



ŞCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea: Inginerie Electrică și Știința Calculatoarelor

Anata-Flavia IONESCU

**ORGANIZAȚIILE VIRTUALE CA
FACILITATORI AI COLABORĂRII ÎNTRE
MEDIUL ACADEMIC ȘI MEDIUL
ECONOMIC**

REZUMAT / ABSTRACT

Conducători științifici

Prof. dr. Dorin-Mircea POPOVICI

Prof. dr. Raluca-Ileana VERNIC

BRAȘOV, 2022

D-lui (D-nei)

COMPONENȚA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Braşov

Nr. din

PREȘEDINTE: Prof. dr. Mihai IVANOVICI, Universitatea Transilvania din Braşov

CONDUCĂTORI ȘTIINȚIFICI:

Prof. dr. Dorin-Mircea POPOVICI, Universitatea Transilvania din Braşov, Universitatea Ovidius
din Constanța

Prof. dr. Raluca-Ileana VERNIC, Universitatea Ovidius din Constanța

REFERENȚI:

Prof. dr. Luca-Dan ȘERBĂNAȚI, Universitatea Politehnică București

Prof. dr. Dorian GORGAN, Universitatea Tehnică din Cluj-Napoca

Prof. dr. Sorin MORARU, Universitatea Transilvania din Braşov

Data, ora și locul susținerii publice a tezei de doctorat:, ora, sala

Eventualele aprecieri sau observații asupra conținutului lucrării vor fi transmise electronic, în timp util, pe adresa anata.ionescu@unitbv.ro.

Totodată, vă invităm să luați parte la ședința publică de susținere a tezei de doctorat.

Vă mulțumim.

CUPRINS

	Pg.	Pg.
	Teză	rezumat
LISTA DE ABREVIERI	10	5
1. INTRODUCERE	11	6
2. SCOP ŞI OBIECTIVE	29	9
3. METODOLOGIA CERCETĂRII	31	11
3.1 Preceptor: O arhitectură de sistem multiagent holonic propusă pentru organizaţii virtuale de instruire.....	33	11
3.1.1 Formularea problemei.....	33	12
3.1.2 Soluţia propusă.....	34	13
3.2 Studiu de caz.....	58	24
3.2.1 Ipoteze.....	65	28
3.2.2 Setări experimentale.....	66	28
4. REZULTATE	67	29
4.1 Abordarea propusă pentru problema de selecţie a partenerilor: Algoritmul Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS).....	67	29
4.2 Studiu de caz: Răspuns la o cerere de organizaţie virtuală de instruire folosind arhitectura Preceptor.....	72	31
5. DISCUŢII	75	34
5.1 Utilitate.....	76	34
5.2 Contribuţii originale.....	76	35
5.3 Limitări.....	77	35
5.4 Direcţii viitoare de cercetare.....	78	36



5.5 Diseminarea rezultatelor cercetării.....	83	37
6. CONCLUZII	85	39
BIBLIOGRAFIE	87	40
ANEXE	97	
Anexa 1, Generator de instanțe aleatorii ale problemei de selecție a	97	
partenerilor		
Anexa 2, Algoritm pentru determinarea holarhiei competențelor pentru	98	
un cursant		
Anexa 3, Mulțimea soluțiilor (S) și mulțimile Pareto folosite în	101	
experimente		
Anexa 4, Boxploturi pentru indicatorii de performanță utilizați în	103	
experimente		
Scurt rezumat	108	42

Lista Abrevierilor

ABM	Agent-Based Modeling
IA	Inteligență Artificială
CH	Competence Holon (Holon Competență)
CPU	Central Processing Unit (Unitate Centrală de Procesare)
HMAS	Holonic Multiagent System (Sistem Multiagent Holonic)
ISP	Instructor Selection Problem (Problema de Selecție a Instructorilor)
MAS	Multiagent System (Sistem Multiagent)
MOEA	Multiobjective Evolutionary Algorithm (Algoritm Evolutiv Multi-Obiectiv)
MOP	Multiobjective Optimization Problem (Problemă de Optimizare Multi-Obiectiv)
PSP	Partner Selection Problem (Problema de Selecție a Partenerilor)
SH	Scheduling Holon (Holon de Programare)
UML	Unified Modeling Language
VE	Virtual Enterprise (Întreprindere Virtuală)
VO	Virtual Organization (Organizație Virtuală)

Capitolul 1

Introducere

În zilele noastre, chiar și pentru învățământul terțiar este dificil să țină pasul cu cerințele din ce în ce mai specializate ale pieței muncii. Pot fi adoptate soluții precum instruirea la locul de muncă, dar aceste abordări au riscuri asociate mari (de exemplu, daune materiale) [6] și pot duce la o scădere temporară a performanței grupului de lucru [2].

O tendință actuală notabilă în învățarea virtuală este personalizarea evaluării [vezi, de exemplu 13], a planului de instruire [de ex. 5] și chiar a conținuturilor [de ex. 4], prin tehnici de inteligență artificială (IA), cum ar fi arbori de decizie, algoritmi genetici, logica fuzzy, rețele bayesiene, rețele neuronale sau modele Markov ascunse [1]. Cu toate acestea, automatizarea puternică a instruirii a fost privită cu rezerve, iar mulți cercetători în domeniul învățării virtuale și-au reorientat atenția către facilitarea colaborării în timp real între cursanți și tutori prin tehnologii precum clasele virtuale sau tabla albă [9]. În plus, politicile UNESCO descurajează înlocuirea profesorilor umani cu IA, deoarece au fost evidențiate efecte negative asupra agenției, motivației, memoriei etc. În schimb, se subliniază nevoia de a redefini rolurile profesorilor ca utilizatori ai tehnicilor IA în activitatea lor didactică [22].

Aceste argumente ne-au condus la concluzia că, în loc să înlocuim instructorii cu IA, o abordare mai eficientă ar fi folosirea IA pentru a-i conecta la cursanți, adunând echipe de instructori optimizate pentru a oferi instruire la cerere. O astfel de abordare poate fi implementată ca o organizație virtuală (VO).

VO sunt alianțe temporare de organizații dispersate geografic, inițiate cu scopul de a servi unui anumit scop [3]). Ideea de a folosi un VO pentru predare datează de mai bine de două decenii (vezi [16]), dar implementările sunt încă rare și, de obicei, se concentrează pe un anumit curriculum. Din cunoștințele noastre, cercetarea existentă în domeniul VO pentru instruire nu abordează încă optimizarea structurii VO ca răspuns la solicitări ad-hoc, adică rezolvarea problemei de selecție a partenerilor (PSP). PSP este un caz special de problemă de optimizare multi-obiectiv (MOP). Presupunem un scenariu tipic de PSP: inițiatorul unui proiect organizează o licitație pentru n activități pe care nu este dispus/capabil să le efectueze intern, căutând să optimizeze combinația de oferte. Tabelul 1.1 rezumă notațiile pe care le folosim



pentru PSP în teză.

Tabela 1.1: Notății.

Notăție	Definiție
n	numărul de activități ale proiectului
A	mulțimea de activități ale proiectului
P	mulțimea de arce care indică constrângerile de precedență între activitățile proiectului
m_i	numărul de oferte ale partenerilor plasate pentru activitatea $i \in A$ din proiect
C_{ij}	costul ofertei partenerului $j \in \{1, \dots, m_i\}$ pentru activitatea $i \in A$
Q_{ij}	calitatea ofertei partenerului $j \in \{1, \dots, m_i\}$ pentru activitatea $i \in A$
D_{ij}	durata ofertei $j \in \{1, \dots, m_i\}$ pentru activitatea $i \in A$
TC	costul total al proiectului
AQ	calitatea medie a proiectului
TD	durata totală a proiectului
$StAv_{ij}$	ora de începere a ferestrei de disponibilitate a ofertei partenerului $j \in \{1, \dots, m_i\}$ pentru activitatea $i \in A$
$EndAv_{ij}$	ora de încheiere a ferestrei de disponibilitate a ofertei partenerului $j \in \{1, \dots, m_i\}$ pentru activitatea $i \in A$
ST_i	ora programată de începere a activității $i \in A$
R	mulțimea tuturor resurselor necesare pentru obiectivele proiectului
RR_{ir}	cantitatea din resursa $r \in R$ cerută de activitatea $i \in A$
RC_r	capacitatea resursei $r \in R$

Vom lua în considerare următoarea formulare a PSP multiobiectiv:

$$\min TC = \sum_{i=1}^n \sum_{j=1}^{m_i} \beta_i^{(j)} C_{ij} \quad (1.1)$$

$$\max AQ = \frac{\sum_{i=1}^n \sum_{j=1}^{m_i} \beta_i^{(j)} Q_{ij}}{n} \quad (1.2)$$

$$\min TD = \max_{\substack{i \in A, \\ j \in \{1, \dots, m_i\}}} \{ST_i + \beta_i^{(j)} D_{ij}\} \quad (1.3)$$

$$\text{cu: } \sum_{j=1}^{m_i} \beta_i^{(j)} = 1, \forall i \in A \quad (1.4)$$

$$ST_i + \beta_i^{(j)} D_{ij} \leq \text{Deadline}, \quad \forall i \in A, j \in \{1, \dots, m_i\} \quad (1.5)$$

$$ST_i + \beta_i^{(j)} D_{ij} \leq ST_k, \quad (1.6)$$

$$\forall i \in A, k \in A \text{ s.t. } (i, k) \in P, j \in \{1, \dots, m_i\}$$

$$\sum_{i=1}^n \sum_{j=1}^{m_i} \beta_i^{(j)} C_{ij} \leq Budget \quad (1.7)$$

$$ST_i \geq StAv_{ij}, \forall i \in A, \quad (1.8)$$

$$j \in \{1, \dots, m_i\} \text{ s.t. } \beta_i^{(j)} = 1$$

$$ST_i + D_{ij} \leq EndAv_{ij}, \forall i \in A, \quad (1.9)$$

$$j \in \{1, \dots, m_i\} \text{ s.t. } \beta_i^{(j)} = 1$$

$$\sum_{i=1}^n a_i^t RR_{ir} \leq RC_r, \forall r \in R, \quad (1.10)$$

$$t \in \{1, \dots, Deadline\}$$

unde

$$\beta_i^{(j)} = \begin{cases} 1, & \text{dacă oferta } j \text{ este selectată pentru activitatea } i \\ 0, & \text{altfel} \end{cases}$$

$$a_i^t = \begin{cases} 1, & \text{dacă activitatea } i \text{ este în derulare la momentul } t \\ 0, & \text{altfel} \end{cases}$$

În formularea de mai sus, ecuațiile (1.1)–(1.3) definesc criteriile/obiectivele. Ecuația (1.1) minimizează costul total al proiectului. Ecuația (1.2) maximizează calitatea medie a serviciilor de instruire oferite de partenerii selectați. Ecuația (1.3) minimizează durata proiectului. Constrângerea (1.4) asigură că este selectată o singură ofertă pentru fiecare activitate din proiect. Constrângerea (1.5) asigură că nicio oră de încheiere a activității nu depășește termenul limită al proiect. Constrângerea (1.6) impune relațiile de precedență între activități. Constrângerea (1.7) garantează că costul total al proiectului nu depășește bugetul alocat. Constrângerile (1.8) și (1.9) impun ca activitățile să aibă loc în ferestrele de timp impuse de ofertele selectate. Constrângerea (1.10) asigură nedepășirea în niciun moment al proiectului, a capacității oricărei resurse.

Au fost propuse multe abordări pentru rezolvarea MOP-urilor (vezi [11]) și a PSP. În prezent, una dintre abordările considerate cele mai potrivite sunt euristiciile de căutare globală (bazată pe populație) [14]. Descriem adaptarea propusă de noi a unei meta-euristici recente, un algoritm pe care l-am numit Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS), în secțiunea 4.1.

În plus, propunem o arhitectură pentru un sistem multiagent (MAS) holonic care să susțină întregul ciclu de viață al unei organizații virtuale (VO) [3] de instruire. Alegerea noastră de a modela sistemul ca MAS a fost motivată de caracteristici VO precum natura descentralizată, orientarea către obiective, formarea bazată pe licitație, decizii bazate pe fluxul de comunicare și informații, procese de negociere etc. [12], [20] Cum Peña et al. [19] susțin utilizarea holonilor ca elemente fundamentale pentru teoria științei informatice, am modelat atât piața ca holon și VO-urile ca subholoni. Din cunoștințele noastre, aceasta este prima lucrare care abordează predarea colaborativă prin implementarea unui VO ca MAS holonic (HMAS) [7].

Capitolul 2

Scop și obiective

În această teză urmărim să propunem o arhitectură pentru o platformă capabilă să susțină întregul ciclu de viață al unui VO de instruire (formare). În acest scop, acordăm o atenție deosebită PSP, care este o problemă cheie care trebuie rezolvată în două din cele patru faze ale ciclului de viață VO: formarea și reconfigurarea. Implementarea managementului concurrent al mai multor procese de rezolvare a PSP este un obiectiv important al cercetărilor noastre. Mai mult, ne ocupăm și de operarea și dizolvarea VO, prin proiectarea de coordonare și a unui sistem de rating.

Abordarea noastră este să pornim de la o formulare PSP destul de generală și să dezvoltăm mai întâi un algoritm generic de rezolvare a PSP. Avantajul unei astfel de abordări este că poate fi aplicată la un set mai larg de probleme, nu numai la problemele enunțate în această lucrare. De dorit ar fi, totuși, ca soluția generală să fie ușor de adaptat la probleme mai specifice, iar această adaptabilitate este, de asemenea, parte din scopul nostru. Ca atare, am ales să dezvoltăm un algoritm euristic care nu necesită niciun tip specific de funcții obiectiv sau o geometrie specifică a frontului Pareto.

