

# SISTEM FLEXIBIL PENTRU ACCESAREA PERSONALIZATĂ A SERVICIILOR DESTINATE PERSOANELOR VÂRSTNICE - WEBAGEING -

Victor Popa

Liliana Constantinescu

vpopa@ici.ro

lconst@ici.ro

*Institutul Național de Cercetare Dezvoltare în Informatică, ICI, București*

**Rezumat:** Accesarea serviciilor publice din diverse domenii incluzând: sănătate, programe sociale, taxe și impozite etc., este o provocare birocratică pentru toți cetățenii, dar persoanele vârstnice sunt cele mai afectate. Pentru a gestiona această stare de lucruri, agențiile guvernamentale, neguvernamentale și alte organizații oferă forme online pentru accesarea serviciilor dedicate acestui segment de populație. Luând în considerare aceste provocări, sistemul WebAgeing implementează o infrastructură de tip „open middleware”, bazată pe tehnologia semantică, capabilă să furnizeze servicii compuse personalizate, destinate vârstnicilor în concordanță cu profilul acestora. Pentru aceasta, se integrează serviciile autonome, eterogene și distribuite geografic, oferite de providerii de servicii (agențiile guvernamentale, neguvernamentale, alte organizații) cetățenilor vârstnici, utilizând un mecanism de înregistrare a serviciilor cu comunitățile de interes, bazat pe algoritmi de matching sintactic și semantic și se compun, în mod personalizat, serviciile providerilor, în scopul execuției cererilor adresate de persoanele vârstnice sistemului WebAgeing.

**Cuvinte cheie:** servicii personalizate compuse, înregistrare servicii cu comunități de interes, algoritmi de matching sintactic și semantic, profil utilizator, ontologia Comunitate, bază de cunoștințe, tehnologia informației și comunicațiilor (IT&C).

**Abstract:** Accessing public services from different domains including: healthcare, social programs, dues and taxes etc., represents a red tape challenge for all citizens, but the senior citizens are most affected. In order to manage this situation, the government and non-government agencies and other organizations offer online configurations for accessing services dedicated to this population segment. Envisaging these challenges, the WebAgeing system implements an “open middleware” type infrastructure based on semantic technology, capable of catering composed personalized services for senior citizens according to their profile. Therefore, we integrate the autonomous, heterogeneous services, distributed geographically furnished by service providers (government and non-government agencies, other organizations) for the elderly, using a recording mechanism of services with interest communities, reckoned on syntactic and semantic matching algorithms and providers’ services are composed in a personalized manner, with the purpose of executing the demands addressed by the elderly to the WebAgeing system.

**Key Words:** composed personalized services, service registration with interest communities, syntactic and semantic matching algorithms, user profile, Community ontology, database, information and communication technology (IT&C).

## 1. Introducere

Accesarea serviciilor publice din diverse domenii incluzând: sănătate, programe sociale, taxe și impozite etc., este o provocare birocratică pentru toți cetățenii, dar persoanele vârstnice sunt cele mai afectate. Pentru a gestiona această stare de lucruri, agențiile guvernamentale, neguvernamentale și alte organizații oferă forme online pentru accesarea serviciilor dedicate acestui segment de populație. Din nefericire, acest efort a avut efecte minore privind îmbunătățirea situației existente.

Luând în considerare aceste provocări, în cadrul proiectul WebAgeing se proiectează și implementează o infrastructură de tip „open middleware” bazată pe tehnologia semantică, ce va furniza servicii compuse personalizate vârstnicilor, în concordanță cu profilul acestora.

Pentru atingerea acestui obiectiv, sunt integrate servicii autonome, eterogene și distribuite geografic, oferite de providerii de servicii (agențiile guvernamentale, neguvernamentale, alte organizații), cetățenilor vârstnici, utilizând un mecanism de organizare a acestora în comunități de interes, de înregistrarea lor cu comunitățile de interes, utilizând algoritmi de matching sintactic și semantic, precum și compunerea, în mod personalizat a serviciilor providerilor, în scopul execuției cererilor adresate, de persoanele vârstnice, sistemului.

