Seminár z UI

Subsumpcia – základná myšlienka tvorby systémov s komplexným správaním

Andrej Lúčny

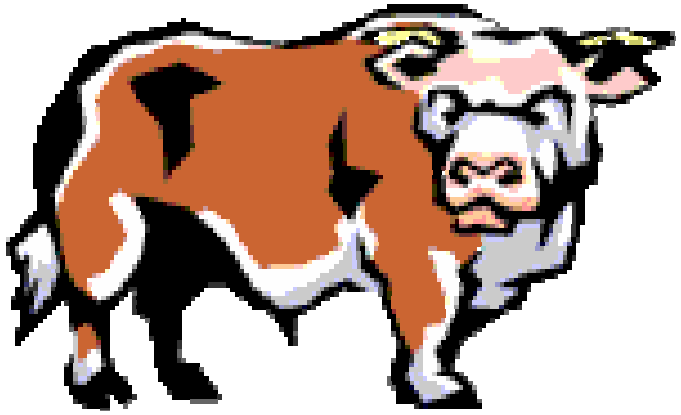
Katedra aplikovanej informatiky, FMFI UK, Bratislava

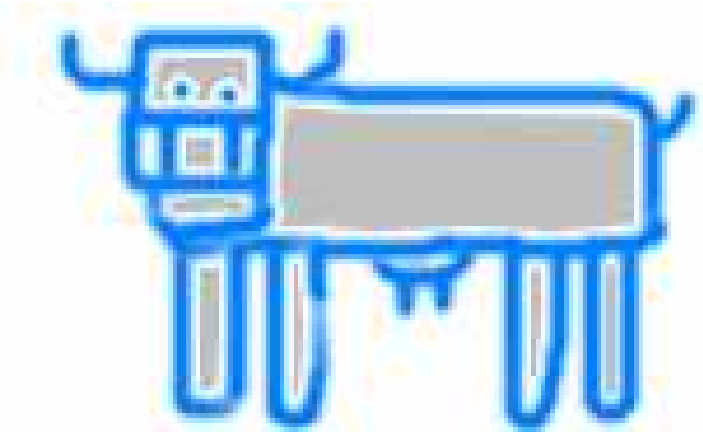
lucny@fmph.uniba.sk

www.microstep-mis.com/~andy

Biomimetický prístup

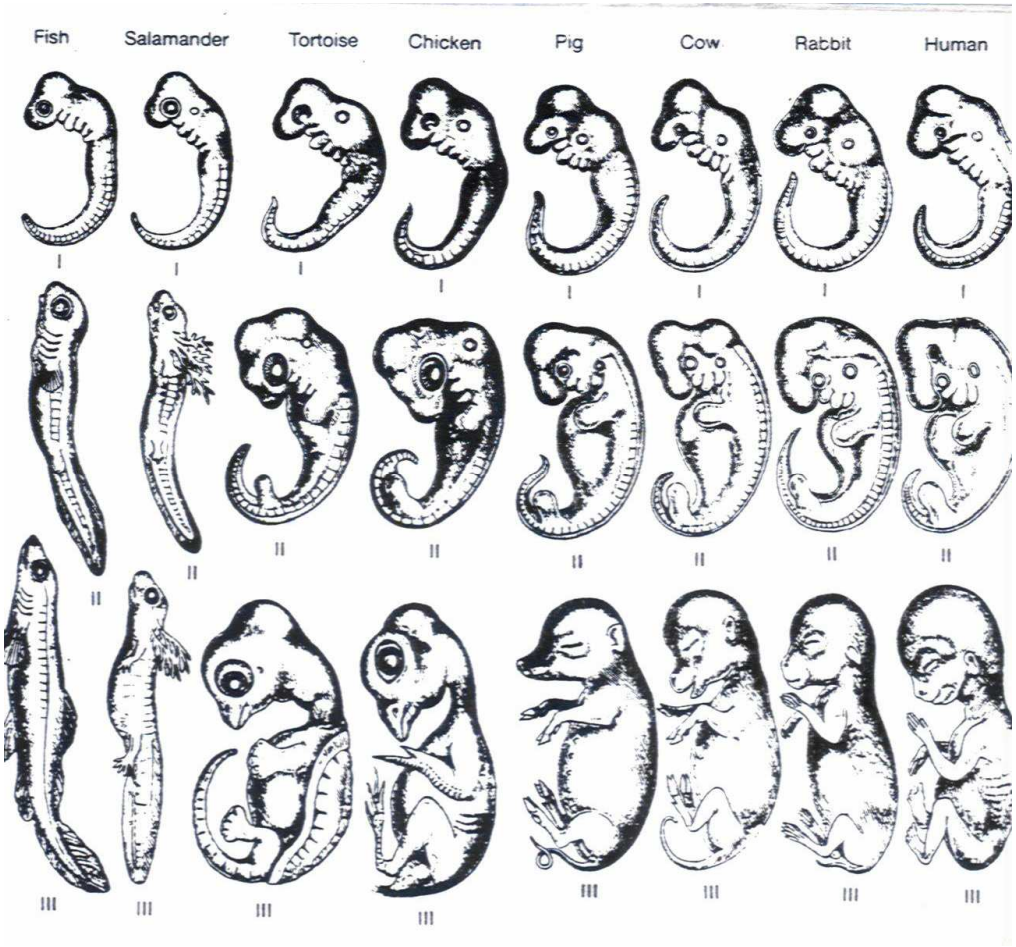
- Napodobňovanie toto čo vytvorila príroda v technickej praxi





Hlavná inšpirácia z biológie

- Štruktúra živých systémov je výsledkom Darwinovskej evolúcie
- Táto štruktúra odráža evolučné kroky, ktoré k nej viedli



Adaptívne systémy vs. Inkrementálny vývoj

- Biomimetický prístup je obyčajne spájaný s oblasťami ako sú genetické algoritmy, genetické programovanie, kde sa vývoj odohráva na populácii jedincov
- Ukážeme iný prístup ako napodobňovať evolúciu a to ručným spôsobom zaoberajúc sa vnútornou štruktúrou jediného jedinca.