Mai mult, gestionarea concomitentă a cererilor de formare VO este parte din scopul nostru. În primul rând, abordăm gestionarea simultană a cererilor prin crearea, pentru fiecare cerere individuală, a unui agent software holic dedicat, însărcinat cu gestionarea licitației și a altor mecanisme de negociere în contextul său interior. În al doilea rând, după ce toate ofertele sunt colectate și se poate calcula mulțimea Pareto peste mulțimea de combinații de oferte, mulțimea Pareto poate conține mai mult de o soluție (sub)optimală. Aceasta înseamnă că este nevoie de sprijin suplimentar pentru procesul uman de luare a deciziilor. Ne propunem să rezolvăm acest lucru prin crearea unui spațiu separat de negociere pentru fiecare potențial VO de instruire pe care cursantul îl poate alege, permițând negocierea concomitentă.

Valorificăm în continuare cadrul holic pentru a proiecta celelalte faze ale ciclului de viață al VO de instruire. În timpul funcționării VO, scopul nostru este de a sprijini colaborarea instructorilor pentru planificarea personalizată a curriculumului prin modelarea instructorilor ca superholoni pentru instructorii responsabili de cerințele preliminare de care au nevoie. Pentru a încheia ciclul și a facilita procesele ul-



terioare de formare a VO, ne referim, de asemenea, la evaluarea calității serviciilor oferite instructorilor la dizolvarea VO. În plus, ne propunem să proiectăm platforma astfel încât instructorii care devin membri ai unuia sau mai multor VO de instruire să își mențină relațiile inițiale competitive în mediul de piață educațional din care fac parte. Principiul principal de proiectare pe care îl propunem în acest scop este modelarea VO de instruire ca subholoni ai pieței educației.

Prin urmare, derivăm următoarele obiective specifice:

1. O1 — propunerea și implementarea unei abordări noi pentru rezolvarea PSP cu programarea activităților în funcție de timp, buget, precedența activităților și constrângeri de capacitate de resurse
2. O2 — implementarea și testarea abordării propuse pe instanțe aleatorii PSP, simulând o licitație pentru formarea VO
3. O3 — proiectarea și implementarea unei arhitecturi HMAS pentru un mediu VO de instruire permițând gestionarea concomitentă a solicitărilor de instruire prin configurarea și suportul pentru operarea, reconfigurarea și dizolvarea VO
4. O4 — realizarea unui experiment pentru a compara eficacitatea a trei structuri organizatorice diferite ale HMAS în timpul procesului de programare a activității de formare.

Restul tezei este organizat după cum urmează. Metodologia pe care o folosim pentru atingerea obiectivelor este descrisă în capitolul 3. Rezultatele corespunzătoare sunt raportate în capitolul 4. O discuție extinsă a acestor rezultate este prezentată în capitolul 5. Capitolul 6 încheie teza.

Capitolul 3

Metodologia cercetării

Am dezvoltat abordarea vizată de obiectivul O1 pe baza unei sinteze a avantajelor și dezavantajelor teoretizate și/sau demonstrate empiric ale abordărilor PSP existente. Abordarea noastră propusă, detaliată în secțiunea 4.1, este o adaptare a uneia dintre euristicile recente a cărei eficiență în rezolvarea problemelor similare cu PSP-ul pe care ne concentrăm a fost demonstrată empiric în mai multe studii.

Pentru a atinge obiectivul O2, am implementat algoritmul propus și algoritmi concurenți în Java, ca o extensie a framework-ului jMetal, versiunea 5 [18]. Pentru a compara algoritmi, am simulat scenarii de formare a VO prin generarea aleatorie a instanțelor PSP. Algoritmul pe care l-am folosit pentru generarea de instanțe aleatorii PSP este prezentat în Anexa 1. Apoi am efectuat experimente numerice comparând algoritmi concurenți în raport cu un set de metrici de performanță pentru optimalitate în sens Pareto. Aceste experimente, precum și experimentele pentru obiectivul O4, au fost efectuate pe un computer Acer cu memorie DDR3L 1600 MHz de 16 GB și un procesor Intel Core i5 de 2,6 GHz care rulează sistemul de operare Windows 10 Pro.

În urmărirea obiectivului O3, am proiectat mai întâi un mediu VO de instruire bazat pe agenți folosind diagrame UML de cazuri de utilizare, de secvențe și de clasă. Designul este descris în secțiunea următoare.

3.1 Preceptor: O arhitectură de sistem multiagent holonic propusă pentru organizații virtuale de instruire

În această secțiune, formulăm o problemă specifică de învățare virtuală, o încadrăm ca un caz special de PSP și propunem o soluție bazată pe o abordare HMAS, apoi descriem proiectarea acesteia.



3.1.1 Formularea problemei

Să presupunem că o persoană care dorește o pregătire specifică (de exemplu, un candidat pentru un loc de muncă) necesită pregătire pentru una sau mai multe competențe pe care nu le stăpânește. Persoana (pe care o vom numi **Cursant**) caută un furnizor de servicii virtuale optimal, dată fiind o mulțime de furnizori de instruire (pe care îi vom numi **Instructori**) care activează pe o piață (pe care o vom numi **Piața educației**), pentru a acoperi toate nevoile specifice de formare ale cursantului. Cursantul stabilește două criterii pentru optimalitate: cost și calitate și emite un tuplu $VORequest \in \mathbb{N} \times 2^C \times \mathbb{R}_0^+ \times \mathbb{N}$, unde C reprezintă mulțimea tuturor competențelor listate pe piață. Tuplul conține numărul de identificare al cursantului, competențele care necesită instruire ($A \subseteq C$), bugetul și termenul limită. Apar constrângeri suplimentare de timp, deoarece unele dintre competențele necesare pot fi prerechizite pentru altele, ceea ce înseamnă că pregătirea pentru acele competențe ar trebui să fie terminată înainte de a începe formarea pentru competențele care depind de ele.

În restul tezei, ne vom referi la problema descrisă mai sus ca **Problema de selecție a instructorilor (ISP)**. ISP poate fi modelată ca o PSP cu următoarele două obiective care urmează să fie optimizate simultan: (1) costul total, de minimizat; (2) calitatea medie a serviciilor de formare care compun VO, de maximizat. Soluțiile sunt supuse unui termen limită și unei constrângeri de buget, sunt compuse dintr-un număr de n activități de formare a competențelor pentru care ar trebui aleasă o ofertă a unui Instructor, iar următoarele caracteristici ale ofertei sunt considerate variabile de decizie: costuri, evaluări ale calității, momentele de început și sfârșit ale ferestrei de disponibilitate și durata serviciului de instruire. Toate activitățile de instruire necesită o singură resursă: timpul (numărul de ore pe zi pe care cursantul este dispus să le aloce activităților din VO). Relațiile de prerechizită între competențe induc constrângeri de precedență. Prin urmare, problema este un caz particular al PSP definit în introducere, obținut prin eliminarea criteriului de timp descris de ecuația 1.3 și reducerea mulțimii de resurse R la o singură resursă.

A doua problemă pe care o abordăm este una pe care o numim **Problemă a strategiei pentru prerechizită**. Presupunem că fiecare competență $c \in C$ este caracterizată de o mulțime de strategii care pot fi aplicate în instruirea ei. Strategiile de instruire pot diferi în ceea ce privește selecția conținutului și/sau secvențierea, precum și în ceea ce privește selecția metodelor de instruire. O altă presupunere pe care o facem este că formarea unei competențe c_1 ca o prerechizită a unei alte competențe c_2 poate necesita o abordare (strategie) diferită față de formarea aceleiași competențe per se, sau ca o prerechizită a c_3 etc.

Fie $P_c \subseteq C \setminus \{c\}$ mulțimea de prerechizite pentru competența $c \in C$ și S_p mulțimea de strategii de instruire pentru prerechizita $p \in P_c$. Atunci, $\forall c \in C, p \in P_c$ definim o funcție de succes $success_{c,p} : S_p \rightarrow [0, 1]$, care asociază fiecare strategie de formare a lui p ca prerechizită pentru c cu o evaluare a succesului. Cu cât evaluarea este mai bună, cu atât strategia este mai reușită. Aceste evaluări pot fi rezultatul diferitelor metode de agregare aplicate evaluărilor furnizate de experți în pedagogie, dar acest lucru depășește scopul acestei teze.



3.1.2 Soluția propusă

Pentru a rezolva ISP și problema strategiei pentru prerechizită, propunem o arhitectură pentru un mediu VO pe care l-am numit Preceptor. Arhitectura propusă acceptă două categorii de utilizatori: Cursanți și Instructori, care folosesc sistemul conform diagramei de cazuri de utilizare din figura 3.1. Mediul VO propus este o implementare holică a Pieței Educației, pentru crearea la cerere a VO-urilor de formare ca subholoni. Preceptor poate fi folosit pentru a stabili planul de instruire personalizat al Cursantului folosind adaptarea propusă de noi a algoritmului Breadth first search [17] pentru prerechizite ale competențelor solicitate (pseudocodul este listat în Anexa 2) și asigură toate fazele ciclului de viață VO: formare, operare, reconfigurare și dizolvare [3].

După stabilirea mulțimii A de competențe care necesită instruire, Cursantul poate emite o Cerere VO (VORequest) de instruire care conține A pe Piața Educației. Figura 3.2 ilustrează modul în care Preceptor gestionează două Cereri VO concomitente. Să presupunem că Cererea VO 1 a specificat o mulțime de competențe, A_1 , care conține trei competențe, în timp ce mulțimea de competențe A_2 specificată prin Cererea VO 2 conținea patru competențe. Atunci, fiecare Negociator VO potențial (PotentialVONegotiator) creat pentru Cererea VO 1 ar trebui să creeze la rândul său trei Holoni competență (CH), în timp ce fiecare Negociator VO potențial pentru cererea VO 2 ar trebui să creeze patru CH. Un Holon de programare (SH) este, de asemenea, creat de fiecare Negociator VO potențial.

Odată ce Cursantul alege o VO, Managerul cererii VO va trece la configurarea VO. O VO configurată are ca holon principal Cursantul. Instructorii vor fi, de asemenea, organizați într-o holarhie conform holarhiei competențelor. Diagrama de secvențe din figura 3.3 ilustrează întregul proces de formare și configurare inițială a VO.

