

Paplašināma daudzāģentu sistēmu modelēšanas vide

Ingars Ribners
Darba vadītājs: Guntis Arnicāns

2014.gada 10.decembris

Saturs

1. Darba tēma
2. Terminu
3. Īsi par daudzāģentu sistēmām (MAS)
 - Kas tās ir?
 - Vēsture
 - Piemēri
4. Darbu plāns. Kas ir izdarīts?
5. Jautājumi

Darba tēma

Promocijas darba ietvaros paredzēts izveidot:

- paplašināmu formālu valodu, kas nodrošinātu iespēju aprakstīt daudzāģentu paradigmā veidotus modeļus un attīstības algoritmus to aģentiem;
- modelēšanas vidi, kura nodrošina modeļu interpretēšanu, kas pierakstīti šajā valodā.

Izveidotā valoda un modelēšanas vide nodrošinās iespēju nozaru speciālistiem izstrādāt un testēt savās nozarēs specifiskus algoritmus kā atsevišķiem aģentiem, tā arī sadarbības algoritmus aģentu grupām.

Saturs

1. Darba tēma
2. Terminu
3. Īsi par daudzāģentu sistēmām (MAS)
 - Kas tās ir?
 - Vēsture
 - Piemēri
4. Darbu plāns. Kas ir izdarīts?
5. Jautājumi

Termini

Modelis – noteiktam pielietojumam izveidots reālās pasaules procesa/parādības vienkāršojums. Modelī ir iekļauts galīgs skaits procesa/parādības pielietojumam būtisko īpašību, pārējās atmetot (bezgalīgs skaits);

Modelēšana – reāla procesa/parādības modeļa izveidošana un aprakstīšana formālā vai neformālā valodā, jeb, citiem vārdiem, subjektam vajadzīgās (galīgas) “bildītes izgriešana” no visas (bezgalīgi daudzpusīgās) apkārtnes.

Modelēšana ir viena no galvenajām funkcijām, ar ko nodarbojas dzīvas būtnes apziņa (taj skaitā, intelekts + viss pārējais)

Termini

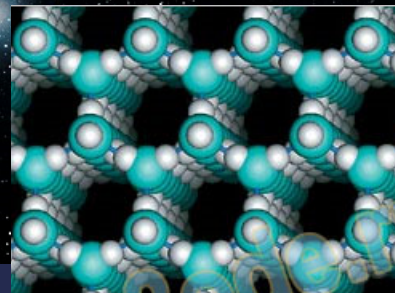
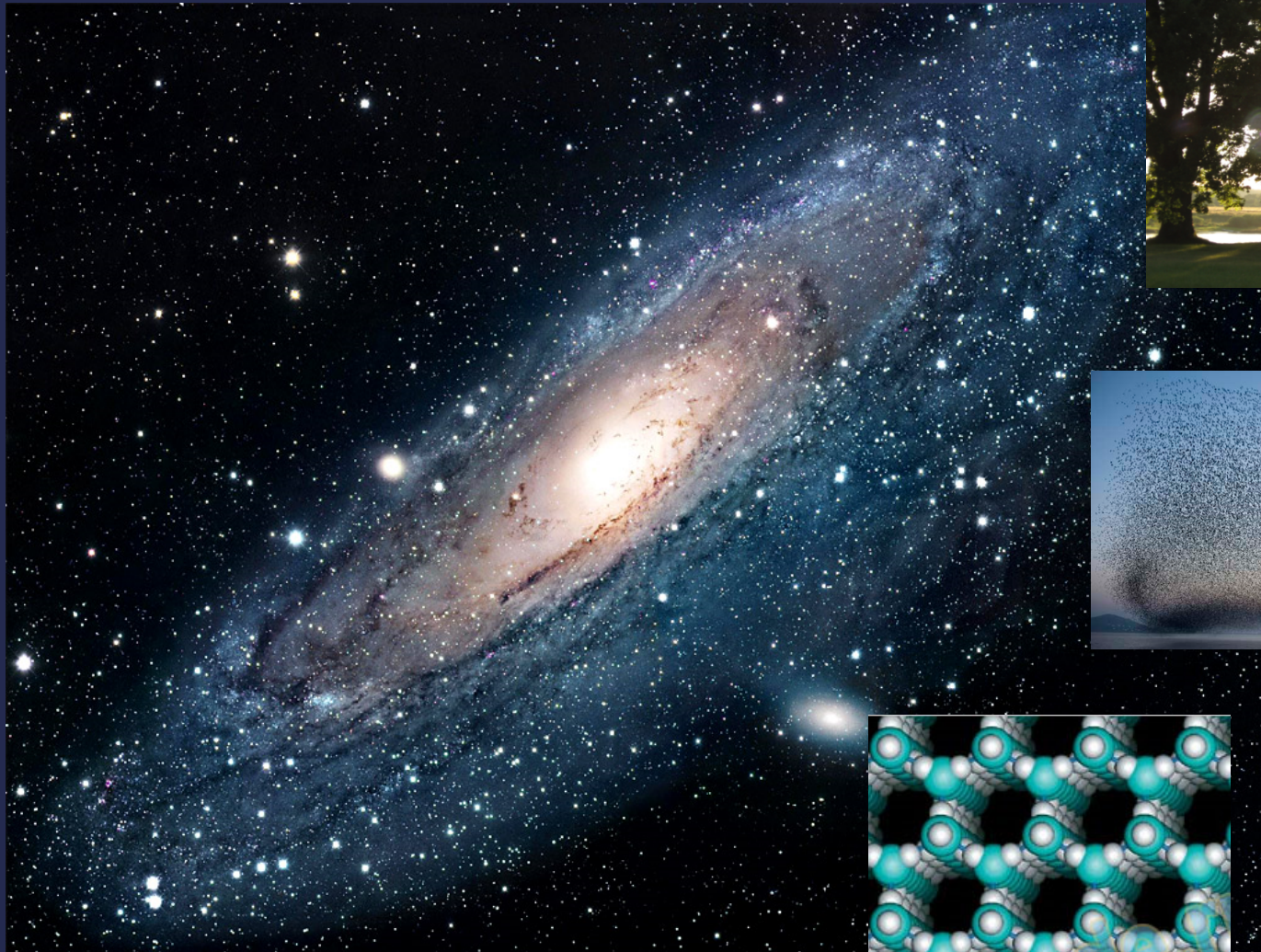
Modeļa interpretācija (“izpilde”) – formālā valodā aprakstīta modeļa apraksta “ielaišana” modelēšanas vidē, kurā modelis atbilstoši tā aprakstam tiek izveidots un sāk “dzīvot”, t.i.:

