

Modelovanie na báze reaktívnych agentov

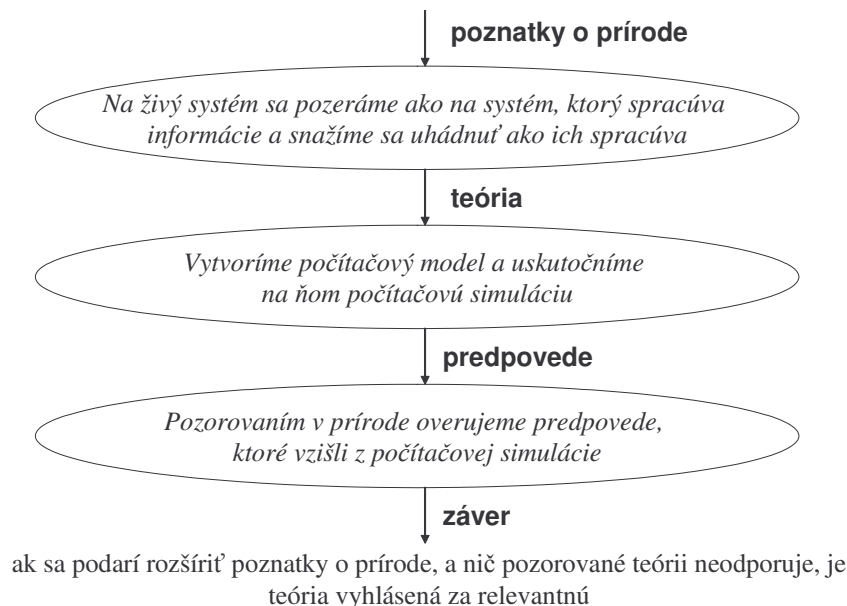
Andrej Lúčny¹

Abstrakt. Pokiaľ modelujeme pozorované správanie nejakého živého tvora, musíme si dať pozor na to, aby sme nemodelovali samotné pozorovanie, ale štruktúry, ktoré sú zaň v danom tvorovi zodpovedné. Môžeme pritom postupovať tak, že sa snažíme povahu týchto štruktúr uhádnuť a z vytvoreného modelu, ktorý vysvetľuje dané pozorovanie, získať určité testovateľné predpovede. Tento postup demonštrujeme na príklade a uvádzame pritom vlastnú predstavu o povahe týchto štruktúr.

1 Fyzikálny prístup k modelovaniu biologických štruktúr

Pokiaľ pozorujeme správanie nejakého jednoduchého tvora a snažíme sa ho namodelovať počítačovou simuláciou, musíme sa vyvarovať jedného častého omylu. Spočíva v tom, že miesto toho aby sme modelovali tvora, ktorý sa pozorovaným spôsobom správa, modelujeme pozorované správanie. Taký model dané správanie ľahko vygeneruje, ale v skutočnosti nepovie vôbec nič, len zaznamená naše pozorovanie. Nás ale nezaujíma ani tak dané správanie ako spôsob, akým povstáva v danom tvorovi z jeho vnútorných štruktúr.

O tom ako sú tieto štruktúry organizované máme už z biológie určitú predstavu. V detailoch ju však pre účely počítačovej simulácie musíme značne dotvoriť, čím vlastne hádame aké softwarové štruktúry v počítači sú analogické biologickým štruktúram zodpovedným za generovanie správania.



Obrázok 1. Fyzikálny prístup k modelovaniu biologických štruktúr

Táto ucelená predstava potom definuje všeobecné možnosti v rámci ktorých sa snažíme nájsť také konkrétne štruktúry, ktoré dané správanie generujú. Takýto model však

¹ KAI FMFIUK Bratislava a MicroStep-MIS, andy@microstep-mis.com, www.microstep-mis.com/~andy

nielen generuje pozorované správanie, ale dokáže poskytnúť množstvo predpovedí, pre situácie odlišné od pozorovaných, ktoré sa potenciálne dajú spätne overiť dodatočným pozorovaním.

Na základe toho si potom môžeme urobiť aj úsudok o tom nakoľko kvalitne sme uhádli biologickú relevantnosť štruktúr použitých pri simulácii.

