

Porazdeljene inteligentne programske tehnologije (PIPT)

Uvodne informacije

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Porazdeljene inteligentne programske tehnologije

O predmetu

- Porazdeljene inteligentne programske tehnologije
- Univerzitetni program FRI, 4. letnik, smer programska oprema
- 3 ure predavanj in 3 ure laboratorijskih vaj tedensko
- Nov predmet, letos se izvaja prvič

- Inteligentni sistemi
 - Fizični, robotski/senzorski, utelešeni sistemi
- Porazdeljeni sistemi
 - Porazdeljene funkcionalne enote enega kompleksnega sistema
 - Večagentni inteligentni sistemi

- Domača stran predmeta:
<http://ucilnica.fri.uni-lj.si/course/view.php?id=113>

Porazdeljene inteligentne programske tehnologije, Uvodne informacije

2

Izvajalci

- Predavatelj: doc. dr. Danijel Skočaj
 - Laboratorij za umetne vizualne spoznavne sisteme
 - e-pošta: danijel.skocaj@fri.uni-lj.si
 - url: <http://vicos.fri.uni-lj.si/danjels/>
 - tel: 01 4768 189
 - prostor: Jadranska 21, soba 9
 - govorilne ure: torek, 10:00-11:00 ali po dogovoru
- Asistenta:
 - Luka Čehvin
 - e-pošta: luka.cehovin@fri.uni-lj.si
 - tel: 01 4768 360
 - prostor: Jadranska 21, soba 8
 - Alen Vrečko
 - e-pošta: alen.vrecko@fri.uni-lj.si
 - tel: 01 4768 769
 - prostor: Jadranska 21, soba 7



Vsebina predavanj

- Intelligentni agenti
 - Kaj so inteligentni agenti
 - Arhitekture inteligentnih agentov
- Intelligentni spoznavni sistemi
 - Zahteve
 - Sposobnosti: zaznavanje, učenje, sklepanje, načrtovanje, komunikacija, mediacija, manipulacija, navigacija
 - Različni scenariji, primeri
 - Nehomogenost spoznavnih sistemov
 - Učenje v spoznavnih sistemih
 - Komunikacija v spoznavnih sistemih
- Arhitekture inteligentnih spoznavnih sistemov
 - Pregled nekaterih arhitektur
 - CAST - Cosy architecture schema toolkit

Vsebina predavanj

- Večagentni sistemi in združbe agentov
 - Komunikacija med agenti
 - Komunikacijski in interakcijski protokoli
 - Združbe agentov
- Porazdeljeno reševanje problemov in načrtovanje
 - Porazdeljevanje nalog, rezultatov
 - Porazdeljeno načrtovanje, izvajanje
- Porazdeljeno odločanje
 - Različne tehnike: glasovanje, dražbe, barantanje, idr.
- Učenje v večagentnih sistemih
 - Koordinacija med učenjem in aktivnostjo
 - Učenje o in od drugih agentov
 - Učenje in komunikacija
- Aplikacije porazdeljene umetne inteligence

Laboratorijske vaje

- Ob torkih po predavanjih (ob 15:00)
- V računalniški učilnici LRI-JD na Jadranski 21
- Asistenta Luka Čehovin in Alen Vrečko

- Vaje praktično usmerjene, z računalniki in kamerami
- Cilj: razviti porazdeljen inteligenten senzorsko/robotski sistem
- Strojna oprema: več aktivnih PTZ kamer nadziranih z računalniki
- Programska oprema: ogrodje za komunikacijo in integracijo različnih komponent CAST
 - Pripravljen sistem za porazdeljeno delovanje (Linux, C++)
 - Pripravljene komponente za nadzor kamer
- Predpostavljeno znanje
 - računalniškega vida osvojeno pri predmetu Računalniško zaznavanje v prvem semestru
 - programskega jezika C++ (v Linux okolju)

Laboratorijske vaje

- Sistem več aktivnih PTZ kamer, ki naj bi bil sposoben opazovati dogajanje na sceni in se primerno odzivati
 - Vsaka kamera je povezana s svojim računalnikom; skupaj tvorita enega agenta
 - Vsak agent zaznava svet preko kamere
 - Agent lahko spreminja pogled kamere - lahko obrača kamero v vodoravni in navpični smeri ter povečuje sliko
 - Vsak agent ima svoj (delen) pogled na svet
 - Agenti morajo skupaj rešiti zadano nalogo
 - Med seboj morajo ustrezno komunicirati in si porazdeliti delo
 - Centralizirano odločanje, porazdeljeno odločanje
- Scenarij:
 - Poligon z različnimi ovirami
 - Sistem kamer mora slediti predmetom, ki se premikajo po poligonu

Izpitni režim

- Seminaraska naloga + pisni izpit + ustni izpit
- Seminaraska naloga
 - Sodelovanje na vajah
 - Razvoj komponente porazdeljenega inteligentnega sistema
 - Zagovor seminarске naloge (jasno izpostavljen prispevek avtorja)
 - (možnost nadgradnje v diplomsko nalogo)
- Pisni izpit
 - Praktične naloge
 - Teoretična vprašanja
- Ustni izpit
 - Preverjanje razumevanja

Literatura

- Glavna literatura:
 - Multiagent Systems - Modern Approach to Distributed Artificial Intelligence
Edited by Gerhard Weiss, The MIT Press
 - Poglavlja: 1, 2, 3, 5, 6 in 9
- Nekatera poročila in članki projekta CoSy - Cognitive Systems for Cognitive assistants
 - <http://cognitivesystems.org/>
- Dodatna literatura:
Michael Wooldridge
An Introduction to MultiAgent Systems
- Gradivo na spletni strani predmeta:
 - <http://ucilnica.fri.uni-lj.si/course/view.php?id=113>



Porazdeljene inteligentne programske tehnologije

Uvod

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Uvod

Porazdeljene inteligentne programske tehnologije, Uvod

Večagentni sistemi

oz. **Porazdeljena umetna inteligenca:**

Pod pojmom Porazdeljena umetna inteligenca razumemo preučevanje, gradnjo, in uporabo večagentnih sistemov, to je sistemov pri katerih več vzajemno delujočih inteligentnih agentov zasleduje enake cilje ali izvajajo enake naloge.

Dolgoročni cilj Porazdeljene umetne inteligence (DAI):

razviti mehanizme in metode, ki bodo omogočale umetnim agentom, da bodo vzajemno delovali tako kot ljudje (ali še boljše) in, da bodo razumeli interakcijo med inteligentnimi sistemi (bodisi ljudmi ali roboti ali obojimi).

Osnovno vprašanje Porazdeljene umetne inteligence:

Kdaj, kako in kateri agenti naj vzajemno delujejo (sodelujejo ali tekmujejo), da bi skupaj dosegli želene cilje?

Porazdeljene inteligentne programske tehnologije, Uvod

2

Agenti

- Agent je računsko enota (program, robot), ki zaznava okolico in aktivno vpliva nanjo.
- Deluje avtonomno in je vsaj deloma odvisen od svojih izkušenj.
- Deluje fleksibilno in racionalno v različnih spreminjajočih se okoljih.
- Fleksibilno obnašanje mu omogočajo njegove bistvene funkcionalnosti:
 - reševanje problemov
 - načrtovanje
 - odločanje
 - učenje
- Bistvena je usmerjenost k izpolnjevanju ciljev in izvrševanju nalog.
- Interakcija, koordinacija med agenti
 - sodelovanje
 - tekmovanje

Lastnosti večagentnih sistemov

- Glavne lastnosti večagentnih sistemov:
 - vsak agent ima samo nepopolno informacijo in je omejen v svojih zmožnostih
 - nadzor sistema je porazdeljen
 - podatki so decentralizirani
 - računanje (obdelava podatkov) je asinhrono

Atributi večagentnih sistemov

- Agenti:

<i>Atribut</i>	<i>Obseg vrednosti</i>
število	od dva naprej
uniformnost	homogeni - nehomogeni
cilji	nasprotni - komplementarni
arhitekture	odzivne - preudarne
Sposobnosti (zaznavanje, spoznavanje,...)	enostavne – napredne

Atributi večagentnih sistemov

- Interakcija:

<i>Atribut</i>	<i>Obseg vrednosti</i>
pogostost	majhna - velika
vztrajnost	kratkoročna - dolgoročna
nivo	posredovanje signalov – posredovanje znanja
vzorec (tok podatkov in nadzora)	decentraliziran – hierarhičen
spremenljivost	nespremenljiva- spremenljiva
namen	tekmovalen – sodelovanje

Atributi večagentnih sistemov

- Okolje:

<i>Atribut</i>	<i>Obseg vrednosti</i>
predvidljivost	predvidljivo – nepredvidljivo
dostopnost in poznanost	neomejena – omejena
dinamičnost	nespremenljivo – spremenljivo
različnost	majhna – bogata
bogastvo z viri	omejeno - obilno

Vrste večagentnih sistemov

- Dve glavni vrsti:
 - Večagentni sistemi, ki koordinirajo njihovo znanje in aktivnosti in razmišljajo o procesu koordinacije
 - Sistemi za porazdeljeno reševanje problemov, kjer se delo potrebno za rešitev nekega problema porazdeli na več vozlišč
- Spoznavni sistemi
 - sestavljeni iz večjega števila specializiranih podsistemov
 - zelo neheterogeni in večmodalni
 - zelo pomembna integracija ter komunikacija med podsistemi

Poglavitna vprašanja

- Kako omogočiti agentom, da bodo znali razdelati njihove cilje in naloge na podcilje in podnalge in jih razdeliti drugim agentom, ter nato spet združiti rezultate skupaj?
- Kako omogočiti agentom komunikacijo? Kakšni jeziki in protokoli naj se uporabijo?
- Kako omogočiti agentom, da predstavijo in sklepajo o akcijah, načrtih in znanju drugih agentov?
- Kako naj agenti predstavijo in sklepajo o stanju njihovi interakcij? Kako naj spremljajo izvajanje svojih akcij in razumejo kdaj so dosegli cilj, vedo kako proces izboljšati, ipd.?
- Kako omogočiti agentom, da najdejo skupni jezik kljub različnim pogledom?
- Kako zgraditi praktične večagentne sisteme?
- Kako zagotoviti dobro razmerje med lokalnim računanjem in komunikacijo?
- Kako se izogniti nepredvidljivem (kaotičnem) obnašanju sistema?
- Kako omogočiti agentom, da se med seboj pogajajo in dogovarjajo?
- Kako omogočiti agentom, da se med seboj povezujejo in tvorijo ekipe?
- Kako formalno opisati večagentne sisteme in interakcijo med njimi?
- Kako realizirati inteligentne procese (reševanje problemov, načrtovanje, odločanje, učenje) v večagentnem kontekstu?

Spoznavni sistemi

Cognitive Systems for Cognitive Assistants - CoSy

The main goal of the project is to advance the science of cognitive systems through a multi-disciplinary investigation of *requirements, design options and trade-offs* for human-like, autonomous, integrated, physical (eg., robot) systems, including requirements for architectures, for forms of representation, for perceptual mechanisms, for learning, planning, reasoning and motivation, for action and communication.

- Človeku podobni, avtonomni, inteligentni, integrirani fizični sistemi (roboti)
- Zelo različne (povezane) funkcionalnosti: zaznavanje, učenje, načrtovanje, sklepanje, motivacija, akcija, komunikacija
- Arhitektura in reprezentacije, ki to omogočajo
- Zelo multidisciplinaren pristop
- Zelo distribuirani, nehomogeni, napredni, komunikativni sistemi, ki naj bi delovali v nepredvidljivem, nenehno spreminjajočem se naravnem (bogatem) okolju

Primer spoznavnega sistema

- Integriran inteligentni sistem – spoznavni sistem
- Različne komponente z različnimi funkcionalnostmi integrirane v enoten inteligenten sistem
- Primer: Projekt Morpha
 - Pogled naprej
 - Scenarij: Robotski asistent za delo v gospodinjstvu in pomoč invalidom in starejšim



Spoznavni sistem

- Kognitivni asistent
 - Razišče okolico in zgradi zemljevid, zna se gibati in izogibati oviram
 - Se nauči prepoznati in identificirati predmete
 - Razume namen in funkcije predmetov ter zna z njimi ravnati
 - Zna verbalno in neverbalno komunicirati z ljudmi v okolici
 - Zazna nove situacije in ustrezno reagira ter načrtuje naslednje akcije
 - Deluje robustno, v vseh pogojih, v realnem domačem okolju
- Vgrajene osnovne funkcionalne sposobnosti, ki jih razvija in nadgrajuje z učenjem
- Vse komponente integrirane v enoten delujoč sistem



Primer večagentnega sistema

- Večagentni porazdeljeni inteligentni sistemi
- Primer: robotski nogomet



Aplikacije multiagentnih sistemov

- v ekonomiji
- v telekominukacijah
- v prometu
- doma
- v vojski
- v vesolju
- v zabavni industriji
- ...

... v ekonomiji

- Elektronsko trgovanje
- Modeliranje in optimizacija trgovinskih poti
- Analiza in optimizacija poslovnih procesov



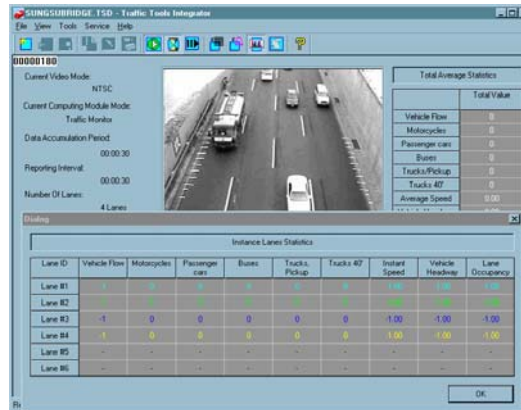
... v telekomunikacijah

- Spremljanje in upravljanje telekomunikacijskih mrež (posredovanje in preklapljanje zvez)
- Upravljanje z informacijami v informacijskih okoljih kot je Internet (zbiranje, filtriranje informacij, itn.)



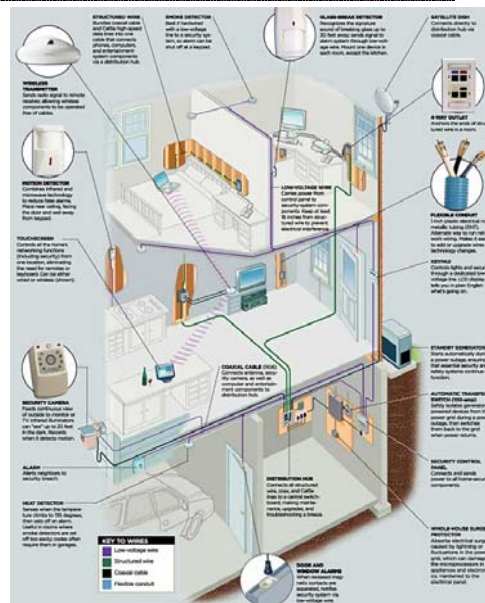
... v prometu

- nadzor cestne infrastrukture, varnost v tunelih
- modeliranje in optimizacije prometnih tokov
- optimizacija logistike
- ...



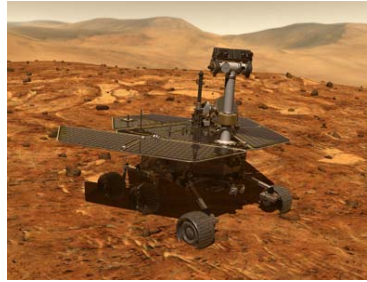
... doma

- Pametna hiša
- Avtomatizacija
 - ogrevanja
 - senčil
 - osvetlitve
 - varnosti
 - ...



... v vojski in vesolju

- Aeronavtika, vesoljske raziskave, vojska
 - Za varno samodejno letenje in pristajanje
 - Večja avtonomija, decentralizirani inteligentni sistemi, decentralizirano odločanje v neznanem, nepredvidljivem in spreminjajočem okolju



... v zabavni industriji

- Računalniške igre



Lastnosti aplikacij

- Porazdeljenost
 - prostorska porazdeljenost
 - časovna porazdeljenost
 - semantična porazdeljenost
 - funkcionalna porazdeljenost
- Kompleksnost

Prednosti

- hitrost in učinkovitost
 - paralelizem, asinhroničnost
- robustnost in zanesljivost
- skalabilnost in fleksibilnost
- nižji stroški
- razvoj in ponovna uporabljivost

Porazdeljene inteligentne programske tehnologije

Inteligentni agenti

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Poglavje 1

Porazdeljene inteligentne programske tehnologije, Inteligentni agenti

Kaj je agent?

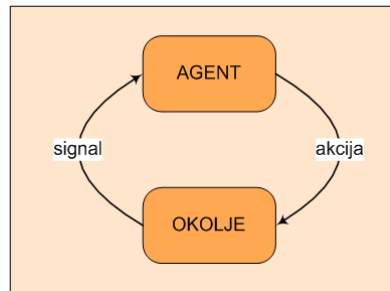
- SSKJ: **agènt** -ènta in -énta m (è`é, è) **1. zastopnik podjetja pri sklepanju kupčij, ki obiskuje stranke na domu: zavarovalni agent; agent za šivalne stroje // samostojen posredovalec kupčij, mešetar: borzni agent; trgovski agent**
2. zaupen sodelavec državne obveščevalne službe: oba agenta sta prisluškovala pogovoru; policijski, tajni agent; agent v civilu / agent provokator ♪
- Wikipedia: Agent je entiteta, ki zaznava in vpliva na okolje skladno z njegovimi aktivnimi lastnostmi ali preferencami in cilji.

Porazdeljene inteligentne programske tehnologije, Inteligentni agenti

2

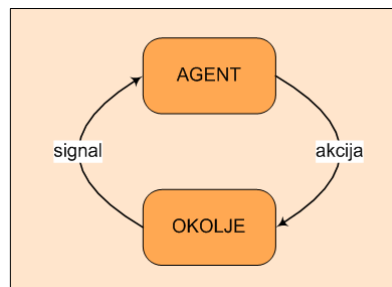
Kaj je agent?

- Agent je računalniški sistem, ki je umeščen v nekem **okolju** in ki je sposoben **samostojno delovati** (izvajati akcije) v tem okolju za doseg zadanih **ciljev**.



Kaj je agent?

- Vhod: signali zaznani s senzorji
- Izhod: akcija s katero agent vpliva na okolje
- Le delni nadzor nad okoljem
 - možnost neuspeha
- Nabor možnih akcij
 - s predpogoji



Okolje

- Kompleksnost agenta je v mnogočem odvisna od okolja, ki ga obdaja in je lahko:
 - Dostopno / nedostopno
 - V dostopne okolju lahko agent kadarkoli dobi popolnoma popolno in natančno informacijo o stanju okolja (precej utopično)
 - Deterministično / nedeterministično
 - V determinističnem okolju enaka akcija vedno da enak odziv
 - Epizoden / ne-epizoden
 - V epizodnem okolju dogajanje temelji na zaporedju med seboj neodvisnih kratkih epizod
 - Statično / dinamično
 - Statično okolje spreminja samo agent, sicer ostaja konstantno
 - Diskretno / zvezno
 - Diskretno okolje vsebuje končno mnogo možnih zaznav (vhodov) in akcij (izhodov)
- V splošnem je realno okolje nedostopno, nedeterministično, ne-epizodno, dinamično in zvezno ☹

Primeri agentov

- Neinteligentni agenti:
- Sobni termostat:
 - Vhod: izmerjena temperatura
 - Akcije: vključiti ali izključiti gretje
 - Odločitveni pravili:
 - Prehladno -> vključiti gretje
 - Temperatura OK -> izključiti gretje
 - Fizično okolje
- Xbiff program
 - Vhod: spremlja prejem nove elektronske pošte
 - Akcija: opozori uporabnika na prejem pošte
 - Programsko okolje



Kaj je inteligenca?

- SSKJ: **inteligénca** -e ž (ę) **1. nadarjenost za umske dejavnosti**: za tako delo je potrebna velika inteligenca; s hitro rešitvijo problema je izpričala svojo inteligenco; izredna, naravna inteligenca; pomanjkanje, preizkus inteligence / s svojo inteligenco je takoj dojel, za kaj gre *bistroumnostjo, bistrostjo*
- Wikipedia: Z inteligenco opisujemo skupno lastnost razuma, ki zaobjema sorodne zmožnosti kot so sklepanje, načrtovanje, reševanje problemov, abstraktno razmišljanje, razumevanje, uporaba jezika in učenje.

Kaj so inteligentni agenti?

- Inteligentni agenti so sposobni **prilagodljivo** in avtonomno izvajati akcije, ki so potrebne za doseg zastavljenih ciljev.
- Prilagodljivost:
 - Odzivnost: zaznavati okolje in zelo hitro odreagirati kadar je to potrebno
 - Pro-aktivnost: agenti stremijo k doseganju zadanih ciljev in temu prilagajajo svoje obnašanje
 - Socialno obnašanje: agenti so sposobni komunicirati z ostalimi agenti (in ljudmi) za doseg svojih ciljev

Inteligentni agenti

- Proaktivni sistemi so relativno enostavni za implementacijo v nespremenljivih okoljih (okolje se med izvajanjem procedure (odločitvene funkcije) ne spremeni).
- V zelo spreminjajočih se okoljih so bolj uspešni odzivni sistemi, ki se hitro odzivajo na spremembe v okolju.
- Pravi izziv: najti pravo razmerje med odzivnostjo in proaktivnostjo!
 - Hočemo, da se agent odziva na spremembe, vendar da se ne odziva in spreminja odločitev bolj pogosto kot je potrebno!
 - Hočemo, da agent zasleduje dolgoročne cilje, vendar jih tudi prilagaja, če je potrebno!
- Socialno obnašanje
 - Za doseg ciljev morajo agenti med seboj komunicirati, sodelovati, se pogajati, itn.

Agenti in objekti

- Objekti v objektno-orientiranem programiranju
 - zaobjemajo neko stanje (vrednosti spremenljivk)
 - izvajajo določene akcije (metode)
 - Komunicirajo med seboj
- Toda:
 - Akcije sprožajo drugi objekti in ne objekt sam; agent se sam odloči ali bo nekaj naredil ali ne
 - Objekti nimajo prilagodljivega avtonomnega obnašanja (s težnjo po zadanem cilju)
 - Agenti delujejo samostojno, neodvisno eden od drugega (v smislu nadzora)

Agenti in ekspertni sistemi

- Ekspertni sistemi rešujejo probleme in svetujejo v neki omejeni domeni znanja
- Toda:
 - Ekspertni sistemi niso utelešeni
 - Ne neposredno zaznavajo okolja
 - Ne neposredno vplivajo na okolje
 - Vse se dogaja preko posrednika
 - Ponavadi ne sodelujejo in komunicirajo z drugimi agenti

Abstraktne arhitekture za inteligentne agente

Formalna definicija agenta:

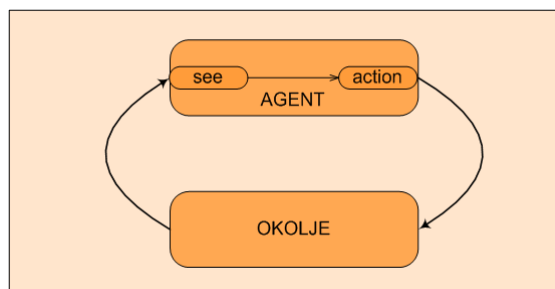
- Množica *stanj okolja*: $S = \{s_1, s_2, \dots\}$
- Množica *razpoložljivih akcij*: $A = \{a_1, a_2, \dots\}$
- *Standardni agent* je funkcija, ki preslika zaporedje stanj okolja v akcije: *action*: $S^* \rightarrow A$
- *Obnašanje okolja*: *env*: $S \times A \rightarrow P(S)$
 - Če je moč vseh $P(S) = 1$, potem je okolje deterministično
- *Zgodovina stanj* (in akcij, ki so povzročile prehod med njimi)
- *Karakteristično obnašanje* agenta v nekem okolju je enako množici vseh možnih zgodovin stanj
- *Invariantna lastnost* je tista, ki je resnična v vseh zgodovinah
- Dva agenta sta *ekvivalentna po obnašanju*, če imata enaka karakteristična obnašanja

Popolnoma odzivni agenti

- Odločajo se samo na osnovi trenutnega stanja, ne glede na preteklost
- Odzivajo se neposredno na okolje
action: S → A
- Primer: termostat

Zaznavanje

- Odločitveno funkcijo razbijemo na podsistema za *zaznavanje* in *akcijo*:
 - *see* zajema zaznavne zmožnosti agenta (senzorji, kamere, sistemski ukazi, ipd.)
 - preslika stanje okolja v *zaznave* (*percepts*) P oz. *see: S → P*
 - *action* zajema odločitveni proces
 - preslika zaporedja zaznav v akcije oz. *action: P* → A*

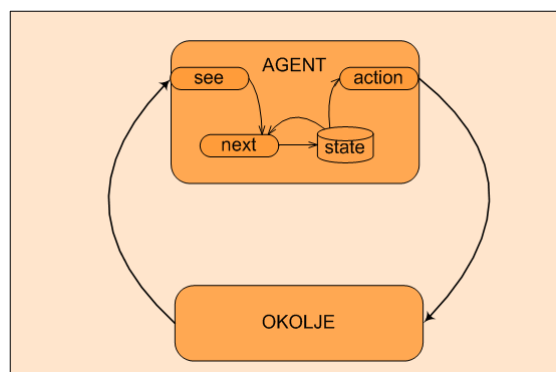


Nerazločljiva stanja

- *Nerazločljivi stanji* sta dve različni stanji, ki sta s strani agenta zaznani na enak način
- Primer s sobnim termostatom:
 - Dva stavka:
 - Temperatura je OK
 - Kosovo je samostojna država.
- Ekvivalenčna relacija, ki razbije množico stanj v množice nerazločljivih stanj
 - *vseveden* agent lahko razloči vsa stanja
 - *neumen* agent – zanj so vsa stanja enaka

Agenti s stanji

- Namesto zgodovine stanj okolja uvedemo *stanje agenta* (I)
 - *see*: $S \rightarrow P$
 - *next*: $I \times P \rightarrow I$
 - *action*: $I \rightarrow A$



Arhitekture za inteligentne sisteme

- Iz abstraktnih arhitektur h konkretnim
- Določimo konkretno notranjo strukturo in operacije agentov
- Štiri razredi agentov:
 - *Agenti osnovani na logiki* temeljijo na logični dedukciji
 - *Odzivni agenti* temeljijo na direktni preslikavi med situacijo in akcijo
 - *Agenti prepričanje-želja-namen* (belief-desire-intention, BDI) temeljijo na manipulaciji s podatkovnimi strukturami, ki predstavljajo prepričanja, želje in namene agenta
 - *Večplastni agenti* sprejemajo odločitve skozi več programskih plasti z različnimi nivoji abstrakcije

Agenti osnovani na logiki

- Tradicionalni pristop k umetni inteligenci (simbolična UI)
 - sistem operira z simboličnimi predstavitvami okolja in ciljev in z njimi sintaktično manipulira
- Nadaljevanje te ideje
 - simbolične predstavitve - logične formule
 - sintaktične manipulacije – logična dedukcija in dokazovanje izrekov
- Agenti so v bistvu dokazovalci izrekov

Preudarni agenti

- Notranje stanje agenta je predstavljeno z množico predikatov logike prvega reda
 - to je informacija, ki jo agent ima o okolju
- L – množica stavkov logike prvega reda
- D – množica množic L
- P – množica zaznav
- A - množica akcij
- *see*: $S \rightarrow P$ preslika stanje okolja v zaznave
- *next*: $D \times P \rightarrow D$ osveži bazo trenutnega stanja agenta
- *action*: $D \rightarrow A$ je definirana z deduktivnimi pravili

Algoritem

- Poglej katero akcijo se lahko izpelje iz trenutnega nabora stavkov, ki opisujejo trenutno stanje
- Poglej katere negacije akcije se ne da izpeljati (je konsistentna s pravili)

Primer

- Svet sesalca
 - Robotski sesalec v majhnem svetu
- Zaznave
 - pozna svojo lokacijo v svetu (x,y)
 - pozna svojo orientacijo (north, south, east, west)
 - zazna umazanijo pod sabo (dirt,null)
- Akcije
 - lahko se premakne naprej za en korak (forward)
 - se obrne za 90° (turn)
 - ali posesa na trenutnem mestu (suck)
- Pravila obnašanja
 - posesaj, če lahko
 - če ne, se premikaj po svetu

Problemi

- Velika računska kompleksnost dokazovalcev izrekov
 - stanje okolja se lahko spremeni med dokazovanjem
- Zaznave morajo biti v obliki simbolov (recimo množica formul)
- V realnem svetu je preslikava signalov v simbole izredno težka (izločanje semantike na sliki)
 - prepad med signali in simboli!
- Zelo težko je predstaviti lastnosti dinamičnega realnega sveta (recimo časovno informacijo)
- Včasih je tudi predstavljanje proceduralnega znanja (kaj narediti) precej okorno

Odzivne arhitekture

- Bistvene lastnosti
 - zavrnitev simbolnih predstavitev
 - ideja, da je obnašanje agenta zelo tesno povezano s samim okoljem; obnašanje ni neutelešeno, je produkt interakcije med agentom in okoljem
 - ideja, da inteligentno obnašanje nastane kot produkt interakcije raznih preprostejših oblik obnašanja
- Poudarek na
 - obnašanju
 - umeščanju v okolje
 - odzivnosti

Vključevalna arhitektura (subsumption arch.)

- Dve karakteristiki:
 - agentovo odločanje je realizirano skozi množico obnašanj, ki izpolnjujejo nalogo
 - situation → action
 - ni kompleksnih simboličnih predstavitev
 - ni simboličnega sklepanja
 - več obnašanj (akcij) se lahko sproži hkrati, nato pa nek mehanizem izbere najboljšo
 - vključevalna hierarhija
 - obnašanja urejena v nivoje z različnimi prioritetami
 - nižji nivoji imajo večjo prioriteto
 - višji nivoji predstavljajo bolj abstraktna obnašanja
- Zelo tesno povezana zaznavanje in akcija
 - slika se ne simbolno intepetira
 - izbira obnašanja temelji na množici obnašanj z relacijo zadrževanja (akcij na višjih nivojih (z nižjo prioriteto))

Algoritem

- Izračunaj množico akcij, ki se lahko sprožijo
- Za vsako izmed teh akcij pogledaj, če obstaja katera, ki jo zadržuje. Če ne, jo vrni.
- Algoritem je računsko precej učinkovit.

Primer

- Raziskovanje oddaljenega planeta; zbiranje vzorcev kamenine
- Scenarij:
 - lokacije vzorcev niso poznane se pa nahajajo v gručah
 - določeno število vozil je na voljo, da se gibljejo po planetu in se vračajo nazaj v matično postajo
 - vozila med seboj ne morejo neposredno komunicirati
- Dva mehanizma (Luc Steels)
 - gradient field: bazna postaja oddaja signal, da vozila vedo kam se vračat (v smeri gradienta)
 - posredna komunikacija: vozila lahko nosijo, spuščajo, detektirajo in pobirajo radioaktivne drobtinice, da ostala vozila vedo, kje se nahajajo kamenine (v nasprotni smeri gradienta)
- Poceni in robustno delovanje

Prednosti in slabosti

- Prednosti vključevalne arhitekture
 - enostavnost, ekonomičnost, praktična računska izvedljivost, robustnost, eleganca
 - enostavno je izdelati nekompleksne agente s preprosto interakcijo z okoljem
- Pomanjkljivosti
 - Če objekti ne modelirajo okolja, morajo imeti dovolj informacije o vsem okolju v lokalnem okolju, ki ga lahko zaznajo
 - Uporabljajo samo lokalno in trenutno informacijo
 - se ne morejo učiti na izkušnjah in izboljševati svoje sposobnosti skozi čas
 - Težko je izdelati takšne agente, ker je težko ugotoviti kako in kakšne akcije, relacije, in splošna obnašanja lahko nastanejo neposredno kot produkt okolja
 - Zelo težko je izdelati kompleksne agente z veliko dinamiko pri komunikacijah

Agenti prepričanje-želja-namen (BDI)

- Belief-desire-intention, BDI agenti
- Korenine ima v praktičnem sklepanju
 - proces odločanja kdaj naj kaj naredimo, da izpolnimo cilje
 - Uporaba sklepanja, da se odločimo kako ukrepati
- Dva procesa
 - *kakšne* cilje naj dosežemo (premišljanje; deliberation)
 - *Kako* bomo dosegli te cilje (means-ends reasoning)

Namen

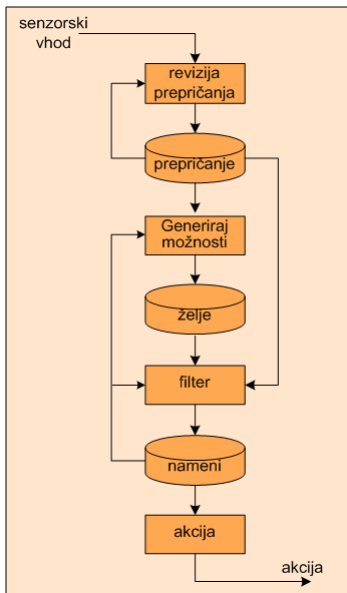
- Definiramo možnosti, izberemo eno izmed njih, se ji zavežemo, postane naš *namen*
- Namen vodi k akciji
- Namen določa način kako bomo dosegli cilje
- Namen omeji nadaljnjo praktično sklepanje
- Namen traja (dolgo časa, vendar ne predolgo)
- Nameni so zelo povezani z našimi *prepričanji* o prihodnosti

- Včasih moramo o namenu ponovno razmisliti
- Problem je najti dobro razmerje med
 - preredkim in
 - prepogostim ponovnim razmišljanjem
- Podobno kot najti pravo razmerje med
 - pro-aktivnimi in
 - odzivnim obnašanjem

Strategije odločanja

- Razmerje med zavezo in ponovnim razmislekom
 - Drzni agenti (se nikoli ne ustavijo in ponovno premislijo)
 - Previdni agenti (to delajo zelo pogosto)
- Stopnja sprememb sveta γ
 - Majhna (okolje se ne hitro spreminja): boljši drzni agenti
 - Velika (okolje se hitro spreminja): boljši previdni agenti
- Različni tipi okolja zahtevajo različne strategije odločanja:
 - Statično okolje: boljše čisto pro-aktivno k cilju usmerjeno obnašanje
 - Dinamično okolje: potrebuje tudi zmožnost reagiranja in hitrega odziva

Splošna BDI arhitektura



Prepričanje -
trenutna informacija o svetu

Želje -
možne smeri akcij

Nameni -
trenutni fokus

Proces odločanja

- Revizija prepričanja
- Generiranje možnosti
 - Proces odločanja *kako* doseči trenutne namene
 - Rekurzivno razdela strukturo hierarhičnega načrta – bolj konkretizira trenutne namene (do izvršljivih akcij)
 - Funkcija generiranja možnosti mora biti:
 - Konsistentna (s trenutnimi prepričanji in nameni)
 - Oportunistična (mora zaznati in izkoristiti spremembe v okolju)
- Funkcija Filter
 - Proces odločanja *kaj* narediti
 - Mora opustiti namene, ki niso več mogoči ali smiselni
 - Mora ohraniti namene, ki še niso izpolnjeni
 - Mora dodati nove namene, če je to potrebno
- Funkcija Akcija
 - Vrne direktne izvršljive akcije

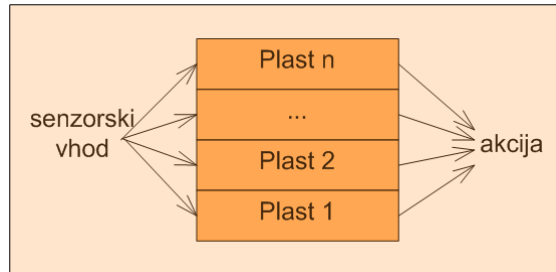
Prednosti in slabosti

- Nameni imajo lahko različne prioritete
- Prednosti
 - Intuitiven model
 - Jasna funkcionalna dekompozicija
- Slabosti
 - Težko je vedeti kako učinkovito implementirati funkcije

Večplastni agenti

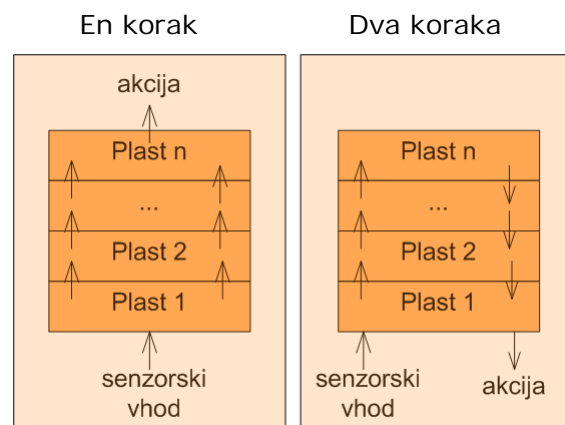
- Če hočemo hkrati imeti funkcionalnost pro-aktivnega in odzivnega agenta -> Dekompozicija na ločene podsisteme
- Hierarhija vzajemno delujočih plasti
- Dva tipa toka podatkov
 - Horizontalne plasti
 - Vsaka plast je povezana z senzorskim vhodom in akcijskim izhodom
 - Vsaka plast zase je agent, ki predlaga akcijo
 - Vertikalne plasti
 - Senzorski vhod in akcijski izhod sta v stiku z največ eno plastjo

Horizontalne plasti



- n različnih hkrati delujočih tekmovalnih plasti
-> nevarnost, da skupno delovanje postane nekoherentno
- *Mediator* – funkcija, ki odloča katera plast prevzame nadzor nad sistemom v nekem trenutku
- Mora upoštevati vse možne interakcije med vsemi plastmi – zelo potratno
- Kontrolni sistem predstavlja ozko grlo agentovega odločanja

Vertikalne plasti



- Kompleksnost interakcije med plastmi je zmanjšana
- Nadzorni tok gre skozi vse plasti - nerobustno

Porazdeljene inteligentne programske tehnologije

Spoznavni sistemi

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Porazdeljene inteligentne programske tehnologije, Spoznavni sistemi

Spoznavnost

- Spoznavni sistem = kognitivni sistem = cognitive system
- SSKJ:
 - **spoznáven** -vna -o prid. (á ā) *nanašajoč se na spoznavanje, spoznanje*: spoznavni nagibi; razčleniti spoznavni proces / človekove spoznavne sposobnosti ... filoz. spoznavna teorija *filozofska disciplina, ki obravnava izvor, strukturo, metodo spoznavanja in veljavnost spoznanja*
 - **spoznávati** -am nedov. (á) **1.** *na osnovi zaznav, podatkov in umske dejavnosti prihajati do a) poznavanja česa*: človek vse bolj spoznava naravo; spoznavati, kako deluje celica / v šoli spoznavati prve črke; spoznavati računalništvo **b) vedenja**: spoznavati lastne zmote; vse bolj je spoznaval, da tako ne more več živeti / spoznavati otrokovo nadarjenost

Porazdeljene inteligentne programske tehnologije, Spoznavni sistemi

2

Cognition

- Britannica Concise Encyclopedia:
Act or process of **knowing**. Cognition includes every mental process that may be described as an experience of knowing (including **perceiving, recognizing, conceiving, and reasoning**), as distinguished from an experience of feeling or of willing.
- Sci-Tech Encyclopedia:
The internal structures and processes that are involved in the acquisition and use of **knowledge**, including **sensation, perception, attention, learning, memory, language, thinking, and reasoning**. Cognitive scientists propose and test theories about the functional components of cognition based on observations of an organism's external behavior in specific situations.

Cognition

- Wikipedia:
The term **cognition** (Latin: cognoscere, "to know") is used in several loosely related ways to refer to a faculty for the **human-like processing of information**, applying **knowledge** and changing preferences. Cognition or cognitive processes can be natural and artificial, conscious and not conscious; therefore, they are analyzed from different perspectives and in different contexts, in *anesthesia, neurology, psychology, philosophy, systemics and computer science*. The concept of cognition is closely related to such abstract concepts as **mind, reasoning, perception, intelligence, learning**, and many others that describe numerous capabilities of the human mind and expected properties of artificial or synthetic intelligence. Cognition is an **abstract property** of advanced living organisms; therefore, it is studied as a direct property of a brain or of an abstract mind on sub-symbolic and symbolic levels.

Cognition

- Wikipedia:

In psychology and in artificial intelligence, it is used to refer to the **mental functions, mental processes** and **states of intelligent entities** (humans, human organizations, highly autonomous robots), with a particular focus toward the study of such mental processes as **comprehension, inferencing, decision-making, planning and learning** (see also cognitive science and cognitivism). Recently, advanced cognitive researchers have been especially focused on the capacities of **abstraction, generalization, concretization/specialization** and **meta-reasoning** which descriptions involve such concepts as **beliefs, knowledge, desires, preferences** and **intentions** of intelligent individuals/objects/agents/systems.

Cognitive science

- Wikipedia:

Cognitive science is most simply defined as the scientific study either of **mind** or of **intelligence**. It is an interdisciplinary study drawing from relevant fields including **psychology, philosophy, neuroscience, linguistics, anthropology, computer science**, and **biology**.

Spoznavna robotika

- Wikipedia:

Cognitive robotics is concerned with endowing **robots** with mammalian and **human-like cognitive capabilities** to enable the achievement of complex goals in complex environments. Robotic cognitive capabilities include **perception processing, attention allocation, anticipation, planning, reasoning about other agents,** and perhaps reasoning about their **own mental states**. Robotic cognition embodies the **behaviour of intelligent agents** in the **physical world**.

