



AGENTŲ PARADIGMA

Medžiagą parengė: Doc. dr. Agnė Paulauskaitė-Tarasevičienė

Agentinė paradigma

Agentinės technologijos yra viena sparčiausiai besivystančių informacinių technologijų tyrimų krypčių. Jų taikymo sritys taip pat nuolat plečiasi:

- kompiuteriniai žaidimai
- “išmaniosios” aplinkos sistemos,
- telekomunikacijos,
- skrydžių valdymas,
- verslo procesų inžinerija,
- duomenų gavyba,
- informacijos paieškos,
- e.prekybos,
- edukologija,
- duomenų bazių ir kitos sistemos.



Agento apibrėžimas

Terminas „agentas” labai platus. Per pastaruosius metus buvo pasiūlyta nemažai agento sąvokos apibrėžimų, tačiau bendro apibrėžimo iki dabar nėra.

To priežastys gali būti dvi:

1. agentų tyrinėtojai vis dar skirtingai suvokia šią sąvoką,
2. nors ir labiausiai artima programinės įrangos sričiai, agento sąvoka apima įvairias tyrimų ir taikymų sritis.



Agentų apibrėžimai

- **Agentas** – programinės įrangos arba techninės įrangos komponentas, kuris sugeba veikti tiksliai siekiant įvykdyti užduotis, kurias jam pateikia vartotojas.

H. S. Nwana

- **Agentas** – esybė, suvokianti aplinką savo jutikliais ir daranti poveikį tai aplinkai savo manipulatoriais.

Russell

- **Agentai** – tai kompiuterinės programos, kurios imituoja žmogiškuosius ryšius, darydamos tai, ką kitas žmogus galėtų padaryti už tave.

T. A. Selker

Agentų paplitimas

Istorija

- *Dirbtinis Intelektas.*
- Su interneto išplitimu ir kompiuterinių žaidimų paplitimu, agentų sąvoka tapo dar plačiau naudojama:
 - elektroninės prekybos,
 - asistavimo sistemos,
 - paieškos įrankiai,
 - socialiniai portalai,
 - kaip sudedamosios dalys didesnių programinės įrangos projektų.



AGENTŲ SAVYBĖS

Bendros savybės

Agento metafora yra daug apimanti, ji taikoma įvairiose situacijose ir skirtingiems tikslams pasiekti. Todėl dažnai tikslaus apibrėžimo suformulavimas keičiamas agento savybių apibūdinimu. Skiriamos šios tipinės agento savybės:

1. AUTONOMIŠKUMAS,
2. REAGAVIMAS,
3. PROAKTYVUMAS,
4. SOCIALUMAS/KOMUNIKAVIMAS.

1. Autonomiškumas

Autonomiškumas yra būtina agentinės sistemos savybė.

Agentas yra autonomiškas, jei:

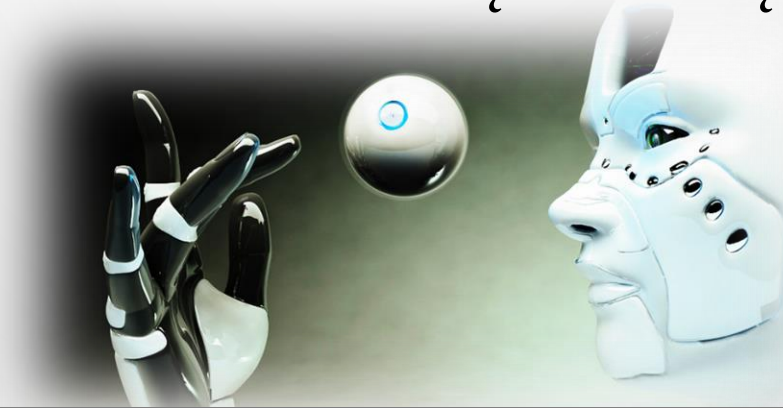
- funkcionuoja be komandų ir nurodymų iš aplinkos,
- agentas kontroliuoja išteklius ir turi įgaliojimus veiksams vykdyti,
- agentinei sistemai nereikia, kad kiekvienas jos žingsnis būtų patvirtintas (naudotojų ar kitų agentų), ji savarankiškai veikia ir sprendžia problemas.

2. Reagavimas

Reagavimas yra būtina agentinės sistemos savybė.

Agentas yra reaguojantis, jei:

- reaguoja į aplinkos pokyčius,
- automatiškai stebi aplinką, gauna apie ją informaciją, identifikuoja pokyčius, galbūt nustato tų pokyčių priežastis ir imasi adekvačių veiksmų.



Todėl agentas privalo turėti reikiamas sudedamąsias dalis: jutiklius, valdiklius, sąsają ir pan.

3. Proaktyvumas

- **Agentas yra proaktyvus** (kitai sakant, siekiantis savo tikslų) jeigu jis ne tik reaguoja į aplinkos pokyčius, bet ir imasi tam tikros iniciatyvos – sprendžia, **kaip** pasiekti tikslą.
- Kad galėtų imtis iniciatyvos, jis privalo turėti aiškiai apibrėžtą tikslą arba tikslų sistemą.

4. Socialumas

Sąveika būtina socialinėms organizacijoms formuoti.

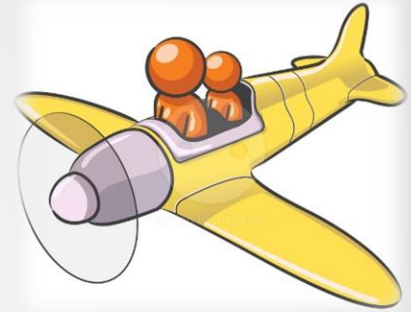
- Kad agentas įvykdytų užduotį, jam gali prireikti ne tik informacijos apie aplinką, bet ir aplinkos paslaugų.
- Sakysime, kad du ar daugiau agentų sąveikauja, jei abipusių veiksmų dėka tarp jų sukuriamas dinaminis ryšys.
- Keisdamiesi informacija, vykdydami tarpusavio įsipareigojimus, darydami vienas kitam įtaką, agentai tampa socialinėmis esybėmis, o jų organizacija įgyja papildomą funkcionalumą.



Pavyzdys. Lėktuvo pilotas

Tikslas: saugus nusileidimas aerouoste. Reikalinga:

- Pro-aktyvumas
 - Planuoti, kaip pasiekti tikslą,
 - Generuoti pagalbinius planus.
- Autonomiškumas
 - Vykdyti planus,
 - Išlaikyti esamą būseną,
 - Reaguoti į aplinkos pokyčius (oro sąlygos, nurodymai iš dispečerio ir t.t.)
- Reaktyvumas
 - Reaguoti laiku
- Socialumas
 - Autopilotas, siekdamas tikslų, bendradarbiauja su valdymo sistemomis.



DIRBTINIS INTELEKTAS IR AGENTAI

Dirbtinis intelektas ir agentai

Dirbtinio intelekto tyrinėtojai išskiria dvi agento sąvokos prasmės grupes:

- **Siauroji** agento sąvoka (*strong*);
- **Plačioji** agento sąvoka (*weak*).

Intelektualizuotas agentas yra toks, kuris nagrinėjamoje situacijoje pasirenka geriausią galimą veiksmą atsižvelgiant į kriterijus, pagal kuriuos vertinama agento sėkmė, atžvilgiu.



Agentas siaurąją prasme

Pagrindinių savybių sąrašas:

- Mentalinės savybės (žinios, tikėjimai,..)
- Mobilumas;
- Gyvybingumas;
- Teisingumas (agentai neskleidžia neteisingos informacijos);
- Racionalumas (agentai neveikia prieš savo tikslus);
- Prisitaikomumas;



AGENTŲ KLASIFIKACIJA

Agentų klasifikavimas pagal savybes

Kadangi agentinių savybių gali būti išskirta labai daug, tam tikra klasifikacija pagal tokias savybes yra tikslinga:

- *Mobilumą* (statinis ar mobilus);
- *Samprotavimo modelį* (svarstymo ar reaktyvųjį);
- *Svarbiausius atributus*: autonomiškumas, kooperacija, mokymąsis.

Pagal šias charakteristikas išskiriami 4 agentų tipai:

- Bendradarbiaujantieji;
- Bendradarbiaujantieji ir besimokantys;
- Sąsajos;
- Sumanieji.
- *Vaidmenis* (informacijos ar interneto);
- *Hibridines filosofijas*;
-

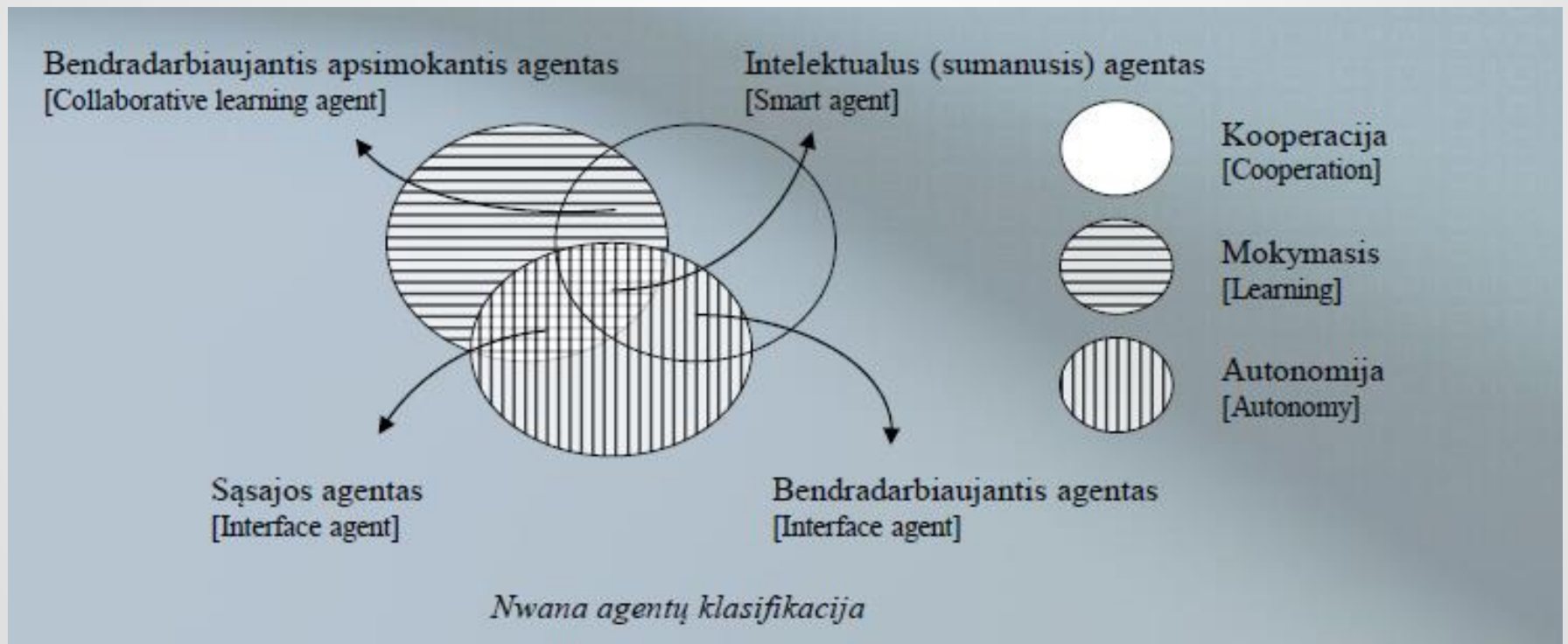
Brustoloni klasifikacija

Brustoloni išskyrė tris agentų klases:

- **Savireguliuojantis agentas.** Pats paprasčiausias agentas. Termostatas yra tokio agento pavyzdys. Šis agentas nieko neplanuoja ir neapsimoko, o tiesiog vykdo baigtinį sąrašą užduočių, esant tam tikrai aplinkos būsenai.
- **Planuojantis agentas.** Tai sudėtingesnis, jame naudojami tam tikri dirbtinio intelekto elementai.
- **Adaptyvus agentas.** Pats sudėtingiausias, nes apart sugebėjimo atlikti veiksmą ir planuoti, agentas mokosi iš sukauptos savo patirties.