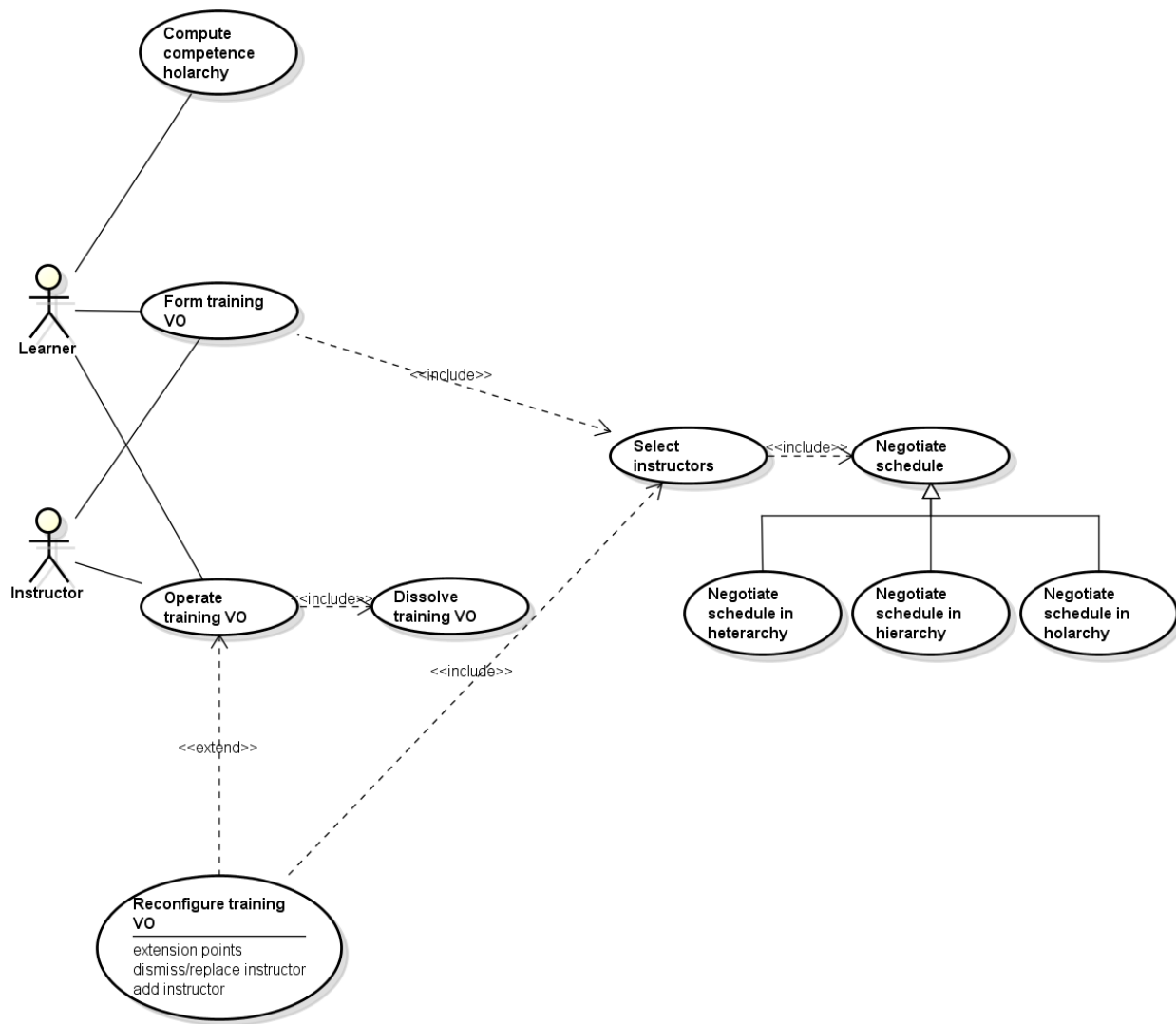


Figura 3.1: Diagrama cazurilor de utilizare pentru Preceptor

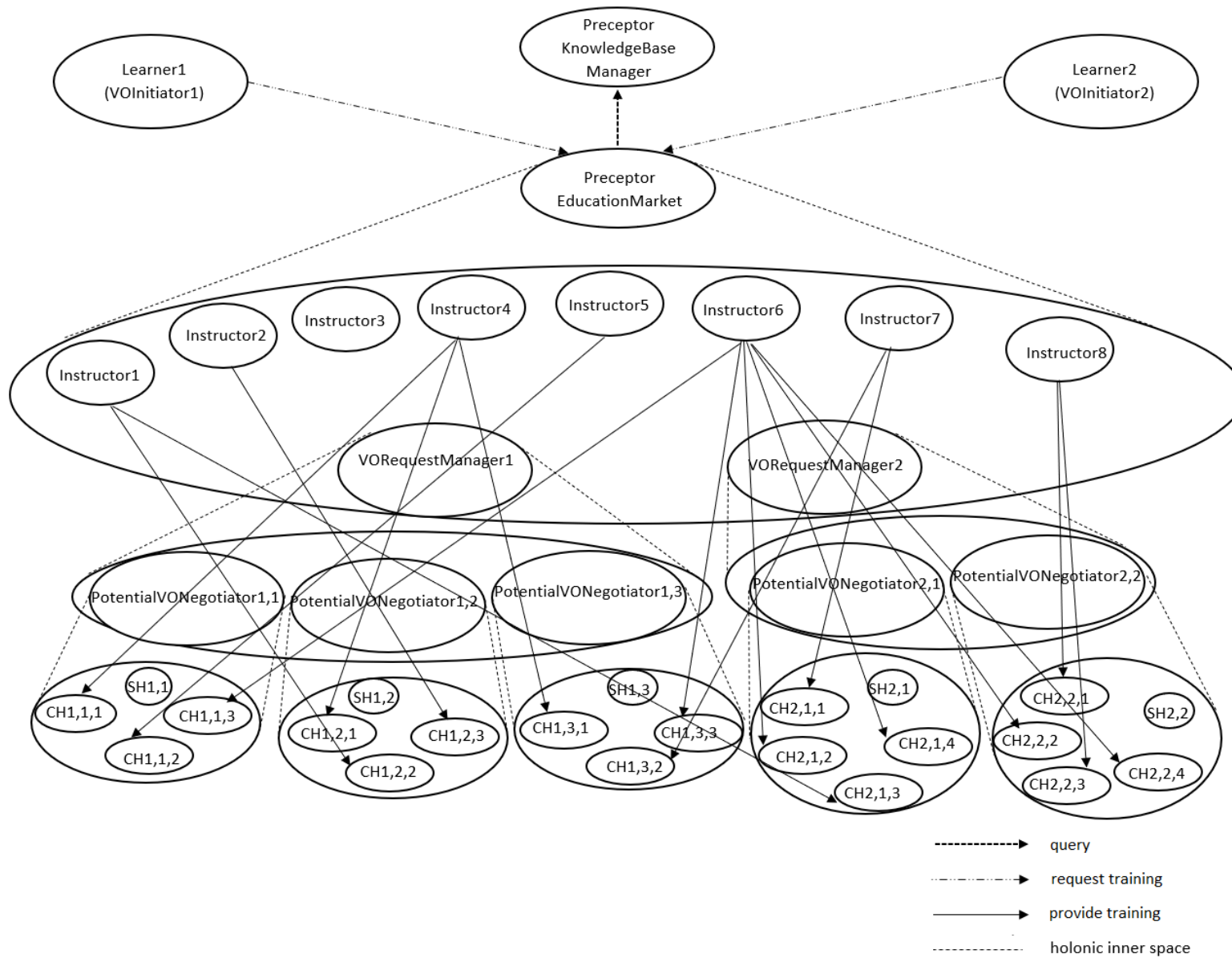


Figura 3.2: Prezentare generală a organizării holonice pentru formarea VO în cadrul Preceptor

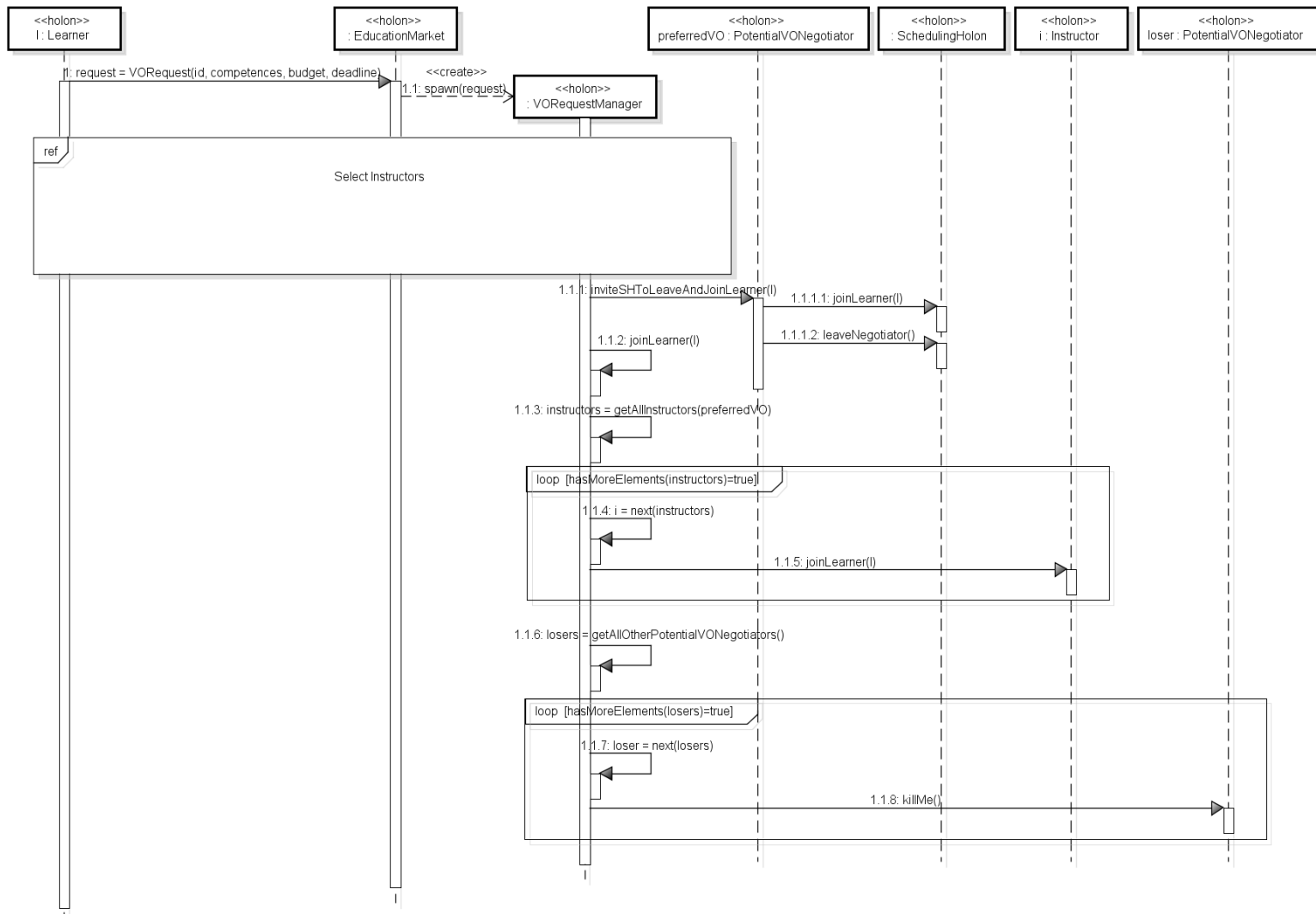


Figura 3.3: Diagrama de secvențe pentru formarea VO





Determinarea sau aproximarea echipei optimale de Instructori care poate acoperi mulțimea A de competențe este un ISP care ar trebui rezolvat pentru a ajunge la starea VO configurată. În acest scop, Managerul cererii VO (VORequestManager) va organiza o licitație pentru competențele solicitate. Fiecare Instructor de pe Piața Educației va depune una sau mai multe oferte pentru o competență din Cererea VO dacă și numai dacă Instructorul este calificat să ofere instruire pentru competența în cauză. O ofertă este un tuplu de forma:

$$bid = (InstructorID, c, StAv, EndAv, D, cost, quality) \quad (3.1)$$

Unde

$InstructorID \in \mathbb{N}$ este identificatorul instructorului

$c \in A$ este competența pentru care instructorul oferă instruire

$StAv$ și $EndAv$ indică începutul și, respectiv, sfârșitul ferestrei de disponibilitate

D , $cost$, și $quality$ denotă durata, costul și evaluarea calității serviciului de instruire oferit de $InstructorID$ pentru c , respectiv.

După închiderea licitației, Managerul de Cereri VO poate proceda la calcularea soluțiilor pentru Cererea VO. Fie B_i , $i \in \{1, \dots, n\}$ mulțimea de oferte pentru activitatea de formare a competențelor i . Definim o soluție s la Cererea VO ca o posibilă combinație de oferte, câte una pentru fiecare competență. Prin urmare, $s \in B_1 \times \dots \times B_n$.

Managerul cererii VO evaluează apoi toate soluțiile posibile. Fiecare soluție s este evaluată în funcție de două obiective: costul total și calitatea medie. Deoarece natura multiobiectiv a acestei probleme justifică căutarea soluțiilor de compromis, trebuie adoptată o metodă de optimizare multiobiectiv. De exemplu, algoritmul MOSOSS poate fi folosit pentru a furniza o aproximare a mulțimii de combinații optimale în sens Pareto. Pentru fiecare soluție din mulțimea Pareto, Managerul cererii VO generează un Negociator VO potențial. În scopul ajungerii la un acord asupra unui program, fiecare Negociator VO potențial generează un SH pentru programare și un CH pentru fiecare activitate de formare a competențelor. Comportamentul SH ca răspuns la o solicitare de programare a unui CH este descris de diagrama de secvențe din Figura 3.4. Încercările de programare ale SH se bazează pe o adaptare a algoritmului de programare propus în Ionescu și Vernic [14] (vezi diagrama de activitate din Figura 3.5).

În starea sa inițială în procesul de programare, CH nu este programat. În paralel, încearcă să-și calculeze programul emițând o solicitare de programare către SH și ascultă notificările de încheiere a activității de la CH care se ocupă de prerechizitele sale, precum și notificări de program complet fezabil din partea Negociatorului VO potențial. După ce SH răspunde printr-o oră de începere propusă, CH își va actualiza cea mai apropiată oră de începere și își va verifica disponibilitatea. Dacă este disponibil, CH va difuza ora calculată de încheiere a activității în contextul său implicit. Starea programului său va fi, de asemenea, actualizată pentru a indica programarea reușită. În caz contrar, va fi difuzat un mesaj de eșec. Când toate CH-urile sunt programate, SH va verifica fezabilitatea programului (adică $\forall c_1 \in A, \forall c_2 \in A \cap P_{c_1}, startTime_{c_2} + duration_{c_2} \leq startTime_{c_1}$, unde $startTime_c$ și $duration_c$ indică ora de începere și, respectiv, durata activității de formare pentru competența $c \in A$).