Ďalšia biologická inšpirácia

- Štruktúra živých systémov sa vyznačuje určitými typickými črtami: v prvom rade paralelizmom a hierarchiou
- Hierarchia je pritom založená skôr na regulácii, než zktivácii

Keď oddelíme úhorovi mozog od miechy, neprestane plávať svojim sinusoidnými pohybmi, tieto pohyby sa iba stanú perfektne pravidelnými a neustálymi. Mozog teda skôr inhibuje a reguluje miechu

Prečo biomimetický prístup ?

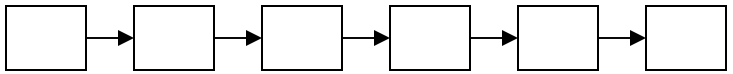
- Otázka: Kedy je inšpirácia z prirody nevyhnutná ?
- Odpoveď: Keď je zamýšľané správanie systému skutočne komplexné

príklad: autonómne vozidlo

Problém škálovateľnosti

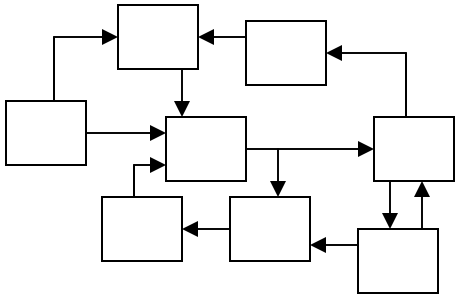
Pre mnoho užitočných priemyselných aplikácii je dostatočné riešiť určitý špecifický problém

V tomto prípad je zvyčajne dostatočnou architektúrou pipeline



Problém škálovateľnosti

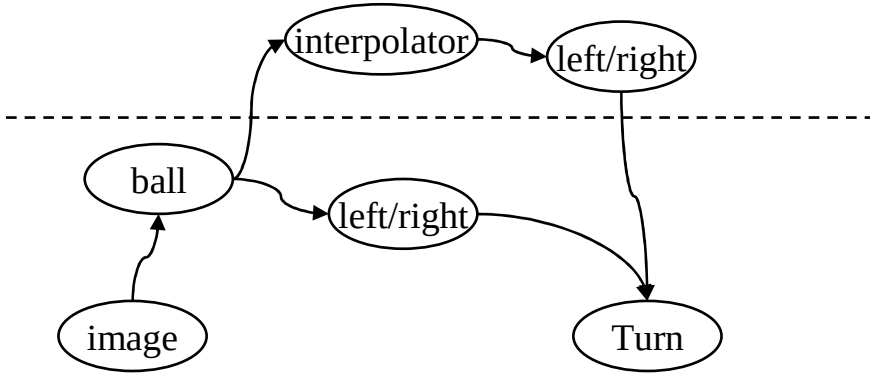
Avšak pri iných úlohách, najmä v oblasti mobilnej robotiky, potrebujeme komplexnejšie správanie a rafinovanejšiu architektúru



Robot COG (AiLab MIT, 1993)

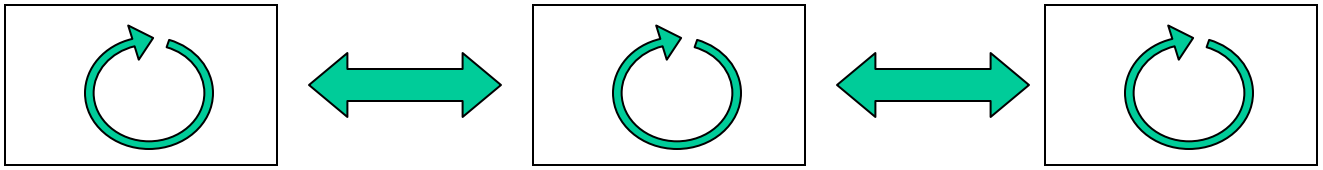
Minského prístup k architektúre

- Systém obsahuje mnoho paralelných modulov (agents, resources)
- Komplexné globálne správanie povstáva z jednoduchého lokálneho správania jednotlivých modulov
- Riadenie = aktivácia správnej množiny modulov v správnom okamihu



Potreba vhodnej architektúry

- Máme veľa paralelných a nepretržite bežiacich modulov
- Máme mechanizmus na výmenu dát medzi nimi
- Pod aktiváciou modulu rozumieme, že modul vyšle dáta významne ovplyvňujúce iné moduly
- Ako môžeme zabezpečiť správnu aktiváciu modulov?



- Jedno možné riešenie: subsumpcia

Subsumpcia

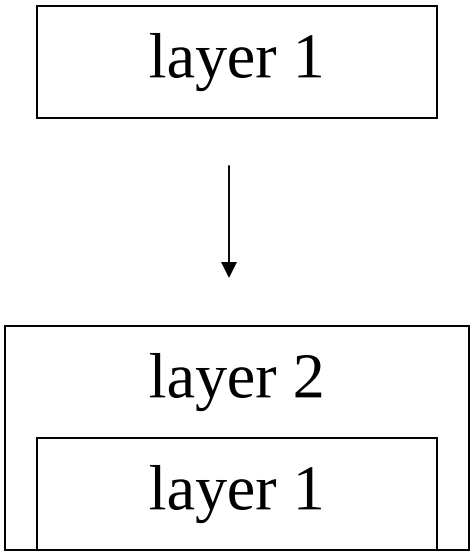
- Vývojová metóda pre umelé systémy s komplexným správaním
- Navrhol ju R. Brooks v roku 1986

