

Agent-Oriented Methodologies – A Comparative Analysis

Asist. Ana-Ramona LUPU

Catedra de Informatică Economică, A.S.E. București

Agent-oriented methodologies and related modeling techniques have become a priority for the development of multi-agent systems. The methodologies proposed remain incomplete: they are either an extension of object-oriented methodologies or an extension of knowledge based methodologies. Trying to assist the decision of which methodology to follow when constructing agent applications, this paper presents a comparative analysis of three existing multi-agent system methodologies.

Keywords: multi-agent system, agent-oriented methodologies.

Introducere

Deși atât comunitatea științifică, cât și mediul de afaceri au recunoscut avantajele potențiale ale sistemelor bazate pe agenți, numărul de astfel de aplicații comerciale nu este mare. Unul din motive este lipsa de maturitate a metodologiilor pentru dezvoltarea de aplicații bazate pe agenți. Evaluarea punctelor bune și a punctelor slabe ale acestor metodologii joacă un rol important în îmbunătățirea lor și dezvoltarea următoarei generații de metodologii. Această lucrare prezintă pe scurt trei dintre metodologiile orientate agent: MaSE, GAIA și MAS-CommonKADS. Deoarece una din problemele importante în construirea de aplicații bazate pe agenți este decizia asupra metodologiei ce va fi folosită, este realizată ulterior și o comparație a acestora. În teorie au fost propuse mai multe metodologii pentru dezvoltarea de sisteme multiagent, dar ele rămân incomplete: ele sunt fie o *extensie a metodologiilor orientate-obiect*, fie o *extensie a metodologiilor bazate pe cunoștințe*. Aceste metodologii nu oferă însă caracteristici adecvate specificului agenților, cum ar fi starea lor sau comportamentul social în contextul sistemelor multiagent.

AOM și Gaia

Agent-Oriented Methodology (AOM) și *Gaia*, care extinde AOM, sunt metodologii care se concentrează pe aspectul modelării sistemelor bazate pe agenți. Construirea sistemului bazat pe agenți este privită ca un proces de proiectare organizațională, plecându-se de la

cerințe, de la concepte abstracte, și ajungându-se în final la proiectarea detaliată, care lucrează cu concepte concrete.

Cea mai abstractă entitate în ierarhia conceptelor este sistemul (sau organizația), privit ca o colecție de roluri aflate în anumite relații unele cu altele și participând sistematic la interacțiuni cu alte roluri. Un rol este privit ca o descriere abstractă a unei funcții a unei entități. Fiecare agent poate juca un anumit rol în cadrul sistemului, sau mai mulți agenți pot juca același rol. Un rol este definit prin trei atribute:

- *Responsabilități* – determină funcționalitatea agentului, fiind împărțite în două categorii: *proprietăți de supraviețuire*, care descriu stările de evoluție a agentului în funcție de condițiile de mediu și *proprietăți de siguranță*, care sunt invariante și descriu starea globală a sistemului sau diverse restricții impuse.
- *Permițiuni* – reprezintă drepturile asociate unui rol și identifică resursele pe care acesta le are la dispoziție pentru realizarea responsabilităților sale.
- *Protocoale* – reprezintă calea prin care un rol poate interacționa cu alte roluri.

Analiza sistemului are ca principal obiectiv înțelegerea optimă a sistemului și a structurii sale, fără referiri la detaliile de implementare. Această fază va avea ca rezultat modelul rolurilor și un model de interacțiune care descrie interacțiunile dintre roluri.

Modelul rolurilor identifică rolurile principale din sistem, funcționalitatea fiecăruia dintre ele fiind definită cu ajutorul responsabilității-

lor ce le revin. Modelul rolurilor este specificat printr-un set de scheme de roluri.

Modelul interacțiunilor reprezintă legăturile dintre diverse roluri, constând dintr-un *set de definiții de protocole*, unul pentru fiecare tip de interacțiune. Aceste definiții au următoarele atribute: acop, inițiator, destinatar, intrări, ieșiri, procesare.

Procesul de analiză presupune reluarea iterativă a următorilor pași: identificarea rolurilor din sistem, rezultând un *model prototip al rolurilor*; identificarea și documentarea protocolelor asociate fiecărui rol, rezultând un *model de interacțiune*; elaborarea modelului rolurilor pe baza modelului interacțiunilor, rezultând un *model elaborat complet*.

Faza de proiectare

Activitatea de proiectare presupune elaborarea a trei modele: modelul agenților, modelul serviciilor și modelul cunoștințelor.

Modelul agenților este definit utilizând un arbore de tipuri de agenți, fiecare nod rădăcină corespunzând unui rol, iar celelalte noduri corespunzând tipurilor de agenți. Tipurile de agenți pot fi adnotate (“n”, ”m.n”, ”*”, ”+”), pentru a indica numărul de agenți de un anumit tip din sistem.