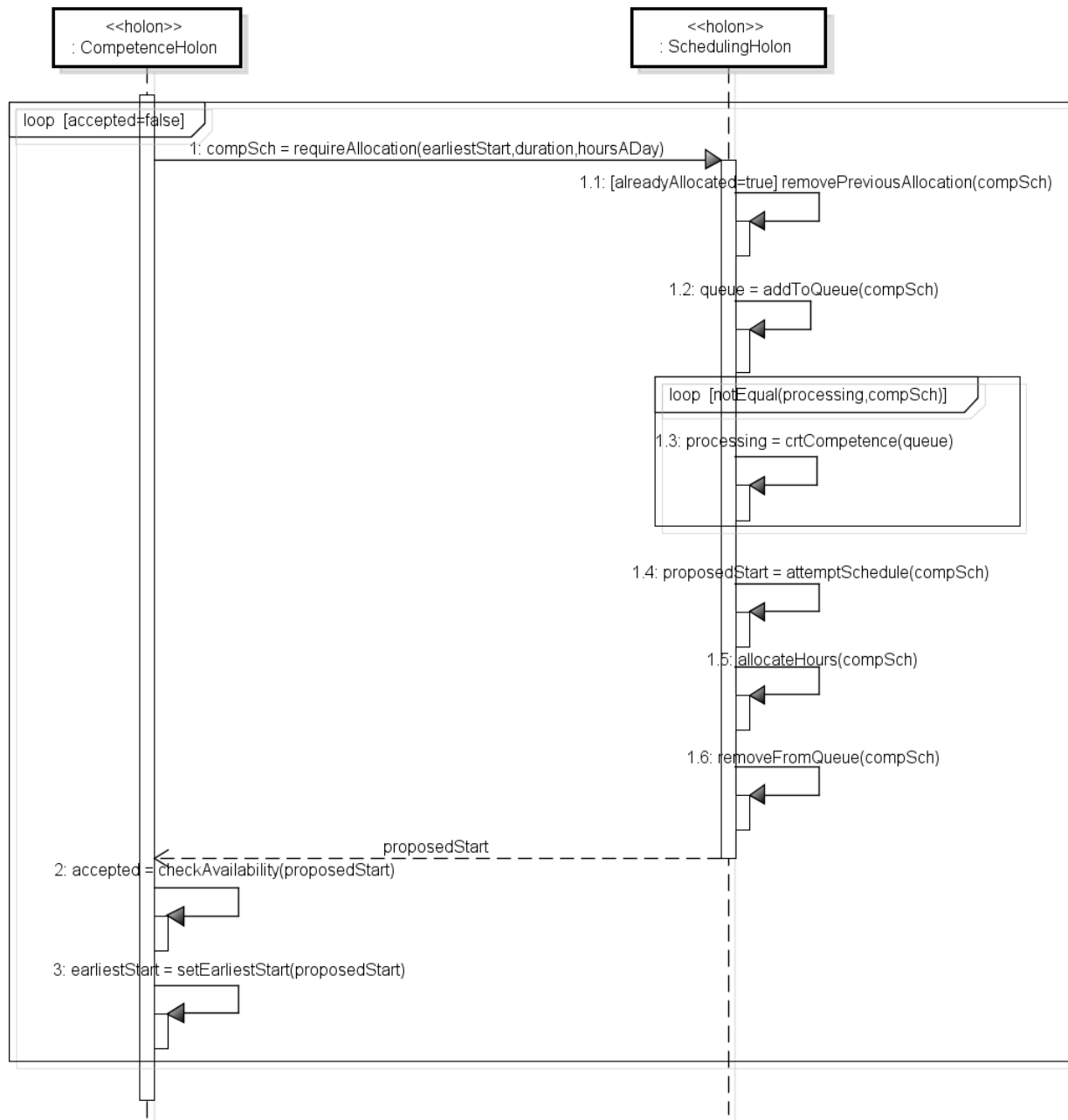


Figura 3.4: Diagrama de secvențe descriind comportamentul unui holon de programare

Când un SH ajunge la un program complet și fezabil, emite programul către Negociatorul VO potențial, care la rândul său îl emite Managerului de cereri VO. În cazul în care Negociatorul VO potențial este in-

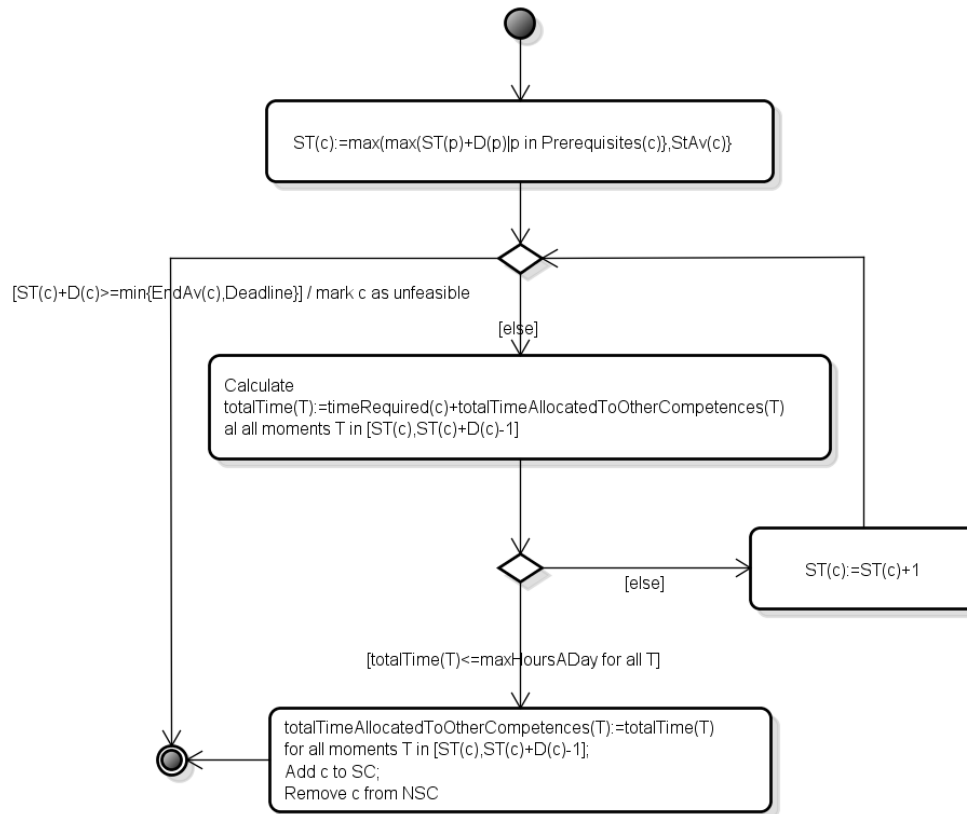
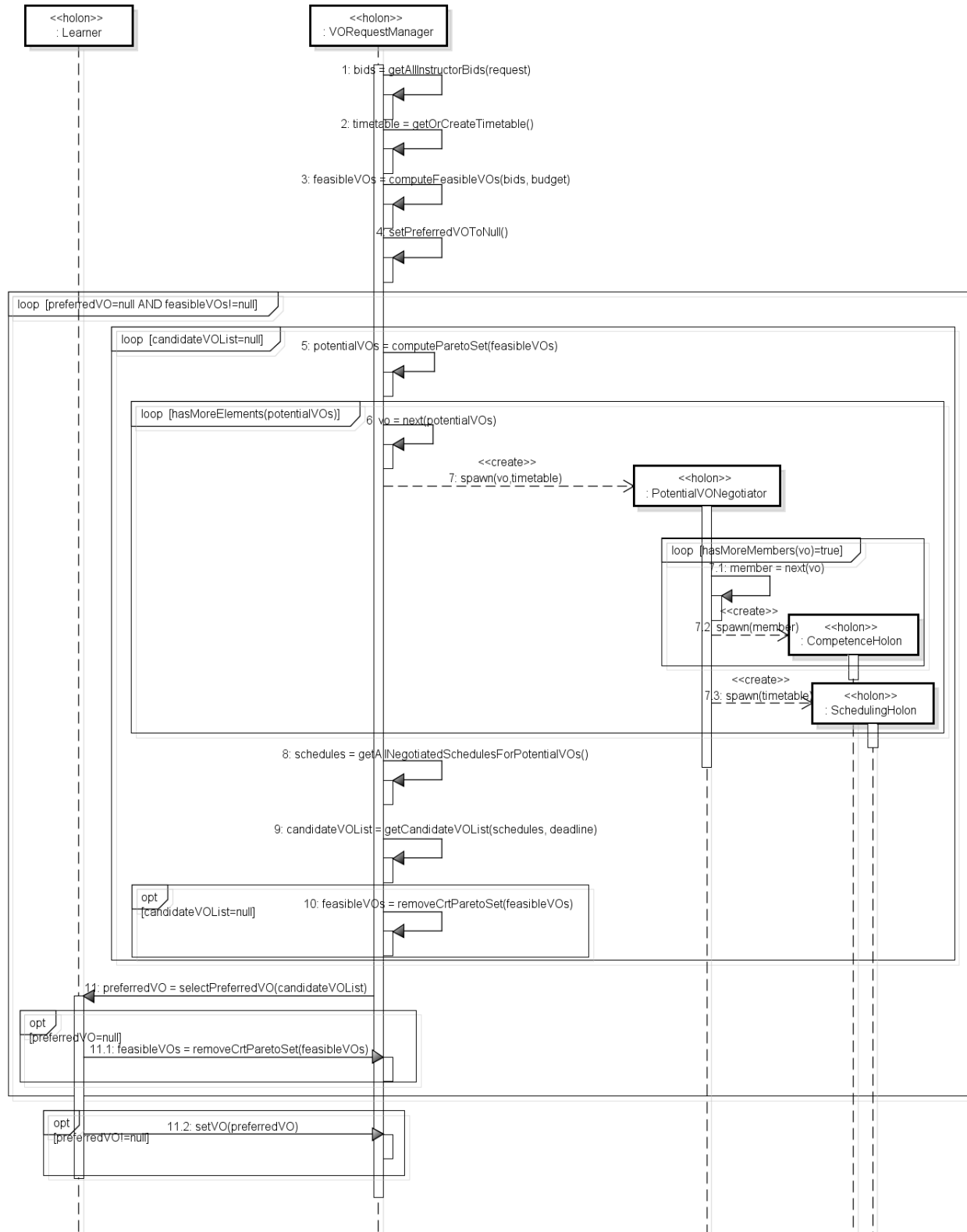


Figura 3.5: Diagrama de activități descriind o încercare de programare

format de către SH că nu poate fi generat niciun program fezabil pentru potențiala VO, Negociatorul VO potențial trimite un mesaj de eșec Managerului de cereri VO.

În cele din urmă, după ce a colectat toate programele (și mesajele de eșec, dacă este cazul) de la Negociatorii VO potențiali, Managerul cererii VO reține potențialele VO care sunt fezabile din punct de vedere al programului și îi cere cursantului să aleagă pe cea pe care o preferă. O perspectivă detaliată a cazului de utilizare pentru selecția instructorilor este oferită în Figura 3.6.





Pentru ca VO potențial ales să devină un VO de formare operațională, fiecare agent Instructor care a plasat o ofertă câștigătoare pentru VO ar trebui să fie notificat de către Managerul cererii VO cu privire la programul său individual, adresa agentului Cursant și adresele superholonilor și ale subholonilor săi imediați. Agenții Instructor sunt holoni multi-part [8]). În primul rând, își păstrează statutul de subholoni ai Pieței Educației. Acest lucru le permite să continue să asculte cererile VO care ar putea fi de interes, având în vedere abilitățile lor de formare. În al doilea rând, prin adeziunea la contextul interior al Cursantului, ei devin subholoni ai Cursantului. În al treilea rând, ei agregă zero sau mai mulți superholoni și zero sau mai mulți subholoni din rândul celorlalți Instructori din VO de instruire. Aceste din urmă relații holonice sunt dictate de relațiile de prerechizită între competențe. Fiecare agent Instructor, pentru fiecare dintre competențele pentru care oferă pregătire în VO, coordonează activitatea de formare a agenților Instructor care formează prerechizitele (dacă există) ale competenței în cauză. Acesta este mecanismul de bază al furnizării unei planificări curriculare personalizate pentru cursant.

O competență poate fi o prerechizită pentru mai mult de o altă competență în VO. Aceasta înseamnă că agentul Instructor care oferă instruire pentru competența $c \in A$ va fi coordonat de toți agenții Instructor din VO care formează orice altă competență pentru care c este o prerechizită. Fiecare dintre acești superholoni poate indica o altă strategie preferată, deoarece ori de câte ori $p \in P_{c_1} \cap P_{c_2}$, $c_1, c_2 \in A$, $c_1 \neq c_2$, este posibil ca $succes_{c_1,p} \neq succes_{c_2,p}$.

Soluția noastră propusă este următoarea: fiecare dintre superholoni ar trebui să-și transmită funcția de evaluare a succesului subholonilor săi (responsabil cu prerechizitele). Un subholon care oferă formare pentru prerechizita p va selecta apoi strategia $s_0 \in S_p$ care maximizează $\sum_{c \in \{c_1 \in A | p \in P_{c_1}\}} succes_{c,p}(s)$.

După cum se arată în Figura 3.7, fiecare agent Instructor va efectua activitatea (sau activitățile) de formare pentru care a fost selectat în VO—adică, corespunzătoare ofertelor sale câștigătoare. O cerere de reconfigurare este un punct de extensie pentru cazul de utilizare pentru operarea VO. Contractele cu subholoni pot fi renegociate dacă Cursantul decide.

