

Androids (2010/2011)

Pohyby agenta

Pohyb agenta pozostáva z fáz. Medzi jednoduché pohyby patrí napríklad kopnutie a medzi zložitejšie resp. cyklické napríklad chôdza.

Tím Androids v špecifikácii uvádzal viacero možností, ktoré by radi implementovali. V nasledujúcej tabuľke sú uvedené všetky. Vypísané sú pohyby, ktoré boli implementované a do kategórie „Návrh do budúcnosti“ sú zaradené tie, ktoré neboli z časového hľadiska implementované.

Pohyb	Implementované	Návrh do budúcnosti
Vstávanie	<ul style="list-style-type: none">• vstávanie z brucha• vstávanie z chrbta• vsávanie zo sedu rozkročmo	<ul style="list-style-type: none">• vstávanie bez využitia prostredia (posúvaním po ihrisku)
Chôdza	<ul style="list-style-type: none">• chôdza dopredu• chôdza dozadu• chôdza do strán (úkroky)	<ul style="list-style-type: none">• rozdelenie chôdze na rýchlu - nestabilnú a pomalú – stabilnú
Otáčanie	<ul style="list-style-type: none">• vylepšovanie už použitých pohybov implementovaných v XML• vlastné pohyby (90°, 180°, vojenské otočenie)	
Kopanie	<ul style="list-style-type: none">• vytvorenie XML pre kopanie• kopanie dopredu špičkou a stranou• porovnanie kopanie celým telom vs. Kopanie kĺbov spodnej končatiny	
Bránenie	<ul style="list-style-type: none">• bránenie pádom brankára s vystretými rukami do strán• bránenie sadnutím na zem a rozkročením nôh	

XML súbory slúžia na vytváranie zostáv pohybov, ktoré pozostávajú z fáz.

Vyššia logika

Po analýze agenta JIM predchádzajúceho tímu zistili, že agent nedisponuje vyššími schopnosťami. Tie teda implementovali. Vyššie schopnosti agenta umožňujú abstrahovať od detailov nižších schopností agenta a tak zjednodušiť ich výber a postupnosť. Definovaním pohybov vyššej logiky je možné kombinovať pohyby vyššej úrovne, ktoré by sa mohli využiť pri plánovaní stratégie.

- **Chôdza** – predstavuje schopnosť dostať sa na určené miesto. Budú využité nižšie schopnosti agenta ako pohyb dopredu, či otočenie sa.
- **Vstávanie** – predstavuje schopnosť vstať z polohy ležmo. Budú použité nižšie schopnosti vstávania z chrbta a vstávania z brucha.
- **Kop do lopty** - predstavuje schopnosť kopnúť do lopty na stanovený cieľ. Týmto cieľom môže byť bránka, spoluagent, prípadne akýkoľvek bod na ihrisku. Použijeme nižšie schopnosti rôznych typov kopania do lopty a otáčania.
- **Vedenie lopty** - predstavuje schopnosť vedenia lopty jedným agentom. Bude pozostávať zo schopností kopu do lopty a chôdze.
- **Zorientovanie sa** – predstavuje schopnosť zistiť svoju vlastnú polohu na základe polohy bránok a polohu lopty. Táto schopnosť bude pozostávať zo schopnosti otáčania sa.
- **Blokovanie strely** – predstavuje schopnosť zablokovať strelu správnym postavením agenta a následným zablokovaním strely nižšími schopnosťami ako pádom, či rozkročením. Túto schopnosť bude využívať predovšetkým brankár.

Editor správania

V editore pohybov implementovali:

- Pridanie panela, ktorý slúži používateľovi na výber, či sa má kĺb ohýbať samostatne alebo spolu s jeho „bratom“ (symetrickým párovým kĺbom) a možnosť voľby, či sa má pohyb vykonať symetricky alebo po zrkadlovej trajektórii.
- Plug-in „Editor správania“, implementovaný v jazyku Java.