Modelul serviciilor identifică serviciile asociate fiecărui rol (cel mult unul pentru fiecare rol) și proprietățile lor. Prin serviciu se înțelege o funcție a agentului, echivalent unei metode în metodologia orientată obiect. Proprietățile unui serviciu sunt: *intrările și ieșirile* – derivate din modelul protocolelor și *pre- și postcondițiile* – derivate din proprietățile de siguranță ale unui rol, reprezintă restricții impuse serviciilor.

Modelul cunoștințelor definește legăturile de comunicare ce vor exista între tipurile de agenți, reprezentate prin căi de comunicare, nu prin formatul/conținutul mesajelor vehiculate. Pe baza modelor precedente se va construi un graf orientat, având tipurile de agenți ca noduri și legăturile de comunicare ca arce.

MaSE

Multiagent System Engeneering (MaSE) are ca obiectiv principal oferirea unei metodologii pentru asistarea proiectării și dezvoltării

sistemelor multiagent pe durata întregului ciclu de viață. Acest proces constă din șapte pași, repartizați în două faze:

Faza de analiză include trei procese:

- *formularea scopurilor* – presupune identificarea scopurilor și a structurii lor și reprezentarea lor sub forma unei ierarhii de scopuri.

- *aplicarea cazurilor de utilizare* – presupune extragerea scenariilor principale din contextul sistemului inițial. Aceste cazuri de utilizare sunt folosite la construirea unui set de diagrame de secvență (similare cu cele din UML).

- *rafinarea rolurilor* – implică construirea *modelului rolurilor* și *modelului sarcinilor concurente*. Modelul rolurilor descrie rolurile din sistem, scopurile, sarcinile pentru fiecare rol și căile de comunicare între roluri. Sarcinile sunt reprezentate apoi detaliat ca un set de automate cu stări finite în modelul sarcinilor concurente.

Faza de proiectare include următoarele procese:

- *Crearea claselor agent* – are ca rezultat diagrama claselor agent care descrie agenții și rolurile pe care le joacă, legăturile între agenți simbolizând conversații și fiind etichetate cu numele acestora

- *Construirea conversațiilor* - realizează o descriere acestor conversații folosind o diagramă a claselor de comunicare sub formă de automat cu stări finite.

- *Asamblarea claselor agent* - presupune definirea arhitecturii agenților și componentelor acesteia. Nu este impusă nici o platformă de implementare.

- *Proiectarea sistemului* – implică construirea unei diagrame de desfășurare care specifică localizarea agenților în cadrul sistemului. Din perspectiva MaSE, agenții sunt o specializare a obiectelor care se coordonează prin conversație și acționează proactiv pentru atingerea scopurilor. Se poate alege între a dezvolta propria arhitectură de agent sau a folosi arhitecturi predefinite, cum este BDI.

MaSE are suport utilitar sub forma agentTool, care implementează toți cei șapte pași MaSE și oferă suport automat pentru

transformarea modelelor de analiză în construcții de proiectare.

MAS Common KADS

Este o metodologie multiagent orientată pe dezvoltarea sistemelor bazate pe cunoștințe care extinde o metodologie mai veche, Common KADS (standardul european pentru modelarea cunoștințelor). Această extensie adaugă tehnici din metodologiile orientate obiect și proiectarea orientată pe responsabilități. Procesul software al acestei metodologii combină iterativ și incremental abordarea orientată pe risc cu cea bazată pe componente. În cazul MAS Common KADS fiecare iterație este organizată în următoarele faze: conceptualizare, analiză, proiectare, programare și testare, integrare și testare globală, desfășurare. Pe parcursul fazei de conceptualizare se urmărește obținerea unei descrieri preliminare a problemei.

Faza de analiză se concentrează pe specificarea cerințelor sistemului multiagent prin dezvoltarea unor modele bazate pe o abordare orientată pe risc. Pașii parcurși sunt:

- *modelarea agenților* – sunt specificate caracteristicile agenților, folosindu-se cazuri de utilizare, șabloane textuale pentru agenți și planșe de colaborare pentru responsabilitățile claselor.
- *modelarea sarcinilor* – sunt descrise sarcinile pe care le poate realiza un agent (numele sarcinii, o scurtă descriere, intrări și ieșiri, structura sarcinii, etc.)
- *modelarea coordonării* – este modelată conversația între agenți și presupune: definirea canalelor de comunicație și construirea unui prototip, urmată de analiza interacțiunilor și determinarea interacțiunilor complexe (cu protocoale de coordonare). Sunt folosite diagramele de flux, de tranziție a stărilor, biblioteci de protocoale și diagrame de secvență pentru descrierea/definirea de protocoale noi.
- *modelarea expertizei* – sunt descrise cunoștințele de care are nevoie un agent pentru atingerea scopurilor sale, atât cele legate de aplicație (domeniu, inferențe, sarcini), cât și cele necesare rezolvării problemelor. Pentru modelare se apelează la un limbaj de modela-

re conceptuală sau digramele modelului de obiecte, structurile de inferență și de descompunere a inferenței bazate pe sarcini.