- “uztvert” apkārtējo vidi;
- reaģēt uz to, atbilstoši saviem algoritmiem un “principiem”.

Modelēšanas vide – “virtuāla mašīna”, kurā formālā valodā aprakstīts sistēmas modelis tiek interpretēts (“izpildīts”).

Modelis daudzāģentu paradigmā – sastāv no liela daudzuma vienkāršiem un vienveidīgiem aģentiem;

Procesi un parādības dabā



Procesi un parādības dabā

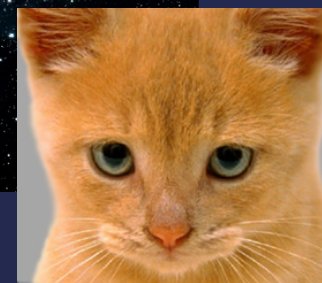
Var novērot, ka jebkuru dabā sastopamu procesu vai parādību vispilnīgāk var aplūkot, ja pieņem, ka tas ir

darītājs (=agents, =aktors)

un, vienlaicīgi, ir arī

darītāju kopums

kurš sastāv no ietvertajiem darītājiem un kurš «dzīvo» to aptverošajos darītājos un blakus peer-darītājiem (kuri veido tā vidi) un kuri, savukārt, atrodas tos aptverošajos darītājos, atrodas dabā, Visumā utt... :)

