

Sinteza unui agent inteligent de timp real folosind g2

Conf.dr. Vasile MAZILESCU

Catedra de Contabilitate si Informatica Economica, Universitatea Dunarea de Jos din Galati

e-mail: mazilescugl@yahoo.com

An Intelligent Agent (InA) uses a knowledge base to interpret process outputs and references inputs, to reason about alternative control strategies, and to generate inputs to the process to improve the performance of the closed-loop system. It can be a specific agent. We compare in this paper the generality of the expert controller's knowledge representation and inference capabilities to that of other intelligent control systems, using G2 as software tool to realize particular simulations for a practical fuzzy balancing problem.

Keywords: intelligent agent, control.

Introducere

O problema importanta în vederea concepiei si simularii unui agent inteligent, notat InA, o reprezinta analogia dintre domeniul problemei si proces [1,2]. Un astfel de InA poate fi orice sistem bazat pe cunostinte care înglobeaza în structura sa de cunostinte un model invers al procesului. *Domeniul problemei* reprezinta multimea de probleme reale pe care sistemul de inteligenta artificiala trebuie sa le rezolve, utilizând o serie de rationamente, nodulate ca sisteme formale (automate sau bgici cuantificate, temporale, dinamice, multivaluate, nuartate etc.). Intrarile si iesirile sunt notate prin $u(t)$, $y(t)$ pentru varianta sistemelor de control conventionale si respectiv u_i , y_i , pentru sistemele de inteligenta artificiala. În raport cu intrarea $u(t)$ a procesului, intrarea u_i a domeniului problemei este o secventa de simboluri. De remarcat este faptul ca domeniul problemei nu poate fi descris întotdeauna prin ecuatii, astfel încât tehnicile conventionale de control nu sunt adecvate pentru sistemele de inteligenta artificiala.

Sistemele bazate pe cunostinte sau InA cu functionare în timp real poseda caracteristici pe care majoritatea sistemelor clasice nu le au: datarea cunostintelor, rationamentul asupra timpului, garantarea timpului de raspuns etc. Au fost concepute în acest sens astfel de sisteme, ale caror trasaturi le prezentam sintetic în continuare, folosind ca exemplu sistemul G2. Specificul generatorului G2 provine din capacitatile sale de con-

trol al unui proces în evolutie constanta, reactie la aparitia unor evenimente, aplicarea cunostintelor procedurale, reprezentarea aspectelor tranzitorii ale unei aplicatii, reprezentarea si utilizarea relatiilor dintre obiecte, achizitia unui mare numar de date de la surse exterioare (locale sau de la distanta), precum si furnizarea unor informatii si comunicarea cu alte sisteme G2.

Caracteristicile InA

Din cauza puternicelor similitudini dintre proces si domeniul problemei, putem evidientia o analogie între modelele utilizate pentru proces si domeniul problemei, pe de o parte, cât si între conceptele sistemice fundamentale, pe de alta parte. Procesul este descris prin ecuatii stohastice (posibil neliniare) sau diferentiale, numite ecuatii de stare si de iesire, care descriu dinamica procesului, structura si interconexiunile sale. Domeniul problemei poate fi descris prin ecuatii simbolice. Exista în continuare analogii matematice puternice între cele doua tipuri de sisteme, studiind anumite proprietati de importanta deosebita în teoria clasica a controlului [3]: controlabilitate, observabilitate, stabilitate, rata sistemului. Exista o analogie structurala între sistemele clasice de control si cele de inteligenta artificiala (planificare, sistem expert), atât în bucla deschisa cât si în bucla închisa.

Caracteristicile fundamentale ale unui InA, de tipul celui elaborat si prezentat în lucrarea de fata, sunt: a) InA poate fi considerat un model de

planificare al expertului uman în acțiune; **b) Starea** planificatorului reprezintă situația în care se află strategia de rezolvare a problemei la un moment dat, fiind alcătuită din informația necesară predicției comportării viitoare a sistemului. **c) Controlabilitatea** reprezintă abilitatea intrărilor de a modifica starea sistemului, considerând pentru analiză un sistem determinist. Domeniul problemei este *complet controlabil* la momentul de timp i , dacă există un moment de timp finit j , cu $j > i$, astfel încât pentru orice stare x_i și orice stare x_j , există o secvență finită de intrări u_1, \dots, u_j (pe care planificatorul le poate produce) care să transfere starea x_i în starea x_j la momentul j , identificată prin $x_i = x_j$. În cazul în care domeniul problemei este complet controlabil, atunci pentru orice stare există un planificator care realizează orice stare specifică, numită scop;

d) Observabilitatea domeniului problemei constă în abilitatea determinării stării acestuia, folosind intrările (secvență de acțiuni), ieșirile și modelul asociat; **e) Minimalitatea** reflectă cât de bine a fost modelat sistemul; **f) Stabilitatea** internă reflectă abilitatea de a rămâne într-o mulțime de stări (invariant) când nu există intrări scop și în prezența perturbațiilor [2].