- A cognitive robot should exhibit:
 - knowledge
 - beliefs
 - preferences
 - goals
 - informational attitudes
 - motivational attitudes (observing, communicating, revising beliefs, planning)

Enota EU cognition



- **Research Rationale:**
By promoting research into systems that have cognitive functions normally associated with people or animals and which exhibit a high degree of **robustness** in coping with **unpredictable situations**, we seek to overcome limitations of today's computers, robots, and other man-made creations to handle simple **everyday situations** with common sense and to work **without pre-programming** in **natural surroundings**, while maintaining and possibly improving the quality of their services.
- **Unit Mission:**
We support research on the construction of **artificial cognitive systems** than can **interpret information** (images, text, speech, video footage) and other forms of sensor data, and **act purposefully** and **autonomously** towards achieving **goals**.
These systems should **learn** and **develop** through individual or social **interaction** with their **environment**. The work should provide an enabling technology that applies across domains such as natural *language understanding, image recognition, automated reasoning and decision support, robotics and automation, sensing and process control, and complex real-world systems*. The work should furthermore borrow insights from the *bio-sciences*, and yield innovative insights about perception, understanding, interaction, learning and knowledge representation.

Definicije raziskovalcev

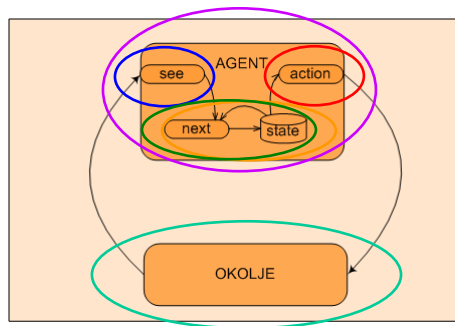
- Cognition is the ability to relate perception and action in a meaningful way determined by experience, learning and memory. *Mike Denham*
- A cognitive system possesses the ability of self-reflection (or at least self-awareness). *Horst Bischof*
- Cognition is gaining knowledge through the senses. *Majid Mermehdi*
- Cognition is the ability to ground perceptions in concepts together with the ability to manipulate concepts in order to proceed toward goals. *Christian Bauckhage*
- An artificial cognitive system is a system that is able to perceive its surrounding environment with multiple sensors, merge this information, reason about it, learn from it and interact with the outside world. *Barbara Caputo*
- Cognition is self-aware processing of information. *Cecilio Angulo*
- Cognitive Systems are ones that are able to extract and (most importantly) represent useful aspects of largely redundant, possibly irrelevant sensory information in a form that is most conducive to achieving a particular high level goal. *Sethu Vijayakumar*
- A cognitive system is a system that can change its behaviour based on reasoning, using observed evidence and domain knowledge. *Bob Fisher*
- Cognition is when I know what I am doing, when I can judge how good or bad it is, and explain why I am doing it. *Markus Vincze*
- Cognition is the ability to plan, reason, adapt and act according to high level motivations or goals and using a range of senses, typically including vision, and may be communicate. *Patrick Courtney*
- A cognitive system is an autonomous anti-entropy engine. *David Vernon*

Definicije raziskovalcev

- Cognition is the ability to relate perception and action in a meaningful way determined by experience, learning and memory. *Mike Denham*
- A cognitive system possesses the ability of self-reflection (or at least self-awareness). *Horst Bischof*
- Cognition is gaining knowledge through the senses. *Majid Mermehdi*
- Cognition is the ability to ground perceptions in concepts together with the ability to manipulate concepts in order to proceed toward goals. *Christian Bauckhage*
- An artificial cognitive system is a system that is able to perceive its surrounding environment with multiple sensors, merge this information, reason about it, learn from it and interact with the outside world. *Barbara Caputo*
- Cognition is self-aware processing of information. *Cecilio Angulo*
- Cognitive Systems are ones that are able to extract and (most importantly) represent useful aspects of largely redundant, possibly irrelevant sensory information in a form that is most conducive to achieving a particular high level goal. *Sethu Vijayakumar*
- A cognitive system is a system that can change its behaviour based on reasoning, using observed evidence and domain knowledge. *Bob Fisher*
- Cognition is when I know what I am doing, when I can judge how good or bad it is, and explain why I am doing it. *Markus Vincze*
- Cognition is the ability to plan, reason, adapt and act according to high level motivations or goals and using a range of senses, typically including vision, and may be communicate. *Patrick Courtney*
- A cognitive system is an autonomous anti-entropy engine. *David Vernon*

Glavni poudarki

- Zaznavanje (perception)
- Akcija (action)
- Sklepanje, načrtovanje (reasoning, planning)
- Cilji (goals)
- Avtonomija, samozavedanje (autonomy, self-awareness)
- Okolje (environment)



Zahteve za spoznavne sisteme

- Kakšne zahteve morajo izpolnjevati spoznavni sistemi:
 - Arhitektura
 - Zaznavanje
 - Predstavitve
 - Učenje
 - Razpoznavanje
 - Akcija
 - Načrtovanje
 - Sklepanje
 - Komunikacija

Arhitektura

- Arhitektura:
 - Odzivna
 - Proaktivna
 - Fleksibilna
 - Učinkovita
 - Skalabilna/razširljiva
 - Mora povezovati različne komponente (zelo nehomogen sistem)
 - Samozavedanje in samorazumevanje
 - Se jo da praktično realizirati/vzdrževati/razvijati,...

Zaznavanje

- Zaznavanje:
 - Vizualna informacija (slika, video; barvna, ČB, IR,...)
 - Zvok (govor, glasba, šum, ...)
 - Haptična informacija - dotik (haptični senzorji, senzorji trka, itn)
 - Globinska/prostorska informacija (globinske slike, 3D modeli, 3D zemljevidi, ...)
 - Veliko različnih modalnosti - spoznavni sistem je zelo večmodalen sistem
- Pozornost
 - Selektivno zaznavanje
 - Obvladovanje kompleksnosti potencialnih vhodnih signalov

Predstavitve

- Predstavitve (reprezentacije) zaznav, sveta, notranjega stanja, načrtov, komunikacije, motivov, itn.
 - **Modalne** predstavitve - predstavitve zaznav (vseh modalnosti), vezane na določeno modalnost
 - **Amodalne** predstavitve – abstrakcija modalnih predstavitev
- Spoznavni sistem mora znati povezovati različne predstavitve ter jih abstrahirati na višji abstrakcijski (semantični) nivo - **binder**
- Dodatne zahteve:
 - Inkrementalno osveževanje predstavitev
 - Različni načini učenja
 - Skalabilnost
 - Primernost za sklepanje in načrtovanje
 - Omogočajo introspekcijo, detekcijo neznanja
- Prirojeno: naučeno

Učenje

- Učenje
 - Različni načini učenja
 - Usmerjano
 - Deloma usmerjano
 - Popolnoma samostojno
 - Kontinuirano učenje
 - Vseživljensko, neprestano učenje
 - Osveževanje predstavitev, tvorjenje novih
 - Detekcija neznanja
 - Učenje v vseh modalnostih
 - Združevanje rezultatov
 - "co-learning"
 - Pozabljanje, popravljanje napak ("unlearning")
 - Robustno, v vsakdanjih pogojih

Razpoznavanje

- Razpoznavanje
 - Objektov
 - Prostorov
 - Akcij
 - Funkcionalnih lastnosti predmetov
 - Govora
 - Namenov,...
- Kategorizacija (razpoznavanje kategorij)
- Večmodalno razpoznavanje

Akcija

- Cikel zaznava-akcija (perception-action)
 - Učinkovitost
 - Robustnost
 - Delovanje v nepredvidljivem okolju
 - Na voljo le delna informacija
 - Fleksibilnost
 - Utelešenost (embodiment)
 - Umeščenost v prostor (situatedness)
- Manipulacija s predmeti (robotska roka)
- Premikanje po prostoru (mobilni robot)

Načrtovanje

- Načrtovanje
 - V nepredvidljivem okolju
 - Brez popolne informacije
 - Z določenimi omejitvami robota
 - V spreminjajočem se okolju
- Kontinuirano načrtovanje
 - Prilagajanje na spremembe v okolju nastale med izvrševanjem načrta
 - Združevanje načrtovanja in izvajanja
- Aktivna detekcija napak
 - Robotska roka
 - Mobilni robot
- Sodelovanje med agenti
 - Sodelovanje/komunikacija s človekom
 - Sodelovanje/komunikacija z drugimi agenti

Sklepanje

- Sklepanje
 - V nepredvidljivem okolju
 - Brez popolne informacije
 - Z določenimi omejitvami robota
 - V spreminjajočem se okolju
- Fleksibilnost in prilagodljivost
- Upoštevanje različnih modalnosti
- Amodalne predstavitve
- Samozavedanje, introspekcija, detekcija neznanja
- Komuniciranje znanja, neznanja

Komunikacija

- Komunikacija
 - S človekom
 - Z drugimi (drugičnimi) agenti
 - V določenem okolju in času
- Prenos znanja
- Razčiščevanje razumevanja
- Koordinacija
- Prezemanje iniciative v dialogu
- Prizemljevanje simbolov - Symbol grounding
- Semantično opisovanje zaznav
- Učenje jezika
 - sintaksa
 - širjenje ontologije
- Učenje z uporabo jezika

Primer spoznavnega sistema

- Hišni robot Robi
- Ukažemo mu: "Prinesi mi pivo".



Primer

- Sosledje dogodkov:
 - Robot mora biti pozoren in poslušati za naš ukaz. [*pozornost, motivacija*]
 - Mora nas slišati in razumeti naš ukaz. [*zaznavanje, razpoznavanje govora, komunikacija*]
 - Postaviti si mora cilj in težiti k temu, da ga izpolni. [*cilj, proaktivnost*]
 - Mora vedeti kje se pivo nahaja, to se je moral prej naučiti. [*učenje*]
 - Mora narediti načrt kako nam bo prinesel pivo. [*načrtovanje*]
 - Mora poiskati najboljšo pot do hladilnika, na osnovi zemljevida, ki si ga je prej zgradil. [*navigacija, gradnja zemljevidov*]
 - Mora se premikati po načrtovani poti. [*akcija – premikanje*]
 - Po poti mora neprestano opazovati kam se giba. [*zaznavanje, akcija*]
 - Po poti se mora izogibati oviram. [*zaznavanje nevarnosti, ponovno načrtovanje, odzivnost*]

Primer

- Ko pride do hladilnika, se mora pravilno postaviti pred njega. [*utelešenost, umeščenost v prostor*]
- Mora znati odpreti hladilnik. [*razpoznavanje funkcionalnih lastnosti*]
- V hladilniku mora znati poiskati pivo (njegov izgled se je moral prej naučiti). [*zaznavanje, kategorizacija, učenje*]
- Načrtovati mora kako ga bo zagrabil. [*načrtovanje*]
- Na pravilen način bo zagrabil steklenico. [*akcija, vizualni nadzor, haptični nadzor*]
- Obrnil se bo in po obratni poti odšel nazaj do nas. [*načrtovanje, navigacija, akcija, zaznavanje nevarnosti, zaznavanje, razpoznavanje*]
- Robi: "Izvoli tvoje pivo". [*komunikacija*]

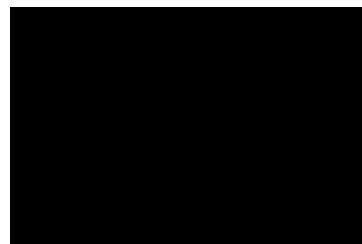
Primer spoznavnega sistema

- Kognitivni asistent
 - Razišče okolico in zgradi zemljevid, zna se gibati in izogibati oviram
 - Se nauči prepoznati in identificirati predmete
 - Razume namen in funkcije predmetov ter zna z njimi ravnati
 - Zna verbalno in neverbalno komunicirati z ljudmi v okolici
 - Zazna nove situacije in ustrezno reagira ter načrtuje naslednje akcije
 - Deluje robustno, v vseh pogojih, v realnem domačem okolju
- Vgrajene osnovne funkcionalne sposobnosti, ki jih razvija in nadgrajuje z učenjem
- Vse komponente integrirane v enoten delujoč sistem



Primer spoznavnega sistema

- DARPA Urban Challenge
- Avtonomni avtomobili
- Vožnja po mestu
- Sposobnosti
 - Zaznavanje (slika, 3D, trk)
 - Načrtovanje
 - Sklepanje
 - Učenje
 - Navigacija
 - Izogibanje oviram
 - Akcija
 - Fleksibilnost
 - Robustnost
 - Učinkovitost
 - ...



Spoznavni porazdeljeni sistemi

- Spoznavni sistemi so v splošnem porazdeljeni
 - Funkcionalno
 - Semantično
 - Fizično
- Posamezni deli
 - so zelo različni
 - uporabljajo različne podatke
 - delujejo asinhrono
- Spoznavni sistem kot nehomogen porazdeljen inteligen agent
 - Sestavljen iz množice podsistemov, ki komunicirajo med seboj in težijo k istemu cilju
- Množica spoznavnih sistemov kot porazdeljen večagentni sistem
 - Sestavljen iz množice inteligentnih agentov, ki komunicirajo med seboj, sodelujejo in težijo k istemu cilju

Atributi spoznavnih sistemov

- Agenti:

<i>Atribut</i>	<i>Obseg vrednosti</i>
število	od ena naprej
uniformnost	homogeni - nehomogeni
cilji	nasprotni - komplementarni
arhitekture	odzivne - preudarne
sposobnosti (zaznavanje, spoznavanje,...)	enostavne – napredne

Atributi spoznavnih sistemov

- Interakcija:

<i>Atribut</i>	<i>Obseg vrednosti</i>
pogostost	majhna - velika
vztrajnost	kratkoročna - dolgoročna
nivo	posredovanje signalov – posredovanje znanja
vzorec (tok podatkov in nadzora)	decentraliziran – hierarhičen
spremenljivost	nespremenljiva- spremenljiva
namen	tekmovalen – sodelovanje

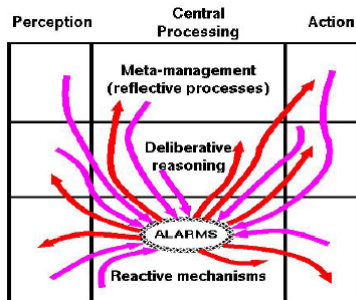
Atributi spoznavnih sistemov

- Okolje:

<i>Atribut</i>	<i>Obseg vrednosti</i>
predvidljivost	predvidljivo – nepredvidljivo
dostopnost in poznanost	neomejena – omejena
dinamičnost	nespremenljivo – spremenljivo
različnost	majhna – bogata
bogastvo z viri	omejeno - obilno

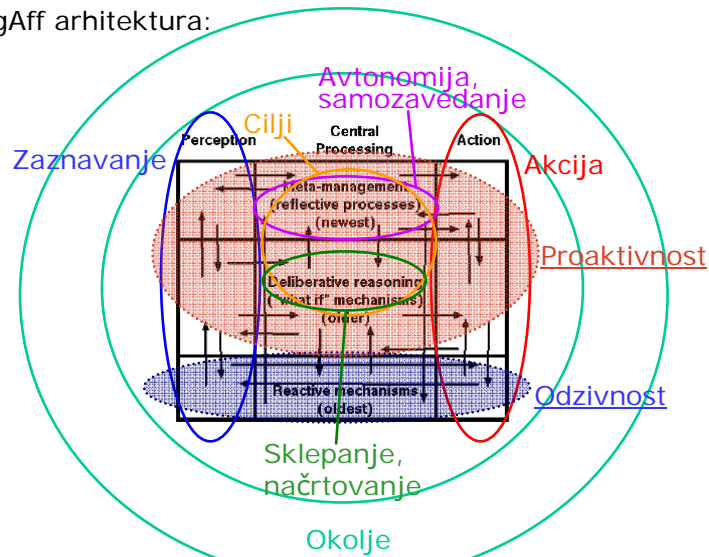
Arhitektura spoznavnih sistemov

- Veliko predlogov: SOAR, ACT, PRODIGY, ICARUS, 3T, APEX, CLARION, CIRCA, EPIC, itn.
- CogAff arhitektura:



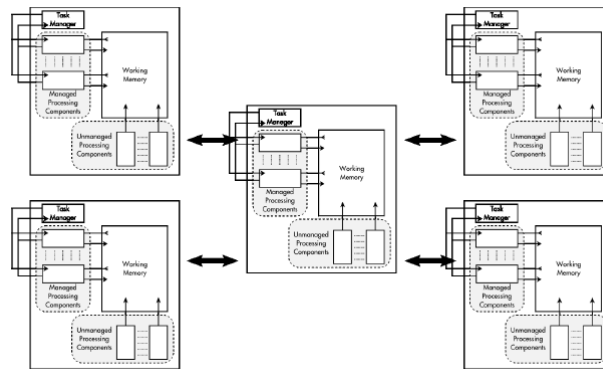
Arhitektura spoznavnega sistema

- CogAff arhitektura:



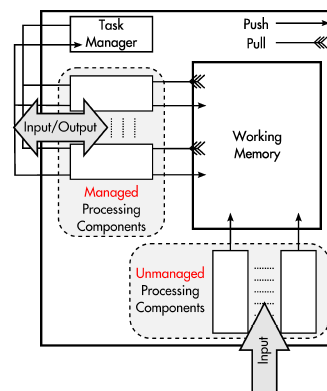
Arhitektura CAS

- CoSy architecture schema
- Zbirka rahlo povezanih podarhitektur
- Zelo različne komponente – zelo nehomogen sistem



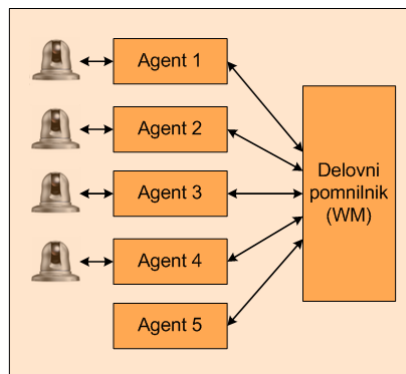
Arhitektura CAS

- Ena podarhitektura:
 - Komponente
 - Osvežujejo strukture v delovnem pomnilniku (WM)
 - Komponente komunicirajo preko delovnega pomnilnika
 - Podarhitekture kmunicirajo z drugimi podarhitekturami preko delovnega pomnilnika
 - Upravitelj (task manager) upravlja s procesiranjem



PIPT CAS arhitektura

- CAS za komunikacijo med agenti



Porazdeljene inteligentne programske tehnologije (PIPT)

Nehomogenost spoznavnih sistemov

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Porazdeljene inteligentne programske tehnologije

Nehomogenost, večmodalnost

- Spoznavni sistemi so porazdeljeni in zelo nehomogeni
- Posamezne komponente imajo na vhodu zelo različne signale in operirajo z zelo različnimi predstavitvami

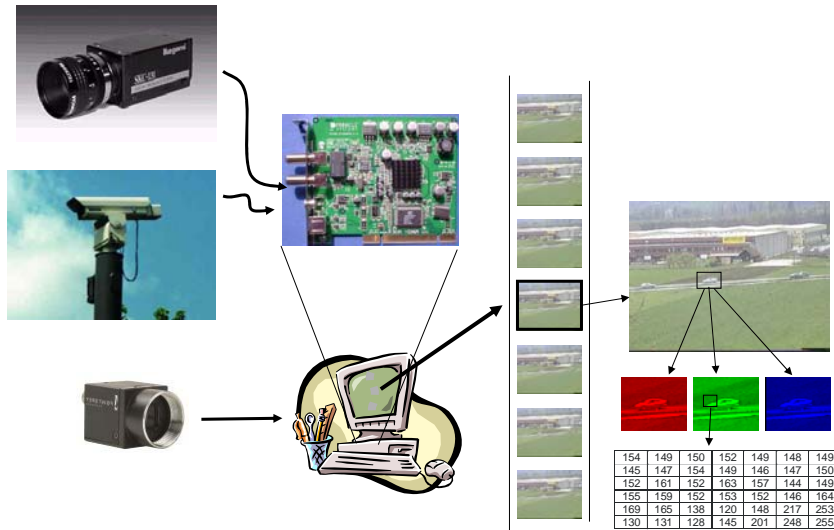
- Kako te predstavitve združevati, kako vedeti katere sodijo skupaj?
-> **Povezovalnik (Binder)**
- Kako se te predstavitve naučiti?
-> **Kontinuirano večmodalno učenje**
- Kakšna arhitektura podpira takšne večmodalne procese?
-> **CAS arhitektura, CAST**



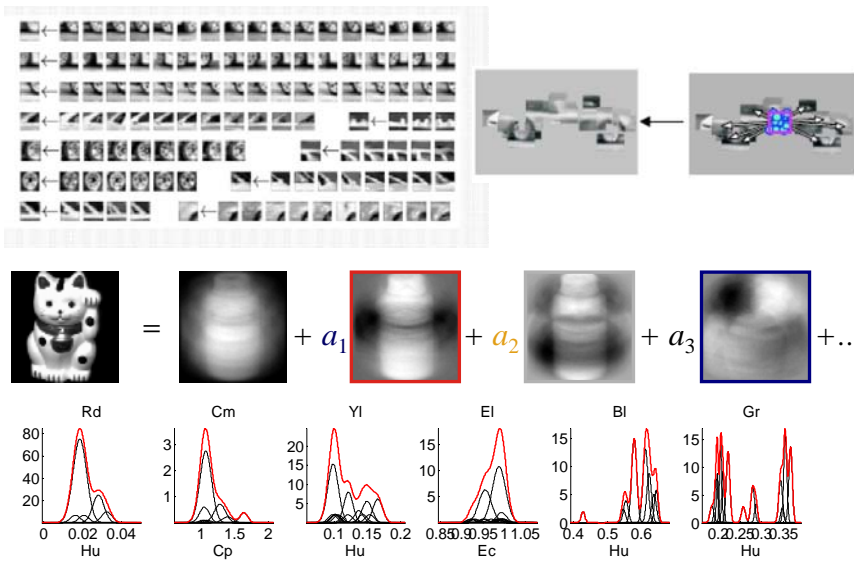
Porazdeljene inteligentne programske tehnologije, Nehomogenost spoznavnih sistemov

2

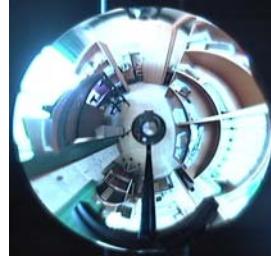
Digitalizacija videa



Predstavitve vizualne informacije



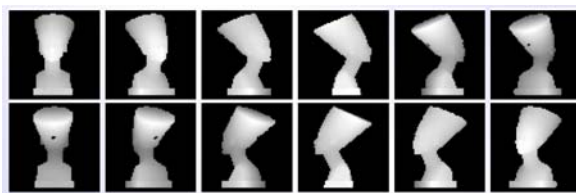
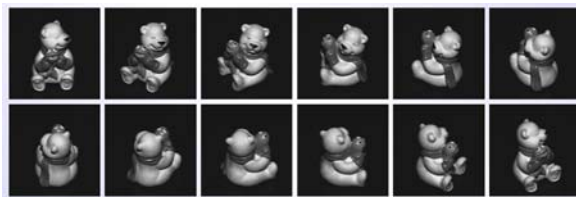
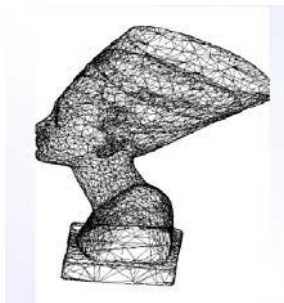
Predstavitve vizualne informacije



Porazdeljene inteligentne programske tehnologije, Nehomogenost spoznavnih sistemov

5

Predstavitve vizualne informacije

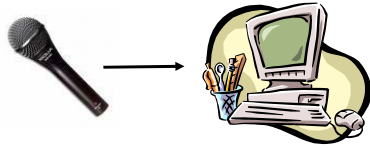


Porazdeljene inteligentne programske tehnologije, Nehomogenost spoznavnih sistemov

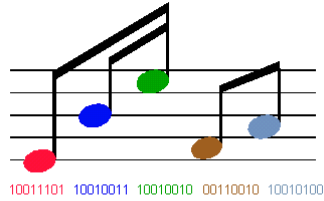
6

Predstavitev avdio informacije

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.



MIDI (digital notes)

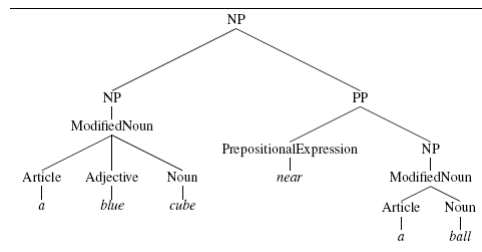


Digital Audio (digitized sound waves)



```
10011101 0010100011101101 1001101011001101010101100110
0010111100101011101000101011011100011001110101101
010111010110101010001001010111010001010110101
100010110111010101011010001001010101110101000
1010011001011010110101100010101010101010101101
0010111010101010101010111010101101101110100
01011101011010110101101011010110101100010101
```

Predstavitev lingvistične informacije



```
S → Command | Statement | Question | S Conjunction S
Command → VP
Statement → NP VP
NP → Pronoun | Modified_Noun | NP RelClause | NP PP | NP
Conjunction NP
Modified_Noun → Noun | Article Noun | Adjective Noun | Article
Adjectives Noun
Noun → Noun_Singular | Noun_Plural
PP → PrepositionalExpression NP
RelClause → RelPronoun VP
```


Sklepanje in načrtovanje

```
(:action move
:parameters (?a - agent ?to - location ?d - door)
:variables (?from - location)
:precondition (and
  (pos ?a : ?from)
  (doorstate ?d : open)
  (entrance ?d ?from) (entrance ?d ?to))
:effect (pos ?a : ?to))
```

```
function CONTINUAL-PLANNING-AGENT(S, G)
while S does not satisfy G do
  if res(S, P) does not satisfy G
    REMOVEOBSOLETE_SUFFIXFROM(P)
    P' = PLANNER(A, res(S, P), G)
    P = CONCAT(P, P')
  if P =  $\emptyset$  then
    return "cannot achieve goal G"
  a = REMOVEFIRSTLEVELACTION(P)
  EXECUTE(a)
  perc = GETSENSORDATA()
  S = UPDATESTATE(S, perc)
return "goal reached"
```

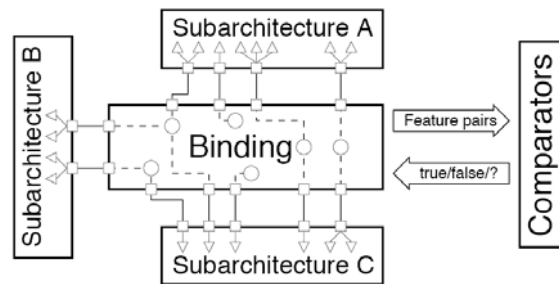
Aktuatorji

- Krmiljenje aktuatorjev
- Povratna informacija
- PTZ kamere
- Robotske roke
- Mobilni roboti...



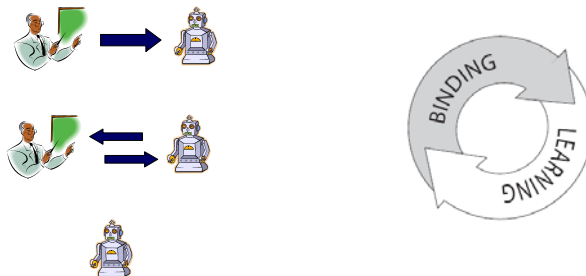
Povezovalnik

- Povezovalnik (binder)
 - povezuje različne modalnosti
 - Ugotavlja katere predstavitve se nanašajo na iste entitete v prostoru



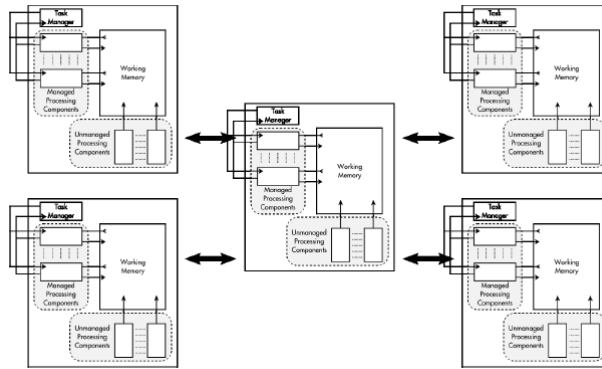
Kontinuirano učenje

- Kontinuirano večmodalno učenje
 - V dialogu s človekom
 - V povezavi s povezovalnikom



Arhitektura sistema

- Arhitektura CAS
- Orodje CAST
- Omogoča distribuiranost, nehomogenost, multimodalnost



Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

Crossmodal Content Binding in Information-Processing Architectures

Henrik Jacobsson¹, Nick Hawes², Geert-Jan Kruijff¹,
Jeremy Wyatt²

¹Language Technology Lab, DFKI GmbH, Germany

²School of Computer Science, University of Birmingham, UK

Dec 2007, Aveiro, LangRo symposium

Outline

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- 1 Introduction
- 2 The Problem
- 3 Motivation
- 4 The Binder
- 5 Discussion

The CoSy Project

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

EU FP6 IST Cognitive Systems Integrated project Cognitive Systems for Cognitive Assistants - CoSy

The main goal of the project is to advance the science of **cognitive systems** through a **multi-disciplinary** investigation of requirements, design options and trade-offs for human-like, autonomous, **integrated**, physical (eg., robot) systems, including requirements for architectures, for forms of **representation**, for perceptual mechanisms, for learning, planning, reasoning and motivation, for action and communication

The CoSy Architecture Schema Toolkit (CAST)

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- One of the main focus of the research in CoSy is to investigate the design space of cognitive robotics
- The architecture toolkit aims at making it possible to investigate a range of possible *instances* of architectures
- An architecture consists of several uniformly designed subarchitectures dedicated to vision, planning, communication, mapping etc.
- The main challenge is the integration effort
 - How to communicate between subarchitectures
 - What to communicate between subarchitectures
 - What to do with information from other subarchitectures
 - When to communicate

CAST Example

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

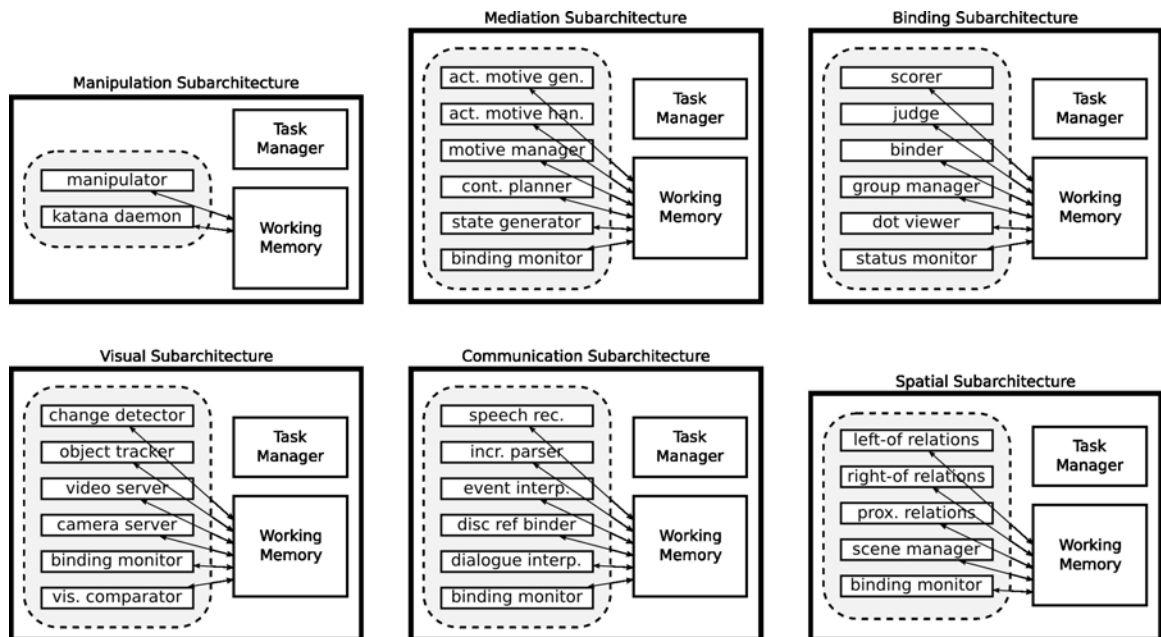
Introduction

The Problem

Motivation

The Binder

Discussion



The Binding Problem

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- a.k.a. symbol grounding
 - but... “[The binder does] not explicitly deal with reality”
- That will not eliminate all problems:
 - 1 Find a common ground for representing information from different sensory modalities and deliberative processes
 - 2 Find a format that facilitates integration of existing and future implementations of subarchitectures
 - 3 Consider other binding problems than those related to language
 - 4 Robustness against “chaotic” dynamics important

Requirements

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- 1 Appropriate level of abstraction
- 2 Nonintrusive and simple
- 3 Stable symbols
- 4 Asynchronous, anytime, incremental production of bindings

Requirements

1. Level of abstraction

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- Dilemma: amodal or modal information? Both make sense! So we support both!
- The entities in our scenarios typically involve
 - Objects (and groups of objects)
 - Actions
 - Relations
- We represent these as entities with sets of describing properties, *binding features*
- These entities are called *proxies* (why? hang on)
- The information fusion of crossmodal contents build upon the assumption that subarchitectures have proxies that may refer to the same entity
 - Independent of temporal or spatial frame

Requirements

2. Nonintrusiveness

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- A subarchitecture only needs to provide:
 - Binding feature *definitions*
 - A binding *monitor* component which create appropriate proxies
 - *Comparators*, that compare pairs of features

Requirements

Nonintrusiveness of Binding Features

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- A subarchitecture can have very specialized representations, e.g.
 - visual features
 - spatial representations
 - linguistic modifiers
 - etc.
- The depth of description would be restricted without them
- Translation into common description is costly and lossy
- A binding feature can therefore in principle be *anything*
 - I.e. anything you can represent in a Java or C++ class
 - If your subarchitecture come across a feature which it doesn't understand, it can only ignore it

Requirements

The Relative Intrusiveness of Binding Monitors

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- The monitors should react to internal data, and make a proper presentations of it in the form of proxies
 - Intramodular binding (e.g. discourse referents or spatial reasoning)
 - Present the currently best hypothesis about objects, actions and relations (i.e. possibly incrementally)
 - Monitors can be context aware (e.g. to withhold irrelevant data)
- Monitors should present data that is likely to be relevant to the task... (not so easy)

Requirements

The Relative Intrusiveness of Comparators

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- If a new feature is added, a function which compare that type feature to other comparable types *should* also be added
 - These functions are called *comparators*
 - In current implementation, they should return *true*, *false* or *indeterminate* for every pair of feature instances (brutally simple)
- The result from the comparisons is the basis for the *binding score* which in turn decides which proxies may in fact refer to the same entity
- Comparators may be based on anything, e.g.
 - hardcoded knowledge (e.g. equivalence testing)
 - ontological reasoning
 - learned mappings
 - context aware agents
 - etc.

Requirements

3. Stability of Symbols

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- A proxy is precisely a ... *proxy* for an entity
- A subarchitecture which creates a proxy will use that proxy as an internal symbol for the represented entity
- The proxy is constant w.r.t the subarchitecture
- Based on the binding score, proxies are unified into *binding unions*
- Unions provide an enriched description of the proxies
- As proxies are added, the existing unions are scored and “compete” to bind
- Unions change frequently, proxies are stable

Representation Summary

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

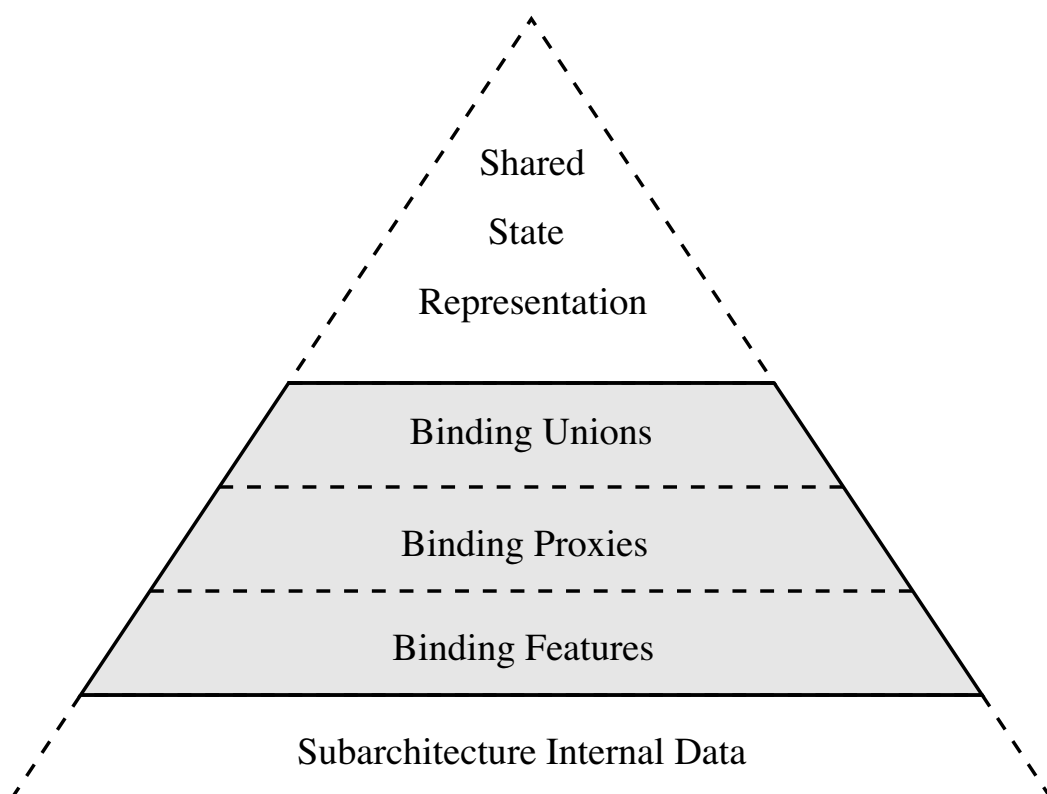
Introduction

The Problem

Motivation

The Binder

Discussion



The Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- The *binder* is a subarchitecture among the others in CAST
- Tasks:
 - Invoke comparators
 - Calculate the binding scores (unions vs. proxies)
 - Create unions
 - Identify disambiguation issues
 - Signal subarchitectures whenever their proxies are bound/rebound
 - Administration...
- The binder does this without a clue about what is represented
- Asynchronous additions and updates of proxies is handled

Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

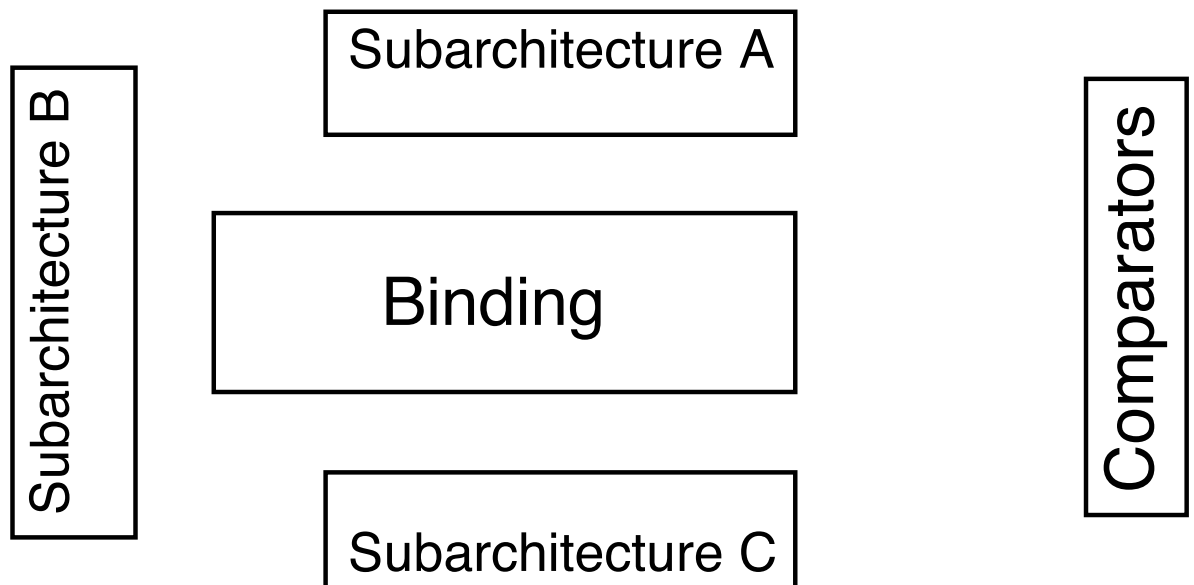
Introduction

The Problem

Motivation

The Binder

Discussion

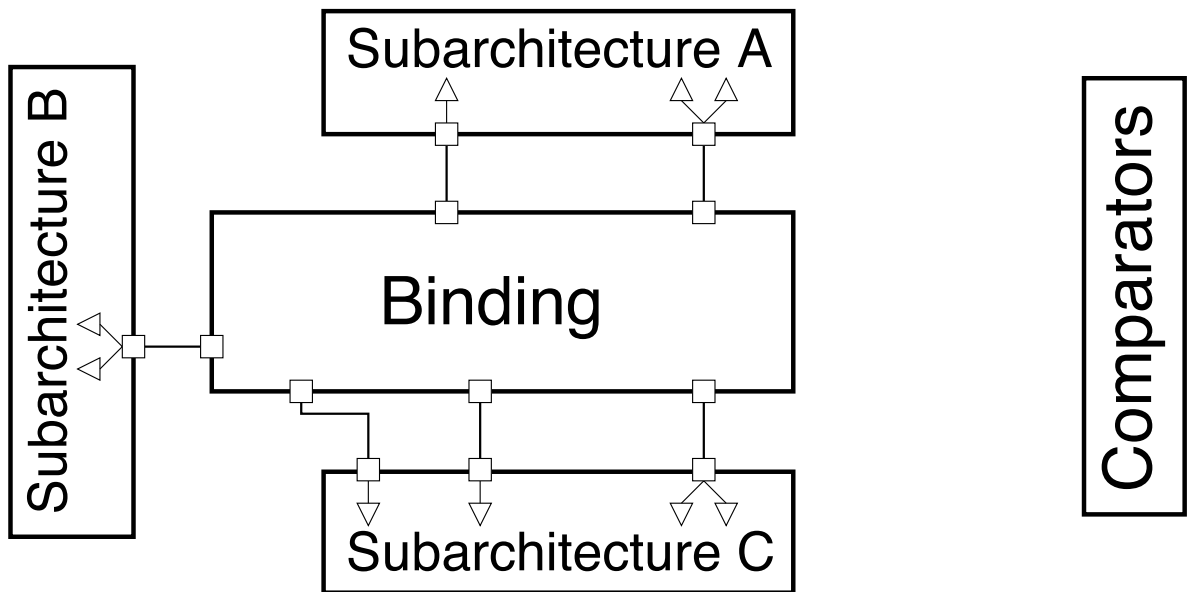


Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction
The Problem
Motivation
The Binder
Discussion