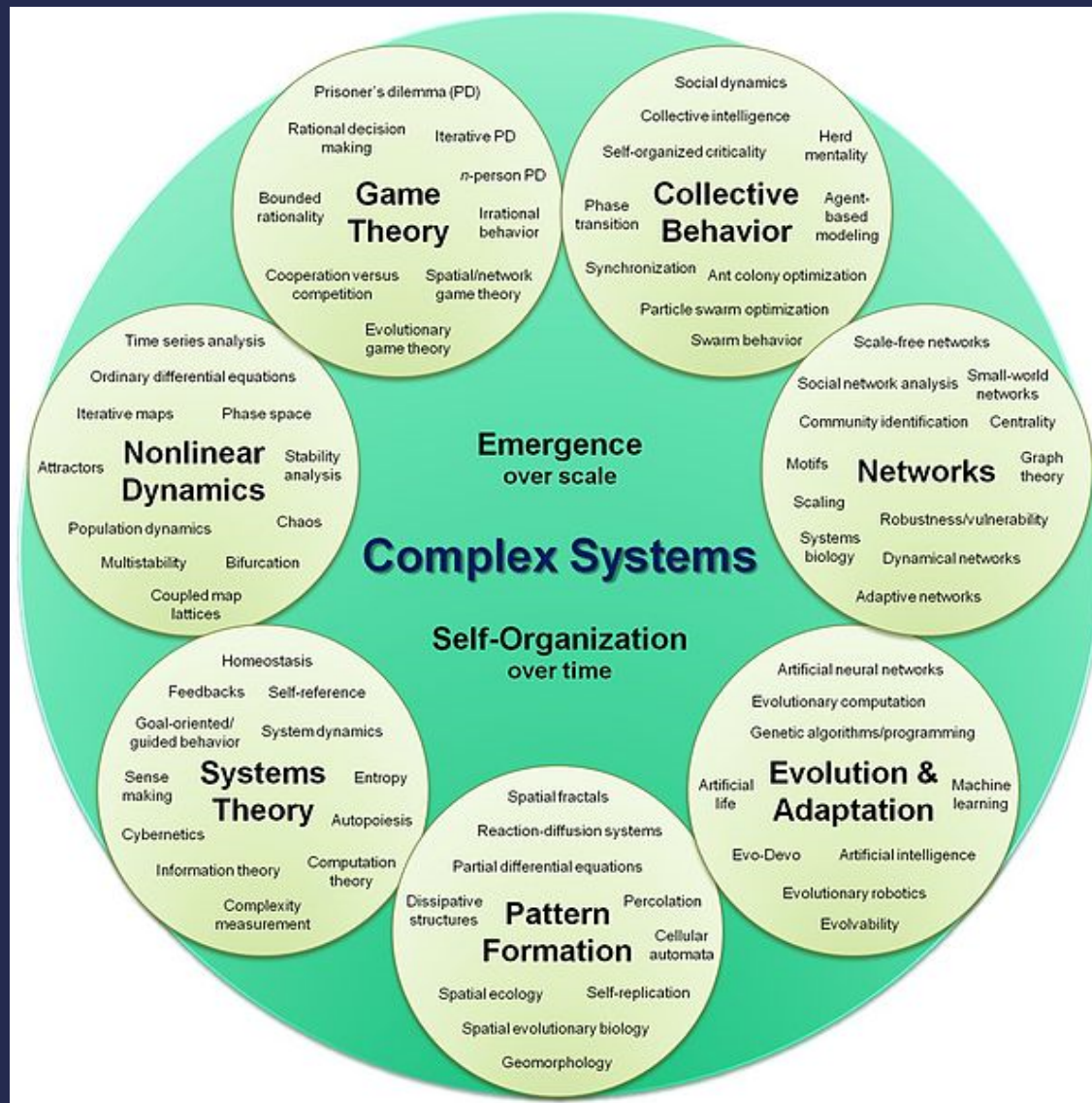


Līdz ar to, ja vēlamies dabā pastāvošu procesu vai parādību modelēt pēc iespējas adekvāti, tad mūsu modelēšanas videi jābūt maksimāli tuvai mūsu «dabīgajam interpretatoram», t.i., tai jāietver visas būtiskās «interpretatora» īpašības.

Termini

- *Aġents* (šeit = “actor”, darītājs) – ikviens aktīvs elements pasaulē, tas, kurš modelē un rīkojas savā apkārtējā vidē. Aġentam ir subjekta īpašības, tas ir autonomš, tam var būt sava atmiņa, tas var sadarboties ar citiem aġentiem, kā arī ar apkārtējo vidi;
- Galvenā aġenta atšķirība no pasīva objekta, klases vai datu entītes:
 - Aġents var būt «dzīvs», t.i. var «atmosties» un reaģēt uz signālu no ārpuses, uz notikumiem, un kaut ko izdarīt vai neizdarīt (*to act*), t.sk. uztvert vai ietekmēt apkārtējo vidi un citus aġentus;
 - Aġentam var būt mērķis;
 - Aġents vienmēr atrodas noteiktā apkārtējā vidē (aptverošajā aġentā) un uz to iedarbojas materiālās vides ierobežojumi. Kā viens no vides ierobežojumiem var būt vietas «lokālums» (2D, 3D u.c. telpā)

Kompleksās sistēmas



Kompleksās sistēmas

Būtība: Par *kompleksu sistēmu* (priekš subjekta) sauc sistēmu, kuru dotais subjekts vada (kontrolē) tikai daļēji.

Bez tam, var būt citi subjekti, kuri veic (mēģina veikt) šīs pašas sistēmas vadīšanu (kontroli).

