

Riadenie simulovaného robota iCubSim pomocou architektúry Agent-Space

Andrej Lúčny – Matúš Kopernický

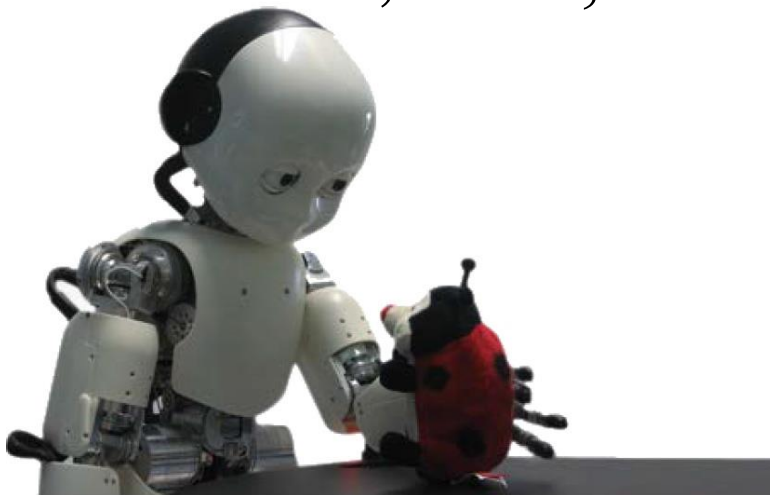
Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

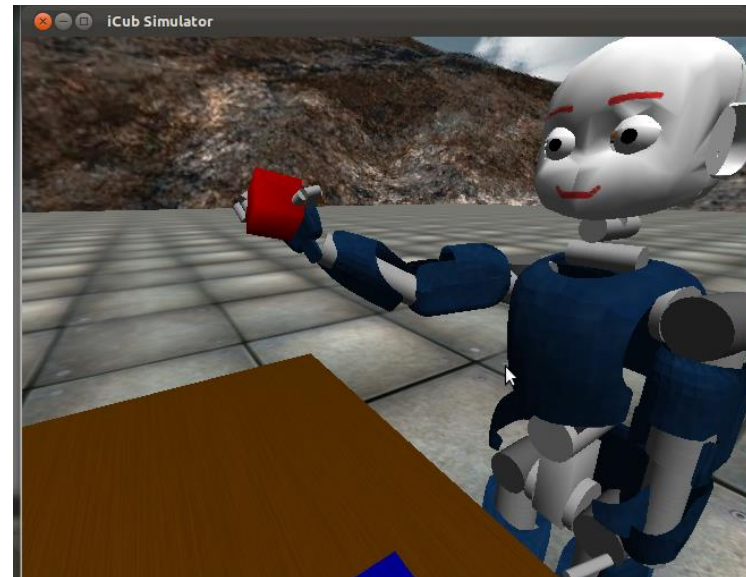
<http://www.agentspace.org/andy>

iCub a iCubSim

- EU IP RoboCub & NoE EuCognition I-III
- Cieľ: overiteľnosť výsledkov výskumu na humanoidných robotoch
- 53 DOF, koža, oči