În cazul în care cererea este una de concediere sau înlocuire, Managerul cererii VO va cere cursantului să evalueze serviciul de formare oferit pentru competența dată. La primirea unui răspuns al Cursantului care conține evaluarea, Managerul cererii VO va actualiza baza de cunoștințe cu evaluarea pentru serviciul de formare a competențelor furnizat sub forma unui tuplu (*Instructor, competen, crt_rating*). Acest tuplu va fi trimis agentului Manager de bază de cunoștințe (KnowledgeBaseManager), care ține evidența tuturor evaluărilor pentru toate perechile Instructor-competență. După ce agentul Manager de bază de cunoștințe primește mesajul de actualizare de la Managerul cererii VO, noua evaluare va fi calculată folosind ecuația 3.2b:

$$rating_0 = \infty \quad (3.2a)$$

$$rating_{n+1} = \frac{n * rating_n + crt_rating}{n + 1}, \forall n \geq 0 \quad (3.2b)$$

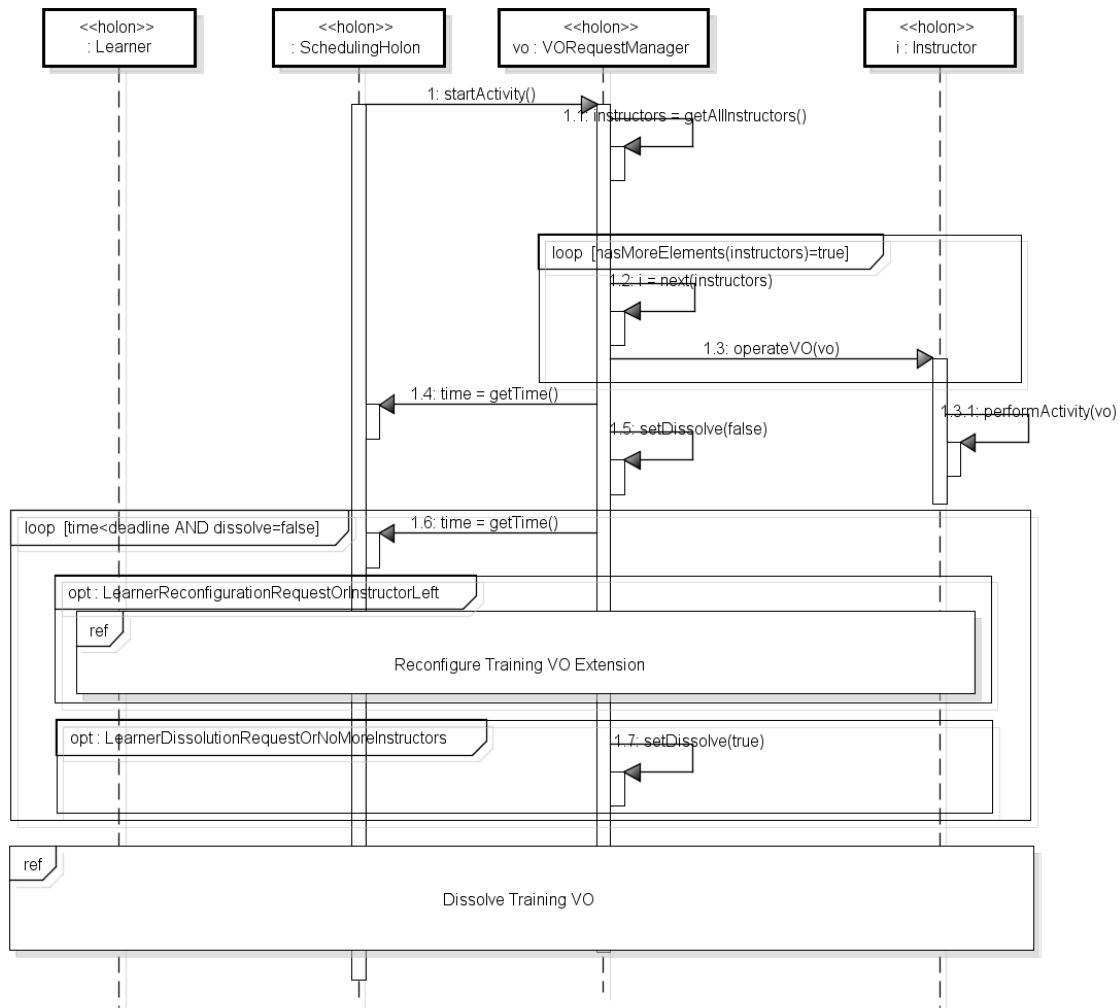


Figura 3.7: Diagrama de secvențe pentru Operarea VO

Dacă cererea de reconfigurare a Cursantului a fost fie una de înlocuire, fie una de adăugare, Cererea VO originală va fi actualizată prin înlocuirea mulțimii inițiale de competențe (A) cu competența din cererea de reconfigurare. Sistemul va reintra în cazul de utilizare Selecția instructorilor cu această Cerere VO actualizată (vezi Figura 3.8).

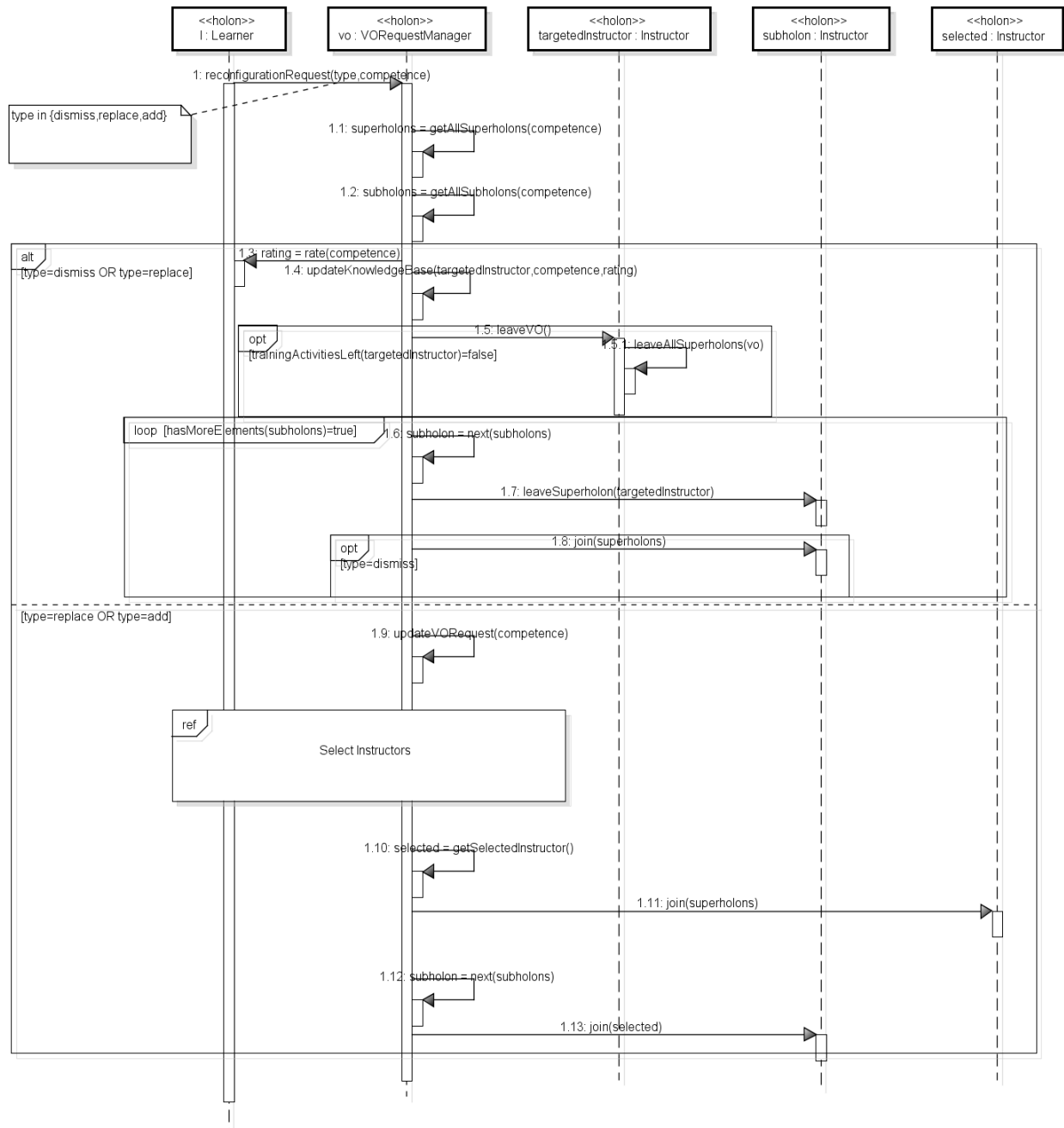


Figura 3.8: Diagrama de secvențe care descrie procesele implicate în extensia de reconfigurare a cazului de utilizare Operarea VO

Ori de câte ori Cursantul solicită în mod explicit dizolvarea, sau VO de instruire rămâne fără Instructori

sau toate activitățile de instruire s-au terminat, VO de instruire se dizolvă (vezi Figura 3.9).

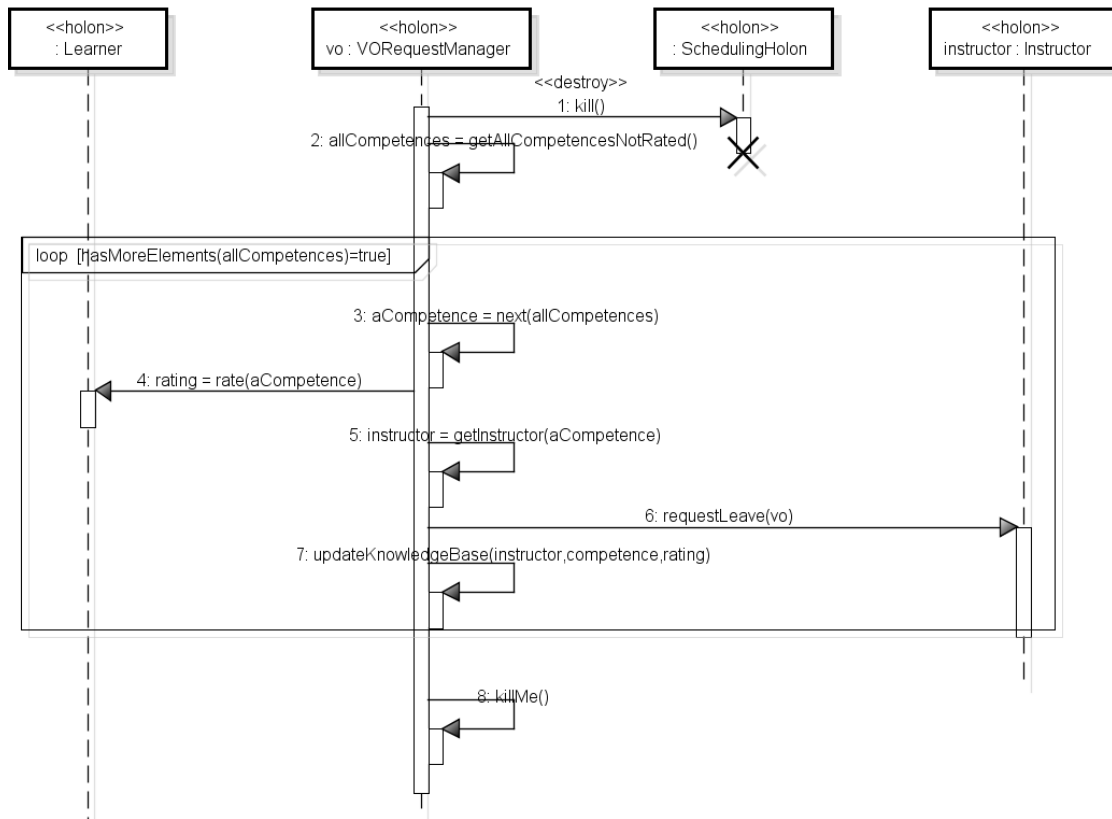


Figura 3.9: Diagrama de secvențe pentru Dizolvarea VO

Am implementat mediul VO de instruire proiectat ca o platformă HMAS codată în limbajul de programare SARL, versiunea 12 [21]. Alegerea noastră a fost motivată de suportul nativ pentru sistemele multiagent holonice oferit de SARL [7]. Ca mediu de execuție pentru HMAS pe care l-am implementat în SARL, am folosit platforma Janus recomandată [8]. Platforma Janus a fost dezvoltată cu scopul de a fi potrivită pentru aplicații mari, modulare. Designul său se bazează pe metamodelul CRIO, adoptând principiul separării rolurilor de holonii care le joacă.

3.2 Studiu de caz

În cele din urmă, pentru a atinge obiectivul O4, am testat arhitectura Preceptor folosind un studiu de caz descris după cum urmează. Un cursant solicită instruire pentru o singură competență: modelarea comportamentului și simularea în medii virtuale. Cursantul și Preceptor interacționează pentru a stabili

Întreaga listă de competențe pentru care cursantul are de fapt nevoie de formare. VO de formare a competențelor corespunzătoare ar trebui, prin urmare, să aibă o structură holonică precum cea reprezentată în Figura 3.10.

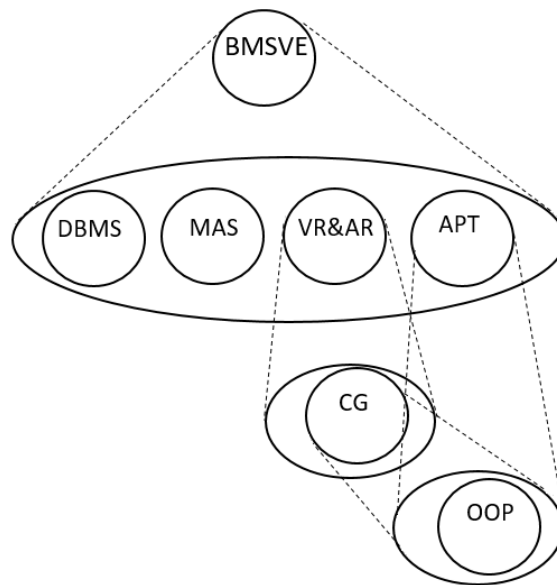


Figura 3.10: Studiu de caz: Holarhia competențelor

Am testat structura organizatorică propusă a Preceptor în timpul procesului de programare a activităților ("heterarhie"/descentralizată) față de două structuri organizatorice alternative ("ierarhie"/centralizată și "holarhie"/holonică). Cele trei structuri organizatorice diferă în ceea ce privește comportamentul Negociatorului VO potențial, al SH și al CH-urilor. În structura organizatorică ierarhică, programarea este responsabilitatea exclusivă a Negociatorului VO potențial (vezi Figura 3.11). În structura holarhică, fiecare CH devine un superholon al tuturor CH-urilor responsabili de oricare dintre prerechizitele sale imediate. După cum este ilustrat în diagrama de secvențe din figura 3.12, CH și SH vor interacționa prin trimiterea de propuneri până când ajung la un acord.

