

1. Realtime / online simulace

offline simulace

- strategické plánování
- taktické plánování
- týká se
 - stabilních objektů (infrastruktura)
 - permanentních mobilních objektů (zdroje)
 - temporálních mobilních objektů (zákazníci, zakázky)
 - organizace interních procesů (postupy, pravidla)
 - organizace IO (regulace vstupů/výstupů)
- průběh
 - Input analysis (sběr a analýza vstupních dat)
 - Modelování, verifikace, validace
 - Simulace (realizace experimentů)
 - Output analysis (sběr a analýza výstupních dat)
- cílem může být robustní plán řešící i náhodné vlivy

online simulace

- operativní plánování
- napojení na reálný systém – průběžné získávání dat
- běží rychleji než realita a “předpovídá možné chyby”
- provozní plán je nutné modifikovat
- způsoby plánování
 - reaktivní – nové plánování až když už je současný plán nepoužitelný
 - iniciativní – postupné a stálé úpravy plánu
- modifikace plánu vychází z předpovědi problému v blízké budoucnosti

školení

- offline simulace
- trenažéry
- vizuální interaktivní simulace
 - lze se přepnout do interaktivního režimu a ovlivňovat simulaci
- vstupem jsou buď náhodná nebo historická data

2. Agenti

Agent

- zapouzdřený systém
- reaguje na okolí
- plní cíl
- vlastnosti
 - autonomní – řídí se sám
 - společenský – komunikuje s okolím
 - reaktivní – reakcí na nějaký vstup je výstup
 - iniciativní – vyvíjí vlastní iniciativu na základě vnitřního stavu, učí se
- klasifikace
 - mobiní a statické (na jednom pc)
 - iniciativní a reaktivní
 - dále autonomní a kooperativní, učí se a neučí se
- proč jsou agenti vhodní
 - decentralizovaní
 - samostatní

- žádný globální super-model
- divide and conquer (každý dělá dobře jednu malou věc)
- při vlastní implementaci se řeší
 - způsob implementace jednotlivých agentů
 - způsob komunikace
 - koncept pro synchronizaci

Reaktivní agenti

- nedisponují interním symbolickým modelem svého okolí
- styl práce podnět -> odezva
- dá se říct že nemají žádnou paměť
- připomínají senzorické systémy
- sestává z kolekce modulů
 - senzorický modul
 - výpočetní modul
 - ...
- moduly komunikují málo a primitivně
- více reaktivních agentů může tvořit inteligentní systém

Architektura ABASim

- architektura definuje
 - základní stavební prvky (událost, aktivita, proces, agent)
 - realizace alokace simulačního výpočtu (monolitická, paralelní, distribuovaná)
- skládá se z reaktivních agentů
- agent
 - má cíl, nebo si jej musí umět sám identifikovat
 - životní cyklus
 - rozpoznání situace
 - rozhodnutí o řešení
 - výkon řešení
 - schopnosti
 - komunikace s okolím
 - navrhovat jednorázová řešení problémů, vytvářet dlouhodobější plány
 - jednorázově se dotazovat na stav systému, nepřetržitě snímat části okolí
 - jednorázová aktivovace akce, spuštění nepřetržitého procesu
- dekompozice agenta
 - důvody dekompozice
 - sdílení komponent
 - vytvoření alternativních komponent a výběr konkrétní až v runtime
 - části
 - manažer – komunikuje s ostatními komponentami a okolím
 - senzory – dotaz (okamžitý) a monitor (delší snímání)
 - řešitelé
 - poradce – okamžitý, optimalizační algoritmus nebo člověk
 - plánovač – předvídá problémy, upozorňuje manažera, dělá plány a aktualizuje je
 - efektory
 - akce – okamžité
 - procesy – předem definovaně něco mění, např. v časových intervalech
 - asistenti - senzory, řešitelé a efekторы dohromady (vše kromě manažera)
 - kontinuální asistenti - procesy, monitory, plánovače – trvající nenulový čas

