

Inteligentní softwaroví agenti Multiagentní systémy Game AI

Ondřej Rohlík

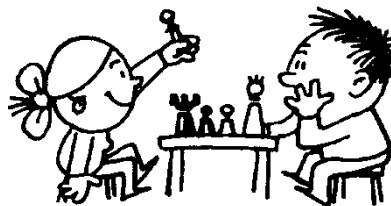
KIV FAV ZČU

rohlik.org

Cíl přednášky

- Inteligentní agenti a MAS
- Aplikace
- Přehled „Game AI“ technik

- Vzbudit zájem o PRJ5/BP/DP/EEG
- Vzbudit zájem o předmět Inteligentní software a obor Inteligentní počítačové systémy



Credits aneb Komu čest, tomu čest

- Jeff Orkin
 - Tvůrce přelomové AI pro F.E.A.R. (a MIT studenty)
- Jakub Gemrot, Michal Bída, Rudolf Kadlec at Pogamut team
 - Pogamut 2 framework a IDE
- Agents Technology Group ČVUT Praha
 - AgentFly
 - Michal Pěchouček, Přemek Volf
- Arnoštka Netrvalová a Václav Matoušek
 - Některé materiály k přednášce

Agent - definice

- In general, **agent** is an entity which interacts with its environment according to its own active properties, or preferences and goals. It can be an abstract, software, physical or human system, as well as a chemical substance.
- In computer science, a **software agent** is a piece of software that acts for a user or other program in a relationship of agency. Such "*action on behalf of*" implies the authority to decide which (and if) action is appropriate. The idea is that agents are not strictly invoked for a task, but activate themselves.

Srovnání abstrakce agentů a objektů

Objekt	Agent
Zapouzdřuje proměnné a metody	Zapouzdřuje chování
Komunikace voláním metod (posíláním zpráv)	Komunikace pomocí vyšších (čti: speciálních) jazyků
Prostředí (kromě kompilačního) nehraje žádnou roli	Prostředí, ve kterém se agent pohybuje, je velmi důležité

[Kubík 2004]

Některé příklady aplikací

- hledání a filtrování informací, data mining
- automatické reagování na signály, řízení
- monitoring a sledování
- hry, film, simulace

Intelligentní agent (1)

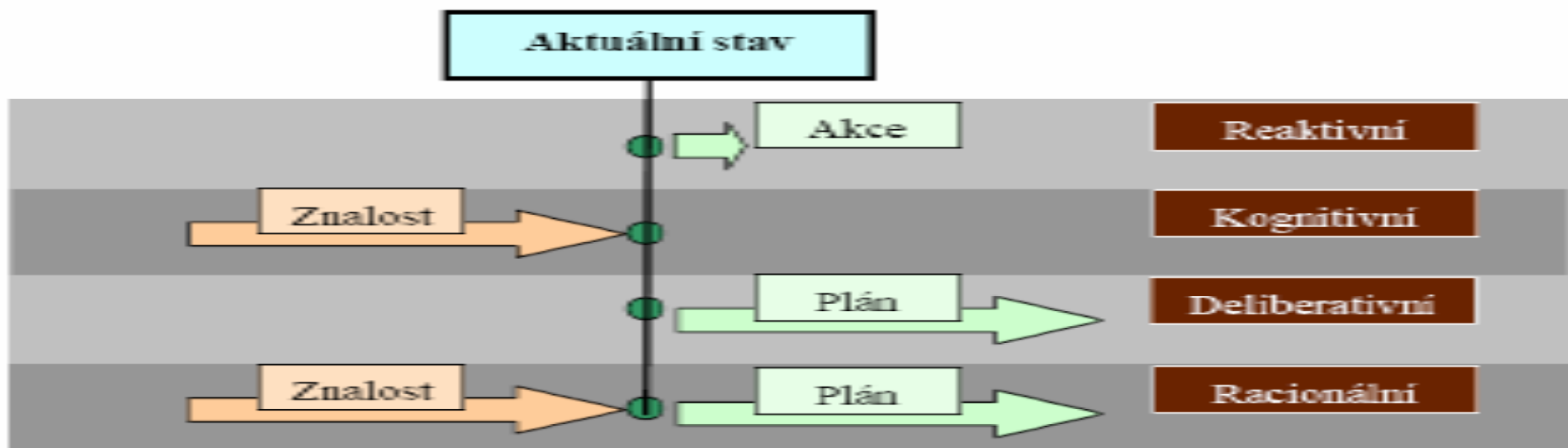
- inteligentní agent je software, který pomáhá uživateli a samostatně jedná podle jeho vůle
- Intelligentní agent se od jiného software odlišuje aspektem pověření agenta uživatelem k samostatnému provádění nějakého úkolu.
Intelligentní agent má následující základní vlastnosti:
 - **agent je autonomní** – má kontrolu nad vlastními akcemi,
 - **agent je řízen cílem** – agent může být program, skript, nebo množina pravidel,
 - **agent reaguje** – registruje změny prostředí a na změny odpovídá provedením akce,
 - **nepřetržitý běh** – bez přítomnosti uživatele.

Inteligentní agent (2)

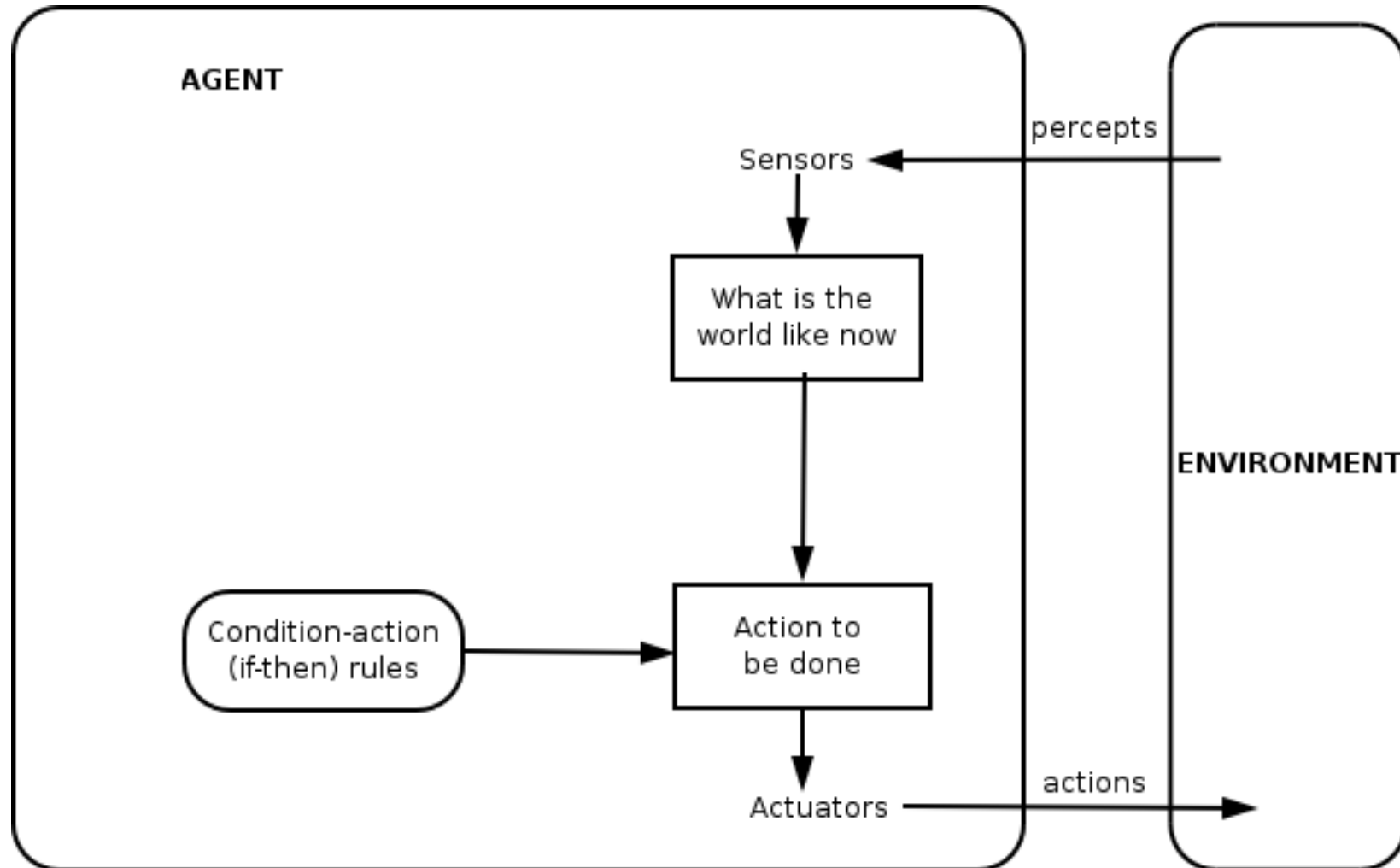
- Inteligentní agent se od jiného software odlišuje aspektem pověření agenta uživatelem k samostatnému provádění nějakého úkolu. Inteligentní agent má následující základní vlastnosti:
 - **agent je autonomní** – má kontrolu nad vlastními akcemi,
 - **agent je řízen cílem** – agent může být program, skript, nebo množina pravidel,
 - **agent reaguje** – registruje změny prostředí a na změny odpovídá provedením akce,
 - **nepřetržitý běh** – bez přítomnosti uživatele.
- Mimo tyto vlastnosti společné pro všechny inteligentní agenty, může mít inteligentní agent další vlastnosti, které určují jeho speciální poslání:
 - **společenskost** – agent může komunikovat s jinými agenty,
 - **přizpůsobivost** – agent mění své chování na základě předchozí zkušenosti,
 - **mobilita** – agent se může přemisťovat v síti z jednoho stroje na druhý.