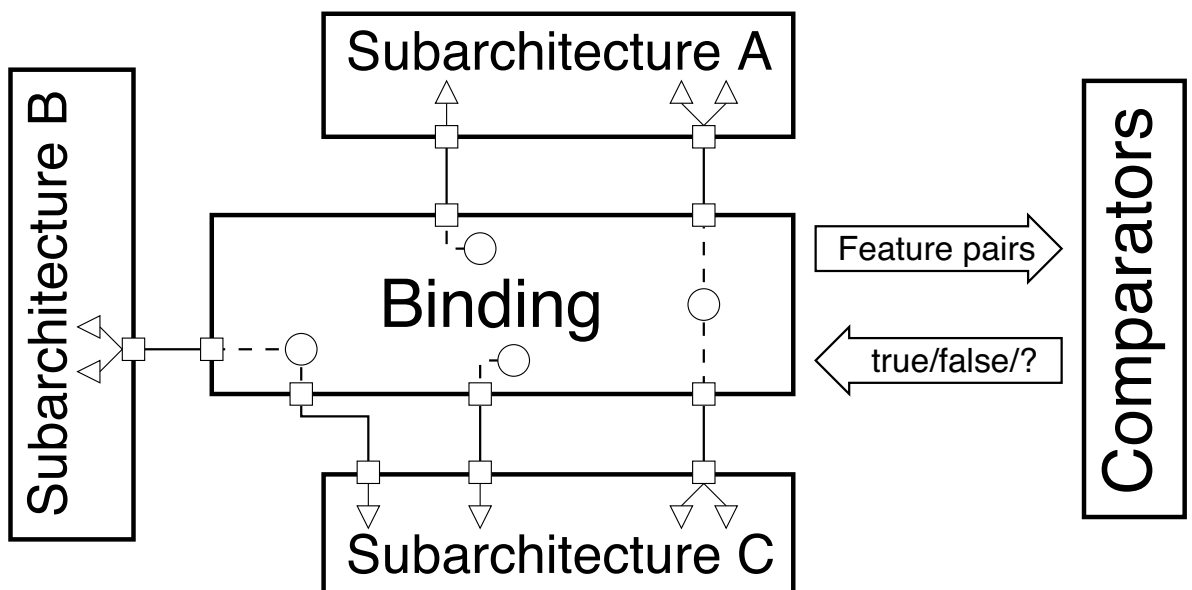


Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction
The Problem
Motivation
The Binder
Discussion

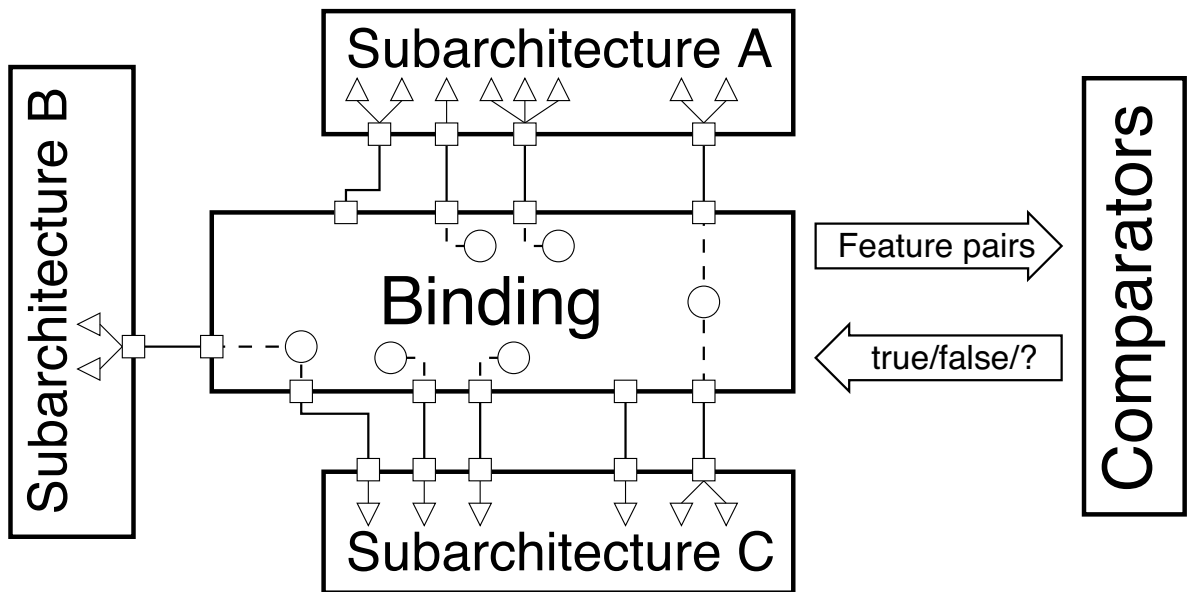


Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction
The Problem
Motivation
The Binder
Discussion

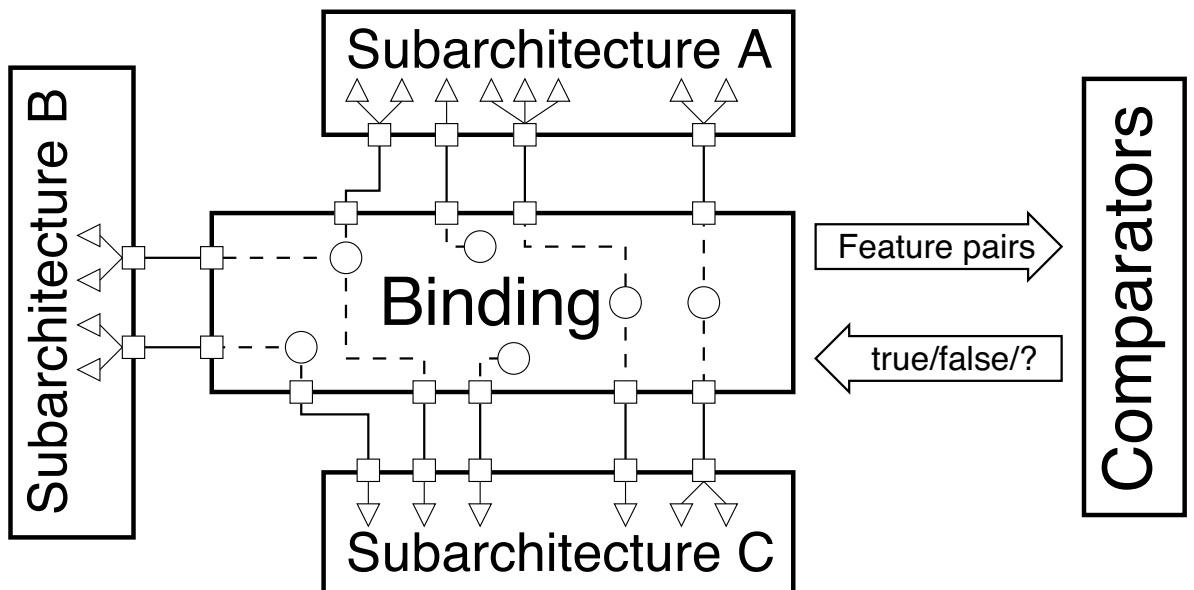


Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction
The Problem
Motivation
The Binder
Discussion



How We Use/Intend To Use the Binder

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- Primary clients
 - Communication
 - Planning
- Scenarios
 - Tabletop scenarios
 - Human augmented mapping
 - Incremental processing (subarchitectures can serve as a source of heuristics for each other)
 - Tutoring scenarios

Some Positive Consequences

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- Representational freedom
- Disambiguation issues identified
- Comparators are implemented by experts
- Modal and amodal representations side by side
- Subsymbolic representation make varying abstraction possible
- Lazy binding
- Incrementality and asynchronous processing
- Scalable
- Small demands on subarchitectures

Less Positive Consequences

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- Information fusion aspect is very limited since the binding score is extremely simple (no fuzzy or Bayesian scoring etc)
 - internally, comparators may be as SOTA as they like, though
- Lack of comparable features can be problematic
- Anything can be a binding feature, not everything should though
- Anyone can propose proxies as they like, but can you trust everyone?
- It's important that subarchitectures are conservative about proposing proxies! (layered binding an option otherwise)

Future Work

Binding

Jacobsson,
Hawes, Kruijff,
Wyatt

Introduction

The Problem

Motivation

The Binder

Discussion

- Incorporate other approaches to symbol grounding as subcomponents (as comparators or monitors)
- Enrich the binding score to accomodate representations of belief of comparators
- Episodic memory



Interactive Learning and Cross-Modal Binding – A Combined Approach

Henrik Jacobsson¹, Nick Hawes²,
Danijel Skočaj³, Geert-Jan Kruijff¹

¹ Language Technology Lab, DFKI GmbH, Germany

² School of Computer Science, University of Birmingham, UK

³ Faculty of CIS, University of Ljubljana, Slovenia

Univerza v Ljubljani

Symposium on Language and Robots 2007
Aveiro, Portugal, 12 December 2007

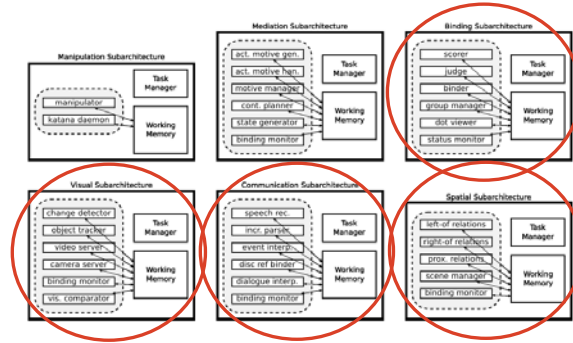


Overview

- Introduction
 - Binding
 - Learning
- Interplay between binding and learning
 - Explicit and implicit learning
 - Co-learning
 - Negation / Unlearning
- Different modes of learning
- Our learning method
- Experimental results
- Integrated system
 - Demo
- Conclusions and work in progress

CoSy Architecture Schema

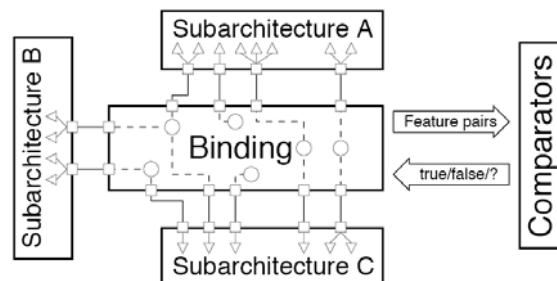
- CoSy Architecture Schema Toolkit (CAST)



- cf. Henrik Jacobsson et. al., Crossmodal Content Binding in Information-Processing Architectures

Binder

- cf. Henrik Jacobsson et. al., Crossmodal Content Binding in Information-Processing Architectures

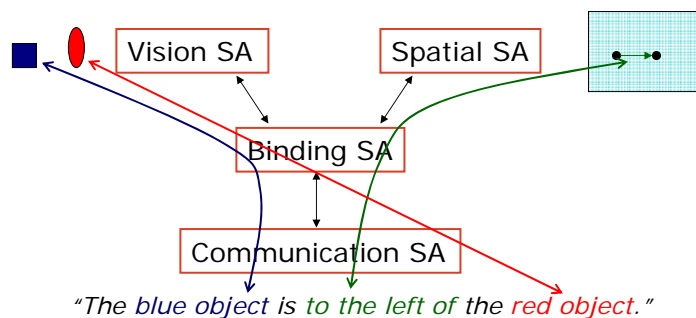


Learning

- All of the knowledge cannot be input to the system by hand => **Learning**
- The information has to be gradually acquired, processed and structured => **Continuous learning**
 - open-ended, life-long, incremental, on-line learning
 - gradually enlarging the ontology
 - No separation on the training and recognition stage
 - the training and recognition are performed in the loop
- The information is provided by a tutor or other modalities => **Interactive cross-modal learning**
 - Communication with the tutor, dialogue
 - Developmental learning, dynamical scaffolding, graded curriculum
 - Co-learning, implicit learning, unlearning
 - Active learning

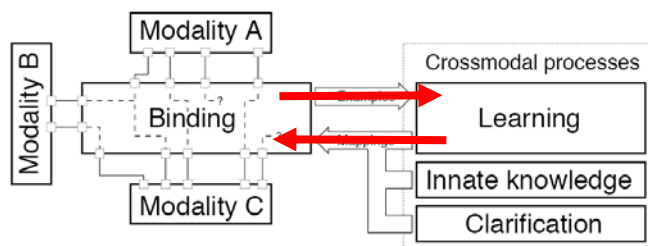
Learning cross-modal associations

- Learning of basic qualitative visual concepts
 - Learning of visual attributes
 - Learning of spatial relationships
- Symbol grounding
- Finding associations between words describing the visual and spatial concepts and automatically extracted visual and spatial features.



Cross-modal binding and learning

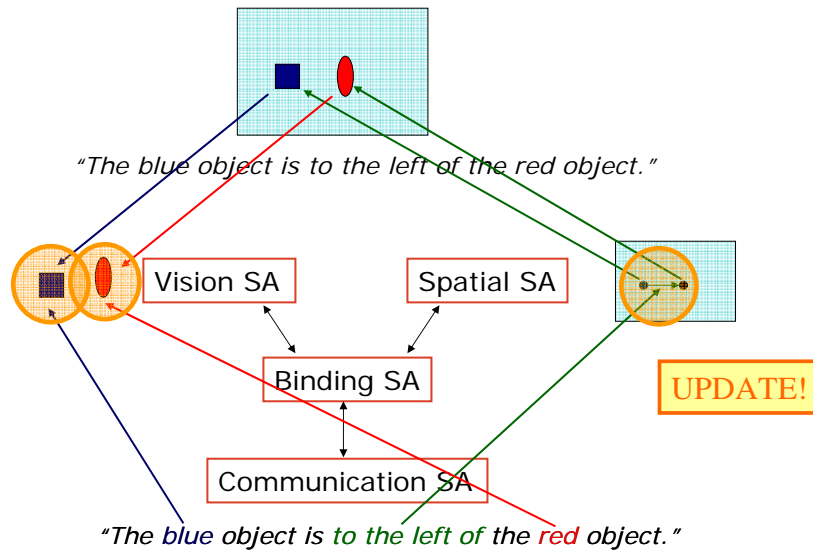
- Interplay between cross-modal binding and interactive learning
 - Learner improves mappings used by binder
 - Binder generates training examples for learner
 - In an incremental and interactive way



Explicit/implicit learning

- Three aspects of intention of a communicative act:
 - Assertions
 - "This is a blue thing."
 - Tutor ascribes new information to the referent.
 - Additional information is a communicative goal (primary purpose) -> **explicit learning**.
 - Use only salience for binding. Update.
 - Command
 - "Put the blue ball to the left of the red cube."
 - Tutor is referring to objects.
 - Additional information is given indirectly as a side effect -> **implicit learning**.
 - Use binder to resolve the referent. Update if reliable.
 - Questions
 - "Is this object blue?"
 - Tutor asks for information.
 - No additional information is given.
 - Use salience for binding. Recognize. Reply. Update if reliable.

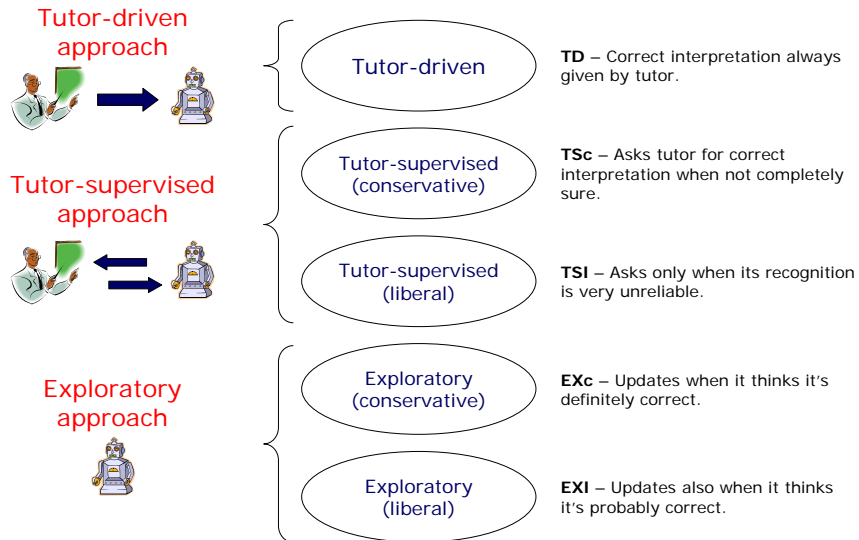
Co-learning



Negation/unlearning

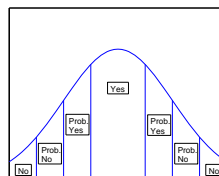
- H: "This is not blue."
- Update the current representations with negative information (negative examples).
- Especially important in an incremental setting and when the system tries to learn autonomously (without tutor supervision)
 - Error propagation
 - Requires error recovery
- Unlearn with false positives

Different modes of learning



Updating rules

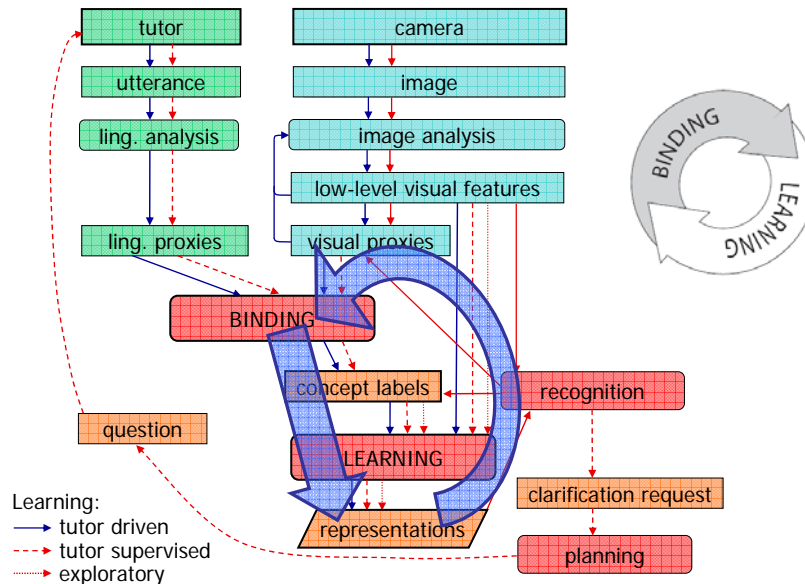
- Recognition
 - reliability of the answer



- Update table:

	YES	PY	PN	NO	DK
TD	ask	ask	ask	ask	ask
TSc	upd	ask	ask	/	ask
TSl	upd	upd	/	/	ask
EXc	upd	/	/	/	/
EXl	upd	upd	/	/	/

Learning process



Learning method

- Statistical learning of basic visual concepts.
- Find a subset/combination of extracted features that determines visual attributes/spatial relations
 - **Consistency** of the feature values extracted from the objects labeled with the same concept label.
 - **Specificity** of these values with respect to the rest of the feature values related to other concept labels.
- Each attribute value is associated with one extracted visual feature
- It is modeled with a distribution of the values of the best extracted visual feature (using **KDE distribution**)
- For redundancy, all feature values are modeled with Gaussian distribution to enable switching association assignments
- Selection of the best feature is based on the Hellinger distance between probability distributions
- Incremental updating of representations
- Facilitates unlearning

Experimental setup

- Learning and recognition of visual attributes
 - 6 visual features
 - 3 appearance features (Hu, Sa, In)
 - 3 shape features (Ar, Pr, Cm)
 - 10 visual attributes
 - 4 colors (Rd, Gr, Bl, Yl)
 - 2 sizes (Sm, Lr)
 - 4 shapes (Sq, Cr, Tr, Rc)
- Interactive system
- Quantitative evaluation
 - Evolution of the recognition score through time
 - Scoring table:



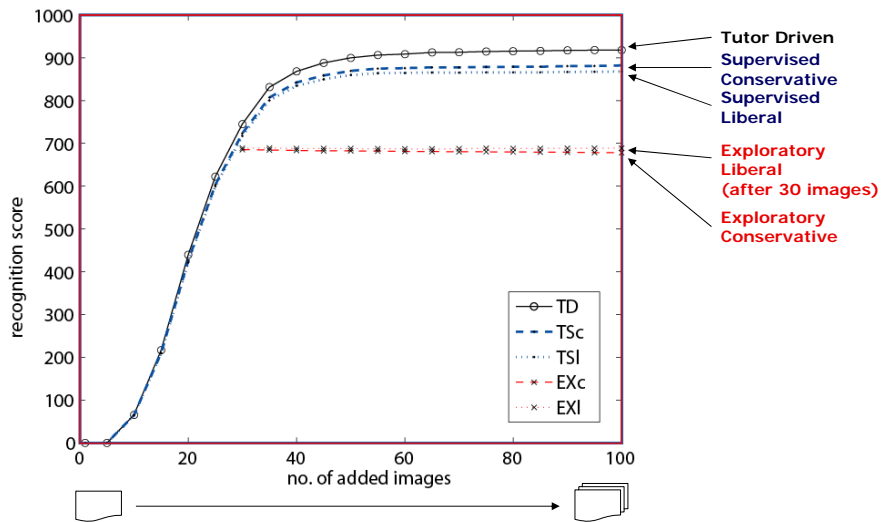
	YES	PY	PN	NO	DK
YES	1	0.5	-0.5	-1	0
NO	-1	-0.5	0.5	1	0

Dialogue example

'H: This is a red, small, and square object.' 'R: OK.' 'H: What do you see?' 'R: I don't know this object.' 'H: It is green, small, and triangular.' 'R: OK.' 'H: What do you see?' 'R: I see a small and square object.' 'H: Yes, that's correct.' 'R: OK.' 'H: It is also blue.' 'R: OK.' 'H: What do you see?' 'R: I see a green object. I think it is also large.' 'H: Yes, that's correct.' 'R: OK.' 'H: It is also rectangular.' 'R: OK.' 'R: I don't know this object. What is it like?' 'H: It is blue, small, and rectangular.' 'R: OK.' 'R: Is this object large?' 'H: Yes, that's correct.' 'R: OK.' 'H: It is also yellow and circular.' 'R: OK.' 'R: Is this object red?' 'H: No, it is not red.' 'R: OK.' 'H: It is yellow.' 'R: OK.' 'H: What do you see?' 'R: I see a red, small, and triangular object.'	Tutor driven
'R: OK.' 'H: It is yellow.' 'R: OK.' 'H: What do you see?' 'R: I see a red, small, and triangular object.'	Tutor supervised

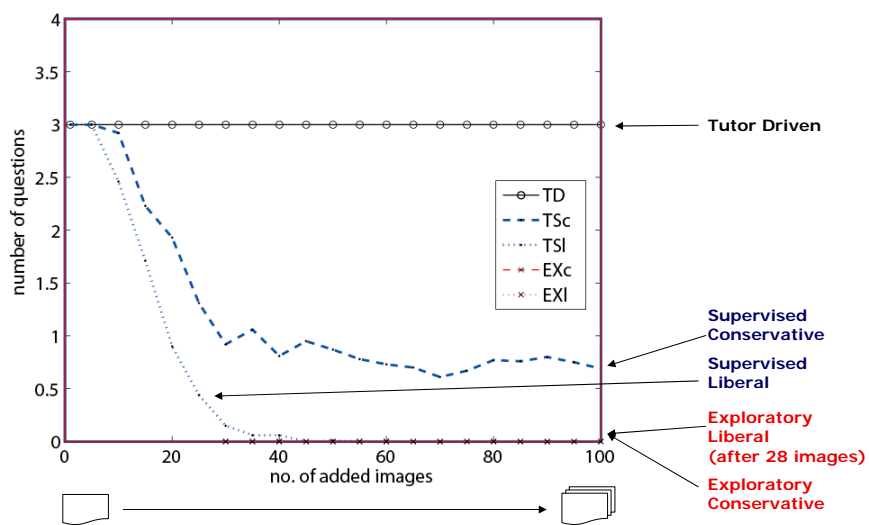
Exp. results – Appearance properties

Recognition score



Exp. results – Appearance properties

Number of questions

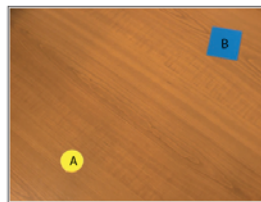


Learning of spatial relations

- 2 objects (A and B)
- 5 features (x,y,dx,dy,d)
- 11 spatial relations:
 - TL: to the left of: "A is to the left of B"
 - TR: to the right of: "A is to the right of B"
 - CT: closer than: "A is closer to me than B"
 - FT: further away than: "A is further away from me than B"
 - NT: near to: "A is near to B"
 - FF: far from: "A is far from B"
 - OL: on the left: "A is on the left"
 - OR: on the right: "A is on the right"
 - IM: in the middle: "A is in the middle"
 - NR: near: "A is near"
 - FA: far away: "A is far away"

Two examples

- Automatic scene description



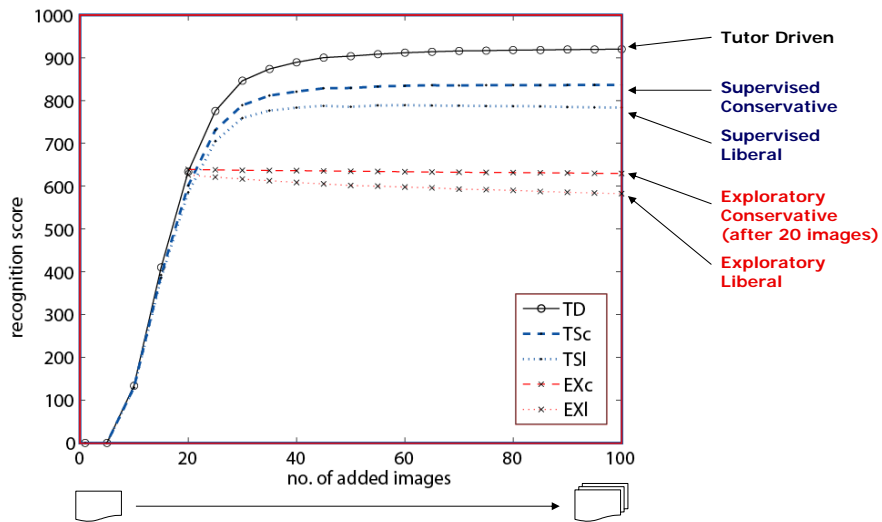
A is yellow, small, and circular.
B is blue, large, and square.
A is on the left.
A is near.
B is on the right.
B is far away.
A is to the left of B.
A is closer than B.
A is far from B.
B is to the right of A.
B is further away than A.
B is far from A.



A is red.
B is yellow.
A is on the right.
A is far away.
B is on the left.
A is to the right of B.
A is further away than B.
A is far from B.

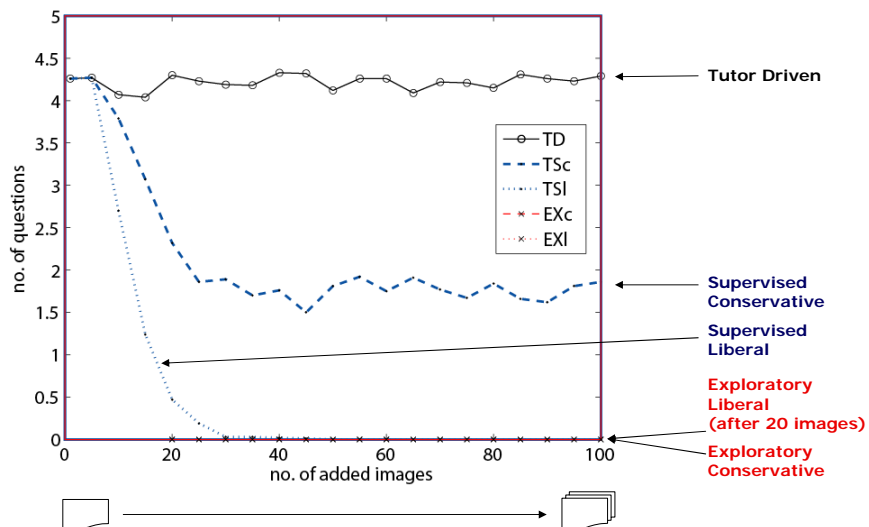
Exp. results – Spatial relations

Recognition score



Exp. results – Spatial relations

Number of questions

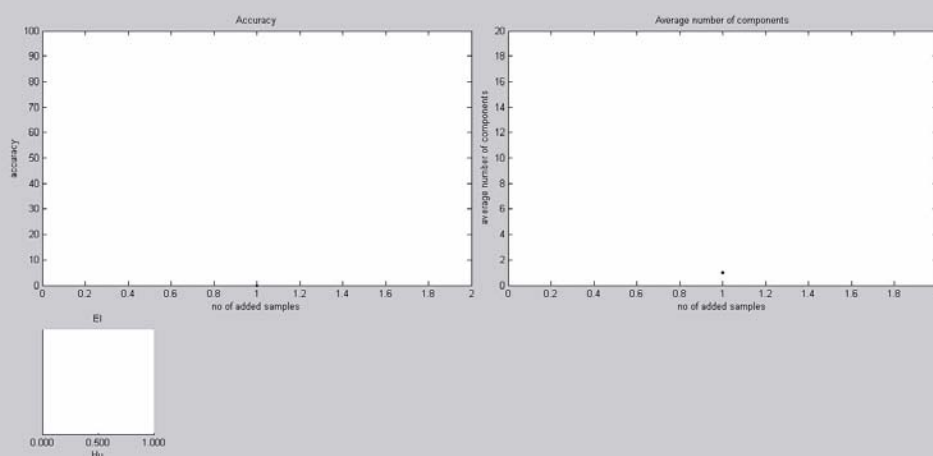


Experimental setup

- Learning and recognition of visual properties of everyday objects
- 6 visual features
 - 3 appearance features (Hu, Sa, In)
 - 3 shape features (Ar, Co, Ec)
- 6 visual properties
 - 4 colors (Rd, Gr, Bl, Yl)
 - 2 shapes (Cm, El)



Experimental results



Demo - learning

The screenshot shows a Linux desktop environment during the 'learning' phase. The desktop includes a 'segmentation' window with a background image, a terminal window running commands, and two windows titled 'LR Visualisation' and 'LRvisModels'. The 'LR Visualisation' window shows a network diagram with nodes labeled 'No', 'Sa', 'In', 'Ar', 'Dr', 'Et', 'Pd', 'Gr', 'B', 'Vf', 'Ck', 'B'. The terminal window displays the output of a learning process, including model parameters and training progress.

Demo - unlearning

The screenshot shows a Linux desktop environment during the 'unlearning' phase. The desktop includes a 'segmentation' window with a background image, a terminal window running commands, and two windows titled 'LR Visualisation' and 'LRvisModels'. The 'LR Visualisation' window shows a network diagram with nodes labeled 'No', 'Sa', 'In', 'Ar', 'Dr', 'Et', 'Pd', 'Gr', 'B', 'Vf', 'Ck', 'B'. The 'LRvisModels' window shows a grid of plots for each node, with the x-axis labeled 'No', 'Sa', 'In', 'Ar', 'Dr', 'Et', 'Pd', 'Gr', 'B', 'Vf', 'Ck', 'B'. The terminal window displays the output of an unlearning process, including model parameters and training progress.

Conclusions

- A system for **continuous interactive** building of **cross-modal associations** between low level modality-specific features and amodal high level concepts
 - A unified framework for learning object basic properties and spatial relations
 - Facilitates unlearning and co-learning
 - Based on reconstructive representations (KDE)
 - Different modes of learning
 - Mixed initiative learning, implicit learning
- Interplay between cross-modal binding and interactive learning
 - Learner improves mappings used by binder
 - Binder generates training examples for learner
 - In an incremental and interactive way

The end



Information-Processing Architectures for Intelligent Robots: Designs, Tools, Examples and Experiments

Nick Hawes

Intelligent Robotics Lab, School of Computer Science, University of Birmingham

Computer Science Research Colloquium, University of
Hertfordshire, 12/12/07



Outline

Motivation

The CoSy Architecture Schema

Illustrations

Experiments

Conclusion



Motivation

- 7-site 4-year EU project building robotic systems aiming to demonstrate both state-of-the-art components and systems.
- We are trying to advance of the *science of building* intelligent systems: **integration** is central.
- We see information-processing architectures as central to this problem.

What Are Architectures?

- Information-processing structures that circumscribe the functionality of system.
- An understanding of *information-processing architectures* is central to the understanding of intelligent integrated systems.
- They are a useful abstraction for integration (more specific than communication frameworks, more general than particular representations).
- As a design and implementation tool they represent the battleground of science and engineering.

Levels of Description

- We use four different levels of description for architectures:
 - High-level principles and *requirements*.
 - A *schema*-level realisation of these.
 - *Instantiations* of a schema in a concrete design.
 - *Implementations* of a design in software and hardware.
- These relate to *niche space* (requirements) and *design space* (designs) as described by Aaron Sloman and the Birmingham CogAff group.

Contributions

Principled approaches for integrating functions (i.e. components and their representations) into a single intelligent robot.

- An **architecture schema**, combining insights from both robotics and AI/cognitive science, designed to support concurrent processing on shared information.
- An approach to **binding** information across multiple modalities into a single amodal representation.
- An investigation into **filtering** in various architecture instantiations.

Some (Selected) Key Problems

- **Filtering:** How does information flow between a subset of components in an architecture?
- **Binding:** How can information about the same thing from different components in an architecture be connected?
- **Incrementality:** How can architectures be easily extended with new capabilities?

Outline

Motivation

The CoSy Architecture Schema

Illustrations

Experiments

Conclusion

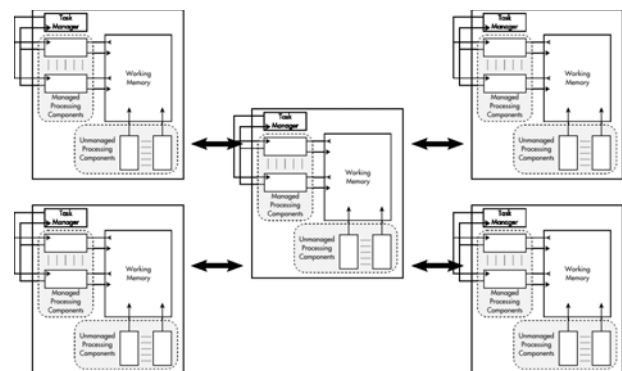
The CoSy Architecture Schema (CAS)

- A schema which defines a limited space of architectures and thus instantiations.
- Based on *requirements* drawn from an analysis of robotic scenarios, and common solutions in implemented systems.
- General enough for experimentation, specific enough to study design commitments.



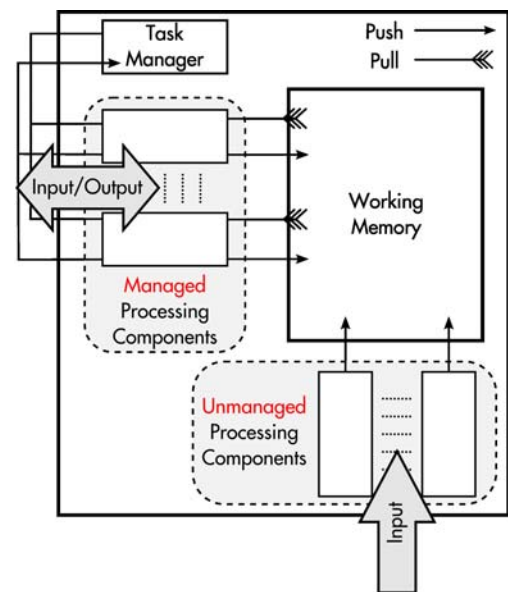
CAS Key Features

- Collection of loosely coupled subarchitectures.



CAS Key Features

- Collection of loosely coupled subarchitectures.
- Each subarchitecture contains processing components that update structures within a working memory (WM).
- Components can read all WMs but only write to the local WM (bar privileged components).
- Processing is controlled by a network of task managers.



CAS in Context

- CAS makes practical use of approaches from cognitive systems.
 - Shared working memories.
 - Management methods for components.
- ...whilst attempting to formalise common practice in robot systems.
 - Multiple concurrent components.
 - Distributed design.
- We are motivated by cognition, although we are not aiming for human-like systems.

CAST: The CAS Toolkit

- 2-layer toolkit: BALT for communication, CAST implements CAS on top.
- Cross language, distributed design, open source, multi-OS. Supports incremental development.
- Biggest system about 30 components running on 5 machines.
- **Key contribution:** separation of architecture from content.

Outline

Motivation

The CoSy Architecture Schema

Illustrations

Experiments

Conclusion

Architectures for Integration

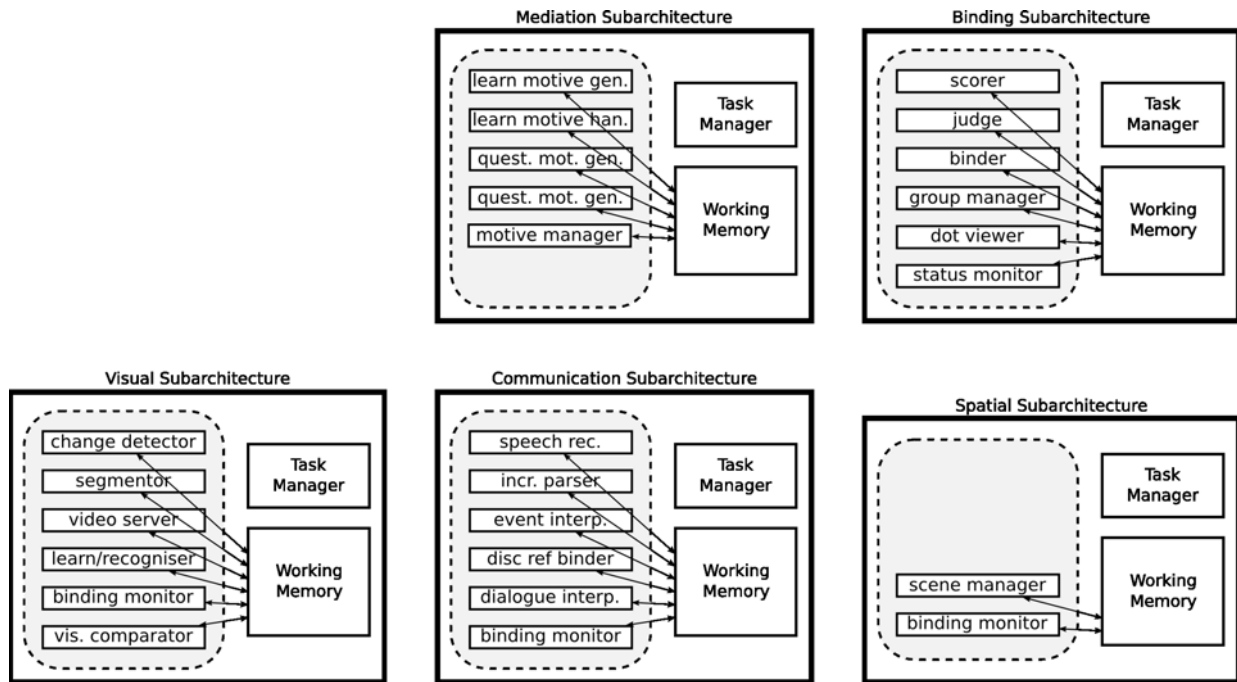
- Over the last two years we have iteratively constructed systems for HRI in a table-top manipulation scenario.
- Each iteration has allowed us to further explore issues in integrated systems, architectures, binding, filtering etc.
- Iterations:
 1. Tutor-driven learning of visual properties.
 2. Language-driven manipulation.

Tutor-Driven Learning of Visual Properties

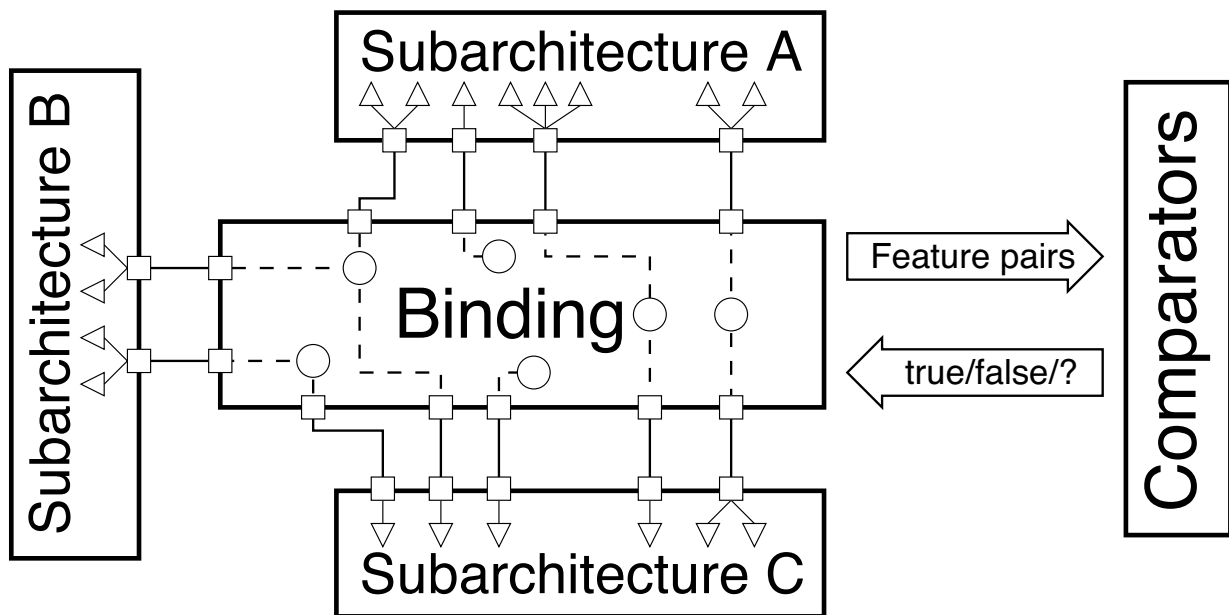
Features

- 2 learning modes:
 - Tutor Driven: Learning task generated via language input.
 - Tutor Supervised: Learning task generated via visual input.
- Spatial WM: Stack of frames of objects in scene, quantitative to qualitative abstraction.
- Mediation: Raises learning goals, posts goals to visual and language SAs.
- Binding SA: Binding linguistic information to visual and spatio-temporal information to generate modality-neutral representations.