- promptní asistenti – akce, dotazy, poradci – okamžité

Multiagentový přístup

- zachycuje hierarchickou strukturu
- delimitace
 - rozdělení oblasti problému do více oddělených částí
 - například – severní hory a jižní hory
 - jedna část o té druhé vůbec nemusí vědět, může fungovat bez ní
- delegace
 - předání části problému jinému podřízenému agentovi
 - například severní hory
 - požár se předá hasičům
 - krádež se předá policii
 - oboje se může stát na tom samém území, území je sdílené
 - například jižní hory
 - požár i krádež se předá šerifovi

Vrstvový model MPE/ABAsim

- MPE = Management, Processing, Entities
- vrstvy managementu, processingu a nepovinně vrstva entit
- management rozhoduje
- processing vykonává (ať už akce nebo vrací třeba výsledky měření)
- entity udržují stav
 - fyzické – stav systému
 - informační – postupy, knowledge base, statistiky
- manažeři ve vzájemné kooperaci za pomoci senzorů a řešitelů startují efektory ve správném čase a za správných podmínek

Konuminační mechanismus ABAsim

- především zprávy
- zprávy jak mezi agenty tak mezi agentem a asistentem
- zprávy manažer-> manažer
 - notice – informuje o čemkoli
 - request – žádost o informaci
 - response – odpověď na žádost
 - zasílá se pouze jako odpověď tomu kdo poslal request
- zprávy manažer -> asistent
 - start – zahájení činnosti
 - break – násilné ukončení činnosti zapnuté startem
 - execute – okamžité vykonání činnosti – asistent ihned vrací výsledek
- zprávy asistent -> někdo
 - finish – manažerovi, ukončení činnosti spuštěné startem
 - notice – manažerovi, informační zpráva o něčem – mezi startem a finishem
 - hold – kontinuální asistent sobě, jako jediná má časové razítko, do doby než asistentovi dorazí (nastane daný čas), tak asistent pokračuje v činnosti, pak může udělat něco jiného
- posun času je pouze jedinou zprávou – hold
- synchronizace je pouze za pomoci kontinuálních asistentů
- v management vrstvě “neplyne” simulační čas
- zprávy
 - adresní – cílový agent musí být znám

- částečně adresní – cílový agent není znám, ale je známo že je to někdo z množiny A_j
- neadresní
- v případě neadresních nebo částečně adresních zpráv se spouští algoritmus vyhledání adresáta...podle toho o jaký typ zprávy se jedná nebo podle dalších kritérií
 - hledají se agenti schopní zpracovat stejnou množinu zpráv
- zlaté stránky
 - přiřazují k určité službě vždy určitého agenta
 - když známe co chceme udělat není problém najít toho kdo to udělá

Příklad

- manžeři
 - správce obsluhy
 - akce přemístění zákazníka, akce obsluha A, akce obsluha B
 - správce zdrojů
 - akce přemístění zdroje
 - okolí
 - akce vstupování zákazníků

3. Synchronizace

- pošta
 - přímé doručování
 - každá komponenta má vlastní poštovní schránku
 - centrální doručování
 - kombinovaný přístup
 - 1. fáze – vyprázdnění vlastních schránek
 - 2. fáze – vybrání všech zpráv s nejmenším časovým razítkem, doručení
 - návrat do fáze 1. a tak stále dokola
- procesy
 - diskrétní – pouze diskrétní aktivity, změny jsou okamžité, provedou se, poté se skočí do dalšího okamžiku kdy je nějaká změna naplánována a zase to samé
 - spojitě – pouze spojitě aktivity, zpracovávání aktivit pomocí metody snímání aktivit
 - kombinované – složen z obou aktivit
- snímání aktivit – s určitou časovou periodou se snímají aktivity a rozhoduje se zda už aktivita skončila nebo ne
- rozdělení do modulů
 - modul diskrétní simulace
 - modul spojitě simulace
 - modul animace
 - modul rozhraní
- diskrétní aktivity
 - aktivita proběhne ihned (spustí se buď startem nebo holdem od předchozí aktivity)
- spojitě aktivity
 - při startu aktivity se zaregistruje aktivita u modulu spojitěho simulátoru
 - ten ji pak posouvá v intervalech simulačního času v nichž je spojitý simulátor aktivní
 - aktivita pošle po dokončení hold tomu procesu který ji registroval
- simulační jádro spojitě simulace