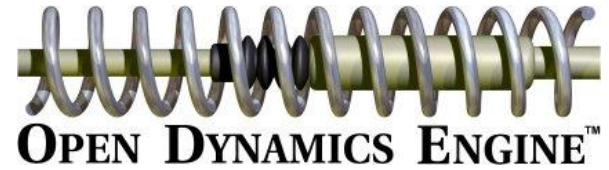


Jager - Meijering 2015

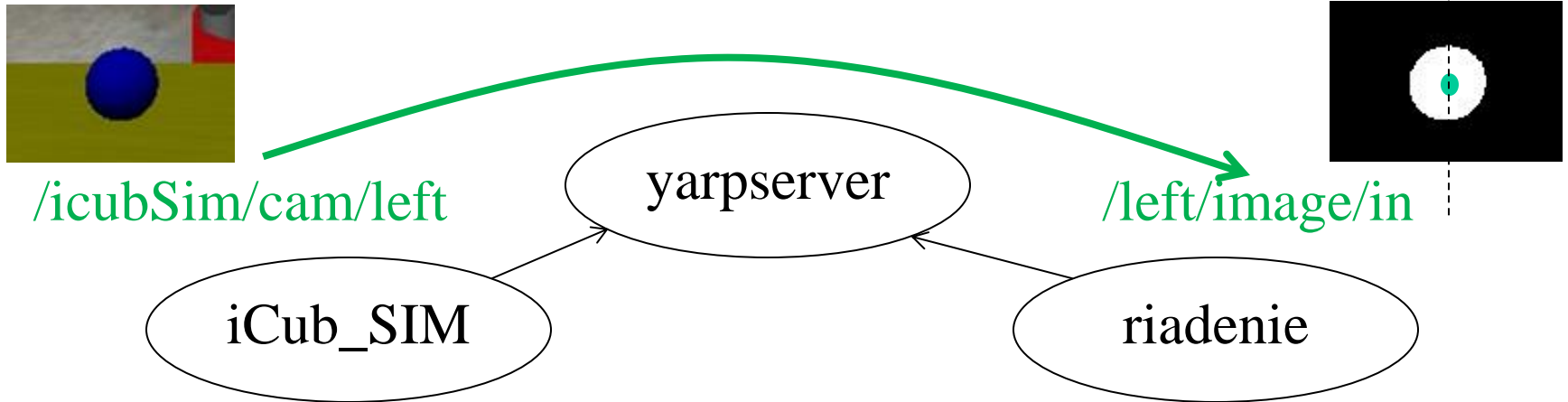


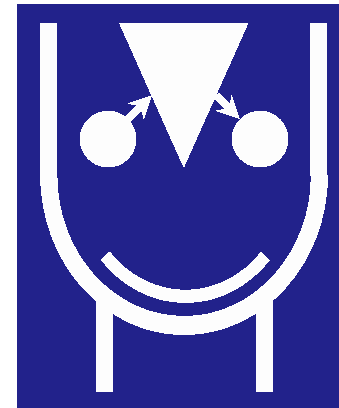
Zdechovan, 2012

iCubSim



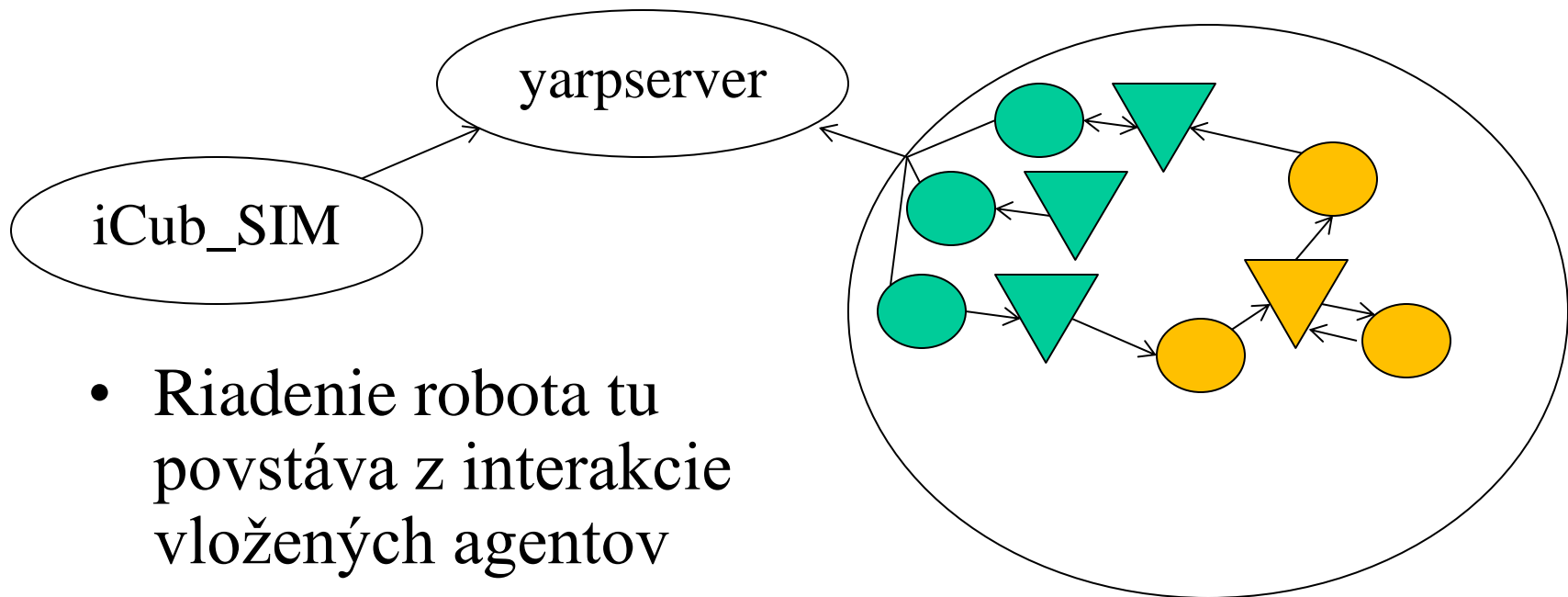
- Simulátor napísaný v C++ používajúci ODE a OpenGL, primárne na platforme Linux
- Na zriadenie toku dát medzi riadiacim programom a simulátorom používa nadstavbu TCP – YARP
- YARP umožňuje definovať na každej strane svoje porty a zrealizovať medzi nimi tok dát.