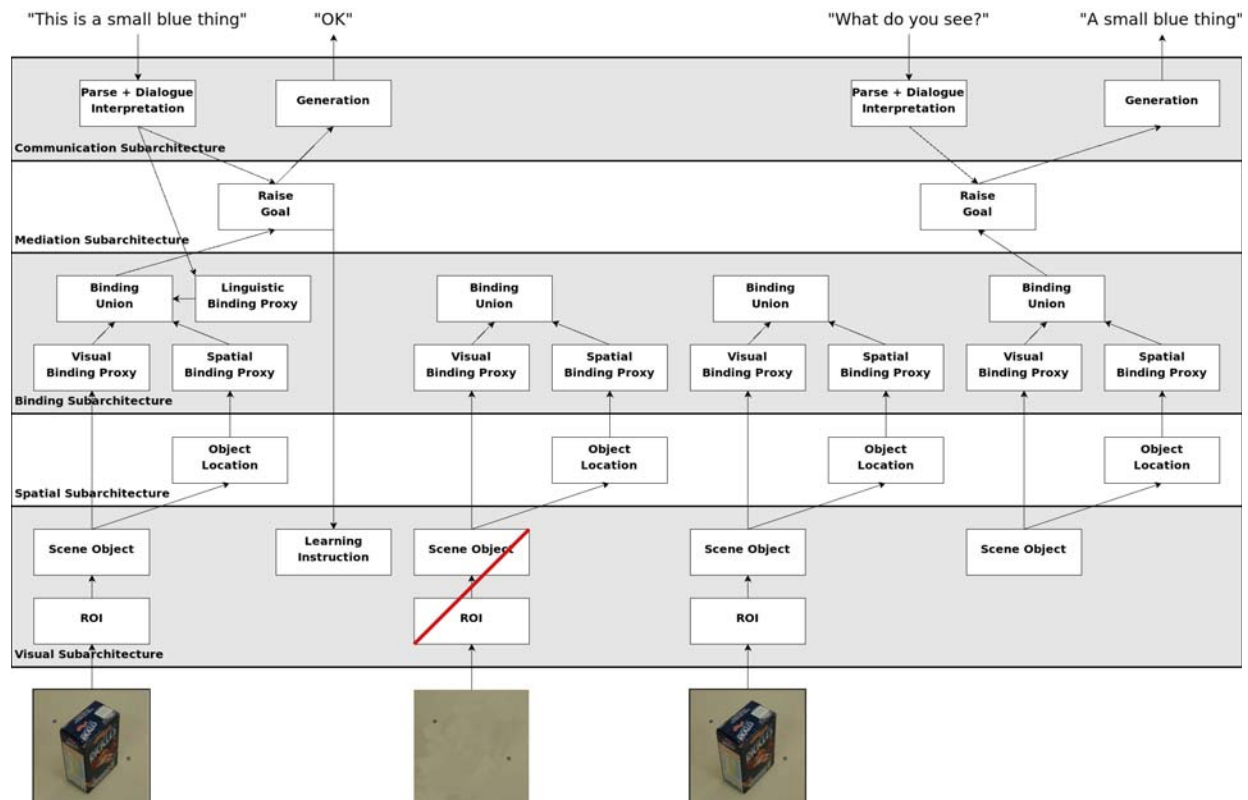
Tutor-Driven Learning of Visual Properties Instantiation



Tutor-Driven Learning of Visual Properties Binding



Timeline

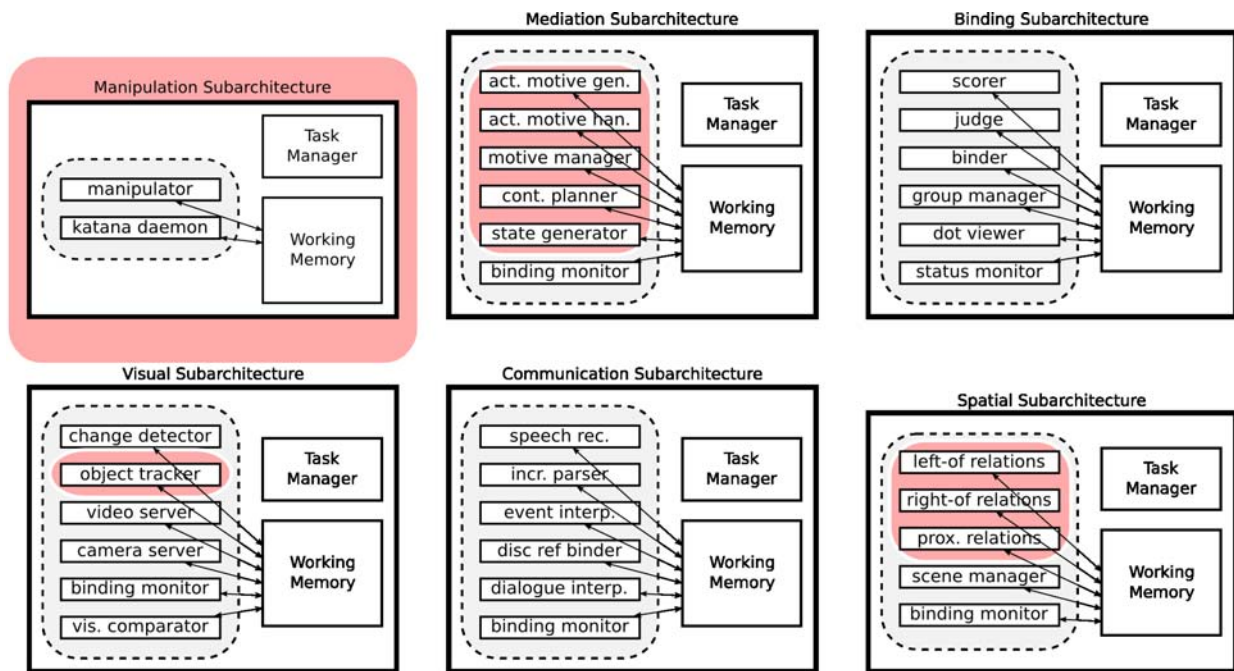


Language-Driven Manipulation Features

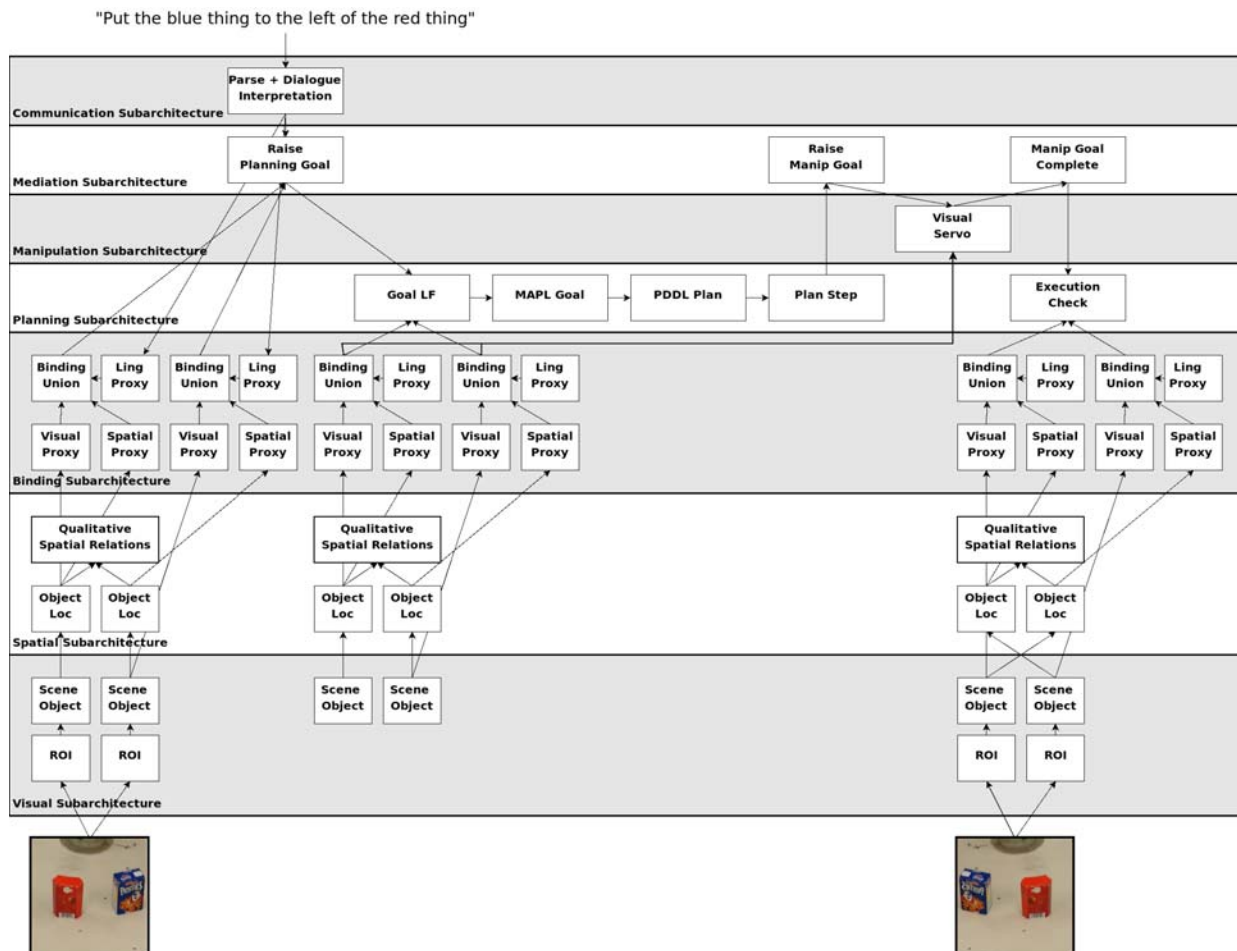
- Goals are raised by language.
- References are made to objects using previously learned features.
- Robot plans intentional actions using a symbolic planner.
- Intention shifting is handled via execution monitoring and continual planning.
 - Symbolic state generated from binding features at regular intervals.
 - Current state checked against expectations during execution.
 - Feedback from manipulator checked during execution.



Language-Driven Manipulation Instantiation



Timeline



Outline

Motivation

The CoSy Architecture Schema

Illustrations

Experiments

Conclusion



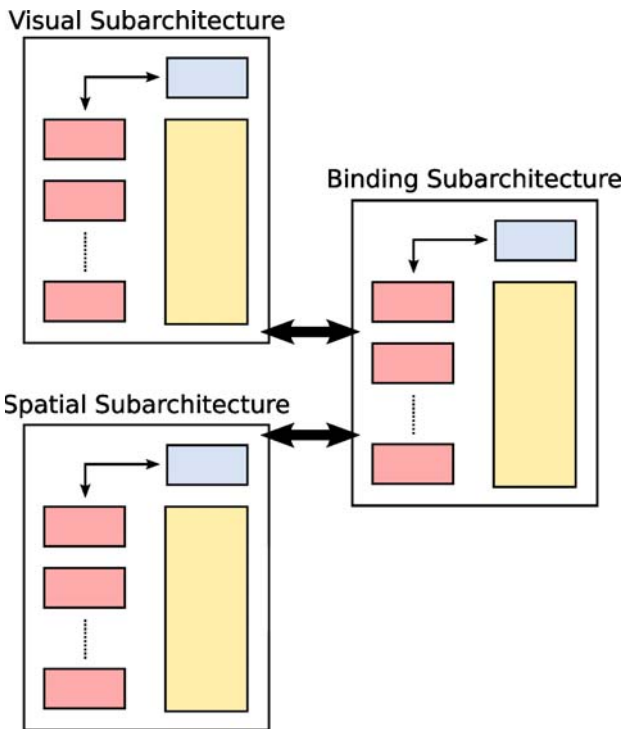
Exploring Design Space

- Given our stated aim of *understanding* systems, building them is not enough.
- Can we use CAST to explore trade-offs in architectural design space?
- Yes!
- **Methodology**: Build systems that represent different points in design space and measure various properties about them to characterise trade-offs.
- **Investigate**: Cost of communication and filtering at three points in design space.



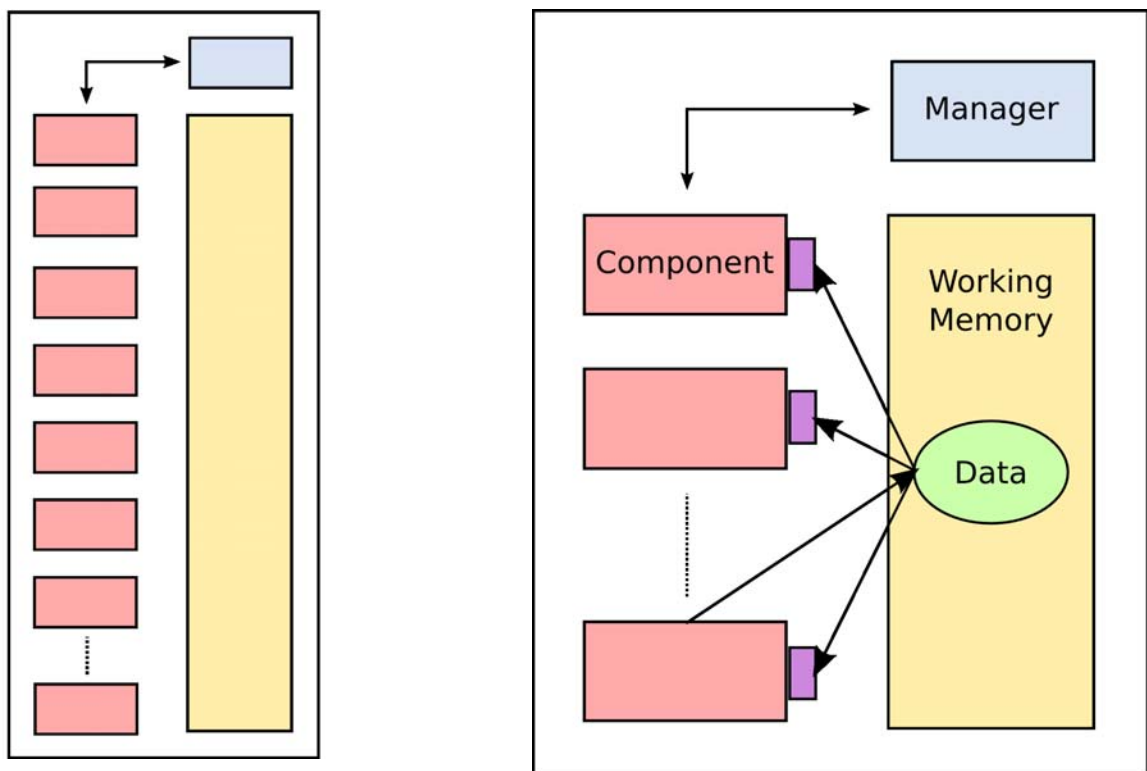
Three Schema Instantiations

N components : 1 subarchitecture



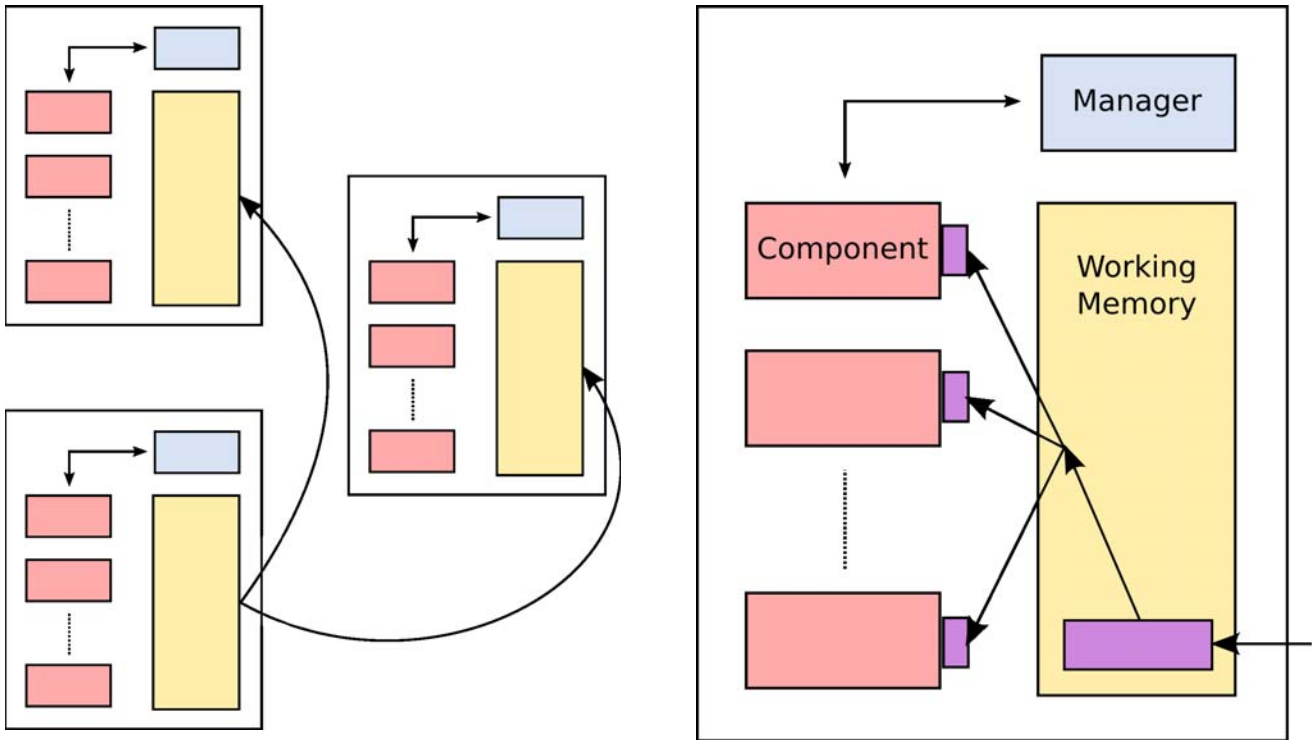
Three Schema Instantiations

N components : 1 subarchitecture



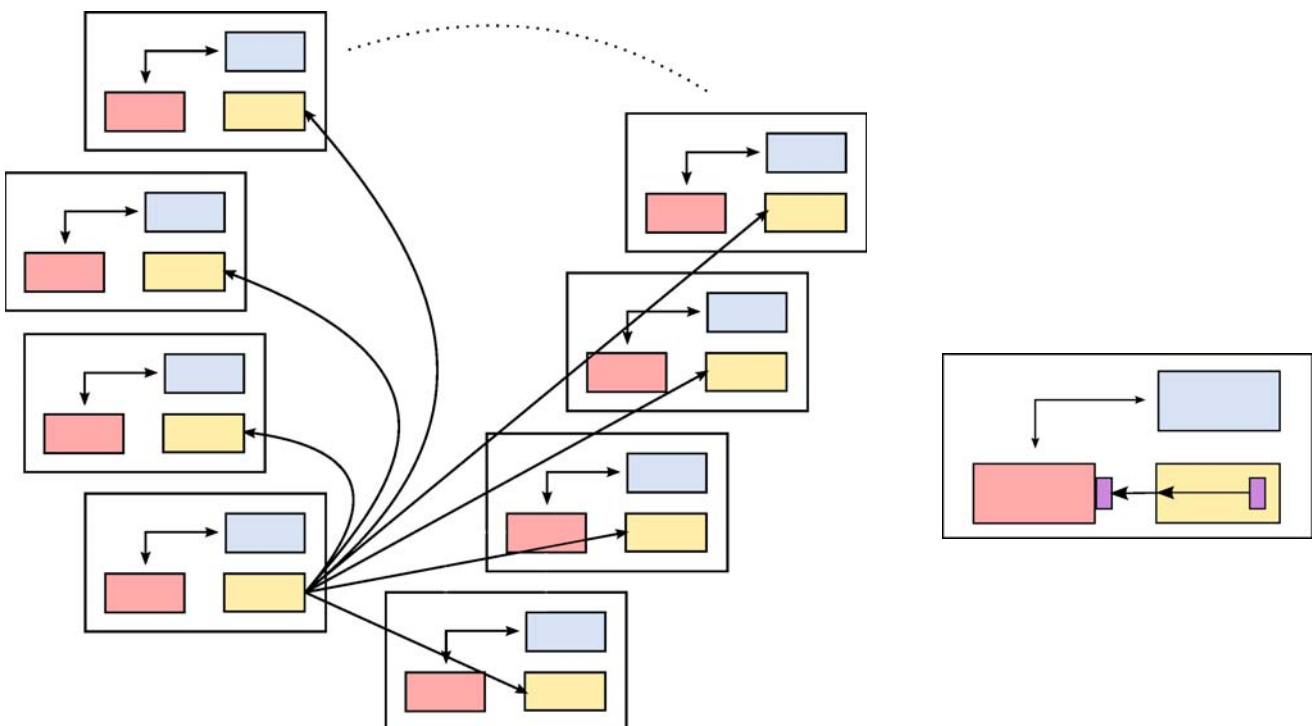
Three Schema Instantiations

N components : M subarchitectures (N > 1)



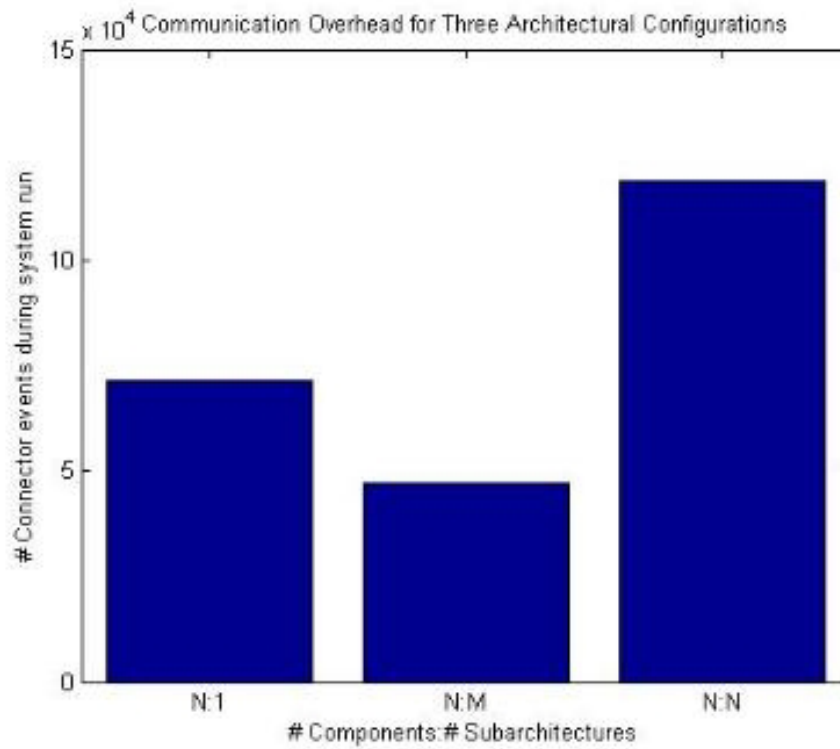
Three Schema Instantiations

N components : N subarchitectures



Results

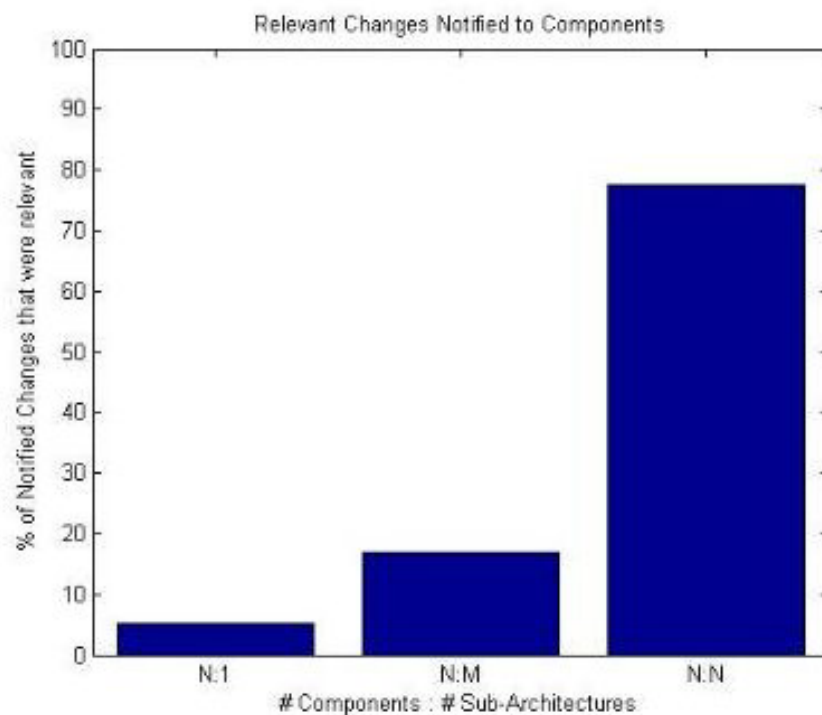
Communication Overhead



Navigation icons: back, forward, search, etc.

Results

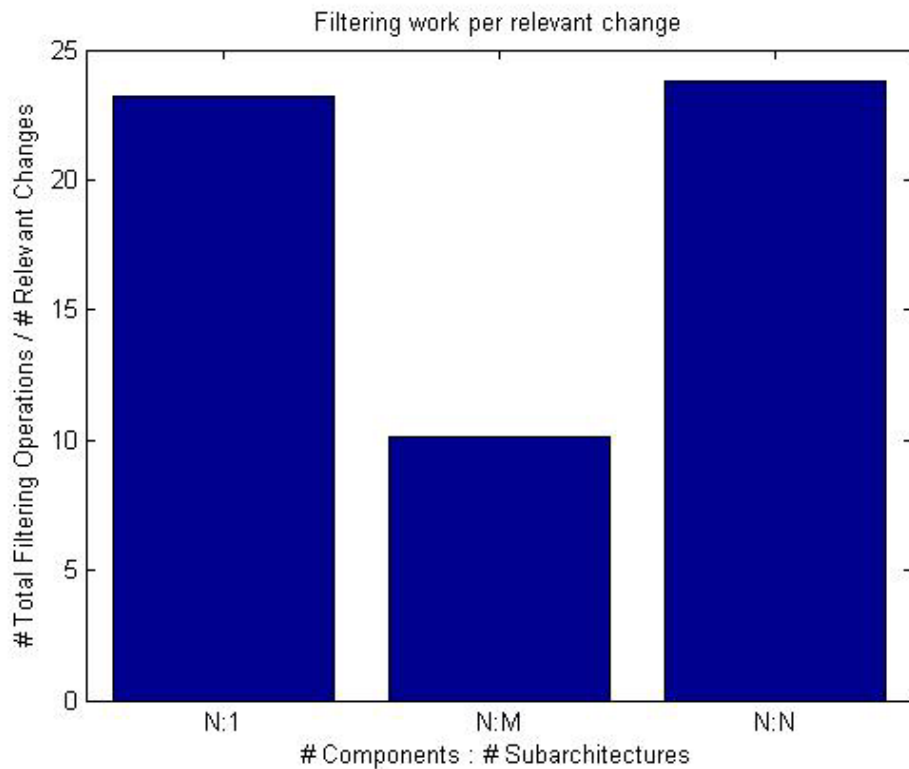
Filtering Relevance



Navigation icons: back, forward, search, etc.

Results

Filtering Relevance



Results Summary

- N:M forms a sweet spot in the space of architectures we explored
- Better for:
 - Communication overhead.
 - Filtering work required to identify relevant information
- This is robust with changes in scene complexity and system complexity



Outline

Motivation

The CoSy Architecture Schema

Illustrations

Experiments

Conclusion



Conclusion

- The CoSy Architecture Schema defines a limited space of possible architectures, allowing us to explore this space in a principled manner.
- A number of CAS instantiations have been implemented for HRI scenarios.
- These instantiations have allowed us to explore approaches to cross-modal binding and aspects of architectural design space.
- All implementations are based on our CAS toolkit. This is available as open source code:
<http://www.cs.bham.ac.uk/research/projects/cosy/cast/>



Acknowledgements

- **Birmingham:** Jeremy Wyatt, Aaron Sloman, Michael Zillich, Marek Kopicki, Somboon Hongeng.
- **DFKI, Saarbrücken:** Henrik Jacobsson, Geert Jan Kruijff, John Kelleher (now at Dublin Inst. Tech).
- **Albert-Ludwigs-Universität, Freiburg:** Michael Brenner.
- **University of Ljubljana:** Danijel Skočaj, Gregor Berginc

The End

Questions?

Porazdeljene inteligentne programske tehnologije

Večagentni sistemi

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Poglavje 2

Porazdeljene inteligentne programske tehnologije

Večagentni sistemi

- Več agentov
- Analizirati, opisati in načrtovati okolja, v katerih v katerih lahko agenti produktivno sodelujejo
- Večagentni sistemi
- Združbe agentov (society of agents)

- **Komunikacijski protokol** določi kakšni tipi sporočil se lahko komunicirajo.
- **Interakcijski protokol** določi na kakšen način (v kakšnem zaporedju, po kakšnem protokolu) se bodo prenašala sporočila.

Porazdeljene inteligentne programske tehnologije, Večagentni sistemi

2

Komunikacijski in interakcijski protokol

- Primer komunikacijskega protokola (možni tipi sporočil):
 - Predlagaj potek akcije
 - Sprejmi potek akcije
 - Zavrni potek akcije
 - Opusti potek akcije
 - Ne strinjaj se s predlogom poteka akcije
 - Podaj svoj predlog poteka akcije
- Primer interakcijskega protokola (ki temelji na zgornjem)
 - Agent1 predlaga potek akcije Agentu2.
 - Agent2 oceni predlog in
 - Pošlje sprejem predloga Agentu1
 - Ali pošlje svoj predlog Agentu1
 - Ali pošlje nestrinjanje s predlogom Agentu1
 - Ali pošlje zavrnitev predloga Agentu1.

Motivacija

- Zakaj porazdeljen sistem
 - Včasih je problem sam porazdeljen
 - Včasih je problem lažje razumljiv/rešljiv
 - Včasih je problem rešljiv samo na porazdeljen način (se podatki ne smejo deliti/odkrivati)
 - Včasih je problem samo tako obvladljiv
- Podatki so porazdeljeni po ločenih informacijskih sistemih, ki so lahko
 - Geografsko porazdeljeni
 - Imajo lahko veliko komponent
 - Vsebujejo ogromno količino informacij
 - Pokrivajo širok spekter informacij
 - Komponente so lahko porazdeljene in nehomogene
 - Topologija sistemov se lahko dinamično spreminja

Porazdeljena umetna inteligenca - DAI

- Štiri tehnike:
 - Modularnost
 - Porazdeljenost
 - Abstrakcija
 - Inteligenca-> Porazdeljena umetna inteligenca (Distributed AI)
- Posamezni agenti so lahko:
 - Inteligentne aplikacije
 - Aktivni viri informacij
 - "wrapperji" okrog klasičnih komponent
 - On-line mrežni servisi
 - Inteligentni roboti
- Posamezni agenti
 - Delujejo avtonomno
 - Upravljajo z lokalno informacijo
 - Sodelujejo z drugimi agenti

Prednosti MAS

- Povezovanje individualnih agentov omogoča:
 - Sodelovanje pri reševanju problemov
 - Deljenje ekspertize
 - Vzporedno delo na skupnih problemih
 - Modularen razvoj in implementacijo
 - Robustnost skozi redundanco
 - Predstavitev različnih pogledov in različnih znanj o problemu
 - Ponovno uporabo

Uporaba MAS

- Računalniki, procesiranje informacije, so vsepovsod prisotni => ogromno potencialnih agentov, ki bi lahko komunicirali med sabo
- Primer: pametna kuhinja
 - Opekač ve, kdaj je kruh opečen.
 - Avtomat za kavo ve, kdaj je kava kuhana.
 - Z medsebojno interakcijo bo kruh opečen istočasno kot bo kava kuhana.
- Od kupa nepovezanih inteligentnih agentov k inteligentnemu večagentnemu sistemu

Značilnosti večagentnega okolja

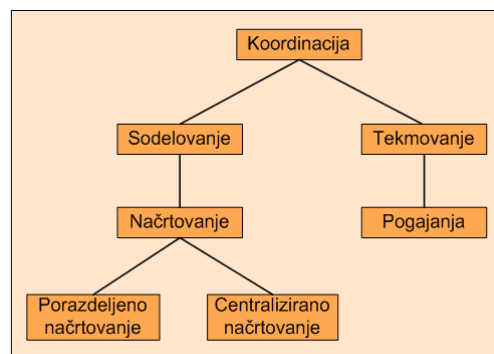
- Večagentna okolja
 - nudijo infrastrukturo za delovanje večagentnih sistemov (komunikacijske in interaktivne protokole)
 - So tipično odprta in nimajo centralnega načrtovalca
 - Vsebujejo agente, ki so avtonomni in porazdeljeni in so lahko sebični ali kooperativni
- Ključne lastnosti večagentnega okolja:
 - Poznavanje (okolja)
 - Predvidevanje (okolja)
 - Kontrolabilnost (okolja)
 - Uporaba zgodovine (ali temelji prihodnost na zgodovini ali samo na sedanjosti)
 - Teološkost (smisel okolja, so še drugi agenti?)
 - Spremenljivost (se lahko okolje spremeni med premišljanjem)

Komunikacija

- Agenti lahko komunicirajo med sabo:
 - Agent lahko zaznava, sklepa in deluje
 - Ima eksplicitno shranjeno znanje, ki ga zna uporabljati za sklepanje
 - Ima sposobnost komuniciranja
 - Zaznavanje (sprejemanje sporočil)
 - Akcija (pošiljanje sporočil)

Koordinacija

- Komunikacija omogoča koordinacijo agentov za lažjo/boljšo doseg skupnih ciljev



Koherenca

- Koherenca pove kako enotno se sistem obnaša
- Problem: kako zagotoviti globalno koherenco brez eksplicitnega globalnega nadzora
- Agenti morajo:
 - Sami določiti skupne cilje
 - Določiti skupne naloge
 - Izogibati se nepotrebnim konfliktom
 - Si deliti znanje
- Agenti morajo biti organizirani

Pomen

- Tri formalni aspekti komunikacije:
 - Sintaksa (struktura komunikacije)
 - Semantika (kaj simboli označujejo)
 - Pragmatika (kako se simboli interpretirajo)
- Pomen = Semantika + Pragmatika
- Agent komunicira, da bi razumel in da bi bil razumljen

Dimenzije pomena

- Opisovalni ali predpisovalni pomen
 - Opis pojava (lažje za razumevanje)
 - Predpisovanje obnašanja (lažje za oponašanje)
- Osebni ali dogovorjeni pomen
 - Osebni – odvisen od agenta
 - Dogovorjeni – v skladu z neko konvencijo, skupnim dogovorom, skupen za vse
- Subjektivni ali objektivni pomen
 - Subjektivno razumevanje učinka akcije
 - Objektivno razumevanje učinka akcije
- Pomen z različnih perspektiv (zornih kotov)
 - Govorca
 - Poslušalca
 - Združbe
- Semantika in pragmatika pomena
 - Pragmatika upošteva stanje udeležencev in okolja (in nista del sintakse in semantike komunikacijskega sporočila)

Dimenzije pomena

- Kontekst
 - Interpretacija sporočila je odvisna od
 - Stanja agenta
 - Trenutnega stanja okolja
 - Zgodovine okolja
- Pokritost
 - Jezik mora biti dovolj velik, da pokrije vse pomene
- Identiteta
 - Identiteta pošiljatelja, naslovnikov (eden ali več)
- Kardinalnost
 - Kolikim agentom se pošilja sporočilo (enemu, nekaterim, vsem)?

Tipi sporočil

- Različni agenti -> različni nivoji komunikacije
- Dialog
 - Agent začenja komunikacijo (glavni; master)
 - Agent odgovarja (podrejeni; slave)
 - Obojestranski pogovor (vrstni; peer)
- Dva glavna tipa sporočil
 - Trditve
 - Vprašanja
- Pasivni agent
 - Sprejme vprašanje
 - Pošlje trditev (odgovor)
- Aktivni agent
 - Lahko postavlja vprašanja in pošilja trditve (tudi zahteve)
- Vrstni (peer) agent
 - Lahko igra obe vlogi

Tipi agentov

- Komunikacijske zmogljivosti različnih tipov agentov:

	Osnovni agent	Pasivni agent	Aktivni agent	Vrstni agent
Sprejema trditve	✓	✓	✓	✓
Sprejema vprašanja		✓		✓
Pošilja trditve		✓	✓	✓
Pošilja vprašanja			✓	✓

Tipi sporočil

Tip sporočila	Namen sporočila
Trditev	Inform.
Vprašanje	Vprašanje
Odgovor	Inform.
Zahteva	Zahteva
Razlaga	Inform.
Ukaz	Zahteva
Dovoljenje	Inform.
Zavrnitev	Inform.
Ponudba	Inform
Sprejem	Inform.
Strinjanje	Inform.
Predlog	Inform.
Potrditev	Inform.
Umik	Inform.
Zavrnitev	Inform.