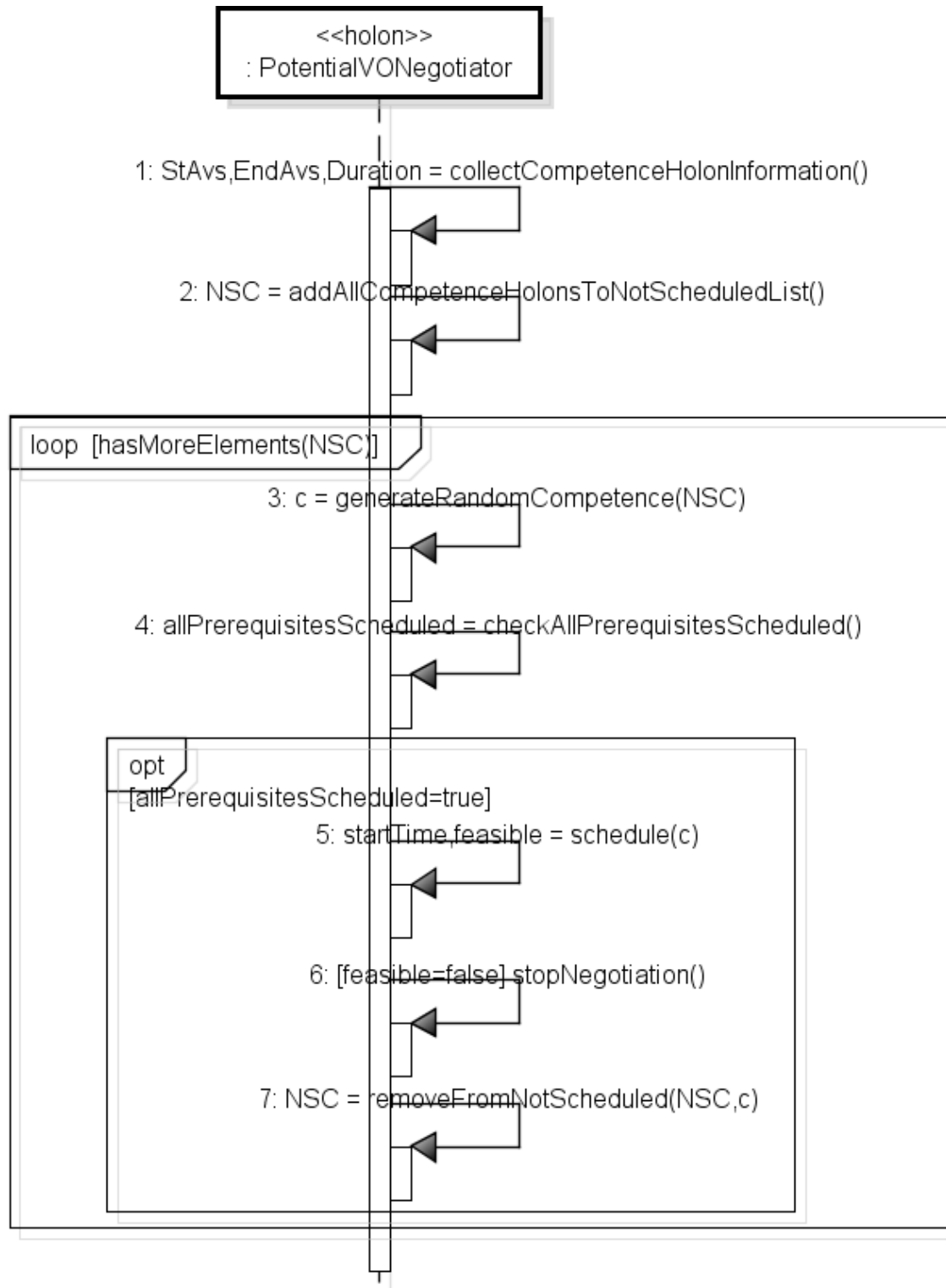


Figura 3.11: Diagrama de secvențe care descrie procesul de programare în structura organizatorică ierarhică

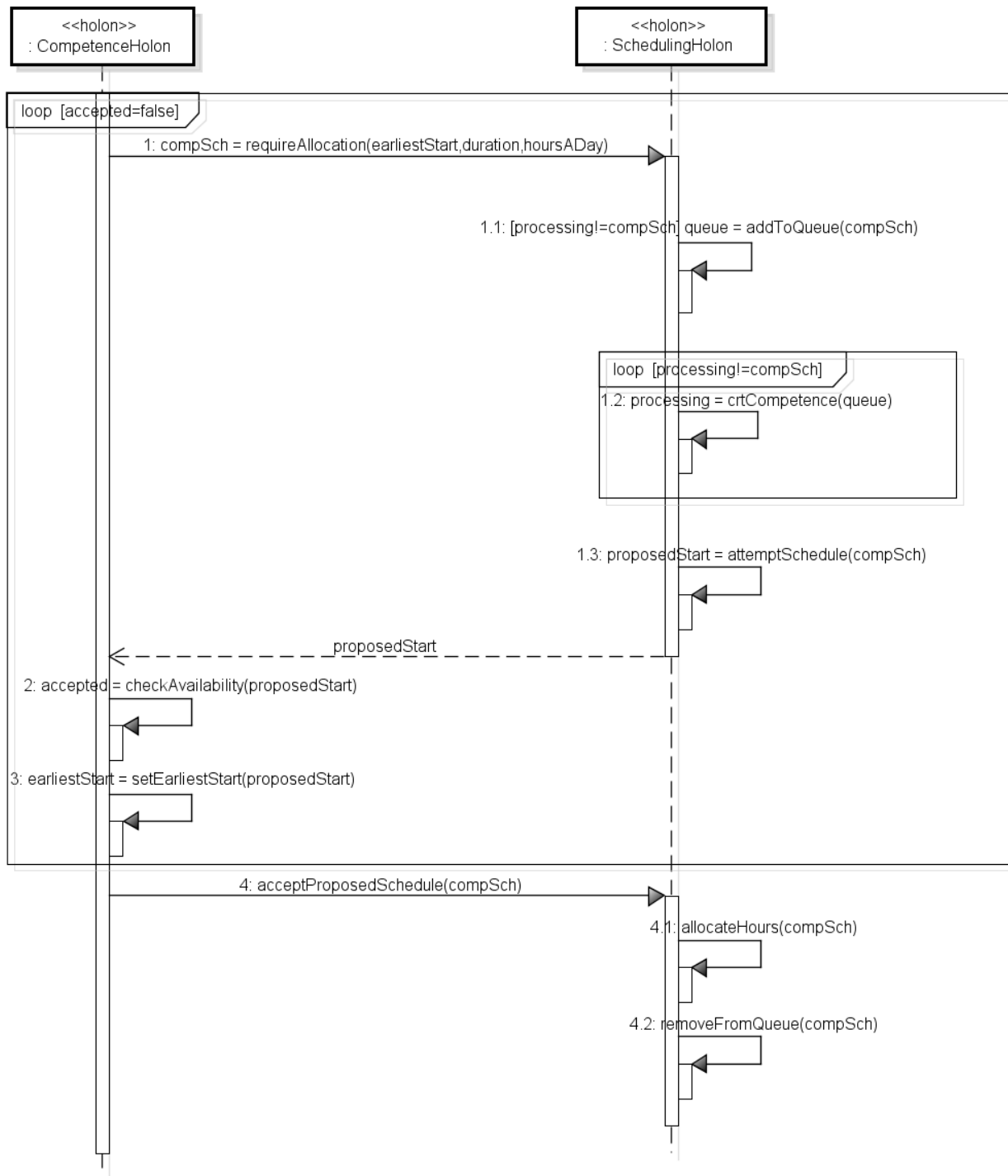


Figura 3.12: Diagrama de secvențe care descrie procesul de programare în structura organizatorică holarhică



3.2.1 Ipoteze

Emitem ipoteza că cele trei structuri organizatorice vor diferi semnificativ în ceea ce privește eficiența lor computațională, luând în considerare trei metrici de performanță: (1) timpul scurs (în milisecunde); (2) utilizarea procesorului (procentul de tick-uri de ceas CPU utilizate de procesele Preceptor din numărul total de tick-uri); și (3) utilizarea memoriei (procentul mediu de memorie utilizat de procesele Preceptor). Pentru fiecare metrică de performanță, ne interesează două valori: valoarea totală pentru întregul proces de configurare și valoarea doar pentru procesul de programare.

Ne așteptăm ca timpul de execuție pentru structura holarhică să fie mai lung, deoarece CH-urile formează și ele o holarhie, inducând astfel întârzieri în comunicare. Prin urmare, emitem ipoteza:

Ipoteza 1. Timpul total de configurare (a) și timpul de programare (b) vor fi semnificativ mai lungi pentru structura organizatorică a holarhică în comparație cu heterarhia și ierarhia.

În ceea ce privește utilizarea CPU, ne așteptăm totuși ca organizarea holoncă să fie superioară cel puțin structurii heterarhice, deoarece holonii așteaptă toate informațiile de care depinde o decizie și comunică altor holoni doar decizia finală. Emitem astfel ipoteza:

Ipoteza 2. Utilizarea totală a CPU (a) și utilizarea CPU în timpul procesului de programare (b) vor fi semnificativ mai mici pentru structura organizatorică holarhică în comparație cu heterarhia și ierarhia.

În ceea ce privește utilizarea medie a memoriei, ne așteptăm ca organizația ierarhică să producă cele mai mari valori, deoarece toate structurile de date sunt stocate în memorie simultan până când Negociatorul VO potențial încheie procesul de programare. Ca atare, emitem ipoteza:

Ipoteza 3. Utilizarea medie a memoriei în timpul configurării (a) și în timpul procesului de programare (b) va fi semnificativ mai mare pentru structurile organizatorice ierarhice în comparație cu heterarhia și holarhia.

3.2.2 Setări experimentale

Pentru a crea setări experimentale care pot fi reproduse cu ușurință, am generat aleatoriu 8 soluții pentru problemă. Fie S mulțimea de 8 soluții. Putem genera primele 50 de combinații de 5 soluții din S și presupunem că fiecare combinație este mulțimea Pareto pentru o mulțime necunoscută de combinații posibile de oferte de instruire, inclusiv ofertele care compun soluțiile din S . Mulțimea de soluții generate aleator (S) și cele 50 de mulțimi Pareto sunt enumerate în Anexa 3. Fiecare dintre cele 50 de mulțimi Pareto generate a fost un caz în experimentele pe care le-am efectuat pentru a compara cele trei structuri organizatorice diferite. Frecvența cu care CH-urile verificau notificările de setare a momentului de încheiere a activității pentru prerechizite în structurile de heterarhie și holarhie a fost setată la o dată la 100 de milisecunde. Rezultatele experimentale sunt raportate în secțiunea 4.2.

Capitolul 4

Rezultate

În acest capitol, prezentăm mai întâi metoda noastră propusă pentru rezolvarea PSP cu constrângeri de timp și resurse (în secțiunea 4.1), care poate fi adaptată cu ușurință la cazuri particulare de PSP, cum ar fi găsirea combinației optime de instructori pentru o cerere de instruire. Apoi, în secțiunea 4.2 raportăm rezultatele pe care le-am obținut în urma utilizării arhitecturii Preceptor pentru studiul de caz descris în secțiunea 3.2.

4.1 Abordarea propusă pentru problema de selecție a partenerilor: Algoritmul Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS)

Două inovații majore care diferențiază MOSOSS de algoritmi consacrați pentru probleme similare sunt operatorii simbiotici (mutualism și comensalism) special proiectați pentru optimizarea combinatorială și tratarea soluțiilor nefezabile (MOSOSS funcționează cu soluții parțial programate).

Codificarea soluției a fost următoarea:

$$\underbrace{bid_1 bid_2 \dots bid_n}_{\text{oferte componente}} \quad \underbrace{ST_1 ST_2 \dots ST_n}_{\text{momente de începere programate}} \quad (4.1)$$

Ori de câte ori programarea unei activități nu este fezabilă din cauza încălcării unei constrângeri de timp, activitatea este programată astfel încât momentul de încheiere să depășească termenul limită cu 1 zi. Astfel, pentru fiecare organism se produce cel puțin un program parțial, având șansa de a fi reparat folosind operatorii evolutivi simbiotici în timpul iterațiilor ulterioare ale algoritmului, în loc să se aplice o metodă suplimentară de reparare. Operatorii simbiotici sunt special concepuți pentru ca soluțiile să profite una de compoziția celeilalte. Ambii folosesc mecanismul de exploatare prezentat în 1.



Algorithm 1 Exploit(sol_i, sol_j)

Require: $sol_i = ((bid_1^{(i)}, \dots, bid_n^{(i)}), (ST_1^{(i)}, \dots, ST_n^{(i)}))$;

$sol_j = ((bid_1^{(j)}, \dots, bid_n^{(j)}), (ST_1^{(j)}, \dots, ST_n^{(j)}))$.

Ensure: $sol'_i = ((bid_1^{(i')}, \dots, bid_n^{(i')}), (ST_1^{(i')}, \dots, ST_n^{(i')}))$, o soluție obținută prin înlocuirea ofertelor în sol_i cu oferte corespunzătoare în sol_j .