Rozšírenie o Agent-Space

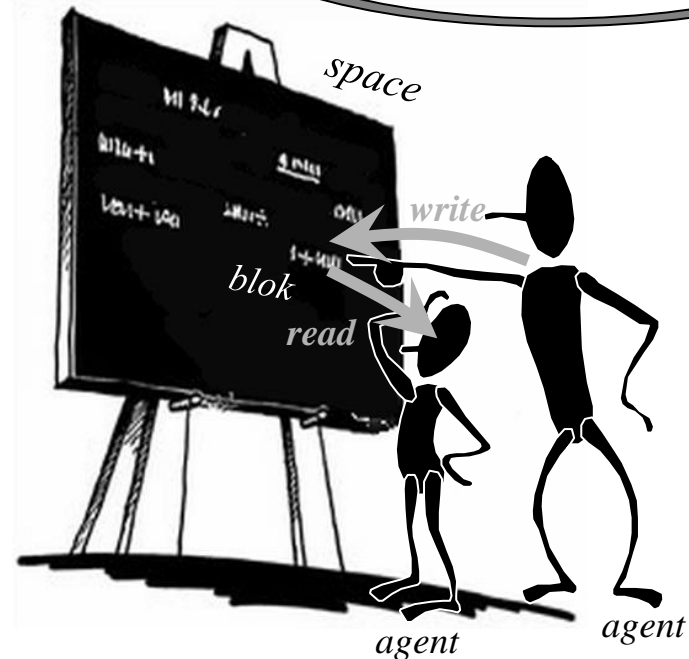
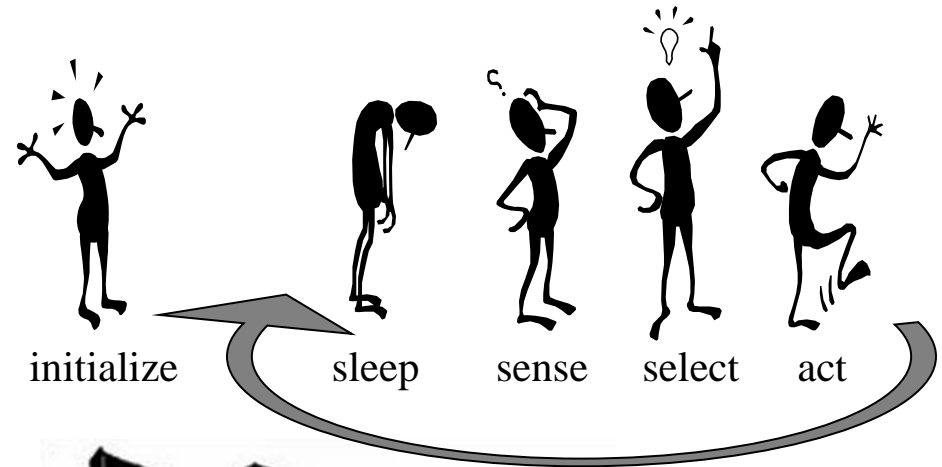
- Riadiaca aplikácia v ktorej sú senzory a akturátory iCubSim dostupné vo forme blokov na čiernej tabuli a do ktorej možno vkladať agentov, ktorí s nimi môžu manipulovať



- Riadenie robota tu povstáva z interakcie vložených agentov

Architektúra Agent-Space

- Systém sa skladá z agentov
- Agenti medzi sebou komunikujú nepriamo cez Space (čiernu tabuľu)