- Napodobňuje zjednodušenú biologickú evolúciu

Subsumpcia

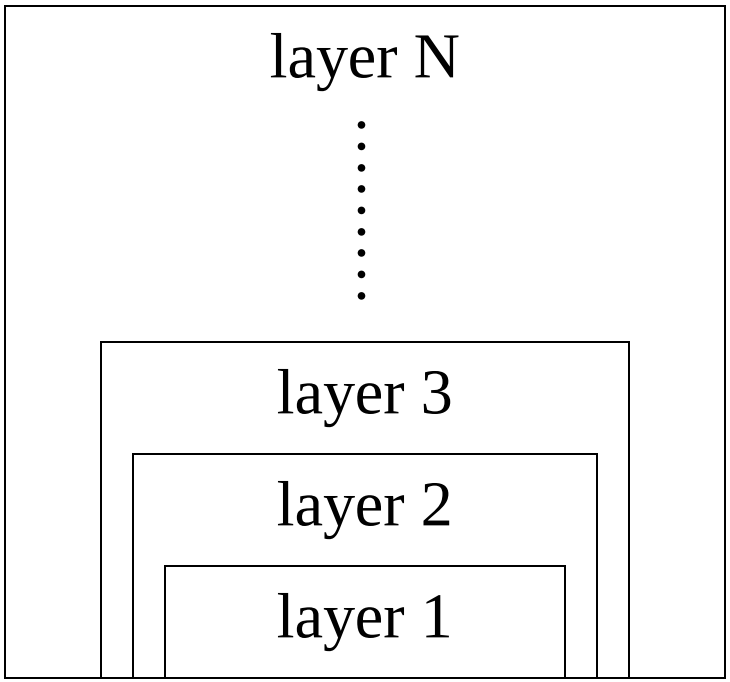
Je založená na evolučnom fakte, že komplexné riadenie pozorované v súčasnosti, má vždy pôvod v jednoduchších predkoch

Vzťah medzi predkom a potomkom je tu však zjednodušený a to tak, že sa predpokladá, že potomok obsahuje presne ten istý riadiaci mechanizmus ako jeho predok, iba k nemu ešte niečo navyše pridáva



Zjednodušenie evolúcie

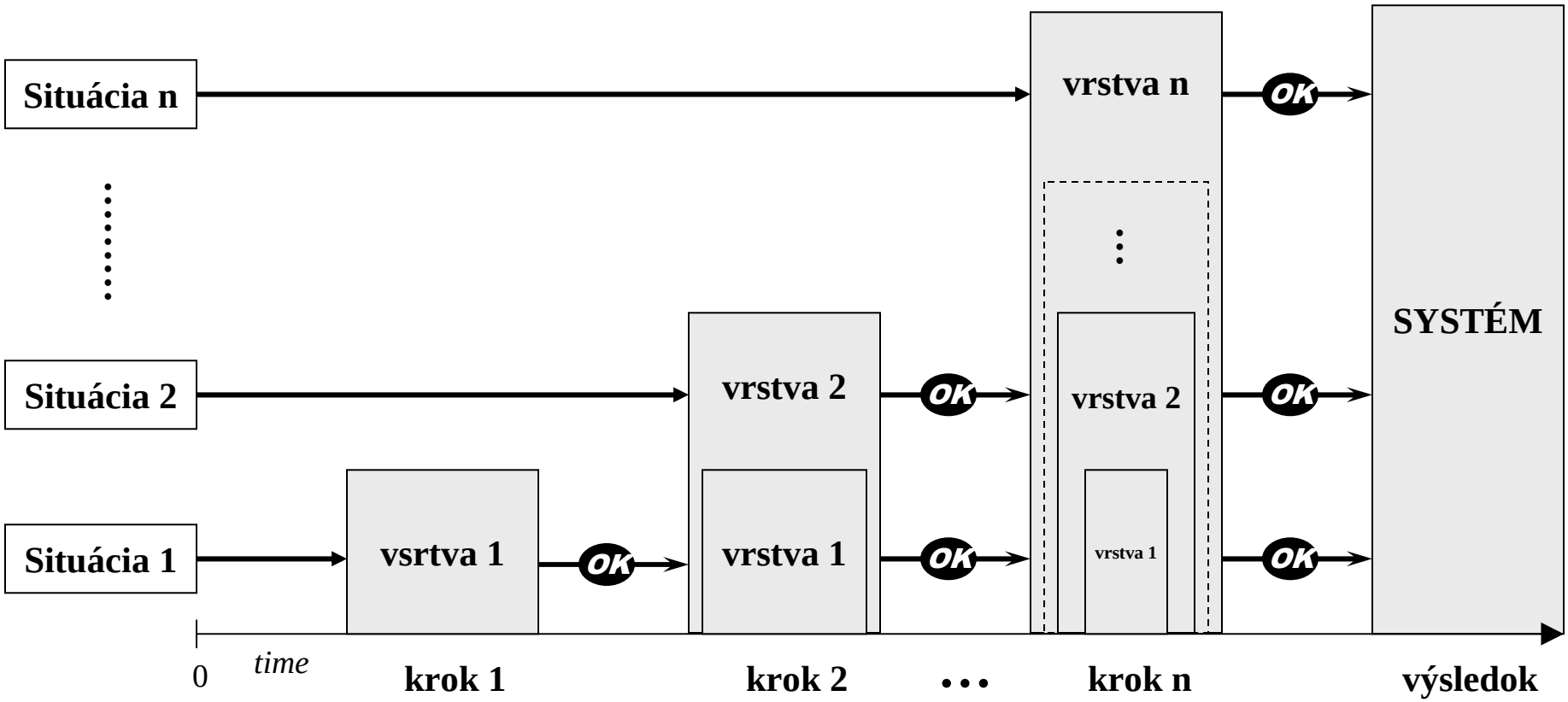
- T.j. mechanizmus potomka zahrňuje (angl. *subsume*) kompletný mechanizmus jeho predkov
- Na základe toho sa tento princíp volá *subsumption*.