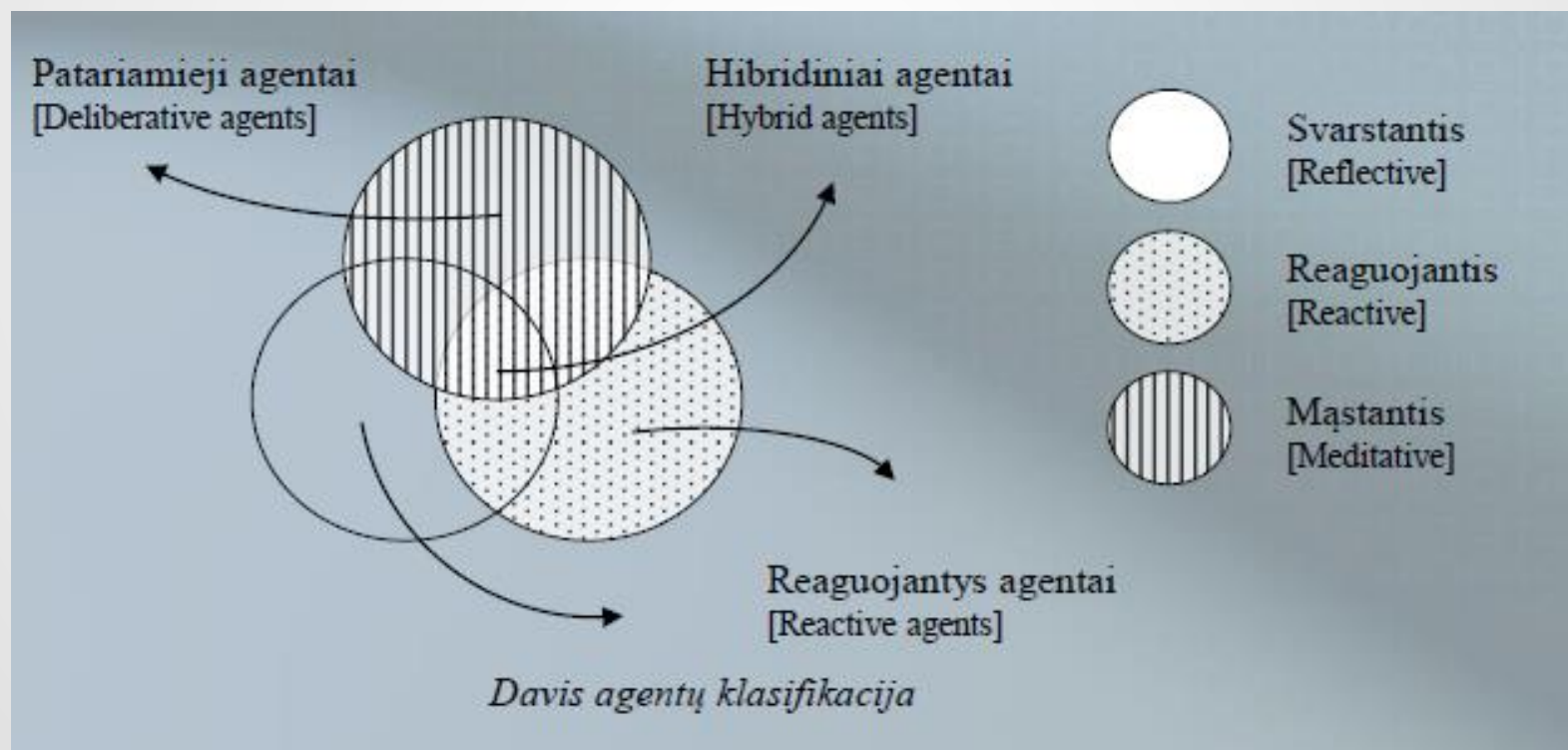
Nwana klasifikacija

- **Nwana** pasiūlė agentus klasifikuoti remiantis agento tipologija, t.y. klasifikavimas pagal tai, kaip stipriai agente išreikštos elgesio savybės: kooperacija, apsimokymas, autonomiškumas.



Davis klasifikacija

D.N. Davis pasiūlyta klasifikacija remiasi trimis intelektinėmis agento savybėmis ir kiek stipriai jos išreikštos agente. Šios intelektinės savybės: gebėjimas svarstyti, gebėjimas reaguoti, gebėjimas mąstyti.



Pagrindiniai Agentų tipai

1. Reaktyvieji agentai
2. Bendradarbiaujantieji agentai
3. Sąsajos agentai
4. Mobilūs agentai
5. Informaciniai agentai
6. Hibridiniai sumanieji



1. Reaktyvieji (reaguojantys) agentai

Savybės

- Agentas neturi sugebėjimo planuoti.
- Agentas neturi atminties.
- Gali atlikti kažkokį veiksmą iš savo baigtinio veiksmų sąrašo.
- Agentui nereikia apsimokyti ir reakcijos laikas priklauso tik nuo sensorių, kuriais agentas jaučia aplinką.
- Agentas gali būti pritaikomas nesudėtingiems uždaviniams spręsti realaus laiko sistemose.

Reaguojantys agentai. Pavyzdys

Braitenberg mašinos

Simboliai

Temperatūra,
šviesa,
.....

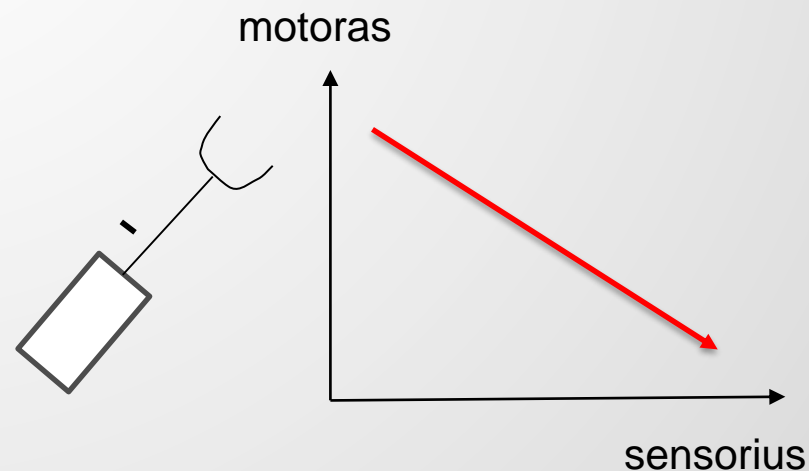
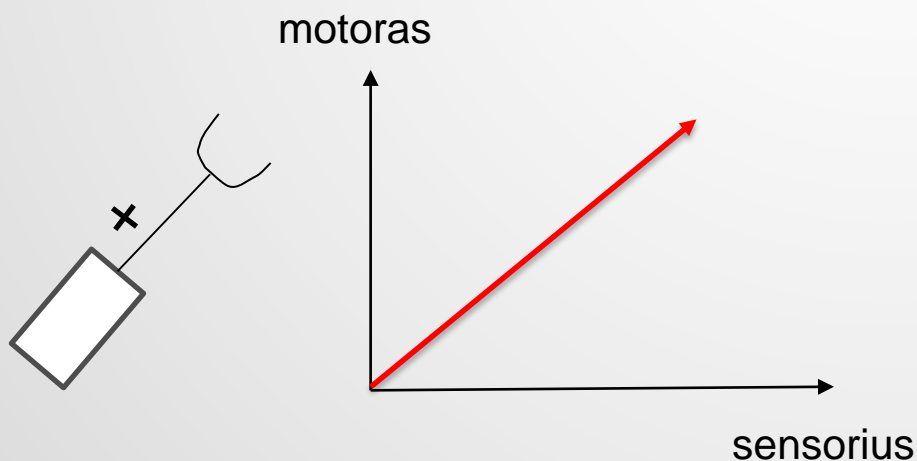


sensorius

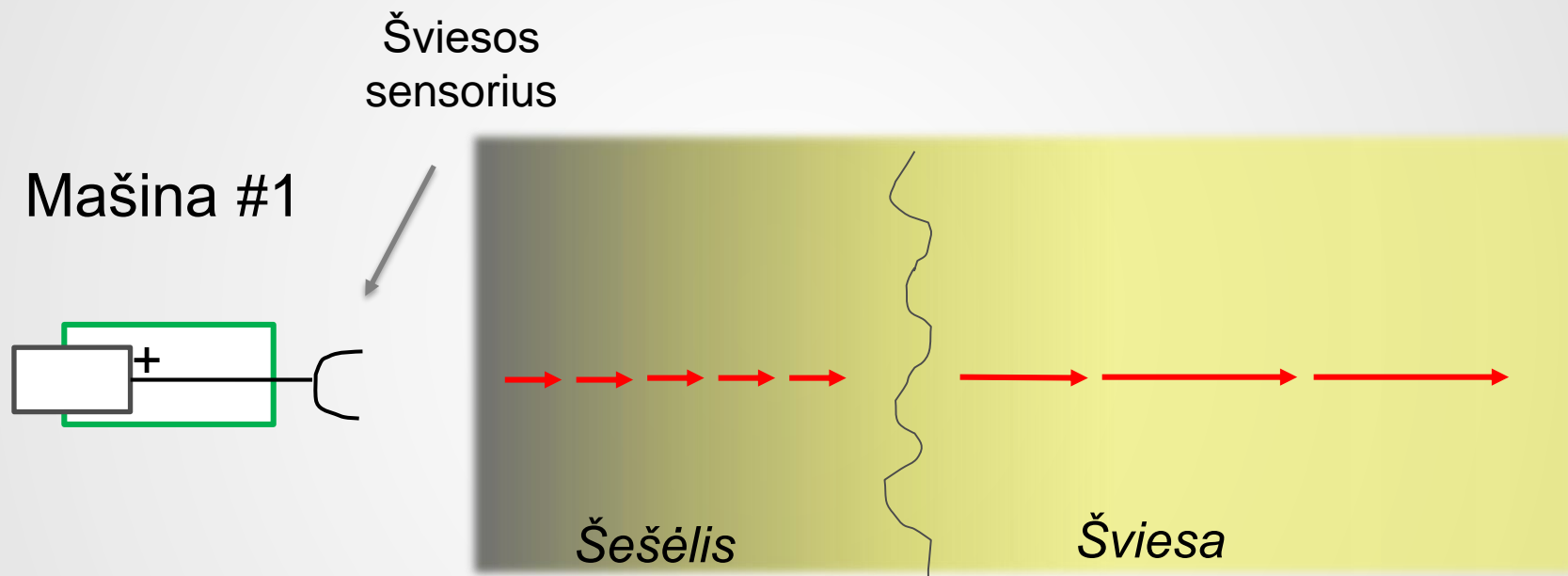


motoras

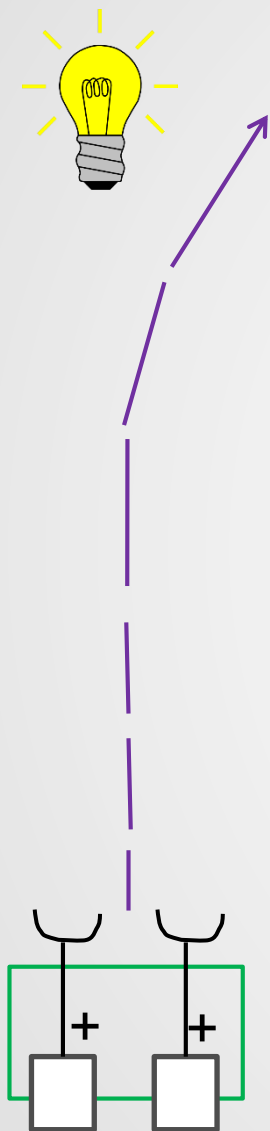
Ratai,
Propeleris,
....



Šviesos sekimo pavyzdys



Kaip judės mašinos?



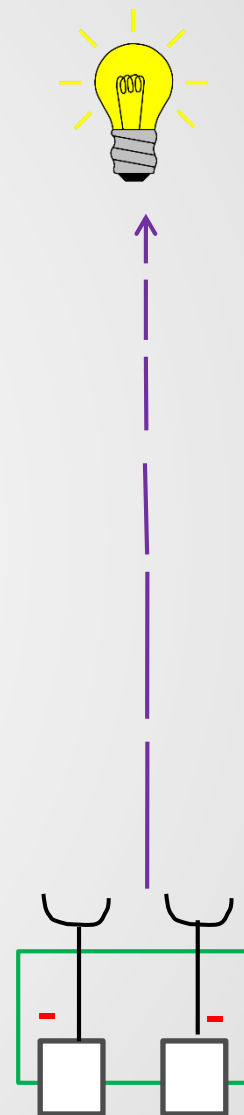
Mašina #2



Mašina #3



Mašina #4



Mašina #5

2. Bendradarbiaujantieji agentai

- Bendradarbiaujantieji agentai pasižymi autonomija ir bendradarbiavimu su kitais agentais tam kad pasiektu savo tikslus. Siekiant sukurti koordinuotą bendradarbiaujančių agentų grupę, jiems gali reikti derėtis norint gauti abipusius agentų sutikimus tam tikrais klausimais.
- Charakteristikos:
 - Autonomiškumas,
 - Socialiniai gebėjimai,
 - Jautrumas,
 - Iniciatyvumas.