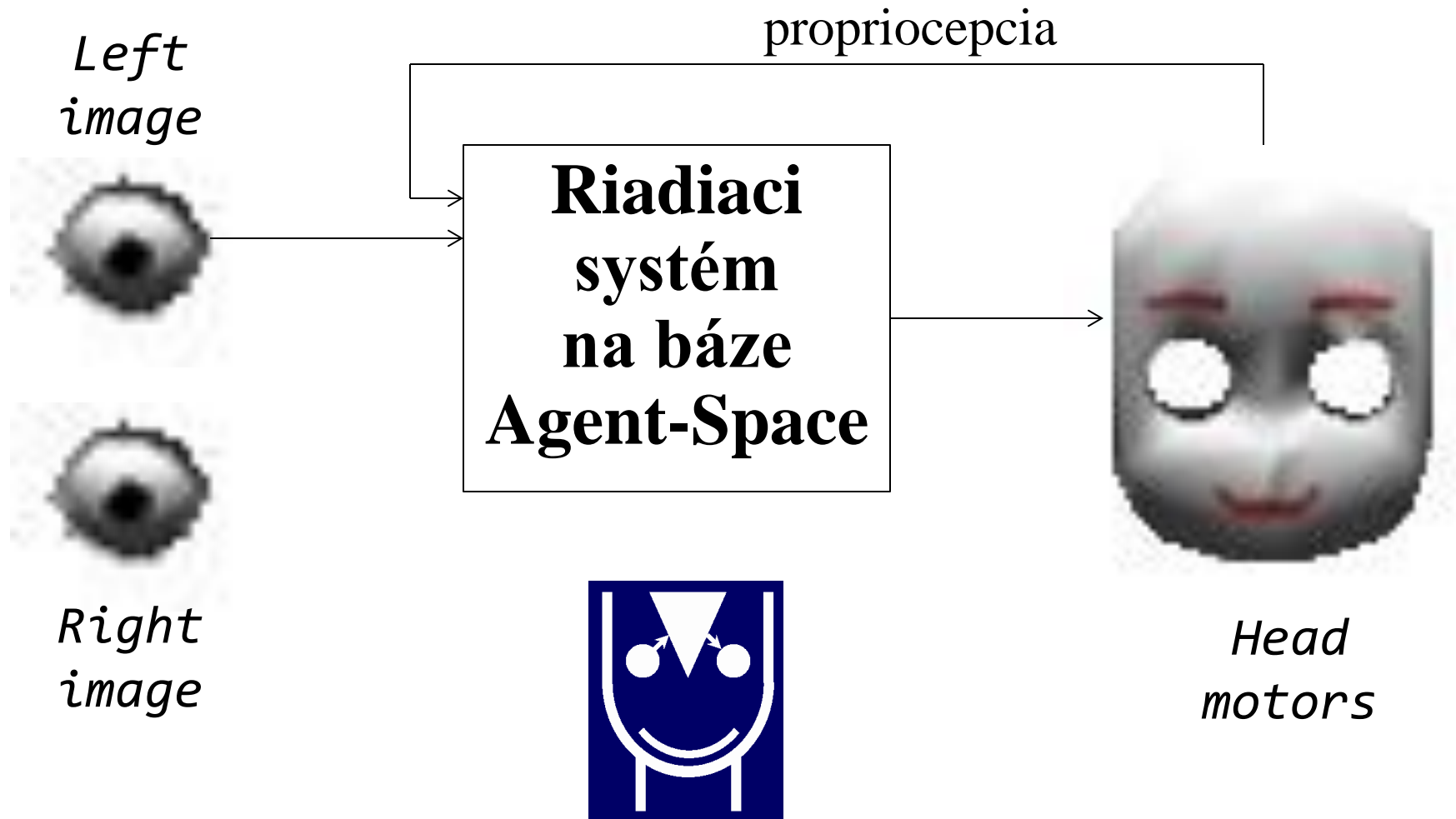


Modelová úloha

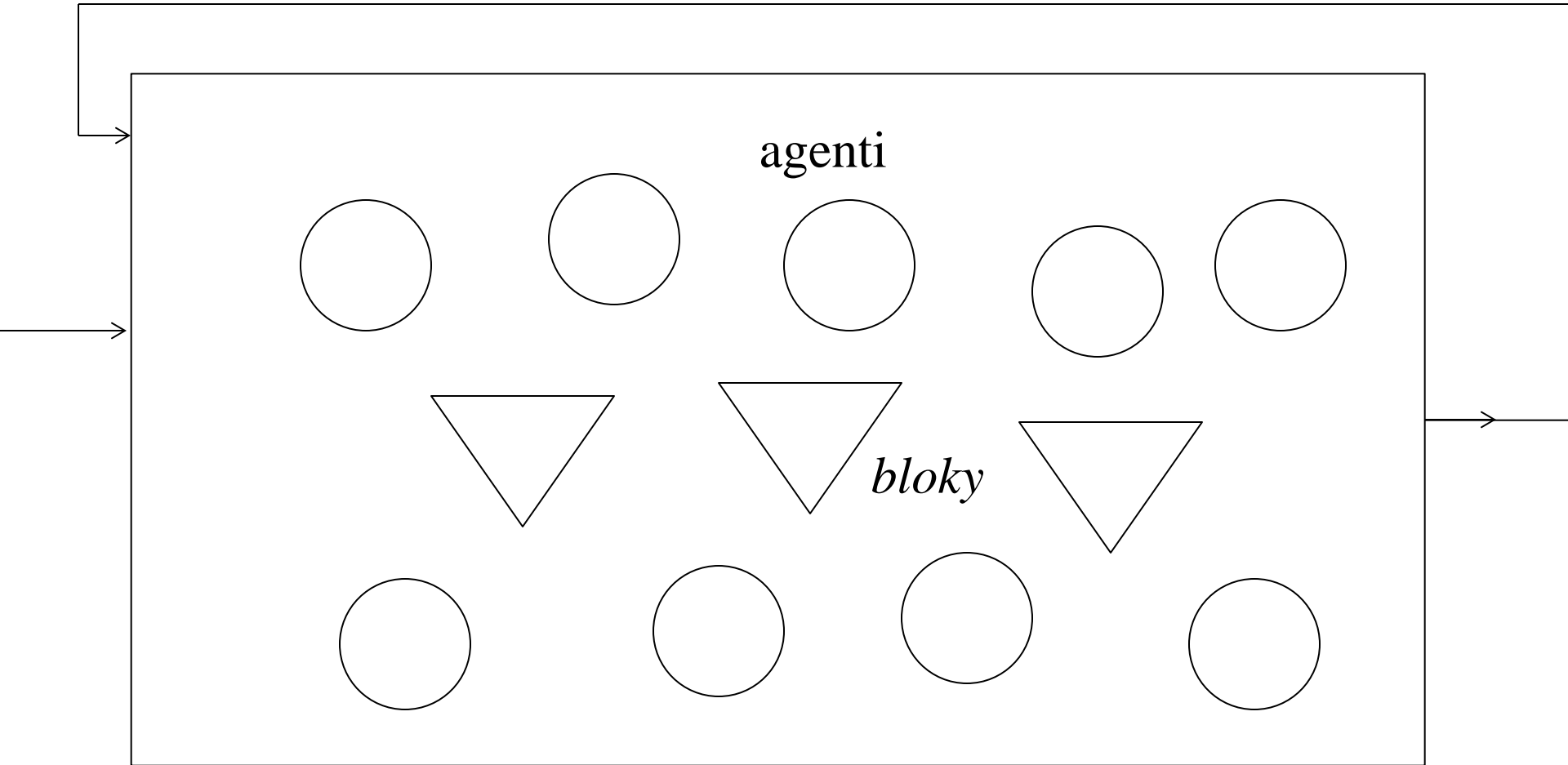
- Hľadanie modrej loptičky v priestore.
- Upriamenie pozornosti.
- Udržanie loptičky v zornom poli