Vývoj pomocou subsumpcie

- Najprv navrhneme vhodné a hlavne dostatočné senzory a aktuátory
- Potom si predstavíme postupnosť evolučných krokov, ktoré by mohli viesť k želanému riadeniu a ktoré začínajú z jednoduchého základu
- Potom postupne vyvinieme štruktúry riadenia zodpovedajúce týmto krokom, pričom jednoduchšiu predchádzajúcu verziu obohacujeme pridaním novej vrstvy
- Od pridaných vrstiev očakávame, že poskytnú novú funkcionalitu, ale nepoškodia tú ktorá už je implementovaná

Vývoj pomocou subsumpcie



Situovanosť

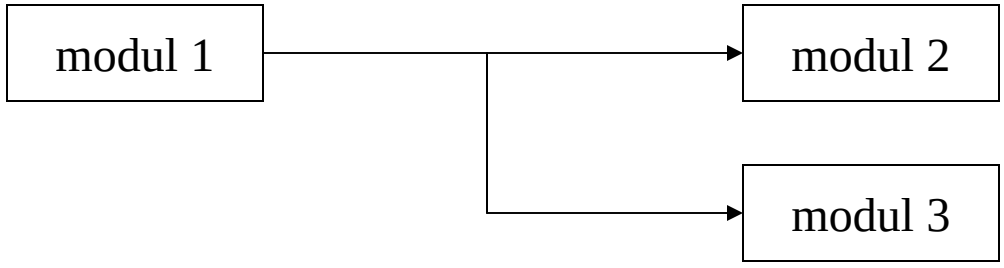
- Postupnosť evolučných krokov môže byť navrhnutá tak, aby riadenie dosiahnuté v určitom kroku korešpondovalo so želaným výsledným riadením za zjednodušených podmienok.
- Jednoduchšia situácia je potom zvládaná evolučne staršími vrstvami.
- Naopak čím zložitejšiu situáciu máme, tým novšie a novšie vrstvy sa do riadenia zapájajú

Potreba vhodnej modularity

- Avšak, ako môžu evolučne novšie vrstvy vplývať na vrstvy evolučne staršie ?
- Evolučne staršie vrstvy boli predsa navrhnuté špecifický účel a nemajú žiadne rozhrania pre neskorší vývoj !
- Riešenie: vrstvy musia mať vhodnú modulárnu štruktúru, ktorá novším vrstvám ich ovplyvňovanie umožní

Subsumpčná architektúra

- Vrstva je zložená s jednoduchých modulov
- Tieto moduly vzájomne komunikujú posielaním správ po tzv. vedeniach



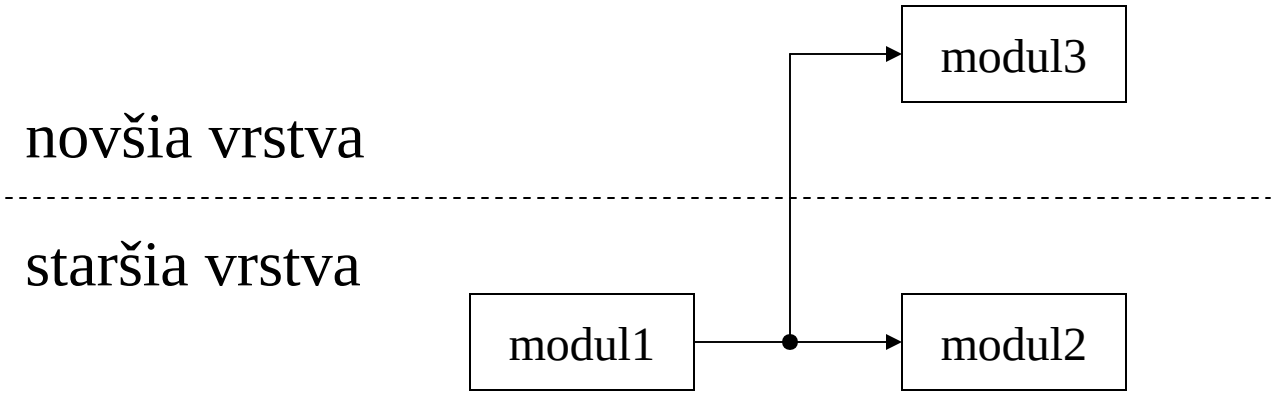
Mechanizmy ovplyvňovania

Subsumpčná architektúra predpokladá tri mechanizmy, ktorými môžu evolučne novšie vrstvy ovplyvňovať vrstvy evolučne staršie:

- odpočúvanie
- inhibícia
- supresia

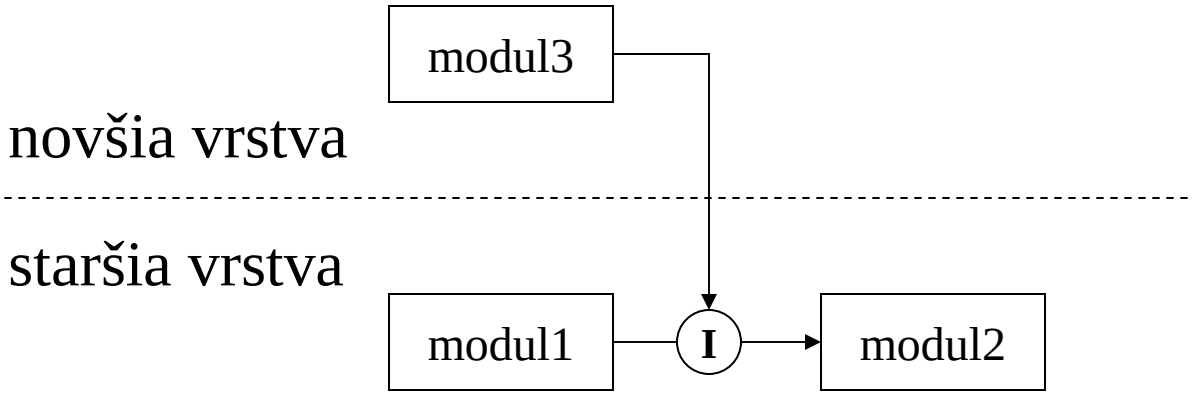
Odpočúvanie

- Novšia vrstva môže odpočúvať vedenie, ktorým si moduly v staršej vrstve posielajú správy



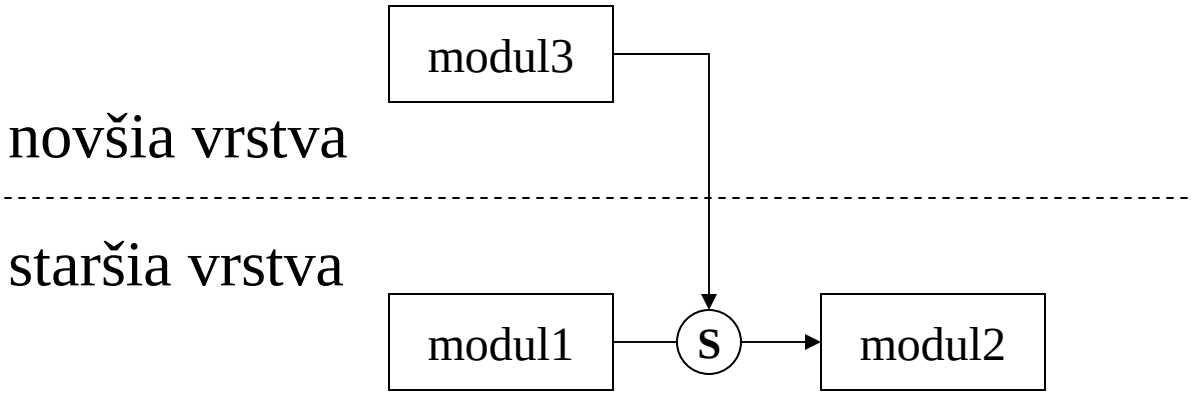
Inhibícia

- Novšia vrstva môže prerušiť komunikáciu na vedení medzi modulmi v staršej vrstve

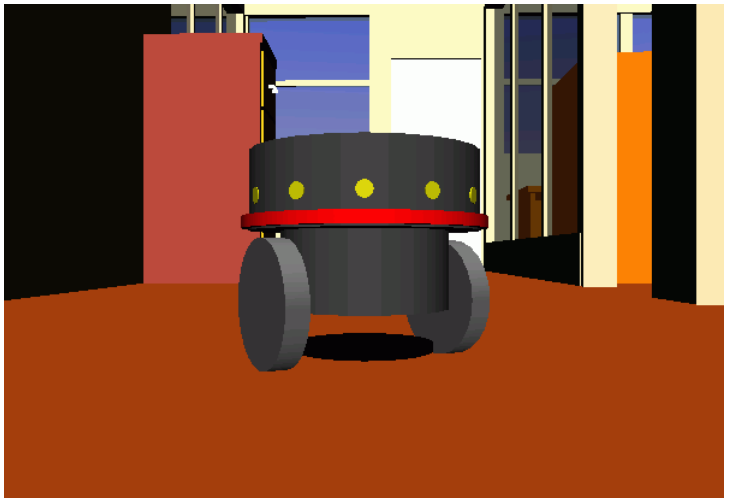


Supresia

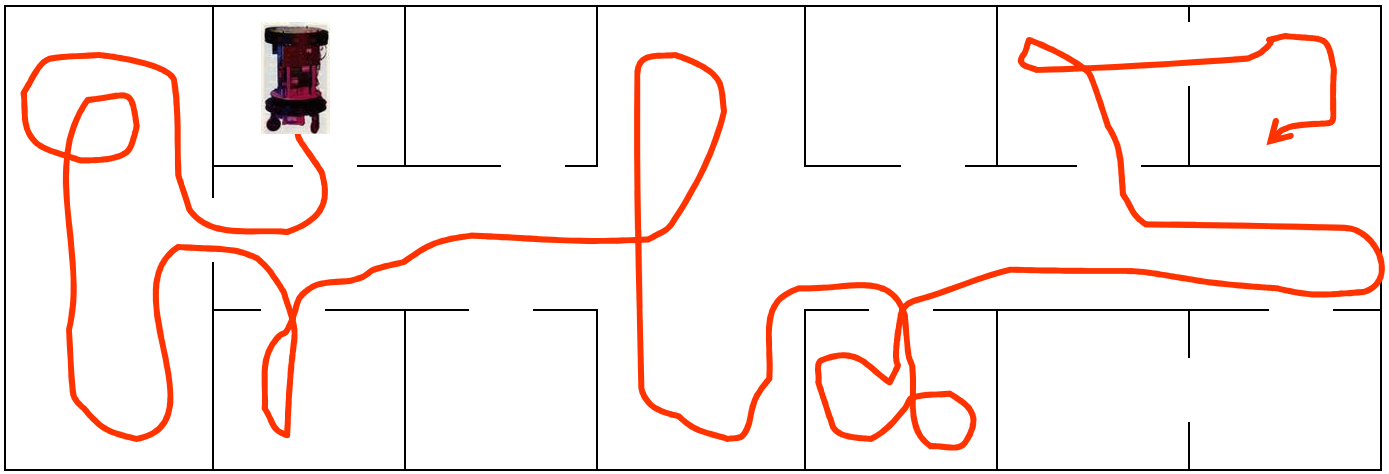
- Novšia vrstva môže na vedenie medzi modulmi staršej vrstvy vyslať náhradnú správu