3. Sąsajos agentai

Sąsajos agentas gali būti apibrėžiamas kaip programa, kuri gali paveikti objektus tiesiogiai manipuluojant sąsaja, bet be tiesioginių vartotojo nurodymų.

- Sąsajos agentas skaito įėjimą, kurį vartotojas paduoda sąsajai ir gali atlikti tam tikrų objektų, kuriuos vartotojas mano ekrane pakeitimus.
- Agentas gali stebėti visą eilę vartotojo įėjimų tam tikrą laiką prieš nusprendžiant atlikti tam tikrą veiksmą arba atvirkščiai vienas vartotojo įėjimas gali iššaukti seriją agento veiksmų.

3. Sąsajos agentai

Sąsajos agentai mokosi tam, kad suteikti geresnę pagalbą vartotojui. 4 pagrindiniai būdai:

1. Stebi ir imituoja vartotoją;
2. Gaudami teigiamus/negiamus atsiliepimus mokosi iš vartotojo;
3. Gaudami tikslias instrukcijas iš vartotojo atlieka veiksmus;
4. Klausdami kitų agentų patarimo.

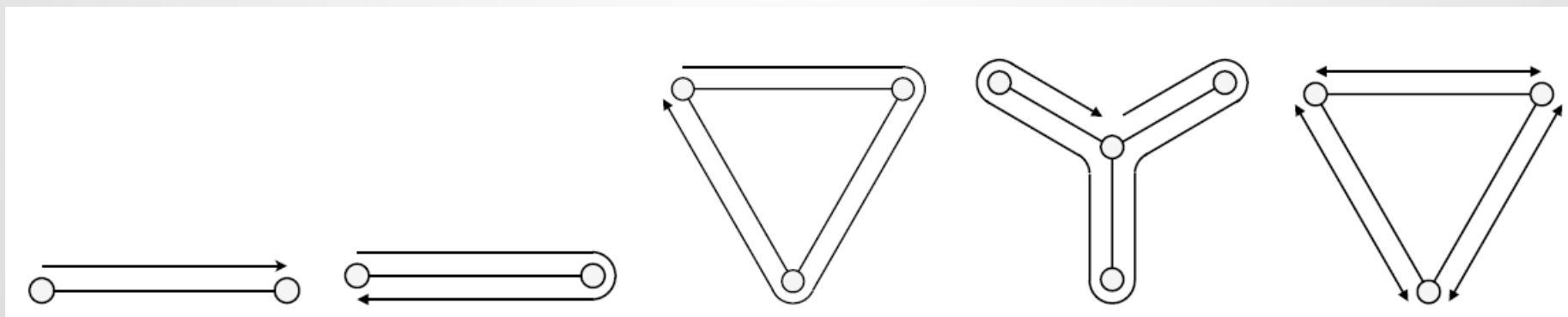


4. Mobilieji agentai

Savybės:

- Gebėjimas keliauti per tinklą;
- Bendrauti su kitais savininkais savo savininko vardu, rinkti informaciją ir grįžti namo atlikus numatytas užduotis;
- Autonomiškumas, bendradarbiavimas keičiantis, perduodant ar surenkant informaciją.

Migravimo būdai



Privalumai

- **Sumažina ryšio kaštus.**
- **Išsprendžia apribotų vietinių resursų problemą.**
- **Lengvesnis koordinavimas.** Vartotojas gali paleisti keletą mobiliųjų agentų, kurie gali atlikti nepriklausomas užduotis. Gražintus rezultatus apdoroja lokali programa.
- **Asinchroninis skaičiavimas.** Vartotojas gali paleisti mobiliuosius agentus ir tuo pačiu metu pakol jie dirba atlikti kitas užduotis..
- **Padidina paskirstytos skaičiavimo sistemos lankstumą.**

Agentų klonavimas

Agentas turėtų svarstyti klonavimo klausimą jeigu:

- Agentas negali laiku atlikti visų jam pavestų užduočių; taip pat negali jas deleguoti kitiems agentas **ir**
- Nėra mažai apkrauto agento, kuris galėtų atlikti užduotis (arba sub-užduotis po užduoties dekompozicijos) **ir**
- Yra pakankamai resursų agento klono sukūrimui (ar toje pačioje vietoje ar nuotolinėje) **ir**
- Klono agento ir originalaus agento laukiamas efektyvumas yra didesnis nei veikiant tik vienam originaliam agentui.



5. Informaciniai/interneto agentai

- Informaciniai agentai sukurti dėl priemonių, kurios turėtų padėti susitvarkyti su vis augančiais informacijos kiekiais stokos.
- **Informacijos agentas** yra savarankiškas, išmanus agentas, kuris turi prieigą prie vieno ar daugiau, nevienalyčių ir geografiškai paskirstytų informacijos šaltinių.
- **Informacijos sintezė ir pateikimas.** Agentas gali sujungti įvairairūšią informaciją ir pateikti vieningą nuomonę dėl atitinkamos informacijos vartotojui.

6. Hibridiniai agentai

Hibridiniai agentai apima kelių agentų savybes.

- Priklausomai nuo to, kokios savybės hibridiniame agente bus išreikštos labiausiai, agentas pasižymės atitinkamu elgesiu.
- Hibridiniame agente panaudotos kelių agentų savybės išreiškiamos tam tikra hierarchine struktūra. Pvz., agente, kuriame išreiškiamos dvi savybės yra du sluoksniai.

Vienas sluoksnis su kitu gali jungtis dviem būdais:

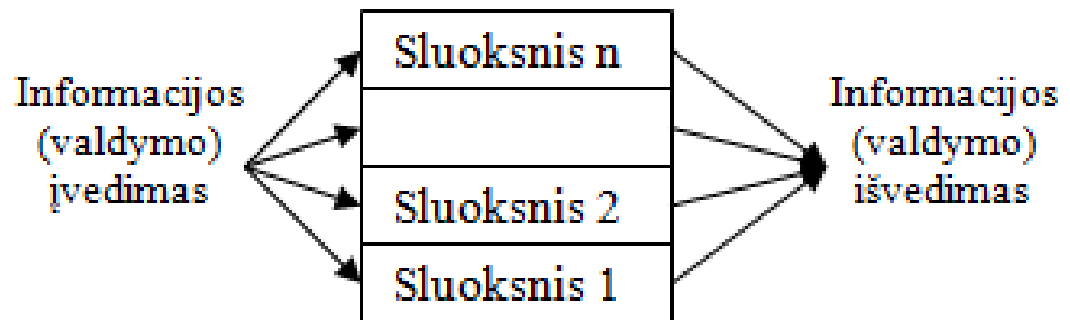
- *horizontaliai*,
- *vertikaliai*.

Horizontali struktūra

Jeigu reikalingas agentas su n skirtingų elgesio modelių, vadinasi agentas gali būti sudarytas iš n horizontaliai išdėstytų sluoksnių.

Trūkumas:

kiekvienas sluoksnis pateikia savo išvadas (pasiūlymus) dėl to sunku nuspėti tokio agento elgesį, kas riboja pritaikymo galimybes. Norint to išvengti į struktūrą yra įtraukiamas elementas kuris nustato, kuris sluoksnis yra pagrindinis, (priimančias galutines išvadas, sprendimus).



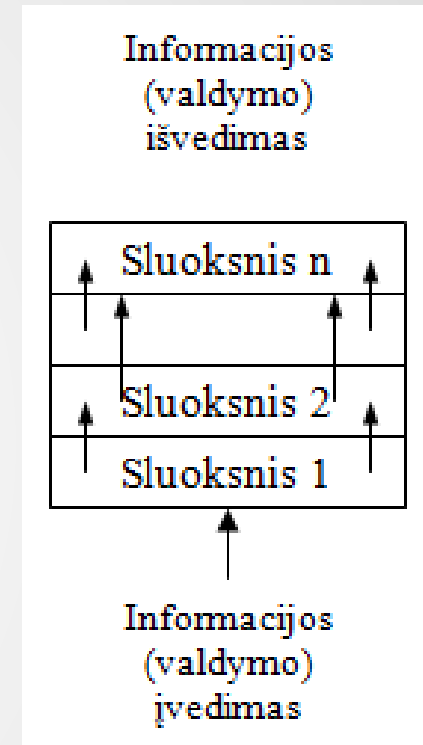
Vertikali struktūra (vienos krypties)

Vertikalus sluoksnių išdėstymas turi dvi modifikacijas:

- vienos krypties sluoksnių išdėstymas,
- dvi-kryptis sluoksnių išdėstymas.

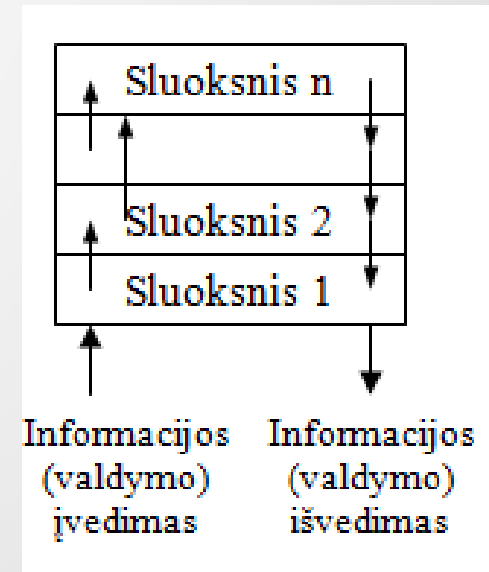
Vienos krypties vertikalios sluoksnių išdėstymo schema:

- Iš sensorių gauta informacija patenka į pirmą sluoksnį. Apdorota informacija keliauja į sekantį sluoksnį ir taip toliau. Paskutinis sluoksnis išveda rezultatą (pvz. kokį veiksmą agentui reikia atlikti).



Vertikali struktūra (dviejų krypčių)

