

Hybridní metody výpočetní inteligence a softwaroví agenti

Roman Neruda a spol.

Ústav informatiky AV ČR Praha
roman@cs.cas.cz

Zadov 6.–7. 10. 2006

Outline

- 1 Co?
- 2 Kdo? (a Jak)?
- 3 Kde? (je ten sémantický web?)
- 4 Co dál?

Hybridní metody



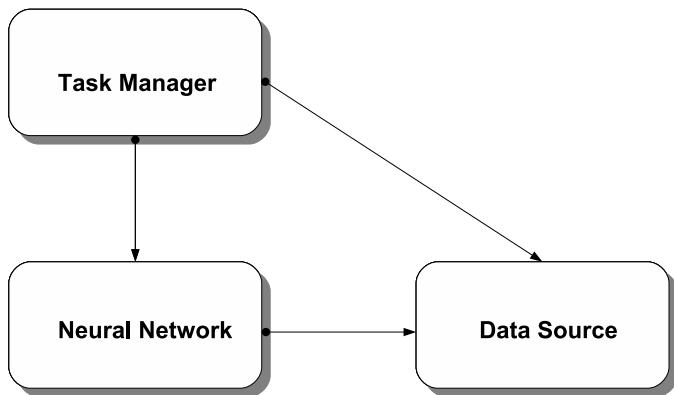
Výpočetní inteligence

- Soft computing (L.Zadeh): “kreativní fúze neuronových sítí, evolučních algoritmů, fuzzy kotrolerů, . . . ”
- výhody oproti individuálním metodám
- nevýhody oproti individuálním metodám
- není žádná jednotná teorie jak fúzovat
- důraz na heuristiky, experimenty, praktické zkušenosti
- kombinace se statistickými, numerickými metodami i formálními metodami
- lepší výsledky

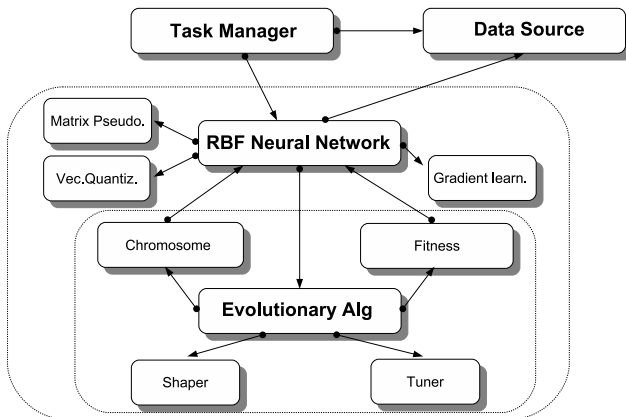
Softwaroví agenti (pokud možno ‘intelligentní’)

- agenti reprezentují jednotlivé metody výpočetní inteligence
- složitější modely (hybridní) jsou množiny komunikujících (kooperujících) agentů
- popsat konfigurace a vztahy agentů pomocí schémat multi-agentních systémů (MAS)
- jednoduché ‘zapojení’ metody (implementované agentem) do MAS
- distribuovaná exekuce
- autonomie agentů
- autonomie MAS