- *modelarea organizării* - sunt definite mediul organizațional al sistemului multiagent și organizarea socială a agenților, folosindu-se notațiile modelului de obiecte, cu simboluri speciale pentru a face distincția între obiect și agent.

Faza de proiectare constă în trei submodele: *modelul rețelei de agenți*, *modelul agenților pentru compunerea sau descompunerea agenților* rezultați din analiză și *modelul platformei* pentru selectarea platformei de dezvoltare a agenților pentru fiecare arhitectură de agenți. Restul fazelor trebuie abordate în maniera obișnuită a metodologiilor pentru inginerie software și nu sunt explicate în detaliu.

MAS-Common KADS poate fi folosită pentru orice tip de agent, dar nu acoperă aspectele legate de mobilitate. Introduce un concept slab de agent, care privește agenții ca entități care pot schimba mesaje printr-un limbaj de comunicare, dar și un concept puternic, care privește agenții ca antități cu stări proprii.

Comparație între metodologiile orientate agent prezentate

Comparația între metodologiile prezentate urmărește mai multe criterii, care sunt structurate pe câteva mari dimensiuni.

Dimensiunea metodologiei urmărește criteriul cum ar fi: etapele proceselor metodologiei, modelul de dezvoltare și posibilitatea reutilizării modelelor, disponibilitatea instrumentelor software pentru asistarea metodologiei respective. Din punctul de vedere al ciclului de viață a dezvoltării sistemului, toate metodele acoperă specificarea cerințelor, proiectarea arhitecturală și proiectarea de detaliu. Faza de implementare însă nu este bine reflectată, exceptând validarea, verificarea și evaluarea ergonomică, toate metodologiile acoperă toate celelalte faze de dezvoltare. MAS-Common KADS ia implicit în considerare verificarea și evaluarea ergonomică, dar nici una din metodologii nu precizează modelele destinate acestui scop sau abordarea folosită. MaSE menționează testarea și desfășurarea,

dar suportul oferit nu este clar specificat. Toate metodele oferă modele reutilizabile și suport software sau metodologic, MAS–Common KADS remarcându-se prin furnizarea unui suport relativ elaborat în acest sens.

Dimensiunea reprezentării urmărește criterii cum ar fi: divizarea sistemului (nivele de abstractizare, generalizare-specializare, strategii-tactici etc), formalizările folosite (diagrame, concepte și reguli), calitatea modelelor (numărul și coeziunea modelelor, complexitatea și complexitatea modelelor).

Toate metodologiile oferă mijloace de control a complexității sistemului dezvoltat, dar doar MAS–Common KADS ia în considerare aspecte legate de strategiile și tacticile acestui sistem. Formalizările de reprezentare folosite pentru date, procese și activități, respectiv pentru dinamica sistemului, se constituie ca o extensie a tehnicilor orientate obiect sau a celor folosite de metodologiile bazate pe cunoștințe. Formalizările folosite de MaSE

permite reutilizarea automată a componentelor și facilitează verificările.

Derivarea este suportată de GAIA și MaSE, și nu este clar specificată în MAS–Common KADS. Modelele metodologiilor nu acoperă toate dimensiunile dezvoltării sistemelor multiagent, aceasta rămânând incompletă. Doar în MAS–Common KADS pare aproape completă, dar modelele sunt complexe, dificil de aplicat în practică.

Dimensiunea agenților urmărește criterii cum ar fi: natura agenților, tipurile de agenți și atributele acestora. Toate metodologiile luate în considerare permit proiectarea sistemelor multiagent bazate pe agenți eterogeni. Modul în care sunt reflectate diferitele criterii este prezentat în tabelul de mai jos. Cea mai completă pare a fi MAS-Common Kads care ia în considerare aproape toate criteriile. Prin modelul său de expertiză aceasta ia în considerare comportamentul deliberativ al agenților.