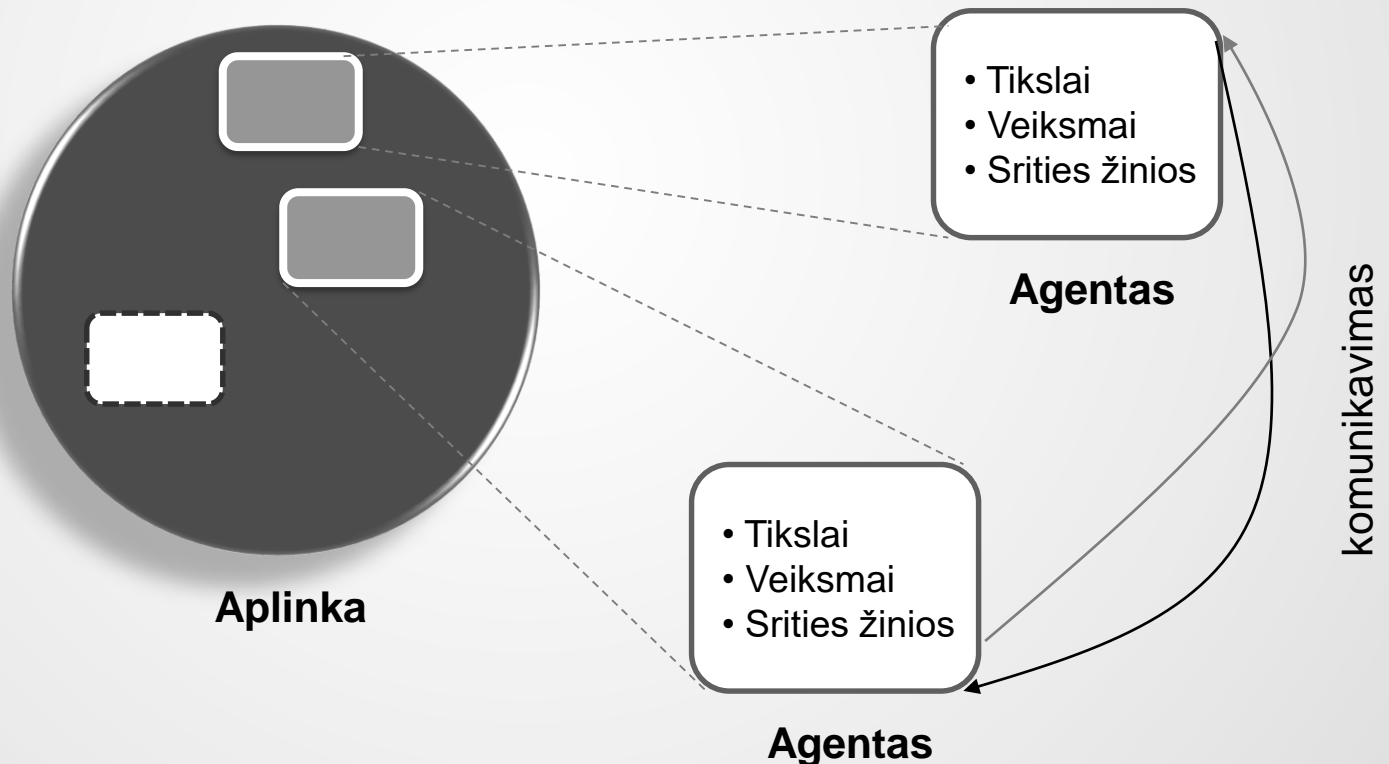
- Dvipusės krypties vertikalios sluoksnių išdėstymo schema veikia panašiai kaip kompanijos organizacinė struktūra, t.y. informacija ateina iš žemiausių grandžių ir kyla iki aukščiausių vadovų, o pastarieji daro sprendimus ir juos teikia žemesnės grandies darbuotojams.
- Kadangi kiekvieno sluoksnio išvada (sprendimas) pereina per kitus sluoksnius, vadinasi jei atitinkamo sluoksnio sprendimas buvo neteisingas, visa sistema veiks neteisingai.



DAUGIA-AGENTINĖ (MULTI-AGENTINĖ) SISTEMA

Multi-agentinės sistemos

- Sistema, kurią sudaro grupė agentų, potencialiai galinčių sąveikauti vienas su kitu vadinama multi-agentine sistema (*liet.* Daugia-agentė sistema).



Daugia-agentės sistemos (1)

- Daugia-agentės sistemos leidžia uždavinį paskirstyti skirtingiems agentams.
- Agentai gali veikti aplinką, o skirtingi agentai veikia skirtingose interesų zonose ir skirtingu laipsniu.
- Jeigu skirtingų agentų interesų zonos aplinkoje susikerta, tada sprendžiami konfliktai. Interesų konfliktai sprendžiami įvairiais būdais, pavyzdžiui, apibrėžiant tam tikrą agentų hierarchijas ir pan.



Daugia-agentės sistemos (2)

Daugia-agentinė sistema turi tenkinti šias sąlygas :

- sistemą sudaro **du** ar **daugiau** agentų,
- bent vienas agentas yra **proaktyvus**,
- Bent du agentai turi sąveikauti tarpusavyje, nes kitaip bus neprasminga nagrinėti kelis savarankiškus agentus kaip vienos sistemos komponentus.
- Be to, sąveika turi padėti agentams siekti tikslų.

Agentinės sistemos projektavimas (1)

Agentinės programinės įrangos kūrimo procesas iš esmės apima tokius pačius kūrimo proceso etapus, kaip ir tradiciniai sistemų kūrimo metodai:

- **specifikavimas** - sistemos funkcijų ir ribojimų identifikavimas, t.y. vartotojo, sistemos bei dalykinės srities reikalavimų analizė;
- **projektavimas ir realizacija** – programinės įrangos kūrimas;
- **validavimas** – kur įranga tikrinama, ar tikrai atitinka reikalavimus,
- **palaikymas.**

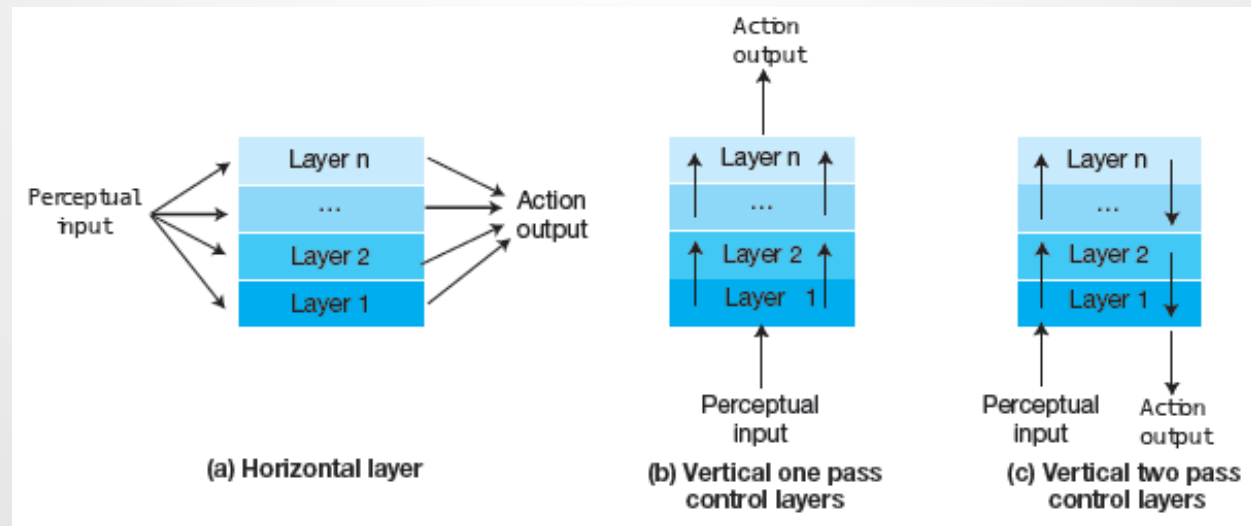
Agentinės sistemos projektavimas (2)

- Analizės bei projektavimo etapuose, gali būti naudojami tiek formalūs, tiek neformalūs metodai.
- Nemažai autorių agentinių sistemų analizei ir projektavimui naudoja formalias specifikavimo kalbas, tokias kaip *ConGolog* ir *CASL*, *Z kalba*.
- Neformalūs/pusiau formalūs agentinių sistemų kūrimo būdai dažniausiai remiasi struktūrizuota kalba ir grafinėmis notacijomis, tokiomis kaip *UML* sekų ar panaudojimo atvejų diagramos.

AGENTŲ ARCHITEKTŪROS

Agentų architektūros

- **Svarstymo architektūros:** Pvz., BDI agentai.
- **Reaktyvios architektūros** (dažniausiai tai paprastos situacija-veiksmas taisyklės).
- **Hibridinės architektūros:** Klasikinių ir alternatyvių metodų junginys, kuriuo siekami paimti tik stipriausias metodų savybes.
- **Sluoksniuota architektūra:** Atvaizduoja sistemas hierarchine struktūra, įtraukiant skirtingas agentų dekompozicijas.



BDI arhitektūra

tikējimai

- *Informacinē būsena*

troškimai

- *Motivacinē būsena*

ketinimai

- *Svarstymo būsena*

Agento būsenos

Yra naudinga išskirti skirtingas agento būsenas pagal jų turinį:

- Taip vadinamas *tikėjimų* arba *pasaulio modelis* talpina vidinę informaciją apie galimas aplinkos situacijas tolimesniems veiksams. Ši informacija yra atnaujinama pagal sensorių įėjimus. Tai vadinama *tikėjimais*, nes tai nebūtinai turi būti teisinga informacija.
- Suvokimo nuostatos susisjusios su sprendimo procesais sukuria ateities būsenas. Jos yra vadinamos kaip *tikslai*, *planai* ir pan. Šie valdo ateities sprendimus ir generuoja veiksmus siekiant užsibrėžto tikslo.



Programos komponentai. Tikėjimų komponentas

Sistemoje yra reikalingi būsenos komponentai, kurie atvaizduotų aplinkos informaciją ir kuri kistu po kiekvieno suvokimo veiksmo.

Tikėjimo komponentas gali būti realizuotas kaip:

- kintamasis,
 - duomenų bazė,
 - aibė loginių išraiškų,
 - duomenų struktūros ir pan.
- Tikėjimai yra sistemos būsenos **informatyvusis** komponentas.

Programos komponentai. Troškimų komponentas

- Būtina, kad sistema turėtų informaciją apie įvykdytus tikslus, arba tiksliau kokia nauda ir praradimai yra susiję su tam tikrais tikslais.
- Toks sistemos komponentas yra vadinamas sistemos *Troškimai (Desires)*, kuris atvaizduoja sistemos motyvacinę būseną.

Pasirinkimo funkcijos perrinkimas

Pasirinkimo funkcijos perrinkimo galimos pasekmės:

- Padidina riziką, jog reikšmingi pokyčiai gali įvykti tuo pat metu, kai bus atliekamas funkcijos perrinkimas.
- Išnaudoja laiką kuris gali būti panaudotas atliekant veiksmus link užsibrėžto tikslo.

Iš kitos pusės, aibės veiksmų vykdymas didina riziką, kad tuo metu gali įvykti reikšmingi pokyčiai ir todėl sistema nepasieks numatytų tikslų ar numatytos naudos.



Ketinimai

- Kiekvienam žingsnyje, veiksmų pasirinkimo pertvarkymas yra nuostolingas ir todėl besąlygiškas veiksmų grupės kaitaliojimas gali neigiamai paveikti sistemą ir ji nepasieks savo tikslų.
- Įmanoma nustatyti ribas kaip dažnai gali būti atliekami pertvarkymai ir tokiu būdu pasiekti balansą tarp per dažnų ir per retų svarstymų.
- Tam tikslui įtraukiamas sistemos būsenos komponentas atvaizduojantis pasirinktų veiksmų grupę esamu momentu.

Pasirinkimo funkcijos išėjimas - sistemos ***Ketinimai*** (**Intentions**), kurie užfiksuoja sistemos svarstymus.

Agentų realizacija

Agentinės sistemos gali būti realizuotos ir įdiegtos daugelyje platformų, tiek specializuotų, standartais paremtų, tiek ir tradicinėse platformose, tokiose, kaip CORBA, Java/J2EE, Microsoft .NET

Realizacijai gali būti naudojamos įvairios programavimo kalbos, tokios kaip:

- Prolog
- Java
- C++
- C#
- ...

Agentų platformos

Agentinių sistemų realizacijai efektyviausia naudotis agentinėmis platformomis, kurios suteikia aibę pakartotinai naudojamų komponentų ir servisų reikalingų agentų realizacijai ir diegimui.

- ☐ JADE

- ☐ Jason

- ☐ Zeus

- ☐ Grasshopper

- ☐ Madkit ir kt.

Agentinių sistemų kūrimo platforma JADE

- Šiandien yra sukurta daug platformų, suteikiančių aplinką agentų egzistavimui ir veikimui, o kartu ir agentinių taikomųjų programų kūrimui palengvinti.
- Viena iš efektyviausių platformų yra **Jade** – Java kalba paremta tarpinė programinė įranga, suteikianti nuo specifinių taikomųjų programų nepriklausantį funkcionalumą, kuris palengvina išskirstytų taikomųjų programų realizaciją.
- Jade remiasi **FIPA** standartais ir palaiko suderinamumą su kitomis platformomis, pagrįstomis šiuo standartu.