- vybere zprávy s nejmenším timestampem z centrální pošty, doručí je a adresáti je zpracují, a tak stále dokola
- simulační jádro kombinované simulace
 - než vybere zprávy z centrální pošty, zjistí kolik simulačního času je mezi posledním výběrem z centrální pošty a časem dalšího výběru (tím že porovná čas posledního výběru s časem zprávy s nejmenším timestampem v poště)
 - pokud je rozdíl času větší než nula, je předáno toto časové kvantum modulu spojitě simulace, po vyčerpání kvanta je předáno řízení zpět diskrétnímu modulu simulace a ten vybere a doručí zprávy (opět s nejmenším ts) a celé se to pak opakuje znovu
 - spojitá simulace probíhá v časech kdy se v rámci diskrétní simulace nic neděje
 - modul spojitě simulace metodou snímání aktivit “posunuje” spojitě aktivity dál
 - modul spojitě simulace nemusí využít kvantum celé pokud nějaká aktivita odešla hold, v tomto případě se provede opět “diskrétní část simulace”
 - to znamená že se buď aktivita ukončila nebo nastala situace která má vliv na diskrétní část
- modul rozhraní
 - obsahuje vyrovnávací paměť kam jsou zapisovány informace od aktivit procesů (spojité i diskrétní)
 - jsou zde informace pro grafické zobrazování
 - jádro animace z ní čte
- modul animace
 - odsimulované časové kvantum spojitou simulací (nebo přeskočené pokud tam není žádná aktivita) je přiděleno modulu animace kde se toto kvantu odanimuje, vyrovnávací paměť už obsahuje potřebná data které tam uložily moduly spojitě i diskrétní simulace při svém provádění
 - animuje se pouze to co se odsimulovalo!!! tzn. pokud přijde ve spojitě simulaci hold, animuje se také pouze stejné kvantum od začátku tohoto simulování po hold
 - animační aktivity se také registrují
 - také využívá metodu snímání aktivit
 - po odanimování je možné vyprázdnit vyrovnávací paměť rozhraní

Existují alternativní standarty pro komunikaci mezi agenty KQML (Knowledge query manipulation language) nebo FIPA-ACL (Agent communication language)

4. Petriho síť

5. Distribuovaná a paralelní simulace

Sdílená paměť => paralelní systémy

- multiCPU / multithread system
- nízká latence

Distribuovaná paměť => distribuované systémy

- clustery, nezávislé počítače atd
- spojení často po síti
- vysoká latence

- mohou být heterogenní
- výhody
 - rychlejší
 - integrace – spolupráce heterogenních systémů
 - geografická distribuce
 - failover – ostatní uzly mohou převzít činnost při výpadku jednoho uzlu

Logický proces = LP

- sekvenční simulační model
- komunikují zasíláním zpráv s časovým razítkem
- dohromady tvoří distribuovaný simulační model

Distribuovaná simulační model

- obsahuje konečnou množinu logických procesů
- obsahuje konečnou množinu komunikačních kanálů mezi LP
- LP nesdílejí proměnné, ale posílají si pouze zprávy
- LP udržuje vlastní hodnotu lokálního virtuálního času

Synchronizace

- každý LP provádí zpracovávání událostí v kauzálním pořadí, tj. s neklesající hodnotou časového razítka
- synchronizační protokol – zaručuje stejné výsledky jako s jedním sekvenčním simulátorem

LCC = local causality constrain

- proces musí zpracovávat zprávy s neklesající hodnotou časového razítka

Metody synchronizace

- konzervativní (striktně dodržují LCC)
- optimistické (při porušení LCC => rollback)