Kompleksas ir lielākā daļa reālās pasaules sistēmu, kā dabā un jebkurā cilvēku veidotā nozarē. Ne-kompleksas sistēmas ļoti neliela klase, kas veidotas laboratorijas apstākļos, kur visa vide pilnībā tiek kontrolēta. Parasti ar tām nodarbojas eksakto zinātņu pārstāvji.

Kompleksas sistēmas vāji iespējams aprakstīt ar formālām metodēm.

Daudzaģentu paradigma (man tā šķiet) ir labākā no zināmajām komplekso sistēmu modelēšanas pieejām.

Kas ir daudzāģentu sistēmas?

Daudzāģentu sistēma ir sistēma, kas uzbūvēta daudzāģentu paradigmā – sastāv no liela daudzuma vienkāršiem un vienveidīgiem aģentiem;

Pētījumi par daudzāģentu sistēmām

1940-tie: John Von Neumann (1903-1957) – pašreplīcējoša mašīna, Stanislaw Ulam (1909-1984) – šūnu automāti (*cellular automata*)



1969: John Conway (1937) – šūnu automātu speciālgadījums «dzīvības spēle» (*The Game of Life*)



1971: Thomas Schelling (1921) – rasu segregācijas modeļi



~1980: Robert Axelrod (1943) – sadarbības izcelšanās pētījumi (programmu sacensības «vairākkārtējā cietumnieku dilemmā» u.c.)

~1988: Craig Reynolds (1953) – bara (*flock*) izturēšanās algoritmi



u.c.

Daudzaģentu sistēmu vēsture

- 1940-ie – Ideja par sistēmu modelēšanu, izmantojot daudzāģentu paradigmu (kuras ietvaros sistēma tiek apskatīta kā liels vienkāršu un vienveidīgu aģentu kopums) pirmo reizi tika izteikta 1940-to gadu beigās (Stanislav Ulam – šūnu automātu ideja (uz papīra); John von Neumann – arhitektūras apraksts mašīnai, kura spēj reproducēt pati sevi). Tomēr, tā kā šāda veida sistēmu modelēšanai nepieciešami diezgan lieli skaitļošanas resursi un nopietniem pielietojumiem - paralēlisma realizācija “dzelzī”, tad praktiska daudzāģentu modelēšanas instrumentu attīstība sākās tikai pēc 1990.gada;
- 1970. gadā John Conway uzkonstruēja pazīstamo šūnu automātu “Game of Life”, kurā virtuālā 2-dimensiju telpā tika modelēta “dzīvu organismu” kolonijas attīstība laikā. Kolonija sastāv no liela daudzuma vienādām “šūniņām”, katras no kurām algoritmi bija ļoti primitīvi;

Daudzaģentu sistēmu vēsture

- 1971 – Viens no agrākajiem daudzģentu modeļu pielietojumiem bija rasu segregāciju modelis (Thomas Shelling, 1971);
- 1984. gadā tika sarīkotas sacensības par labāko stratēģiju “cietumnieku dilemmas” risināšanā, kurā risinājumi tika pārbaudīti vidē, izmantojot daudzģentu pieeju (Robert Axelrod);
- 1986. gadā pirmo reizi tika modelēti bioloģiski procesi ar sociuma raksturiezīmēm, piemēram, putnu kāša modelis (Craig Reynolds, flocking model);

Daudzaģentu sistēmu vēsture

- 1991 – Vārda “aģents” pirmo lietojumu mūs interesējošā nozīmē noteikt nav vienkārši. Kā viens no kandidātiem ir raksts “Artificial Adaptive Agents in Economic Theory” (John Holland, John Miller);
- 1990-ie – Pirmie daudzģentu sistēmu izstrādes rīki programmētājiem tika izstrādāti 1990-tajos gados (1990 – StarLogo; 1995 - NetLogo, SWARM; 1998 – MAML u.c.);
- 2000-ie – pēc daudzģentu programmēšanas rīku parādīšanās – aktīvi notiek daudzģentu pielietojumu algoritmu attīstība nozarēs (fizikā, ķīmijā, bioloģijā, medicīnā, sabiedrības mācībā, apziņas mācībā, robotikā u.c.), kura turpinās arī šodien;
- Pēdējos 10-15 gadus notiek attīstība arī sistēmu modelēšanas metodoloģiju izstrādē, izmantojot daudzģentu paradigmu. Vienota pieeja modelēšanas metodoloģijās pagaidām gan vēl nav izkristalizējusies.