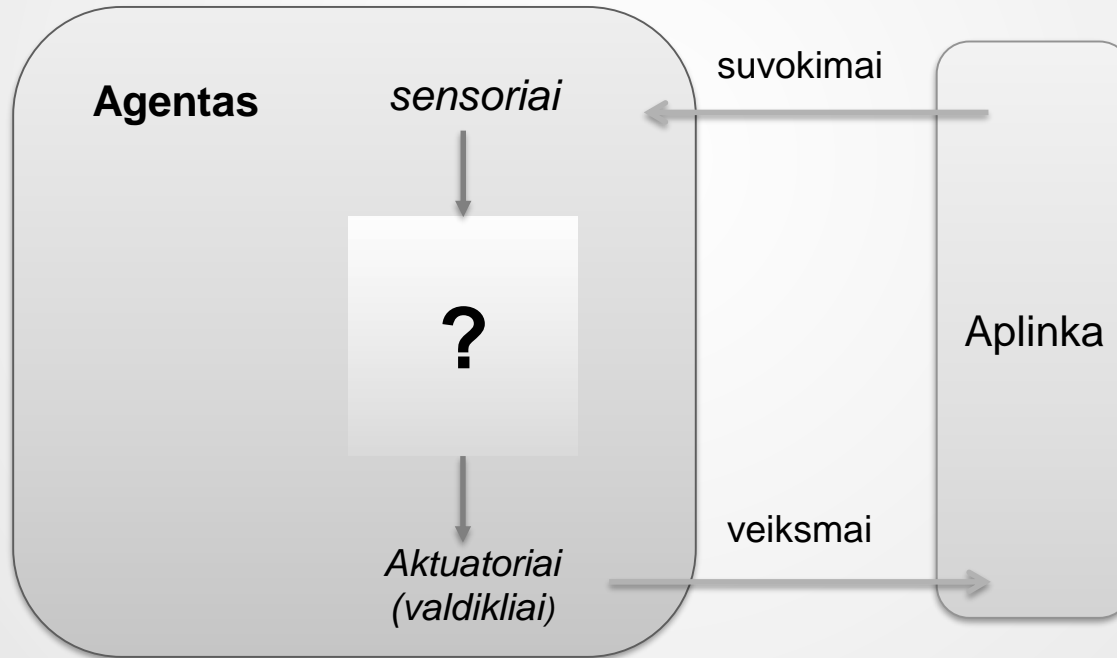
Agentų tipai

- Racionalieji agentai
- Agentų aplinkos tipai
- Programinių agentų galimos struktūros



Programinio Agento sąvoka

Agentas yra esybė, kuris sensorių (jutiklių) pagalbą jaučia ir suvokia aplinką, ir tą aplinką veikia naudojant vykdiklius (aktuatorius).



Jutikliai ir vykdikliai

Žmogus

- **jutikliai:** akys, ausys,...
- **vykdikliai:** rankos, kojos, burna,...

Robotas
agentas

- **jutikliai:** kamera, infraraudonųjų spindulių jutikliai,...
- **vykdikliai:** motoras, ratai,....

Programinis
agentas

- **jutimai:** klavišų paspaudimas, failo turinys,..
- **vykdymas:** išvestis ekrane, įrašymas į failą, tinklo paketų siuntimas,...

Agentai ir aplinka

Suvokimas yra agento gauti įėjimai esamoje būsenoje.

Suvokimų seka – pilna agento suvokimų istorija.

Bendru atveju agento veiksmai priklauso nuo suvokimų sekos.

Agento funkcijos apibrėžia agento elgseną. **Agento funkcija** atvaizduoja duotąją suvokimų seką į veiksmą (veiksmus).

Agento programa realizuoja agento funkcijas.

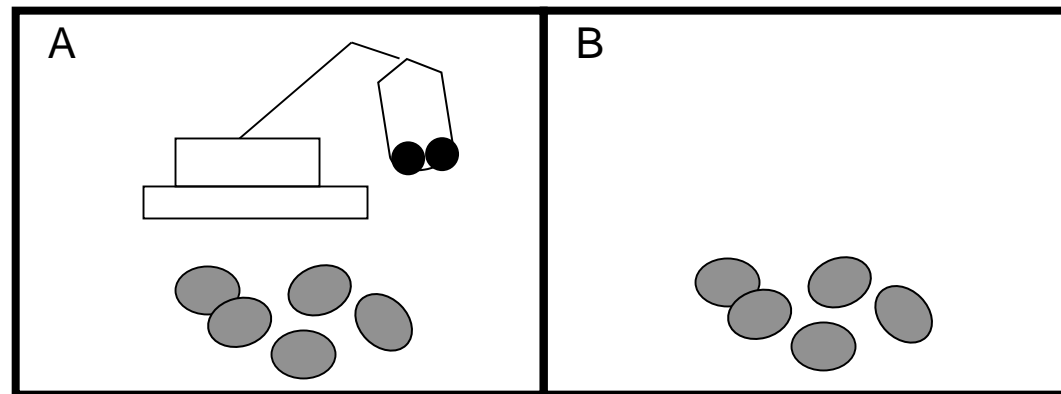
Agento funkcija yra abstrakti matematinė išraiška, agento programa yra konkreti realizacija.

Pavyzdys: Siurblys

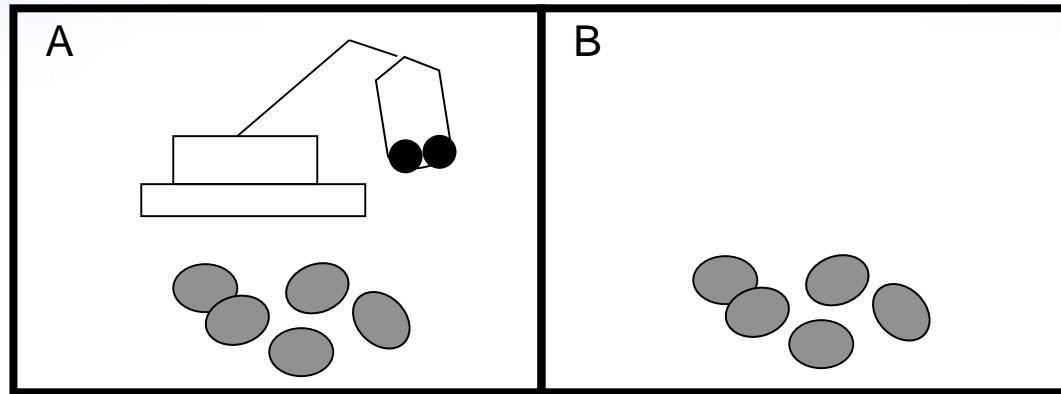
- 2 vietos: kvadratai A ir B
- **Jutikliai:** purvo sensorius ir pozicijos sensorius.
- **Vykdikliai:** Siurbimo įrenginys, variklis, ratai.

Siurblio agentas supranta, kuriame kvadrate jis yra ir ar jame yra purvo.

- **Veiksmai:** judėti į kairę/dešinę, siurbti, nieko nedaryti.
- **Agento funkcija:** jei kvadrate yra purvo -> siurbti, jei ne -> judėti į kitą kvadratą.



Pavyzdys: Siurblys (metamatinė išraiška)



Percepts: $x_1(t) \in \{A, B\}$, $x_2(t) \in \{\text{clean}, \text{dirty}\}$

Actions: $\alpha_1(t) = \text{suck}$, $\alpha_2(t) = \text{right}$, $\alpha_3(t) = \text{left}$

$$\mathbf{x}(t) = \begin{pmatrix} B \\ \text{clean} \end{pmatrix} \Rightarrow \mathbf{a}(t) = \begin{pmatrix} - \\ - \\ \text{left} \end{pmatrix}$$

$$\mathbf{x}(t) = \begin{pmatrix} A \\ \text{dirty} \end{pmatrix} \Rightarrow \mathbf{a}(t) = \begin{pmatrix} \text{suck} \\ - \\ - \end{pmatrix}$$

$$\mathbf{x}(t) = \begin{pmatrix} A \\ \text{clean} \end{pmatrix} \Rightarrow \mathbf{a}(t) = \begin{pmatrix} - \\ \text{right} \\ - \end{pmatrix}$$

$$\mathbf{x}(t) = \begin{pmatrix} B \\ \text{dirty} \end{pmatrix} \Rightarrow \mathbf{a}(t) = \begin{pmatrix} \text{suck} \\ - \\ - \end{pmatrix}$$



Pavyzdys: dulkių siurblys. Suvokimų seka

Suvokimų seka	veiksmas
[A,Clean]	Right
[A,Dirty]	Suck
[B,Clean]	Left
[B,Dirty]	Suck
[A,Clean], [A,Clean]	Right
[A,Clean], [A,Dirty]	Suck
...	...
[A,Clean], [A,Clean], [A,Clean]	Right
[A,Clean], [A,Clean], [A,Dirty]	Suck



Racionalus agentas

- **Racionalus agentas** – toks, kuris daro, ką reikia, t.y. kiekvienas agento funkcijos lentelės įrašas teisingas. Racionalus agentas kiekvienai suvokimų sekai pasirenka veiksmą, kuriuo tikimasi maksimizuoti jo veiklos įverčius.

***Rational Agent:** one that does the right thing*

- Kaip nuspręsti, kada agentas geras, kada blogas?
- Ką reiškia **teisingai** elgtis?
 - Teisingas veiksmas, kurį agentas pasirenka ir kurį pasirinkus agentui labiausiai “pasiseka” (agento našumas yra didžiausias)
- **KAIP** ir **KADA** tą sėkmę pamatuoti?



Elgsenos įvertis

Elgsenos įvertis - sėkmingą agento elgseną apibrėžiančio kriterijaus išraiška.

KAIP ?

- Kokie kriterijai įvertinantys agento elgsenos sėkmę.
- Nėra vieno bendro matavimo vieneto, kuris tiktų visiems agentams.
- Galima būtų paklausti agento, kaip jis jaučiasi, bet tikriausiai nesugebės atsakyti.



KAIP tą sėkmę pamatuoti?

Pavyzdys: Siurblys



Siurblio agento **elgsenos įvertinimas**:

- Kiek purvo susiurbė per tam tikrą laiką.
- Kiek suvartojo elektros ir kiek sugeneravo triukšmo.

Visada geriau modeliuoti elgsenos įvertinimus, pagal tai, ko iš tikrųjų siekiama aplinkoje, nei pagal tai, kaip agentas turėtų elgtis.

- Apdovanoti agentą už švarias grindis. Tarkime už kiekvieną švarų kvadrata gaunamas taškas ir minusuojami taškai už elektros sunaudojimą ar triukšmą.

KADA tą sėkmę pamatuoti?

- **KADA** - vertinti taip pat svarbu:
 - Jei vertintume pagal tai, kiek purvo susiurbė pirmąją dienos dalį, aukštesnį įvertį gautų agentas, kuris greičiau dirbo pirmąją dienos dalį, nors ir paskui dirbo lėtai.
 - „nuskriaustume“ tuos, kurie dirbo tolygiai visą dieną.
- Taigi, vertinant reikėtų pasirinkti ilgesnį laiko tarpą.
 - pvz. 8 valandų darbo dieną ar pan.



Nuo ko priklauso racionalumas

Racionalumas duotuoju laiko momentu priklauso nuo:

1. Elgsenos įverčio, kuris apibrėžia sėkmės kriterijų;
2. Agento išankstinių žinomų apie jį supančia aplinką;
3. Veiksmų, kuriuos agentas gali atlikti;
4. Agento suvokimų sekos duotuoju laiko momentu.



Ar siurblio agentas racionalus?

Siurblio agento pasaulis. Tarkime:

- Elgsenos įvertis: suteikiamas 1 taškas už kiekvieną išvalytą kvadratą kiekviename žingsnyje;
- Agento gyvavimo ciklas: 1000 žingsnių.
- Aplinkos „geografija“ žinoma iš anksto, tačiau purvo pasiskirstymas ir agento pradinė pozicija nežinoma.
- Švarūs kvadratai išlieka švarūs, o siurbdamas agentas išvalo nešvarųjį. Su *Left* agentas juda į kairę, su *Right* į dešinę, nebent tai išeina už aplinkos ribų
- Vieninteliai veiksmai: *Left*, *Right*, *Suck*, *NoOp*.
- Agentas teisingai suvokia savo vietą ir ar joje yra purvo.

Probleminės srities specifikavimas

Prieš kuriant aplinką pirmiausia reikia apsibrėžti problemine srities aplinką ir kaip galima pilniau. (angl. *task environment*):

PEAS:

- Elgsenos įvertis
- Aplinka
- Vykdikliai
- Jutikliai

Performance
Measure

Environment

Actuators

Sensors



Pavyzdys: Automatinis taksi vairuotojas

Elgsenos
įvertis:

- Pasiiekti reikiamą adresą
- minimizuoti kuro sąnaudas ir nusidėvėjimą
- minimizuoti kelionės laiką ar kainą
- nepažeisti eismo taisyklių
- netrukdyti kitiems vairuotojams
- užtikrinti maksimalų keleivio saugumą ir komfortą
- maksimizuoti uždarbį



Pavyzdys: Automatinis taksi vairuotojas

Aplinka:

- Įvairūs keliai;
- kiti automobiliai, pėstieji, gyvūnai, policija;
- sąveika su keleiviais;
- eismo sąlygos: sniegas, lietus....