Prezentăm în continuare problemele importante care pot fi evidențiate pentru sistemele în buclă deschisă. Dacă domeniul problemei este necontrolabil, atunci nu există un planificator capabil să rezolve problema, iar dacă este controlabil, existența unui astfel de planificator este asigurată. Evaluarea situației și monitorizarea execuției în cazul sistemelor în buclă deschisă nu pot fi asigurate, neexistând posibilitatea unei replanificări. În acest caz planificatorul este sensibil la variațiile care apar în domeniul problemei, planificatoarele în buclă deschisă neputând reduce această sensibilitate. Aceste planificatoare prezintă interes numai pentru perturbații nesemnificative, ceea ce nu este cazul problemelor reale, care necesită modele complexe și detaliate. Întrucât ieșirile sistemului nu sunt sesizate în intrare, dacă domeniul problemei este instabil (intrare-iesire sau intern), atunci el nu va fi niciodată stabilizabil în planificarea în buclă deschisă. Stabilizarea în buclă deschisă impune în mod absolut o cantitate

de cunoștințe exacte despre domeniul problemei. Cum acest lucru nu poate fi obținut, orice perturbare poate fi catastrofică. Planificatoarele în buclă deschisă prezintă avantaje din punct de vedere a simplității. Dacă domeniul problemei este stabil și perturbațiile sunt nesemnificative, atunci aceste planificatoare pot fi utile. Ele prezintă de asemenea avantajul unor costuri reduse, nefiind necesare măsurări ale stărilor și ieșirilor aferente domeniului problemei.

Sistemele de planificare în buclă închisă reprezintă analogul sistemelor convenționale de control în buclă închisă și de cele mai multe ori nu utilizează evaluarea situației [2]. Sunt permise în acest caz monitorizarea execuției și replanificarea. Domeniul problemei poate să nu fie complet controlabil fie din cauza elementelor de execuție care afectează serios starea domeniului, fie din cauza reconsiderării modelului aferent domeniului problemei, întrucât controlabilitatea este o proprietate a modelului matematic utilizat. Studiile de controlabilitate și observabilitate pot fi utile ca ghid în sinteza modelului pentru domeniul problemei. Evaluarea situației nu este necesară în planificare dacă întreaga stare a domeniului problemei este măsurabilă, această proprietate fiind analogul reacției totale pe stare. În cazul în care studiul observabilității evidențiază anumite stări neobservabile ale domeniului problemei, se pot implementa traductoare suplimentare care să furnizeze informațiile necesare de stare. Se remarcă în acest fel existența unui compromis între costurile implementării unui sistem de planificare și complexitatea acestuia. Poate fi mai scumpă soluția implicării mai multor traductoare pentru evaluarea întregii stări. În acest caz nu mai este necesară existența unui evaluator de situații, ceea ce conduce la o structură de planificator mai simplă. Dacă domeniul problemei este complet controlabil și observabil în raport cu intrările și ieșirile alese, atunci există un planificator în buclă închisă care stabilizează domeniul problemei, dacă acesta este instabil. Sistemul de planificare examinează diferența dintre situația de ieșire curentă și scopul dorit, în vederea executării anumitor acțiuni. Eroarea în sistemele de planificare de inteligență artificială sau în cazul sistemelor

expert nu este atât de ușor de obținut, în comparație cu sistemele clasice de control, întrucât distanța dintre simboluri este mult mai greu de cuantificat. Cu atât mai dificilă rămâne în cazul InA bazate pe cunoștințe imprecise evaluarea similitudinii dintre stări, în conformitate cu un criteriu dat. Sistemele de planificare pot avea la orice moment de timp funcția scop constantă (sisteme de planificare tip regulator) sau nu (sisteme de planificare cu urmărirea scopului).

Stabilitatea reprezintă o proprietate a sistemului în buclă închisă, valabilă și pentru cazul sistemelor de planificare bazate pe cunoștințe. Chiar dacă se gândește sistemul de planificare în buclă închisă cu abilitatea de a stabili orice sistem care este stabilizabil, el poate destabiliza domeniul specific al unei alte probleme. Este cazul de exemplu aplicării unui plan corect cu o anumită întârziere. Sistemele sunt destabilizate când se încearcă creșterea ratei sistemului peste o anumită valoare de siguranță, aceasta fiind o specificitate a sistemului în buclă închisă. Un avantaj major al sistemelor de planificare în buclă închisă în raport cu cele în buclă deschisă îl reprezintă abilitatea acestora de a rejecta perturbările care pot apărea în raport cu domeniul problemei, fiind nesensibile la variații ale domeniului problemei (atingerea unui obiectiv, chiar dacă modelul aferent domeniului problemei încorporează în mod obiectiv cunoștințe aproximative). În teoria sistemelor de control clasice, acest obiectiv se realizează folosind sisteme robuste.