Taxonomie agentů (jedna z mnoha)

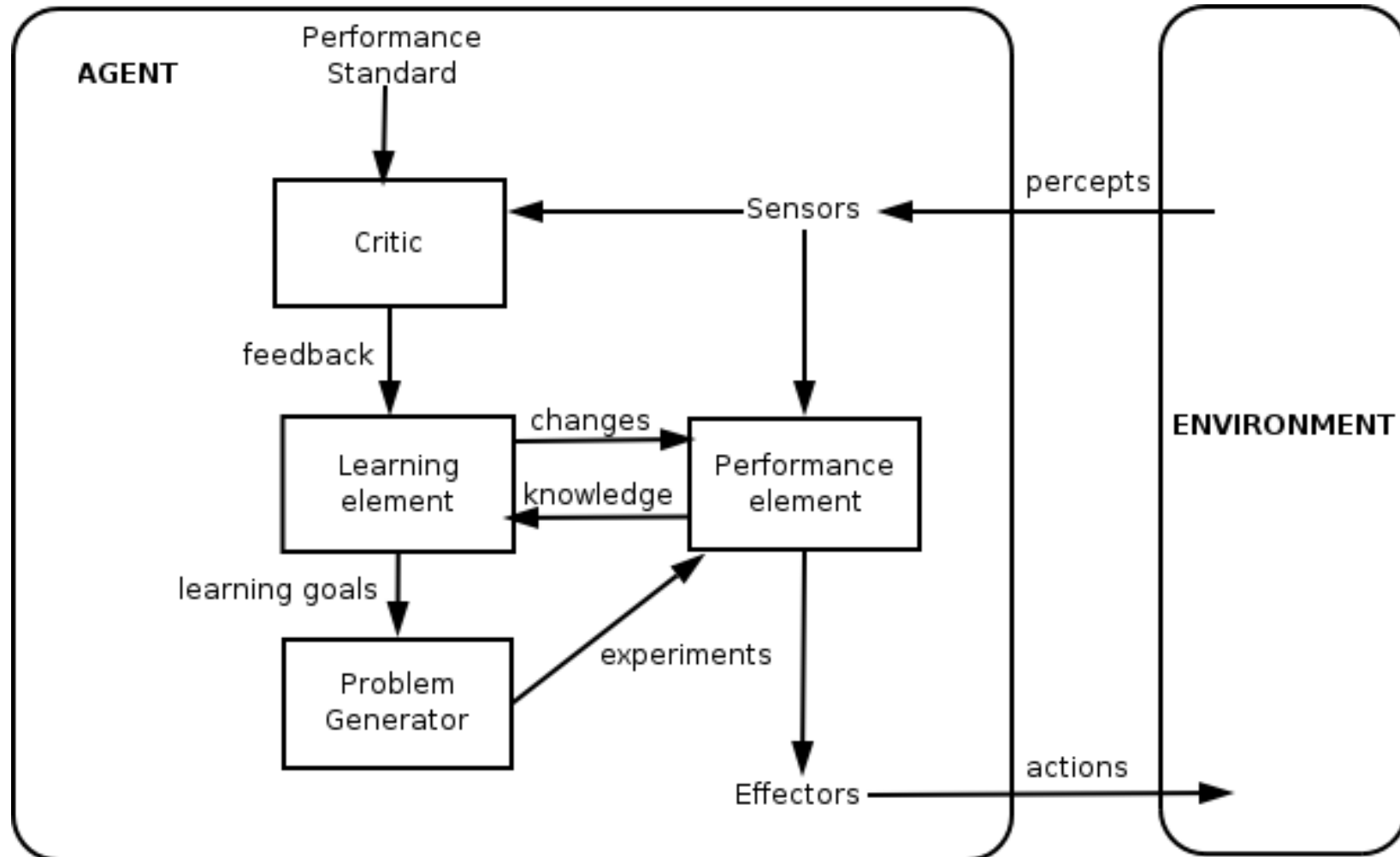
- **Agent reaktivní** – má schopnost reakce na podněty
- **Agent deliberativní (uvažující)** – má schopnost **plánovat** postup svých akcí, provádět výpočty, ovlivňovat prostředí tak, aby získal výhodu (proaktivita)
- **Agent kognitivní** – má schopnost vyvozovat logické závěry ze svých pozorování okolního prostředí. Je schopen se **učit** a vytvářet bázi znalostí (ukládá tam informace a znalosti získané dedukcí). Musí mít deliberativní schopnosti. Pak může provádět vnitřní akce jako je analýza scény, překlad a získávání dalších znalostí.
- **Agent racionální** – musí mít všechny výše uvedené schopnosti. Jeho struktura obsahuje plánovací jednotku i kognitivní jednotku včetně báze znalostí. Na základě svých poznatků je schopen se učit a pak racionálním způsobem plánovat svou činnost pro dosažení cílů.



Reaktivní agent



Učící se agent



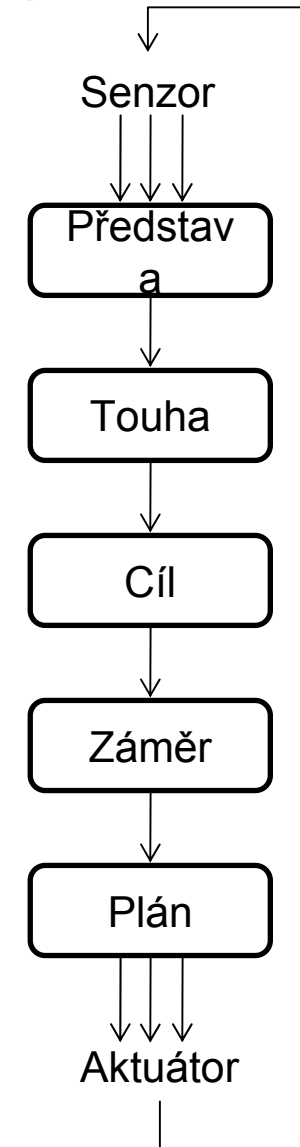
BDI Agent – Belief-Desire-Intention

Beliefs represent the informational state of the agent - in other words its beliefs about the world (including itself and other agents). Beliefs can also include inference rules, allowing forward chaining to lead to new beliefs. Typically, this information will be stored in a database (sometimes called a belief base), although that is an implementation decision. Using the term belief - rather than knowledge - recognizes that what an agent believes may not necessarily be true (and in fact may change in the future).

Desires (or goals) represent the motivational state of the agent. They represent objectives or situations that the agent would like to accomplish or bring about. Examples of desires might be: find the best price, go to the party or become rich. Usage of the term goals adds the further restriction that the set of goals must be consistent. For example, one should not have concurrent goals to go to a party and to stay at home - even though they could both be desirable.