Napodobňujeme tu teda spôsob akým sa vo fyzike skúma vnútorná štruktúra hmoty. Určitý experiment tu vedie k postulovaniu teórie zahrňujúcej určitú predstavu o štruktúre, ktorá ho vysvetľuje. Zároveň ale táto teória generuje určité predpovede, ktoré sa dajú overiť ďalšími experimentmi. Keď sú predpovede potvrdené, posilňuje to presvedčenie, že ide o dobrú teóriu, hoci nikto netvrdí, že je dokonalá (obrázok 1).

Takto získané poznatky potom predstavujú nielen určité biologicky relevantné poznanie, ale aj zdroj inšpirácie pre programovanie tých umelých systémov, ktoré živé systémy do istej miery napodobňujú.

2 Príklad modelovaného správania

Za príklad na demonštrovanie našej myšlienky si zvolíme modelovanie zakladania potomstva kutavkou [4]. Ide o jedno z najzložitejších individuálnych správání v ríši hmyzu vôbec a zjednodušujúcim faktorom tu je, že toto správanie je evidentne vrodené (žiadna kutavka nevidí svoju matku zakladať potomstvo).

Kutavka je samotárska osa (obrázok 2) známa najmä pre hrôzostrašný spôsob akým zabezpečuje pre svoje larvy dostatok čerstvej potravy: žihadlom pichne svoju korisť (každý druh loví špecifickú obeť z radov chrobákov, napríklad sedlovku) následkom čoho táto ochrne a odtiahne ju do podzemnej komôrky, kde na ňu nakladie vajíčko. Z toho sa vyľahne larva, ktorá potom paralyzovanú obeť za živa zje, pričom jej to trvá niekoľko mesiacov. Kutavka pritom realizuje vždy rovnaký rituál²: najprv vyhrabe komôrku, potom uloví obeť, položí si ju pred komôrku, vlezie do komôrky skontrolovať ju, potom do nej vtiahne obeť, znesie na ňu vajíčko, vylezie z komôrky a uzavrie jej vchod.



Obrázok 2. Kutavka

S týmto správaním sa dá zaujímavo pohrať. Pokiaľ by ste kutavke vo fáze, keď vlezie na kontrolu do komôrky, sedlovku odtiahli, ale nie príliš ďaleko – aby ju kutavka ľahko našla, opäť by ju priniesla ku komôrke. Ale opäť by ju položila pred vchod a ide vykonať kontrolu (ktorú už práve pred chvíľou vykonala), čiže vlezie do komôrky. Takto má experimentátor možnosť opakovať odtiahnutie sedlovky od vchodu a udržiavať kutavku v bezvýchodiskovom cykle. Kutavka 30-40 krát zakaždým sedlovku pritiahne ku vchodu komôrky, ale nepoučí sa a nevtiahne ju hneď dnu, naopak, ide vykonávať

² Tento rituál sa odlišuje v rámci jednotlivých druhov kutaviek, tu uvádzame podobu z [4]

kontrolu, ktorú už vykonala mnoho krát a s pričinením experimentátora opäť o sedlovku príde. E. Gál položil v [3] provokujúcu otázku: čo stojí za týmto z nášho hľadiska dosť tupým správaním kutavky?

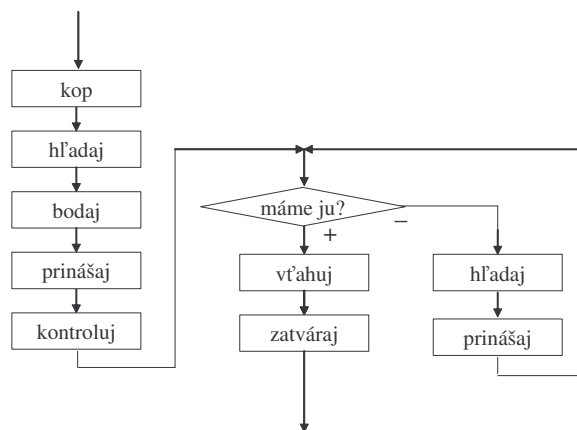
Nejaké jednoduché vysvetlenie tu neobstojí, lebo napríklad na rozdiel od kontroly komôrky, pri opätovnom nájdení odtiahnutej sedlovky, ju kutavka nejde druhý raz pichať žihadlom. Určité fázy vie teda zopakovať, iné preskočiť.

Navyše kutavka si evidentne dokáže zapamätať, že už vykopala komôrku a dokonca kde to bolo. Jej schopnosti učiť sa orientovať pri lete podľa význačných pozemných objektov, boli jednoznačne preukázané.