$sol'_i \leftarrow sol_i$

$replace \leftarrow False$

Dacă sol_i are o activitate în așteptare, înlocuiește oferta din i' cu oferta din j :

if $\exists k$ s.t. $Pending_i = k$ **then**

$bid_k^{(i')} \leftarrow bid_k^{(j)}$

$evaluate(sol'_i)$

if $sol'_i \succ sol_i$ **then**

$replace \leftarrow True$

end if

if $replace = False$ **then**

for $k \in \{1, \dots, n\}$ **do**

if $ST_k^{(i)} = -1$ or $Pending_i = k$ **then**

$bid_k^{(i')} \leftarrow bid_k^{(j)}$

$evaluate(sol'_i)$

if $sol'_i \succ sol_i$ **then**

$replace \leftarrow True$

end if

end if

end for

end if

else

Resetează programul sol'_i :

$ST_k^{(i')} \leftarrow -1, \forall k \in \{1, \dots, n\}$

$Pending_{i'} \leftarrow NULL$

Generează un număr aleator de oferte de înlocuit, $rand \in \{1, \dots, n\}$.

Înlocuiește ofertele pentru $rand$ activități aleatoare cu cele corespunzătoare de la sol_j :

for $k \in \{1, \dots, rand\}$ **do**

Generează o activitate aleatoare $act_k \in \{1, \dots, n\}$

if $bid_{act_k}^{(i)} \neq bid_{act_k}^{(j)}$ **then**

$bid_{act_k}^{(i')} \leftarrow bid_{act_k}^{(j)}$

end if

end for

$\{totalCost'_i, totalDuration'_i, avgQuality'_i, c'_i\} \leftarrow Evaluate(sol'_i)$

end if

return sol'_i



Am efectuat experimente numerice (raportate în [14]) comparând MOSOSS cu MOSOS consacrat și cu NSGA-II folosind următorii șase indicatori de performanță pentru optimalitatea Pareto: ϵ aditiv, GD, IGD, IGD+, HV și Δ . Rezultatele au susținut superioritatea MOSOSS față de algoritmi concurenți pentru toți indicatorii de performanță și pentru toate instanțele PSP de testare generate aleatoriu. Instanțele de testare au fost 15, 5 instanțe cu 3 activități, 5 cu 4 activități și alte 5 cu 5 activități. Am demonstrat, de asemenea, viteza de convergență mai mare a MOSOSS, care a acoperit mai mult de 90% din mulțimea Pareto după mai puține iterații decât au avut nevoie algoritmi concurenți pentru a găsi orice soluție. Cititorul interesat este îndrumat către [14] pentru detalii suplimentare.

4.2 Studiu de caz: Răspuns la o cerere de organizație virtuală de instruire folosind arhitectura Preceptor

În această secțiune, raportăm rezultatele studiului de caz descris în secțiunea 3.2.

Statisticile descriptive pentru cei șase indicatori de performanță sunt raportate numeric în Tabelul 4.1 și vizual în Figura din Anexa 4.

Ne-a interesat existența diferențelor între structurile organizatorice în ceea ce privește indicatorii performanței. În primul rând, trebuia să testăm efectul factorului nostru (structura organizatorică, cu trei valori posibile: heterarhie, ierarhie și holarhie) asupra fiecărei variabile dependente (indicator al performanței).

Am folosit pachetul R `rstatix` [15] pentru a aplica testul Friedman și testele post-hoc. Un rezultat semnificativ al testului Friedman ar implica faptul că cel puțin două dintre cele trei structuri organizatorice diferă în ceea ce privește variabila dependentă în cauză. După cum este raportat în Tabelul 4.2, rezultatele testului Friedman au indicat o diferență semnificativă ($p < .05$) între scenariile numai în ceea ce privește timpul și utilizarea CPU. Nu s-a constatat că utilizarea medie a memoriei diferă semnificativ între cele trei scenarii. Prin urmare, ipoteza 3 nu a fost susținută.

Următorul pas a fost efectuarea de comparații pe perechi pentru a stabili mai exact care structuri organizatorice diferă semnificativ în raport cu variabilele dependente. Prin urmare, am efectuat comparații post-hoc pe perechi folosind testul tuturor perechilor Nemenyi-Wilcoxon-Wilcox din `rstatix` [15]. Rezultatele sunt raportate în tabelele 4.3–4.4.

După cum arată tabelele, rezultatele testelor indică diferențe semnificative ($p < .05$) între toate perechile atât pentru timpul total, cât și pentru timpul de programare. După cum era de așteptat, în ceea ce privește timpul scurs, organizația ierarhică a depășit heterarhia, care, la rândul său, a fost mai rapidă decât holarhia. Prin urmare, ipoteza 1 a fost pe deplin susținută.

Cu toate acestea, interpretând rezultatele prezentate în tabelul 4.4, atât holarhia cât și heterarhia au depășit ierarhia în ceea ce privește utilizarea totală a CPU ($p < .05$). Ipoteza 2a a fost astfel parțial

Tabela 4.1: Statistici descriptive pentru indicatori de performanță

Indicator de performanță	Structură Organizatorică	Medie	SD	Mediană	IQR
Timp (total)	Heterarhie	1701	95.2	1710	110.0
	Hierarhie	1135	88.3	1114	104
	Holarhie	2000	145.0	2002	101
Timp (programare)	Heterarhie	743	78.9	705	106
	Hierarhie	153	91.5	150	97.8
	Holarhie	1002	142.0	996	99
Utilizarea CPU (total)	Heterarhie	11,8	5,76	11	3,75
	lerarhie	13.4	6.67	12	4.75
	Holarhie	11.6	4.64	11	3
Utilizarea CPU (programare)	Heterarhie	17.7	7.67	17	7.75
	lerarhie	19.0	18.2	14.5	14.8
	Holarhie	11.6	3.97	11	6
Utilizarea memoriei (total)	Heterarhie	21.5	6.00	22.0	7.42
	lerarhie	23.1	6.82	22.8	8.30
	Holarhie	25.1	8.64	25.2	12.0
Utilizarea memoriei (programare)	Heterarhie	27.7	8.04	27.4	8.88
	lerarhie	32.8	11.6	32.8	12.4
	Holarhie	29.0	9.59	30.1	11.2

SD = abatere standard; IQR = amplitudine intercuartilică

Tabela 4.2: Rezultatele testului Friedman

Indicator al performanței	F	p
Timp (total)	100	1.93e-22
Timp (programare)	100	1.93e-22
Utilizare CPU (total)	12.3	0.00217
Utilizarea procesorului (programare)	12.5	0.00190
Utilizarea memoriei (total)	4.68	0.0963
Utilizarea memoriei (programare)	3.88	0.144

susținută. Diferența dintre holarhie și heterarhie nu a fost semnificativă. În ceea ce privește utilizarea CPU pentru procesul de programare, doar diferența dintre holarhie și heterarhie a fost semnificativă, holarhia producând o utilizare mai eficientă a CPU. Ipoteza 2b a fost astfel parțial susținută.



Tabela 4.3: Comparații pe perechi între structurile organizatorice: Timp

Structură organiza- torică	Timp total			Timp programare		
	Heterarhie	Ierarhie	Holarhie	Heterarhie	Ierarhie	Holarhie
Heterarhie	-			-		
Ierarhie	1.7e-06 ▲	-		1.7e-06 ▲	-	
Holarhie	1.7e-06 ▽	2.4e-14 ▽	-	1.7e-06 ▽	2.4e-14 ▽	-

Note. Valorile raportate în tabel sunt valorile p pentru testul Nemenyi-Wilcoxon-Wilcox.

▲ = structura organizatorică de pe linie este mai performantă decât cea de pe coloană;

▽ = structura organizatorică de pe linie este mai puțin performantă decât cea de pe coloană;

- = diferența dintre structura organizatorică de pe linie și cea de pe coloană nu este semnificativă.

Tabela 4.4: Comparații pe perechi între structurile organizatorice: Utilizarea CPU

Structură organiza- torică	CPU total			CPU programare		
	Heterarhie	Ierarhie	Holarhie	Heterarhie	Ierarhie	Holarhie
Heterarhie	-			-		
Ierarhie	0.0141 ▽	-		0.2456 -	-	
Holarhie	0.9661 -	0.0065 ▲	-	0.0014 ▲	0.1386 -	-

Note. Valorile raportate în tabel sunt valorile p pentru testul Nemenyi-Wilcoxon-Wilcox.

▲ = structura organizatorică de pe linie este mai performantă decât cea de pe coloană;

▽ = structura organizatorică de pe linie este mai puțin performantă decât cea de pe coloană;

- = diferența dintre structura organizatorică de pe linie și cea de pe coloană nu este semnificativă.

Capitolul 5

Discuții

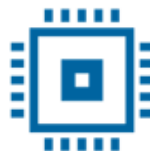
În această teză, am propus o abordare a furnizării de instruire individualizată prin intermediul unui HMAS strategic, urmată de sugestii pentru proiectarea și implementarea unui sistem care să susțină astfel de funcționalități și experimente numerice comparând trei arhitecturi diferite (structuri organizatorice) pentru negocierea programului.

Argumentele teoretice și rezultatele noastre sugerează că un model de organizare complet holonic în timpul formării VO nu numai că permite un mediu VO mai dinamic, dar este și mai eficient în ceea ce privește utilizarea CPU. Cu toate acestea, este mai costisitor din punct de vedere al timpului de calcul decât o structură organizatorică heterarhică sau ierarhică. Rezultatele experimentale sugerează că, deși structurile organizatorice descentralizate (holarhie și heterarhie) funcționează mai bine în ceea ce privește utilizarea CPU, structura organizatorică ierarhică are un avantaj important—acela de a fi mai rapidă. Cu toate acestea, unul dintre principalele dezavantaje ale structurii ierarhice este lipsa de flexibilitate. Dacă subholonii au cel mult drept de veto, un acord cu toți subholonii poate fi greu de atins. Dacă formarea și reconfigurarea VO sunt complet centralizate, fără a accepta întreruperi de la subholoni, Negociatorul VO potențial produce configurații și programe rigide.

5.1 Utilitate

Arhitectura propusă abordează planificarea personalizată a curriculum-ului la cerere. Concret, asistă novicii în trei domenii majore: identificarea nevoilor lor de formare din punct de vedere al competențelor, găsirea unei combinații optime de instructori pentru acele competențe care necesită pregătire și generarea unui plan personalizat de învățare prin coordonarea holonică a instructorilor.

Structura holonică a platformei creează un echilibru între centralizare și descentralizare, permițând holonilor de la diferite niveluri ale holarhiei să-și gestioneze subholonii. Modelarea VO ca subholoni ai Pieței



Educației oferă avantajul de a permite relații simultane de colaborare (ca formatori pentru competențe pentru aceeași cerere) și competitive (ca ofertanți pentru cereri diferite) între instructori.

5.2 Contribuții originale

Cele mai importante contribuții originale ale prezentei teze pot fi rezumate după cum urmează:

- propunerea unui cadru pentru predarea colaborativă la cerere. Am raportat despre proiectarea și implementarea unei soluții la cererea tot mai mare de formare specializată, care respectă recomandările UNESCO și recenziile critice recente ale rolului IA în educație, prin aceea că evită înlocuirea profesorilor umani cu IA.
- propunerea unei adaptări a unei noi euristici prin reproiectarea completă a operatorilor MOSOS, astfel încât să servească mai bine scopului rezolvării PSP ca MOP combinatorială cu programarea activităților supusă unor constrângeri de timp, buget, resurse și prioritate a activităților. Oferim, de asemenea, o abordare inovatoare a programării activităților prin evoluția soluțiilor parțial programate.
- identificarea unei verigi între mediile VO și VO în sine, prin propunerea unei organizații holonice în care VO sunt subholoni ai mediului VO („piață”). Aplicabilitatea acestei idei poate fi extinsă dincolo de sfera acestei teze (formare), la ipotetic orice VO.
- propunerea, testarea și compararea abordărilor concurente ale negocierilor pentru programarea activităților în timpul ISP. Pe baza studiului nostru de caz, am evidențiat avantajele și dezavantajele diferitelor structuri organizatorice în timpul acestor procese.

5.3 Limitări

În prezenta teză, ne-am limitat la furnizarea unei dovezi a conceptului în scopul de a demonstra o parte a funcționalității pe care arhitectura Preceptor este menită să o ofere. Câteva limitări importante care ar trebui abordate într-o agendă de cercetare viitoare sunt enumerate în cele ce urmează.

În primul rând, arhitectura nu a fost încă validată pe scenariii din lumea reală și trebuie măsurată satisfacția părților interesate față de calitatea serviciului. În plus, trebuie efectuate mai multe experimente care compară structurile organizatorice concurente în timpul formării VO pentru a analiza impactul parametrilor cum ar fi frecvența sarcinilor de ascultare/monitorizare și pentru a le ajusta pe baza rezultatelor analizelor de sensibilitate. În al doilea rând, mai multe structuri alternative pentru diferite faze ale ciclului de viață VO merită să fie explorate în lucrări viitoare. În al treilea rând, ar trebui luat în considerare un sistem de evaluare mai avansat pentru calitatea instruirii, cum ar fi evaluările pentru colaborare.



5.4 Direcții viitoare de cercetare

Pe baza limitărilor menționate mai sus, precum și a potențialului de extindere al arhitecturii, pot fi identificate următoarele direcții pentru o viitoare agendă de cercetare.

În primul rând, comportamentul Preceptor în timpul fazei de operare a VO ar putea fi rafinat în implementări mai concrete ale colaborării profesorilor. Pentru ca acestea să fie eficiente, ar trebui să se bazeze pe o analiză a nevoilor, a opiniilor și a preferințelor instructorilor. De exemplu, eforturile de colectare a datelor ar putea viza membrii Universităților Europene¹ sau alți furnizori de instruire care oferă un curriculum flexibil, centrat pe cursant și/sau formarea de echipe transdisciplinare orientate spre obiective, cum ar fi Școala de Afaceri UE² sau UNIR³.