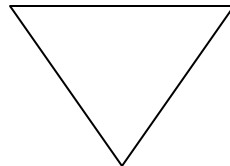
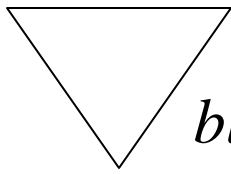
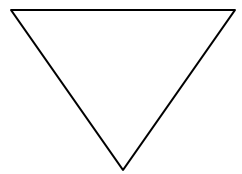
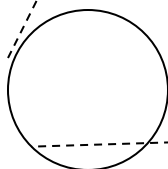
Použitie architektúry Agent-Space



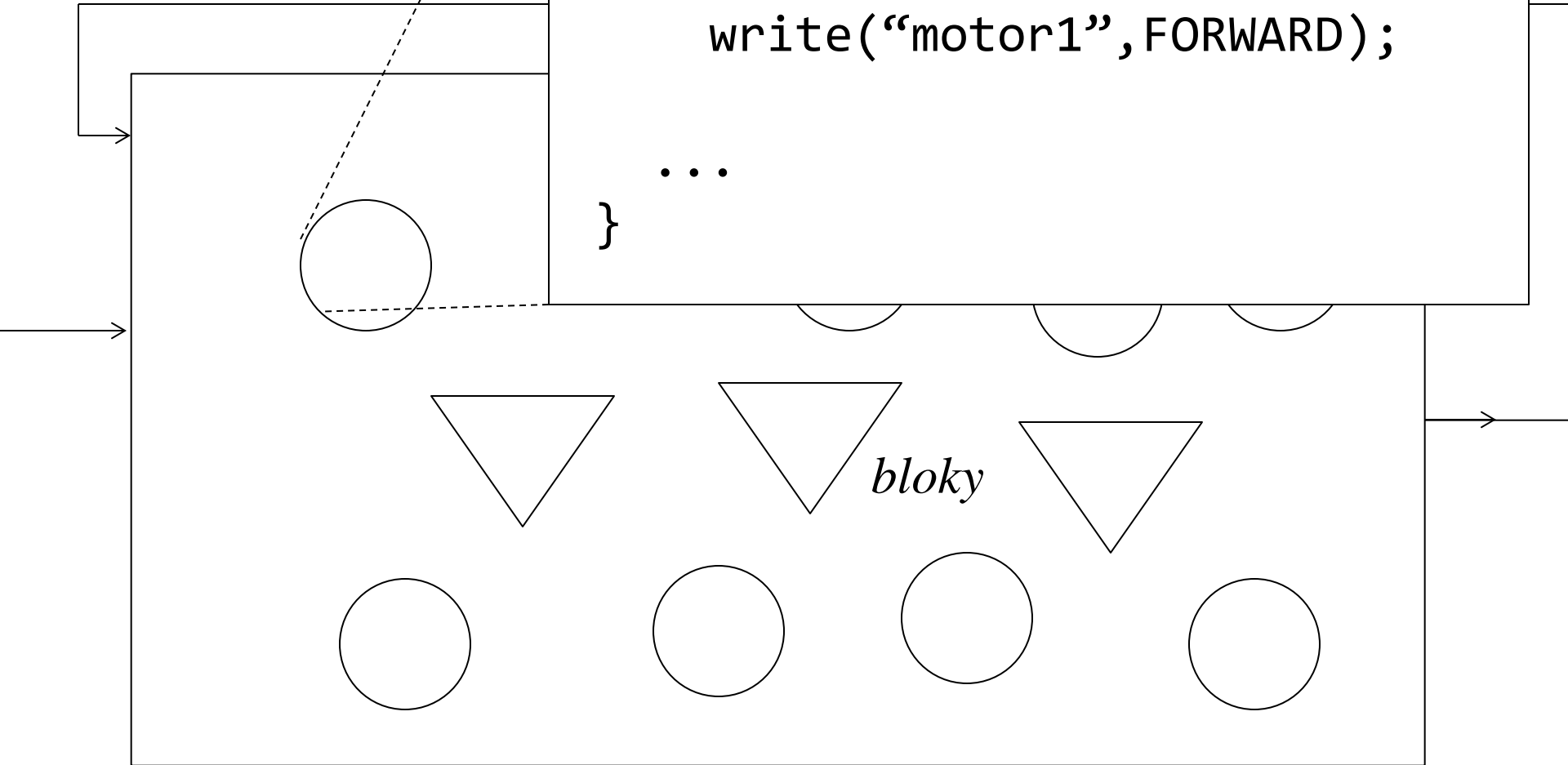
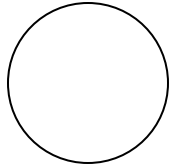
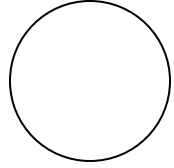
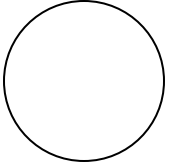
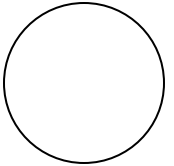
Riadiaci systém na báze Agent-Space




```
...  
for (;;) {  
    if (!Ball.see(read("image"))  
        write("motor1", FORWARD);  
    ...  
}
```