S predstavou, že kutavka je z hľadiska zložitosti svojho riadiaceho mechanizmu len jednooký medzi slepými (teda relatívne jednoduchý tvor vykonávajúci pomerne zložité správanie) sa teraz pokúsime zostrojiť počítačový model uvažovaného správania.

3 Tradičný model

Pokiaľ by sme chceli zvolené správanie zaznamenať algoritmickým spôsobom, zrejme by sme na úvod vytvorili niečo podobné obrázku č. 3. Pokiaľ by sme to naozaj spravili takto, dostali by sme sa k už uvedenému problému zámenny modelovania pozorovania a pozorovaného. Simulátor založený na interpretácii takéhoto algoritmu totiž na jednej strane vykazuje výbornú zhodu so zvoleným pozorovaním, avšak na strane druhej nič iné než pozorované správanie nedokáže vygenerovať.

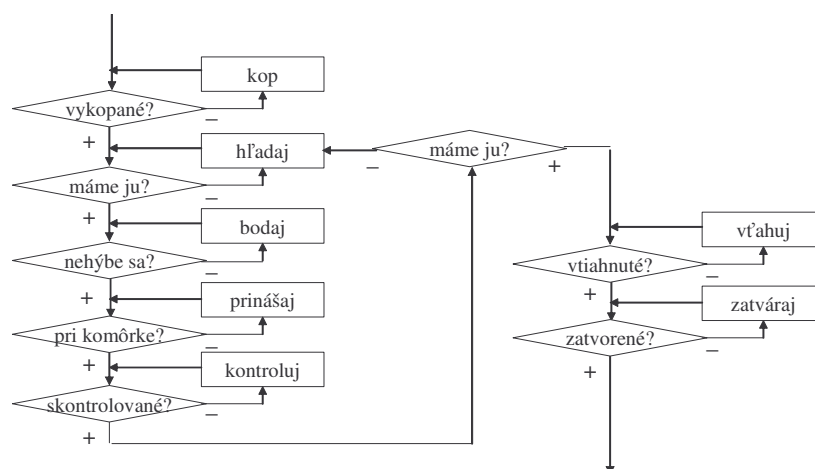


Obrázok 3. Modelovanie pozorovania (tradičný postup)

Ak potom simulátor čo len trochu vychýlime zo podmienok zvoleného experimentu, napríklad počas kontroly komôrky ochrnutú sedlovku odstránime a nahradíme neochrnutou, dostaví sa zlyhanie. Pri prechode od hľadania k prinášaniu sa prinášanie s neochrnutou sedlovkou nepodarí, čo z hľadiska počítačovej simulácie zodpovedá vygenerovaniu výnimky.

Príroda si ale vždy nájde cestu, čo môžeme premietnuť aj do daného algoritmického vyjadrenia modelu. Riešenie na obrázku 4 napríklad predpokladá, že každá výnimka je ošetrená vynechaním danej fázy. Takýto model už dokáže generovať určité predpovede, napríklad, že ak vymeníme sedlovku za neochrnutú, tak ju kutavka bodne, zatiaľ čo pri ponechaní pôvodných podmienok bodnutie vynechá. Potiaž je, že tu ľahko nájdeme aj predpovede načisto nesprávne, napríklad, že keď zakopeme komôrku, tak sa ju do nekonečna snažíme kontrolovať. Dotiahnutie tohto modelu by si

vyžadovalo veľké množstvo rôznych podmieňovacích príkazov. Boli by sme samozrejme radšej keby sme ich vedeli zakódovať do modelu čo najúspornejším spôsobom.



Obrázok 4. Čiastočný prechod k modelovaniu pozorovaného (tradičný postup)

Ďalšie problémy predstavujú časové parametre modelu. Oba uvedené modely napríklad nevysvetľujú ako dôjde k tomu, že kutavka napriek dlhému cykleniu nakoniec pokusy o kontrolu komôrky zanechá. Doplnenie by tu spočívalo v inkrementálnom počítaní „netrepezlivosti“ pri prechodoch cyklami alebo volaním generátora náhody v podmienkach, čo by za ešte skomplikovalo celý model.

Zvyšovanie zložitosti modelu sa pritom nepriaznivo odráža nielen v našej schopnosti mu porozumieť, ale aj v znižovaní jeho schopnosti generovať predpovede. Tie totiž zodpovedajú tomu, že explicitným zanašáním určitých vlastností do modelu v skutočnosti pridávame aj niečo čo sme vlastne pridať nezamýšľali, niečo čo zanašame implicitne. A pri štruktúrach tradičného programovania sa takáto situácia objavuje načisto výnimočne.