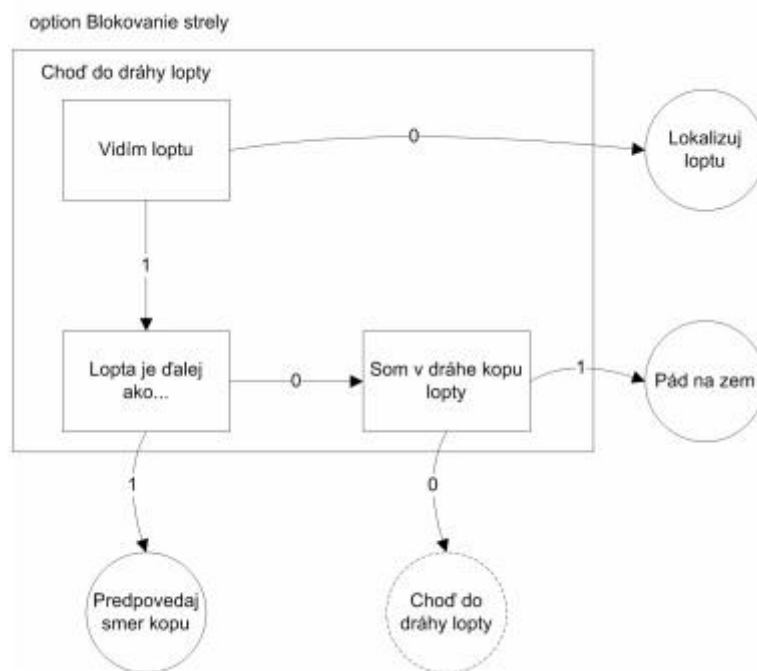
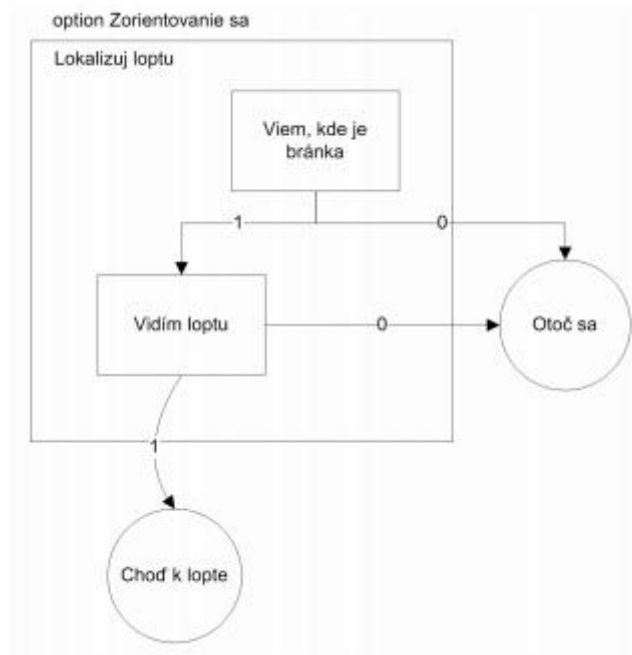
Definovanie vyššej logiky pomocou XABSL

Ako alternatíva produkčného systému plánovali použiť jazyk na definovanie vyššej logiky XABSL (Extensible Agent Behavior Specification Language), ktorý je vhodný na opísanie vyššej logiky hráča. K dispozícii je aj Java XABSL library. Opis správania pomocou XABSL môže byť výhodné najmä, ak komplexnosť správania začne rásť a implementovanie v jazyku Java by začalo byť neefektívne.

Agent v XABSL pozostáva z menších celkov – tzv. options, čo sú vlastne moduly správania, z ktorých každý je konečným automatom. V našom prípade pôjde o:

- **zorientovanie sa**
- **blokovanie strely**

Tieto sú zoskupené v strome, ktorého koncové uzly tvoria prvky nižšieho správania, ktoré vyvolajú požadovanú akciu. Prechody stavov realizujeme ako rozhodovacie stromy.