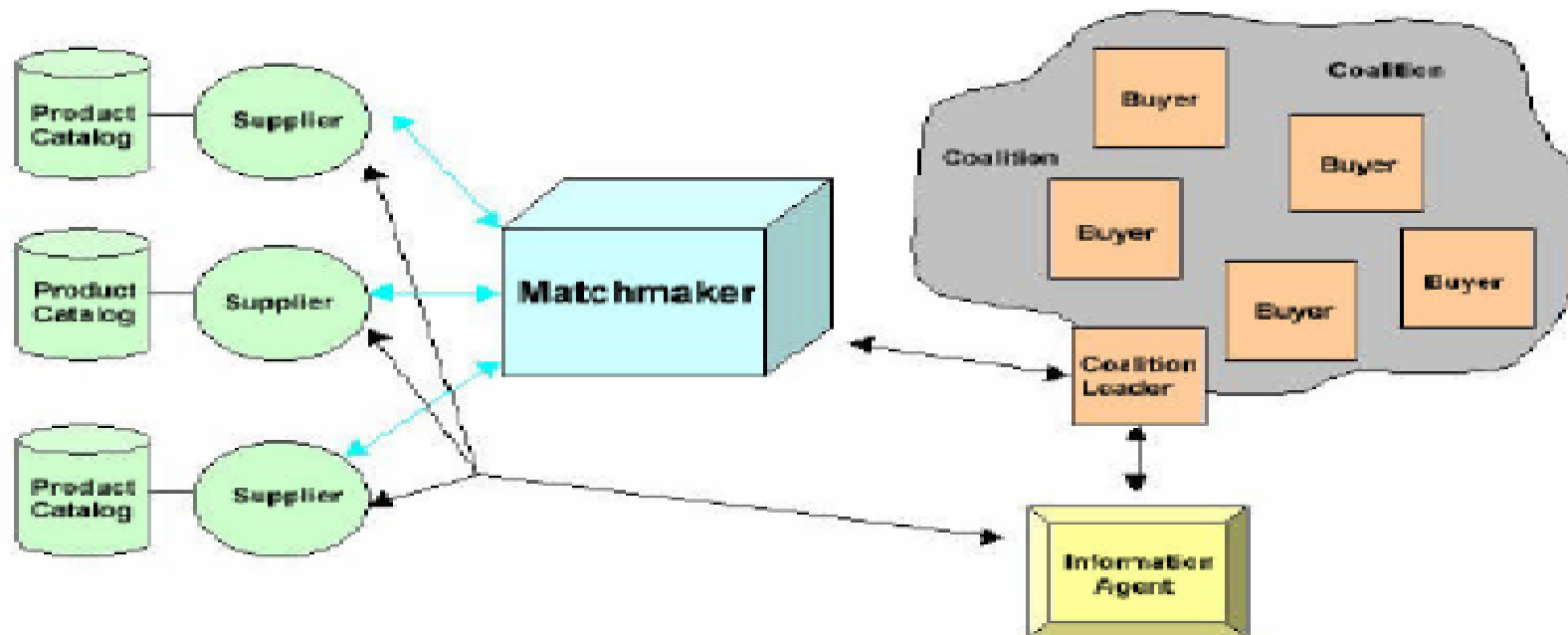
Intentions represent the deliberative state of the agent: what the agent has chosen to do. Intentions are desires to which the agent has to some extent committed (in implemented systems, this means the agent has begun executing a plan).

Plans are sequences of actions that an agent can perform to achieve one or more of its intentions. Plans may include other plans: my plan to go for a drive may include a plan to find my car keys. This reflects that in Bratman's model, plans are initially only partially conceived, with details being filled in as they progress.



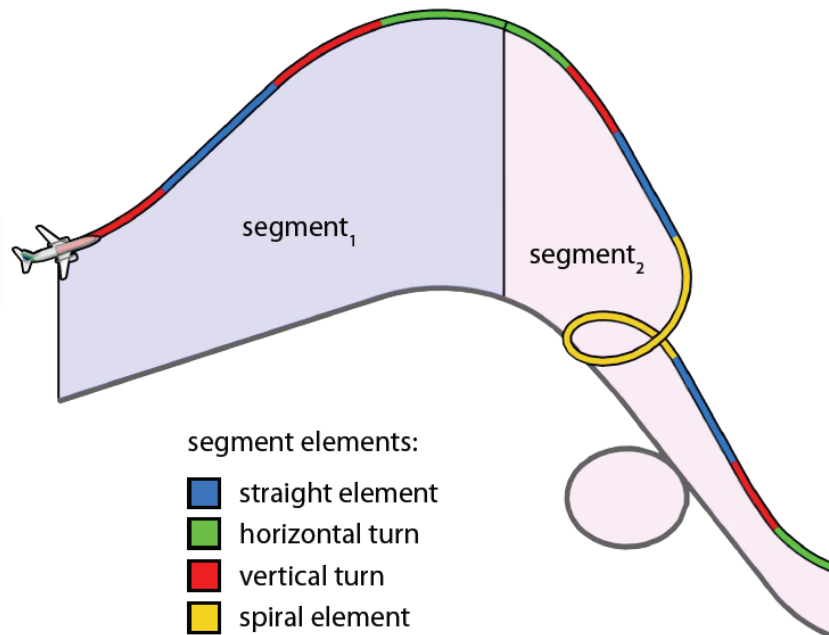
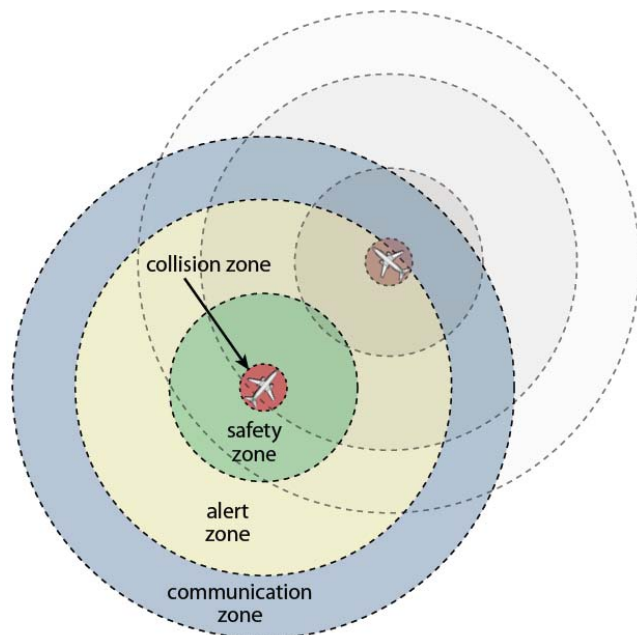
Multiagentní systémy (MAS)

- Více agentů => komunikace, kooperace, koordinace => soulad nebo kolize záměrů
- Vyjednávání, dohody, závazky a koalice



Aplikace – AgentFly [Pěchouček et al]

- 3D Space and Time Planning
- Planning the mission (= flight trajectory) is constrained by the series of:
 - time-specific waypoints
 - set of no-flight zones - landscape, static zones (eg. powerplants, airport areas) and dynamic zones (other airplanes, unsuitable weather condition).
- Replanning - based on continual situation assessment the system performs seamlessly instant replanning of the planned flight trajectory.
 - invoked by the change of the mission or by application of the evasion manoeuvres during the collision avoidance
 - reusing of the plan parts - plan composed from segments and elements



video 126

7.5.2008 ~ KIV FAV ZČU

Aplikace – Umělé bytosti

- Umělý tvor s **virtuálním tělem, který "žije" ve virtuálním prostředí** podobném přirozenému prostředí
- Aplikace
 - Hry (pro zábavu)
 - Storytelling př. Facade
 - Virtuální herci v simulacích (policie, armáda př. VBS)
 - Testování pracovních postupů
 - Filmový průmysl
 - <http://www.massivesoftware.com/>



Façade – Virtuální drama



NLP

Serious Games – Tactical Iraqi



Computer games – pro zábavu

- Boti v FPS jako agenti
- Nevýhoda: je jich velká spotřeba ;-)
- Průměrná doba života nepřátelské AI:
12.23 sekundy
- Počet AI které umřou v FPS hrách za den:
19 789 203