Caracteristicile generatorului G2

Realizarea sistemelor de tip InA impune noi modalități de reprezentare a cunoștințelor, adecvate pentru diminuarea timpului de exploatare a lor. Inferența este întreruptă de evenimente externe asincrone, ce necesită prelucrarea unor date noi. Pentru tratarea a două sau mai multe evenimente cu aceeași prioritate, InA trebuie să fie capabil să execute inferențe în paralel. Multe din datele noi pot fi contradictorii cu cele deja existente. Din acest motiv este necesar ca baza de cunoștințe să nu aibă o construcție monolitică. Valorile provenite din mediu sau de la un anumit proces vor fi transformate în simboluri logice cu care se

operează. Aplicațiile realizate cu G2 utilizează un limbaj structurat într-un mediu grafic intuitiv de nivel înalt. Acest mediu permite conceptorului prototipizarea rapidă a aplicației sale și extinderea prototipului la un sistem complet de timp real. Conceptorul descrie fiecare clasă (sau tip) de obiecte corespunzătoare aplicației sale, reprezentarea sa grafică, atributele și caracteristicile specifice. Se creează astfel un model al părților permanente ale aplicației, plasând obiectele sale într-un spațiu de lucru (workspace). El poate de asemenea să conecteze aceste obiecte între ele. Marimile utilizate de sistem sunt de două tipuri: variabile și parametri. Parametrii sunt utilizați pentru valorile datelor cert valide, în timp ce variabilele sunt utilizate pentru valorile datelor evolutive. Este posibilă asocierea unei perioade de validitate atasată valorilor și de constituire a unui istoric cu vechile lor valori. Când validitatea valorii unei variabile s-a terminat, G2 caută o nouă valoare, fie cu ajutorul regulilor sau procedurilor, fie cu ajutorul unor formule de simulare. Regulile de producție sunt scrise într-un limbaj cvasinatural și un formalism de ordinul întâi. Ele sunt construite cu ajutorul unui editor contextual care ghidează conceptorul la fiecare pas. Conceptorul poate asocia fiecărei reguli o prioritate sau un mod de activare (înlanțuire înaintea, înapoi sau mixta, cu activare periodică). Formulele de simulare pot fi ecuații algebrice sau diferențiale de ordinul întâi. Este posibilă utilizarea formulelor de simulare pentru descrierea unor modele complexe (liniare sau nu).

G2 permite dezvoltarea și exploatarea aplicațiilor inteligente datorită unor caracteristici prezentate pe scurt în cele ce urmează.

Caracteristici de dezvoltare: a) *Un mediu de programare în limbaj natural.* Se folosește un limbaj natural structurat pentru exprimarea obiectelor, regulilor, modelelor și procedurilor în vederea capturării rapide a celor mai semnificative cunoștințe aferente aplicației; b) *Ingineria cunoștințelor.* Tot ceea ce se achiziționează și se dezvoltă în G2, se stochează în baze de cunoștințe. Aplicația poate încărca una sau mai multe baze de cunoștințe și toate bazele de cunoștințe pot partaja informații;

c) *Modelare si simulare dinamica*. Se pot modela procesele în mod dinamic, folosind obiectele specifice aplicatiei sub forma schematica sau diagrame. Utilizând abilitatile de rationament în raport cu obiectele conectate, se pot realiza simulari ale aplicatiei; **d) Abilitatea rapida si puternica de prototipizare**; **e) Dezvoltarea rapida si incrementală a aplicatiei folosind ciclul de viata**. Legarea icoanelor alese din biblioteca de obiecte permite crearea proiectului conceptual. Simularea si testarea subsistemelor are un impact deosebit asupra procesului de conceptie. Pe masura ce elementele hardware sunt disponibile, se înlocuiesc obiectele simulate în vederea validării performantelor sistemului.

Caracteristici de exploatare: **a) Portabilitatea aplicatiilor**. Pentru platforme noi nu este necesara recompilarea; **b) Arhitectura de timp real**. Aceasta permite ca mai multe reguli, proceduri si modele sa fie active în paralel, astfel încât aplicatiile pot urma multiple linii de rationament în timp real; **c) Modificarea interactiva a aplicatiilor fara nici un fel de întreruperi**; **d) Sporirea fiabilitatii aplicatiei on-line**. Acest lucru se realizeaza prin posibilitatea salvării stării curente a aplicatiei si istoricelor acesteia; **e) Accesul concurent la multiple surse de date si schimburi de date prin intermediul unei interfete standard G2**; **f) Partajarea concurenta a aplicatiilor G2 de catre utilizatori multipli**, prin intermediul produsului *Gensym Telewindows*; **h) Documentare extinsa on-line**, realizata cu ajutorul unui help senzitiv de context.