Componentele tehnologice cheie ale sistemului sunt:

1. O rețea de comunități, fiecare comunitate corespunzând unui domeniu de interes (sănătate, programe sociale, taxe și impozite etc.), utilizată la înregistrarea serviciilor oferite persoanelor vârstnice de către agențiile guvernamentale, neguvernamentale și alte organizații. Comunitățile sunt definite de furnizorii de comunități și includ colecții de operații generice, asigurate la operațiile concrete ale serviciilor Web, în momentul înregistrării serviciilor Web cu comunitățile respective. În scopul descoperirii comunităților atașate unor domenii de interes, sistemul implementează registrul de comunități. Pentru înregistrarea serviciilor furnizate de agențiile guvernamentale, neguvernamentale și de alte organizații cu comunitățile de interes (Health, Elderly etc.) este nevoie de un mecanism care să asigneze operațiile generice ale

comunităților la operațiile concrete ale serviciilor. Comunitățile sunt interconectate într-o rețea de comunități, pe baza relațiilor de înrudire dintre domeniile de interes atașate comunităților;

2. O abordare de compoziție a serviciilor Web în scopul furnizării de servicii personalizate, specifice persoanelor vârstnice, constând dintr-un selector de operații generice, un generator de servicii compuse personalizate și o mașină pentru execuția serviciilor personalizate generate;
3. Un management al metadatelor utilizate în definirea comunităților (ontologia Comunitate), definirea serviciilor Web (WSDL extins cu attribute semantice), înregistrarea serviciilor cu comunitățile de interes (ontologia Mapping), definirea cererilor și a profilului utilizator;
4. Sistemul demonstrează modul cum suportă modelarea comunităților și serviciilor, înregistrarea serviciilor cu comunitățile de interes, compunerea personalizată a serviciilor utilizând profilul utilizator și execuția serviciilor compuse. Necesitatea unui astfel de sistem este motivată, pentru diferitele categorii de utilizatori, astfel:
  - pentru agențiile și organizațiile care oferă servicii pentru persoanele vârstnice prin necesitatea organizării acestor servicii în comunități de interes,
  - pentru funcționarii publici, care lucrează cu persoane vârstnice, prin necesitatea rezolvării cererilor de servicii și programe, adresate de persoanele vârstnice,
  - pentru persoanele vârstnice care preferă să solicite servicii și programe, de la computerul din locuință, de necesitatea construirii de servicii personalizate compuse care prin execuția lor să furnizeze răspunsuri la cererile adresate,
  - pentru furnizorii de comunități prin necesitatea definirii unei ontologii care să servească drept template de definire a comunităților.

Pentru realizarea acestor obiective, WebAgeing se axează pe organizarea serviciilor destinate persoanelor vârstnice în comunități bazate pe domenii de interes, comunități ce sunt definite printr-un set de attribute semantice ce identifică comunitatea (categoria comunității, sinonimele comunității, specializările comunității). O colecție de operații generice, unde fiecare operație generică este identificată, de asemenea, prin attribute semantice (categoria operației, scopul operației, sinonimele operației, specializările operației), include parametrii de intrare-ieșire, reguli de eligibilitate ale operației etc. Definirea comunităților de interes pentru persoanele vârstnice se bazează pe analiza cerințelor acestora privind accesarea serviciilor, cât și pe diverse dicționare și taxonomii de termeni, existente în prezent: WordNet, NAICS (North American Industry Classification System), UNSPPSC (Universal Standard Products and Services Classification), Rossetta Net, cXML, EDI etc.

## 2. Obiective

Obiectivul principal al sistemului este construirea unei infrastructuri bazate pe servicii Web și pe tehnologii semantice, în scopul îmbunătățirii accesului la serviciile și programele destinate persoanelor vârstnice.

Obiectivele specifice, cuantificabile, sunt:

1. cerințele privind accesarea serviciilor și programelor destinate persoanelor vârstnice, prin care se analizează nevoile cetățeanului vârstnic în vederea accesării acestora și a programelor specifice;
2. Metodologia de Modelare a Serviciilor, instrumentele suport, care constă în modelarea serviciilor Web și a instrumentele suport. Aceasta se referă la modelarea serviciilor utilizând standardul WSDL extins cu attribute semantice și calitative;
3. Ontologia de Modelare a Comunităților, instrumentele suport, pentru modelarea comunităților utilizând attribute sintactice, semantice și operaționale;
4. metodologia de înregistrare a serviciilor Web cu comunitățile de interes, instrumentele suport pentru înregistrarea serviciilor cu comunitățile de interes, algoritmi de matching, utilizați în potrivirea operațiilor generice cu operațiile serviciilor, mediarea potrivirii parametrilor de intrare-ieșire ai operațiilor generice cu parametrii de intrare-ieșire ai operațiilor serviciilor;
5. metodologia de specificare a cererilor utilizator, instrumentele suport pentru specificarea cererilor utilizator;
6. metodologia de compunere și execuție a serviciilor, instrumentele suport privind compunerea personalizată a serviciilor, execuția serviciilor compuse, care se referă la algoritmi pentru generarea personalizată a serviciilor compuse, execuția serviciilor compuse generate, instrumentele suport;
7. prototipul WebAgeing, cu toate componentele, ce constituie o platformă completă, care permite providerilor de comunități să le definească, utilizând ontologia Comunitate, providerilor de servicii



să-și înregistreze serviciile cu comunitățile de interes, iar persoanelor vârstnice să-și specifice cererile; o implementare consistentă a prototipului pentru o platformă completă WebAgeing va fi produsă prin exploatarea pe cât posibil a tehnologiilor existente;