BEFORE



Game bots

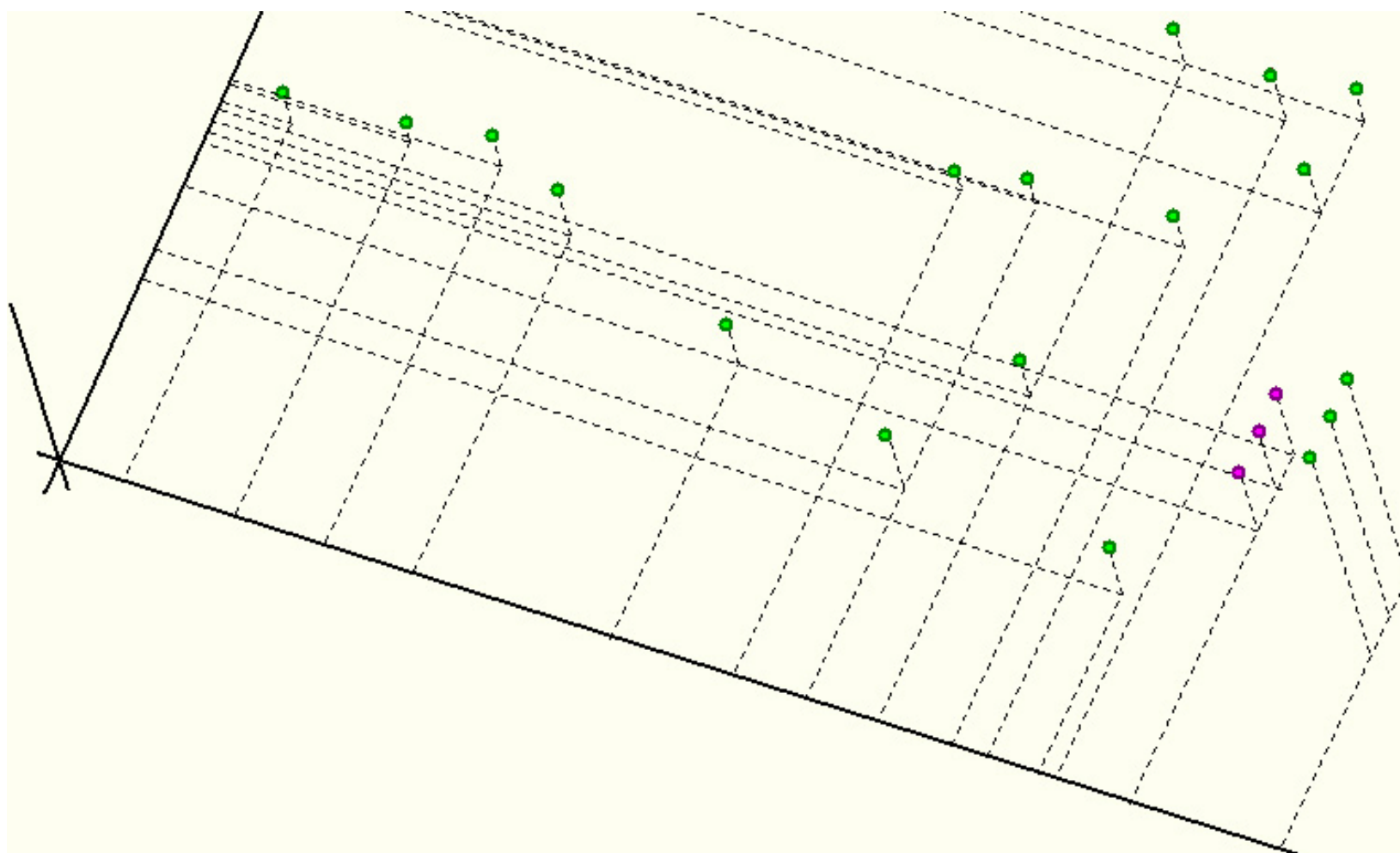
- **Environment**
 - Navigation points, graph
 - Floor coverage
 - Rectangles, circles
 - NavMesh
 - Visibility matrix
 - Doors, levers, lifts, jumping and sniper spots
- **Logic**
 - If-then rules
 - FSM
 - Fuzzy logic
 - Planners
 - Traditional
 - Reactive
 - Anytime planners would be nice
 - Or see next slides

navigace a plánování

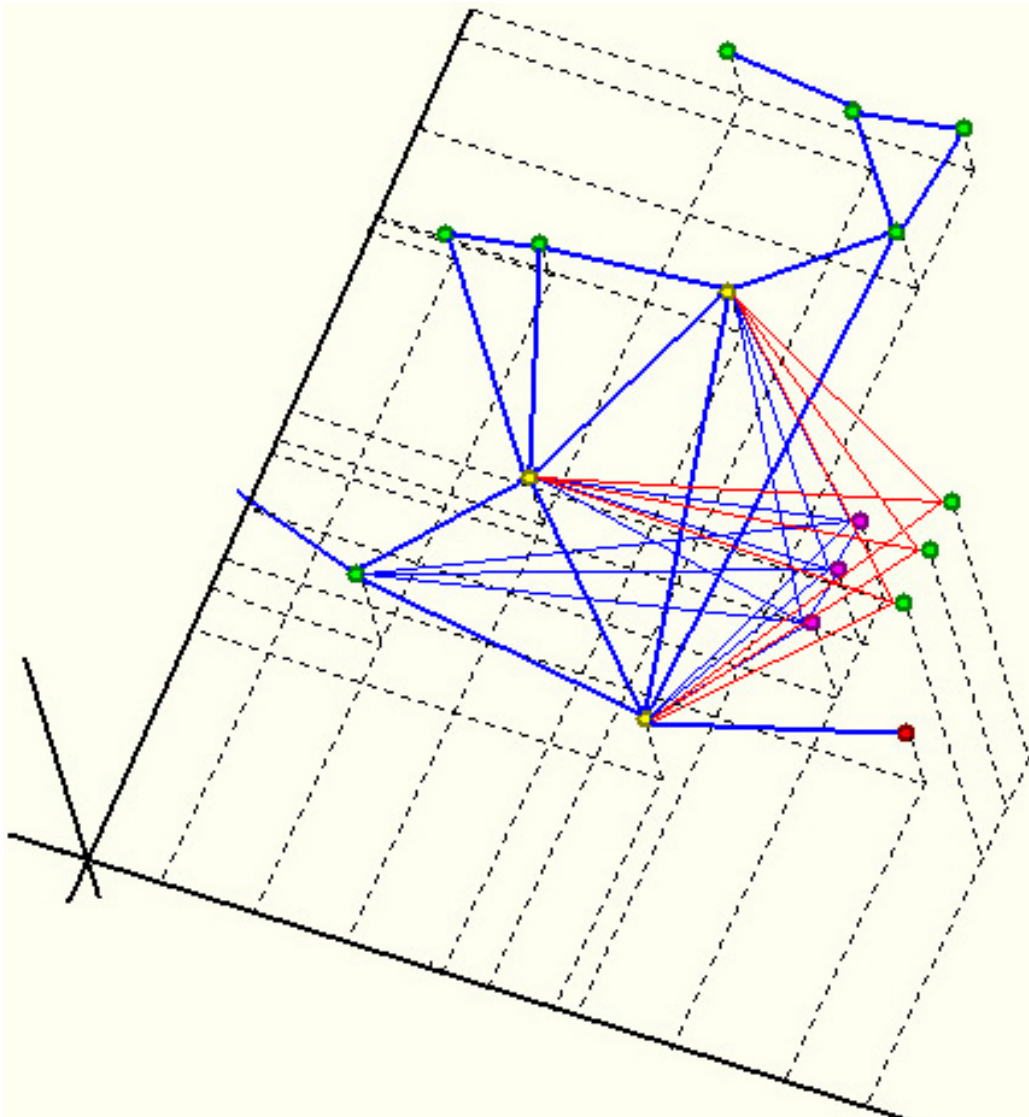
FPR (UT) a Navigation Points



Jako to vidí bot?



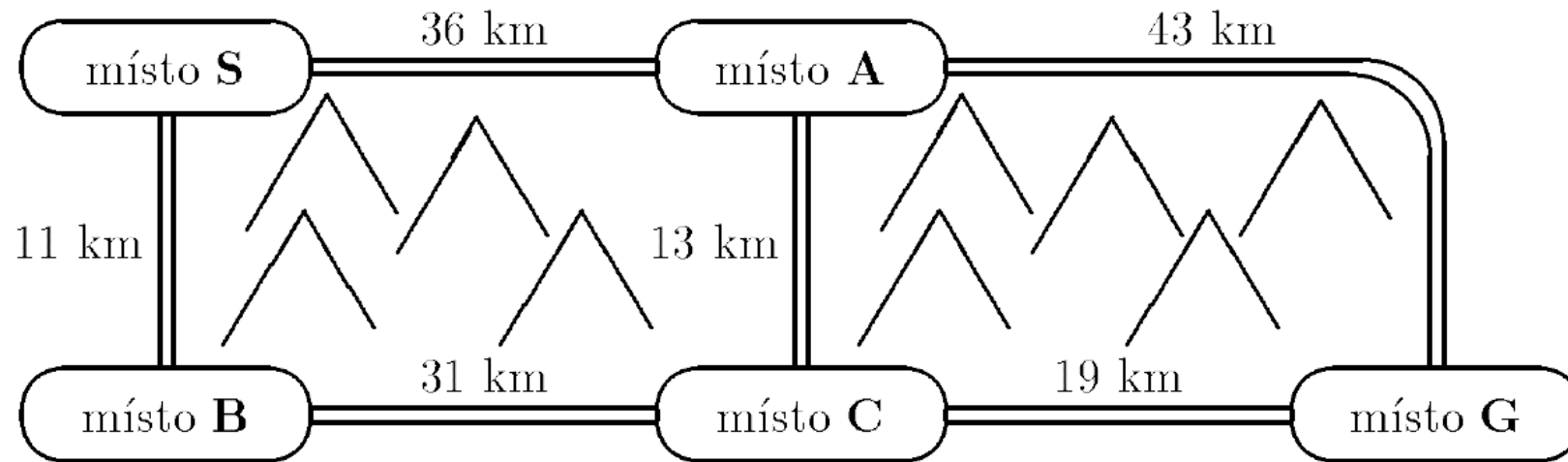
Hrany a značkování



- Reachability
 - Running / Jumping
 - Flying (translocator)
- Points flags
 - Item (+description)
 - Threat
 - Fire translocator from here
 - Ambush / sniping
 - ...

A* algoritmus

- Pamatujete na Alpy ?