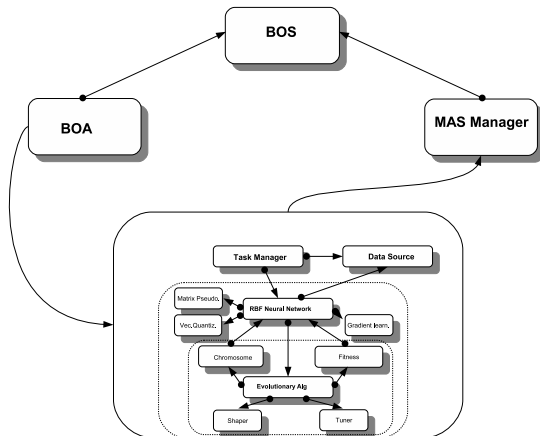
MAS



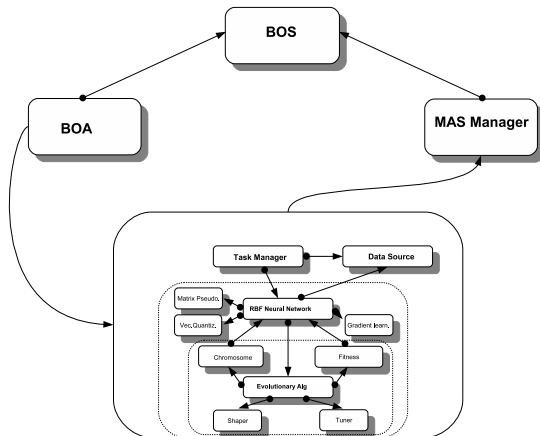
bigMAS



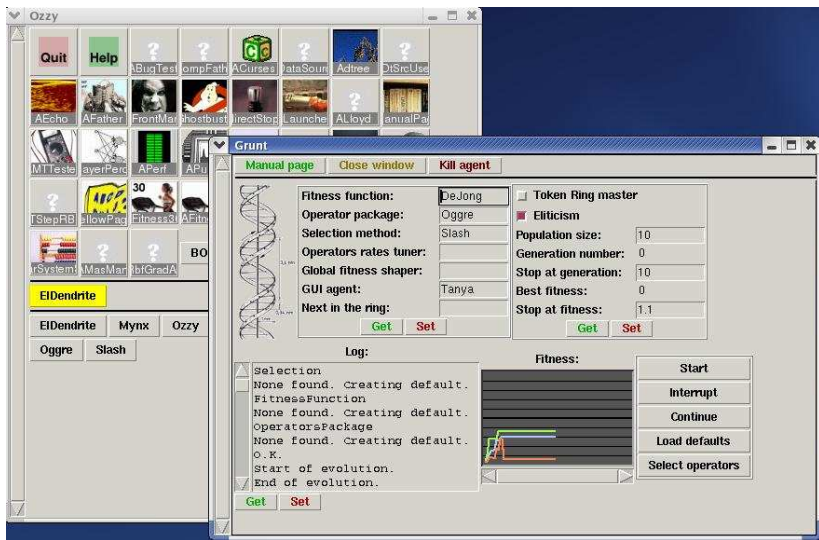
megaMAS



megaMAS



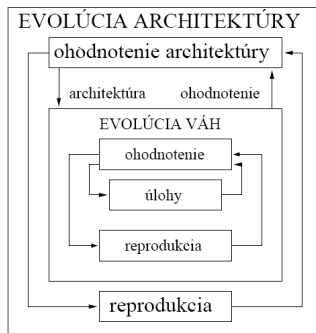
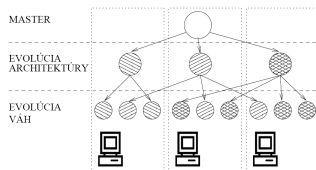
megaMAScolor



Outline

- 1 Co?
- 2 Kdo? (a Jak)?
- 3 Kde? (je ten sémantický web?)
- 4 Co dál?

Kombinace NS+EA: P.Kudová, S.Slušný



Formální popisy agentů, skládání MAS: G.Beuster

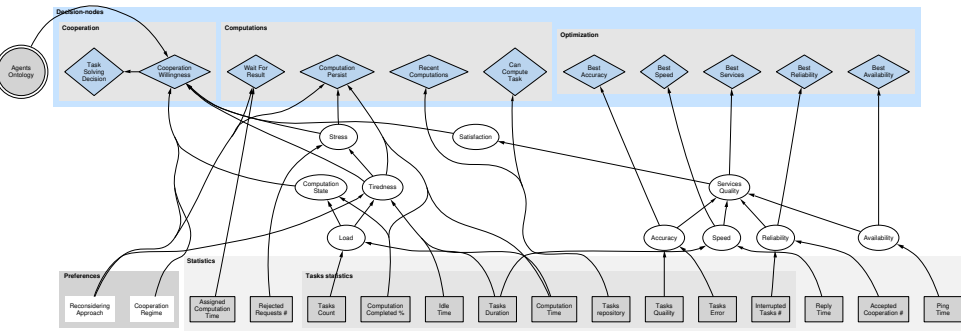
Concepts	
mas(C)	C is a Multi-Agent System
class(C)	C is the name of an agent class
gate(C)	C is a gate
m_type(C)	C is a message type
Roles	
type(X,Y)	Class X is of type Y
has_gate(X,Y)	Class X has gate Y
gate_type(X,Y)	Gate X accepts messages of type Y
interface(X,Y)	Class X understands mess. of type Y
instance(X,Y)	Agent X is an instance of class Y
has_agent(X,Y)	Agent Y is part of MAS X

```

comp_MAS(MAS) ←
  type(CAC, computational_agent) ∧
  instance(CA, CAC) ∧
  has_agent(MAS, CA) ∧
  type(DSC, data_source) ∧
  instance(DS, DSC) ∧
  has_agent(MAS, DS) ∧
  connection(CA, DS, G) ∧
  type(TMC, task_manager) ∧
  instance(TMC, TM) ∧
  has_agent(MAS, TM) ∧
  connection(TM, CA, GC) ∧
  connection(TM, GC, GD)