Komunikacijski nivoji

- Tri nivoji komunikacijskih protokolov:
 - Spodnji nivo: metoda za medsebojno povezavo
 - Srednji nivo: format (sintaksa)
 - Zgornji nivo: pomen (semantika)
- Kardinalnost protokolov
 - Binarni
 - en pošiljatelj in en sprejemnik
 - N-narni (broadcast, multicast)
 - en pošiljatelj in več sprejemnikov
- Specifikacija protokola:
 1. Pošiljatelj
 2. Sprejemnik(i)
 3. Jezik v protokolu
 4. Kodirne in dekodirne funkcije
 5. Akcije, ki naj jih izvede sprejemnik

Govorna dejanja

- Teorija govornih dejanj (speech act theory)
 - Človeški pogovorni jezik je dejanje, povzročča dejanja
- Tri aspekti govornega dejanja:
 - Izraz (locution): fizično izreče izraz govorca
 - Pomen (illocution): namenjen pomen izraza
 - Dejanje (perlocution): dejanje, ki izhaja iz pomena
- Interpretacija pomena je včasih težavna ("Zebe me.")
- Določitveni glagoli, nameni izrazov (performative verbs)
 - povedati, vprašati, zahtevati ("Povem, da...", "Zahtevam, da...")
 - Odpravijo vsak dvom o pomenu izraza
- Komunikacijski protokol poskrbi za nedvoumnost tipa in namena komunikacije
 - Še vedno pa je sporočilo samo lahko dvoumno

Komunikacijski protokol

- V komunikacijskem protokolu je temeljna ločitev
 - sintakse (ki mora biti neodvisna od domene) in
 - semantike (ki je lahko odvisna od domene)
- Komunikacijski protokol
 - Morajo uporabljati vsi agenti udeleženi v komunikaciji
 - Mora biti jedrnat

KQML

- Jezik za povpraševanje in manipulacijo
- KQML – Knowledge Query and Manipulation Language
- Protokol za izmenjavo informacij in znanja med agenti in aplikacijami
- Vsa informacija potrebna za razumevanje sporočila je zajeta v sami komunikaciji
- Osnovna struktura:

```
(KQML_performative
      :sender      <word>
      :receiver   <word>
      :language   <word>
      :ontology   <word>
      :content    <expression>
      . . .)
```

KQML

- Semantika stavka (strukture) je neodvisna od domene
- Semantika sporočila je definirana s polji
 - **:content** (sporočilo)
 - **:language** (jezik v katerem je zapisano sporočilo)
 - **:ontology** (ontologija, besednjak)
- Sporočilo samo je zaobjeto v strukturo, ki vsebuje vse informacije, ki so potrebne za njegovo razumevanje
- Polja s parametri za posredovanje sporočil
 - **:sender** (pošiljatelj)
 - **:receiver** (sprejemnik)
 - **:reply-with** (odgovori)
 - **:in-reply-to** (v odgovor)
- Komunikacija je lahko asinhrona (**reply-with** in **in-reply-to** poskrbita za povezave med sporočili)

KQML

- Primer : "Kocka A je na kocki B."
 - Uporablja se jezik KIF (Knowledge Interchange format)
 - Domena je svet lesenih kock zaobjet v ontologiji Blocks-World

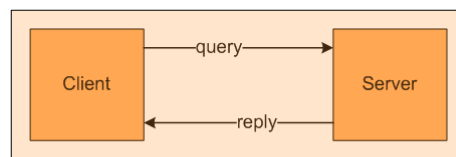
```
(tell
  :sender      Agent1
  :receiver   Agent2
  :language   KIF
  :ontology   Blocks-World
  :content    (AND(Block A)(Block B)(On A B))
```

- Namesto KIF-a bi lahko uporabljali kakšen drugi jezik (PROLOG, LISP, SQL, ipd.)
- Ontologija je odvisna od domene

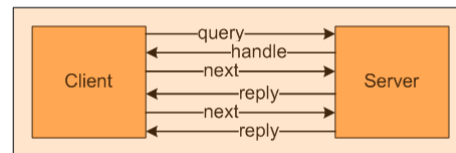
KQML - Model odjemalec strežnik

- Agenti med seboj delujejo kot strežniki in odjemalci

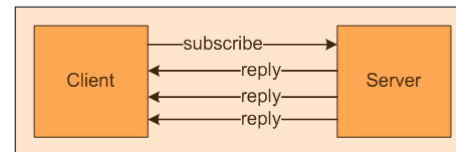
- Sinhrona komunikacija
 - Čaka na odgovor



- Strežnik vzdržuje stanje
 - Odgovori se pošiljajo po potrebi na zahtevo



- Asinhrona komunikacija
 - Odgovori se pošiljajo med delovanjem strežnika



KQML

- Gnezdenje sporočil

```
(forward
  :from      Agent1
  :to        Agent2
  :sender     Agent1
  :receiver   Agent3
  :language   KQML
  :ontology   kqml-ontology
  :content    (tell
               :sender     Agent1
               :receiver   Agent2
               :language   KIF
               :ontology   Blocks-World
               :content     (AND(Block A)
                             (Block B)(On A B))))
```

KQML

- Kategorije KQML struktur (nameni izrazov) – performatives
 - Osnovni (*evaluate, ask-one, ask-all,...*)
 - Povpraševanja po večjem številu odgovorov (*stream-in, stream-all,...*)
 - Odgovori (*reply, sorry,...*)
 - Splošni informacijski (*tell, achieve, cancel, untell, unachieve,...*)
 - Generatorji (*standby, ready, next, rest,...*)
 - Določevalci sposobnosti (*advertise, subscribe, monitor,...*)
 - Mrežni (*register, unregister, forward, broadcast,...*)

KQML primer

- **advertise** – za obveščanje o svojih sposobnostih
- Agen2 sporoči Agentu1

```
(advertise
  :sender      Agent2
  :receiver    Agent1
  :language    KQML
  :ontology    kqml-ontology
  :content     (ask-all
                :sender      Agent1
                :receiver    Agent2
                :in-reply-to id1
                :language    Prolog
                :ontology    Blocks-World
                :content     "on(X,Y)"))
```

KQML primer

- Agent1 vpraša:

```
(ask-all
  :sender      Agent1
  :receiver    Agent2
  :in-reply-to id1
  :reply-with  id2
  :language    Prolog
  :ontology    Blocks-World
  :content     "on(X,Y)"))
```

- Agent2 odgovori:

```
(tell
  :sender      Agent2
  :receiver    Agent1
  :in-reply-to id2
  :language    Prolog
  :ontology    Blocks-World
  :content     "[on(a,b),on(c,d)]")
```

Format za izmenjavo znanja

- Naravni jezik je zelo ekspresiven, vendar velikokrat so izjave dvoumne
- KIF – Knowledge Interchange Format
- Jezik, ki temelji na logiki (predikatih prvega reda)
- Z njim opišemo vse kar nas zanima (v realnem svetu)
- Prefiksen zapis logičnih predikatov prvega reda

KIF - primeri

- Podatki o plači za tri osebe

```
(salary 0394-232 finance 72000)
(salary 0847-475 management 92000)
(salary 0233-756 production 9000)
```

- Prvi čip je večji od drugega

```
(> (* (width chip1) (length chip1))
    (* (width chip2) (length chip2)))
```

- Če je x realno, n pa celo število, potem je $x^n > 0$

```
(=> (and (real-number ?x)
         (even-number ?n))
     (> (expt ?x ?n) 0))
```

Ontologije

- Ontologija je specifikacija objektov, konceptov in relacij na nekem področju zanimanja
- Koncepti: unarni predikati logike prvega reda
- Relacije: več-arni predikati logike prvega reda
- Primer: kocka je fizični objekt

```
∀x (Block x) ⇒ (PhysicalObject x)
```

- Primer: pravilo, ki definira hierarhijo tipov

```
(class Block)  
(Class PhysicalObject)  
(subclassOf Block PhysicalObject)  
∀x,y,z (instanceOf x y) ∧ (subclassOf y z)  
⇒ (instanceOf x z)
```

Ontologije

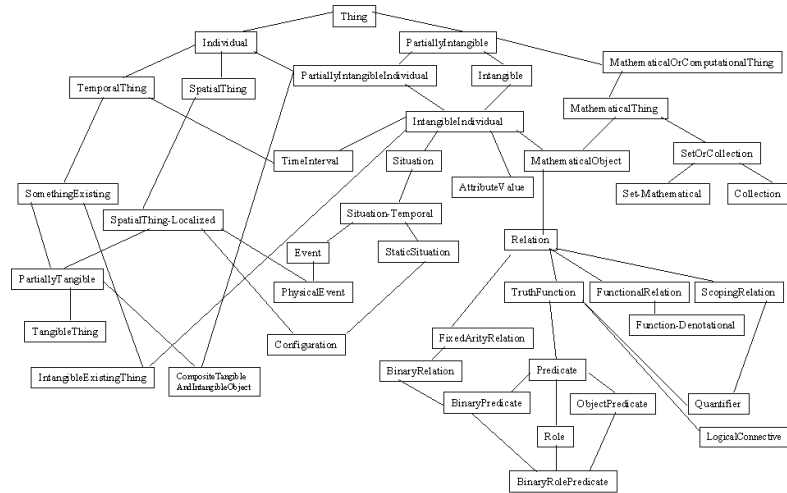
- Ontologija opisuje tudi relacije
- Vsebuje razrede – primerke razredov pa ne nujno
- Analogija s shemo podatkovne baze, ne s podatki v njej
- Agent mora predstavljati svoje znanje z uporabo besednjaka izbrane ontologije
- Vsi agenti, ki uporabljajo isto ontologijo razumejo besede, ki se uporabljajo pri komunikaciji
- Bolj podrobna definicija relacij:

```
(domain On PhysicalObject)  
(range On PhysicalObject)
```

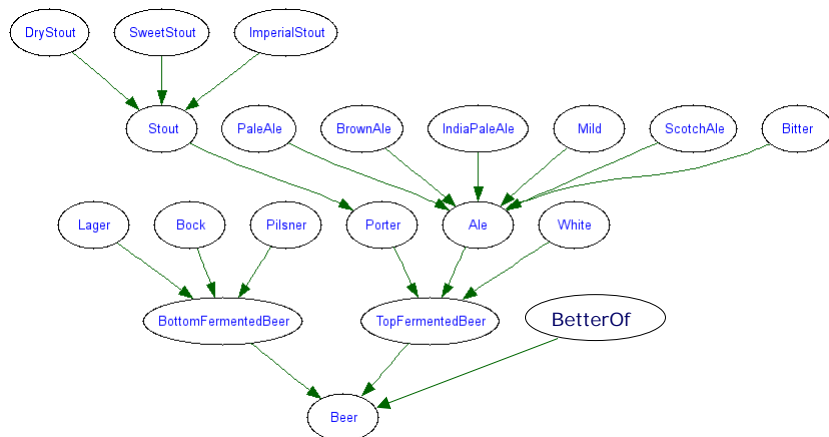
▪ OK: (On A B)

▪ NOK: (On A Dream1)

Primer ontologije



Primer ontologije



Interakcijski protokoli

- **Komunikacijski** protokol opiše mehanizme za interakcijo **posameznih** sporočil (vrste, namen, ipd.)
- **Interakcijski** protokol opiše/predpiše potek izmenjave **zaporedja** sporočil – pogovor
- V primeru agentov, ki sodelujejo med sabo je cilj interakcijskega protokola:
 - Zagotoviti koherentnost delovanja sistema kot celote
 - Brez eksplicitnega globalnega nadzora
 - (ob upoštevanju avtonomije posameznega agenta)
- Pomembni aspekti:
 - Določiti skupne cilje
 - Določiti skupne naloge
 - Izogibati se nepotrebnim konfliktom
 - Združiti znanje in zaznave

Interakcijski protokoli

- Različni protokoli in mehanizmi
 - Koordinacijski protokoli
 - Protokoli za sodelovanje
 - Pogodbene mreže (Contract nets)
 - Sistemi s tablo (Blackboard systems)
 - Pogajanja (Negotiations)
 - Ohranjanje prepričanja večagentnega sistema (Multiagent Belief Maintenance)
 - Trži mehanizmi (Market mechanisms)
- Združbe agentov

Koordinacijski protokoli

- V omejenem okolju morajo agentje koordinirati aktivnosti:
 - Akcije agentov so soodvisne
 - Skupaj morajo težiti k skupnemu cilju
 - Upoštevati morajo globalne omejitve
 - Nobeden izmed agentov nima vse potrebne
 - Kompetence
 - Virov
 - Informacije
- Agenti morajo
 - Posredovati informacijo drugim
 - Sinhronizirati akcije
 - Se izogibati redundantnemu reševanju problemov

Porazdeljen nadzor

- Porazdeljen koordiniran sistem -> Porazdeljenost podatkov in nadzora
 - Vsak agent je avtonomen
 - V generiranju novih akcij
 - V odločanju h katerim ciljem težiti
 - Znanje je razpršeno po sistemu
- > težje zagotoviti koherentnost globalnega delovanja

Graf ciljev

- Akcije agentov -> iskanje po AND/OR grafu ciljev
 - Predstavitev odvisnosti med cilji in sredstvi potrebnimi za doseganje preprostih ciljev (listi v grafu)
 - Posredne odvisnosti med akcijami
- Aktivnosti ki zahtevajo koordinacijo:
 - Definirati graf ciljev
 - Določiti dele grafa posameznim agentom
 - Nadzorovati odločitve o tem katere dele grafa raziskovati
 - Prečkati graf
 - Poročati o uspešnem prehodu

Zaveze in konvencije

- Vsak agent mora vedeti kaj nameravajo drugi
 - Kakšne zaveze so dali (obljube)
 - Kakšne konvencije uporabljajo (kako spreminjajoče okolje lahko vpliva na te obljube)
- **Zaveze** zagotavljajo visok nivo predvidljivosti
 - Ostali agenti lahko načrtujejo njihove akcije glede na to
- **Konvencije** zagotavljajo prilagodljivost zavez
 - Omejujejo pogoje v katerih se lahko zaveze ne izpolnijo, oz. se ponovno pretehtajo
 - Predpisujejo kaj se v takih situacijah zgodi
 - Zaveza se obdrži in se mora izpolniti
 - Zaveza se lahko popravi glede na trenutno situacijo
 - Zaveza se opusti

Zaveze in konvencije

- Če se situacija ne spremeni, agent teži k izpolnitvi zavez
⇒ njegove zaveze morajo biti med seboj konsistentne ter konsistentne z njegovim prepričanjem
- Konvencije pomagajo urejati zaveze
- Ko se spremeni cilj (zaveza), ki vpliva tudi na druge agente, mora agent to spremembo sporočiti ostalim (zainteresiranim) agentom -> **socialna konvencija**

Socialna konvencija

- Primer socialne konvencije v primeru omejenih komunikacijskih zmogljivosti:

```
LIMITED_BANDWIDTH SOCIAL CONVENTION

INVOKE WHEN
    Local commitment dropped
    Local commitment satisfied

ACTIONS
Rule1: IF Local commitment satisfied
    THEN inform all related commitments

Rule2: If local commitments dropped because
    unattainable or motivation not present
    THEN inform all strongly related commitments

Rule2: If local commitments dropped because
    unattainable or motivation not present
    AND communication resources not overburned
    THEN inform all weakly related commitments
```

Skupne zaveze

- Skupna akcija => skupni cilj
- Izmenjavati si morajo vsaj informacije o:
 - Status njihove obveze k skupnemu cilju
 - Status njihove obveze do ekipe
- Če se ta status spremeni, morajo biti obveščeni vsi člani ekipe
- Če agent spozna, da se je zaveza nekega drugega agenta s katerim sta skupaj v akciji, spremenila, mora premisliti tudi svoje zaveze

Konvencija za skupne zaveze

BASIC JOINT-ACTION CONVENTION

INVOKE WHEN

Status of commitment to joint action changes
Status of commitment to attaining action in
present team context changes

ACTIONS

Rule1: IF Status of commitment to joint action changes
OR
IF Status of commitment to present team context
changes
THEN inform all other team members of these
changes

Rule2: If Status of joint commitment of a team member
changes
THEN Determine whether joint commitment still
viable

Zaveze in konvencije

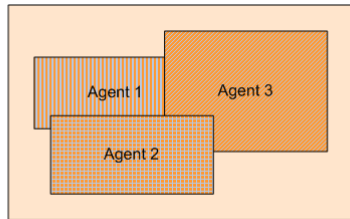
- Zaveze in konvencije so glavni gradniki **koordinacije**
 - **Zaveze** zagotavljajo strukturo predvidljive interakcije
 - **Socialne konvencije** zagotavljajo medsebojno podporo

Protokoli za sodelovanje

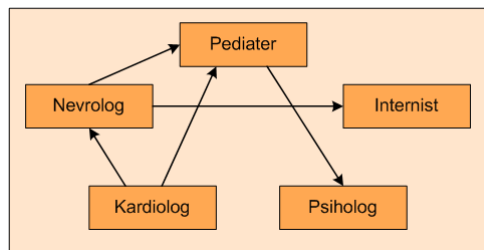
- Osnovna strategija je ponavadi razčleniti in porazdeliti naloge
 - Alternativne dekompozicije
 - Upoštevati je potrebno
 - zmogljivosti agentov
 - Interakcije med podnalogami
 - Konflikte med agenti
- Naloge lahko razdeli
 - Načrtovalec sistema (jih vprogramira)
 - Agenti s hierarhičnim načrtovanjem
 - Razdelitev je že zajeta v predstavitvi problema (AND-OR graf)
- Naloge so lahko razdeljene
 - Prostorsko (glede na položaj vira informacije ali odločitve)
 - Funkcionalno (glede na ekspertizo)

Dekompozicija nalog

- Prostorska dekompozicija:



- Funkcionalna dekompozicija:



Razdelitev nalog

- Kriteriji za razdelitev nalog:
 - Izogibati se preobremenitvi kritičnih virov
 - Dodeliti naloge agentom s primernimi zmogljivostmi
 - Agenti s širšim pogledom naj dodeljujejo naloge ostalim agentom
 - Naj bo del odgovornosti deljen za zagotovitev koherentnosti
 - Zelo medsebojno odvisne naloge dodeliti sosednim agentom (prostorsko, semantično)
 - Ponovno je potrebno razdeliti naloge, če je potrebno izvršiti nujno nalogo.

Mehanizmi za razdelitev nalog

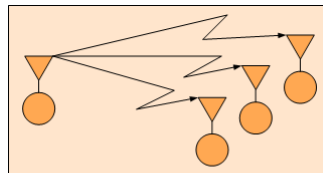
- Pogosti mehanizmi za razdelitev nalog:
 - Mehanizmi trga (Market mechanisms)
 - Naloge so porazdeljene s skupnim dogovorom ali vzajemnim izborom
 - Pogodbene mreže (Contract net)
 - Cikli naznanilo, ponudba, nagrada
 - Večagentno načrtovanje (Multiagent planning)
 - Načrtovalni agenti imajo odgovornost za razdelitev nalog
 - Organizacijske strukture
 - Agenti imajo stalne obveznosti za določene naloge

Pogodbene mreže

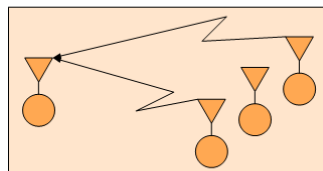
- Interaktivni protokol za sodelovanje med agenti pri reševanju skupnega problema
- Modelira dogovore v poslovanju pri menjavi blaga in storitev
- Rešitev za problem povezave (connection problem): najti primerne agenta za delo na določeni nalogi
- *Manager* – agent, ki ima problem, ki ga je potrebno rešiti
- *Pogodbeniki* – agenti, ki (mogoče) lahko rešijo problem

Pogodbene mreže - proces

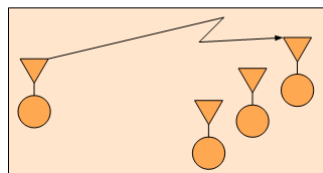
- Manager objavi svoj problem vsem potencialno zainteresiranim agentom.



- Agenti ocenijo sporočil. Nekateri agenti pošljejo ponudbe.



- Manager sklene pogodbo z najustreznejšim ponudnikom. Komunikacija nato po potrebi nadaljujeta neposredno.



Pogodbene mreže - proces

- Proces pri managerju:
 - Objavi sporočilo o nalogi
 - Sprejmi in oceni ponudbe
 - Skleni pogodbo z najugodnejšim ponudnikom
 - Sprejmi in sestavi rezultate
- Proces pri pogodbeniku
 - Sprejmi objavo o nalogi
 - Oceni svojo zmožnost za odgovor
 - Odgovori (zavrni ali ponudi)
 - Opravi nalogo, če je njegova ponudba sprejeta
 - Sporoči rezultate

Pogodbene mreže - proces

- Vloge managerja in pogodbenika niso v naprej določene
- Lahko so pri določenih nalogah obrnjene
- Struktura za objavo naloge:
 - Naslovljenec
 - Specifikacija primernosti za izvedbo
 - Kratek opis naloge
 - Specifikacija ponudbe (kaj le-ta mora vsebovati)
 - Rok za ponudbe
- Potencialni pogodbeniki ovrednotijo objavo naloge in se zanjo lahko potegujejo
- Manager potem sprejme zanj najugodnejšo ponudbo (lahko tudi pred iztekom roka)
- Pogodbenika obvesti z ustreznim sporočilom
 - Ni mu pa potrebno obvestiti ostalih zainteresiranih

Pogodbene mreže

- Manager lahko tudi ne dobi nobene ponudbe
 - če so vsi potencialni pogodbeniki zasedeni
 - Če je potencialni ponudnik prost, a oceni nalogo kot manj pomembno od ostalih
 - Nobeden izmen ponudnikov ni sposoben opraviti zadane naloge
- Z tem primeru lahko manager zahteva takojšen odgovor:
 - Sposoben vendar zaseden
 - Nesposoben
 - Nezainteresiran
- Manager nato ustrezno prilagodi svojo zahtevo
- V pogodbenih mrežah je pogodba sklenjena takoj, brez pogajanj
 - sprejet pogodbenik lahko pogodbo samo sprejme ali zavrže
 - Enostaven mehanizem, ki lahko poveča učinkovitost

Sistemi s tablo

- Blackboard systems
- Skupina agentov (specialistov) skupaj rešuje problem
- Problem in začetni podatki se zapišejo na tablo
- Specialisti spremljajo delne rezultate in gledajo kdaj lahko uporabijo svojo ekspertizo
- Če uspejo rešiti del naloge rezultate napišejo na tablo
- Ostali udeleženci sprejmejo te rezultate in nandalje poizkušajo rešiti nalogo
- Ta proces se nadaljuje dokler naloga ni rešena
- Specialist = vir znanja (Knowledge Source)

Sistemi s tablo

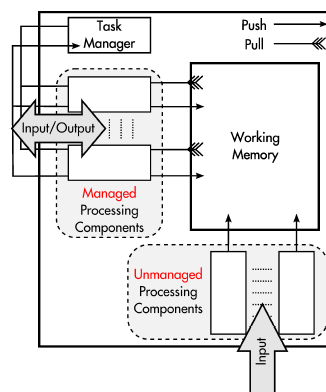
- Značilnosti sistemov s tablo:
 - Neodvisnost ekspertiz
 - Vsak specialist je neodvisen in nepovezan z otalimi specialisti.
 - Različnost tehnik za reševanje problemov
 - Vsak specialist lahko uporablja svoje metode in predstavitve, ki so ostalim skrite.
 - Fleksibilna predstavitev informacije na tabli
 - Ni predpisana narava informacije, ki se predstavlja na tabli.
 - Skupni interakcijski jezik
 - Specialisti se morajo sporazumevati v vsem razumljivem jeziku.
 - Aktivacije sprožajo dogodki
 - Dodajanje, sprememba ali brisanje informacije
 - Tabla obvešča specialiste o zanimivih spremembah
 - Postopna gradnja rešitve
 - Specialisti lahko izboljšujejo/popravljajo že obstoječe podrešitve ali začnejo nov način reševanja
 - Nadzorni mehanizem

Sistemi s tablo

- Nadzorni mehanizem
 - Nadzoruje potek reševanja problema
 - Specialist za usmerjanje reševanja problema
 - Izbira specialiste, ki bodo reševali (pod)problem
 - Vsak specialist obvesti nadzornega specialista o kvaliteti in ceni njegovega opravila
 - Nadzorni specialist se odloči, ali mu bo nato naloga zaupana ali ne

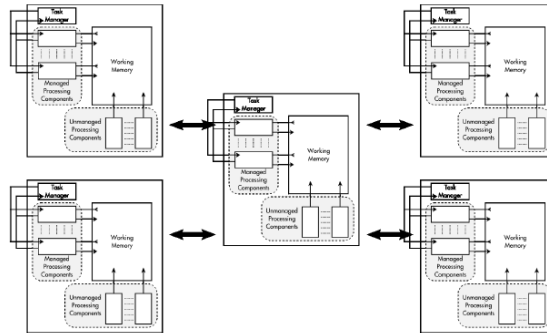
Primer - Arhitektura CAS

- Ena podarhitektura:
 - Agenti = Komponente
 - Tabla = delovni pomnilnik (WM)
 - Komponente komunicirajo preko delovnega pomnilnika
 - Nadzorni specialist = upravitelj (task manager) - upravlja s procesiranjem



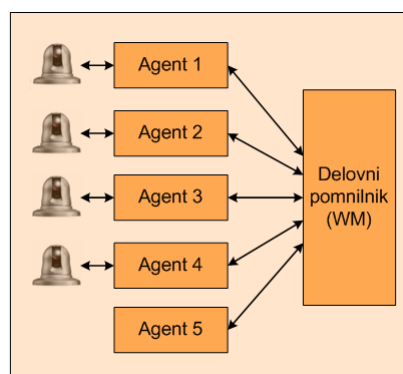
Primer - Arhitektura CAS

- CoSy architecture schema
- Zbirka rahlo povezanih podarhitektur
- Zelo različne komponente – zelo nehomogen sistem
- Podarhitekture komunicirajo z drugimi podarhitekturami preko delovnega pomnilnika



Primer PIPT CAS arhitektura

- CAS za komunikacijo med agenti
- Delovni pomnilnik kot tabla



Pogajanja

- Pogajanje je proces v katerem se doseže skupna odločitev med dvema ali več agenti od katerih vsak teži k izpolnitvi svojega cilja.
- Agenti najprej komunicirajo svoje predloge (ki so lahko nasprotujoči)
- Nato delajo kompromise ali iščejo alternativne rešitve, da se zblížajo v stališčih
- Glavne značilnosti:
 - Jezik, ki ga uporabljajo udeleženci
 - Protokol za pogajanja
 - Odločitveni proces vsakega udeleženca
- Tehnike sistemov za pogajanja se osredotočajo
 - Na okolje
 - Na agenta

Mehanizem za pogajanja

- Tehnike, ki se osredotočajo na **okolje**:
 - Kako naj bodo pravila, ki veljajo v okolju, oblikovana, tako da bodo agenti v tem okolju medsebojno delovali produktivno in pošteno?
- Tak mehanizem za pogajanja naj bi izpolnjeval kriterije:
 - Učinkovitost
 - Agenti naj ne bi zapravljali virov pri doseganju sporazuma
 - Stabilnost
 - Vsi agenti naj bi upoštevali dogovorjene strategije
 - Enostavnost
 - Pogajalski mehanizem naj bo čim manj potraten (računsko in v pasovni širini)
 - Porazdeljenost
 - Mehanizem naj ne zahteva centralnega odločevalca
 - Simetričnost
 - Vsi agenti naj bodo obravnavani enako

Mehanizem za pogajanja

- Okolja, ki so osredotočena na nalogo:
 - Vsi agenti lahko rešijo nalogo brez pomoči drugih
 - Lahko pa agenti s sodelovanjem povečajo koristnost
- Primer: agenti za snemanje datotek z interneta
 - Skupne datoteke bi lahko sneli samo enkrat in si jih delili
- Mehanizem:
 - Vsak agent napove dokumente, ki jih potrebuje
 - Skupni dokumenti se dodelijo posameznim agentom z žrebom
 - Agenti plačajo za dokumente, ki jih snamejo
 - Agenti imajo dostop do svojih in skupnih dokumentov
- Tak mehanizem je enostaven, simetričen, porazdeljen in učinkovit

Mehanizem za pogajanja

- Tehnike, ki se osredotočajo na **agenta**:
 - V danem okolju, kakšna je najboljša strategija za agenta
- Ekonomsko racionalni agenti
 - Množica agentov je majhna
 - Agenti imajo skupni jezik in abstrakcijo problema
 - Agenti težijo k skupni rešitvi
- Poenoten pogajalski protokol:
 - Agenti sklenejo *sporazum* – skupni načrt, ki bi izpolnil vse njihove cilje
 - *Koristnost sporazuma* – vsak agent jo hoče maksimizirati
 - Agenti se dogovarjajo o *pogajalski množici* (množica vseh sporazumov, ki so koristni za vse agente)

Ohranjanje prepričanja večagentnega sistema

- Multiagent Belief Maintenance
- Truth Maintenance System (TMS) ohranja integriteto znanja agenta, ki naj bo
 - Stabilno
 - Dobro osnovano
 - Logično konsistentno
- Stabilno stanje
 - Vsak podatek za katerega obstaja dokaz je pozitiven (is believed)
 - Vsak podatek, za katerega ne obstaja dokaz, je negativen (is disbelieved)
- Dobro osnovano (well-founded)
 - Nobena množica prepričanj ni odvisna od druge množice prepričanj
- Logično konsistentno stanje znanja
 - Stabilno, brez kontradikcij
- Ostale zaželjene lastnosti baze znanja:
 - Popolnost, jedrnatost, natančnost, učinkovitost

Večagentni TMS

- Enoagentni TMS – ohranja stabilna stanja baze znanja s prilagajanjem pozitivnih in negativnih podatkov
- Večagentni TMS morajo ohranjati integriteto njihovega znanja in integriteto komunicirane informacije
- Vsak podatek je označen kot
 - **INTERNAL** (pozitiven, ker ga je agent potrdil)
 - **EXTERNAL** (pozitiven, ker ga je neki drugi agent potrdil)
 - **OUT** (negativen)
- Agenti komunicirajo o podatkih in labelah
- Labele morajo biti konsistentne po vseh agentih
- Nek podatek mora biti **INTERNAL** vsaj enemu agentu in **INTERNAL** ali **EXTERNAL** ostalim
- Strategije za ustrezno osveževanje label

Tržni mehanizmi

- Market mechanisms
- Tržni mehanizmi so primerni za koordinacijo aktivnosti večjega števila agentov z minimalno direktno komunikacijo med agenti samimi
- Vsi relevantni podatki na osnovi katerih agent sprejema odločitve so zajeti v *ceni*
- Dva tipa agentov
 - Potrošniki, ki izmenjujejo dobrine
 - Proizvajalci, ki predelujejo dobrine v druge dobrine
- Agenti ponujajo različne ponudbe za dobrine, izmenjave pa se zgodijo po trenutnih tržnih cenah
- Vsak agent teži k temu, da bi povečal dobiček ali svojo vrednost

Tržni mehanizmi

- Za prevedbo problema v ustrezno obliko je potrebno opredeliti:
 - Dobrine s katerimi se trguje
 - Agente potrošnike
 - Agente proizvajalce z njihovimi proizvodnimi zmogljivostmi
 - Predvideno obnašanje udeležencev pri trgovanju
- Cena ene dobrine vpliva na ponudbo in povpraševanje drugih dobrin
- Trg doseže ravnotežje
 - Potrošniki težijo k maksimizaciji njihove vrednosti ob upoštevanju razpoložljivega proračuna
 - Proizvajalci težijo k maksimizaciji njihovega dobička ob upoštevanju njihovih tehnoloških zmogljivosti
 - Neto povpraševanje je nič za vse dobrine
- Ravnotežje ustreza neki porazdelitvi virov in določa akcije, ki jih morajo agenti izvesti

Tržni mehanizmi

- Ekonomsko racionalno obnašanje agenta
 - Agentove preference so podane
 - Znanje o učinkih agentovih akcij je podano
 - Racionalna akcija maksimizira njegove preference
- Vse faktorje je potrebno nekako pretvoriti v ceno
- Kvantitativno in probabilistično modeliranje stroškov in učinkov akcij
 - > učinkoviti načrti akcij

Združbe agentov

- Agenti delujejo v sodelovanju z drugimi -> več agentov -> veliko agentov -> združbe agentov
- Velika informacijska okolja, npr. inteligenen nadzor nad avtocestami (inteligentna vozila)
 - > porazdeljen zajem podatkov, sklepanje, načrtovanje, nadzor
- Agenti morajo
 - Dolgo živeti
 - Se prilagajati (se učiti)
 - Biti socialni
 - Morajo vzajemno sodelovati z ostalimi agenti za dosego njihovih ciljev
 - Morajo se zanašati na ostale agente in uporabljati njihovo znanje

Združbe agentov

- Agenti v združbi igrajo različne vloge
 - Skupina definira vloge
 - Vloge definirajo obveze
 - Agenti se (prostovoljno) pridružijo skupini preko določene vloge – sprejmejo vlogo in obveze
 - Skupina določi socialni kontekst v katerem agenti sodelujejo med sabo

Združbe agentov

- Veliko agentov -> sociologija, teorija organizacije
- Od togih odjemalcev/strežnikov arhitektur k fleksibilni povezavi enakovrednih porazdeljenih agentov
- Tržni mehanizmi, ekonomski modeli, zreducirajo agente na sebične agente
 - > agenti potrebujejo nekaj več kot maksimizacijo dobička

Socialna odvisnost

- Socialne zaveze so zaveze enega agenta do drugih agentov
- Socialne zaveze določajo (omejujejo) obnašanje agentov
- Socialna odvisnost:
 - Agent x je odvisen od agenta y v odnosu do akcije a za dosego cilja p
 - Če je p cilj agenta a
 - In je x nezmožen opraviti a
 - In je y zmožen opraviti a
 - Socialna odvisnost je lahko prostovoljna (se sprejme z vlogo)
 - Agent se je lahko ne zaveda
 - Vzajemna odvisnost
 - x in y sta odvisna drug od drugega pri doseganju skupnega cilja p
 - Potrebno sodelovanje
 - Recipročna odvisnost
 - x in y sta odvisna drug od drugega pri doseganju vsak svojega cilja p_x in p_y

Ekipe socialnih agentov

- Skupina socialnih agentov tvori ekipo, ko
 - Imajo vsi agenti skupni cilj
 - Vsak agent mora prispevati svoj delež pri doseganju skupnega cilja skupine ali podskupine
 - Vsak agent sprejema zahteve za njegov prispevek.

Porazdeljene inteligentne programske tehnologije

Porazdeljeno reševanje problemov in načrtovanje

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Poglavje 3

Porazdeljene inteligentne programske tehnologije

Porazdeljeno reševanje problemov

- Distributed problem solving
- Področje Porazdeljene umetne inteligence
- Agenti naj delajo skupaj s ciljem učinkovitega reševanja problemov, ki zahtevajo skupinsko delo
- Porazdeljeni viri:
 - Znanje
 - Zmožnosti
 - Informacije
 - Ekspertiza
- Agent ne zmore sam rešiti problema
- ali ga lahko v skupini zmore rešiti bolje
 - Hitreje
 - Bolj popolno
 - Bolj natančno
 - Bolj gotovo

Porazdeljene inteligentne programske tehnologije, Porazdeljeno reševanje problemov in načrtovanje

2

Koherenca in kompetenca

- Reševanje porazdeljenih problemov zahteva:
 - Koherenco v skupini (agentje želijo delati skupaj)
 - V primeru PRP so agenti načrtovani na ta način
 - **Kompetenco** v skupini (agentje vedo kako delati skupaj)
 - V PRP poudarek na kompetenci


Primeri porazdeljenih problemov

- Načrtovanje avtomobila
 - Splošne zahteve
 - 4 kolesa
 - Zmogljiv motor
 - Ne prevelik motor
 - Mehki sedeži
 - Moderen design
 - Aerodinamična oblika
 - Potovalni računalnik
 - Ustrezati vsem predpisom, itn.
 - Rešitev (načrt avtomobila) mora ustrezati določenim kriterijem (po njem nato avtomobil izdelajo)
 - Zelo heterogen problem
 - Vsak agent se ukvarja z enim (ali večimi) podproblemi
 - Nato se delne rešitve združijo v rešitev celotnega problema
- Nadzorni sistem za nadzor zračnega prometa
- Nadzor proizvodnje
- Krizni management

Porazdeljeno načrtovanje

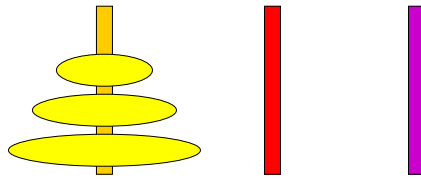
- Agentje v porazdeljenem okolju morajo načrtovati njihovo skupno delo:
 - Razdeliti probleme v podprobleme
 - Si porazdeliti te podprobleme
 - Si izmenjati rešitve teh podproblemov
 - Združiti delne rešitve v skupne rešitve
- Porazdeljeno načrtovanje je zelo povezano s porazdeljenim reševanjem problemov

Primeri porazdeljenih problemov

- Različne motivacije za porazdeljeno reševanje problemov
 - Paralelizem za pospešitev reševanja problema
 - Distribuirane sposobnosti
 - Porazdelitev prepričanj, procesiranja, podatkov,...
 - Porazdelitev rezultatov
- 
- Hanojski stolpi
 - Porazdeljena mreža senzorjev za nadzor premikov vozil na velikem področju
 - Geografsko porazdelitev agentov
 - Porazdelitev zajemanja podatkov
 - Porazdeljeni nadzor premikov vozil
 - Porazdelitev procesiranja, razmišljanja, načrtovanja...
 - Potrebne manj komunikacije
 - Porazdeljena dostava
 - Porazdeljeno načrtovanje (brez enega odpravitelja)
 - Ponovno načrtovanje ob spremembah okoliščin
 - Potrebne manj komunikacije

Hanojski stolpi

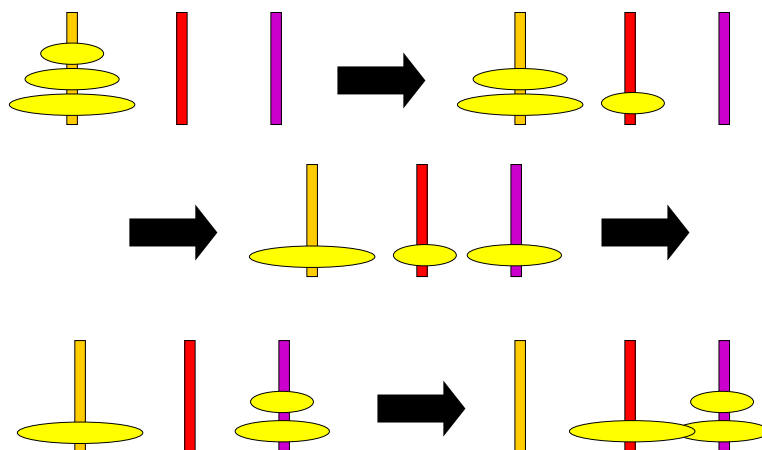
- Problem hanojskih stolpov
 - Imamo 3 palice
 - Na prvi palici je n diskov
 - Vse diske je potrebno prestaviti na drugo palico
 - Naenkrat se lahko prestavi samo en disk
 - Nikoli ne sme biti večji disk položen na manjšega



Porazdeljene inteligentne programske tehnologije, Porazdeljeno reševanje problemov in načrtovanje

7

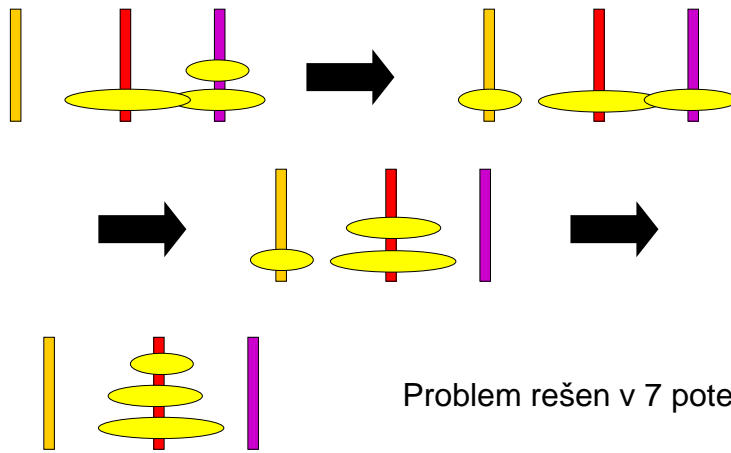
Hanojski stolpi



Porazdeljene inteligentne programske tehnologije, Porazdeljeno reševanje problemov in načrtovanje

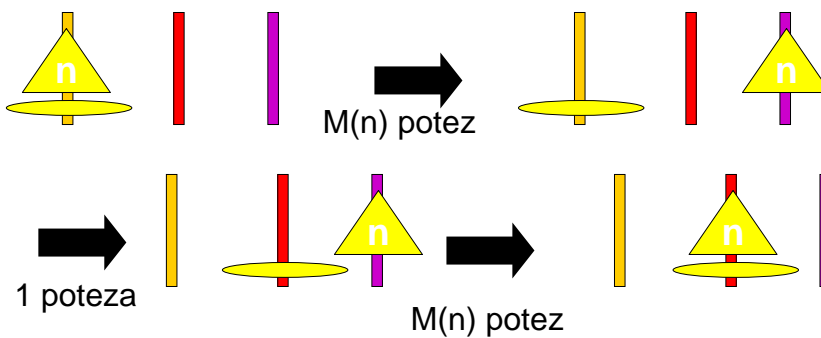
8

Hanojski stolpi



Hanojski stolpi

Dekompozicija problema z $n+1$ diski:



$$\text{Skupaj: } M(n+1) = 2 M(n) + 1 = 2^n - 1$$

$N=3$: 7 potez

$N=64$: 18.446.744.073.709.551.615 potez (580 bilijonov let)

Delitev nalog

- Strategija za porazdeljeno reševanje problemov
- Task sharing, task passing
- Ideja: ko ima agent veliko nalog, vpokliče na pomoč manj obremenjene agente
- Glavni koraki:
 - Dekompozicija naloge
 - Generiraj množico nalog (podnalog)
 - Dodelitev nalog
 - Dodeli naloge primernim agentom
 - Izvedba nalog
 - Posamezni agenti izvedejo zadane naloge
 - Pri tem lahko tudi naprej razdelijo naloge drugim agentom
 - Sestavi rezultate
 - Agent rezultate svoje naloge posreduje primernemu agentu
 - Rezultate je potrebno sestaviti v končni rezultat

Delitev nalog – Hanojski stolpi

- Dekompozicija naloge
 - Analiza problema
 - Premakni največji disk, ki še ni na končnem položaju
 - Rekurzivna dekompozicija:
 - Preidi v stanje, ko se lahko največji disk premakne
 - Premakni največji disk
 - Reši še za preostale diske
- Dodelitev nalog
 - Naključno določi nalogo enemu od poljubno mnogo razpoložljivih agentov
- Izvedba nalog
 - Rekurzivna dekompozicija se ustavi, ko se nalog ne da deliti naprej
- Sinteza rezultatov
 - Ko agent reši problem, pošlje rezultate naprej
 - Ko prejme rešitve vseh agentov, ki jim je dodelil naloge, rezultate sestavi in pošlje naprej

Hanojski stolpi – časovna kompleksnost

- Problem hanojskih stolpov je optimalen za hierarhično dekompozicijo
- Časovna kompleksnost se drastično zmanjša:
 - Izčrpnem preiskovanju prostora rešitev
 - > polinomska kompleksnost
 - Hierarhična dekompozicija problema
 - > Linearna časovna kompleksnost
 - Vzporedno reševanje podproblemov
 - > Logaritemska časovna kompleksnost
- V primeru naloge, ki je idealna za dekompozicijo, se lahko s paralelizmom reševanje problema zelo izboljša!

Predpostavke

- Ponavadi problemi ne ustrezajo vsem predpostavkam optimalnega problema za dekompozicijo:
 - Vsak podproblem je neodvisen od drugih
 - Hierarhična rešitev najde enako optimalno rešitev kot izčrpno preiskovanje
 - Število abstraktnih nivojev lahko narašča z velikostjo problema
 - Delitev nalog na podnaloge (razmerje med velikostmi nivojev) je optimalno
 - Problem se lahko razdeli v enako velike podprobleme (*In aurea media*)
 - Število razpoložljivih agentov je dovolj veliko (kot je listov v drevesu)
 - Procesi dekompozicije problema, porazdeljevanja podproblemov in zbiranja rezultatov zahtevajo le zanemarljiv čas

Delitev nalog – heterogeni sistemi

- Težko (ali predrago) je zgraditi agente (ali naučiti ljudi), da bi bili kompetentni za reševanje vseh problemov
- Po potrebi se poveže kompetentne specialiste za posamezna področja in kombinira njihove ekspertize za rešitev kompleksnih problemov
- Ko so kompetence agentov različne, dodeljevanje nalog ni trivialno
- Tabela s kompetencami agentov
- Dinamično dodeljevanje nalog s komunikacijo (kateri agenti so na razpolago, itn.)

