznatkov, automatizované usudzovanie, učenie sa*
- *Systémy, ktoré myslia rozumne (správne myslenie – logika)*
- *Systémy, ktoré konajú rozumne (dosiahnutie svojich vopred daných ciel'ov)*



Porovnanie umelej a prirodzenej inteligencie.

- *UI:*
 - *Je stabilnejšia*
 - *Nemožno ju rozmnrožovať a rozširovať*
 - *Je lacnejšia*
 - *Je stála*
- *Prirodzená inteligencia:*
 - *Je tvorivá*
 - *Používa vnemy*
 - *Kontext skúseností*
 - *intuícia*



Čo zahŕňame do oblasti UI?

- *Expertné a znalostné systémy*
- *Neurónové siete*
- *Genetické algoritmy*
- *Distribuovaná UI*
- *Multiagentové systémy*
- *Nové web technológie (sémantické siete, ontológie....)*



Pojem „agent“

- *Agent je vlastne konatel', mn.číslo agenty*
- *Agent*
 - *vníma svoje prostredie (senzory),*
 - *je s ním v spätej väzbe (efektory),*
 - *koná na základe informácií z prostredia,*
 - *je opísaný je bázou poznatkov*



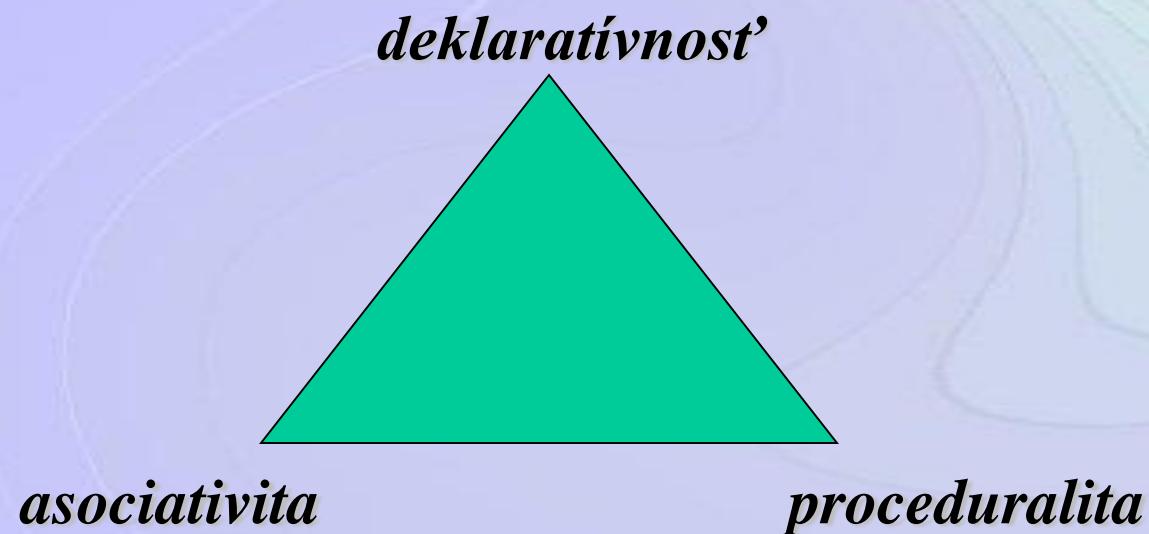
Poznatky – znalosti.

- *Úroveň abstrakcie*



Atribúty poznatku.

Tri základné vlastnosti:



Deklaratívnosť = vyjadritel'nosť

Proceduralita = použitel'nosť (poznanok je informacia v akcii)

Asociativita = poznatky navzájom súvisia, nedajú sa izolovať



Stručný prehľad histórie UI.

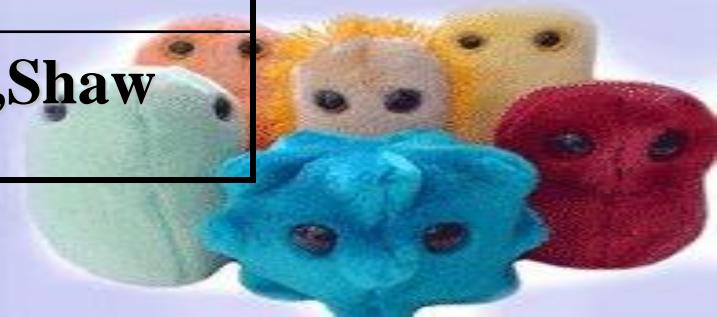
- *Umelá inteligencia je horšia než prirodzená blbost'.*

Murphyho počítačové zákony



Stručný prehľad histórie UI a ES.