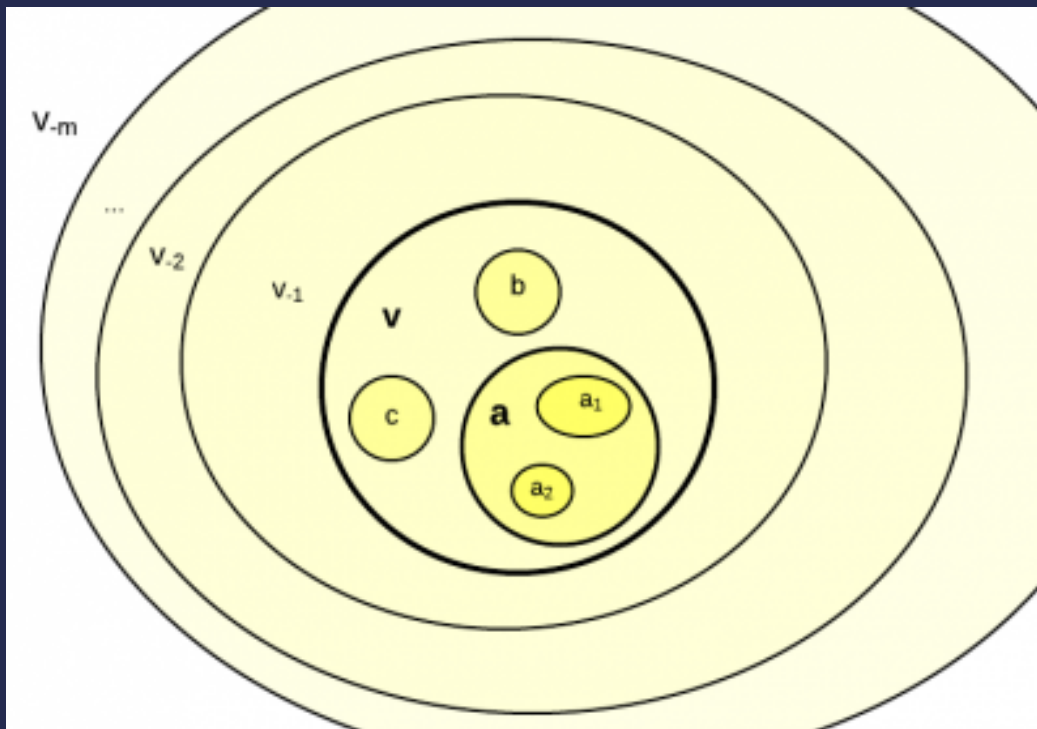
Paralēlisms datortehnikā

- Pēdējos gados datortehnikas jaudas palielināšana notiek vairs ne tik daudz paaugstinot taksts frekvenci, bet gan realizējot vienlaicīgas apstrādes iespējas. Piemēram, daudzkodolu procesori (Intel Xeon Phi, 60-kodolu processor), datoru klāsteri;
- Valodu/instrumentu atbalsts. Tehnoloģija Erlang tika izstrādāta Ericsson laboratorijā 1980-to gadu beigās (Joe Armstrong un citi, 1987). Erlang ietver sevī kā virtuālo mašīnu (BEAM), tā arī programmēšanas valodu (Erlang/OTP). Erlang programmas arhitektūra ir neatkarīgu procesu (aģentu) kopums, kuri apmainās ziņojumiem un sadarbojas virtuālajā mašīnā. Kopš 1998 – open source;
- Bibliotēkas darbam ar paralēliem procesiem un “aktoriem” (pēc Erlang piemēra) pēdējos gados tiek izstrādātas visās galvenajās programmēšanas valodās;