Pogodbene mreže

- Razdeljevanje nalog po principu pogodbenih mrež
 - Neposredne pogodbe (manager se obrne neposredno na želenega pogodbenika)
 - Usmerjeno naslavljanje (manager sporoči podnaloge skupini izbranih potencialnih pogodbenikov)
- Manager zbere ponudbe in nato izbere pogodbenike
- Problem nastane, če nobeden od agentov ni na razpolago

Razširjeno naslavljanje

- Manager mogoče ne pozna vseh potencialnih pogodbenikov (njegova tabela ni popolna)
- Manager lahko objavi naloge širši množici agentov (tudi tistim za katere še ne ve)
- Neposredno in usmerjeno naslavljanje se uporablja kot "predpomnilnik":
 - za pohitritev sklepanja pogodb
 - Za zmanjšanje potrebne komunikacije
- Tabela sposobnosti agentov se dinamično posodablja

Ponovni poizkus

- Če nobeden agent ni na voljo, lahko manager počaka, nato pa ponovno pošlje poziv
- Časovni presledek med posameznimi poskusi
 - Ne sme biti predolg (se zapravljajo njegovi resursi)
 - Ne sme biti prekratek (preveč komunikacije)
- Lahko se obrne protokol na glavo:
 - Potencialni pogodbeniki objavijo njihovo razpoložljivost
 - Manager po prejemu takšnega obvestila pošlje ponudbo
- Sistem lahko tako iterira med objavami za
 - Naloge in
 - Razpoložljivost

Revizija objave

- Manager pošlje potencialnim pogodbenikom v objavi nalog tudi specifikacijo primernosti za izvedbo
- Mogoče je ta specifikacija preveč omejujoča, zato jo lahko nato sprosti
- V specifikaciji se lahko navedejo managerjeve preference glede pogodbenikov
- Manager se lahko s tem odloči
 - Da bo (lahko dolgo) čakal na optimalnega pogodbenika
 - Da bo (čimprej) sklenil pogodbo s suboptimalnim pogodbenikom

Alternativne dekompozicije

- Manager lahko poizkusiti razgraditi problem na drugi način, tako da bodo na voljo agent za rešev na novo nastalih podproblemov
- Manager lahko najprej preveri razpoložljivost agentov in nato na tej osnovi razgradi problem

Delitev nalog – porazdeljena mreža senzorjev

- Vsak agent dobi za nalogo nadzor določenega geografskega dela ozemlja
- Agent ima potrebno znanje, vendar ne more zaznati celotnega območja (samo lokalno)
- Agent mora spoznati, da ne more sam rešiti celotne naloge, ampak, da mora zato vpoklicati na pomoč druge agente
- Razstavi nalogo, objavi podnaloge, izbere pogodbenika...

- **Objava naloge:**
 - Specifikacija primernosti za izvedbo
 - Primerni agenti morajo biti locirani na pravem mestu in morajo imeti primerne senzorje
 - Kratek opis naloge
 - Agent oceni, če je zanj zanimiva
 - Specifikacija ponudbe
 - Agent sestavi ponudbo po tej specifikaciji

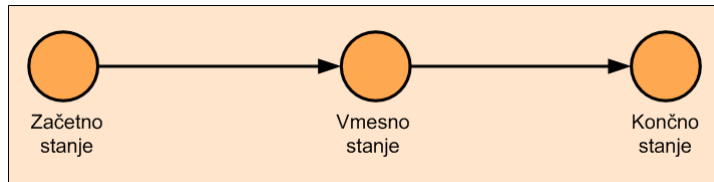
Delitev nalog – porazdeljena mreža senzorjev

- Če manager ne dobi nobene ponudbe:
 - Lahko odneha (ne reši naloge)
 - Lahko ponovno poizkusi (po določenem času)
 - Lahko naslovi širši krog agentov
 - Lahko zmanjša kriterije za primernost agentov
 - Poveča zahtevano območje
 - Zmanjša zahtevano ločljivost
 - Zmanjša zahtevano hitrost kamere (framerate)
 - Lahko razstavi nalogo drugače
- Manager lahko tudi dobi ponudbe, ki zanj niso sprejemljive

- Če manager dobi eno ali več sprejemljivih ponudb:
 - Lahko dodeli nalogo enemu (ali večim) agentom
- Pošiljanje in sprejemanje ponudb je prostovoljno
- Ekipe se izoblikujejo skozi medsebojno selekcijo

Delitev nalog pri neodvisnih nalogah

- Naloge, ki nas pripeljejo iz enega stanja v drugo so neodvisne



- Te naloge se lahko neodvisno in vzporedno izvajajo
- Za končno sintezo rezultatov moramo samo počakati, da se izvedejo vse podnaloge

Delitev nalog pri odvisnih nalogah

- Ponavadi so naloge vsaj deloma odvisne
- Agent, ki mora izvesti nalogo, ki je odvisna od drugih mora počakati, da drugi agenti prej izvršijo svoje (pod)naloge
 - Načrtovanje procesa proizvodnje mora počakati na zaključek izdelave načrta izdelka
 - Načrtovanje se lahko začne prej ob določenih predpostavkah glede na napredek odvisne naloge
- Velikokrat se odvisnost pojavi med samim izvajanjem in ne pri dekompoziciji nalog
 - Pri nadzoru vozil se a odvisnost spreminja glede na premike vozil skozi različna področja
 - Porazdelitev nalog za pokritje področja s senzorji je lažja kot porazdelitev nalog za njegovo uporabo (nadziranje prometa)

Delitev rezultatov

- Različni agenti lahko rešijo nalogo na različne načine
- Z delitvijo rezultatov lahko skupine agentov izboljšajo rezultate:
 - Zaupanje
 - S primerjanjem različnih rezultatov se lahko poveča zaupanje v končni rezultat
 - Popolnost
 - Agenti pokrivajo vsak svoj del naloge, skupna rešitev je precej bolj popolna
 - Zemljevid združen iz lokalnih zemljevidov
 - Natančnost
 - Rešitev drugega agenta lahko pomaga pri povečanju natančnosti rešitve dane naloge
 - Časovni vidik
 - Četudi bi lahko agent rešil nalogo sam, lahko z upoštevanjem rezultatov drugih naloge reši precej hitreje
- Prednosti so jasne – kako delitev rezultatov realizirati?
 - Kaj narediti z deljenimi rezultati
 - Kaj izmenjavati

Funkcionalno natančno sodelovanje

- V primeru hanojskih stolpov so vse podnaloge natančno rešene
- V primeru porazdeljenega nadzora vozil ima vsak agent le delno informacijo o svetu
 - Precej zanesljivo informacijo o svoji ožji lokaciji
 - Manj zanesljivo informacijo o svoji širši lokaciji
 - Nima informacije o preostanku sveta
- Te delne rešitve je potrebno združiti
 - Kreirati je potrebno redundantne poizkusne rezultate
 - Jih na pravi način združiti
- Funkcionalno natančno sodel. (Functionally Accurate Cooperation)
- Funkcionalna natančnost
 - Dobi odgovor (mogoče na začetku napačen)
- Sodelovanje
 - Zahteva iterativno izmenjavo rezultatov

Funkcionalno natančno sodelovanje

- Rezultati, ki jih proizvedejo agenti, se smatrajo kot poizkusni, začasni – hipoteze
- Šele, ko se soočijo z rezultati ostalih, se sprejmejo kot pravi
- Izmenjava poizkusnih delnih rezultatov vpliva na
 - Popolnost
 - Natančnost
 - Zaupanje
- Predstavitve rezultatov morajo omogočati takšno kombiniranje rezultatov
- Skozi iteracije naj bi se rezultati izboljševali
- Če je komunikacije (izmenjevanja rezultatov) preveč, lahko pride do odvrčanja od pravih problemov (vsi agenti se ukvarjajo z istimi (ne najbolj pomembnimi) podproblemi)

Deljeni repozitoriji

- Namesto kopice sporočil poslanim večim naslovnikom uporabljamo en skupni repozitorij
- Sistemi s tablo
- Porazdeljen problem zadovoljevanja omejitev (distributed constraint satisfaction problem – DCSP), razen:
 - Agenti ne vedo čigave omejitve se lahko spremenijo zaradi njihovega načrta
 - > uporaba skupnega repozitorija
 - Agenti lahko po potrebi (v stiski) sprostijo omejitve
 - Agenti lahko vsak trenutek izbirajo med
 - Izboljšanjem nekaterih rešitev
 - Zavrnitvijo nekaterih rešitev
 - Zmanjšanjem zahtev (nekatero prej nesprejemljivo rešitve postanejo sprejemljive)

Iskanje s pogajanjem

- Negotiated search
- Operatorji za izboljšanje porazdeljeno reševanje problemov:
 - **Initiate-solution** : predlagaj novi začetek gradnje rešitve
 - **Extend-solution** : izboljšaj že obstoječo rešitev
 - **Critique -solution** : komentiraj že obstoječo delno rešitev
 - **Relax-solution-requirement** : spremeni lokalne zahteve za sprejem rešitve
- Agent mora ob vsakem času izbrati kateri operator bo uporabil
 - Sistematično preiskovanje po prostoru rešitev
 - Hevristično iskanje
 - Npr. uporabiti **Relax-solution-requirement**, ko je napredek premajhen
 - Katere zahteve omiliti,...

Porazdeljeno hevristično iskanje z omejitvami

- Problemi pri DCSP nastanejo tudi zaradi tekmovanja za vire
- Vsakemu viru je dodeljen en agent, in drugi agenti dobivajo rezultate preko tega agenta
- Dva pristopa:
 - Tržno usmerjeno programiranje (market-oriented programming)
 - z viri so asocirane dražbe, ki zagotavljajo ravnovesje med dodeljenimi viri
 - Viri izračunajo svoje skupne zahteve, ki jih nato upoštevajo agenti, ki tekmujejo za njih

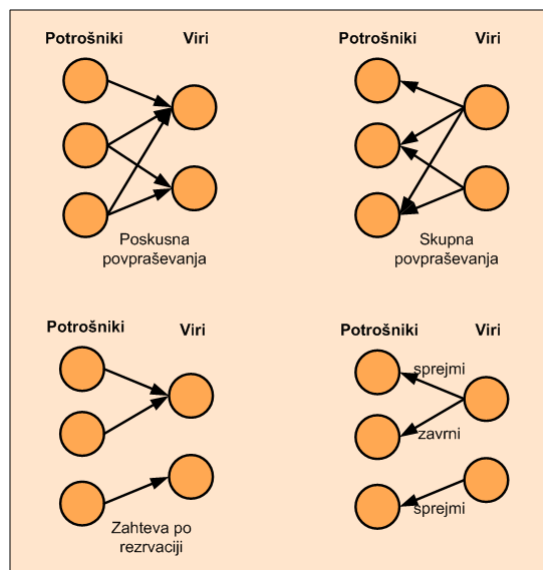
Porazdeljeno hevristično iskanje z omejitvami

- Distributed constrained heuristic search (DCHS):
 1. Agent začne z definiranjem stanja problema, z dekompozicijo problema (definira naloge, razmerja med njimi, vključno z omejitvami)
 2. Agent propagira omejitve v njegovem stanju problema
 1. Detektira nekonsistence
 2. Določi kakšni viri so potrebni in kdaj ter naredi profil povpraševanja za temi viri
 3. Sistem pošlj te profile virom (na začetku ali ob spremembah)
 4. Vir izračuna skupno povpraševanje in o tem obvesti agente
 5. Agent uporabi ta skupna povpraševanja za spremembo njegovih želja – izbere preferenčen vir-čas.interval-akcijo
 6. Agent zahteva od vira rezervacijo izbranega čas. Intervala
 7. Vir postopoma
 1. odobri rezervacijo, če je možno, in spremeni svoj urnik razpoložljivosti
 2. Zavrne zahtevo, če ta ni možna
 8. Agent obdela odgovor vira:
 1. Če je rezervacija sprejeta gre na korak 2 (popravi svoje stanje)
 2. Če ni sprejeta poskusi s ponovno rezervacijo (gre na korak 6)

Porazdeljene inteligentne programske tehnologije, Porazdeljeno reševanje problemov in načrtovanje

31

Porazdeljeno hevristično iskanje z omejitvami



Porazdeljene inteligentne programske tehnologije, Porazdeljeno reševanje problemov in načrtovanje

32

Organizacijske strukture

- Smiselno je zmanjšati komunikacijo, jo omejiti samo na potrebno
 - Komunicirajo naj agenti, za katere je ta komunikacija pomembna
 - Dekompozicijska struktura naj omogoča določanje takšnih kanalov (parov agentov)
 - Npr.,
 - V primeru nadzora prometa naj se vzpostavi komunikacija med agentoma, ki prekrivata sorodno območje, ki ga prečka avtomobil
- Organizacijska struktura
 - Definira, vloge, odgovornosti in preference agentov v združbi sodelujočih agentov
 - Definira nadzor in komunikacijski vzorec med njim
 - Za vsakega agenta določi naloge, ki jih lahko opravi, ter prioritetni vrstni red med njimi
 - Določa kateri agent naj komu kaj sporoči (delne rezultate)
 - Določa koliko naj agent upošteva dobljene delne rezultate

Organizacijske strukture

- Organizacijska struktura mora biti globalno koherentna
- Lokalno mora samo vsak agent zase skrbeti za predpisane naloge
 - Kaj naj kdaj naredi
 - Komu naj kaj pošlje
 - Od koga naj sprejme informacijo
 - Kako naj upošteva to informacijo
- Organizacijske strukture so ponavadi implementirane v obliki pravil vzorec-odgovor
- Dva pristopa pri načrtovanju organizacijske strukture:
 - Od spodaj navzgor: meje med vlogami agentov se lahko premikajo med samim delovanjem in reševanjem problema, da se poveča učinkovitost
 - Od zgoraj navzdol: prostor alternativnih struktur se razišče in evaluiira že pred implementacijo

Komunikacijske strategije

- Organizacijske strategije zagotavljajo statične informacije
- Upoštevati je potrebno tudi časovno komponento
 - Agent mora poslati informacijo drugemu agentu, ko ga ta zanima
- Dva ekstrema:
 - Pošiljati vse delne rezultate
 - prejemnik potem oceni, če jih potrebuje ali ne
 - Zahteva veliko komunikacije
 - Zapolni pomnilnik
 - Pošiljati samo (lokalno) popolne končne rezultate
 - Prejemnik ni obveščen o tekočem dogajanju
 - Prejemnik se že prej lahko zadovolji že z nedokončnimi rezultati
- Vmesne rešitve:
 - Pošlje se krajši delni rezultat zelo zgodaj

Komunikacijske strategije

- Rezultate lahko agenti pošiljajo drugim prostovoljno
 - Če je več potencialno zanimivih rezultatov kot dejansko proizvedenih rezultatov
 - Zakasnitve so manjše
 - Lahko povzroči odvečno komunikacijo
- Rezultate lahko pošiljajo agenti na zahtevo
 - Če je zgeneriranih rezultatov veliko več kot jih je dejansko potrebnih
 - Zakasnitve so enkrat daljše
- Zanesljivost komunikacije
 - Agent lahko zahteva potrditev sprejema
 - Dodatna obremenitev komunikacijskih poti
 - Agent poizkusi ugotoviti ali je prejemnik prejel njegovo sporočilo z opazovanjem njegovega odziva
 - Težje izvedljivo

Strukture nalog

- V opisu naloge nekega agenta lahko predvidimo učinke rezultatov te naloge
 - Rešitev je potrebna za rešitev druge naloge
 - Rešitev izboljša aktivnosti sprejemnika
- Takšna eksplicitna struktura nalog omogoča lažjo uporabo različnih komunikacijskih mehanizmov
 - Nek agent lahko da višje na svojo prioriteto nalogo, ki je zelo potrebna za nekega drugega agenta
 - Lahko mu pošlje rešitev, ko je ta ravno dovolj natančna
- Statične organizacijske strukture in strukture nalog imajo evidentne omejitve
 - Za bolj kompleksne probleme je morajo agent analizirati trenutno stanje in načrtovati načrte za interakcijo za vnaprej

Porazdeljeno načrtovanje

- Porazdeljeno načrtovanje: porazdeljeno reševanje problemov, kjer je problem narediti načrt
- Porazdeljeno načrtovanje:
 - Načrt je porazdeljen med več sistemov
 - Porazdeljeno načrtovanje sistema

	Centraliziran načrt	Porazdeljen načrt
Centralizirano načrtovanje		
Porazdeljeno načrtovanje		

Central. načrtovanje za porazdeljene načrte

- Deloma urejeni načrti (partial order plans)
- Centralizirani koordinator (načrtovalec za deloma urejena načrta) razbije načrt v vzporedne niti z sinhronizacijo med njimi:
 - Vhod: opis cilja, množica operatorjev, opis začetnega stanja -> Poišči delno urejeni načrt (s čim manj odvisnostmi)
 - Razstavi načrt v podnačrte
Urejene aktivnosti naj se ohranijo znotraj podnačrtov
 - Dodaj sinhronizacijske (komunikacijske) aktivnosti v podnačrte
 - Dodeli podnačrte posameznim agentom
 - V primeru neuspeha se vrni na prejšnje korake
 - Razstavi načrt na drugačen način
 - Generiraj drugačen deloma urejen načrt
 - V primeru uspeha
 - Dodaj še ostale pomembne povezave v podnačrte (imena agentov ipd.)
 - Inicializiraj aktivnosti za izvrševanje načrta
 - Spremljaj izvajanje (sestavi odzive agentov, itn.)

Central. načrtovanje za porazdeljene načrte

- Dodeljevanje nalog: razgradi-dodeli-izvrši-sestavi tehnika
- Cilj: Najti načrt, ki je najbolj razgradljiv in se da najbolj učinkovito porazdeliti
- Problem: razpoložljivost agentov ponavadi ni znana v naprej
- Komunikacija (komunikacijski kanal) igra zelo veliko vlogo:
 - Pri rahlo povezanih agentih (ali ozkih komunikacijskih analih): rahla dekompozicija, malo velikih podnalog
 - Pri krepko povezanih agentih (z deljenim pomnilnikom): krepka dekompozicija, visoka paralelizacija

Porazdeljeno načrtovanje za centraliz. načrte

- Sestavljanje zahtevnih kompleksnih načrtov v skupini sodelujočih načrtovalcev specialistov
- Proces načrtovanja porazdeljen, vsak specialist zgradi del načrta
- Delitev (skupna raba) nalog:
 - Celoten načrt se razstavi in porazdeli specialistom
 - Vsak specialist zgradi svoj del načrta
 - Načrt se postopoma izpolnjuje in posreduje naprej
- Delitev (skupna raba) rezultatov:
 - Načrtovalni agentje generirajo vzporedno delne načrte
 - Nato si izmenjujejo in sestavljajo te podnačrte v enovit načrt

Porazdeljeno načrtovanje za porazd. načrte

- Porazdeljeni so tako načrtovalni proces kot tudi proizvedeni načrt
- Ni potrebno imeti sestavljenega celotnega načrta skupaj kjerkoli v sistemu
- Porazdeljeni deli načrta morajo biti kompatibilni
 - Ne sme biti konfliktov med njimi
 - Morajo si eden drugemu pomagati
- Različni pristopi
 - Sestavljanje načrtov
 - Iterativno izdelovanje načrtov
 - Izdelovanje načrtov s pogajanji

Sestavljanje načrtov

- Vsak agent sestavi svoj individualni načrt, nato morajo zagotoviti, da se lahko vsi načrti skupaj izvedejo brez konfliktov
- Glavni cilj/problem: identifikacija in reševanje konfliktov
- Vsak agent zgradi popolnoma urejen načrt
- Poišče se pare akcij med agenti, ki so komutativne (se lahko izvedejo v obeh vrstnih redih)
- Takšne akcije s opusti iz analize -> načrti agentov se bistveno skrajšajo
- Konfliktne situacije in njihovo reševanje se išče samo na preostalih akcijah (ki jih je lahko zelo malo)
 - Preišče se vse možne kombinacije akcij
- Pri nevarnih interakcijah med akcijami
 - Se dodajo sinhronizacijske akcije (ki ustavijo izvajanje določenih akcij za čas, ko so v konfliktu z drugimi)

Sestavljanje načrtov

- Načrti so lahko sinhronizirani
 - Glede na akcije, na iztek določenih dogodkov (preko komunikacije)
 - Glede na čas, na določene roke
- Časovni roki za izdelavo posameznih delov
 - V okviru teh rokov so lahko načrti precej neodvisni
 - Koordinacija med načrti mora torej najti ustrezne roke

Iterativno izdelovanje načrtov

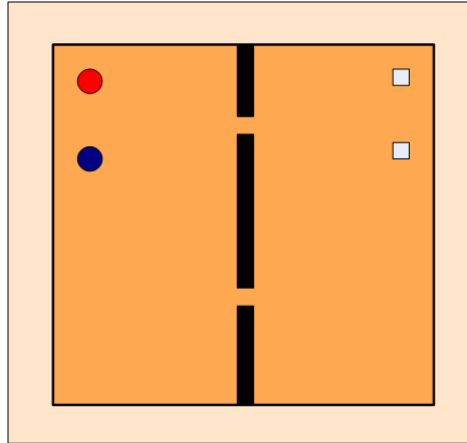
- Problemi pri sestavljanju načrtov nastanejo zaradi odvisnosti med akcijami
=> pri načrtovanju lokalnega načrta je dobro upoštevati tudi globalne omejitve, omejitve drugih agentov, odvisnosti med posameznimi akcijami
- Vsak posamezen agent naj ne proizvede samo enega načrta (ki se potem kombinirajo ampak vse možne načrte, ki vodijo do izpolnitve njegovega cilja
- Porazdeljen načrtovalski proces nato poišče med temi množicami načrtov, takšno kombinacijo načrtov, ki sodijo skupaj

Iterativno izdelovanje načrtov

- Iskanje kombinacij načrtov
 - Začni z množico vseh možnih načrtov
 - Postopoma zmanjšuj to množico in konvergiraj k optimalnemu načrtu
- Porazdeljeno hierarhično načrtovanje
 - Agenti hierarhične razdelajo načrte
 - Nato jih nadaljujejo razčlenovati v vedno bolj delajlne nivoje
 - Hierarhično iskanje po prostoru obnašanj (hierarchical behavior-space search)
 - Vsak agent predstavi svoje obnašanje na različnih nivojih abstrakcije
 - Če se lahko konflikt reši na višjih abstraktnih nivojih se reši
 - Krajše iskanje rešitve
 - Manj sporočanja
 - Možne so nepotrebne omejitve na nižjih nivojih
 - Dolgoročne, enostavne, lahko neučinkovite odločitve
 - Če ne, se posreduje naprej na nižji nivo in se poskuša rešiti tam
 - Manj nepotrebnih omejitev
 - Več komunikacije
 - Bolj odzivne, relno-časovne komunikacijske odločitve

Primer

- Primer hierarhičnega iskanja po prostoru obnašanj
- Porazdeljeno načrtovanje raznašanja



Pogajanja v porazdeljenem načrtovanju

- Ko se ugotovi konflikt med načrti – kateri izmed agentov bo spremenil svoj načrt?
- Možen pristop: pogajanja
- Primer: smeri poletov letal
 - letali, ki letita proti isti točki navedeta svoje alternativne smeri
 - Letalo, ki ima več alternativnih možnosti, spremeni smer

Predstavitve porazdeljenih načrtov

- Predstavitve načrtov, znanja, mnenj, akcij, pogojev, povpraševanj,...
- Agenti se morajo razumeti med seboj, govoriti morajo isti jezik
- Na višjem nivoju: razumeti namen komunikacije
- Na nižjem nivoju: razumeti vsebino komunikacije
- Pogosto odvisno od domene problema

Porazdeljeno načrtovanje in izvajanje

- Načrt se naredi z namenom, da bo tudi izveden
- Odnos med načrtovanjem in izvajanjem je zelo kompleksen (spremljanje izvajanja, ugotavljanje napak in nesreč, ponovno načrtovanje, ipd.)
- Ta odnos je v porazdeljenem načrtovanju še kompleksnejši
- Več strategij za kombiniranje koordinacije, načrtovanja in izvajanja

Koordinacija po načrtovanju

- Postopek združevanja načrtov:
 - Načrtuj
 - koordiniraj načrte
 - Izvedi
- Če med izvajanjem spodleti en načrt, spodleti načrt v celoti
- Načrtovanje z različnimi možnostmi (contingency planning)
 - Agent poleg osnovnega načrta naredi še alternativne načrte za različne možnosti do katerih lahko pride med izvajanjem (več vej)
 - Združiti in koordinirati je sedaj potrebno celotne načrte (vključno z vejami)
 - Označi možnosti s pogoji in koordiniraj samo načrte (dele načrtov) z neizključujočimi pogoji

Koordinacija po načrtovanju

- Spremljanje in ponovno načrtovanje
 - Vsak agent spremlja izvrševanje svojega načrta
 - Če pride do odstopanja od načrta se zaustavi izvajanje načrtov vseh agentov in se ponovi celoten cikel načrtovanja, koordiniranja in izvajanja načrtov
 - Če je to pogosto, je zelo drago
 - Popravljanje načrtov, uporaba ponovno uporabljivih načrtov
- Zaželeno je, da se lahko odstopanje od načrta popravi lokalno
 - Načrt se zgradi na visokem abstraktnem nivoju
 - Popravki se nato delajo na nižjih nivojih in ne globalno vplivajo na višje abstrakcijske nivoje

Koordinacija pred načrtovanjem

- Še preden agenti začnejo načrtovati se lahko dogovorijo (koordinirajo) dovoljenih aktivnostih, tako da med izvajanjem načrtov ne pride do konfliktov
- Socialna pravila
 - Socialno pravilo je prepoved določenih izbir akcij v določenih kontekstih
 - Določi se "prepovedana stanja", nato se ugotovi katere akcije (oz. kombinacije akcij) lahko privedejo do teh stanj in se jih prepove

Prepleteno načrtovanje, koordinacija in izvajanje

- Koordinacija
 - pred načrtovanje: agenti imajo popolne načrte, ki jih koordinirajo
 - po načrtovanju: načrtovanje vodijo skupna pravila, ki poskrbijo za koordinacijo
 - Vmesna možnost: načrtovanje in koordinacija sta prepletene
 - Načrtovanje in koordinacija se iterativno izvajata, nadzorujeta in popravljata

Delno globalno načrtovanje

- Partial Global Planning
- Načrtovanje in koordinacija sta prepletena
- Dekompozicija naloge
 - Naloge se da razgraditi na podnaloge
 - Agent na začetku ne pozna drugih agentov in ne ve kakšne naloge imajo oni
 - Cilj komunikacije je postopoma zgraditi to zavedanje in se uskladiti, da se lahko izvedejo zahtevane naloge
- Formulacija lokalnega načrta
 - Agent mora razumeti kakšne cilje naj bi dosegel in s kakšnimi akcijami
 - Lokalni načrti so pogosto nezanesljivi, z različnimi vejami in alternativnimi akcijami

Delno globalno načrtovanje

- Abstrakcija lokalnega načrta
 - Podrobnosti načrta kako se odzvati v različnih situacijah so pomembne za agenta, so pa lahko precej nepomembne s stališča komunikacije z drugimi
 - Agenti identificirajo čim večje korake v načrtu, ki bi lahko bili zanimivi za ostale
 - Ostale korake v komunikaciji navzven abstrahirajo
- Komunikacija
 - Agenti morajo komunicirati o svojih načrtih, da bi lahko zgradili skupne načrte
 - MLO (Meta-Lever Organization) določa informacije in pretok nadzora med agenti
 - Kdo mora poznati načrte določenega agenta
 - Kdo lahko določi nove načrt, ipd.

Delno globalno načrtovanje

- Določanje delnega globalnega cilja
 - Izmenjava lokalnih načrtov omogoča agentom, da lahko identificirajo kdaj se lahko določeni lokalni cilji smatrajo kot podcilji enega globalnega cilja
 - Tako so lahko znani samo deli globalnega cilja
 - Problem interpretacije: iz množice komponent (lokalnih ciljev) zgradi globalno interpretacijo (globalni cilj)
- Gradnja in popravljanje delnega globalnega načrta
 - Delni načrti se združujejo v delni globalni načrt (ki teži k izpolnitvi delnega globalnega cilja)
 - Analiza vzporednih aktivnosti omogoča izboljšave načrta:
 - Olajša se doseganje ciljev drugih agentov s tem, da se večjo prioriteto da nalogam, ki jih potrebujejo tudi drugi
 - Izogibati se je potrebno redundantnim nalogam
 - Temu ustrezno se spremeni vrstni red nalog

Delno globalno načrtovanje

- Načrtovanje komunikacije
 - Načrtuje se interakcije med agenti (komunikacija rezultatov nalog)
 - Glede na delni lokalni načrt, agent lahko ve kdo mu lahko kdaj priskrbi kakšne rezultate in ustrezno temu načrtuje komunikacijo s tem agentom
 - Če je potrebna sinteza rezultatov za to poskrbi en agent in sestavi delne rezultate v skupni rezultat
- Izvrševanje delnih globalnih načrtov
 - Glede na izgrajen delni globalni načrt (in preureditve akcij), se popravijo lokalni načrti
 - Popravijo se abstraktne predstavitve lokalnih načrtov
 - Te predstavitve se nato uporabijo za določanje konkretnih akcij, ki sledijo

Delno globalno načrtovanje

- Sprotno spreminjanje načrta
 - Aktivnosti agentov ali spremembe v okolju lahko zahtevajo spremembe načrtov
 - Če so te zahteve manjše, ne vplivajo na abstraktni nivo načrta, tako da stane koordinacija nespremenjena
 - Če so zahteve večje, je o tem potrebno obvestiti ostale agente
 - Kako ugotoviti kako pomembne so spremembe v lokalnih načrtih, da je potrebna komunikacija in ponovna koordinacija?
- Ponovno dodeljevanje nalog
 - Po začetni delitvi nalog so lahko nekateri agenti zelo obremenjeni, nekateri pa podobremenjeni
 - Pri izmenjavi abstraktnih načrtov nek agent lahko ve kako obremenjeni so drugi agenti
 - Agent lahko predlaga nov delni globalni načrt pri čemer predvidi, da bo del svojih nalog opravil drugi agent
 - Drugi agent lahko ta predlog sprejme ali predlaga drugačen delni načrt
 - Na ta način lahko pride do prerazporeditve nalog

Delno globalno načrtovanje

- Delno globalno načrtovanje je še posebej primerno v aplikacijah, ki tolerirajo določeno mero nekoordiniranih aktivnosti
- Lokalni načrti se neprestano osvežujejo (lahko niti nikoli nimamo množice stabilnih lokalnih načrtov)
- Dinamično osveževanje načrtov na učinkovit način v nepredvidljivem svetu
- (ne pa optimizacija načrtov za statična in predvidljiva okolja)

Sprotna koordinacija načrtov brez komunikacije

- Včasih agenti ne morejo (ali ne smejo) komunicirati informacije svojih načrtih
- Koordinacija brez eksplicitne komunikacije:
 - Agentje razpoznajo načrte drugih z opazovanjem njihovega delovanja
 - Agentje razpoznavajo namene drugih agentov pri čemer se naslanjajo na njihovo racionalnost ali na njihov pogled

Porazdeljene inteligentne programske tehnologije

Porazdeljeno odločanje

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Poglavlje 5

Porazdeljene inteligentne programske tehnologije

Porazdeljeno odločanje

- Agenti delajo skupaj a imajo različne cilje
- Vsak agent skuša maksimizirati svoje koristi (ne glede na skupno dobro)
- Vsi agenti upoštevajo skupni interakcijski protokol, a vsak ima svojo strategijo delovanja
- "Sebični" agentje
- Protokoli morajo upoštevati nekooperativnost
- Protokol zagotavlja, da je za vsakega agenta najboljša njegova zelena lokalna strategija – zato jo uporablja
- Agenti so si tekmeči, tekmujejo med seboj
- Različne agente lahko naredijo različni razvijalci

Porazdeljene inteligentne programske tehnologije, Porazdeljeno odločanje

2

Evaluacijski kriteriji

- Protokoli (mehanizmi) za pogajanja (dogovarjanja) se lahko ocenijo glede na različne kriterije:
 - Socialna blaginja (Social welfare)
 - Pareto učinkovitost (Pareto efficiency)
 - Individualna racionalnost (Individual rationality)
 - Stabilnost (Stability)
 - Računska učinkovitost (Computational Efficiency)
 - Porazdeljenost in komunikacijska učinkovitost (Distribution and Communication Efficiency)

Evaluacijski kriteriji

- Socialna blaginja (social welfare)
 - Vsota koristi vseh agentov v dani rešitvi
 - Meri globalno skupno dobro agentov
 - Primerja se alternativne mehanizme s primerjanjem rešitev do katerih vodijo ti mehanizmi
- Pareto učinkovitost (Pareto efficiency)
 - Globalni kriterij
 - Rešitev x je Pareto učinkovit, če ne obstaja nobena rešitev x' v kateri je vsaj en agent na boljšem kot v x in noben agent ni v x' na slabšem kot v x .
- Individualna racionalnost (Individual rationality)
 - Sodelovanje v pogajanjih je za agenta individualno racionalno, če izpogajana rešitev ni slabša kot, če agent ne bi sodeloval v pogajanjih
 - Mehanizem je individualno racionalen, če je sodelovanje v pogajanjih individualno racionalno za vse agente

Evaluacijski kriteriji

- Stabilnost (Stability)
 - Mehanizem mora biti stabilen, mora motivirati (sebične) agente, da se obnašajo kot je treba (in ne kot bi njim najbolj odgovarjalo)
 - *Dominantna strategija* zagotavlja najboljše rezultate, ne glede na strategije drugih agentov (včasih jo lahko postavimo)
 - *Nashovo ravnotežje* upošteva strategije drugih – vsak agent izbere strategijo, ki je najboljši odgovor na strategije drugih agentov
- Računska učinkovitost (Computational Efficiency)
 - Zaželjena je velika računska učinkovitost
- Porazdeljenost in komunikacijska učinkovitost (Distribution and Communication Efficiency)
 - Zaželjena je porazdeljenost in čim manj potrebne komunikacije

Glasovanje

- Vsi agenti glasujejo (prispevajo svoj delež na vhodu v mehanizem)
- Mehanizem nato izbere na osnovi vseh glasov rešitev
- Vsi agenti nato upoštevajo to rešitev
- Glasovanje naj bi na fer način pretvorilo volje posameznih agentov v skupno voljo
- A – množica agentov
- O – množica možnih rešitev
- \succ_i – preferenčna relacija med dvema možnostma za agenta i
- \succ^* - socialna (skupna) preferenčna relacija
- Mehanizem (glasovanje) pretvori preferenčne relacije agentov $(\succ_1, \succ_2, \dots, \succ_{|A|})$ v skupno socialno preferenčno relacijo \prec^*

Arrow teorem

- Šest želenih lastnosti za skupno izbiro:
 1. Socialna preferenčna urejenost \succ^* naj obstaja za vse možne vhode (individualne preference)
 2. \succ^* naj bo definirana za vsak par izbir o, o' iz O
 3. \succ^* naj bo asimetrična in tranzitivna po O
 4. Rezultat naj bo Pareto učinkovit:
če za $\forall i \in A$, $o \succ_i o'$, potem $o \succ^* o'$.
 5. Shema naj bo neodvisna od nerelevantnih sprememb.
Nepomembne spremembe (nove možnosti) naj ne vplivajo na izid glasovanja.
 6. Noben agent ne sme biti diktator in s silo uveljavljati svoje preference drugim agentom.
- Arrow teorem: Nobeno pravilo ne more ugoditi vsem šestim željam.

Množinski protokol

- Plurality protocol
- "navadna večina"
- Vsak agent glasuje za eno izbiro, zmaga tista, ki je dobila največ glasov
- Rezultat je zelo odvisen od irelevantnih alternativ

Kandidat	1. krog		2. krog	
	Glasov	%	Glasov	%
Lojze Peterle (neodvisen, podpora NSi, SDS in SLS)	283,412	28.73	312,012	31.74
Danilo Türk (neodvisen, podpora SD, DeSUS in AS)	241,349	24.47	671,018	68.26
Mitja Gaspari (neodvisen, podpora LDS)	237,632	24.09		
Zmago Jelinčič Plemeniti (SNS)	188,951	19.16		
Darko Krajnc (Stranka mladih Slovenije)	21,526	2.18		
Elena Pečarič (neodvisna, podpora stranke Akacije)	8,830	0.89		
Monika Piberl (Glas žensk)	4,729	0.48		

Množinski protokol

- Primer z ameriških volitev 2000:

X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇
Bush	Bush	Bush	Gore	Gore	Nader	Buchanan
Buchanan	Nader	Nader	Bush	Nader	Gore	Nader
Nader	Gore	Buchanan	Buchanan	Bush	Bush	Gore
Gore	Buchanan	Gore	Nader	Buchanan	Buchanan	Bush

Rezultati:

- Bush: 3
- Gore: 2
- Nader: 1
- Buchanan: 1

Za drugo mesto:

- Nader: 3
- Gore: 2
- Buchanan: 2

Binarni protokol

- "turnirski sistem"
- Alternative se obravnavajo v parih, zmagovalec gre naprej
- Rezultat je odvisen od irelevantnih alternativ
- Rezultat je tudi odvisen od agende – v kakšnem vrstnem redu se vršijo primerjave
- Lahko vodi do Pareto neučinkovit rešitve

Binarni protokol

X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇
Bush	Bush	Bush	Gore	Gore	Nader	Buchanan
Buchanan	Nader	Nader	Bush	Nader	Gore	Nader
Nader	Gore	Buchanan	Buchanan	Bush	Bush	Gore
Gore	Buchanan	Gore	Nader	Buchanan	Buchanan	Bush

Vrstni red 1:

- Bush > Nader: 4/7
- Bush > Buchanan: 6/7
- Gore > Bush: 4/7
- zmagovalac: Gore

Vrstni red 2:

- Nader > Gore: 5/7
- Nader > Buchanan: 4/7
- Bush > Nader: 4/7
- zmagovalac: Bush

Borda protokol

- Vsak agent razvrsti alternative in jim glede na to pripiše določeno število točk (od 1 do $|O|$).
- Vse točke vseh agentov se nato seštejejo.
- Zmaga alternativa z največjim številom točk

- Hitro, ker ni potrebno preverjaje po parih
- Zelo odvisno od irelevantnih alternativ

Borda protokol

	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇
4	Bush	Bush	Bush	Gore	Gore	Nader	Buchanan
3	Buchanan	Nader	Nader	Bush	Nader	Gore	Nader
2	Nader	Gore	Buchanan	Buchanan	Bush	Bush	Gore
1	Gore	Buchanan	Gore	Nader	Buchanan	Buchanan	Bush

Rezultati:

- Bush: 20
- Gore: 17
- Nader: 19
- Buchanan: 14
- Rezultat: Bush > Nader > Gore > Buchanan

Borda protokol

- Brez najslabše alternative (Buchanan)

	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇
3	Bush	Bush	Bush	Gore	Gore	Nader	Nader
2	Nader	Nader	Nader	Bush	Nader	Gore	Gore
1	Gore	Gore	Gore	Nader	Bush	Bush	Bush

Rezultati:

- Bush: 14
- Gore: 13
- Nader: 15
- Rezultat: Nader > Bush > Gore

Diktatorski protokol

- Vedno se upošteva želja enega
- Enostaven, ni fer

X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇
Bush	Bush	Bush	Gore	Gore	Nader	Buchanan
Buchanan	Nader	Nader	Bush	Nader	Gore	Nader
Nader	Gore	Buchanan	Buchanan	Bush	Bush	Gore
Gore	Buchanan	Gore	Nader	Buchanan	Buchanan	Bush

Vedno upoštevaj x1!