Vykdikliai:

- greičio valdymas akseleratoriaus pagalba;
- vairas vairavimui;
- stabdžiai;
- ekranas,...

Jutikliai

- TV kameros;
- spidometras;
- akselerometras;
- GPS,...

(PEAS) pavyzdžiai

Agento tipas	Veiklos įvertis	Aplinka	Valdikliai	Sensoriai
Sveikatos diagnozės centras	Sveikas pacientas, minimizuoti kaštai	Pacientas, ligoninė, darbuotojai	Ekranas (klausimai, testai, diagnozės, gydymas)	Klaviatūra (įvesti paciento simptomus, atsakymus)
Dalių- surinkimo robotas	Dalių teisingose konteineriuose procentas	Konvėjero juosta su dalimis, konteineriai	Lankstomos rankos ir plaštakos	Kamera, kampų jutikliai
Interaktyvus anglų kuratorius	Maksimizuoti studentų testų balus	Studentų aibė	Pateikti užduotis, pasiūlymus, taisymus	Klaviatūros įvestis

Aplinkos rūšys

- Probleminių sričių, kuriose sprendžiami AI uždaviniai yra labai daug.
- Pagal aplinkos rūšį taikomi ir specifiniai agentų tipai problemoms spręsti.
- Aplinkos rūšys:
 - Pilnai/Dalinai stebima aplinka
 - Deterministinė/Stochastinė aplinka
 - Epizodinė/nuosekli aplinka
 - Statinė/dinaminė aplinka
 - Diskreti/Ištęstinė aplinka

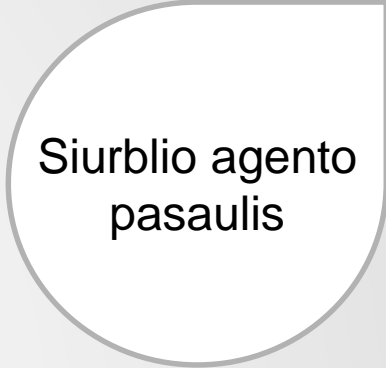


Pilnai matoma aplinka vs Dalinai matoma aplinka

Pilnai matoma aplinka:

- Jei agento jutikliai suteikia jam pakankamai informacijos apie aplinką bet kuriuo laiko momentu;
- Jutikliai turi identifikuoti visus aspektus, kurie yra svarbūs veiksmo pasirinkimui;

Tokia aplinka patogi, nes agentui nereikia saugoti vidinės būsenos ir žinoti kas vyksta aplink.

A white rounded rectangle with a grey border, representing a full environment.

Siurblio agento
pasaulis

Dalinai matoma aplinka

- Triukšmas aplinkoje ir netikslūs jutiklio parodymai.

Jutiklių trūkumas → informacijos trūkumas

A white rounded rectangle with a grey border, representing a partially observable environment.

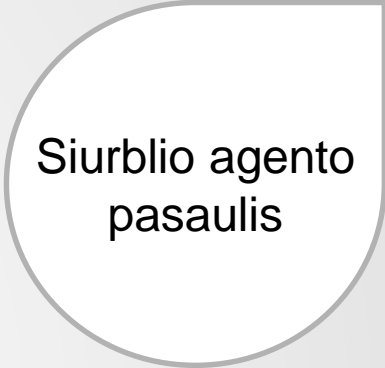
Taksi
vairuotojas

Deterministinė aplinka vs Stochastinė aplinka

- **Deterministinė aplinka:**

- Jei aplinkos būseną visiškai priklauso nuo dabartinės būsenos ir agento įvykdyto veiksmo. Priešingu atveju tai stochastinė aplinka

Agentui pilnai stebimoje ir deterministinėje aplinkoje nėra dėl ko jaudintis, nes nėra neaiškių situacijų.



Siurblio agento
pasaulis

- **Stochastinė aplinka**

- Daug kas priklauso nuo atsitiktinumų;
- Jei aplinkai dalinai stebima tai galima sakyti jog tai stochastinė aplinka.



Taksi
vairuotojas

Epizodinė aplinka vs nuosekli aplinka

Epizodinė aplinka:

- Epizodinėje aplinkoje agento patirtis padalijama į atskirus epizodus, kur kiekvienas jų atitinka tam tikrą agento suvoktą ir įvykdytą veiksmą.
- Sekantis epizodas **nepriklauso** nuo praeitame epizode atliktų veiksmų.
- Veiksmo pasirinkimas kiekvienu epizodu priklauso tik nuo pačio epizodo.
- Dabartinis sprendimas neturi jokios įtakos kitam sprendimui.

“Nuoseklios sekos” aplinka:

- Dabartinis sprendimas gali turėti įtakos visiems būsimiems sprendimams

Klasifikavimo
uždaviniai

šachmatai

Statinė aplinka vs dinaminė aplinka

Dinaminė aplinka:

- Aplinka gali pasikeisti, kol agentas samprotauja.

Taksi
vairuotojas

Statinė aplinka:

- Daug paprastesnė, nes agentas sprendamas uždavinius neturi stebėti pasaulio, nei rūpintis dėl einančio laiko.

„Puzzle“
delionės

Pusiau-dinaminė aplinka:

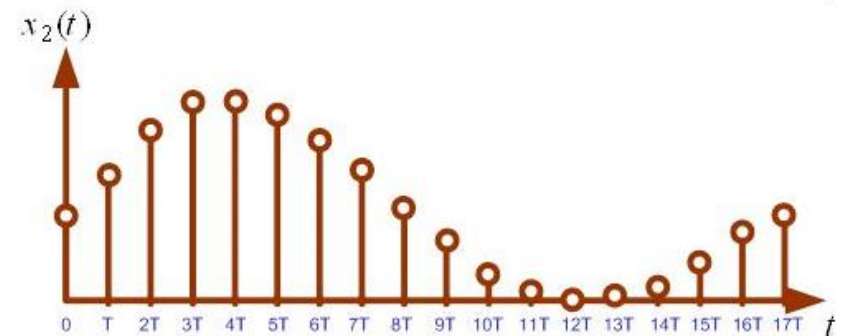
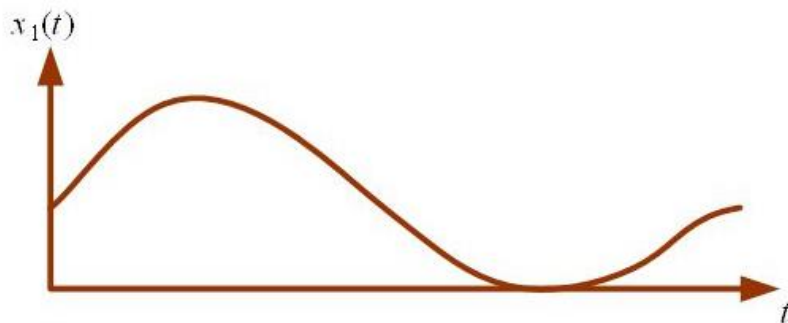
- Jei aplinka laikui bėgant nesikeičia, tačiau agento elgsenos įvertis keičiasi.

Šachmatai
su laikrodžiu

Diskretinė vs tolydinė

Diskretinė ir tolydinės aplinkos yra taikomos aplinkos būsenai pagal tai kaip yra traktuojamas laikas bei agento suvokiami ir veiksmai.

- Šachmatų žaidimas yra diskretinės aplinkos pavyzdys. Užduodamas baigtinis būsenų skaičius.
- Taksi vairuotas yra tolydinės aplinkos pavyzdys (greitis, vieta iš lėto kinta laike).



Agentų programos

Agentas = Architektūra + Programa

- Programa – realizuoja agento funkcijas, kurios atvaizduoja agento suvokimus į vykdomus veiksmus.
- Architektūra - įrenginys su fiziniais sensoriais ir aktuatoriais.

Visos agentų programos pagrįstos bazine agento programa (angl. *skeleton program*)

- Agento programa gauna tik vieną jutimą kaip įvestį iš sensorių ir grąžina veiksmą aktuatoriams;
- Jeigu agento veiksmai priklauso nuo visos suvokimų sekos tai jis turi prisiminti visus suvokimus.

“Table-driven” Agentas

Paprasčiausias būdas parašyti programą – sudaryti lentelę.

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  static: percepts, a sequence, initially empty
           table, a table of actions, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```



“Table-driven” Agento trūkumai

- Trūkumai:
 - Milžiniški lentelių dydžiai
 - Joks agentas neturėtų tiek atminties lentelei išsaugoti
 - Užtrunka labai ilgai kūrėjui sukurti lentelę
 - Agentas nėra autonomiškas
 - Netgi jei agentas ir turėtų mokymosi mechanizmą, lentelės mokymasis užtruktų labai ilgai

Pagrindinis AI iššūkis – išsiaiškinti, kaip rašyti programas, kurios užtikrintų racionalią elgseną realizuojant kuo mažesniu programos kodo kiekiu.



Agentų programų tipai

Analizuosime 4 agentų programų rūšis:

- Paprastas refleksinis agentas (angl. *simple reflex agent*)
- Refleksinis agentas su vidine būseną (angl. *reflex agent with internal state* arba *model-based agent*)
- Į tikslą orientuotas agentas (angl. *goal-based agent*)
- Į naudingumą orientuotas agentas (angl. *utility-based agent*)

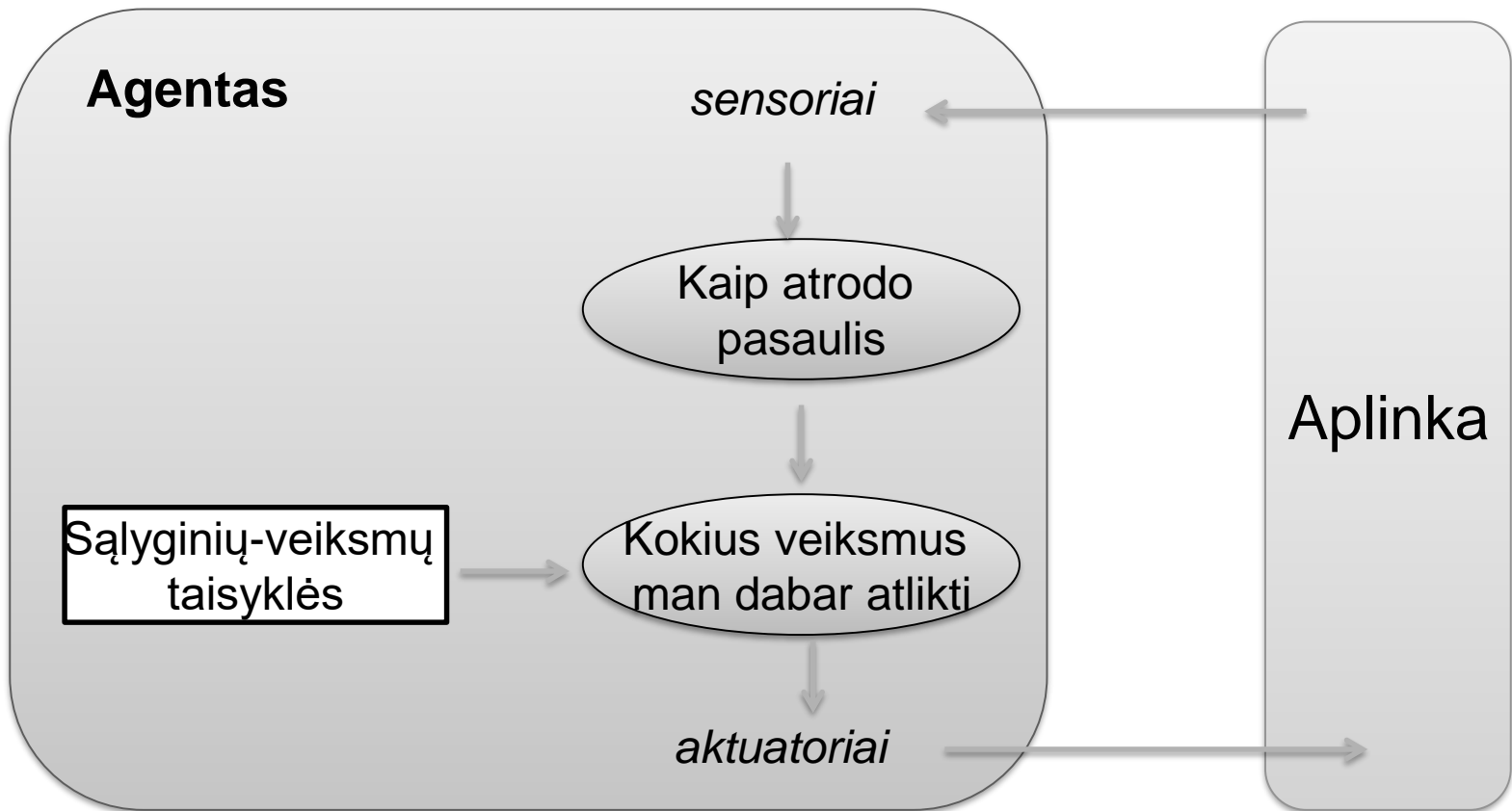
Agentai, gebantys mokytis:

- Apsimokantis agentas (angl. *learning agent*)



Paprastas refleksinis agentas

Reaguojančių agentų architektūra ir realizacija labai paprasta. Agento veiksmo parinkimas remiasi tik dabartiniu jo suvokimu (ignoruoiant suvokimų istoriją).