Aplicatiile elaborate cu ajutorul mediului de baza G2 si a produselor suplimentare (*GDA, NeurOn-Line, GSPC, ReThink, Fault Expert*), compatibile cu G2 si realizate de firma *Gensym* împreuna cu o serie de alte firme parteneri, pot fi utilizate pentru: **i) Monitorizarea, diagnoza si tratarea alarmelor** (*G2 Diagnostic Assistant – GDA*) ofera un mediu de programare pentru aplicatii inteligente de conducere si diagnoza a proceselor, help on-line, predictie si identificare, rezolvarea unor probleme care pot apare în proces, cu sprijinul sau nu al operatorului uman); **ii) Planificare dinamica si logis-**

tica; **iii) Optimizarea proceselor** (se utilizeaza G2 împreuna cu produsul *ReThink*, aplicabil în probleme de reinginerie, analiza, optimizarea si conducerea afacerilor complexe, în procese de productie); **iv) Supraveghere si control avansat** (se foloseste suplimentar produsul *NeuroOn-Line* ideal pentru validarea datelor provenite de la senzori, recunoasterea de forme, clasificarea defectelor, în aplicatii de control a calitatii); **v) Proiectarea si simularea proceselor (GSPC)**; **vi) Gestiunea inteligenta a retelelor (Fault Expert)**.

Mediul G2 este utilizat pentru o larga varietate de sisteme suport în luarea de decizii si în aplicatii de conducere, care sprijina marile firme în satisfacerea anumitor deziderate (cresterea productivitatii, utilizarea mai eficienta a resurselor, sporirea calitatii produselor si serviciilor, satisfacerea mai rapida a cererilor clientilor etc.) si flexibilizarea activitatii acestora în conditii de concurenta puternica. Sistemele de inteligenta artificiala realizate cu G2 contribuie serios la atingeria obiectivelor sus mentionate. G2 este un mediu complet orientat pe obiecte pentru dezvoltarea si exploatarea aplicatiilor de inteligenta artificiala. Spre deosebire de alte sisteme sau generatoare, mediul G2 faciliteaza diminuarea timpului de raspuns în elaborarea aplicatiilor, prin îmbunatatirea tuturor etapelor caracteristice ciclului de viata a unui proiect: de la conceptie, dezvoltare, simulare pâna la exploatarea si mentenanta sistemului. Mijloacele traditionale cum ar fi C, C⁺⁺ sau instrumentele CASE sunt utile în etapele de dezvoltare ale unui proiect. Cu cât proiectul avanseaza catre etapele finale cum ar fi testare, exploatare si mentenanta, productivitatea mediilor traditionale descreste rapid.

Caracteristica de baza a mediului G2 o reprezinta integrarea tehnologiilor necesare în dezvoltarea si exploatarea rapida a unei aplicatii de inteligenta artificiala: executie concurenta în timp real, proiectare orientata pe obiecte, grafica interactiva, reguli, proceduri si metode, limbaj natural structurat, modelare si simulare dinamica, procesare distribuita, conectivitate cu date on-line. G2 ofera un mediu de dezvoltare integrat, un nucleu

ce înglobează caracteristici puternice de timp real cu o proiectare orientată pe obiecte. Câteva din aceste caracteristici vor fi prezentate sumar în continuare.

Execuția concurentă în timp real. G2 este conceput în special pentru aplicații de timp real. Capacitățile puternice de timp real au fost realizate în mai mulți ani pe baza dezvoltării unor aplicații on-line. Nucleul de timp real al mediului G2 îl reprezintă un modul de planificare pentru executarea concurentă a regulilor, procedurilor, modelelor și altor task-uri. Mai multe linii de raționament, modele și alte activități sunt executate concurrent. Ciclurile motorului sunt de ordinul milisecundelor. Fiecare task are o prioritate, astfel încât activitățile critice sunt procesate corespunzător. Proprietățile de timp real ale mediului G2 includ: planificarea evenimentelor (ordin de milisecunde), planificare multiînlanțuită pentru execuția concurentă a regulilor, procedurilor și ale altor task-uri, istorice de date, procesarea de mii de reguli și proceduri pe secunda, datarea cunoștințelor, realizarea de raționamente într-un interval de timp impus și în lipsa unor date.

Tehnologia obiectuală. Proiectarea orientată pe obiecte constituie fundamentul dezvoltării în G2, oferind coduri de aplicație reutilizabile și o mai mare simplitate în înțelegerea structurii aplicației. Obiectele capturează proprietăți și metode, reprezintă obiecte fizice, putând fi grafice și interactive. O aplicație se dezvoltă sub G2 prin crearea diagramei de obiecte, organizată ierarhic în spațiul de lucru al lui G2 în care operează regulile și procedurile. Întrucât toate obiectele din G2 sunt definite ierarhic în clase, se pot scrie reguli și proceduri generice, aplicabile la orice nivel în cadrul ierarhiei, rezultând o aplicație bine structurată. Spațiul de lucru acționează ca un container în vederea păstrării componentelor aplicației, fiind utilizat tipic într-un mod care reflectă modularitatea aplicației. El apare grafic ca o fereastră, conține obiecte, reguli și alte elemente de date însoțite de caracteristicile funcționale ale acestora. Majoritatea partilor unei aplicații G2 se află în spațiul de lucru sau în conjuncție cu acesta. Spațiul de lucru este mai mult decât un container, fiind structurat ierarhic și reprezintă o