Darbu plāns

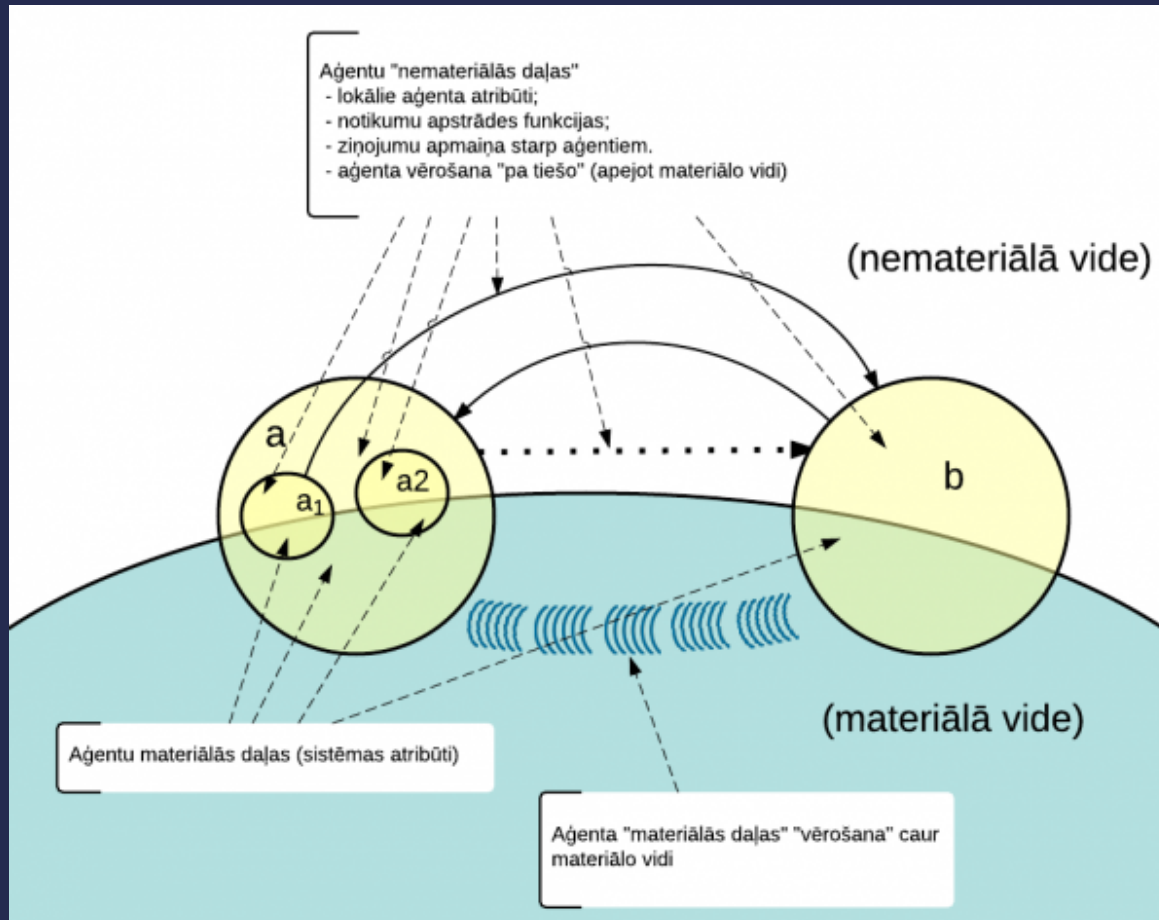
Izpazīšanās ar daudzāģentu modelēšanas metodoloģijām un programmēšanas vidēm	✓
Modelēšanas vides arhitektūra	✓
Modelēšanas valoda(s)	•
Lietotāja saskarnes (text, web)	
Standartbibliotēkas un sistēmas servisi	•
API ārējām sistēmām	
Modelēšanas vides pielietojumi	•
Disertācijas teksts	