4 Alternatívny model

Potrebovali by sme teda do nášho modelu vložiť takú predstavu o štruktúrach generujúcich uvažované správanie, kde jednotlivé jemné odtiene a variácie tohto správania implicitne povstávajú z relatívne malej informácie, explicitne uloženej v týchto štruktúrach.

Samozrejme tutovkou sa tu zdá byť voľba štruktúr veľmi podobných biologickým korelátom ako sú neurónové siete. My si volíme cestu inú a to takú, že sa snažíme odpozorovať len všeobecné vlastnosti biologických štruktúr, presnejšie povahu ich modularity, organizácie a aktivácie. Vede nás k tomu fakt, že pri tomto prístupe sme stále blízko tradičného programovania, pomocou ktorého – na rozdiel od súčasných neurónových sietí – vieme dosiahnuť jednak škálovateľnosť riešení, jednak jasné rozlíšenie medzi vrozeným a naučeným. (V princípe je ale naša cesta len jedna z mnohých.)

Takže akými dôležitými vlastnosťami sa vyznačujú biologické štruktúry produkujúce správanie? Podľa nás týmito štyrmi:

1. **decentralizovanou modularitou s vysokým paralelizmom.** O paralelnej povahe spracovania informácií v biologických systémoch zrejme nikto nepochybuje. O tom, že z hľadiska riadenia ide o decentralizovaný systém svedčí rôznorodosť a vysoká špecifickosť porúch, ktoré môžu vplyvom zranenia, alebo dedičnej vady nastať.
2. **nízkou kontinuitou vykonávania inštrukcii.** Na rozdiel od počítača, ktorý dokáže interpretovať rutinu pozostávajúcu z neobmedzene dlhej sekvencie inštrukcii, v prírode je s niečím podobným problém. Pri jednoduchších tvoroch ide z hľadiska určitého správania o jednotky, maximálne desiatky úkonov. Aj medzi ľuďmi sú jedinci disponujúci schopnosťou čo i len zapamätať si dlhú sekvenciu dát nie tak dokonalí, ale tak retardovaní.
3. **obmedzenou časovou platnosťou operačných dát.** Na rozdiel od počítača, kde sú údaje dokonale perzistentné a ich zabúdanie sa musí špecificky implementovať, v prírode je veľa indikácií (vrátane termodynamickkej povahy pamäti), že problém je skôr dáta trvalo uchovať.
4. **možnosťou infiltrácie kooperácie dvoch modulov inými modulmi, pričom prevláda potláčanie a korekcia aktivity nad jej vyvolávaním.** Je známych niekoľko experimentov, ktoré svedčia v prospech toho, že hierarchicky vyššie vrstvy skôr potláčajú a usmerňujú aktivitu nižších, než že by ju vyvolávali. Napríklad keď (dostatočne jemne) dekapitujeme pakobyľku, tak pekne rovnomerne kráča. Jej nervové uzly v hlave teda uzly v hrudi nestimulujú, ale potláčajú. Nie sú teda s nimi zapojené v sérii, ale hierarchicky narúšajú komunikáciu v uzloch hrude.

Náš alternatívny model bude preto pozostávať z viacerých paralelných jednotiek. Budeme ich nazývať agentami. Ich vzájomná komunikácia bude realizovaná pomocou globálnych premenných, ktoré môže viacero agentov (naraz) zapisovať aj čítať. Každý agent bude vykonávať pomerne krátku a jednoduchú sekvenciu inštrukcií, a to buď pravidelne v čase, alebo na základe zmeny hodnoty niektorej globálnej premennej. Oproti tradičnému chápaniu premenných sa tie naše budú vyznačovať dvomi novinkami:

- hodnota sa do premennej bude dať zapísať aj na obmedzený čas, po ktorom sa stratí
- hodnote budeme môcť definovať prioritu, ktorá spôsobí, že hodnota sa do premennej vôbec nepriradí ak je v nej hodnota s vyššou prioritou

Časovú platnosť i prioritu definuje agent pri priradení hodnoty do premennej. Kým časovú platnosť potrebujeme podľa bodu 3, priority nám umožnia definovať hierarchiu pri naplňaní bodu 4.