Príklad

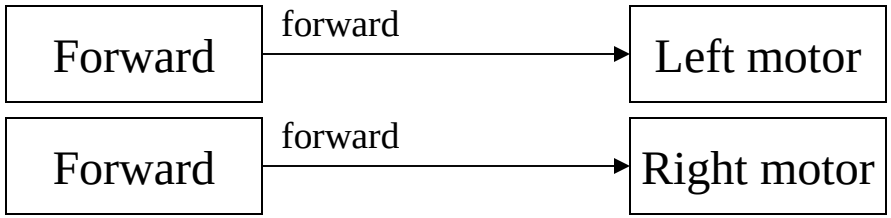


Mobilný robot pohybujúci sa v kancelárskych priestoroch (mod. reimplementácia robota ALLEN, Brooks 1986)



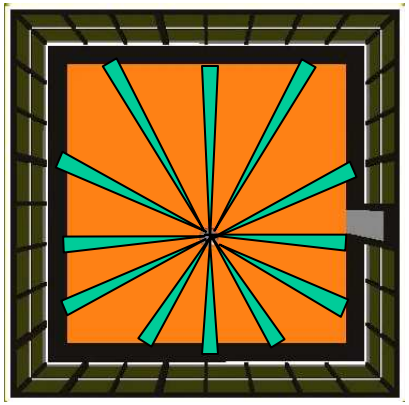
Príklad – krok 1

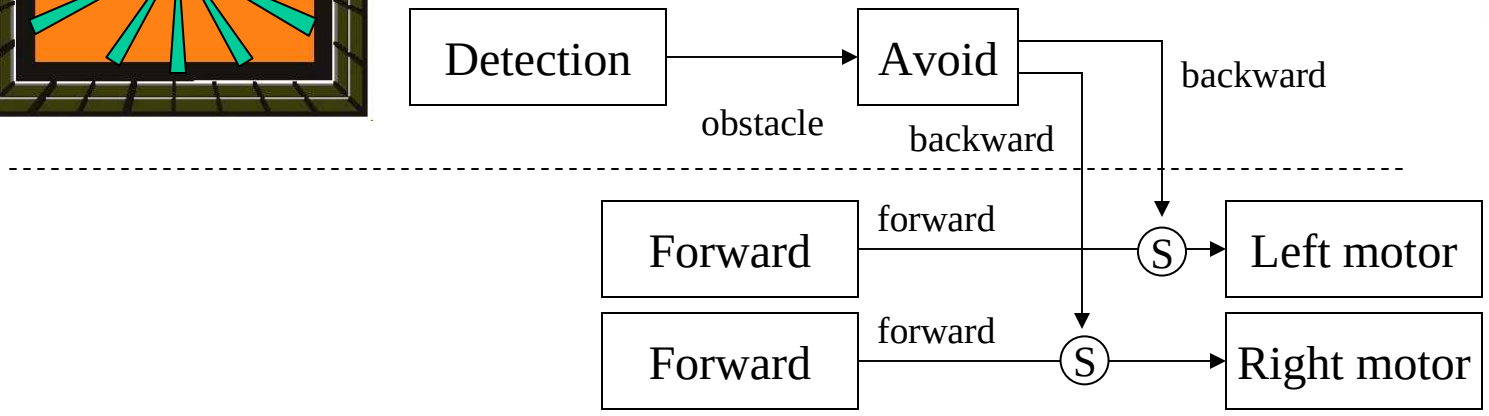
- *Začneme s robotom ktorý iba ide dopredu*



Príklad – krok 2

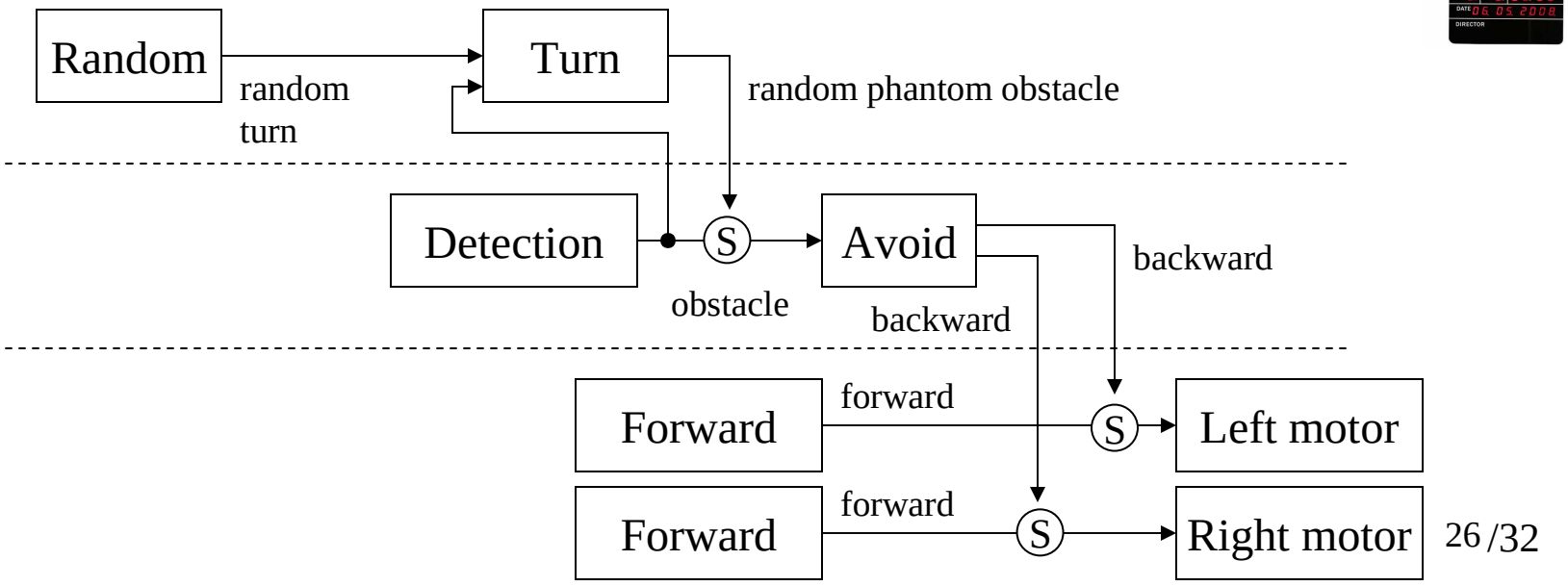
- *Pridáme vrstvu, ktorá rozpoznáva prekážky a pokiaľ sú prítomné v smere pohybu, signál vpred je na vhodnejšom kolese nahradený signálom vzad. Následkom toho robot nenaráža na prekážky.*