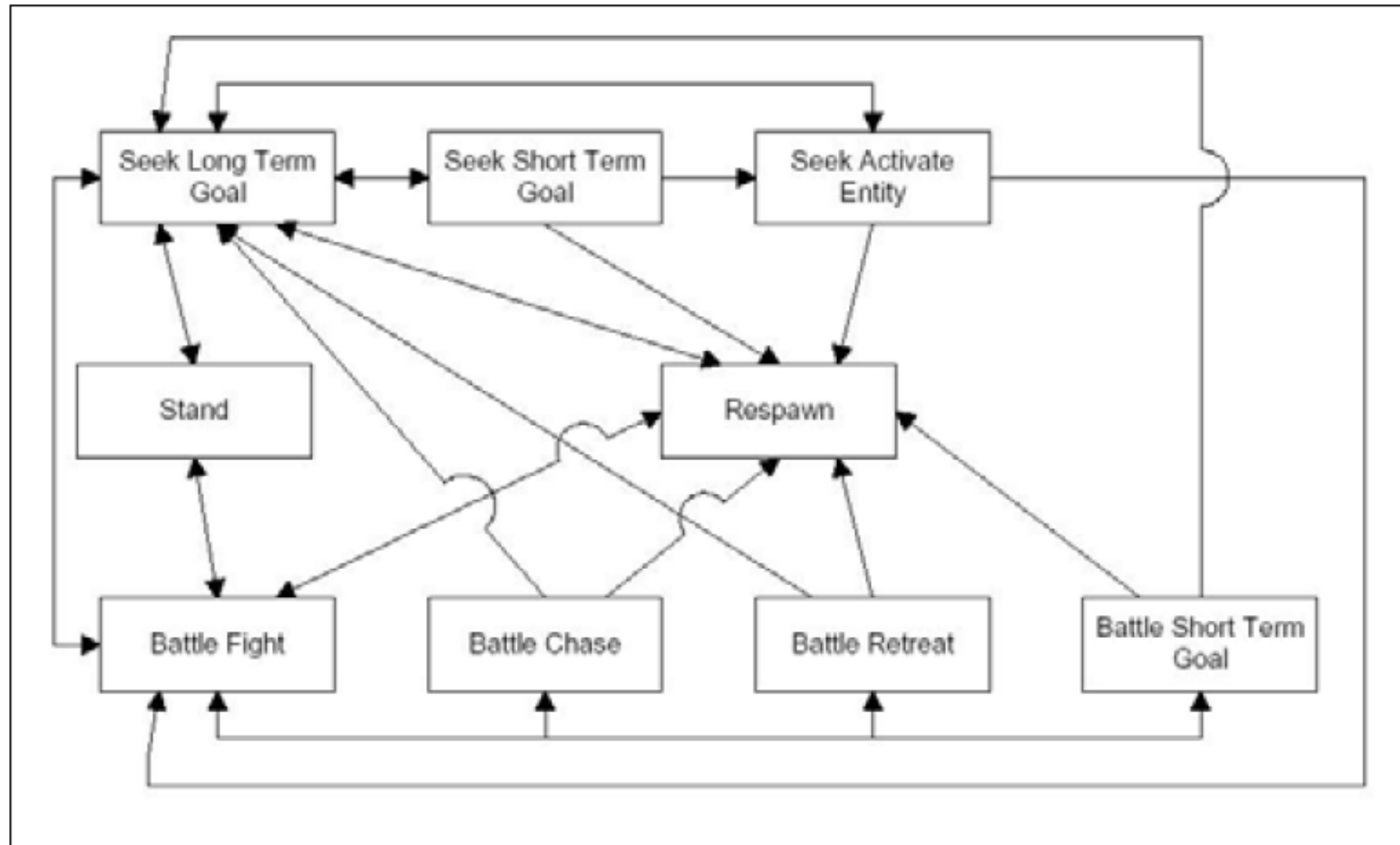


- Klasické použití (a zdroj nedorozumění):
pathfinding
 - <http://www-cs-students.stanford.edu/~amitp/gameprog.html#paths>
 - <http://www-cs-students.stanford.edu/~amitp/game-programming/a-star-flash/test.html>
- Obecné použití: hledání řešení

Action selection: What to do next?

- Konečné automaty (FSM = Finite State Machines)
 - Unreal Tournament (native support for FSM in UnrealScript)
 - The Quake III Arena bot (Jean-Paul Waveren)
- Plánování
 - je proces hledání posloupnosti **akcí**, které vedou k **cíli**.
 - Proces se jmenuje „Formulování plánu“
 - Reaktivní plánovače
 - (S)POSH
 - A*
 - STRIPS
- Chytré prostředí
 - AAS v Q3
 - Path-finding, ručně vložené nápovědy
- Složitost roste při koordinaci týmu

Konečný automat FSM



FSM jako tabulka

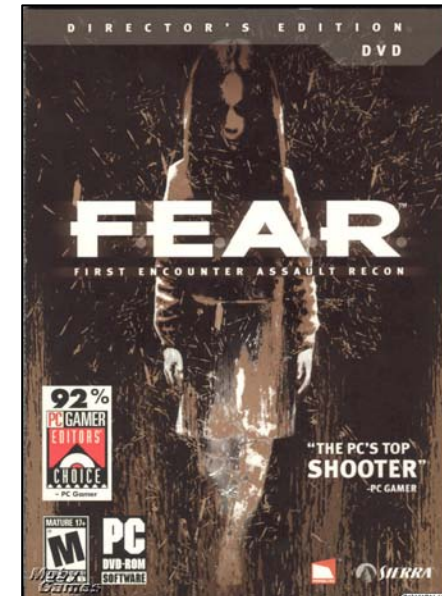
- Ekvivalentní zápis tabulkou:

Stav	Podmínka	Přechod
Runaway	Safe	Patrol
Attack	WeakerThanEnemy	RunAway
Patrol	Theatened & StrongerThanEnemy	Attack
Patrol	Theatened & WeakerThanEnemy	RunAway
...



State-of-the-art: Plánování ve F.E.A.R.

- STRIPS plánovač (r. 1970)
- F.E.A.R. SDK
 - <http://aigamedev.com/source/fear-sdk>



State: (phone#, recipe, hungry?)

Goal: (-- , -- , **NO**)

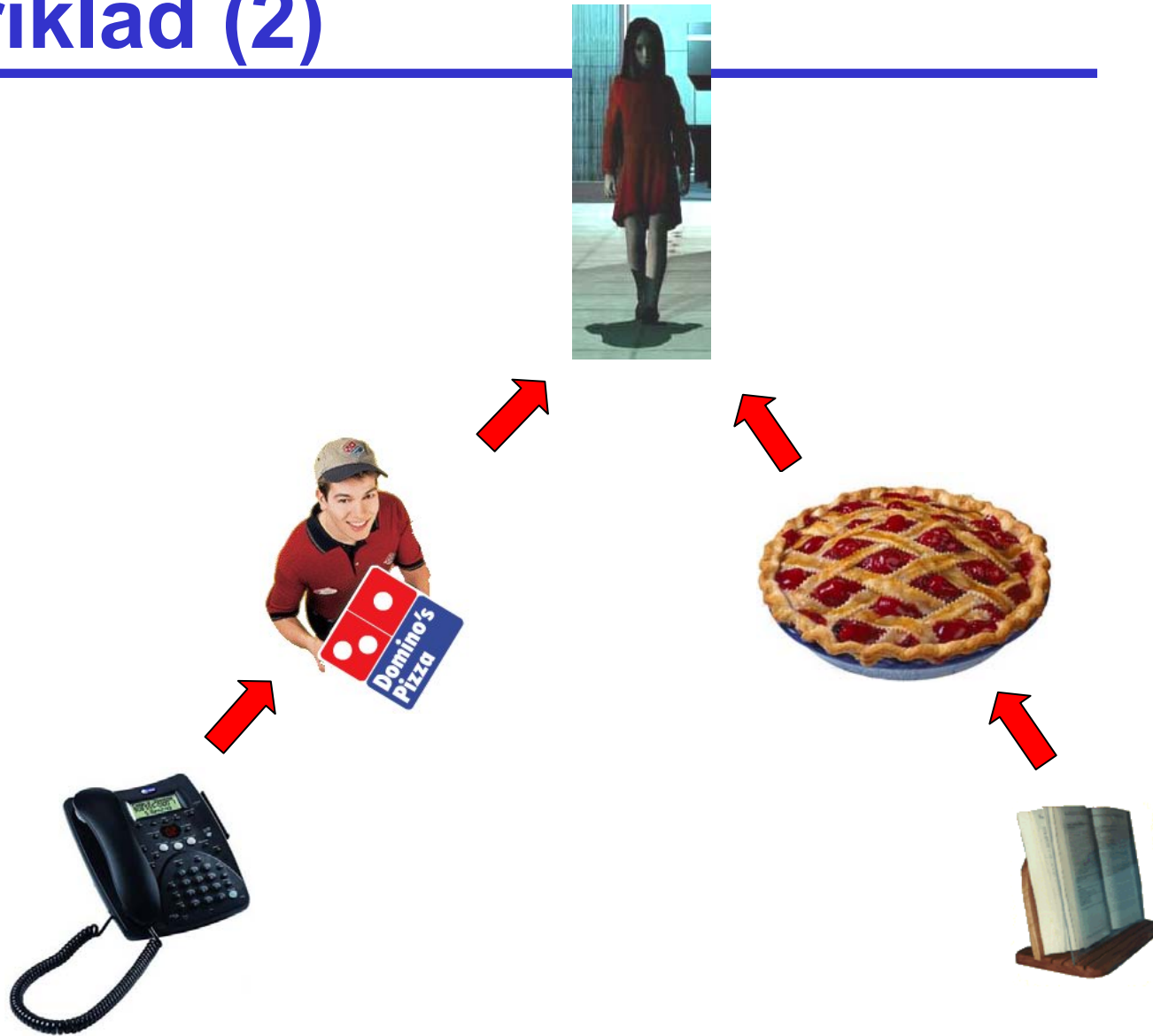
[Jeff Orkin]