1943	model neurónu	McCulloch, Pitts
1950	Turingov test	Turing
1956	Darthmouthská konferencia	Minsky, Simon, Newell, McCarthy
1957	Perceptron	Rosenblatt
1957	formálne gramatiky	Chomsky
1957	GPS	Newell, Simon, Shaw



Stručný prehľad histórie UI a ES.

1973	MYCIN	Shortliffe, Buchanan
1975	rámce	Minsky
1978	R1-XCON	McDermott
1979	ReTe algoritmus	Forgy
1981	jap. projekt počítačov piatej generácie	
1982	Hopfieldova neurón. siet'	Hopfield



Riešenie problémov UI.

- *obecné riešenie úloh (prehl'adávanie stavového priestoru)*
- *percepcia (reč, videnie)*
- *porozumenie prirodzeného jazyka*
- *tvorba expertných systémov*
- *dokazovanie teorém (hypotéz)*
- *hranie hier (strategické postupy v ekonomike)*
- *využitie algoritmov a postupov umelej inteligencie pri získavaní dát (datamining, datawarehouse)*
- *hl'adanie minimálnej cesty (obchodný cestujúci)*
- *navigovanie robotov (strojové učenie)*
- *navrhovanie technických a technologických postupov*



Stavový priestor (riešenie problémov)

- *univerzum, v ktorom systém (agent) pracuje je popísaný v určitom čase stavmi (zmenami stavov)*
- *stavový priestor-abstraktný priestor stavov sveta, v ktorom dochádza k riešeniu problému*
- *stavový priestor-charakterizovaný množinami stavov a operátorov*
- *stav-reprezentovaný symbolovou štruktúrou*
- *operátor-reprezentovaný výpočtovou procedúrou*
- *Ciel'ový stav: s_t (konfig. na konci riešiaceho postupu)*
 - *Začiatočný stav problému: s_0 (konfig. pred začatím riešenia)*



Stavový priestor (riešenie problémov)

Definujeme: *Problém je dvojica $P_s = (s_0, S_t)$*

- *ciel'ový stav nemusí byť určený jenoznačne tj. môže byť množina ciel'ových stavov $\Rightarrow P_s = (s_0, T), T \subseteq S$*
- *problémy:*
 - *riešitel'né*
 - *neriešitel'né*
 - *patologický prípad-triviálne problémy, kedy nemusíme použiť ani jeden operátor na riešenie ($s_0 \in T$)*
- *Riešenie problému-postupnosť „zmien konfigurácií“ postupnosť krokov, postupnosť použitia jednotlivých operátorov*



Riešenie problému.

- *Hľadanie riešenia:*
 - vytváranie usporiadaných postupností a zistovanie, či náhodou nejde o riešenie

Poznámka: na formulovanie stavového priestoru a problému je možné použiť teóriu grafov (riešenie problému je cesta od počiatočného do cieľového stavu)

Priklady: hlavolam s posúvateľnými poličkami - zjednodušený prípad, ktorý sa štandardne používa v UI pri skúmaní algoritmov heuristického hľadania; problém 8 dám; kryptooritmetické hlavolamy



Charakteristiky problémov

- *podľa typu riešenia:*
 - *riešením je cesta*
 - *riešením je stav (resp. množina stavov)*
- *podľa rozložiteľnosti problému*
 - *ak sa dá problém rozložiť na podproblémy*
 - *ak sa nedá rozložiť*
- *podľa spôsobu práce s krokmi v procese riešenia*
 - *s ignorovateľnými krokmi riešenia*
 - *s odčinitel'nými krokmi riešenia*
 - *s neodčinitel'nými krokmi riešenia*



Procedúry hľadania:

Kritériá efektívnosti algoritmu hľadania:

- **Úplnosť**=ak existuje riešenie problému, algoritmus ho vždy nájde
- **časová zložitosť**
- **pamäťová zložitosť**
- **Prípustnosť**= ak existuje aspoň jedno optimálne riešenie problému algoritmus ho vždy nájde (nemusí nájsť všetky)

Procedúry neinformovaného hľadania:

- a) základná procedúra slepého prehľadávania
- b) prehľadávanie do šírky
- c) prehľadávanie do hĺbky
- d) obojsmerné prehľadávanie
- e) nedeterministické prehľadávanie