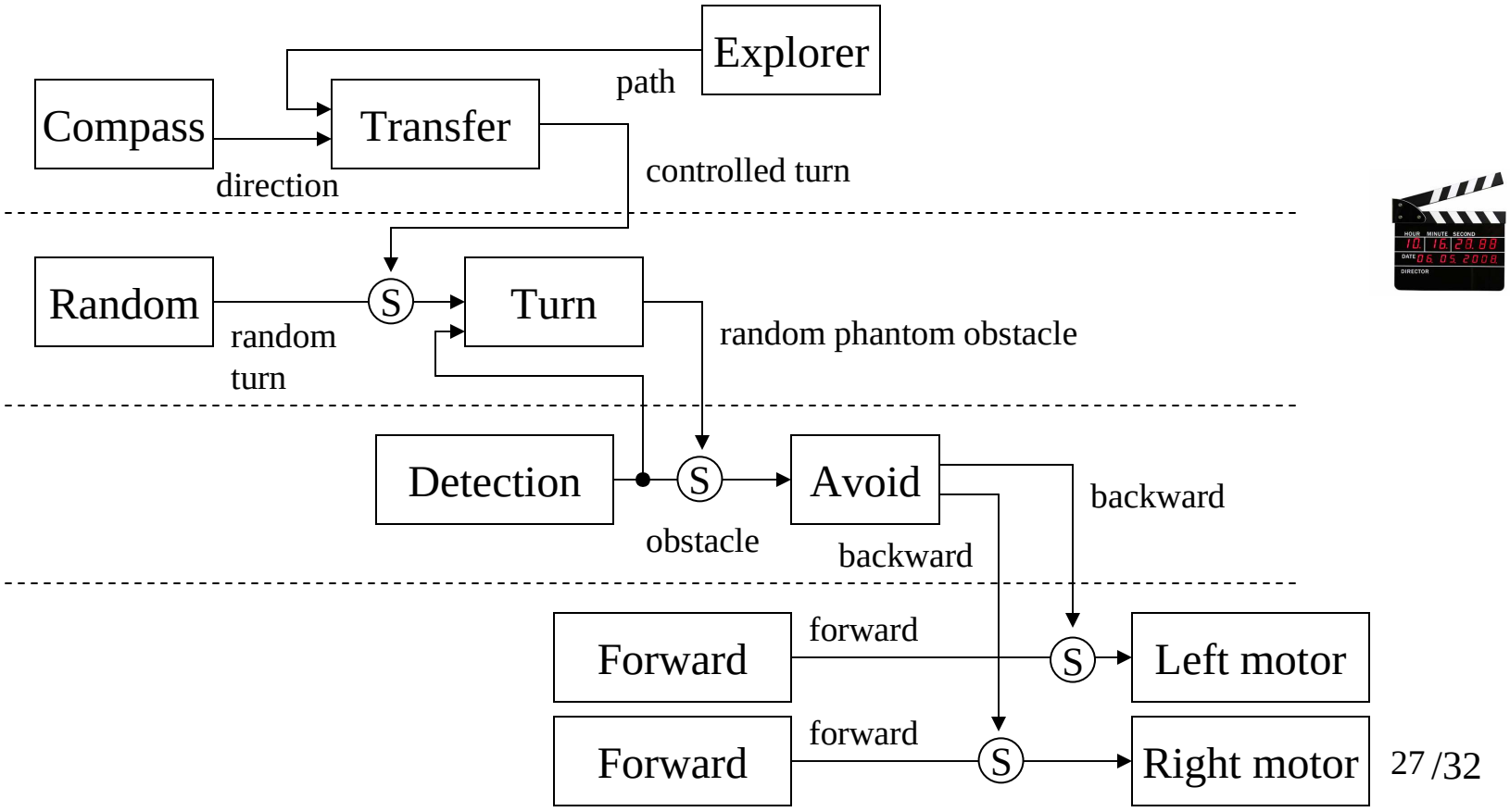
Príklad – krok 3

- *Pohyb takéhoto robota sa ľahko dostane do cyklu. Preto pridáme vrstvu ktorá sem-tam náhodne zmení smer robota. Túto zmenu smeru implementujeme dosť netradične: ako reakciu na fantomickú prekážku.*