parte integrantă din cunoștințele aplicației. El este utilizat de asemenea în mod necondiționat pentru construirea interfeței grafice a aplicației. Conexiunile și relațiile sunt două modalități distincte pentru reprezentarea dependențelor dintre obiecte. Conexiunile sunt grafice iar relațiile descriptive. Abilitatea de reprezentare a dependențelor în G2 este un aspect important în vederea conceptualizării cunoștințelor în cadrul aplicației. Utilizarea variabilelor simbolice ca obiecte date și cu posibilități de păstrare a unor istorice, permite reprezentarea proprietăților de timp real.

Reguli, proceduri și metode. Acestea reprezintă o mare parte din cunoștințele specifice unei aplicații G2. Abilitatea de crea reguli, proceduri și metode generice, aplicabile tuturor claselor și obiectelor, diminuează timpul de dezvoltare a aplicației. Motorul de inferență din G2 invocă reguli, prin evaluarea antecedentului acestora. Trebuie determinate valorile atributelor referite de antecedentele regulilor. Un parametru are întotdeauna o valoare, a cărei durată de validitate dacă a expirat, G2 poate reactualiza această valoare de la diverse surse: server de date, utilizator, formule de simulare, reguli exploatate prin înlanțuire înapoi, formule. Motorul de inferență poate realiza următoarele probleme: scanarea regulilor la perioade determinate, focalizare asupra obiectelor prin invocarea regulilor asociate acestora, funcționarea înapoi pentru determinarea anumitor valori, înlanțuirea înainte în prezenta datelor necesare. G2 furnizează proceduri pentru îndeplinirea unor acțiuni secvențiale complexe și a unor calcule. Aceste proceduri diferă de reguli (care evidențiază doar modul de a răspunde la o situație dată), fiind ideale în aplicații de tipul procese secvențiale, planificarea evenimentelor, algoritmi de control complecsi, procese concurențiale, etc.

Limbaaj natural structurat. Se pot exprima enunțuri în mod natural folosind un editor senzitiv de context. Gensym este furnizorul principal de software și servicii pentru dezvoltarea și exploatarea sistemelor de inteligență artificială, care includ: monitorizare, diagnoză, controlul dinamic al evenimentelor în medii complexe și cu restricții puternice de timp, controlul calității în timp real,

gestiunea retelelor, planificare dinamica. Conducerea proceselor corespunde utilizarii intensive a cunostintelor. Aplicarea logicii sau "inteligentei" necesara pentru conducerea proceselor este în mod inerent un demers complex, care face dificila dezvoltarea unor astfel de aplicatii cu ajutorul produselor software traditionale. Produsul *Gensym G2 GDA* sprijina marile companii în elaborarea aplicatiilor de conducere inteligenta în mod simplu si rapid, pe toata durata ciclului de viata a acestuia, fiind un mediu integrat. Caracteristicile produsului *G2 GDA* se refera cu precadere la analiza operatiilor, conducerea în regim de supervizare si suport în luarea de decizii în timp real. Componentele grafice din *G2 GDA* au fost elaborate în vederea unei implementari rapide a unor functii de conducere, care fac apel la cantitati mari de cunostinte, pentru: corelarea tendintelor de evolutie a procesului si determinarea caracteristicilor de baza ale acestora, recunoasterea de forme, discriminarea evenimentelor semnificative din variatii aleatoare, anticiparea perturbatiilor si filtrarea informatiilor redundante, tranzitia între diferite strategii de control si multe altele.

Conducerea avansata a necesita sisteme suport în luarea de decizii inteligente, pe care metodele traditionale de tip *SCADA* sau *MMI* nu le pot satisface. Utilizatorii recunosc calitatile acestor sisteme din familia *G2*, care sprijina continuu strategiile de rezolvare a problemelor. Din acest motiv, *GDA* a contribuit la programele comune de calitate *ISO-9000*, *TQM*. *GDA* furnizeaza componente (grafice) bloc, predefinite, care pot comunica între ele date, stari logice sau obiecte. Componentele bloc includ filtre, functii matematice si statistice, serii de timp, generatoare de semnale utile în faza de prototipizare si încarcare a aplicatiei. Blocurile de inferenta sunt utilizate pentru detectarea caracteristicilor individuale sau a modelelor, pentru clasificarea si generarea unor concluzii ce pot genera secvente de actiuni sau explicatii si includ modele logice booleene, temporale, trivalente, fuzzy. Caracteristicile tehnice ale produsului *Gensym G2 GDA* sunt urmatoarele: **i) Statistici integrate pentru conducerea proceselor** (blocurile de statistici pot lucra