```


Adaptivní 'inteligentní' agenti: R.Vaculín



Implementace middleware: P.Krušina

```
#include "../base/bang.h"

class ADataSink : public AFatherT3 {
    AGENTIMPLCD( ADataSink );
    LABEL(FileName) CString name;
    FILE* file;
    CRox* save(CRox& x);
};

#endif
```

```
class ADataSink : public AFatherT3 {
    AGENTIMPLCD3( ADataSink )
    /*HERE STARTS H_MID*/
    CRox * Trgr_____mod_adatasink_bc_0(T3TRGRHIDARGS, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_2(T3TRGRHIDARGS, CString & x, CRox & y, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_6(T3TRGRHIDARGS, CString & x, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_10(T3TRGRHIDARGS, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_12(T3TRGRHIDARGS, CRox & x, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_13(T3TRGRHIDARGS, CString & x, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);
    CRox * Trgr_____mod_adatasink_bc_17(T3TRGRHIDARGS, CRox & x, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol);

    /*HERE STARTS H_END*/
    Binder T3TFLAG Trgr_____mod_adatasink_bc_0
    static int Initfn_colseq_____mod_adatasink_bc_0()//(((Trgr_____mod_adatasink_bc_0
    {
        colseq_____mod_adatasink_bc_0 = new T3ColMsg_t( new CXml("root", 0, new CXml("msg", 0, new CXml("request", 0, new CXml("init", 0, (void*) 0), (void*) 0), (void*) 0), new CXml("!", 0, 0,
    )
    )
    AUTOEXEC( AE_BLSMSGs, Trgr_____mod_adatasink_bc_0, "init trgr from _____mod_adatasink_bc:line", Initfn_colseq_____mod_adatasink_bc_0());

    int AttachTrgr_____mod_adatasink_bc_0()
    {
        AttachTrigger( colseq_____mod_adatasink_bc_0, &GlobTrgr_____mod_adatasink_bc_0, ADataSink::TypeID );
        return 0;
    }

    AUTOEXEC( AE_ATTACHs, Trgr_____mod_adatasink_bc_0, "attach trgr from _____mod_adatasink_bc:line", AttachTrgr_____mod_adatasink_bc_0()); //!!!! _____mod_adatasink_bc:line
    //)))
    #else
    CRox * ADataSink::Trgr_____mod_adatasink_bc_0(T3TRGRHIDARGS, CGate & sender, CRox & interface, CRox & conversation, CRox & protocol)
    {
        BERR("Should not be called."); return 0;
    }
    #endif
    static int Initfn_colseq_____mod_adatasink_bc_1()
    {
        colseq_____mod_adatasink_bc_1 = new T3ColMsg_t( new CXml("ok", 0, (void*) 0), 0);
        return 0;
    }
}
```

Outline

- 1 Co?
- 2 Kdo? (a Jak)?
- 3 Kde? (je ten sémantický web?)**
- 4 Co dál?

Mluvíme stejnými jazyky



- formální popis: deskriptivní logika + prolog / FOL (KR-Hyper)
- ontologie (Racer), OWL-DL
- komunikace ve FIPA-ACL, inzerce služeb
- interface na webové služby

Můžeme pomoci

- nástroj pro modelování:
 - rychlý prototyp modelu z dostupných komponent
 - poloautomatické testování úspěšnosti metod
 - distribuované prostředí
- výzkumný nástroj:
 - nové metody
 - nové způsoby skládání metod

Potřebujeme pomoci

- 'správné' otologie agentů,
- popis MAS, dedukce vlastností
- zatím in vitro
- blíž k datům a dataminingu
- ontologie dat
- inzerce služeb

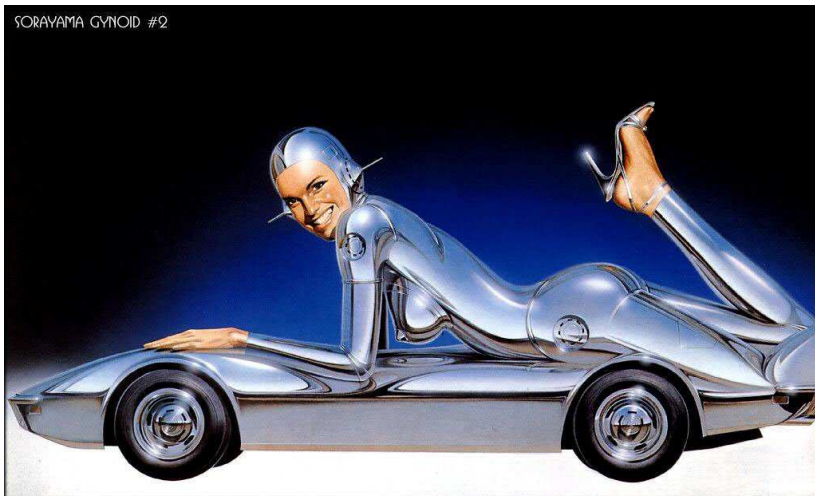
Outline

- 1 Co?
- 2 Kdo? (a Jak)?
- 3 Kde? (je ten sémantický web?)
- 4 Co dál?**

Více agentů, více agentů, více agentů



Lepší agenti



... a ještě

- rutinní práce se schématy
- perzistence agentů, dynamická klasifikace
- sklady zmražených expertů
- propojení popisů MAS s automatickou konfigurací
- hybridní evoluce schémat MAS
- webové služby/OWL-S