- Rezultat: Bush > Buchanan > Nader > Gore

Izberi agenta naključno:

- Rezultat: x7 (Buchanan > Nader > Gore > Bush)

Dražbe

- Sporazum med dvema agentoma: dražbarjem in ponudnikom
- Dražbar hoče maksimizirati svoj dobiček
- Cilj ni skupno dobro

- Dražbar hoče prodati predmet po najvišji ceni, ponudniki pa bi ga kupili po čim nižji ceni
- Dražbar hoče oddati naloge po čim nižji ceni, ponudniki pa bi jih opravili po čim večji ceni

Različna ozadja

- Različna ozadja dražb:
 - Dražba z osebnimi vrednostmi
 - Private value auctions
 - Ocena vrednosti blaga temelji zgolj na agentovih lastnih preferencah
 - Najboljši ponudnik ne bo preprodajal kupljenega blaga
 - Ponudnik ponavadi dobro pozna prav ceno blaga
 - Dražba s skupnimi vrednostmi
 - Common value auctions
 - Ocena vrednosti blaga temelji zgolj na ocenah vrednosti drugih agentov
 - Zaradi simetričnosti relacije, so te ocene enake oceni agenta
 - Agent kupi blago za preprodajo
 - Dražba s koreliranimi vrednostmi
 - Correlated value auctions
 - Ocena vrednosti blaga deloma temelji na agentovih lastnih preferencah, deloma pa na ocenah vrednosti drugih agentov
 - Agent lahko sam opravi nalogo ali pa jo da opraviti drugemu agentu

Dražbeni protokoli

- Različni dražbeni protokoli:
 - Angleški protokol
 - Zbiranje ponudb
 - Nizozemski protokol
 - Vickrey protokol

Angleška dražba

- English auction
- First-price open-cry auction
- Vsak ponudnik lahko viša svojo ponudbo
- Ko nobeden ponudnik ne več poviša zadnje ponudbe, najboljši ponudnik dobi blago
- Včasih dražbar enakomerno povišuje ponudbo
- Včasih morajo agenti najaviti svoj odstop od dražbe
- V angleških dražbah z osebnimi vrednostmi je agentova dominantna strategija vedno malce višati najvišjo ponudbo dokler le-ta ne doseže osebne vrednosti

Zbiranje ponudb

- First-price sealed-bid auction
- Vsak ponudnik pošlje eno ponudbo
- Ne ve kakšne ponudbe so/bodo poslali ostali ponudniki
- Ponudnik z najvišjo ponudbo zmagaja
- Ponudnik si želi ponuditi čim nižji ponudbo (ki je pod njegovo oceno vrednosti), ki bi pa še zmagala

Nizozemska dražba

- Duch (descending) auction
- Dražbar začne z visoko ceno za blago in jo neprestano znižuje, dokler eden izmed ponudnikov blaga ne kupi
- Strateško je ekvivalentna zbiranju ponudb

Vickrey dražba

- Vickrey auction
- Second-price sealed-bid auction
- Vsak ponudnik pošlje eno ponudbo
- Ne ve kakšne ponudbe so/bodo poslali ostali ponudniki
- Ponudnik z najvišjo ponudbo zmaga
- Plača pa ceno enako drugi najvišji ponudbi
- Dominantna strategija ponudnika je ponuditi njegovo oceno
 - Ponudnik lahko odda ponudbo ne glede na to kdo so drugi ponudniki
 - Ponudniku ni potrebno prilagajati taktiko glede na obnašanje ostalih
- Pogosto uporabljan protokol v MAS

Ekvivalenca donosa

- Vsi štiri dražbeni protokoli proizvedejo enako višino pričakovanih dohodkov dražbarja pri dražbah z osebnimi vrednostmi, če so ponudniki nevtralni glede na tveganja.
- Če se ponudniki izogibajo tveganju, Zbiranje ponudb in Nizozemska dražba dajejo višji pričakovani dohodek dražbarju
- Če se dražbar izogiba tveganju, višji pričakovani dobiček dobi z Angleško in Vickrey dražbo
- Večina dražb ni povsem tipa z osebnimi vrednostmi
 - Angleška dražba ponavadi zagotovi večji prihodek dražbarju kot Vickrey dražba
 - Angleška in Vickrey dražba ponavadi zagotovita večji prihodek dražbarju kot Zbiranje ponudb in Nizozemska dražba

Tajni dogovor

- Problem z dražbami je, da lahko tajni dogovori porušijo koncept dražbe
 - Ponudniki se lahko med seboj dogovorijo za nižje cene
- Bolj varni za dražarja sta Zbiranje ponudb in Nizozemska dražba
 - Vsakdo se boji, da bo kdo drugi prelomil obljubo
- Bolj varni za goljufa sta Vickrey dražba in Angleška dražba
 - Pri Vickrey ponudi svojo ocenjeno vrednost
 - Pri Angleški dražbi lahko spremlja ostale ponudbe

Lažnjivi dražbar

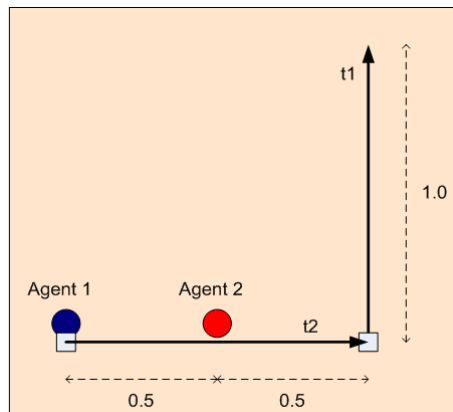
- Pri Angleški dražbi, zbiranju ponudb in Nizozemski dražbi ponudnik vedno plača v višini svoje ponudbe
- Pri Vickrey dražbi plača drugo ponudbo in ne svoje – dražbar se lahko o tem zlaže
- Dražbar ima lahko pomagače, da umetno napihujejo ceno pri Angleškem protokolu
- Lahko tudi sam odda ponudbo v višini pričakovane vrednosti
 - Pri Vickrey dražbi lahko še vedno izgubi
 - Bi dejansko moral postaviti višjo ceno

Prekletstvo zmagovalca

- Pri dražbah s skupnimi in koreliranimi vrednostmi, velja pravilo Prekletstvo zmagovalca
 - Ko agent zmagaja, ve da je plačal več kot ostali
 - => drugi cenijo izdelek manj
 - => izdelek je manj vreden
 - => ga je preplačal
- => Agent bi moral ponuditi manj, kot ocenjuje, da je vredno

Dražba z več povezanimi posli

- Na dražbi je več različnih predmetov
- Vsak predmet se prodaja posebej
- Agentove ocene teh predmetov so med seboj povezane
- Primer:



Nashovo ravnotežje

- Množica strategij je v Nashevem ravnotežju, če nobeden izmed agentov ne more izboljšati rezultata samo s spremembo svoje strategije (pri čemer ostanejo strategije drugih agentov nespremenjene)



John Forbes Nash



Primer

- Dva igralca hkrati izbereta število od 0 do 4
- Oba dobita število točk enako manjšemu od obeh števil
- Če en agent izbere večje število kot drugi, mu mora dati dve točki

	Player 2 chooses '0'	Player 2 chooses '1'	Player 2 chooses '2'	Player 2 chooses '3'
Player 1 chooses '0'	0, 0	2, -2	2, -2	2, -2
Player 1 chooses '1'	-2, 2	1, 1	3, -1	3, -1
Player 1 chooses '2'	-2, 2	-1, 3	2, 2	4, 0
Player 1 chooses '3'	-2, 2	-1, 3	0, 4	3, 3

Nashevo ravnotežje je v točki (0,0)

Zapornikova dilema

- Dva zapornika, nimata možnosti komunikacije
- Vsakdo od njiju lahko molči ali izda pajdaša
- Ponujene različne kazni

	B molči	B izda
A molči	A: 6 mes. B: 6 mes.	A: 10 let B: prost
A izda	A: prost B: 10 let	A: 5 let B: 5 let

Dominantna strategija: izdaja

Nashevo ravnotežje: oba izdata

Max. skupno dobro: oba molčita

Pareto učinkovitost: oba molčita

Barantanje

- Agenti se lahko sporazumejo, vendar imajo različne interese o katerem sporazumu naj se sporazumejo
- Dve podpodročji teorije o barantanju
 - Aksiomatična
 - Strateška

Aksiomatično barantanje

- Namesto ravnotežja se išče rešitev, ki ustreza določenim aksiomom
 - Pogosto obstaja več rešitev za Nashevo ravnotežje
 - Dva agenta hočeta razdeliti en dolar
 - Karkoli agent 1 predlaga agent 2 sprejme (ker je več kot 0).
 - Nashevo ravnotežje velja za vsak r , $0 < r < 1$
 - Potrebne so dodatne omejitve
- => Nashova rešitev barantanja

Nasheva rešitev barantanja

- Aksiomi Nasheve rešitve barantanja
 - Invarianca: pomemben je samo vrstni red izidov (koristnosti), ne pa tudi dejanske vrednosti
 - Anonimnost (simetrija): če zamenjamo labele agentov, se izid ne spremeni
 - Neodvisnost od nerelevantnih alternativ
 - Pareto učinkovitost
- Edina rešitev, ki zadovolji zgornje štiri aksiome je tista, ki maksimizira $[u_1(o) - u_1(o_{\text{fallback}})][u_2(o) - u_2(o_{\text{fallback}})]$.
- Delitev dolarja
 - Nasheva rešitev barantanja je 0.5

Strateško barantanje

- Ni aksiomov
- Barantanje je modelirano kot igra in rešitev je osnovana na analizi, ki išče ravnovesje med strategijami igralcev
- Zaporedno barantanje – agenti izmenično ponujajo ponudbe po nekem vrstnem redu
- Primer delitve dolarja
 - Zadnji ponudnik bo dobil $1 - \epsilon$, drugi agent pa ϵ
 - Agent mora torej samo čakati do konca, pa dobi skoraj cel dolar

Strateško barantanje

- Uvedemo še zmanjševanje vrednosti dolarja (za δ na vsakem koraku)
 - Končno število korakov
 - Primer: $\delta = 0.9$:
 - Maksimalne sprejemljive ponudbe:

Round	1's share	2's share	Total value	Offerer
t-3	.819	.181	$.9^{t-4}$	2
t-2	.91	.09	$.9^{t-3}$	1
t-1	.9	.1	$.9^{t-2}$	2
t	1	0	$.9^{t-1}$	1

- 90% ob času t-1 pomeni isto kot 100% ob času t

Rubensteinova rešitev barantanja

- Neomejeno število korakov
- Zmanjševalna faktorja :
 - Za agenta 1: d_1
 - Za agenta 2: d_2
 - Maksimalne sprejemljive ponudbe:

Round	1's share	2's share	Offerer
t-2	$1 - d_2(1 - d_1P)$		1
t-1		$1 - d_1P$	2
t	P		1

- Rubensteinova rešitev barantanja:
 - Agent 1 dobi $(1-d_2)/(1-d_1d_2)$
 - Agent 2 dobi $1 - (1-d_2)/(1-d_1d_2)$
 - Sporazum je sklenjen v prvem koraku

Model s stroški barantanja

- Vrednost se ne zmanjšuje
- Uvede se stroške barantanja
 - Za agenta 1: c_1
 - Za agenta 2: c_2
- Če je $c_1 < c_2$ potem agent 1 dobi cel dolar
- Če je $c_1 > c_2$ potem
 - agent 1 dobi c_2
 - agent 2 dobi ostalo

Pogodbene mreže

- Problem dodeljevanja nalog med sebične agente
 - Množica nalog T ,
 - Nožica agentov A ,
 - Funkcija stroškov $c: C_i: 2^T \rightarrow \mathfrak{R} \cup \{\infty\}$
 - Začetna dodelitev nalog med agente

$$\langle T_1^{init}, \dots, T_{|A|}^{init} \rangle$$

kjer

$$\bigcup_{i \in A} T_i^{init} = T; T_i^{init} \cap T_j^{init} = \emptyset, i \neq j$$

Mejni stroški

- Pogodbe se sklepajo na osnovi izračunov mejnih stroškov
- Individualna racionalnost – pogodba je za agenta individualno racionalna, če se agentu bolj splača sprejeti pogodbo kot ne
- Agent pogodbo sprejme, če je plačan več kot so njegovi mejni stroški za izvedbo naloge T^{contract} :

$$MC^{\text{add}}(T^{\text{contract}} | T_q) = C_q(T^{\text{contract}} \cup T_q) - C_q(T_q)$$

- Agent dodeli nalogo T^{contract} drugemu agentu, če mu za to mora plačati manj kot privarčuje, če naloge ne naredi:

$$MC^{\text{remove}}(T^{\text{contract}} | T_r) = C_r(T_r) - C_r(T_r - T^{\text{contract}})$$

Dodeljevanje nalog

- Agenti tako predlagajo pogodbe eden drugemu
- Pri odločitvah upoštevajo te mejne stroške
- Svoje naloge lahko tako naprej dodeljujejo drugim
- Agenti ne poznajo funkcije stroškov drugih agentov
- Dodelitev nalog se lahko na ta način samo izboljšuje (skupno dobro se izboljšuje, skupni stroški se zmanjšujejo)
- Skupno dobro se povečuje monotono
- Pogajanja lahko prekinemo kadarkoli (anytime algoritem)

Globalno optimalna razdelitev nalog

- Želja po konvergenci h globalno optimalni razdelitvi nalog
- Štiri tipi pogodb:
 - Originalna pogodba (Original contract - O): ena naloga se prerazporedi od enega agenta k drugemu
 - najbolj pogosta
 - Se lahko ujame v lokalni optimum
 - Gručasta pogodba (Cluster contract - C): množica nalog se hkrati prerazporedi od enega agenta k drugemu
 - Menjalna pogodba (Swap contract - S): Par agentov si izmenja nalogi.
 - Večagentna pogodba (Multiagent contract - M): več kot dva agenta sta vpletena v izmenjavo nalog

Globalno optimalna razdelitev nalog

- Za vsakega izmed štirih tipov pogodb (O, C, S, M) obstajajo razdelitve nalog, kjer nobena individualno racionalna pogodba z ostalimi temi tipi pogodb ni možna, je pa možna s četrtem tipom.
- Obstajajo primeri razdelitve nalog, kjer nobeno zaporedje individualno racionalnih pogodb ne vodi od začetne razdelitve nalog do optimalne razdelitve nalog z uporabo O-, C-, S- in M-pogodb.
- Ne zadošča niti nobena podmnožica pogodb
- Samo en tip (ali več njih) ne zadošča za konvergenco h globalno optimalni razdelitvi nalog

OCSM pogodbe

- OCSM pogodba je definirana s parom $\langle T, \rho \rangle$ $|A| \times |A|$ matrik
 - $T_{i,j}$ je množica nalog, ki jih agent i da agentu j
 - $\rho_{i,j}$ je znesek, ki ga agent i plača agentu j .
- OCSM pogodbe omogočajo poljubno prerazporeditev nalog z eno pogodbo.
- Če sta $|A|$ in $|T|$ končni množici in pogodbeni protokol omogoča OCSM pogodbe, potem vedno obstaja zaporedje individualno racionalnih pogodb, ki pripelje v končnem številu korakov do globalno optimalne razdelitve nalog
- Če kakšna OCSM-pogodba ni dovoljena, vedno obstaja kakšen problem, kjer ne obstaja nobeno IR zaporedje, ki bi pripeljalo do globalno optimalne razdelitve nalog

Neiskreni agenti

- Agentje se ponavadi pošteno pogajajo glede na svoje (resnične) mejne stroške
- Agenti lahko tudi lažejo:
 - Sprejmejo individualno neracionalne pogodbe
 - Zavrnejo individualno racionalne pogodbe
- Agenti lažejo
 - O svoji vrednosti za neko nalogo (in čakajo, da bodo v prihodnosti pogoji boljši)
 - O nalogah, ki jih imajo
 - Skrivajo naloge, ki jih imajo
 - Prikažejo naloge, ki jih nimajo
 - Prikažejo naloge, ki jih nimajo, a jih lahko zgenerirajo

Formiranje koalicij

- Včasih bi se (sebičnim) agentom splačalo med seboj formirati koalicije
- Nashevo ravnotežje: Množica strategij je v Nashovem ravnotežju, če nobeden izmed agentov ne more izboljšati rezultata samo s spremembo svoje strategije (pri čemer ostanejo strategije drugih agentov nespremenjene)
 - Se ga da izigrati s koalicijami (zapornikova dilema)
- Močno Nashevo ravnotežje: Nobena podmnožica agentov (koalicija) ne obstaja, ki bi lahko skupno spremenila svojo strategijo na način, ki bi vsem članom koalicije povečal dobiček, pri čemer nečlani koalicije ne bi spremenili strategije
 - Premočna zahteva, pogostokrat ne obstaja
- Koalicijsko odporno Nashevo ravnotežje: Nobena podmnožica agentov (koalicija) ne obstaja, ki bi lahko naredila vzajemno koristno spremembo strategije, pri čemer je ta sprememba stabilna glede na isti kriterij.
 - Kakšen agent bi raje šel v koalicijo z kakšnim nečlanom koalicije
 - Tudi ne obstaja vedno

Igra karakteristične funkcije

- Characteristic function game (CFG)
- Vrednost koalicije S je podana s karakteristično funkcijo v_S .
- Vrednost te funkcije je odvisna samo od članov koalicije in ne neposredno nečlanov.
- Posredno pa je vseeno odvisna od nečlanov:
 - Negativni vplivi (externalities)
 - Skupni viri (ki so lahko zasedeni)
 - Nasprotujoči cilji (ki lahko koalicijo oddaljijo od cilja)
 - Pozitivni vplivi
 - Prekrivajoči se cilji (ki lahko koalicijo pomaknejo bližje k cilju)

Tri aktivnosti

- Formiranje koalicij v CFG vsebuje tri aktivnosti:
 1. Generiranje strukture koalicije
 - Razbitje množice agentov na vse možne koalicije – na množico neprerivajočih koalicij, ki vsebujejo vse agente – struktura koalicij (coalition structure CS)
 - Agenti v koaliciji med seboj koordinirajo aktivnosti, jih pa ne koordinirajo z agenti izven koalicije
 - Primer: trije agenti {1,2,3}
 - 7 možnih koalicij: {1}, {2}, {3}, {1,2}, {1,3}, {2,3} in {1,2,3}.
 - 5 možnih struktur koalicij: {{1}, {2}, {3}}, {{1}, {2,3}}, {{1,2}, {3}}, {{1,3}, {2}} in {{1, 2, 3}}.
 2. Reševanje optimizacijskega problema za vsako koalicijo
 - Cilj koalicije je maksimirati dobiček: čimveč denarja dobiti od zunaj (minus stroški virov)
 3. Delitev vrednosti generirane rešitve med agente
- Te aktivnosti so lahko prepletene in niso neodvisne

Korak 1: Generiranje strukture koalicije

- Superaditivnost: Superaditivne igre so igre pri katerih velja
$$V_{S \cup T} \geq V_S + V_T$$
za vse neprekrivajoče koalicije
$$S, T \subseteq A$$
- CSG v superaditivnih igrah je enostaven, ker lahko vsi agenti skupaj kar tvorijo eno veliko koalicijo in sodelujejo med sabo
- Upoštevati je treba tudi ceno formiranja koalicij, komunikacije, čas potreben za komunikacije, ipd.
- V ne-supearditivnih igrah CSG ni trivialen

Generiranje strukture koalicije

- Cilj je maksimizirati skupno dobro agentov tako, da se poišče strukturo koalicij

$$CS^* = \arg \max_{CS \in \text{partitionsof}A} V(CS)$$

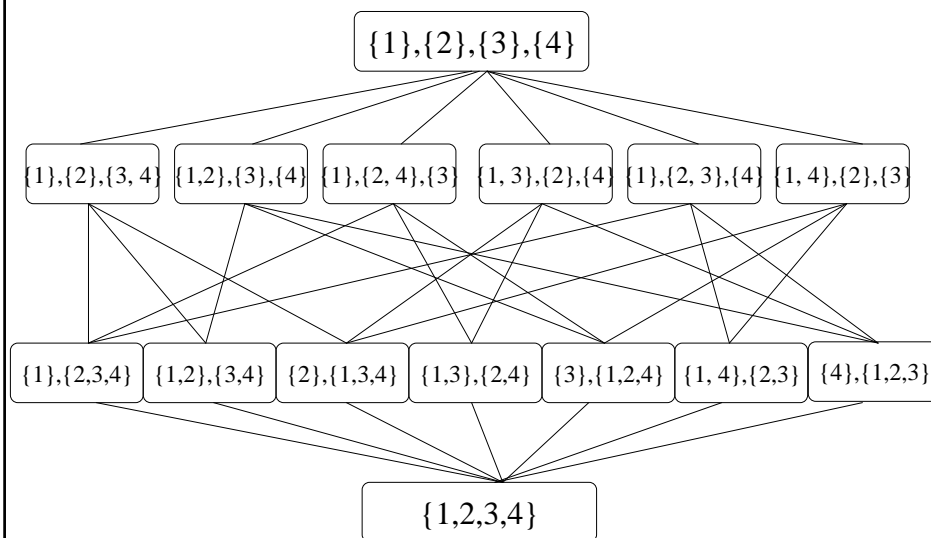
$$\text{kjer } V(CS) = \sum_{S \in CS} v_S$$

- Problem: število možnih struktur koalicij je ogromno $\Omega(|A|^{|A|/2})$
- Rešitev: išči po podmnožici N vseh možnih razdelitev A in izberi najboljšo koalicijo videno do takrat

$$CS_N^* = \arg \max_{CS \in N} V(CS)$$

- Generiranje strukture koalicije = iskanje po grafu struktur koalicij

Graf strukture koalicij



Iskanje po grafu struktur koalicij

- Želimo, da je najdena struktura koalicij v nekih mejah odstopanja od optimalne strukture

$$k \geq \frac{V(CS^*)}{V(CS_N^*)}$$

- Dovolj je preiskati spodnja dva nivoja strukture koalicij. Pri tem je $k=|A|$ in število preiskanih vozlišč je $n=2^{|A|-1}$
- Noben algoritem za generiranje struktur koalicij ne more doseči meje k ne da bi preiskal $n=2^{|A|-1}$ vozlišč.
- Mejo (worst case) se da doseči, ne da bi pregledali vse strukture koalicij.
- Če preiščemo več kot spodnja dva nivoja lahko dosežemo boljši rezultat od spodnje meje.

Korak 2: Reševanje optimizacijskega problema

- Reši se optimizacijski problem za vsako koalicijo posebej in tako definira njeno vrednost
- Včasih je reševanje optimizacijskega problema drago -> približna ocena rešitve
- Primerjati kvaliteto rešitve in ceno reševanja

Korak 3: Delitev dobička

- Na kakšen način razdeliti izplen koalicije med agente?
 - Mora biti fer
 - Mora motivirati agente, da ostanejo v koaliciji
- Delitev dobička glede na jedro
 - Jedro CFG-ja je množica konfiguracij razdelitve dobička, ki se naredi na tak način, da nobena podskupina agentov ni motivirana, da bi zapustila koalicijo in se pridružila drugi.
- Delitev dobička glede na Shapleyevo vrednost
 - Shapleyeva vrednost agenta i je mejni prispevek agenta i k strukturi koalicij, povprečen čez vse vrstne rede združevanj

Porazdeljene inteligentne programske tehnologije

Učenje v večagentnih sistemih

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Literatura: MAS, G. Weiss, Poglavje 6

Porazdeljene inteligentne programske tehnologije

Učenje

- Kaj je učenje?
 - inteligentni agenti so sposobni učenja
 - učeči se agenti so inteligentni
 - nadgrajevanje znanja
 - izboljševanje sposobnosti
 - povečevanje verjetnosti, da se bo v prihodnosti boljše odrezal
 - učenje na izkušnjah
 - prilagajanje na novo zaznane vhodne signale
 - posplošitev znanja

Porazdeljene inteligentne programske tehnologije, Učenje v večagentnih sistemih

2

Strojno učenje in MAS

- Strojno učenje (machine learning, ML)
 - uveljavljena disciplina
 - zelo veliko predlaganih metod za učenje
 - aplicirana na zelo veliko domenah
 - ponavadi centralizirano učenje
 - izoliran, samostojen proces
- Večagentni sistemi (MAS)
 - porazdeljeni sistemi
 - prepleteni procesi
 - ponavadi strukturna organizacija in funkcionalno obnašanje predvideni v naprej in fiksni
- Presek ML in MAS -> učenje v večagentnih sistemih!
 - nadgraditi MAS s sposobnostjo učenja
 - razširiti pogled na strojno učenje z večagentnim učenjem (lahko izboljša razumevanje splošnih principov učenja)

Zakaj učenje v MAS

- Razlogi za učenje
 - večagentni sistemi naj bi delovali v kompleksnem okolju
 - veliko, odprto, dinamično, nepredvidljivo okolje
 - nemogoče je v naprej predvideti vse mogoče situacije
 - kakšno bo okolje
 - koliko agentov bo prisotnih
 - kako se bodo agenti odzivali
 - kako se bo spreminjalo okolje...
 - nemogoče je v naprej specificirati sistem in programirati vsega potrebnega znanja
- => potrebno je omogočiti učenje – da agenti izboljšujejo svoje in skupne sposobnosti
- učenje v MAS je kvalitativno drugačno kot izolirano
 - vpliv drugih agentov na učenje (pomagajo, zavirajo, omogočajo učenje...)
 - sodelovanje, posredovanje znanja, komunikacija...

Karakterizacija učenja

- Učenje v MAS je večstranski problem
- različni pristopi, različna terminologija:
 - mutual learning
 - cooperative learning
 - collaborative learning
 - co-learning
 - team learning
 - social learning
 - shared learning
 - plularistic learning
 - organizational learning
 - interactive learning
 - active learning

Glavni kategoriji

- Dve glavni kategoriji
 - centralizirano (izolirano) učenje
 - decentralizirano (interaktivno) učenje
- Učni proces: vse aktivnosti (načrtovanje, sklepanje, odločanje, popravljanje predstavitev, ipd.), ki so izvedene s ciljem doseči določeni učni cilj (se nečesa naučiti).
- Centralizirano učenje:
 - celoten proces učenja je izveden s strani enega agenta
 - agent je pri tem popolnoma neodvisen od ostalih agentov
 - obnaša se, kot bi bil sam
- Decentralizirano učenje
 - več agentov je vpletenih v en učni proces
 - učne aktivnosti se izvajajo na več agentih
 - potrebnih je več agentov

Različni tipi učenja

- V večagentnem sistemu je lahko hkrati aktivnih več centraliziranih učečih se agentov
- lahko imajo različne ali isti cilj
- lahko je več agentov ali skupin agentov, ki so lahko vpleteni v decentralizirane učne procese
- lahko imajo različne ali isti cilj
- vsak agent je lahko vključen v več centraliziranih ali decentraliziranih učnih procesov

Značilnosti različnih tipov učenja

- Stopnja decentralizacije
 - porazdeljenost
 - en agenta
 - veliko agentov v učnem procesu
 - paralelizem
 - zaporedno izvajanje
 - popolnoma paralelizirano izvajanje

Značilnosti različnih tipov učenja

- Interakcija
 - stopnja interakcije
 - samo opazovanje
 - enostavno pošiljanje signalov
 - izmenjava informacij
 - kompleksni dialogi in pogajanja
 - vztrajnost interakcije
 - kratkoročna
 - dolgoročna
 - pogostost interakcije
 - redka
 - zelo pogosta
 - urejenost vzorca interakcije
 - popolnoma nestrukturiran
 - popolnoma hierarhičen
 - spremenljivost interakcije
 - fiksna
 - spremenljiva

Značilnosti različnih tipov učenja

- Vpletenost agenta
 - relevantna vpletenosti
 - vpletenost nekega agenta ni nujno potrebna
 - cilj se ne da doseči brez sodelovanja agenta
 - vloga igrana med vpletenostjo
 - splošen agent (agent lahko izvede vse učne aktivnosti)
 - specialist (agent je specializiran za določene učne aktivnosti)
- Cilji
 - tip izboljšave, ki se jo hoče doseči z učenjem
 - izboljšanje posameznega agenta
 - izboljšanje skupine agentov
 - kompatibilnost učnih ciljev, ki jih zasledujejo agenti
 - konfliktni
 - komplementarni učni cilji

Značilnosti različnih tipov učenja

- Učna metoda
 - rutinsko učenje
 - vgrajeno znanje, ni potrebno sklepanje pri učenju
 - učenje z navodili in nasveti
 - operacionalizacija – pretvorba podane informacije v obliko, ki jo uporablja učenec in integracija s predhodnim znanjem
 - učenje iz primerov in izkušenj
 - izločanje in izboljševanje znanja ter generalizacija le tega iz pozitivnih in negativnih primerov in izkušenj
 - učenje z analogijo
 - prenos znanja z rešenega na nov problem
 - učenje z raziskovanjem
 - zajemanje novega znanja z opazovanjem, izvajanjem eksperimentov, generiranjem in testiranjem hipotez)

Značilnosti različnih tipov učenja

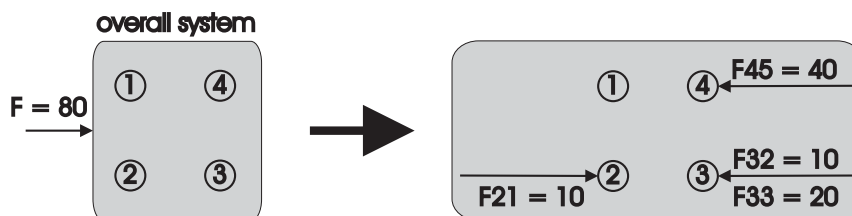
- Odziv pri učenju
 - nadzorovano učenje (supervised learning)
 - odziv je v naprej podan, učenje mora nadgraditi model tako, da se bo rezultat ujema z odzivom
 - okolje (ali drugi agent, človek) deluje kot učitelj
 - vzpodbujevalno učenje (reinforcement learning)
 - (spodbujevano učenje, učenje z ojačitvijo, ojačano učenje, vsiljeno učenje, učenje z vzpodbujanjem, vzpodbujevano učenje, vzpodbujevalno učenje)
 - za odziv učenec dobi podatek o koristi učne akcije
 - cilj je dolgoročno maksimizirati koristi
 - okolje deluje kot kritik
 - nenadzorovano učenje
 - učenec ne dobi nobenega odziva
 - sam mora ugotoviti koristi učnih akcij s poskušanjem ali samoorganizacijo
 - okolje deluje zgolj kot opazovalec

Problem dodelitve zaslug (CAP)

- Credit-Assignment Problem (CAP)
- Kako pravilno upoštevati odziv – dodeliti zasluge (pozitivni odziv) ali penale (negativni odziv) za skupno spremembo v učinkovitosti (pozitivno ali negativno)
- Kako vedeti katera aktivnost v učnem procesu je bolj prispevala k spremembi?
- V večagentnem sistemu je ta problem še bolj izrazit
- Dva podproblema:
 - med-agentni problem (inter-agent CAP)
 - določitev eksternih akcij agentov, ki jim gredo zasluge za celotno spremembo učinkovitosti
 - katera zunanja akcija katerega agenta je koliko prispevala k spremembi
 - znotraj-agentni problem (intra-agent CAP)
 - določitev internih akcij (sklepanj, odločitev) agenta, ki jim gredo zasluge za določeno eksterno akcijo
 - katero znanje, sklepanja in odločitve (notranje akcije) so vodile k neki zunanji akciji
- V praksi je zelo težko ločiti ta dva podproblema!

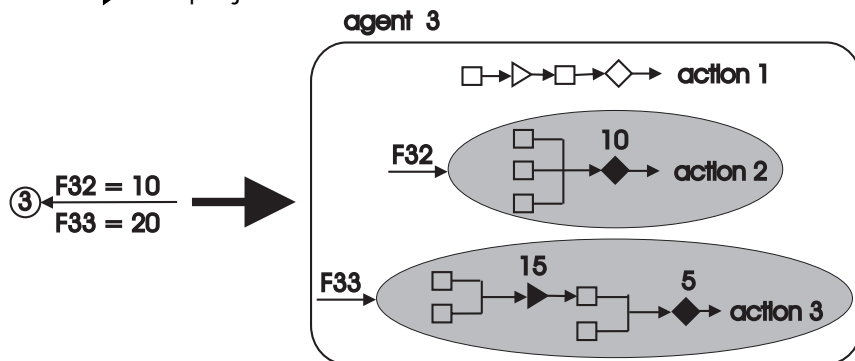
Med-agentni CAP

- Primer
 - štiriagentni sistem
 - izvedenih je bilo več akcij, s končnim odzivom $F=80$
 - Dekompozicija na osnovi akcij posameznih agentov:
 $F_{i,j}$ - i -ti agent in j -ta eksterna akcija



Znoraj-agentni CAP

- Primer za Agenta 3, ki je izvedel tri interne akcije
- Notranje akcije temeljijo na:
 - □ notranje znanje
 - ◇ odločitve
 - ▷ sklepanja



Učenje in koordinacija aktivnosti

- Kako naj se agenti naučijo primerno koordinirati njihove aktivnosti
- Agenti se lahko koordinirajo na v naprej določen način, s pravili, pogajanji, ipd.
- Lahko pa se tudi naučijo in prilagajajo novim situacijam
- Agenti se morajo naučiti prilagajati e ostalim agentom in koordinirati aktivnosti
- Dva primera:
 - Vzpodbujevalno učenje za koordinacijo v večagentnih sistemih, kjer se agenti ne zavedajo drug drugega
 - skupina sodelujočih učencev

Vzpodbujevalno učenje

- Reinforcement learning
- Agenti imajo opis trenutnega stanja in morajo izbrati naslednjo akcijo, ki naj bi maksimizirala odziv
- Okolje učenca se lahko modelira z Markovim procesom z diskretnim časom in končnim stanjem
- Okolje v MDP je predstavljeno s četverčkom $\langle S, A, P, r \rangle$
 - S je množica stanj
 - A je množica akcij
 - $P: S \times S \times A \mapsto [0,1]$
 - $r: S \times A \mapsto \mathfrak{R}$ skalarna funkcija nagrade (odziva)
- Vsak agent ohranja strategijo π , ki preslika trenutno stanje v željeno akcijo, ki naj bi bila izvedena nad tem stanjem
- Različni pristopi k vzpodbujevalnemu učenju

Q-učenje

- Poišči strategijo π^* , ki maksimizira vsoto vseh prihodnjih nagrad za vsa možna stanja $s \in S$
- Odločitvena funkcija je predstavljena s funkcijo (tabela)
 $Q: S \times A \mapsto \mathfrak{R}$
= dolgoročna nagrada za vsak par stanje-akcija
- Q-vrednosti se sproti spreminjajo

Učenje klasifikatorskih sistemov

- Klasifikatorski sistemi
 - temeljijo na pravilih
 - se učijo z uravnavanjem moči posameznih pravil
 - ob upoštevanju odziva okolja
 - in z odkrivanjem novih pravil z genetskimi algoritmi
- En klasifikator ima en pogoj in eno akcijo:
 - klasifikator i : (c_i, a_i)
 - moč klasifikatorja i ob času t : $S_t(c_i, a_i)$

BBA algoritem

- Cikel:
 - Vsem klasifikatorjem se na začetku priredijo naključne moči
 - na vsakem koraku se izbere eden izmed klasifikatorjev, ki vžge
 - na osnovi njegve akcije se dobi odziv iz okolja
 - moči klasifikatorjev se ustrezno popravijo
- Cikel se ponovi, niz takih ciklov tvori poizkus
- BBA algoritem (Bucket Brigade Algorithm) za dodeljevanje zaslug:
 - klasifikatorju, ki vžge se pripišejo skoraj vse zasluge
 - del zaslug se pripiše klasifikatorju, ki je vžgal pred njim
- Shema učenja zelo podobna Q-učenju

Izolirani, sočasni, vzpodbujevalni učenci

- Isolated, Concurrent Reinforcement Learners
- Pri vzpodbujevalnem učenju agenti tvorijo preslikave med zaznavami in akcijami in s primerno izbiro akcij skušajo maksimizirati odziv okolja
- Uporabijo se lahko tudi, ko imajo agentje malo predhodnega znanja in poznavanja sposobnosti in ciljev drugih agentov
- Naj agentje izberejo komunikacijo pri učenju koordinacije?
 - komunikacija pogosto olajša skupinske procese
 - ne garantira koordiniranega obnašanja
 - je časovno potratna
 - lahko odvrta od drugih problemov
 - naj bi se ne prenašale velike količine informacij
 - naj bi se agentje ne preveč zanašali na komunikacijo

Izolirani, sočasni, vzpodbujevalni učenci

- Izolirani, sočasni, vzpodbujevalni učenci
- vsak agent se uči zase, brez komunikacije z ostalimi
- upošteva samo odziv okolja
- ne modelira drugih agentov
- Ali je možno uporabiti enak učni mehanizem za kooperativna in ne-kooperativna okolja?

Karakteristike CIRL

- Običajna predpostavka, da dinamika okolja ni odvisna d drugih agentov, ne drži
- Karakteristike CIRL (concurrent, isolated, reinforcement learning):
 - povezovanje agentov (agent coupling)
 - tesno povezan sistem: akcije enega agenta pogosto in močno vplivajo na načrte drugih agentov
 - šibko povezan sistem: akcije enega agenta redko in šibko vplivajo a načrte drugih agentov
 - relacije med agenti
 - kooperativni agenti: agenti sodelujejo v skupini pri reševanju skupnega problema
 - nevtralni agenti: agenti komunicirajo, ker imajo skupne vire, nimajo pa posebnih odnosov
 - agenti tekmeči: agenti imajo nasprotujoče interese
 - čas odziva
 - odziv je na voljo takoj ali čez nekaj časa
 - kombinacije optimalnega obnašanja
 - koliko vrst obnašanja optimalno reši problem (od ena do neskončno pri različnih problemskih domenah)

CIRL eksperimenti

- Potiskanje kocke:
 - dva agenta se individualno naučita potiskati kocko po začrtani poti od starta do cilja
 - kooperativna situacija (oba agenta meta isti cilj)
 - kompetitivna situacija (vsak agent ima svoj cilj)
 - odziv temelji na razdalji kocke od začrtane poti
 - karakteristike:
 - sočasno delovanje dveh agentov
 - tesno povezan sistem
 - takojšen odziv
 - več optimalnih obnašanj

CIRL eksperimenti

- Souporaba virov:
 - nalogi sta podani
 - dva agenta se morata naučiti deliti vire skozi čas
 - karakteristike:
 - tesno povezan sistem
 - zapoznel odziv
 - eno samo optimalno obnašanje
- Navigacija robota
 - dva robota se naučita navigirati po sekajočih se poteh na mreži brez trčenja
 - karakteristike:
 - spremenljivo močno povezovanje
 - takojšen odziv
 - več optimalnih obnašanj

Zaključki CIRL eksperimentov

- Tako prijatelji kot nasprotniki se lahko naučijo koordinacije s CIRL
- Ni potrebno ne predhodno znanje, ne ekspliciten model drugih agentov
- Slabše se CIRL izkaže, ko
 - je sistem tesno povezan
 - odziv je zapoznel in
 - obstaja samo eno ali malo optimalnih obnašanj
- Ta problem bi lahko rešili z učenjem po fazah
 - nekaj časa bi se agent učil
 - potem bi izvrševal trenutno strategijo
 - potem bi se spet učil, itn.
 - dva agenta bi se lahko koordinirala, da se eden uči, ko se drugi ne
- V kooperativnih situacijah se agenti naučijo komplementarne strategije (se specializirajo)
- Agenti lahko prenesejo znanje na podobne situacije

Interaktivno vzpodbujevalno učenje koordinacije

- Agenti med seboj eksplicitno komunicirajo ob odločitvah o individualnih ali skupinskih akcijah
- Prilagojen BBA algoritem
 - agent lahko spremlja množico akcij, o katerih premišljajo drugi agenti
 - temu ustrezno lahko nekompatibilne akcije izloči iz svoje množice akcij
- Dva algoritma (ki izvirata iz BBA)
 - ACE (Action Estimation)
 - AGE (Action Group Estimation)
 - zahtevata različni nivo vpletenosti in koordinacije med vpletenimi agenti
- Agenti imajo isti cilj – sodelujejo, da bi optimizirali skupinski cilj

ACE – Action Estimation Algorithm

- Podana je zaznava S_i trenutnega stanja S (agenta a_i)
- vsak agent najprej izračuna množico akcij $A_i(S)$, ki jih lahko izvede v stanju S
- za vsako akcijo A_j izračuna relevantno $E_j(S)$ glede na cilj
- za vse akcije nad pragom agent izračuna in objavi ponudbo
- za izvajanje se izbere akcija z najvišjo ponudbo
- nekompatibilne akcije so izločene
- ta proces se ponavlja dokler niso vse akcije, za katere so bile poslana ponudba, bodisi izločene bodisi izbrane
- izbrane akcije tvorijo kontekst aktivnosti (activity context)
- BBA mehanizem nato zmanjša oceno izbrane akcije za vrednost ponudbe, ki se razporedi med akcije v prejšnjem kontekstu akcij
- če se po izvedbi akcij v trenutnem kontekstu aktivnosti dobi odziv, se le-ta enakomerno razporedi med vse akcije

AGE – Action Group Estimation Algorithm

- naprej se zberejo vse izvršljive akcije vseh agentov v danem stanju okolja
- nato se izračunajo množice kontekstov akcij, pri čemer je vak kontekst sestavljen iz množice med seboj kompatibilnih akcij
- za vsak kontekst aktivnosti se nato zbere ponudbe od vsakega agenta za vse njegove akcije
- izbere se kontekst aktivnosti, ki ima najvišjo vsoto ponudb
- izbrane akcije se izvedejo na dotičnih agentih
- spremenijo se ocene relevanc za akcije $E_j(S,A)$
- kupna vrednost ponudb se prerazporedi med akcije v prejšnjem kontekstu aktivnosti

Primerjava ICE in AGE algoritmov

- Oba algoritma omogočata učenje koordiniranega obnašanja
- agenti se naučijo precej bolj učinkovito izvrševati naloge kot pri naključni izbiri akcij
- AGE algoritem se nauči bolj učinkovite koordinacije
- AGE algoritem je računsko prostorsko in časovno bolj kompleksen
- algoritma nista dosegla globalno optimalnega obnašanja
 - zaradi omejenega zaznavanja
 - ker nista mogla razločiti nekaterih različnih globalnih stanj

Učenje o in od drugih agentov

- ne samo učenje koordinacije
- agenti se učijo izboljšati njihove individualne sposobnosti
- včasih gre to izboljšanje na račun drugih

- agenti naj se učijo o drugih, da bi bolje izkoristili priložnosti
- agenti se učijo, kako lahko na njihovo učenje vplivajo drugi agenti

- predvidevanje, napoved obnašanja drugih agentov
- interakcija z drugimi agenti
- skupni pogled na svet

Učenje o in od drugih agentov

- Tri pristopi:

- Učenje organizacijskih vlog
 - agentje v skupini se naučijo različnih komplementarnih vlog
 - na ta način izboljšajo skupno učinkovitost

- Učenje v tržnih pogojih
 - informacijski agenti, ki prodajajo in kupujejo na trgu
 - se prilagajajo trenutnim razmeram na trgu
 - gradijo modele drugih agentov

- Učenje boljše strategije za igro proti nasprotniku
 - namizne in podobne igre (med dvema igralcema)
 - zgrajen model nasprotnikove strategije pomaga izboljšati lastno strategijo

Učenje organizacijskih vlog

- Agenti se učijo prilagajanja na vloge, specifične za dano situacijo, v kooperativnih problemih
- vsak agent lahko igra eno ali več vlog v vsaki situaciji
- naloga je izbrati najboljšo vlogo (ki bo bolj pomagala pri reševanju problema)
- Uporaba ocen UPC (Utility, Probability and Cost) posamezne vloge v dani situaciji
 - korist, verjetnost, cena
 - vsa stanja sveta S so preslikana v manjšo množico situacij
 - korist (utility): ocena končne koristi v končnem stanju, če agent sprejme dano vlogo v dani situaciji
 - verjetnost (probability): verjetnost, da bo agent dosegel uspešno zadnje stanje, če bo sprejel dano vlogo
 - cena (cost): računski stroški, ki pri tem nastanejo
 - potencial (potential) vloge: uporabnost vloge pri odkrivanju primerne globalne informacije in omejitev

Učenje organizacijskih vlog

- Naj bodo S_k, R_k množice vektorjev situacij in vlog agenta k
- Agent hrani do $|S_k| \cdot |R_k|$ vektorjev vrednosti UPC in potencialov za različne vloge v različnih situacijah

- Med učno fazo je verjetnost za izbiro vloge r v situaciji s

$$\Pr(r) = \frac{f(U_{rs}, P_{rs}, C_{rs}, Potential_{rs})}{\sum_{j \in R_k} f(U_{js}, P_{js}, C_{js}, Potential_{js})}$$

kjer je f funkcija, ki se uporablja za ocenitev vloge

- Ko je učna faza mimo, se vloga izbere najboljše ocenjeno vlogo: $r = \arg \max_{j \in R_k} f(U_{js}, P_{js}, C_{js}, Potential_{js})$

Učenje organizacijskih vlog

- UPC vrednosti se naučijo z vzpodbujevalnim učenjem
- Naj bodo $\hat{U}_{rs}^n, \hat{P}_{rs}^n, \hat{Potential}_{rs}^n$ vrednosti po n korakih za vlogo r v situaciji s
- Naj bodo S situacije srečane med začetkom uporabe vloge r v situaciji s in končnim stanjem F (s končno koristjo U_F)
- Vrednosti za koristi (utility) se osvežujejo tako:
$$\hat{U}_{rs}^{n+1} \leftarrow (1-\alpha) \cdot \hat{U}_{rs}^n + \alpha \cdot U_F$$
- Osvežijo se vse vloge v vseh situacijah v S (na poti do F)
- Naj $O: S \rightarrow [0,1]$, pove ali je dano stanje uspešno ali ne
- Osvežitveno pravilo za verjetnost: $\hat{P}_{rs}^{n+1} \leftarrow (1-\alpha) \cdot \hat{P}_{rs}^n + \alpha \cdot O(F)$
- Naj $Conf(S)$ vrne 1, če obstajajo na poti do F konfliktne poti, ki se razrešijo
- Osvežitveno pravilo za potencial:
$$\hat{Potential}_{rs}^{n+1} \leftarrow (1-\alpha) \cdot \hat{Potential}_{rs}^n + \alpha \cdot Conf(S)$$

Učenje organizacijskih vlog

- Osvežitvena pravila za cene so odvisna od domene/problema
- Najtežji problem je kako oceniti funkcijo f
- Primer:
$$f(U, P, C, Potential) = U * P + Potential$$

Učenje v tržnih pogojih

- Agenti lahko kupujejo in prodajajo dobrine na elektronskem trgu
- trg je odprt za nove agente
- agenti so sebični in hočejo maksimizirati svoj dobiček, vsak igra zase
- tržni mehanizem uravnava trgovanje
- agenti lahko dostopajo enako do vseh ostalih agentov

- kupec objavi povpraševanje
- prodajalci objavijo ponudbe
- kupec izbere najbolj ugodno ponudbo in plača
- prodajalec izroči dobrino kupcu

- kupec hoče maksimizirati vrednost (odvisna od cene in kvalitete)
- prodajalec hoče maksimizirati dobiček (odvisen d cene in stroškov)

Tri tipi agentov

- Agenti 0. stopnje
 - ne modelirajo obnašanja drugih agentov
 - cene postavljajo na osnovi akumuliranih izkušenj
- Agenti 1. stopnje
 - Agenti analizirajo preteklo obnašanje ostalih agentov in skušajo predvideti njihove preference glede nakupov in prodaj
 - ostale agente modelirajo le kot agente 0. stopnje (ne upoštevajo, da lahko tudi ti agenti modelirajo druge agente)
- Agenti 2. stopnje
 - Agenti modelirajo druge agente kot agente 1. stopnje

Učenje za zmago nad nasprotnikom

- Namizne igre med dvema igralcema
- Minimax algoritem – minimiziraj maksimalno možno izgubo
- Predvidi, da bo nasprotnik naredil zate najslabšo potezo
- če nasprotnik lahko zmaga v naslednjem koraku, mu to prepreči
- M^* algoritem je posplošitev Minimax algoritma
 - uporabi model nasprotnika za izbiro naslednje poteze
 - M^0 ustreza minimax algoritmu (če je evaluacijska funkcija f_0 znana)
 - višji redi algoritma M iščejo rekurzivno bolj v globino
 - igralec M^n predpostavlja, da je nasprotnik igralec M^{n-1} itn.
 - igralec M^1 igra boljše kot igralec M^0 , če je model nasprotnika poznan
 - Vprašanje pa je kako dobiti natančen model nasprotnika!
 - Nauči se model nasprotnika!