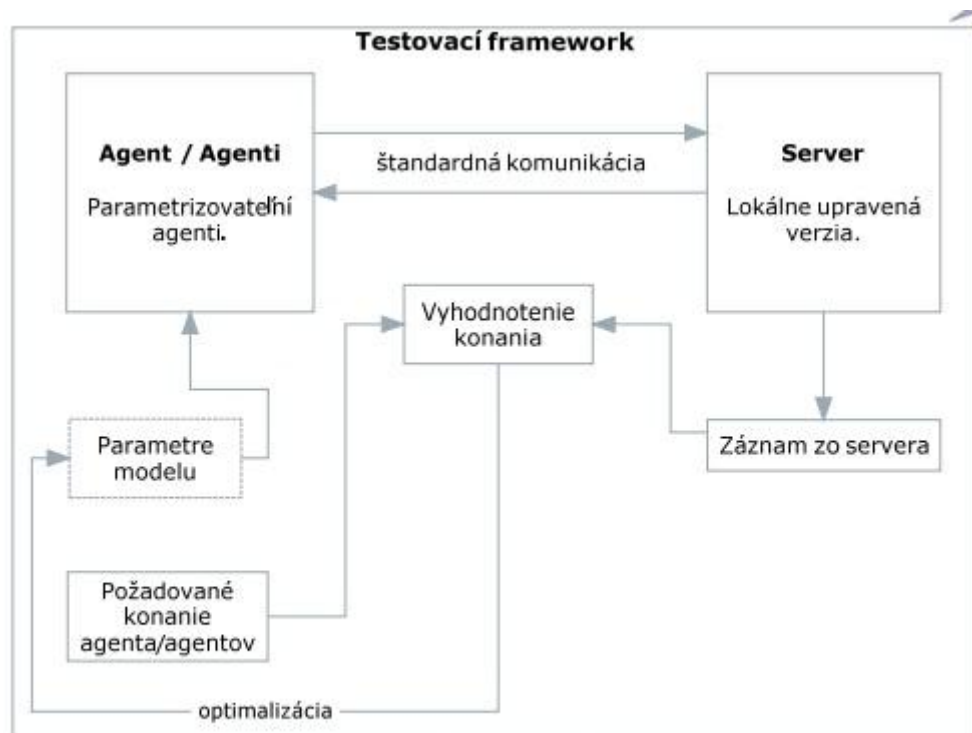


Testovací framework

Z dôvodu zrýchlenia simulácie zasiahli do testovacieho frameworku a tak docielili efektívnejšie testovanie agenta.

Architektúra testovacieho frameworku

Nakoľko požadovaná funkcionálnosť pracuje na úrovni nad samotným agentom, testovací framework musí umožňovať vloženie agenta do systému. To platí v prípade testovania schopností nižšej úrovne. V prípade testovania vyššej logiky, musí framework umožňovať vloženie viacerých agentov do systému. Testovanie agenta vyžaduje špecifikovanie očakávaného výsledku. Následne po uskutočnení simulácie sa pomocou záznamu o zápase vyhodnotí, či výsledok dosiahol požadovanú úroveň. Ak nie, framework umožní technikami umelej inteligencie hľadať lepšie nastavenie parametrov. Vzhľadom na povahu problému bude vhodné použiť genetické algoritmy. Na obrázku je architektúra systému znázornená diagramom.



Architektúra testovacieho frameworku

V nasledujúcej časti sú opísané netriviálne bloky, ktoré sa nachádzajú na obrázku:

- **Parametre modelu.** Blok špecifikuje jednoducho meniteľné parametre agenta, resp. agentov, ktoré budú môcť byť ovplyvňované v rámci optimalizácie. Parametre modelu predstavujú v skutočnosti komunikačný protokol pre použité genetické algoritmy a agenta.
- **Záznam zo servera.** Záznam zo servera slúži na získanie informácií o simulácii. Informácie, ktoré sa týmto spôsobom získajú, je následne potrebné pretransformovať do opisu, ktorý bude kompatibilný s opisom v bloku Požadované konanie agenta/agentov.

- **Požadované konanie agenta/agentov.** Požadované konanie je blok, do ktorého zadáva informácie používateľ. Tieto informácie sú opisom očakávaných udalostí, ktoré sa v bloku Vyhodnotenie konania budú porovnávať s informáciami získanými zo Záznamu zo servera.
- **Vyhodnotenie konania.** V predchádzajúcich opisoch bol uvedený účel bloku Vyhodnotenie konania. Je potrebné uvedomiť si, že jeho zložitosť závisí od miery rozdielu medzi informáciami získanými od používateľa a informáciami, ktoré získava blok zo Záznamu zo servera. Nakoľko informácie zo servera nie sú rýdzo deterministické, je potrebné vytvoriť opis, ktorý bude vždy môcť uvažovať s vopred stanovenou mierou nepresnosti.

Prototyp

Do prototypu boli zahrnuté nasledujúce funkcie:

- **Implementácia testovacieho frameworku**
- **Pohyby agenta**
- **Úpravy v editore pohybov**

Bližšie sú opísané v analýze tímu Android na wiki:

<http://team17-11.ucebne.fiit.stuba.sk/wiki/Androids - TP 2010/2011>

Použité zdroje:

- Dokument: http://labss2.fiit.stuba.sk/TeamProject/2010/team05is-si/files/dok_prod_verFinal.pdf
- Prezentácia: <http://labss2.fiit.stuba.sk/TeamProject/2010/team05is-si/files/team05issi-technicka.ppt>