Vyššie uvedené je podstatou nami navrhnutej architektúry agent-space, prednostne určenej na tvorbu riadiacich systémov napr. mobilných robotov. Jej technické detaily sú pomerne zložité a čitateľ ich môže nájsť v [8]. Na pochopenie myšlienky tohto článku však nie sú nevyhnutné.

Z filozofického hľadiska každý agent predstavuje v systéme určitú zabudovanú reakciu, určitý reflex. Možno teda povedať, že tu ide o snahu **vymodelovať správanie systému ako kolekciu reflexov nad vnútornou pamäťou, ktorá určitú špeciálnu povahu** (potenciálne ohraničenú časovú platnosť a priority).

Podme sa teraz pozrieť ako to bude vyzeráť na zvolenom príklade. V modeli budeme mať niekoľko globálnych premenných reprezentujúcich vnímanie kutavky:

nestDetected	kutavka sa nachádza pri svojej komôrke
victimDetected	kutavka vidí sedlovku
paralyzedVictimDetected	kutavka vidí sedlovku, ktorá sa nehýbe
insideNest	kutavka vidí, že je vo vnútri komôrky
nestClosed	kutavka vidí uzavretú komôrku

ktoré dopĺňa premenná reprezentujúca vnímanie vlastného tela:

inseminated	kutavka je oplodnená
-------------	----------------------

Ďalej budeme mať premenné reprezentujúce vnútornú pamäť kutavky:

nestReady	komôrka je vykopaná
checkingPerformed	kontrola komôrky je vykonaná
checking	kontrola komôrky prebieha
victimInNest	sedlovka je v komôrke

a na niekoľko základných správání (ktoré by sme si ale v detailoch tiež vedeli predstaviť ako zložené z atomickejších úkonov realizovaných podobnými agentovými štruktúrami – na to by sme však museli uvažovať jemné detaily tela kutavky, čo by v tejto chvíli bolo zbytočne zložité):

digNest()	kop komôrku
lookForVictim()	hľadaj sedlovku
paralyzeVictim()	paralyzuj sedlovku
carryToNest()	zanes paralyzovanú sedlovku ku komôrke
checkNest()	kontroluj komôrku
pullIntoNest()	vťahni paralyzovanú sedlovku do komôrky
closeNest()	uzavri vchod do komôrky

Pod ich „zavolaním“ si musíme predstaviť vypustenie výstupu z daného správania na aktuátory kutavky na relatívne krátky čas, presnejšie pokým nejaký iný agent „nezavolá“ správanie iné. Ak teda v kóde „zavoláme“ digNest() myslíme tým len malý krôčik v kopaní hniezda a „volanie“ agenta sa vôbec nezablokuje (tradične by sme predpokladali, že ho zablokuje, až kým sa nevykope celé hniezdo). Modelované správanie kutavky potom možno vysvetliť súčinnosťou siedmich agentov:

Agent1

```
for (each 1 second)
  if (inseminated && !nestReady) {
    if (insideNest) nestReady = true for 1 day;
    else digNest();
  }
```

Agent2

```
for (each 1 second)
  if (nestReady && !victimDetected)
    lookForVictim();
```

Agent3

```
for (each 1 second)
  if (victimDetected && !paralyzedVictimDetected)
    paralyzeVictim();
```

Agent4

```
for (each 1 second)
  if (paralyzedVictimDetected && !nestDetected)
    carryToNest();
```

Agent5

```
for (each 1 second)
  if (!checkingPerformed && ((paralyzedVictimDetected
    && nestDetected) || checking) {
    checking = true for 7 seconds;
    checkNest();
    if (insideNest)
      checkingPerformed = true for 40 seconds;
  }
```

Agent6

```
for (each 1 second)
  if (checkingPerformed && paralyzedVictimDetected
    && !victimInNest)
    if (insideNest) victimInNest = true for 1 hour;
    else pullIntoNest();
```

Agent7

```
for (each 1 second)
  if (victimInNest & !nestClosed) closeNest();
```

V tomto modeli už pravda nenahliadneme tak ľahko, čo sa vlastne za akých podmienok robí a ak sa nechceme pomýliť, treba prejsť ku počítačovej simulácii. V princípe sa ale za normálnych okolností agenty aktivizujú v uvedenom poradí, t.j. istý čas je aktívny agent1, potom agent2, atď. Avšak spôsob ako jeden agent aktivizuje druhého je rôzny od prípadu k prípadu.