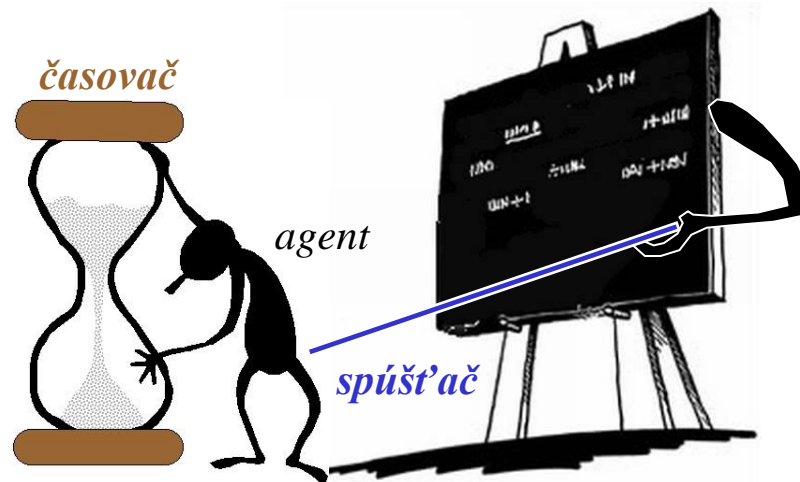


bloky



Implementácia v C++

- Multivláknové prostredie pthreads
- Každý agent má vlastné vlákno
- Može volať metódy singletonu Space
- vlákno agenta je blokované na časovač alebo spúšť'ač



Príklad použitia

```
#include <iostream>
#include <conio.h>
#include "agentspace.h"
using namespace std;

class MyAgent1 : public Agent {
private:
    int i;
protected:

    void init (string args) {
        i = 0;
        timer_attach(1000,1000);
    }

    void sense_select_act (int pid) {
        i++;
        cout << "a := " << i << endl;
        space_write("a",i,1500);
    }

public:
    MyAgent1 (string args) :
        Agent(args) {};
};
```

```
class MyAgent2 : public Agent {
protected:

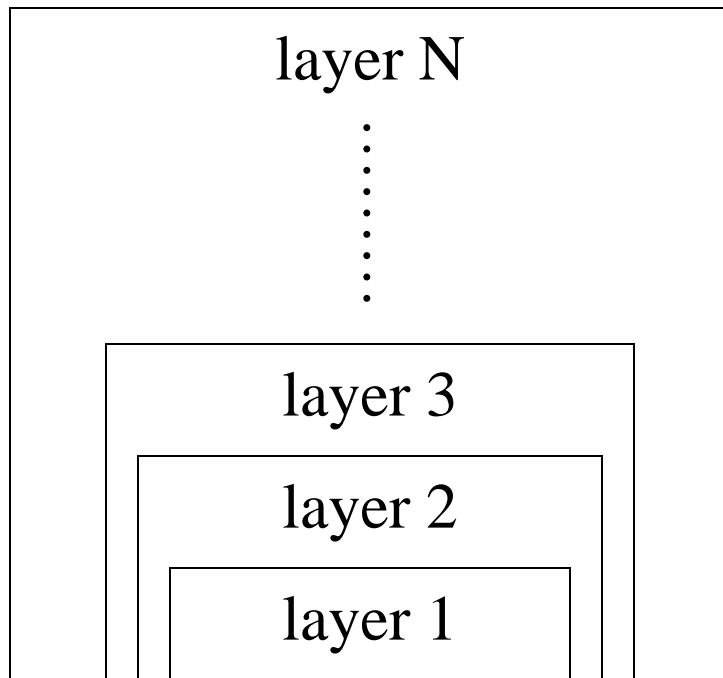
    void init (string args) {
        trigger_attach("*",TRIGGER_MATCHING);
    }

    void sense_select_act (int pid) {
        int a = space_read("a",0);
        cout << "a = " << it->value << endl;
    }

public:
    MyAgent2 (string args) :
        Agent(args) {};
};

int main () {
    MyAgent1 a1("");
    MyAgent2 a2("");
    getch();
}
```

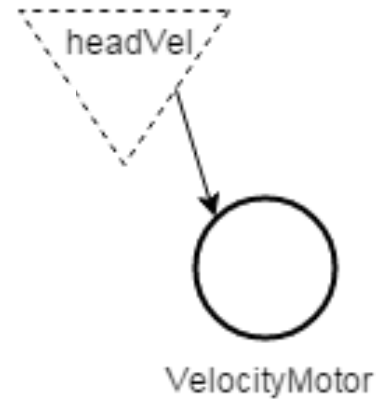
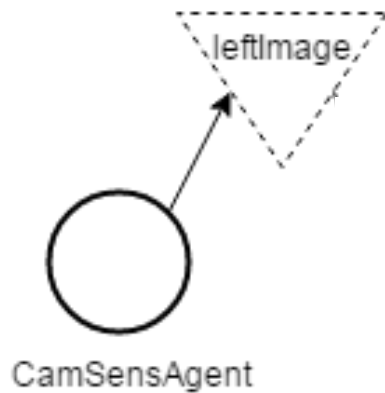
Vývojová stratégia: subsumpcia