Modelēšanas vides arhitektūra



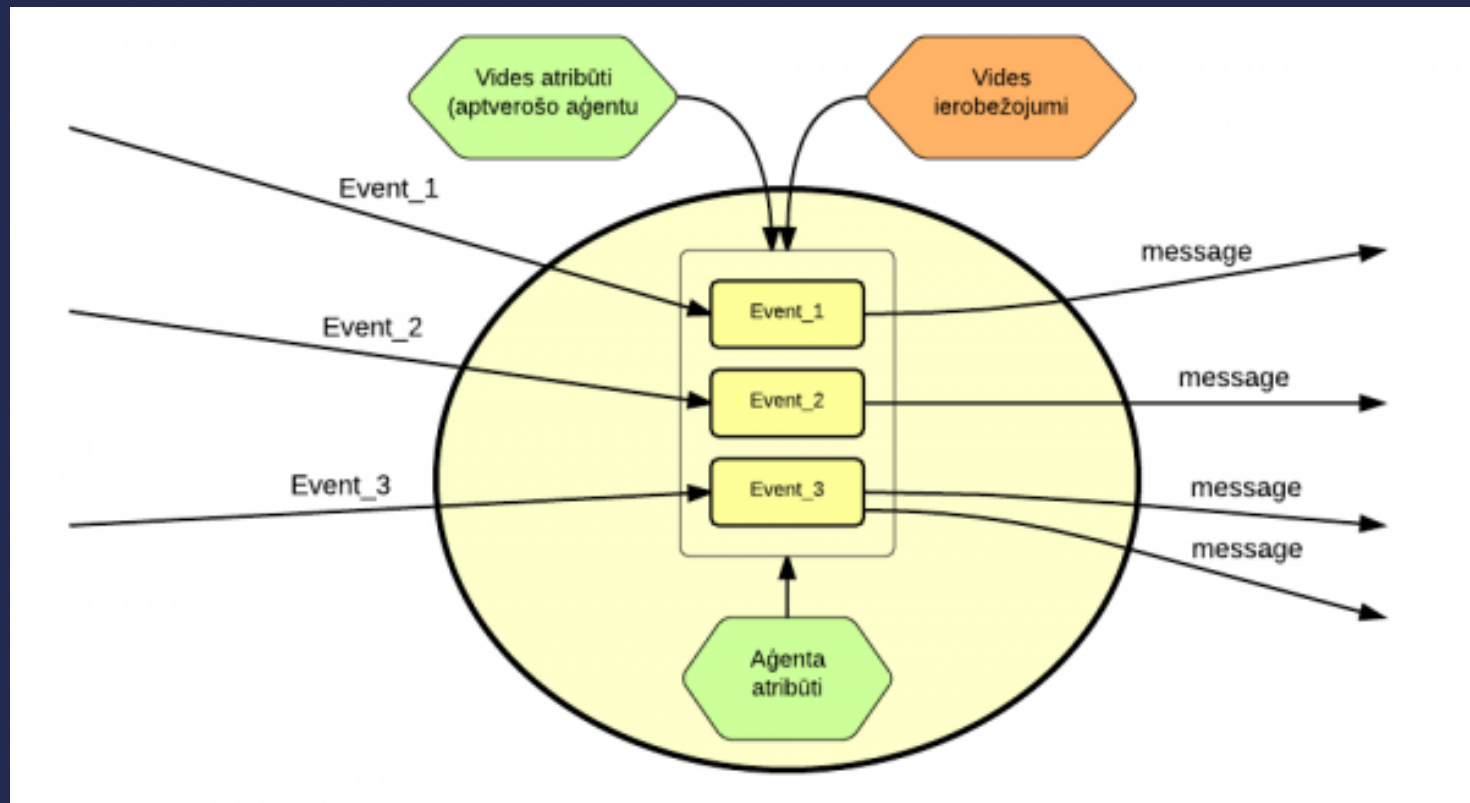
Aģentu hierarhijas piemērs

Modelēšanas vides arhitektūra



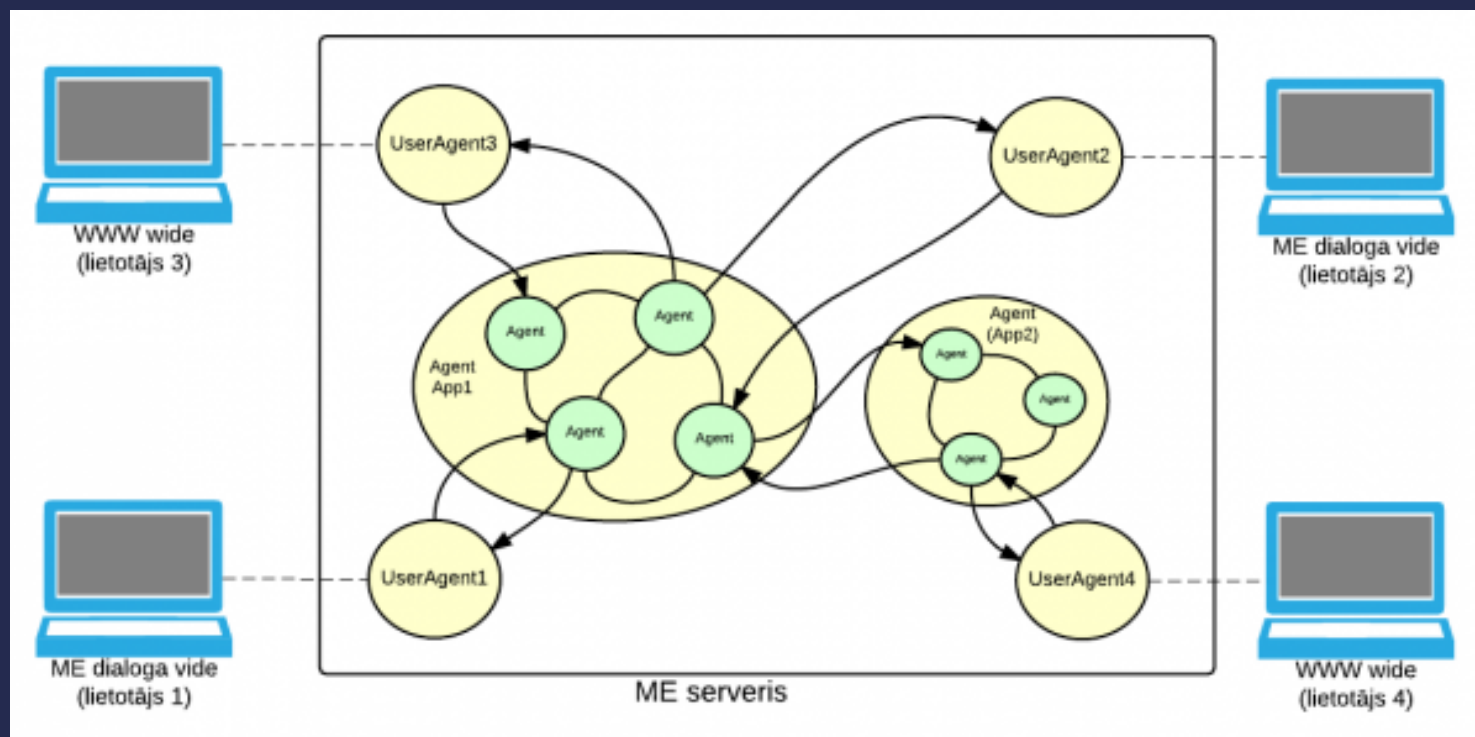
Aģenta materiālā un nemateriālā daļa / mijiedarbības veidi