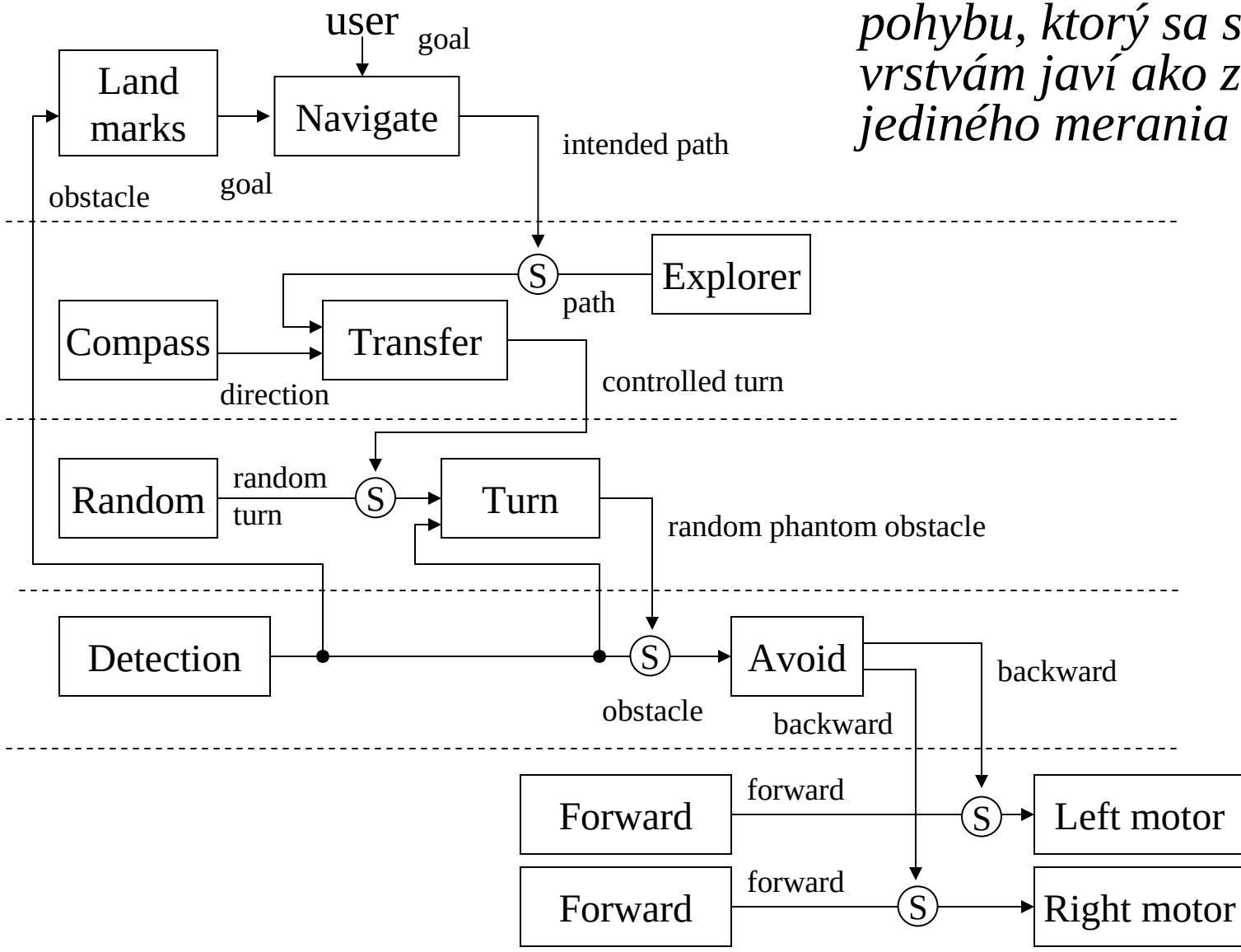
Príklad – krok 4

- *Ďalšia vrstva poskytuje globálnejší pohyb v absolútnom smere – z jednej časti priestoru do inej. Po zvolení vhodného smeru, implementujeme vhodnú korekciu pohybu presnými otočeniami, ktoré sa starším vrstvám javia ako čisto náhodné*



Príklad – krok 5

- *Ďalšia vrstva monitoruje prostredie a môže cielene voliť vhodnejší smer pohybu, ktorý sa starším vrstvám javí ako zistený z jediného merania*



Deriváty subsumpčnej architektúry

• *behaviorálne architektúry*: iba supresia na výstupoch z vrstiev (vrstva implementovaná jediným modulom)

• *jemnozrnná architektúra*: jednotný typ dát, dátová fúzia, rozbitie vrstvy na moduly podobné neurónom

• mnoho ďalších

Derivát implementovaný na FMFI UK

- dlhodobá tradícia vďaka J. Kelemenovi (1992 pôsobil u Marvina Minského na MIT)

- *architektúra agent-space*: moderné rozšírenie subsumpčnej architektúry ktoré prekonáva obmedzenia dané napodobňovaním usporiadania hardwaru (software umožňuje viac) (A. Lúčny 2004)

Záver

- Subsupčná architektúra je biologicky inšpirovaná metóda vývoja systémov s komplexným správaním
- Typické vlastnosti: inkrementálny vývoj. situovanosť, decentralizácia, ovplyvňovanie inhibíciou a supresiou
- Literatúra: Brooks, R.: Cambrian Intelligence, MIT Press, Cambridge, 1999

Vďaka za pozornosť

Andrej Lúčny

Katedra aplikovanej informatiky,
FMFI UK, Bratislava
lucny@fmph.uniba.sk
www.microstep-mis.com/~andy