Konzervativní asynchronní metoda synchronizace

- předpoklad – zprávy posílané po lince z daného spojení jsou v neklesající posloupnosti
- LP má pro každé propojení jednu prioritní frontu – utříděné podle TS
- LP čeká dokud není v každé frontě alespoň jedna zpráva a pak zpracovává tu s zprávu s nejmenším časovým razítkem
- koncepce výhledu
 - zaručí že žádný proces nemůže naplánovat jakoukoliv událost v čase $T + L$ a menším
 - pak je jasné že lze zpracovat všechny zprávy a razítkem menším $T + L$, protože alespoň do doby $T + L$ nemůže nic nového přijít
 - bez výhledu proces zpracovává pouze všechny zprávy s nejmenším TS
- uvážnutí = deadlock
 - řešení předcházením
 - při každé změně pošle LP NullMessage na všechny spojení

- NullMessages lze posílat buď vždy nebo na pouze na požádání
- řešení s detekcí a zotavením
 - řídicí proces si ukládá v jakých stavech jsou ostatní procesy (buď aktivní nebo blokováný)
 - 1. řídicí proces informuje jeden nebo více LP o tom které zprávy jsou bezpečné (ty nemohou způsobit narušení LCC)
 - 2. LP zpracují bezpečné události a ty mohou způsobit další události
 - 3. Nemůželi LP dále zpracovávat zprávy – je blokováný
 - 4. Jestliže je blokováný řídicí proces, je simlace ve stavu uváznutí a jede se znovu od bodu 1.
 - nutno identifikovat bezpečné zprávy
 - zpráva s nejnižším TS je vždy bezpečná

Konzervativní synchronní metoda synchronizace = bariéry

- bariéra určuje že žádná zpráva s časem menším než T nemůže být vyprodukována
- LP po dokončení činnosti čeká na bariéru, bariéra je uvolněna až když všechny LP dokončí činnost, čímž je zajištěna synchronizace
- nutno zabezpečit
 - blokování procesů před bariérou
 - žádné tranzitní zprávy (odeslané, ale ještě nedoručené) – lze řešit počítadlem zpráv
- Centralizovaná bariéra
 - jedna hlavní bariéra, soustředění komunikace na jednom místě, hodně zpráv na jednom místě
- Stromová bariéra
 - LP jsou ve vyváženém stromu
 - LP bude volat bariérovou synchronizaci až po tom co mu všichni jeho potomci řekli že také dosáhli bariéry
 - při otevření bariéry to zase LP sdělí svým potomkům
 - kořen stromu po dosažení bariéry a obdržení od potomku bariéru uvolní
- Motýlkový bariéra
 - postupná párová synchronizace procesů

Optimistická metoda synchronizace = TimeWarp algoritmus

- dvě složky řízení
 - lokální řídí činnost LP – nezávisle od ostatních LP
 - globální – provádění nevratných operací, např. IO, uvolňování paměťového prostoru
 - disponuje znalostí všech LP
- lokální řízení
 - každý LP má kalendář událostí – může obsahovat i události jiných LP
 - LP zpracovává události, ale netestuje bezpečnost LCC
 - pokud je doručena opožděná zpráva s nižším TS, je třeba provést rollback
- rollback
 - nutno navrátit stav tak jak byl předtím a zrušit poslané zprávy
 - kopírování stavu nebo incrementální ukládání změny stavu

- inkrementální se může hodit pokud se mění jen malá část stavového prostoru
- zrušení odeslaných zpráv
 - antizprávy – každé odeslané zprávě odpovídá antizpráva, ta se pošle a ruší účinek zprávy
 - LP si udržuje připravené antizprávy ve frontě
 - pokud přijde antizpráva
 - pokud nebyl originál zpracován, originál se smaže – anihilace
 - pokud byl zpracován – sekundární rollback
 - pokud ještě nedorazil – počká se na něj a anihiluje se
 - existuje možnost livelocku – antizpráva zruší jinou zprávu se stejným TS, opatření další atribut zprávy
- globální řízení
 - provádí nevratné operace
 - řeší problémy se správou paměti – stanovení událostí které již není možné vrátit
- globální virtuální čas
 - je roven minimální hodnotě TS všech nezpracovaných a částečně zpracovaných zpráv a antizpráv
 - GVT označuje čas kdy je jisté že více zpět už se rollback neprovede

Konzervativní metody, nižší režie, nižší paralelizmus

Optimistické metody, vyšší režie, lze lépe paralelizovat