Příklad (1)



[Jeff Orkin]

Příklad (2)

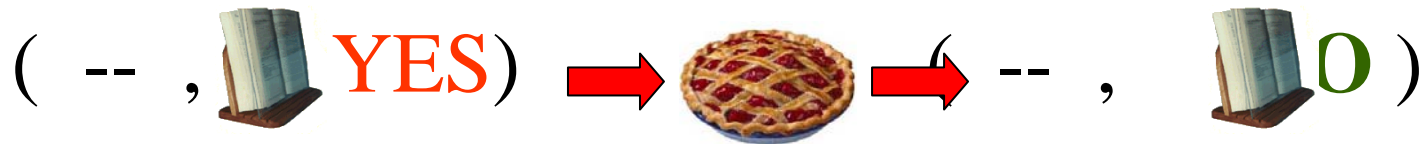
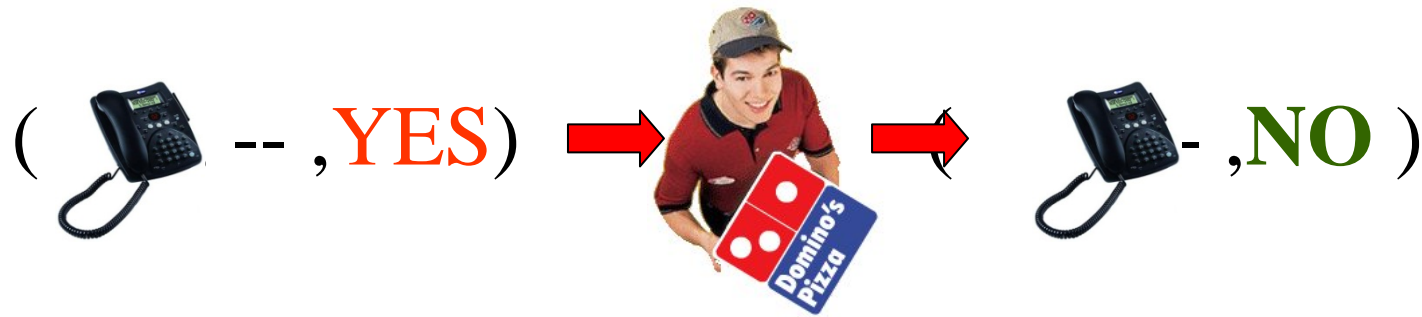


[Jeff Orkin]

Příklad (3)

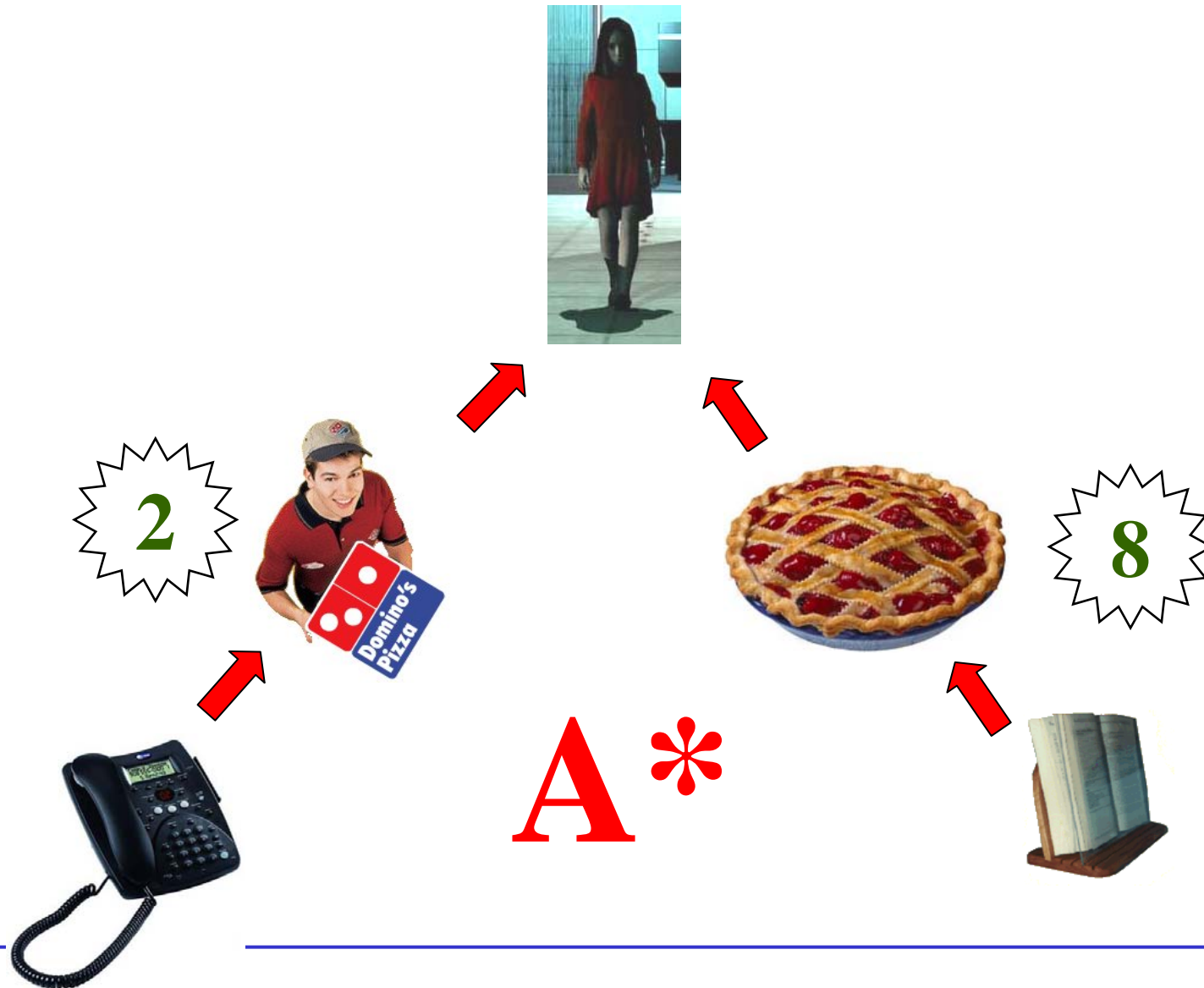
State: (phone#, recipe, hungry?)