Tabelul 1. Caracteristici generale ale agenților în sistemele dezvoltate

Criteriu	Valori criteriu	Gaia	MaSE	MAS-Common KADS
Natura agenților	Omogenă	N	N	N
	Eterogenă	D	P	D
Tipuri de agenți	Agenți inteligenți	P	P	D
	Agenți de interfață	P	P	D
	Agenți mobili	P	P	D
	Agenți informaționali	P	D	D
	Agenți autonomi	P	P	D
Atribute agenți	Adaptabilitate	P	P	P
	Autonomie	P	P	P
	Comportament cooperativ	P	P	P
	Capacități de inferență	P	P	P
	Abilități de comunicare	P	P	P
	Mobilitate			
	Personalitate	P	P	D
	Reactivitate	P	P	D
	Continuitate temporală	P	P	
	Comportament deliberativ	P	P	D

Dimensiunea organizațională definește structura pe care trebuie să o aibă sistemul dezvoltat și caracteristicile mediului în care acesta va funcționa, indicând dacă metodologia spe-

cifică în mod explicit această structură și caracteristici.

În teorie, toate cele trei metodologii suportă sisteme distribuite, dar nici una nu oferă su-

port real pentru sisteme deschise. Structurile organizaționale (tipul de organizație – sistem ierarhic/distribuit/deschis/holonic, natura sau tipul mediului, caracteristicile datelor prelucrate) nu sunt identificate cu claritate.

Dimensiunea cooperării urmărește criteriile cum ar fi: modul și limbajul de comunicare, modelul de cooperare (negociere, planificare, delegarea sarcinilor), și modul de interacțiune (static, dinamic, cu protocoale de interacțiune implicite sau explicite, mecanisme pentru rezolvarea stărilor necooperative între agenți etc.).

Toate metodologiile pot modela comunicarea între agenți autonomi și între agenți și utilizatori. Cu excepția GAIA, modul de comunicare folosit de aceste metodologii poate fi direct, sincron sau asincron și limbajul de comunicare poate fi bazat pe semnale și acte de vorbire. Tipul de control, definit în teorie ca fiind distribuit, rămâne cel implicit, mai mult sau mai puțin specificat.

În MAS–Common KADS interacțiunea este dinamică, și este statică în celelalte două cazuri. Toate metodologiile suportă protocoale de interacțiune. Pentru rezolvarea stărilor necooperative, GAIA și MAS–Common KADS oferă un mecanism de interacțiune pentru rezolvarea conflictelor simple. Așadar, metodologia GAIA pare să ofere un suport slab pentru cooperarea agenților.

Concluzie

Metodologiile prezentate oferă un suport consistent pentru conceptele de bază ale orientării agent: autonomie, proactivitate, reactivitate etc. În privința procesului de dezvoltare, toate oferă exemple și euristici pentru a asista dezvoltarea din momentul stabilirii cerințelor până la proiectarea de detaliu, dar multe dintre aspectele legate de faza de implementare rămân neclare.

Metodologiile prezentate sunt încă proiecte experimentale, astfel că unele acoperă doar o parte a ciclului de viață, concentrându-se asupra anumitor aspecte ale procesului de dezvoltare. Documentația apare cel mai adesea sub forma unor rapoarte de cercetare sau lucrări de doctorat. Alte probleme ar fi legate de lipsa de software utilitar de calitate pentru

asistarea construirii sistemelor, instrumentele propuse oferind doar suport parțial și trebuind dezvoltate pentru a acoperi toate etapele și pașii, inclusiv generarea de cod.

O restricție foarte importantă a metodologiilor prezentate este aceea de modelare doar a sistemelor închise. În condițiile în care intensificarea utilizării Internetului a condus la dezvoltarea sistemelor dinamice deschise, pentru a putea fi folosite în sisteme reale metodologiile vor trebui să includă aceste caracteristici ale sistemelor.

Eforturile făcute în direcția standardizării metodologiilor, platformelor și mediilor orientate agent sunt încă la început și mai au un drum lung de parcurs. Succesul paradigmei agenților reclamă metodologii sistematice pentru specificarea analizei și proiectării aplicațiilor multiagent reale.

Bibliografie

- 1.P. Cuesta, A. Gomez, J.C. Gonzalez, F.J. Rodrigues – A Framework for Evaluation of Agent Oriented Methodologies, Technical Report IT1/2003, Departamento de Informatica (Universidade de Vigo), 2003.
- 2.Sabas, S. Delisle, M. Badri – A Comparative Analysis of Multiagent System development methodologies: Towards a Unified Approach, în Third International Symposium from Agent Theory to Agent Implementation, Vol. 2, 2002.
- 3.K.H. Dam, M. Winikoff - Comparing Agent-Oriented Methodologies, Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2003) at AAMAS03, Melbourne, Australia, July 2003.
- 4.W. Shen – Evaluation of Modeling Techniques for Agent Based Systems, A Tutorial Report for SENG 609.22, University of Calgary, 2002.
- 5.M. Wooldridge, N.R. Jennings, D. Kinny – The GAIA Methodology for Agent-Oriented Analysis and Design, Journal of Autonomous Agents and MAS 3, 2000.