în extensie cu produsul *GSPC*); **ii) Arbori de decizie interactivi** (desi blocurile de inferenta pot în mod eficient clasifica diferitele situatii, exista uneori insuficiente date on-line pentru sinteza anumitor concluzii); **iii) Intrari manuale provenite de la operatorul uman** (exista definita o multime completa de blocuri care permit achizitionarea de valori numerice, logice sau text de la operator); **iv) Facilitati de explicatie**. Ele sunt folosite în cazul detectarii de evenimente importante. Explicatiile se bazeaza pe texte si pot fi realizate automat sau manual; **v) Reguli exprimate în limbaj natural**. Drept caracteristici avansate ale produsului *Gensym G2 GDA* mentionam în mod deosebit: arhitectura orientata pe obiecte, integrarea logicii fuzzy si a logicii temporale, integrarea retelelor neuronale.

Simularea unei baze de cunostinte în G2

În scopul evidentierii caracteristicilor inferentiale ale generatorului *G2*, am elaborat o serie de teste, utilizând ca proces un sistem flexibil de productie, similar cu cel din lucrarea [2]. Baza de reguli a fost rescrisa conform sintaxei generatorului *G2*, creându-se un spatiu de lucru al regulilor (*workspace*). Structura fiecărei reguli contine în clar variabilele de lucru pentru încarcarile momentane ale fiecarui subsistem, precum si elemente din structura vectorului de stare a *InA*, de tipul:

```
IF (the charge of m1>=the charge of
m2) and (the charge of m1>=the
charge of m3) and (the charge of
m1>=the charge of m4) and (the
charge of m1>=the charge of m5)
and (the charge of m1>=the charge
of m6) and (xb[1]/=0) and
(xb[3]/=0) and (the charge of
m1/=the charge of m2)
THEN in order conclude that xb[0]=2
and inform the operator on
INFER_NBR_FILE that "[the name of
this rule], xb1[xb[1]],
xb2[xb[2]], xb3[xb[3]],
xb4[xb[4]], xb5[xb[5]],
xb6[xb[6]], xb7[xb[7]],
xb8[xb[8]], xb9[xb[9]],
xb10[xb[10]]" and start
modify_charge (xb[0]) and conclude
that xb[1]=1 and conclude that
```

`xb[3]=1 and conclude that xb[5]=1
and conclude that xb[6]=1 and con-
clude that xb[8]=1 and conclude
that xb[9]=1 and conclude that
xb[10]=1 and conclude that xb[2]=0`

Transferul de pe un subsistem pe altul se realizeaza în cadrul modelului de conducere cu ajutorul procedurii *modify_charge(tip_eveniment)*, a carui parametru formal reprezinta evenimentul produs asupra procesului, adica asupra sistemului flexibil, reprezentat în regim de simulare prin modelul sau de cunostinte. Aceasta procedura este apelata odata cu selectarea si activarea unei reguli. Testele au fost realizate folosind diferite valori initiale si diferiti vectori de prioritate. Numarul de inferente, timpul de executie precum si încarcarile finale depind esential de starea initiala a sistemului si de strategia de selectie a regulilor, care este conforma cu posibilele strategii de conducere utilizate de om în vederea atingerii obiectivului propus.

Pentru cele zece teste prezentate mai jos, raspunsul sistemului InA elaborat apare sub forma unor siruri de reguli activate, care prezinta partial sau global anumite proprietati de periodicitate. În cazul testului zece, raspunsul sistemului este sub forma unei fractii periodice mixte, cu partea ne-periodica $n_1=7$ si lungimea partii periodice $n_2=5$. Rezulta ca aceasta problema de conducere este decidabila [3]. Semnificatia numarului k aflat între paranteze este ca grupul de inferente care apare ulterior neboldat se repeta de k ori. De exemplu secventa **(27) 2,3 5,4,2,3,10,9** indica faptul ca grupul de reguli (2,3) se repeta de 27 de ori, dupa care urmeaza în ordine regulile **5,4,2,3,10,9**.

Test 1 // time=real, infer_nbr=114, prioritati egale, timp_executie=2 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=100$, $m4=0$, $m5=40$, $m6=80$, încarcarile finale sunt $m1=m2=m4=72.917$, $m3=m5=68.542$, $m6=64.167$, prima regula activata este R5 // 5,4, **(27) 2,3 5,4,2,3,10,9, (25) 2,3, 2,4**

Test 2 // time=fast, infer_nbr=920, prioritati egale, timp_executie=26 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=$