Goal: (-- , -- , **NO**)



[Jeff Orkin]

Cost per Action



F.E.A.R.: Actions

Soldier

+	
[-]	Action
1	AI/Actions/Attack
2	AI/Actions/AttackCrouch
3	AI/Actions/SuppressionFire
4	AI/Actions/SuppressionFireFromCover
5	AI/Actions/FlushOutWithGrenade
6	AI/Actions/AttackFromCover
7	AI/Actions/BlindFireFromCover
8	AI/Actions/AttackGrenadeFromCover
9	AI/Actions/AttackFromView
10	AI/Actions/DrawWeapon
11	AI/Actions/HolsterWeapon
12	AI/Actions/ReloadCrouch
13	AI/Actions/ReloadCovered
14	AI/Actions/InspectDisturbance
15	AI/Actions/LookAtDisturbance
16	AI/Actions/SurveyArea
17	AI/Actions/DodgeRoll
18	AI/Actions/DodgeShuffle
19	AI/Actions/DodgeCovered
20	AI/Actions/Uncover
21	AI/Actions/AttackMelee

Assassin

+	
[-]	Action
1	AI/Actions/Attack
2	AI/Actions/InspectDisturbance
3	AI/Actions/LookAtDisturbance
4	AI/Actions/SurveyArea
5	AI/Actions/AttackMeleeUncloaked
6	AI/Actions/TraverseBlockedDoor
7	AI/Actions/UseSmartObjectNodeMounted
8	AI/Actions/MountNodeUncloaked
9	AI/Actions/DismountNodeUncloaked
10	AI/Actions/TraverseLinkUncloaked
11	AI/Actions/AttackFromAmbush
12	AI/Actions/DodgeRollParanoid
13	AI/Actions/AttackLungeUncloaked
14	AI/Actions/LopeToTargetUncloaked
	+

Rat

+	
[-]	Action
1	AI/Actions/Animate
2	AI/Actions/Idle
3	AI/Actions/GotoNode
4	AI/Actions/UseSmartObjectNode
	+

[Jeff Orkin]

F.E.A.R.: Dynamic Problem Solving



[Jeff Orkin]

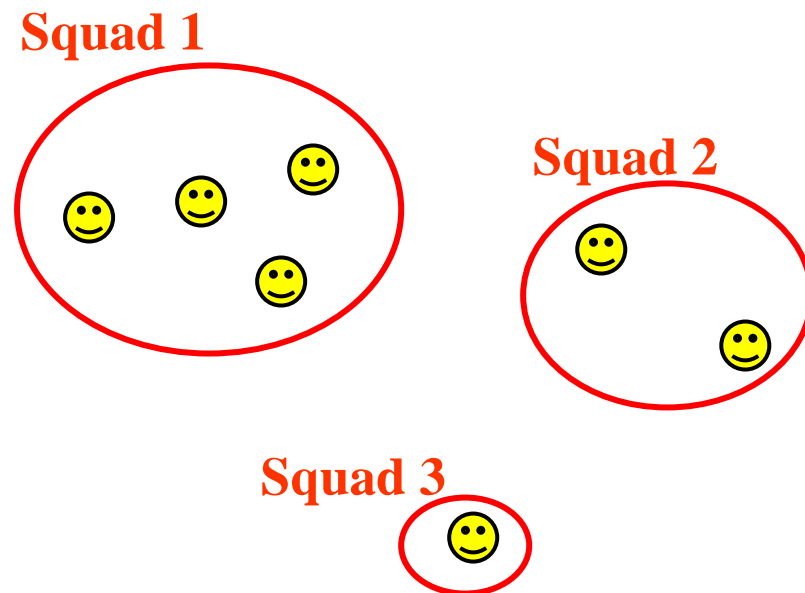
FSM vs. Plánování

- FSM
 - jak to udělat
 - procedurální
- Plánování
 - co udělat
 - deklarativní

Budoucnost je v plánování

Tým AI & Multiagentní systémy

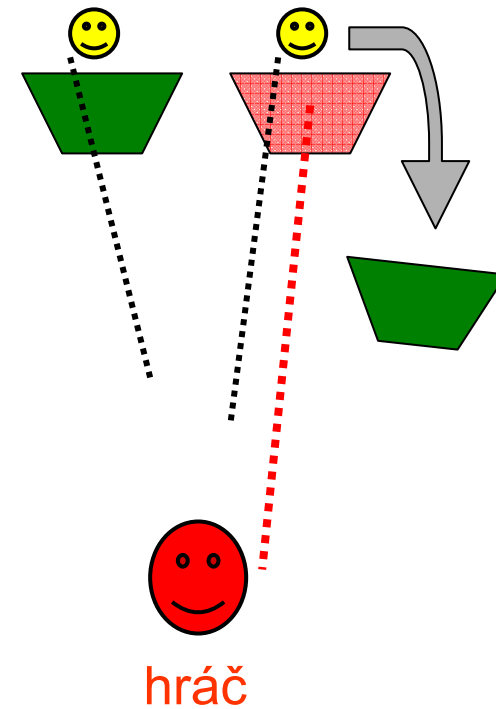
- MAS v Game AI dnes nejsou (Téma PRJ5)
- Behaviours
 - Suppression fire
 - Goto positions
 - Get to cover
 - Search in pair
 - Follow leader
 - Flanking
 - Coordinated strikes
 - Retreats
 - Reinforcements



[Jeff Orkin]

Get-to-Cover Behaviour

1. Find participants
2. Send orders
3. Monitor progress
4. Succeed or fail



[Jeff Orkin]

Komunikace

TAKE THE WINE

"I don't know what a WINE is"

TAKE THE FLASK

"I don't know what a FLASK is"

TAKE THE DAMN BOTTLE

"I don't know what a DAMN is"

TAKE THE BOTTLE

"You have taken the bottle."

Raději dialog než pouze zvolání
„Dostal jsem to!“

Vysvětlit, proč AI/NPC „nic nedělá“

- AI1: “He’s aiming at you!
Get out of there!”
- AI2: “I’ve got nowhere to
go!”

OpenDoor -or- “Open the door!”

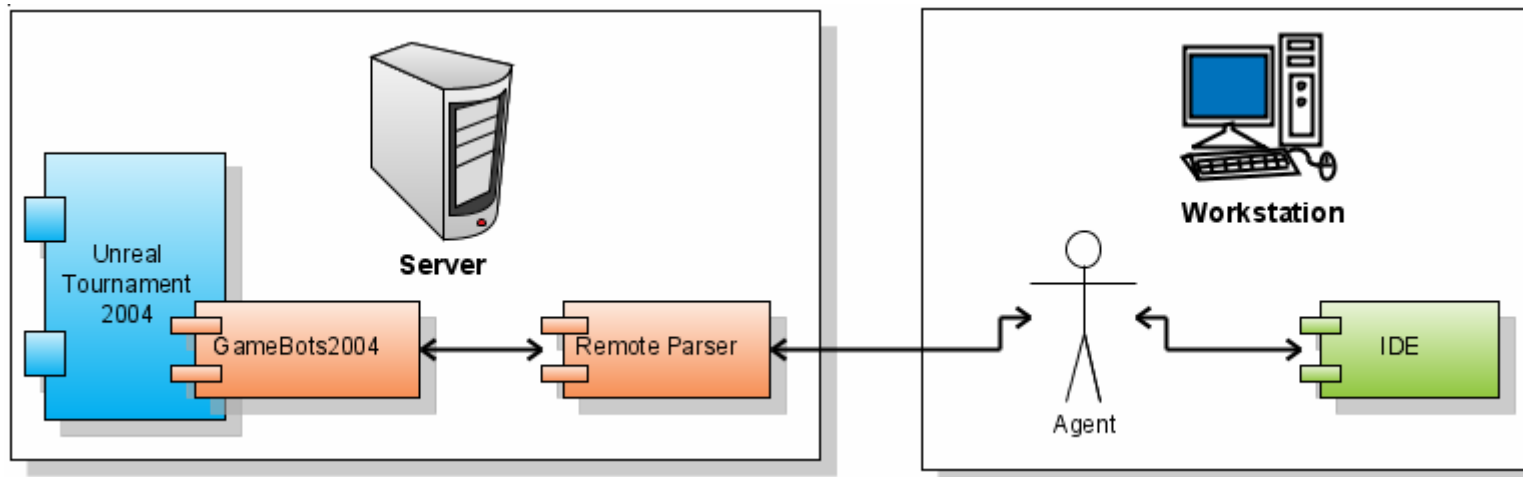


Pamatujme

- Game AI is behavioral, not scientific \Rightarrow Nejde o to vytvořit inteligenci, ale zlepšit hratelnost skrze iluzi intelligence.
- To zní jednoduše, ale má to řadu komplikací
- Vztah k tomu, co jste dosud slyšeli na UIR, je velmi vlažný :-)
- Přesto, programování botů je super
 - zábava
 - možnost prototypovat AI algoritmy
 - a proto [next slides]

Programování botů má svá úskalí

- Co je třeba vyřešit, než začnete psát AI
 - jak napojit bota na server
 - základní senzomotorická primitiva
 - jak číst mapu, navigace
 - jak interpretovat data z game engine (př. pozice ostatních botů a hráčů)