În al doilea rând, având în vedere accentul care a fost pus în mod constant în literatura de specialitate VO pe agilitate, ar fi deosebit de interesant să se facă distincția între diverși factori declanșatori ai reconfigurării, cum ar fi modificări ale competențelor cerute sau modificări ale programului impus de Cursant. Comportamentele optime ale holonilor implicați în procesul de reconfigurare pot diferi pentru diferiți factori declanșatori, astfel încât fiecare dintre aceste scenarii merită o analiză mai detaliată.

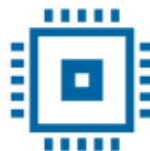
În al treilea rând, bazându-se pe modelul pieței resurselor, viitoarele extensii ale Preceptor pot restrânge căutarea potențialilor parteneri VO prin rezolvarea ISP-ului în două etape. În primul rând, într-o etapă pasivă, Preceptor ar putea crea o listă de oferte de formare care să fie relevante pentru competențele cerute și să satisfacă constrângerile impuse de Cursant. În al doilea rând, în stadiul activ, ar putea notifica agenții Instructor din lista asupra cererii de instruire și permite Cursantului să negocieze cu aceștia.

În al patrulea rând, dezvoltarea arhitecturii Preceptor servește ca pas preliminar la ceea ce poate fi denumit E-learning as a Service (EaaS) [10]. Ofertele instructorilor sunt servicii care pot fi descrise și descoperite folosind web-ul semantic. În plus, Preceptor se pretează să fie extins la o platformă web care acceptă trei categorii de utilizatori: (1) Angajatorii, care trebuie să recruteze candidați pentru un anumit loc de muncă și să folosească platforma pentru a posta anunțul de angajare; (2) Instructorii, care ar putea folosi platforma pentru a posta oferte de formare; (3) Cursanții, care ar putea să se aboneze la platformă pentru a citi anunțurile de angajare postate de angajatori și pentru a posta cereri de formare pentru competențele necesare care le lipsesc. Toate interacțiunile utilizatorilor cu platforma ar trebui să fie efectuate printr-o interfață web care să permită contribuții la baza de cunoștințe ale tuturor categoriilor de utilizatori. De exemplu, angajatorii ar putea contribui prin descrierea noilor competențe de care au nevoie și prin specificarea relațiilor de prerechizită între aceste competențe. O ontologie a competențelor poate fi dezvoltată în colaborare în acest mod.

¹a se vedea Inițiativa Universităților Europene: https://ec.europa.eu/education/education-in-the-eu/european-education-area/european-universities-initiative_en

²<https://www.euruni.edu/>

³<https://en.unir.net>



5.5 Diseminarea rezultatelor cercetării

Recenziile și rezultatele cercetării originale prezentate în această teză au fost, de asemenea, raportate în următoarea listă de publicații:

5.5.1 Articole în reviste științifice indexate Web of Science

A.-F. Ionescu and R. Vernic, "MOSOSS: An adapted multi-objective symbiotic organisms search for scheduling," *Soft Computing*, vol. 25, no. 14, pp. 9591–9607, 2021. WOS:000641235700007

5.5.2 Articole/Capitole indexate Web of Science

A.-F. Ionescu, "Methods and algorithms for creating and reconfiguring virtual organizations," *Decision Making in Social Sciences: Between Traditions and Innovations*, Springer, 2020, pp. 49–63. WOS:000640236300003

5.5.3 Alte capitole în volume editate

A.-F. Ionescu, "Multi-objective evolutionary algorithms: Decomposition versus indicator-based approach," in *Algorithms as a Basis of Modern Applied Mathematics*, Springer, 2021, pp. 69–85.

A.-F. Ionescu and D.-M. Popovici, "Applications of multi-agent systems in social sciences: Virtual enterprises as an example," in *Models and Theories in Social Systems*, Springer, 2019, pp. 311–325.

5.5.4 Lucrări în volumele unor conferințe internaționale indexate Web of Science

A.-F. Ionescu, "Designing virtual learning systems: Current trends and evaluation," in *Proceedings of the 14th International Conference on Virtual Learning (ICVL)*, 2019, pp. 303–308. WOS:000506084800044

A.-F. Ionescu, "E-learning as a Service: Benefits of the semantic web and SOA for virtual learning," in *Proceedings of the 14th International Conference on Virtual Learning (ICVL)*, 2019, pp. 401–407. WOS:000506084800059

A.-F. Ionescu and D. Sburlan, "AdABI: An adaptive assessment system based on Bayesian inference," in *Proceedings of the 15th International Scientific Conference on eLearning and Software for Education (eLSE)—New Technologies and Redesigning Learning Spaces*, 2019, pp. 288–295. WOS:000473322400039



Universitatea
Transilvania
din Braşov



Universitatea
Transilvania
din Braşov

FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

5.5.5 Prezentări orale la conferințe internaționale

A.-F. Ionescu and D.-M. Popovici, "Preceptor: A Proposed Architecture for an On-Demand Virtual Learning Platform", paper accepted for presentation at the 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2021.

Capitolul 6

Concluzii

În concluzie, toate obiectivele tezei au fost îndeplinite. Pentru a atinge obiectivele O1 și O2, ne-am pe o sinteză a literaturii din domeniul metodelor de rezolvare a PSP. Pentru a obține O3, am detaliat proiectarea soluției software propuse. Pentru a ajunge la O4, am descris un studiu de caz folosit pentru a testa soluția dezvoltată și am raportat rezultatele—algoritmul propus pentru PSP și validarea acestuia, precum și teste ale eficienței mediului VO dezvoltat.

Putem aminti câteva contribuții originale importante ale cercetării prezentate în această teză:

- proiectarea și implementarea unei soluții la cererea tot mai mare de formare specializată: un mediu pentru predarea colaborativă la cerere.
- încadrarea explicită a formării/reconfigurării VO ca un caz particular de PSP și integrarea literaturii PSP și a optimizării multi-obiectiv în cadrul VO
- propunerea unei adaptări a unei euristici recente (MOSOS) pentru rezolvarea PSP cu programarea sarcinilor sub constrângeri de timp, buget, resurse și prioritate a sarcinilor.
- identificarea unei verigi între mediile VO și VO, prin propunerea unei organizări holonice a VO ca subholoni ai mediului.
- propunerea, testarea și compararea abordărilor concurente pentru negocierea programării activităților în timpul ISP.

Arhitectura Preceptor poate servi ca bază pentru dezvoltarea unei platforme web cu scopul de a facilita planificarea personalizată a curriculumului și formarea profesională la cerere. Atât cursanții, cât și instructorii ar putea beneficia de pe urma utilizării platformei, iar extinderile viitoare ar putea servi și angajatorilor în căutarea candidaților pentru locuri de muncă. Ca atare, platforma are potențialul de a conecta instructorii și cursanții la cerințele locului de muncă și astfel crește rata de angajare.

Bibliografie

- [1] K. Almohammadi, H. Hagraş, D. Alghazzawi și G. Aldabbagh, "A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, nr. 1, pp. 47–64, 2017.
- [2] V. Buş, "Relația dintre siguranța psihologică, învățarea în echipă și performanță: O perspectivă longitudinală," în "Horia D. Pitariu" National Conference of Industrial and Organizational Psychology, Cluj-Napoca, 2016.
- [3] L. M. Camarinha-Matos și H. Afsarmanesh, "Virtual enterprise modeling and support infrastructures: applying multi-agent system approaches," în *ECCAI advanced course on artificial intelligence*, Springer, 2001, pp. 335–364.
- [4] N. Capuano și D. Toti, "Experimentation of a smart learning system for law based on knowledge discovery and cognitive computing," *Computers in Human Behavior*, vol. 92, pp. 459–467, 2019.
- [5] V. Caputi și A. Garrido, "Student-oriented planning of e-learning contents for Moodle," *Journal of Network and Computer Applications*, vol. 53, pp. 115–127, 2015.
- [6] C. Cooper și I. Rothmann, *Organizational and work psychology: Topics in applied psychology*. Routledge, 2013.
- [7] M. Cossentino, N. Gaud, S. Galland, V. Hilaire și A. Koukam, "A holonic metamodel for agent-oriented analysis and design," în *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Springer, 2007, pp. 237–246.
- [8] S. Galland, S. Rodriguez și N. Gaud, "Run-time environment for the SARL agent-programming language: the example of the Janus platform," *Future Generation Computer Systems*, vol. 107, pp. 1105–1115, 2020.
- [9] A.-F. Ionescu, "Designing Virtual Learning Systems: Current Trends and Evaluation," în *Proceedings of the 14th International Conference on Virtual Learning (ICVL)*, 2019, pp. 303–308.
- [10] A.-F. Ionescu, "E-learning as a Service: Benefits of the Semantic Web and SOA for Virtual Learning," în *Proceedings of the 14th International Conference on Virtual Learning (ICVL)*, 2019, pp. 401–407.
- [11] A.-F. Ionescu, "Multi-objective Evolutionary Algorithms: Decomposition Versus Indicator-Based Approach," în *Algorithms as a Basis of Modern Applied Mathematics*, Springer, 2021, pp. 69–85.



- [12] A.-F. Ionescu și D.-M. Popovici, "Applications of multi-agent systems in social sciences: virtual enterprises as an example," în *Models and Theories in Social Systems*, Springer, 2019, pp. 311–325.
- [13] A.-F. Ionescu și D. Sburlan, "AdABI: an Adaptive Assessment System Based on Bayesian Inference," în *15th International Scientific Conference on eLearning and Software for Education (eLSE)—New Technologies and Redesigning Learning Spaces*, 2019, pp. 288–295.
- [14] A.-F. Ionescu și R. Vernic, "MOSOSS: an adapted multi-objective symbiotic organisms search for scheduling," *Soft Computing*, vol. 25, nr. 14, pp. 9591–9607, 2021.
- [15] A. Kassambara, "rstatix: Pipe-friendly framework for basic statistical tests," R package version 0.6.0, 2020.
- [16] H. Mayr, "Teaching software engineering by means of a virtual enterprise," în *Proceedings Tenth Conference on Software Engineering Education and Training*, IEEE, 1997, pp. 176–184.
- [17] E. F. Moore, "The shortest path through a maze," în *Proc. Int. Symp. Switching Theory*, 1959, 1959, pp. 285–292.
- [18] A. J. Nebro, J. J. Durillo și M. Vergne, "Redesigning the jMetal multi-objective optimization framework," în *Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation*, 2015, pp. 1093–1100.
- [19] F. D. de la Peña, J. A. Lara, D. Lizcano, M. A. Martínez și J. Pazos, "A New Approach to Computing Using Informons and Holons: Towards a Theory of Computing Science," *Foundations of Science*, vol. 25, nr. 4, pp. 1173–1201, 2020.
- [20] N. Protogeros, *Agent and web service technologies in virtual enterprises*. IGI Global, 2007.
- [21] S. Rodriguez, N. Gaud și S. Galland, "SARL: a general-purpose agent-oriented programming language," în *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, IEEE, vol. 3, 2014, pp. 103–110.
- [22] UNESCO, *Beijing Consensus on Artificial Intelligence and Education: Outcome Document of the International Conference on Artificial Intelligence and Education "Plan Education in the AI Era: Lead the Leap"*, 2019.



Rezumat

Prezenta teză se concentrează pe învățarea virtuală personalizată la cerere, ca răspuns la o solicitare specifică de instruire emisă de un cursant. Teza este structurată astfel:

În capitolul 1, oferim o introducere în învățarea virtuală și o perspectivă critică asupra limitărilor sau a lacunelor din cercetarea actuală. O atenție specială este dedicată organizațiilor virtuale ca o potențială soluție pentru învățarea virtuală.

În capitolul 2, menționăm scopul tezei—de a proiecta, implementa și testa un mediu de tip piață pentru a asigura instruire individualizată la cerere pentru un cursant prin intermediul unei organizații virtuale (VO)—și derivăm obiective specifice.

Capitolul 3 prezintă metodologia de cercetare pe care am folosit-o în studiile noastre. În primul rând, oferim justificarea alegerilor noastre de proiectare în dezvoltarea euristicii propuse pentru rezolvarea problemei de selecție a partenerilor (PSP) multi-obiectiv, apoi încadrăm problema de selecție a instructorilor ca un caz particular de PSP și introducem arhitectura Preceptor pentru un mediu holonic pentru VO de tip piață care susține întregul ciclu de viață al unei VO de formare. În cele din urmă, folosind un studiu de caz, oferim un test al arhitecturii și o comparație a trei structuri organizatorice concurente care pot fi utilizate în timpul procesului de negociere a programului.

În capitolul 4 prezentăm mai întâi algoritmul nostru nou dezvoltat pentru instanțe PSP generice cu programarea activităților cu constrângeri de timp, buget, prioritate a activităților și resurse, Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS). Analizăm apoi studiul de caz, comparând trei structuri organizatorice diferite ale VO de formare în procesul de formare/reconfigurare a VO—heterarhie, ierarhie și holarhie. Rezultatele testului Friedman și testului pentru toate perechile Nemenyi-Wilcoxon-Wilcox au arătat că, în timp ce o structură organizatorică ierarhică poate duce la un timp de execuție mai scurt în comparație cu structurile descentralizate, o structură organizatorică complet holonică poate fi mai eficientă în ceea ce privește utilizarea procesorului.

O discuție a propunerilor și a rezultatelor noastre, precum și a limitărilor și a direcțiilor viitoare de cercetare, este prezentată în capitolul 5. Contribuțiile originale ale cercetării raportate sunt rezumate în capitolul 6, care încheie teza.