Niekedy je aktivácia sprostredkovaná následkom činnosti predchádzajúceho agenta, napríklad paralyzovanie obete sa spúšťa následkom toho, že ju nájdeme. V tomto prípade teda nie je potrebná vnútorná pamäť.

V iných prípadoch treba aktiváciu realizovať cez vnútornú pamäť, ale pokiaľ ide o dlhodobu platný údaj, ako napríklad, že komôrka je vykopaná, dá sa to ešte vyjadriť dosť jednoducho.

Najzložitejšie vyjadrenie vyžaduje práve tá inkriminovaná kontrola komôrky, ktorá sa dá oprieť výlučne o vnútornú pamäť, lebo prostredie sa jej vykonaním nemení a navyše ide o krátkodobo platné zistenie. Musí sa tam nejako zabezpečiť aby kutavka po istom čase kontrolu znovu vykonala a zabrániť, aby tak neurobila ihneď po skončení predošlej kontroly. Navyše túto fázu štartujú podmienky, ktoré už tesne po jej zahájení prestanú platiť (ulovenú sedlovku necháme pri kontrole vonku). To sa najjednoduchšie dosiahne použitím časovej platnosti premenných. Akákoľvek eliminácia časových

platností, tu vyžaduje implementáciu zabúdania a povedie teda k zložitejšiemu kódu v agentoch. Tupé správanie kutavky sa tu teda dá vysvetliť na základe toho, že kým kutavka sedlovku opäť nájde, zabudne, že komôrku skontrolovala. Príroda si vybrala túto možnosť zrejme preto, že jej kód je jednoduchší, hoci logicky nie je stopercentne správny. Navyše treba vziať v úvahu, že sa uvažované zlyhanie prejavuje za podmienok, ktoré sa v prírode nevyskytujú a teda nemôže nijako ovplyvniť evolučný výber. Zbytočnou kontrolou v nastrojených podmienkach sa tu teda asi platí za jednoduchosť riadiacich štruktúr postačujúcich v prirodzených podmienkach.

Čo sa týka vysvetlenia, prečo po čase – hoci dlhom čase - kontrolu predsa len kutavka zanechá, v rámci tohto modelu možno odpovedať tým, že opätovné nájdenie sedlovky netrvá vždy rovnako dlho a opakovaním sa môže zdokonaľovať. Tým sa môže podať kutavke vrátiť sa ku komôrke v čase, keď ešte časová platnosť kontroly nevypršala. (Navyše, keď zvažíme biologické možnosti implementácie časovej platnosti – časté zapisovanie takého údaju by zrejme viedlo k posilneniu schopnosti zapamätať si ho na dlhší čas a tým pádom by sa po väčšom počte opakovaní platnosť kontroly dostatočne predĺžila.)

Výhodou tohto modelu je, že okrem zhodovania sa s normálnym správaním i správaním počas uvažovaného experimentu dáva dobré predpovede pre iné okrajové situácie. Predovšetkým zahŕňa v sebe návrat k vhodným predošlým fázam v prípade výskytu výnimky. Napríklad, keď počas prinášania, kutavka sedlovka vypadne, model automaticky prejde do fázy hľadania. Ak pri opätovnom prinášaní nájde inú – ešte neuloženú – sedlovku, paralyzuje ju. Ak však nájde tú istú, paralyzovanie vynechá. V kritickom prechode od kontroly k vtiahnutiu sa na základe straty sedlovky vráti do fázy hľadania.

Vieme však urobiť aj zaujímavé predpovede, u ktorých nevieme odhadnúť ako by dopadlo ich otestovanie v prírode? Hneď niekoľko:

1. čo sa stane, keď pri kontrole ochrnutú sedlovku odstránime a dáme miesto nej inú nechrnutú, ale zato napríklad uviazanú na nitku aby neušla? Z modelu vychádza, že by ju mala paralyzovať a vtiahnuť.
2. aký vplyv bude mať vzdialenosť odtiahnutia sedlovky na počet opakovaní? Tu nám vychádza, že ak bude vzdialenosť dostatočne malá, žiadne opakovanie sa nedostaví. Inak vo všeobecnosti, čím menšia vzdialenosť, tým menej opakovaní.
3. dá sa zväčšiť počet opakovaní z tých 30-40? Podľa modelu by sa to malo podať tak, že budeme stále vhodne zväčšovať vzdialenosť o ktorú sedlovku odťahujeme.
4. čo sa stane keď komôrku zakopeme? Tu sa zo značnej časovej platnosti údaju o vykopaní komôrky dá usúdiť, že kutavka sa dosť dlho bude správať načisto bezradne – dlho bude hľadať neexistujúcu komôrku. Nakoniec si však predsa len vykope komôrku novú.
5. Bude sa pri opakovaní prinesenia sedlovky čas tohto prinesenia skracovať (tréningom) alebo predlžovať (únavou)? Pre náš model by bolo priaznivé keby sa rekordný čas stále zmenšoval (kolísat' pritom môže koľko chce).