Učenje in komunikacija

- Kako sta učenje in komunikacija povezana?
- zahteve za učinkovito izmenjavo informacij med agenti
- Dve glavni relaciji:
 - Učenje za komunikacijo
 - z učenjem se zmanjša količina informacije med agenti
 - komunikacija je počasna in draga, učenjem se jo lahko zmanjša
 - Komunikacija za učenje
 - s komunikacijo se omogoči izmenjava informacij, ki jih agenti potrebujejo pri učenju
 - učenje potrebuje informacije, komunikacija jih lahko priskrbi

Učenje in komunikacija

- Relevantna vprašanja:
 - Kaj komunicirati?
 - kakšna informacija je zanimiva za ostale
 - Kdaj komunicirati?
 - do kolikšne mere naj agent sam poskuša rešiti problem predno vpraša druge za pomoč
 - S kom komunicirati?
 - katere agente zanima naša komunikacija, kateri nam lahko pomagajo
 - Kako komunicirati?
 - na katerem nivoju
 - kateri jezik in protokol naj uporabljajo
 - neposredno ali preko table

Skupna ontologija

- Agentje morajo vedeti o čem se pogovarjajo kaj se na kaj nanaša
- Glavna naloga učenja v porazdeljenem sistemu je zagotoviti enak/skupni pomen simbolov
- Prizemljitev simbolov (symbol grounding problem)
 - prizemljitev (povezava) simbolov v realen svet
 - asociacije med visokonivojskimi simboli in nizkonivojskimi (realnimi) signali (ali entitetami izpeljanimi iz njih)
 - pogoj za realen inteligentni sistem?
- Problem skupnega pomena (shared meaning problem)
 - prizemljitev simbolov v MAS

Učenje za komunikacijo

- Zmanjševanje količine komunikacije z učenjem
- Pogodbene mreže
 - Manager objavi naloge
 - potencialni pogodbeniki pošljejo ponudbe
 - manager izbere najboljšo ponudbo, sklenitev pogodbe
- Pošiljanje ponudb vsem je lahko zelo potratno v velikih sistemih z veliko agenti in/ali veliko nalogami
- Rešitve:
 - usmerjeno naslavljanje
 - neposredno komuniciranje
 - včasih težko izvedljivo, naslovniki niso vedno v naprej poznani
- Elegantnejša rešitev:
 - učenje naslovnikov

Učenje naslovnikov

- Glavni cilj: zmanjšanje količine komunikacije z zajemanjem znanja o drugih agentih (oz. njihovih sposobnostih)
- Z naučenim znanjem je lahko naslavljanje bolj usmerjeno ali neposredno
- Sklepanje na osnovi primerov (Case-based reasoning):
 - problem se reši na osnovi rešitve podobnega primera v preteklosti
 - tvorjenje parov problem-rešitev
 - ko pride nov problem, se najprej pogleda, če že obstaja podoben rešen problem (iskanje problema)
 - če je problem popolnoma nov, se reši od začetka
 - če obstaja podoben problem, se njegova rešitev uporabi kot začetni približek in se ga nato reši
 - Vsi problemi se skupaj z rešitvami shranijo v bazi primerov (case storage)

Učenje naslovnikov

- Vsak agent hrani svojo bazo primerov
- Primer je sestavljen iz
 - specifikacije naloge
 - informacijo o tem kateri agent je že rešil to nalogo in kako dober je pri tem bil
- Specifikacija naloge:
 - atributi naloge
 - vrednosti atributov
- Mera podobnosti med nalogami
 - mera podobnosti med atributi in njihovimi vrednostmi
- Meri podobnosti med atributi in vrednostmi so odvisni od domene, recimo: 1, če sta enaki ali 0 sicer (identiteta)
- Podobnosti upragovimo, da dobimo množico podobnih nalog
- Odločitev komu poslati povpraševanje:
 - za vsakega agenta agent izračuna množico podobnih nalog
 - izračuna primernost agenta z obtežitvijo s preteklimi uspehi
- Agent nato obvesti samo najbolj primerne agente!

Komunikacija za izboljšanje učenja

- Agent ni vsemogočen niti vseveden (razen v zelo omejenih pogojih)
- Pomanjkanje informacije se lahko nanaša na
 - okolje, v katerem je umeščen in problem, ki naj ga reši
 - ostale agente
 - odvisnosti med različnimi aktivnostmi in učinek nekaterih agentovih aktivnosti na okolje in na aktivnosti drugih agentov
- Pomanjkanje informacij se lahko zmanjša s komunikacijo med agenti in se izboljša učenje
 - učenje, ki temelji na nizkonivojski komunikaciji (izmenjava manjkajočih delov informacije)
 - informacija v skupni rabi
 - učenje, ki temelji na visokonivojski komunikaciji (bolj kompleksna interakcija, pogajanja, medsebojno razlaganje in kombiniranje znanja)
 - skupno razumevanje

Komunikacija za izboljšanje učenja

- Komunikacija kot orodje za izboljševanje učenja
- Komunikacija je del učnega procesa (zajemanja znanja)
- Naj bo učenje obogateno s komunikacijo?
 - Kako hitro se učni rezultati dosežejo z ali brez komunikacije?
 - kakšna je kvaliteta rezultatov doseženih z ali brez komunikacije?
 - kako kompleksen je celotni učni proces z ali brez komunikacije?
- Komunikacija lahko zelo poenostavi učenje (zagotovi nove informacije)
- Komunikacija lahko tudi upočasni učenje (če pride do napake in se te napake uporabijo v učnem procesu/naučenih modelih)

Primer 1

- Lovimo skupaj!
- Domena lovec-plen
- kvadratna mreža, 4 možne poteze
- žrtve se gibljejo naključno
- lovci želijo ujeti čim več plena
- lovci imajo omejen vidi doseg
- strategija lovcev temelji na Q-učenju
- uspeh lahko izboljšajo s komunikacijo:
 - Komuniciranje senzorskih podatkov
 - podatki o opaženem plenu, o medsebojnih premikih
 - centraliziran nadzor porazdeljenih senzorjev
 - Komuniciranje odločitvenih strategij in odločitev o akcijah
 - sporočajo si odločitve/aktivnosti (recimo vrednosti $Q(ws,a)$)
 - centraliziran nadzor porazdeljenih efektorjev

Primer 2 – Učenje s konsenzom

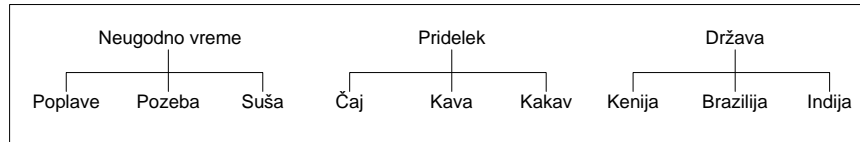
- Učenje z visokonivojsko komunikacijo
- Učenje s konsenzom
- Skupina agentov komunicira z uporabo table
- Uporabljajo enostaven jezik, ki vsebuje naslednjih 9 operatorjev za hipoteze:
 - dodajanje in odstranjevanje hipotez s table:
 - ASSERT(H)
 - PROPOSE(H,C)
 - WITHDRAW(H)
 - evaluacija hipotez:
 - CONFIRM(H,C)
 - DISAGREE(H,C)
 - NOOPINION(H)
 - MODIFY(H,G,C)
 - sprememba stanja hipotez in sprejem hipotez:
 - AGREED(H,T)
 - ACCEPT(H)

Učenje s konsenzom

- Agent predlaga hipotezo H (s PROPOSE)
- ostali agenti odgovorijo (s CONFIRM, DISAGREE, NOOPINION ali MODIFY)
- Prvi agent nato ugotovi skupno stopnjo zaupanja T v hipotezo H.
 - upošteva tako pozitivne odgovore (CONFIRM in MODIFY)
 - kot tudi negativne odgovore (DISAGREE)

Učenje s konsenzom - primer

- Koliko bo stala skodelica kave?
- Tri agenti, ki bi radi vedeli kako se bodo spreminjale cene kave, čaja in kakava
- Taksonomije znanja agentov:



Učenje s konsenzom - primer

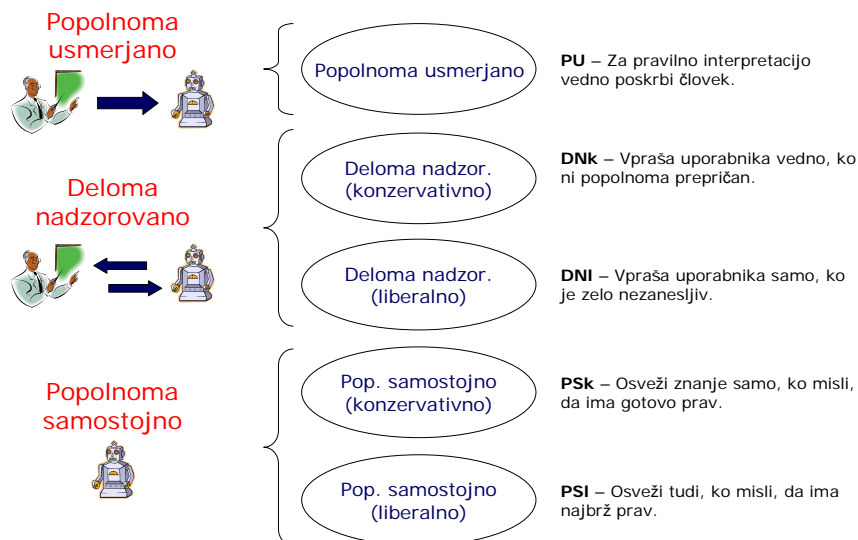
- Lokalno znanje:
 - *Agent1: Glavni-Proizvajalec(Kenija, Kava)*
Glavni-Proizvajalec(Kenija, Čaj)
 - *Agent2: Glavni-Proizvajalec(Brazilija, Kava)*
Glavni-Proizvajalec(Brazilija, Kakav)
 - *Agent3: Glavni-Proizvajalec(Indija, Čaj)*
- Zaznani podatki in generalizacije:
 - *Agent1: Vreme(Kenija, Suša), Cena(Čaj, Raste)*
Vreme(Kenija, Suša), Cena(Kakav, Stabilna)
Vreme(Kenija, Pozeba), Cena(Kava, Raste)
GEN: Vreme(Kenija, Neugodno) in
Glavni-Proizvajalec(Kenija, Pridelek) ->
Cena(Pridelek, Raste)
 - *Agent2: Vreme(Brazilija, Pozeba), Cena(Kava, Raste)*
Vreme(Brazilija, Poplave), Cena(Kakav, Raste)
GEN: Vreme(Brazilija, Neugodno) -> Cena(Pridelek, Raste)
 - *Agent3: Vreme(Indija, Poplave), Cena(Čaj, Raste)*
GEN: Vreme(Indija, Poplave) -> Cena(Čaj, Raste)

Učenje s konsenzom - primer

- Hipoteze:
 - H1= *Vreme(Indija,Poplave) -> Cena(Čaj,Raste)*
 - H2= *Vreme(Država,Neugodno) -> Cena(Pridelek,Raste)*
 - H3= *Vreme(Država,Neugodno) in Glavni-Proizvajalec(Država,Pridelek) -> Cena(Pridelek,Raste)*
- *Potek interakcije med agenti:*

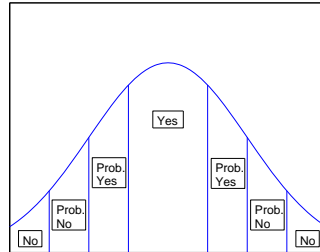
<i>Agent3: PROPOSE(H1,0.6)</i>	<i>Agent3: WITHDRAW(H1)</i>
<i>Agent2: MODIFY(H1,H2,0.5)</i>	<i>Agent2: WITHDRAW(H2)</i>
<i>Agent1: MODIFY(H1,H3,0.55)</i>	<i>Agent1: AGREED(H3,0.8)</i>
<i>H1: 0.775</i>	<i>Agent3: ACCEPT(H3)</i>
<i>Agent1: MODIFY(H2,H3,0.45)</i>	<i>Agent2: ACCEPT(H3)</i>
<i>Agent3: CONFIRM (H2,0.6)</i>	
<i>H2: 0.78</i>	
<i>Agent3: CONFIRM (H3,0.6)</i>	
<i>Agent2: CONFIRM (H3,0.5)</i>	
<i>H3: 0.8</i>	

Primer - Različni načini učenja



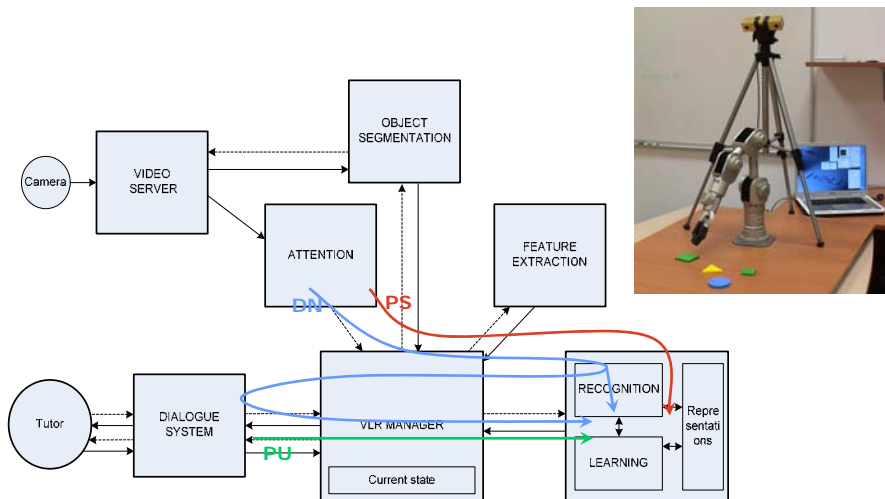
Pravila za osveževanje znanja

- Učni proces:
 - zaznaj predmet
 - segmentiraj predmet
 - poizkušaj razpoznati predmet
 - odloči se za naslednji korak
 - osveži
 - vprašaj
 - osveži predstavitev
- Tabela za osveževanje:



	Da	Najbrž da	Najbrž ne	Ne	Ne vem
PU	vprašaj	vprašaj	vprašaj	vprašaj	vprašaj
DNK	osveži	vprašaj	vprašaj	/	vprašaj
DNI	osveži	osveži	osveži	/	vprašaj
PSk	osveži	/	/	/	/
PSI	osveži	osveži	/	/	/

Implementiran sistem



Primer dialoga

H: This is a red, small, and square object.
 R: OK.
 H: What do you see?
 R: I don't know this object.
 H: It is green, small, and triangular.
 R: OK.
 H: What do you see?
 R: I see a small and square object.
 H: Yes, that's correct.
 R: OK.
 H: It is also blue.
 R: OK.
 H: What do you see?
 R: I see a green object. I think it is also large.
 H: Yes, that's correct.
 R: OK.
 H: It is also rectangular.
 R: OK.
 R: I don't know this object. What is it like?
 H: It is blue, small, and rectangular.
 R: OK.
 R: Is this object large?
 H: Yes, that's correct.
 R: OK.
 H: It is also yellow and circular.
 R: OK.
 R: Is this object red?
 H: No, it is not red.
 R: OK.
 H: It is yellow.
 R: OK.
 H: What do you see?
 R: I see a red, small, and triangular object.

Popolnoma
usmerjano

Deloma
nadzorovano

Eksperimenti

- Učenje in razpoznavanje vizualnih lastnosti
 - 6 vizualnih značilnic
 - 3 značilnice videza (Hu, Sa, In)
 - 3 značilnice oblike (Ar, Pr, Cm)
 - 10 vizualnih lastnosti
 - 4 barve (Rd, Gr, Bl, Yl)
 - 2 velikosti (Sm, Lr)
 - 4 oblike (Sq, Cr, Tr, Rc)
 - Pravilni opisi poznani – simulirani odgovori uporabnika
 - Evolucija uspešnosti razpoznavanja skozi čas

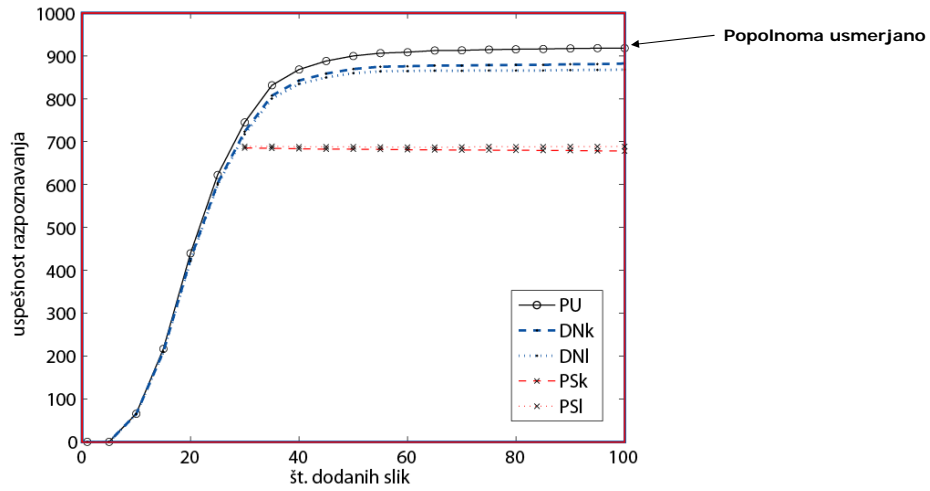


Tabela uspešnosti:

	YES	PY	PÑ	NO	DK
YES	1	0.5	-0.5	-1	0
NO	-1	-0.5	0.5	1	0

Eksp. rezultati – vizualne lastnosti

- Uspešnost razpoznavanja

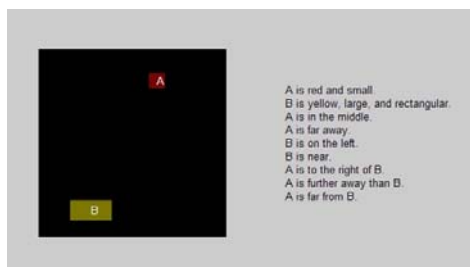


Učenje prostorskih relacij

- 2 objekta (A and B)
- 5 značilnic (x, y, dx, dy, d)
- 11 prostorskih relacij:
 - levo od (TL: to the left of: "A is to the left of B")
 - desno od (TR: to the right of: "A is to the right of B")
 - bližje kot (CT: closer than: "A is closer to me than B")
 - dlje kot (FT: further away than: "A is further away from me than B")
 - blizu do (NT: near to: "A is near to B")
 - daleč od (FF: far from: "A is far from B")
 - na levi (OL: on the left: "A is on the left")
 - na sredini (OR: on the right: "A is on the right")
 - na desni (IM: in the middle: "A is in the middle")
 - blizu (NR: near: "A is near")
 - daleč (FA: far away: "A is far away")

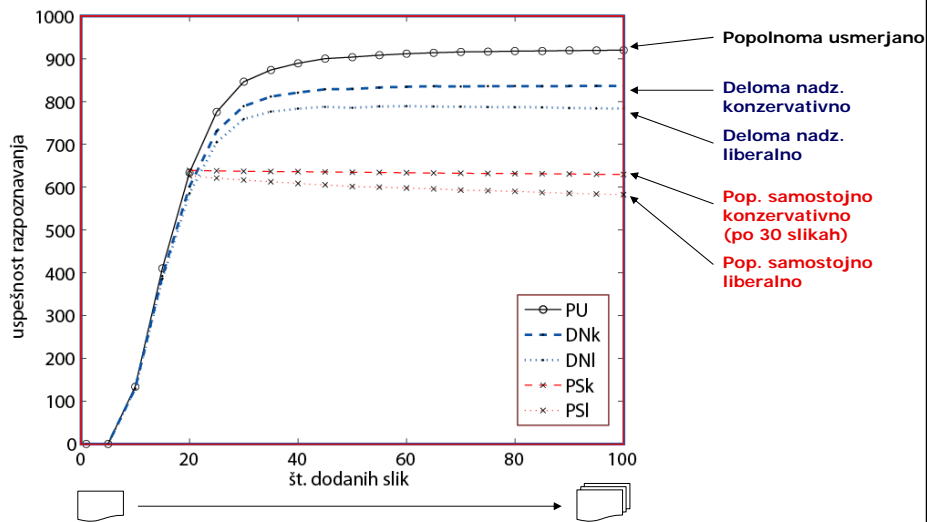
Dva primera

- Avtomatski opis prizora



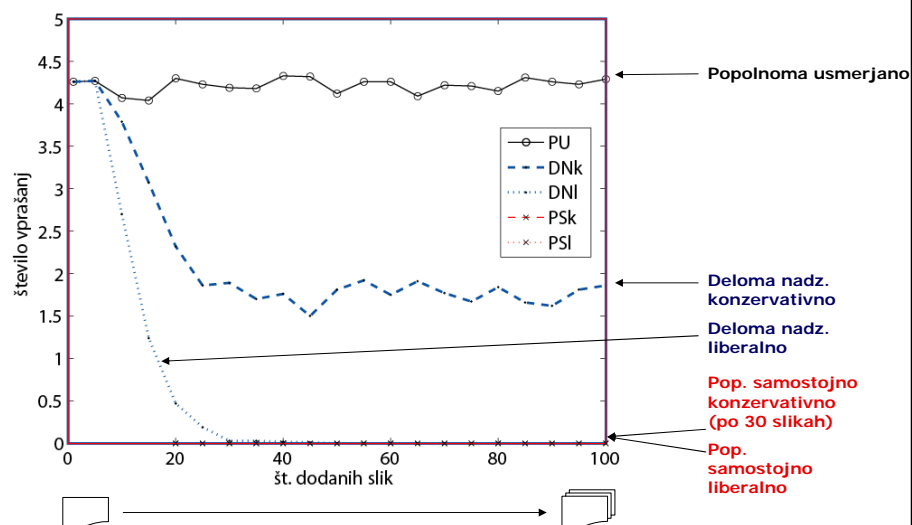
Eksp. rezultati – prostorske relacije

Uspešnost razpoznavanja



Eksp. rezultati – prostorske relacije

Število vprašanj

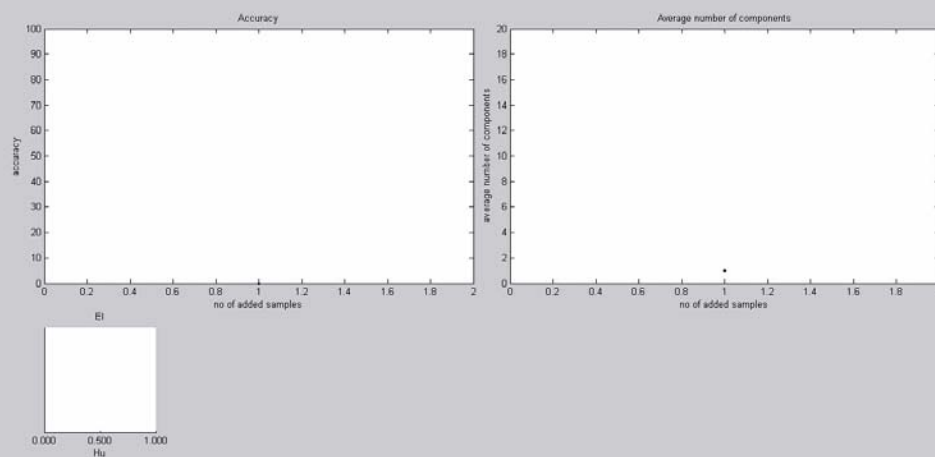


Eksperimentalni rezultati

- Učenje in razpoznavanje vizualnih lastnosti vsakodnevnih predmetov
 - 6 vizualnih značilnic
 - 3 značilnice videza (Hu, Sa, In)
 - 3 značilnice oblike (Ar, Co, Ec)
 - 6 vizualnih lastnosti
 - 4 barve (Rd, Gr, Bl, Yl)
 - 2 obliki (Cm, El)

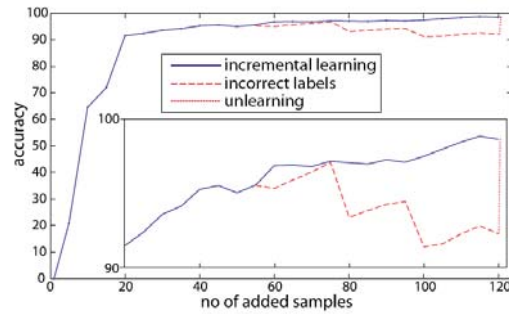


Eksperimentalni rezultati



Eksploimentalni rezultati

- Učenje z napačnimi podatki zelo poslabša rezultate učenja
 - slaba komunikacija
 - nezanesljiv vir informacije
- Rešitev: popravljanje naučenih predstavitev (unlearning)



Porazdeljene inteligentne programske tehnologije

Aplikacije

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Porazdeljene inteligentne programske tehnologije, Aplikacije

MAS v industriji

- Zakaj uporabiti porazdeljene večagentne sisteme v industriji?
- Uporabiti v problemih, ki se lahko uspešno rešijo s takšnimi sistemi!
- Primerne aplikacije:
 - modularne
 - decentralizirane
 - spreminjajoče
 - slabo definirane
 - kompleksne

Porazdeljene inteligentne programske tehnologije, Aplikacije

2

Modularnost

- Agenti so proaktivni objekti
- Najbolj so primerni za aplikacije, ki se jih da naravno razbiti na module
 - Agent zajema entiteto
 - ki je samozadostna
 - ima določeno množico spremenljivk stanj, ki se ločijo od tistih v okolju
 - se lahko določi vmesnik med agentom in okoljem
 - fizične dekompozicije so bolj primerne kot funkcionalne dekompozicije

Decentraliziranost

- Agenti so proaktivni objekti, ki reagirajo sami, brez potrebnega zunanje impulza – so samoiniciativni
- Primerni so za aplikacije, ki se lahko razbijejo v množico samostojnih procesov, od katerih lahko vsak samostojno deluje, ne da bi bil nadzorovan od zunaj.
- Izdelava izdelkov je pogosto zelo decentralizirana
- Nabava sestavnih delov je zelo decentralizirana
=> decentralizirani agenti

Spremenljivost

- Večagentni sistemi so zelo primerni v spremenljivih okoliščinah
 - modularnost omogoča spremenljivost samo določenih komponent
 - decentraliziranost zmanjšuje vpliv spremembe ene komponente na druge
- Sistem se torej lahko spreminja/nadgrajuje hitro, pogosto in brez škodljivih stranskih učinkov
 - velika prednost pri majhnih serijah, prilagajanju posameznim zahtevam, potrošnikom, ipd.
- Omogoča
 - ponovno uporabo obstoječe kode (reuse)
 - samokonfiguracijo

Slabo definirani problemi

- Tradicionalno morajo biti vse zahteve sistema znane v naprej, problem mora biti natančno definiran
- Včasih temu ni tako
 - problem (okolje) je slabo definiran
 - Vsa informacija o strukturi problema ni znana v naprej
 - "paraliza z analizo"
- Pri večagentnih sistemih je
 - Jaz poznan in definiran
 - Okolje je lahko deloma nedefinirano in se lahko spreminja (v določenih mejah)
- Večagentni sistemi omogočajo torej delovanje v dinamičnih spreminjajočih se okoljih
 - načrtuje se množico struktur sistemov in ne samo natančno ene strukture
 - večja prilagodljivost
 - enostavnejša rekonfiguracija

Kompleksnost

- Mera kompleksnosti sistema: na koliko različnih načinov se lahko obnaša?
- Ena možnost je, da se vsa možna obnašanja definirajo med načrtovanjem sistema
 - ob upoštevanju vseh kombinacij posameznih modulov (oz. njihovih obnašanj)
 - eksponentna zahtevnost (prostorska in časovna)
- Boljša možnost je, da se nova obnašanja generirajo po potrebi med samim delovanjem
 - večagentni sistemi!
 - časovno in prostorsko veliko bolj učinkovito

Aplikacije multiagentnih sistemov

- v ekonomiji
- v telekomunikacijah
- v prometu
- doma
- v vojski
- v vesolju
- v zabavni industriji
- ...

... v ekonomiji

- Elektronsko trgovanje
- Modeliranje in optimizacija trgovinskih poti
- Analiza in optimizacija poslovnih procesov



Porazdeljene inteligentne programske tehnologije, Aplikacije

9

... v telekomunikacijah

- Spremljanje in upravljanje telekomunikacijskih mrež (posredovanje in preklapljanje zvez)
- Upravljanje z informacijami v informacijskih okoljih kot je Internet (zbiranje, filtriranje informacij, itn.)

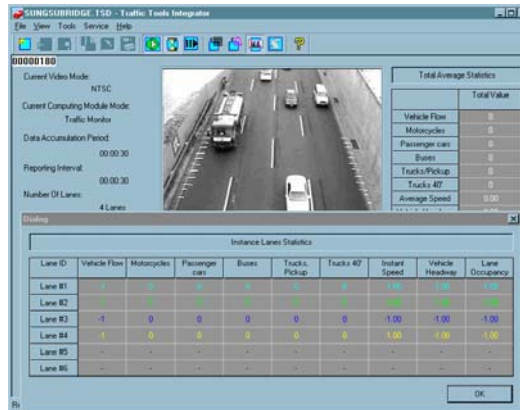


Porazdeljene inteligentne programske tehnologije, Aplikacije

10

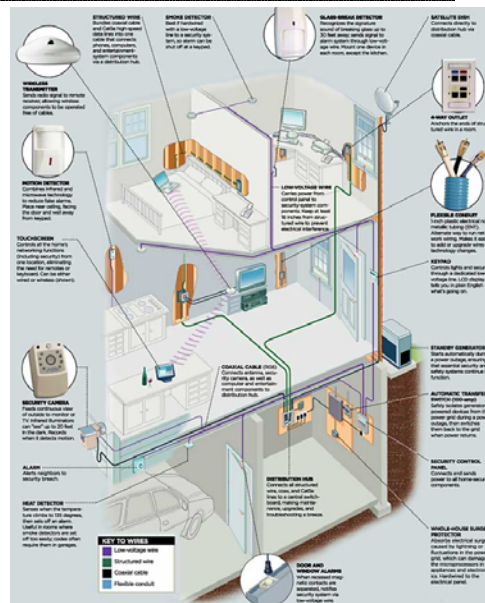
... v prometu

- nadzor cestne infrastrukture, varnost v tunelih
- modeliranje in optimizacije prometnih tokov
- optimizacija logistike
- ...



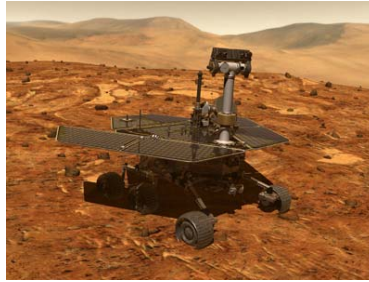
... doma

- Pametna hiša
- Avtomatizacija
 - ogrevanja
 - senčil
 - osvetlitve
 - varnosti
 - ...



... v vojski in vesolju

- Aeronavtika, vesoljske raziskave, vojska
 - Za varno samodejno letenje in pristajanje
 - Večja avtonomija, decentralizirani inteligentni sistemi, decentralizirano odločanje v neznanem, nepredvidljivem in spreminjajočem okolju



Porazdeljene inteligentne programske tehnologije, Aplikacije

13

... v zabavni industriji

- Računalniške igre



Porazdeljene inteligentne programske tehnologije, Aplikacije

14

Lastnosti aplikacij

- Porazdeljenost
 - prostorska porazdeljenost
 - časovna porazdeljenost
 - semantična porazdeljenost
 - funkcionalna porazdeljenost
- Kompleksnost

Prednosti

- hitrost in učinkovitost
 - paralelizem, asinhroničnost
- robustnost in zanesljivost
- skalabilnost in fleksibilnost
- nižji stroški
- razvoj in ponovna uporabljivost

Porazdeljene inteligentne programske tehnologije (PIPT)

Zaključek

Danijel Skočaj
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Vsebina predavanj

- Porazdeljeni večagentni sistemi
 - Inteligentni agenti
 - Večagentni sistemi
 - Porazdeljeno reševanje problemov
 - Porazdeljeno načrtovanje
 - Porazdeljeno odločanje
 - Učenje v večagentnih sistemih
 - Aplikacije
- Porazdeljeni spoznavni sistemi
 - nehomogenost spoznavnih sistemov
 - arhitektura spoznavnih sistemov
 - učenje v spoznavnih sistemih
 - komunikacija v spoznavnih sistemih

Inteligentni agenti

- Kaj je agent?
- Kaj je inteligentni agent?
- Abstraktne arhitekture za inteligentne agente
 - Popolnoma odzivni agenti
 - Zaznavanje
 - Agenti s stanji
- Arhitekture za inteligentne sisteme
 - Agenti osnovani na logiki
 - Odzivni agenti
 - Agenti prepričanje-želja-namen
 - Večplastni agenti

Večagentni sistemi

- Motivacija
- Porazdeljena umetna inteligenca – DAI
- Pomen in tipi sporočil
- Komunikacijski protokol
 - KQML
 - KIF
 - Ontologije
- Interakcijski protokoli
 - Koordinacijski protokoli
 - Protokoli za sodelovanje
 - Pogodbene mreže
 - Sistemi s tablo
 - Pogajanja
 - Ohranjanje prepričanja večagentnega sistema
 - Tržni mehanizmi
- Združbe agentov

Porazdeljeno reševanje problemov

- Primeri porazdeljenih problemov
 - Paralelizem za pospešitev reševanja problema
 - Porazdeljene sposobnosti
 - Porazdelitev prepričanj, procesiranja, podatkov
 - Porazdelitev rezultatov
- Delitev nalog
 - Hanojski stolpi
 - heterogeni sistemi
 - porazdeljena mreža senzorjev
 - neodvisne naloge
- Delitev rezultatov
 - Funkcionalno natančno sodelovanje
 - Deljeni (skupni) repozitoriji
 - Organizacijske strukture
 - Komunikacijske strategije
 - Strukture nalog

Porazdeljeno načrtovanje

- Različni tipi porazdeljenega načrtovanja:
 - Centralizirano načrtovanje za porazdeljene načrte
 - Porazdeljeno načrtovanje za centralizirane načrte
 - Porazdeljeno načrtovanje za porazdeljene načrte
- Predstavitve porazdeljenih načrtov
- Porazdeljeno načrtovanje in izvajanje
 - Koordinacija po načrtovanju
 - Koordinacija pred načrtovanjem
 - Prepletено načrtovanje, koordinacija in izvajanje
 - Delno globalno načrtovanje
 - Sprotna koordinacija načrtov brez komunikacije

Porazdeljeno odločanje

- Evaluacijski kriteriji
 - Socialna blaginja
 - Pareto učinkovitost
 - Individualna racionalnost
- Glasovanje
 - Množinski protokol
 - Binarni protokol
 - Borda protokol
 - Diktatorski protokol
- Dražbe
 - Angleški protokol
 - Zbiranje ponudb
 - Nizozemski protokol
 - Vickrey protokol
- Nashovo ravnotežje
- Barantanje
- Formiranje koalicij

Učenje v večagentnih sistemih

- Strojno učenje in MAS
- Karakterizacija učenja
 - Centralizirano učenje
 - Decentralizirano učenje
 - Značilnosti različnih tipov učenja
 - Problem dodelitve zaslug
- Učenje in koordinacija aktivnosti
 - Vzpodbujevalno učenje
 - Izolirani, sočasni, vzpodbujevalni učenci
 - Interaktivno vzpodbujevalno učenje koordinacije
- Učenje o in od drugih agentov
 - Učenje organizacijskih vlog
 - Učenje v tržnih pogojih
 - Učenje za zmago nad nasprotnikom
- Učenje in komunikacija
 - Učenje za komunikacijo
 - Komunikacija za izboljšanje učenja

Aplikacije

- Primerne aplikacije:
 - modularne
 - decentralizirane
 - spreminjajoče
 - slabo definirane
 - kompleksne
- Aplikacije multiagentnih sistemov
 - v ekonomiji
 - v telekominukacijah
 - v prometu
 - doma
 - v vojski
 - v vesolju
 - v zabavni industriji

Porazdeljeni spoznavni sistemi

- Spoznavnost
- Zahteve za spoznavne sisteme
 - Arhitektura
 - Zaznavanje
 - Predstavitve
 - Učenje
 - Razpoznavanje
 - Akcija
 - Načrtovanje
 - Sklepanje
 - Komunikacija
- Primer spoznavnega sistema
- Atributi spoznavnih sistemov

Arhitektura spoznavnih sistemov

- Različne arhitekture spoznavnih sistemov
- CAS – CoSy aritekturna shema
 - skupen povezanih podarhitektur
 - procesne komponente
 - skupni delovni pomnilnik
 - porazdeljen sistem
 - vzporedno delovanje v realnem času
 - zelo heterogen sistem
 - vizualna PA
 - PA za komunikacijo
 - prostorska PA
 - povezovalna PA
 - mediacijska PA
 - PA za manipulacijo

Nehomogenost spoznavnih sistemov

- Nehomogenost, večmodalnost
 - Predstavitve vizualne informacije
 - Predstavitve avdio informacije
 - Predstavitve lingvistične informacije
 - Nehomogeno sklepanje in načrtovanje
 - Aktuatorji
- Povezovalnik (Binder)
 - Povezovanje med različnimi modalnostmi
 - Sodelovanje med povezovalnikom in učenjem

Učenje v spoznavnih sistemih

- Interaktivno kontinuirano večmodalno učenje
- Učenje večmodalnih asociacij
- Večmodalno učenje in povezovalnik
- Eksplicitno in implicitno učenje
- co-learning, unlearning
- Različni načini učenja
 - Popolnoma usmerjano
 - Deloma nadzorovano
 - konzervativno
 - liberalno
 - Popolnoma samostojno
 - konzervativno
 - liberalno

Literatura

- Glavna literatura:
 - Multiagent Systems - Modern Approach to Distributed Artificial Intelligence Edited by Gerhard Weiss, The MIT Press
 - Poglavja: 1, 2, 3, 5, 6 in 9
 - Nekatera poročila in članki projekta CoSy - Cognitive Systems for Cognitive assistants
 - <http://cognitivesystems.org/>
 - Dodatna literatura:
 - Michael Wooldridge
An Introduction to MultiAgent Systems
- Gradivo na spletni strani predmeta:
 - <http://ucilnica.fri.uni-lj.si/course/view.php?id=113>