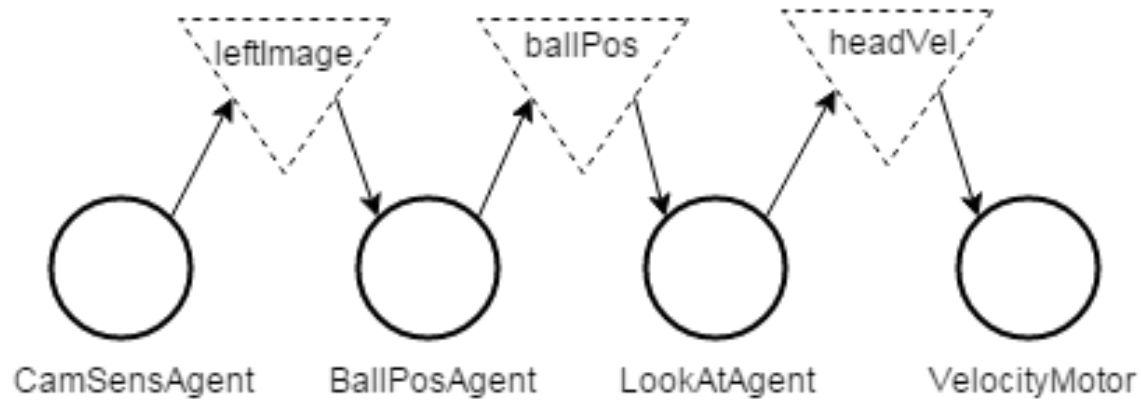
- „evolučne staršie“
vrstvy sa už
nemenia
- „evolučne novšie“
vrstvy ovplyvňujú
„evolučne staršie“
vrstvy