Keby nejaký etológ zistil – či už z existujúcich prác alebo vykonaním experimentu, nakoľko sú tieto predpovede úspešné, dozvedeli by sme sa nielen či sme spravili dobrý model, ale by sme aj posilnili či oslabili našu vieru, že správne hádame s akými štruktúrami príroda pracuje.

5 Záver

V tomto príspevku sme sa snažili čitateľa presvedčiť, že pri modelovaní živých systémov v počítači by sme sa nemali zameriavať na ich vonkajšie prejavy, ale na ich vnútorné štruktúry, ktoré sú za ne zodpovedné. Žiadali sme pritom, aby sme modelovaním skúmali povahu týchto štruktúr podobným spôsobom ako sa vo fyzike skúma štruktúra hmoty, predovšetkým aby tieto modely poskytovali dostatočne zaujímavé predpovede, ktoré sa potenciálne dajú otestovať.

Na príklade z etológie hmyzu sme demonštrovali vlastnú predstavu o povahe týchto štruktúr, ktorá je síce len jednou z mnohých, ale ktorá sa vyznačuje zaujímavým potenciálom. Otvorili sme také základné otázky z počítačovej simulácie ako „čo je premenná?“ a „ako jeden kus kódu volá druhý?“. Ukázali sme aký význam má uvažovať u premennej obmedzenú časovú platnosť a prioritu a aký význam má modularita založená na agentoch. V podstate, ak je vôbec slušné povedať, že príroda programuje svoje výtvary, tak my k tomu dodávame, že používa trochu iné premenné a inak organizovaný kód, než si bežne pod programovaním predstavujeme. S tým asi bude súhlasiť takmer každý, my ale máme aj určitú konkrétnu predstavu aký charakter tieto premenné a kód majú.

Ako vedľajší efekt sme sa snažili ukázať, že niektoré zaujímavé vlastnosti získavajú živé systémy prostoduchejším spôsobom, než by sme predpokladali. Doslova ich majú vďaka svojej nedokonalosti. Jednoznačným dôsledkom tohto faktu je, že pokiaľ chceme zostrojiť stroje s podobnými vlastnosťami, mali by sme túto nedokonalosť použiť ako konštrukčný princíp. Toto zároveň poukazuje na rozdiely medzi strojmi a živými tvormi. Vo svojom správaní sa zrejme nepodobajú práve preto, že pri konštrukcii našich strojov sa snažíme o ich dokonalosť.

Literatúra

- [1] Brooks, R.: *Cambrian Intelligence*, The MIT Press, Cambridge, Mass., 1999.
- [2] Brooks, R.: *Robot – The Future of Flesh and Machines*. Penguin Books, London, 2002
- [3] Gál, E.: *Intuitívna psychológia a kognitívne vedy*. In: Kognitívne vedy III (Kvasnička V., Pospíchal J. eds.), CHTF, STU, Bratislava, 2000
- [4] Hass, H.: *The Human Animal: The Mystery of Man's Behavior*. G. P. Putnam's Sons., 1970, pp. 27
- [5] Kelemen, J.: *Strojovia a agenty*. Archa, Bratislava, 1994
- [6] Kelemen, J.: *The Agent Paradigm*. Computing and Informatics, Vol.22. (2003), pp. 513-519
- [7] Lúčny, A.: *Hľadanie kvalitatívneho rozdielu*. In: Kognice a umělý život (Kelemen, J. – Kvasnička, V. – Pospíchal, J., eds.), FPF SU, Smolenice, 2001, 167-176
- [8] Lúčny, A.: *Building Complex Systems with Agent-Space Architecture*. Computing and Informatics, Vol. 23 (2004), pp. 1001-1036
- [9] Minsky, M.: *The Society of Mind*. Simon&Schuster, New York, 1986
- [10] Minsky, M.: *Emotion machine*. <http://web.media.mit.edu/~minsky/>, 2004