$m4=m5=0$, $m6=420$, încarcarile finale sunt $m1=52.5$, $m2=m3=m4=m5=m6=73.5$, prima regula activata este R10 // **10,9,6,3,7,(27)10,8,6,(26)10,8,7,6,(26)10,8,7,6,(25)10,8,7,6,(24)10,8,10,7,6,(24)10,8,7,6,10,8,4,8,5,(21)10,8,10,7,6,10,8,4,8,5,(21)10,8,7,6,10,8,4,8,5,(19)10,8,7,6,10,8,4,8,5,(19)10,8,7,6,10,8,4,8,5,(18)10,8,10,7,6,10,8,4,8,5,(18)10,8,7,6,10,8,4,8,5,(17)10,8,7,6,10,8,4,8,5,(17)10,8,7,6,10,8,4,8,5,(16)10,8,7,6,10,8,4,8,5,(15)10,8,7,6,10,8,4,8,5,(13)10,8,10,7,6,10,8,4,8,5,(14)10,8,7,6,10,8,4,8,5,(13)10,8,7,6,10,8,4,8,5,(6)10,8...// oprit.**

Test 3 // time=fast, infer_nbr=723, prioritati egale, timp_executie=20 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=m4=m5=0$, $m6=420$, încarcarile finale sunt $m1=0$, $m2=m3=m4=m5=m6=84.00$, prima regula activata este R9 // **9, (27) 10,8, 10, 6, (27) 10,8, 7, 6, (27) 10,8, 7,6, (27) 10,8, 7,6, (26) 10,8, 7,6, (26) 10,8, 7, 6, (25) 10,8, 7, 6, 10, 8, 4, 8, 5, (22) 10,8, 10, 7, 6, 10, 8, 4, 8, 5, (22) 10,8,7, 6,10, 8, 4, 8, 5,(20)10,8,10,7,6,10,8,4,8, 5, (20) 10,8,7,6,10, 8,4, 8,5,(20)10,8, 7, 6, 10, 8, 4, 8, 5 (19) 10,8, 10, 7, 6, 10, 8, 4, 8, 5, (19) 10,8, 7, 6, 10, 8, 4, 8, 5, (18) 10,8, 7, 6, 10, 8, 4, 8, 5, (17) 10,8, 10, 7, 6, 10, 8, 4, 8, 5, (17) 10,8, 7, 6, 10, 8, 4, 8, 5, (13) 10,8 ...// oprit.**

Test 4 // time=fast, infer_nbr = 867, prioritati egale, timp_executie =16 minute, încarcarile initiale ale subsistemelor sunt $m1 = m2 = m3 = m4=m5=0$, $m6=420$, încarcarile finale sunt $m1 = m3=0.0$, $m2=m4=m5=m6=105.0$, prima regula activata este R9 // **9,(27)10,8,10,6,(27)10,8,6,(25)10,8,10,6,(25)10,8,10,6,(25) 10, 8,6,(24)10,8,6,(24)10, 8, 10, 6, (21)10, 8, 10, 6, (21)10,8,10,6,(20)10,8,10,6,(20)10,8,6,(19)10, 8, 6, (18) 10,8, 10, 6, (17) 10,8, 10, 6, (16) 10,8, 6, (16) 10,8, 6, (15) 10,8, 10, 6, (14) 10,8, 10, 6, (13) 10,8, 10, 6, (12) 10,8, 6, (12) 10,8, 6, (6) 10,8 ... // oprit.**

Test 5 // time=fast, infer_nbr=859, vector prioritati p1, timp_executie=15 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=m4=m5=0$, $m6=420$, încarcarile finale sunt $m1=m3=0.0$, $m2=m4=m5=m6=105.0$, prima regula activata este **R9** // 9,(27),10,8,10,6, (27)10, 8, 6, (25)10, 8, 10, 6, (25)10, 8,10,6, (25)10,8,6,(24)10,8,6, (23)10,8,10,6, (21)10,8, 10,6,(21)10,8,10,6,(20)10,8, 10, 6, (20)10, 8, 6, (19)10,8,6,(18)10,8,10,6,(17)10,8,10,6,(16)10, 8, 6,(16)10,8,6, (15) 10,8, 10, 6, (14) 10,8, 10, 6, (13) 10,8, 10, 6, (12) 10,8, 6, (12) 10,8, 6, 10,8,10,8 ... // oprit.

Test 6 // time=fast, infer_nbr = 210, vector prioritati p1, timp_executie =3 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=00$, $m4=0$, $m5=40$, $m6=80$, încarcarile finale sunt $m1=m2=75.309$, $m3=m5=67.654$, $m4=74.074$, $m6=60.0$, prima regula activata este **R5** // 5, (26) 2,3, 5,2,3,9, (25) 2,3, 5, (25) 2,3, 5, (24) 2,3, 2, 4.

Test 7//time=fast, infer_nbr=159, vector prioritati p2, timp_executie=1 minut, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=100$, $m4=0$, $m5=40$, $m6=80$, încarcarile finale sunt $m1=m2=77.037$, $m3=m5=58.519$, $m4=75.556$, $m6=73.333$, prima regula activata este **R5** // 5, (26) 2,3, 5,2,3,10, (25) 2,3, 5, (25) 2,3, 2,4.