Naštěstí existuje Pogamut

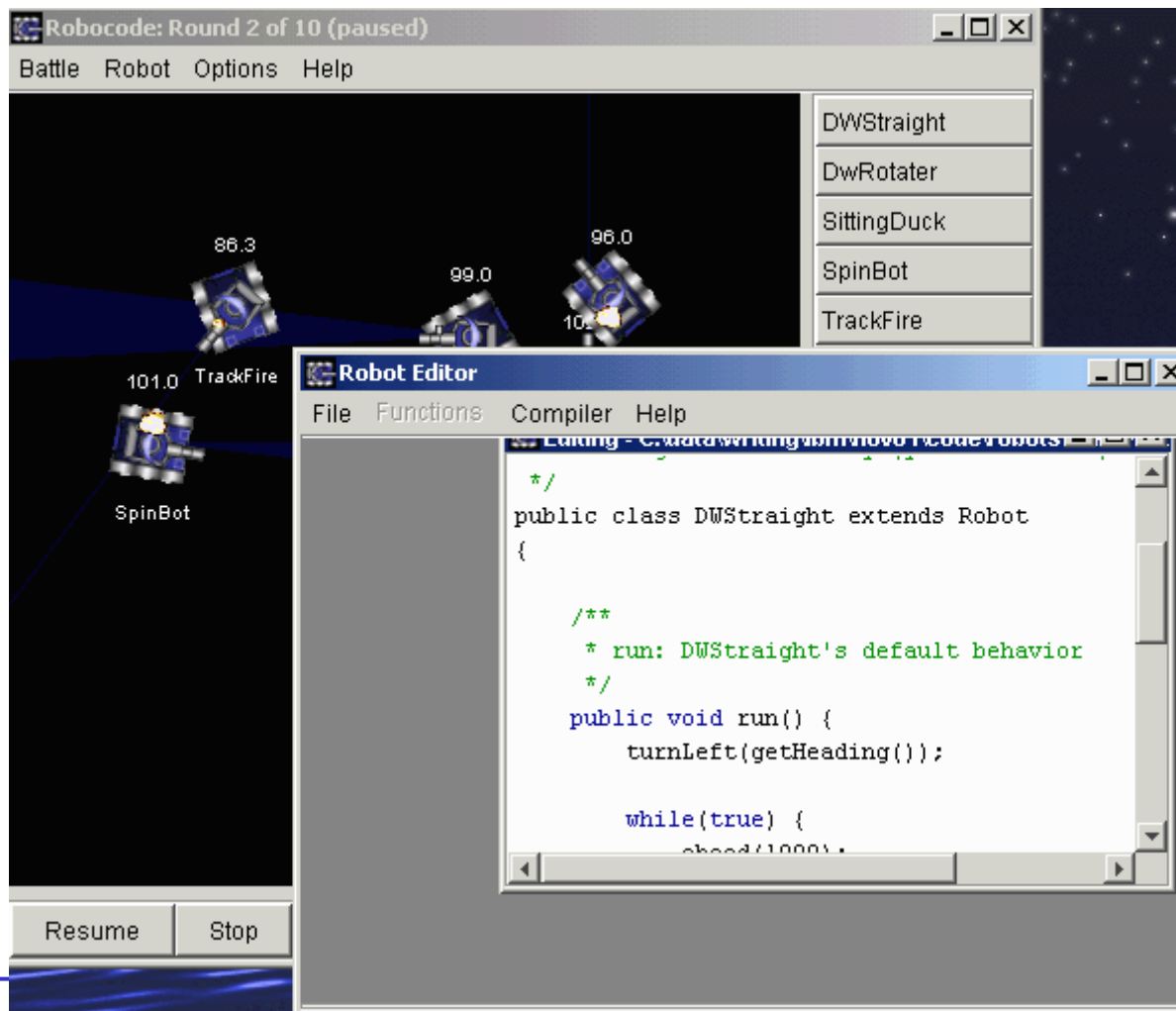
- General tool for creating bot's behavior
- Bots are embodied in Unreal Tournament 2004
- Pogamut is implemented mainly in Java and UnrealScript , minor parts are in XML, POSH and Python
- Plugin for Netbeans IDE
- Support for debugging the agent
 - Logs and filters for logs
 - Project management
 - Servers
- Tutorials, videos
- **<http://artemis.ms.mff.cuni.cz/pogamut>**

Pogamut: možnosti rozšíření (PRJ5)

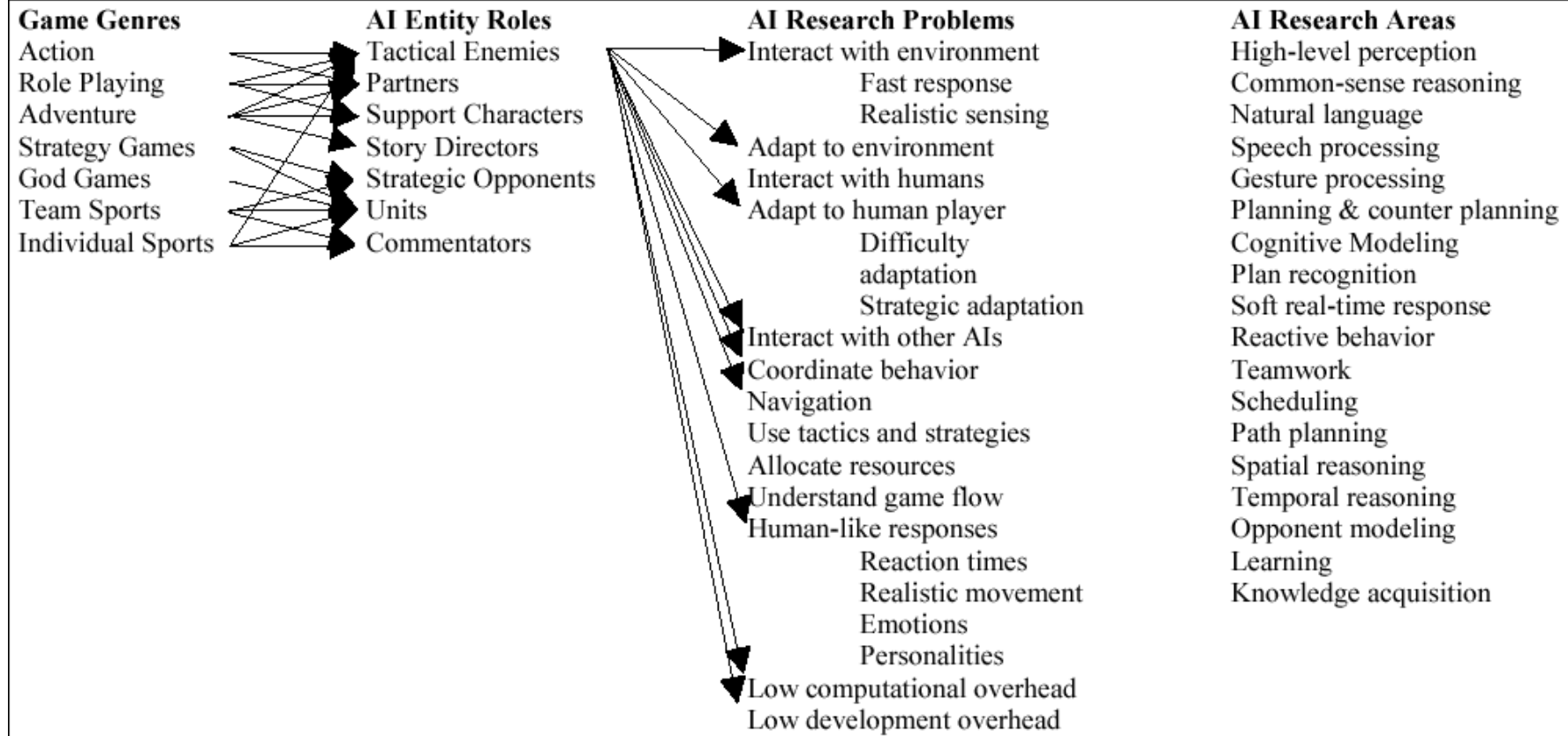
- Integration with other DMSs: Soar, Drools, natural language ...
- Tracking of human players
- Real bot debugging
- UT3 port
- Enhanced system for communication between bots (ala Jade)
- Porting gamebots to other environments
- And much more

Alternativa: RoboCode

- Prototypování MAS

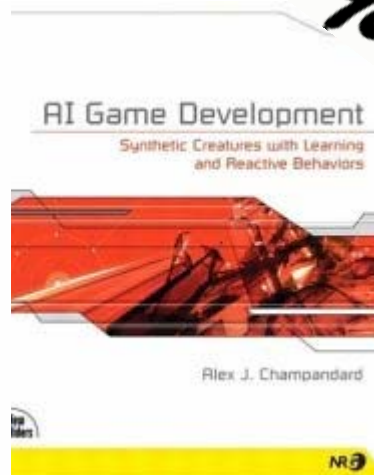
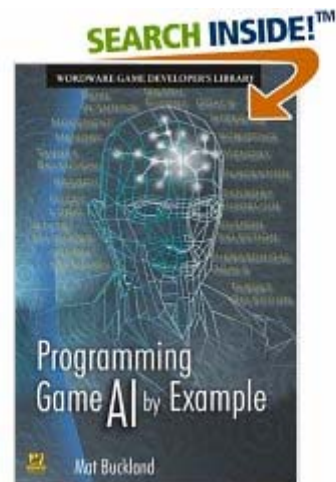


BTW, co se skrývá za pojmem Game AI?



A následujících 10 slidů přeskočíme ...

Dopručené čtení



AI Game Dev .COM