Paprastas refleksinis agentas

```
function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules, a set of condition-action rules

  state  $\leftarrow$  INTERPRET-INPUT (percept)
  rule  $\leftarrow$  RULE-MATCH (state, rules)
  action  $\leftarrow$  RULE-ACTION [rule]
  return action
```

A simple reflex agent works by finding a rule whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule.

Paprastas refleksinis agentas

- Sąveikai su aplinka reikalinga surašyti visas sąlyginių veiksmų taisykles
 - jei automobilis priekyje stabdo → pradėti stabdyti
- Agentas labai paprastas, neintelektualus, aplinka turi būti pilnai matoma.

Trūkumai:

- Agentas veiks teisingai tik tuo atveju, jei gali priimti sprendimą priklausomai nuo dabartinio jutimo.
- Taisyklių rinkinys gali būti milžiniškas (priklausomai nuo probleminės srities).
- Ilgai užtrunka sukurti taisyklių rinkinį.
- Jokių žinių apie aplinką, jei nėra specifinių jutiklių.
- Neadaptyvus.

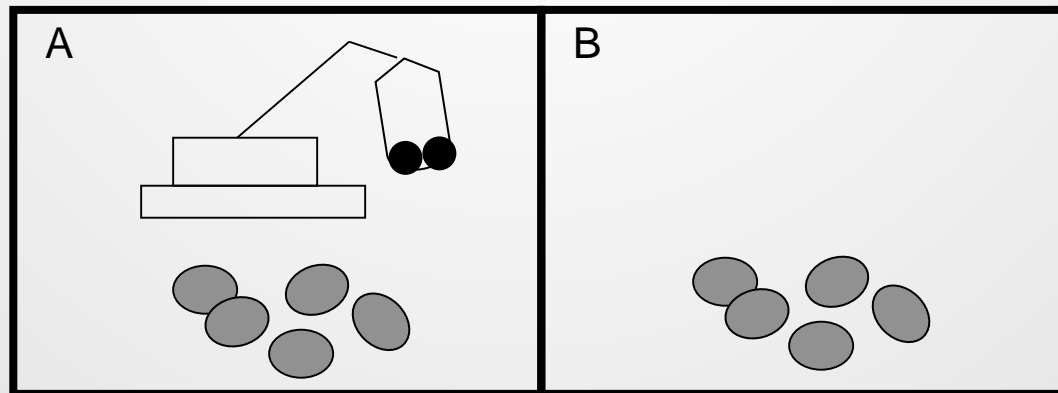
Dulkių siurblys - paprastas refleksinis agentas

- Sensorius: dulkių sensorius.
- Suvokimai: 2 suvokimai [*Dirty*] ir [*Clean*].
- Veiksmai: kai [*Dirty*] tai ***Suck***. Judėti ***Left, Right***.

Tarkim agentas neturi vietos nustatymo sensoriaus.

Ką daryti kai [*Clean*]???, ėjimas į kairę kai jis yra A kvadrate, bus ciklas, ėjimas į dešinę kai yra B kvadrate ciklas.

Išeitis: atsitiktinis veiksmų parinkimas. Jeigu švaru eiti į kairę ar dešinę atsitiktinai.



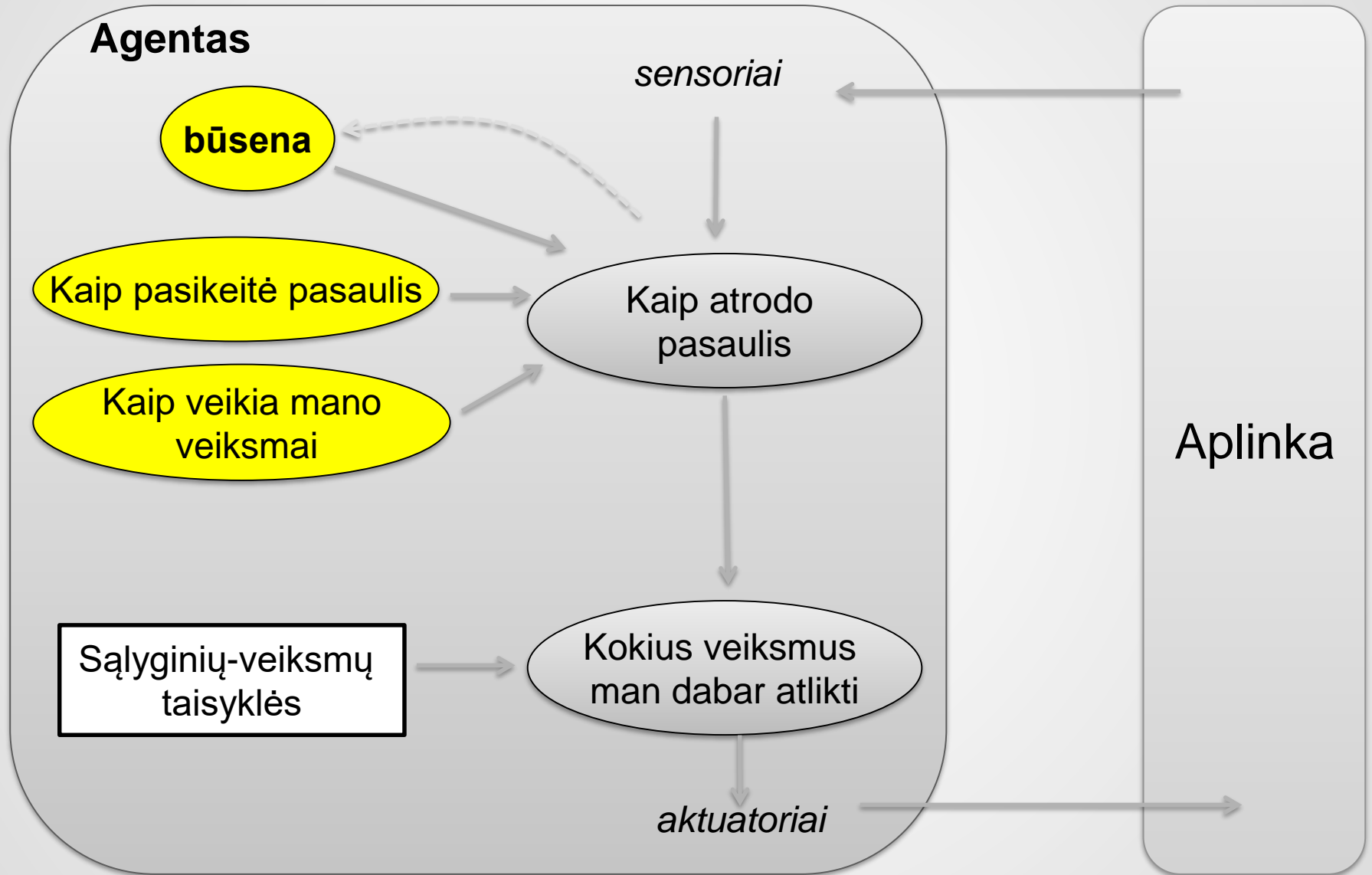
Modeliu-paremtas agentas

- Vidinės informacijos atnaujinimas laike reikalauja 2 tipų agento žinių:
 - informacijos apie tai, kaip elgiasi aplinka, nepriklausomai nuo agento veiksmų
 - informacijos, kaip pačio agento veiksmai paveikia pasaulį.
- Žinios apie tai kaip veikia pasaulis, nepriklausomai, kaip tai bus realizuota, vadinama pasaulio **MODELIU**.
- Agentas, naudojantis tokį modelį, vadinamas **Model-based agentu**.

Modeliu-paremtas refleksinis agentas

- Refleksinis su vidine būseną agentas veiksmą pasirenka, atsižvelgdamas ne tik į jutiklių pateiktą informaciją, bet ir į saugomą aplinkos modelį.
- To reikia tais atvejais, kai jutiklių informacija neišsami ir jos nepakanka dvejoms skirtingoms aplinkos būsenoms atpažinti.
- Aplinkos būsenos vadinamos skirtingomis jei, reaguojant į jas, turi būti vykdomi skirtingi veiksmai.

Modeliu-paremtas refleksinis agentas



Modeliu-paremtas refleksinis agentas

Gavęs jutiklių informaciją, agentas atsižvelgia į esamą suvokimą, atnaujiną vidinę būseną ir parenka tinkamą taisyklę.

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action
  static: state, a description of the current world state
           rules, a set of condition–action rules
           action, the most recent action, initially none

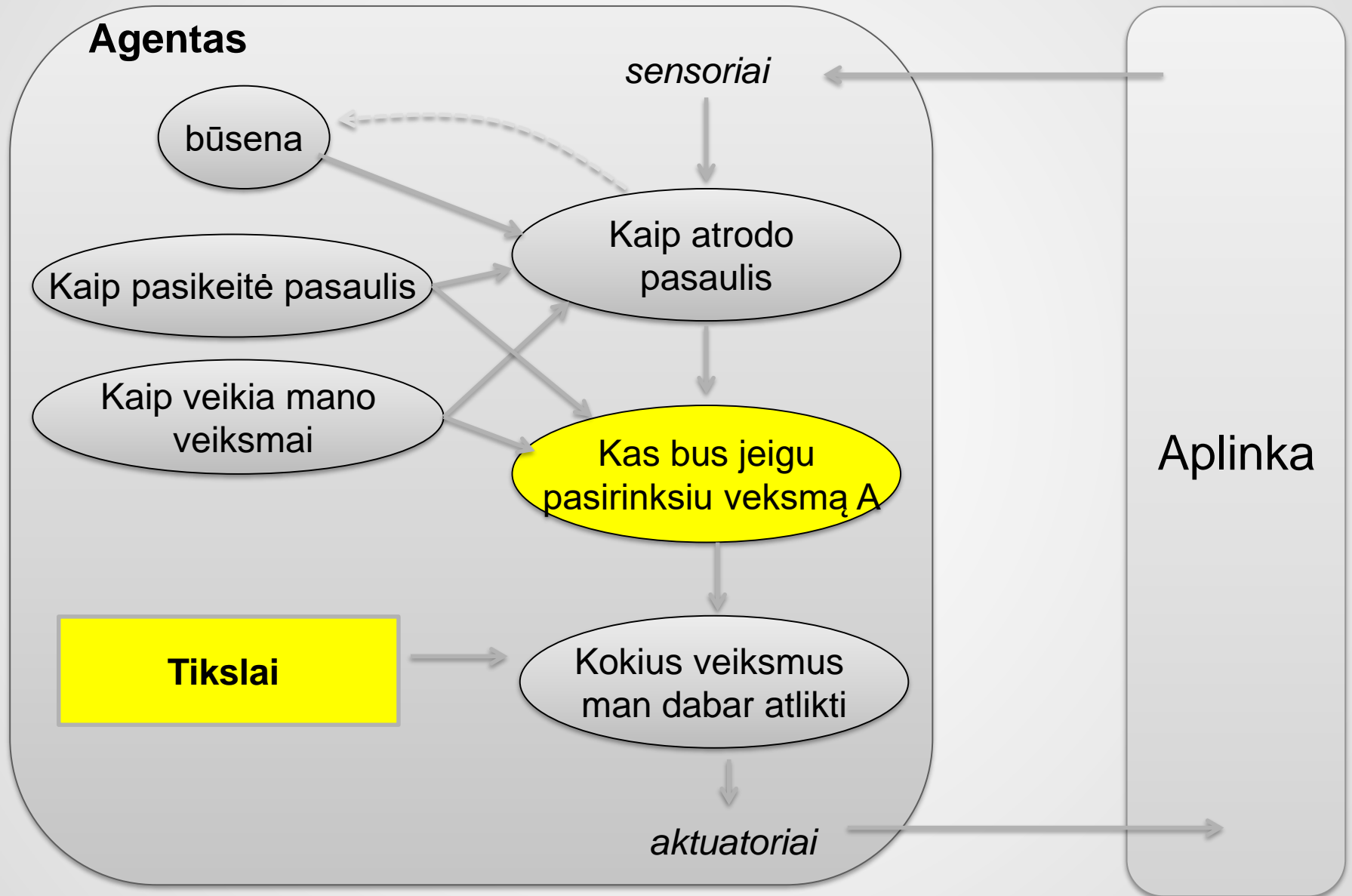
  state ← UPDATE-STATE(state, action, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```


Agento tikslai

- Aplinkos dabartinės būsenos žinojimas ne visada yra pakankamas tam kad agentui žinoti kaip elgtis duotoje situacijoje.
 - Pavyzdžiui, tam tikroje situacijoje gali būti vykdomas ne vienas, o vienas iš aibės leistinų veiksmų (sankryžoje gali suktį į kairę, dešinę, arba važiuoti tiesiai.)
- Tai reiškia, kad agentas turi žinoti ne tik dabartinę būseną, bet ir **tikslo** informaciją, kuri apibrėžia pageidaujamą būseną (situaciją).