Modelēšanas vides arhitektūra



Aģenta arhitektūra

Modelēšanas vides arhitektūra



Lietotāju-sistēmas sadarbība

Modelēšanas valoda(s)

Aģenta reakciju uz notikumiem (behaviour) aprakstīšanas valoda (A-L)	•
Valodas konstrukcijas aģenta zināšanu un principu aprakstīšanai (AKP-L)	•
Valodas konstrukcijas valodas paplašināšanai (metaprogrammēšana, DSL) (A-MPL)	•

Lietotāja saskarnes

Lietotāja dialoga vide un dialoga valoda darbam ar modelēšanas vidi klienta-servera režīmā (teksta saskarne, līdzīga Oracle SQL*Plus vai MySQL teksta videi)	
Grafiska “vadības pults” darbam ar sistēmu (html / js / svg)	

Standartbibliotēkas un sistēmas servisi

Aģentu manipulāciju primitīvi (A-QL) - komandas darbam ar atsevišķiem aģentiem (izveidošana, dzēšana u.c.)	•
Daudzaģentu sistēmu manipulāciju primitīvi (MAS-QL) – komandas darbam ar sistēmām un tās ierobežojumiem (izveidošana, dzēšana, saglabāšana, nolasīšana u.c.)	•
Aģenta loģikas aprakstīšanas valodā izmantojamās standartfunkciju bibliotēkas	•
Darbs ar lieliem aģenta atribūtiem (faktiski, datubāzēm)	
Neironu tīklu izmantošana aģentu pašvēstures pierakstam un apmācībai	

Darbu plāns

Modelēšanas vides arhitektūra	✓
Modelēšanas valoda(s)	•
Lietotāja saskarnes (text, web)	
Standartbibliotēkas un sistēmas servisi	•
API ārējām sistēmām	
Modelēšanas vides pielietojumi	•
Disertācijas teksts	

Jautājumi?

Darba novitāte

- Modelēšanas valoda nav piesaistīta konkrētai pielietojuma jomai (domēnam), bet valodā ir realizēti rīki, kas ļauj lietotājiem pašiem veidot jomai specifiskus modelēšanas primitīvus (DSM) un valodas (DSL), ar ko aprakstīt šo primitīvu reakciju uz notikumiem;
- Modelēšanas vides realizācija ar “klienta-servera” arhitektūru, kur “serveris” var būt gan atsevišķs dators, gan klasteris, kas sastāv no vairākiem datoriem. Tiek izmantotas šodienas tehnoloģijas, kas ļauj modeļus izpildīt daudzkodolu un klasteru arhitektūrās;
- Modelēšanas vides arhitektūra ir modulāra, saskarnes starp moduļiem ir formāli aprakstītas (modeļi un to interpretācijas vide ir nodalīta no modeļa vizualizācijas mehānismiem, no lietotāja saskarnēm, no perifēro iekārtu saskarnēm, no modeļu transformācijas rīkiem uz citu veidu modeļiem u.c.)