0. etapa vývoja – senzory a aktuátory

Prepojenie senzorov a aktuátorov s blokmi v Space

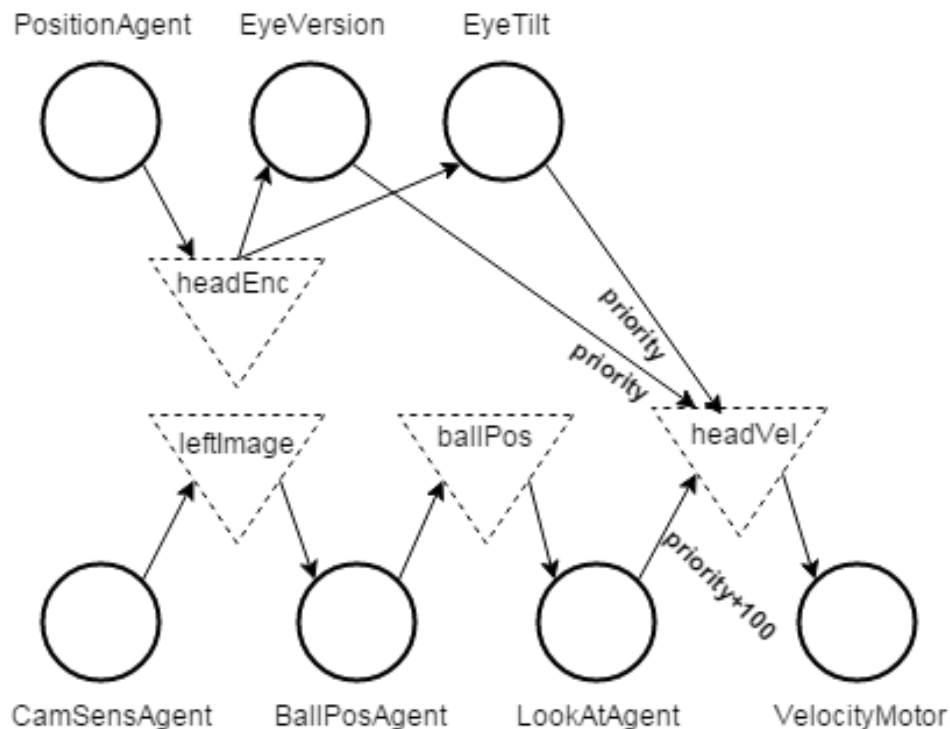


1. etapa vývoja - dátová výmena



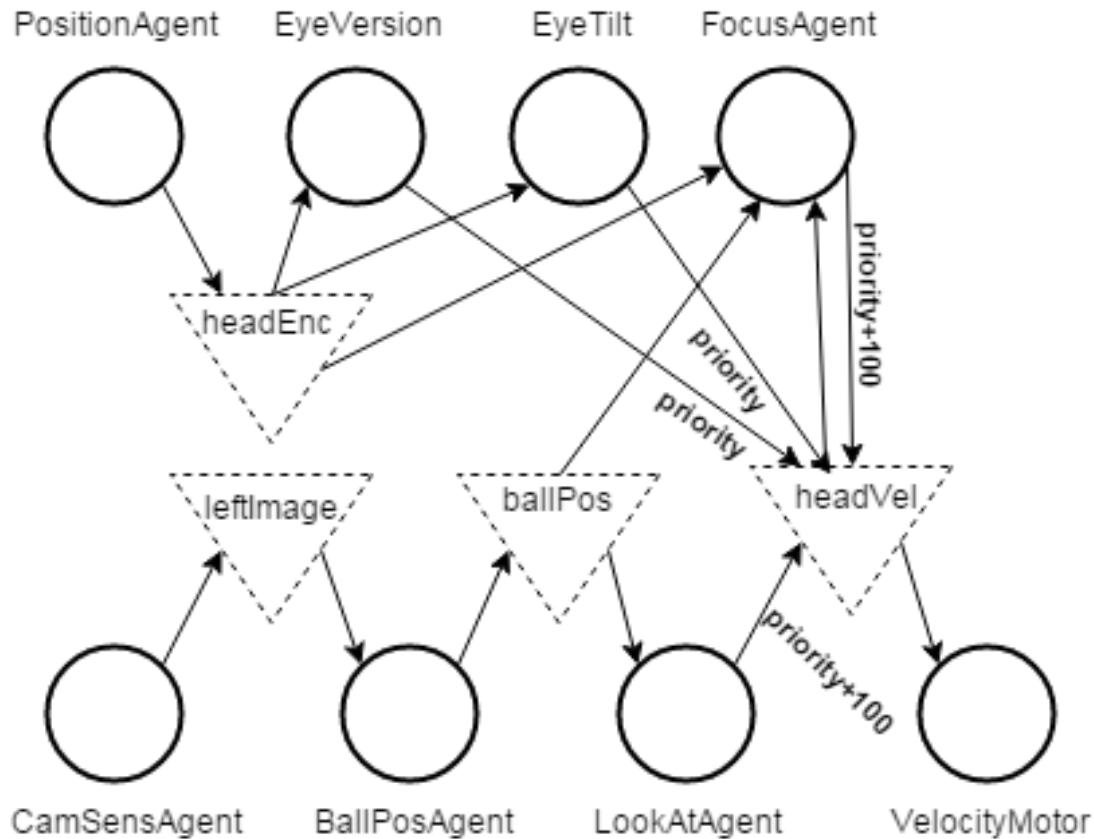
2. etapa vývoja

- Hľadanie loptičky
 - Rozširovanie zorného pola
 - PositionAgent, EyeVersionAgent, EyeTiltAgent



3. etapa vývoja

- Upravenie pohľadu na loptičku - FocusAgent



Princíp Embodiment-u

telo ako výpočtové zariadenie +
senzomotorický prístup

Tradičné riešenie

algebraické výpočty

máme model $x = M(\Phi, \Psi, \Omega)$

hľadáme riešenie $(\Phi_t, \Psi_t, \Omega_t)$ rovnice

$$x_t = \left(\frac{\delta M}{\delta \Phi}(\Phi_0), \frac{\delta M}{\delta \Psi}(\Psi_0), \frac{\delta M}{\delta \Omega}(\Omega_0) \right) (\Phi_t, \Psi_t, \Omega_t)^T$$

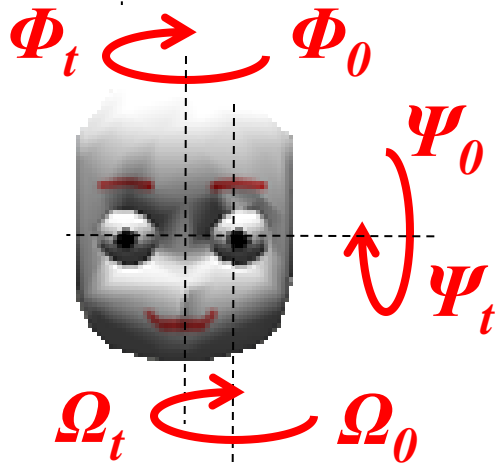
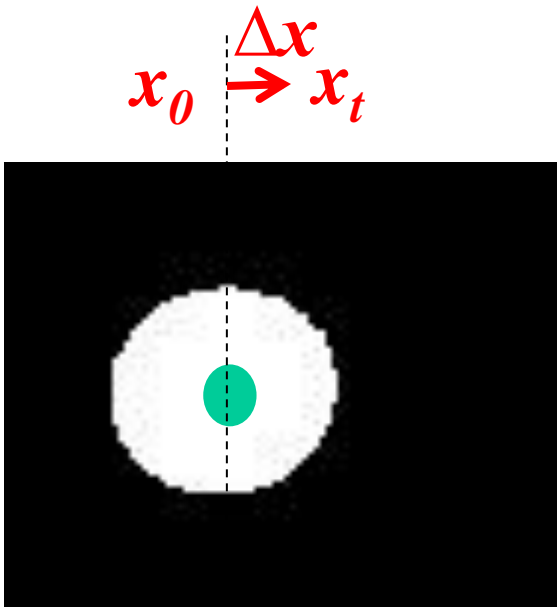
Embodiment

$$\Delta \Phi = c_{\Phi} \Delta x \quad \text{analogicky } \Psi, \Omega$$

alebo

$$\Delta \Phi = \text{random} \quad \text{analogicky } \Psi, \Omega$$

$$\text{aby } |\Delta x| \rightarrow 0$$



Záver

- Rozbehali sme simulátor iCubSim
- Do simulátora iCubSim sme implementovali architektúru Agent-Space
- Prepojili sme senzory a aktuátory iCub-u s architektúrou
- Vyvinuli sme pomocou nej riadenie hlavy iCub-u tak, aby sa pozerala za modrou loptičkou na stole (čo najprirodzenejším spôsobom)
- Vyvíjali sme na princípe subsumcie a embodimentu
- Ďalšie info: www.agentspace.org/icub

Ďakujeme za pozornosť

Riadenie simulovaného robota iCubSim
pomocou architektúry
Agent-Space

Andrej Lúčny – Matúš Kopernický

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

www.agentspace.org/andy