Tikslo agentas



Naudingumo funkcija

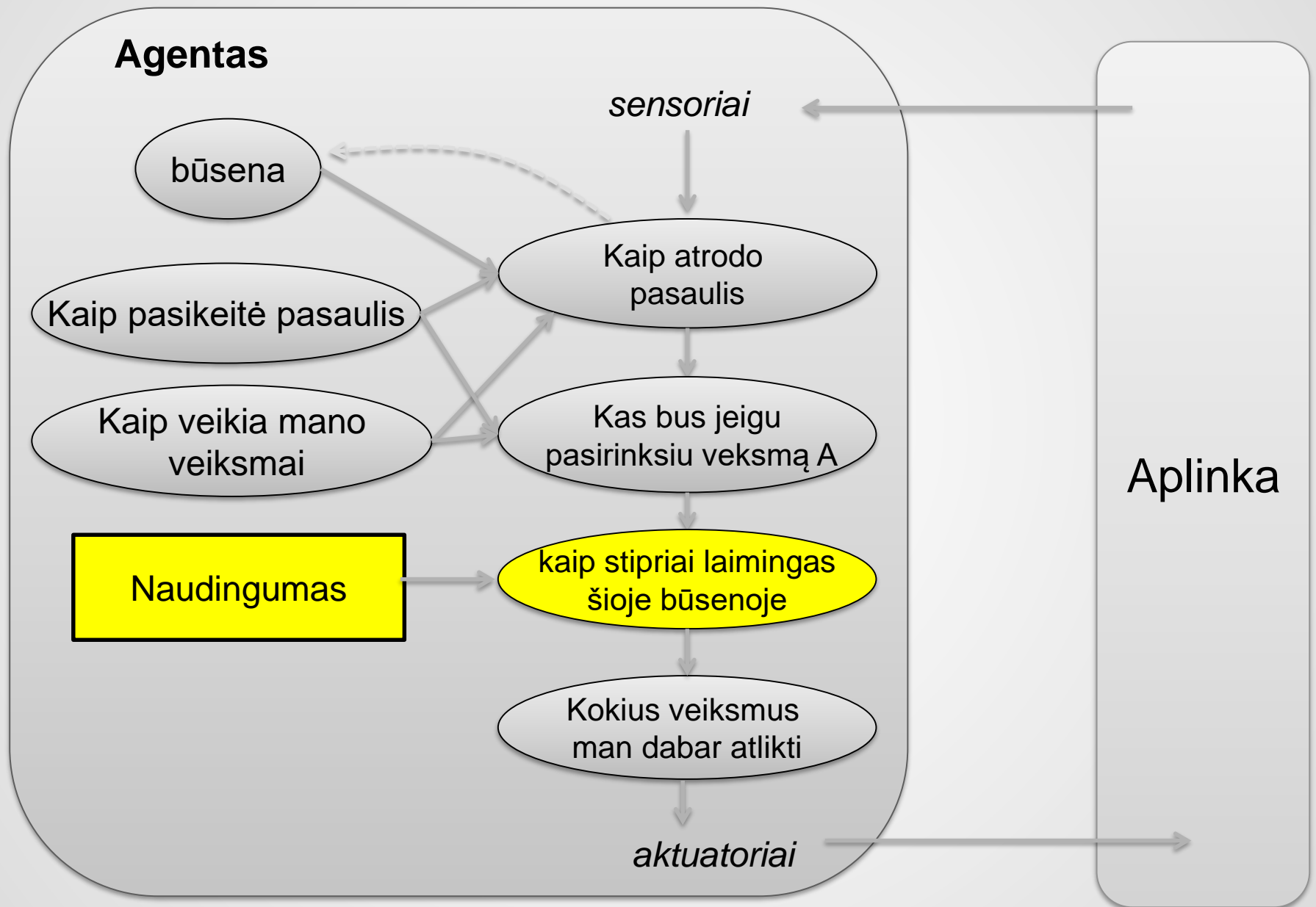
- Sumodeliuoti intelektualią agento elgseną dažnai gali nepakakti net ir tikslų.
 - pvz. pasiekti tikslą, t.y. nuvažiuoti į tam tikrą vietą taksi gali daugeliu maršrutų, tačiau kai kurie iš jų yra greitesnis, saugesni, pigesni ir pan.
 - tikslas tik leidžia agentui būti vienoje iš būsenų „happy“ arba „unhappy“, tačiau reikalingas elgsenos įvertis, nustatantis „how happy“.
- Jei viena pasaulio būseną yra labiau norima, lyginant su kitomis, ji atneša agentui daugiau **naudos**.
- **Naudingumo funkcija** susieja būseną (ar jų seką) su realiu skaičiumi, kuris nurodo atitinkamą „laimingumo“ laipsnį.

Kada reikalinga naudingumo funkcija

- Naudingumo funkciją geriau naudoti nei tikslus dviem atvejais:
 - kai tikslai konfliktuoja tarpusavyje (pvz. saugumas ir greitis)
 - kai agentas turi ne vieną tikslą ir nėra garantijų, kad jie bus pasiekti.



Naudingumo agentas



Apsimokantys agentai

- Kiekvienas agentas gali pagerinti savo veikimą apsimokant. Agentas pajėgus apsimokyti keičiant ar koreguojant veiksmus nusakančias taisykles.
- Nuoseklių aplinkos būsenų stebėjimas leidžia agentui “išmokti kaip keičiasi ir vystosi pasaulis”, o jo veiksmų padarinių įvertinimas leidžia agentui “išmokti ką daro mano veiksmai”.



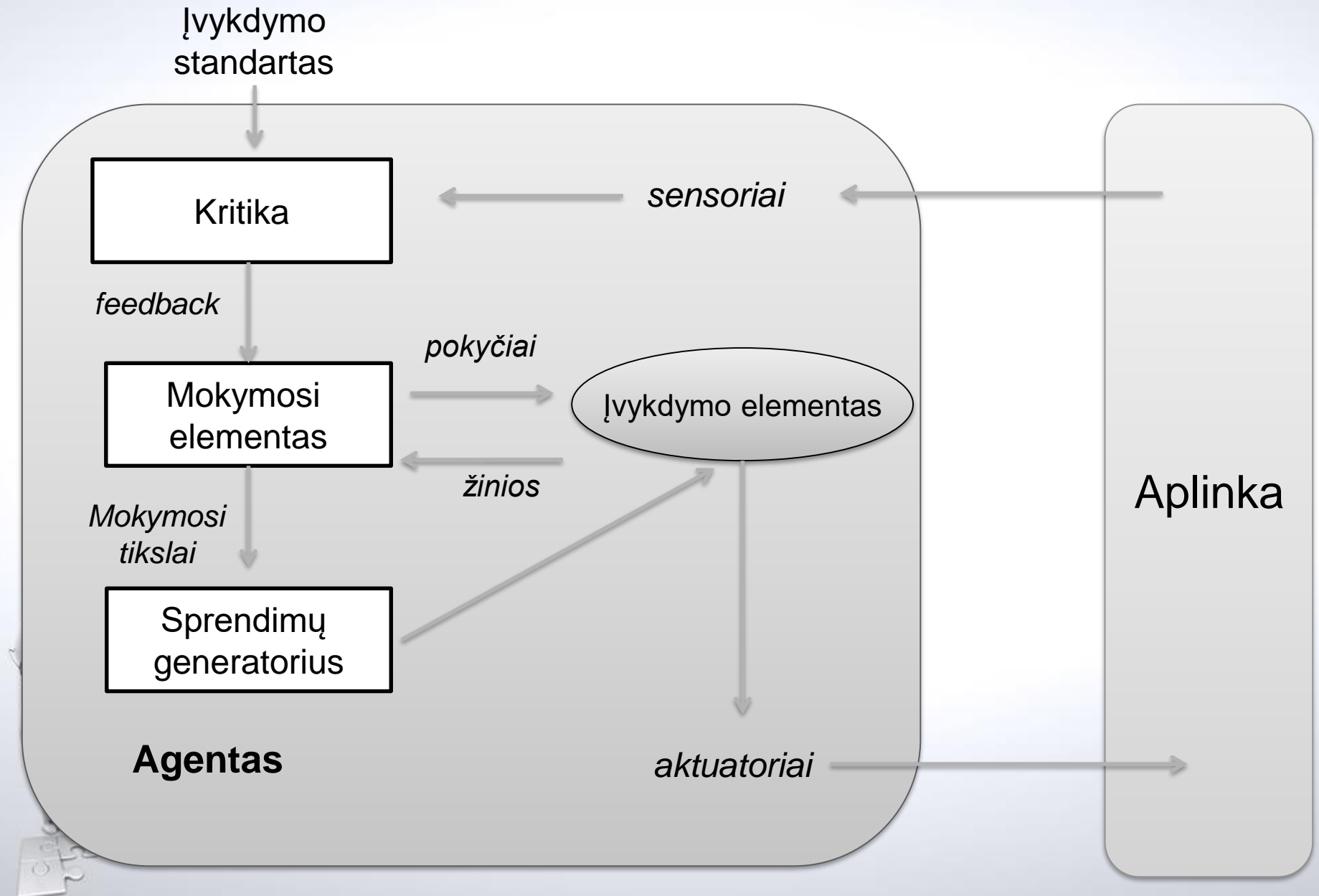
Apsimokantis agentas

Apibendrintą apsimokančio agento modelį sudaro 4 konceptualūs komponentai:

- Mokymosi elementas
- Įvykdymo (Elgsenos) elementas
- Kritikos elementas
- Problemų generatorius



Apsimokantis agentas



Apsimokantis agentas

- **Įvykdymo elementas** – atsakingas už veiksmų parinkimą priklausomai nuo suvokimų.
- **Kritikos elementas** (angl. *critic*) pasako mokymosi elementui, kaip agentas elgiasi, lyginant su standartine elgsena
 - jis reikalingas, nes jautiklių informacija nepasako, ar tai, ką agentas suvokia yra gerai ar blogai.
 - elgsenos standartas turi būti fiksuotas. Nes agentas negali turėti galimybės tai koreguoti pagal savo elgseną.

Apsimokantis agentas

- **Mokymosi elementas** – atsakingas už tobulėjimą.
 - naudoja grįžtamąją informaciją iš kritikos elemento, apie tai, kaip agentui sekasi ir nusprendžia, kaip įvykdymo elementas turėtų būti pakeistas, geresniam veikimui ateityje.
 - sprendžiamas klausimas ne „kaip agentas to išmoks?“, bet „kokio įvykdymo elemento agentui reiks, kai jis jau išmoko kaip tą daryti.“
- **Sprendimų generatorius** (angl. *problem generator*)
 - turi pasiūlyti veiksmus, kurie ilgainiui pagerintų agento veikimą