Test 8 // time=fast, infer_nbr=579, prioritati p3, timp_executie=10 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=100$, $m4=0$, $m5=40$, $m6=80$, încarcarile finale sunt $m1=m2=m3=m4=m5=70.292$, $m6=68.542$, prima regula activata este **R5** // 5,4, (27) 2,3, 5, 4, 2, 3, 10, (25) 2,3, 2,4, 9, 2, 3, 6, 7, 3, (24) 2,3, 5, 4, 2, 3, 2, 3, 6, 7, 3, (21) 2,3, 5, 4, 2, 3, 6, 7, 3, (22) 2,3, 5, 4, 2, 3, 6, 7, 3, (20) 2,3, 2, 5, 4, 2, 3, 6, 7, 3, (20) 2,3, 5, 4, 2, 3, 6, 7, 3, (18) 2,3, 2, 5, 4, 2, 3, 6, 7, 3, (18) 2,3, 2, 5, 4, 2, 3, 6, 7, 3, (18) 2,3, 5, 4, 2, 3, 6, 7, 3, (13) 2,3, ... // oprit.

Test 9 // // time=fast, infer_nbr=632, vector prioritati p3, timp_executie=10 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=m3=m4=m5=0$, $m6=420$, încarcarile finale sunt $m1=m2=m3=m4=m5=70$, $m6=69.998$, prima

regula activata este **R10** // 10,6,10,7, 6,10,3,7,6,10,4,7,6,10,7,6,3,10,7,6,10,5,4,7,6, 3, 10, (25) 2,3, 5, 4, 2, 3, 7, 6, 10, 7, 6, 3, 10, (21) 2,3, 2, 5, 4, 2, 3, 7, 6, 10, 7, 6, 10, 7, 6, 10, 5, 4, 7, 6, 3, 2, 3, 10, (22) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (22) 2,3, 5, 4, 2, 3, 7, 6, 3, (21) 2,3, 5, 4, 2, 3, 7, 6, 3, (19) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (19) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (18) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (18) 2,3, 5, 4, 2, 3, 7, 6, 3, (17) 2,3, 5, 4, 2, 3, 7, 6, 3, (16) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (15) 2,3, 2, 5, 4, 2, 3, 7, 6, 3, (14) 2,3, 5 // oprit.

Test 10 // time=fast, infer_nbr=129, vector prioritati p3, timp_executie=0.5 minute, încarcarile initiale ale subsistemelor sunt $m1=m2=0.0$, $m3=420$, $m4=m5=m6=0$, încarcarile finale sunt $m1=m2=m3=m4=m5=m6=70$, prima regula activata este **R5** // 5, 4, 6, 3, 8, 6, 4, (5,3,8,6,4).

Concluzii

Testele elaborate cu ajutorul generatorului \mathcal{G} ne-au evidentiat pe de o parte comportarea satisfacatoare a modelului de cunostinte pentru sistemul flexibil de productie, iar pe de alta parte au confirmat limitele utilizarii unui generator, în conditiile în care nevoia de modelare a rationamentului si a cunoștințelor specifice dintr-un domeniu sunt diferite fata de posibilitatile conceptuale ale generatorului. În plus, pentru o formula scop, rezulta ca putem sintetiza un InA de timp real care sa permita ca formula sa fie valida în timp polinomial. Acesta este un aspect deosebit de important privind decidabilitatea conducerii, bazata pe integrarea unor caracteristici logice temporale. În scopul realizarii unor sisteme de sprijin în luarea de decizii este necesara elaborarea de sisteme de inteligenta artificiala, care sa permita reprezentarea si exploatarea unor cunostinte ce depind esential de factorul timp. Frecvent, si mai ales în aplicatiile de timp real, trebuie avuta în vedere relatia dintre un eveniment si timpul sau de aparitie, sau relatia temporală dintre cauza si efect, din perspectiva logicilor temporale. Imprecizia cunoștințelor se poate datora unei lipse de informatie despre evenimentele trecute sau viitoare, fiind adesea dificil de determinat precis valorile anumitor marimi. Aceste

observatii au intervenit esential în conceptia sistemului InA, sau a oricarui agent dintr-o structura multiagent.

Bibliografie

1. Gupta M. Madan, Sinha K. Naresh, 1996 – *Intelligent Control Systems / Theory and Applications*, IEEE PRESS, ISBN 0-7803-1063-2
2. Mazilescu V., 1998 - *Fuzzy real time expert control systems. Characteristics and semantic of fuzzy compiled models*, Applications of Artificial Intelligence in Industrial Automation, (Tempus Intensive Course Postprints), Ed. Academica Publishing, May 22-27, ISBN 973-97816-8-3, p. (26) 1-15, from TEMPUS M-JEP 11467/96-“COMPANION”, May 22-27
3. Mazilescu V., Mânzu V., Georgescu C., 2000 - *Algorithm for tasks - to - workstations assignment in single or mixed-model assemble lines*, 2nd CIRP International Seminar on Intelligent Computation in Engineering - ICME 2000, 21-23 June, Capri, Italy, p. 245-250