8. implementarea prototipului prin dezvoltarea unui test integrat implicând accesarea de servicii pentru populația vârstnică, care permite evaluarea nivelului de utilizare și acceptare a utilizatorului în temei de acoperire a nevoilor sale, precum și al ușurinței utilizării.

O etapă importantă în realizarea WebAgeing, prezentată în cadrul articolului, se concentrează pe analiza privind:

- identificarea cerințelor funcționale pentru ontologia Comunitate;
- identificarea comunităților de interes și cazuri de utilizare a acestora;
- identificarea cerințelor funcționale pentru înregistrarea serviciilor cu comunitățile de interes;
- identificarea cerințelor funcționale pentru execuția serviciilor compuse.

## 2.1. Analiza cerințelor privind modelarea comunităților

O ontologie este definită ca o specificație formală și explicită a unei conceptualizări acceptată în comun de mai mulți utilizatori. Ontologiile reprezintă instrumente cheie ce permit prelucrări și accesări ale datelor bazate pe semantică. Combinarea celor două concepte puternice (servicii Web și ontologii) a condus la apariția serviciilor Web semantice. Acestea au provocat o intensă activitate de cercetare atât în industrie, cât și în mediile academice privind descrierea, selecția, monitorizarea și compoziția lor.

Sistemul WebAgeing permite compunerea automată a serviciilor Web, oferite persoanelor vârstnice, și generează provocări privind înțelegerea completă a descrierii și organizării serviciilor Web, selectarea serviciilor relevante unei cereri utilizator, componența serviciilor Web partenere, precum și generarea descrierii serviciilor compuse.

Organizarea semantică a serviciilor Web este o cerință importantă pentru sistemele viitoare datorită creșterii rapide a numărului acestora. Furnizorii de servicii pentru persoanele vârstnice sunt localizați în diverse puncte geografice, iar răspândirea pe scară largă a acestora, dinamica și eterogenitatea lor diminuează înțelegerea semanticii și, astfel, îngreunează managementul lor. Pentru rezolvarea acestor aspecte, sistemul propune o metodă semantică pentru organizarea și descrierea serviciilor Web pentru persoanele vârstnice. Astfel, serviciile vor fi organizate în comunități plecând de la domeniul lor de interes. O comunitate este un container pentru o anumită arie de interes. Toate serviciile Web aparținând unei comunități au în comun aceeași arie de interes. Comunitățile furnizează descrieri ale serviciilor dorite fără a face referiri la serviciile Web actuale. Ontologia Comunitate este o ontologie metadată, ce furnizează concepte pentru descrierea comunităților. [4]

Comunitățile sunt create de către providerii de comunități (grupuri de agenții, organizații non-profit și alte organizații care au în comun un domeniu de interes) folosind ontologia Comunitate ca template de creare. Ca exemple de comunități se pot enumera:

- comunitatea serviciilor pentru bătrâni din sfera medicală,
- comunitatea serviciilor pentru bătrâni din sfera socială,
- comunitatea serviciilor pentru bătrâni din sfera asigurări etc.

Ontologia Comunitate, a cărei structură este prezentată în continuare, definește categoria comunității (domeniul de interes, sinonimele, specializările), operațiile generice, relațiile dintre operațiile generice, regulile de eligibilitate atașate operațiilor. Relațiile „pre-operation” și „post-operation” între operațiile generice ale comunităților se bazează pe legile și politicile care regularizează accesul la serviciile și programele destinate cetățenilor vârstnici. Aceste relații oferă mijloace pentru a defini procese ce includ operații generice aparținând la una sau mai multe comunități. Sistemul WebAgeing adoptă diagramele de activitate UML pentru a reprezenta grafic relațiile „pre-operation” și „post-operation”. Regulile de eligibilitate, atașate operațiilor, utilizează condiții logice, care includ variabile a căror asignare este făcută utilizând valori din profilul utilizator.



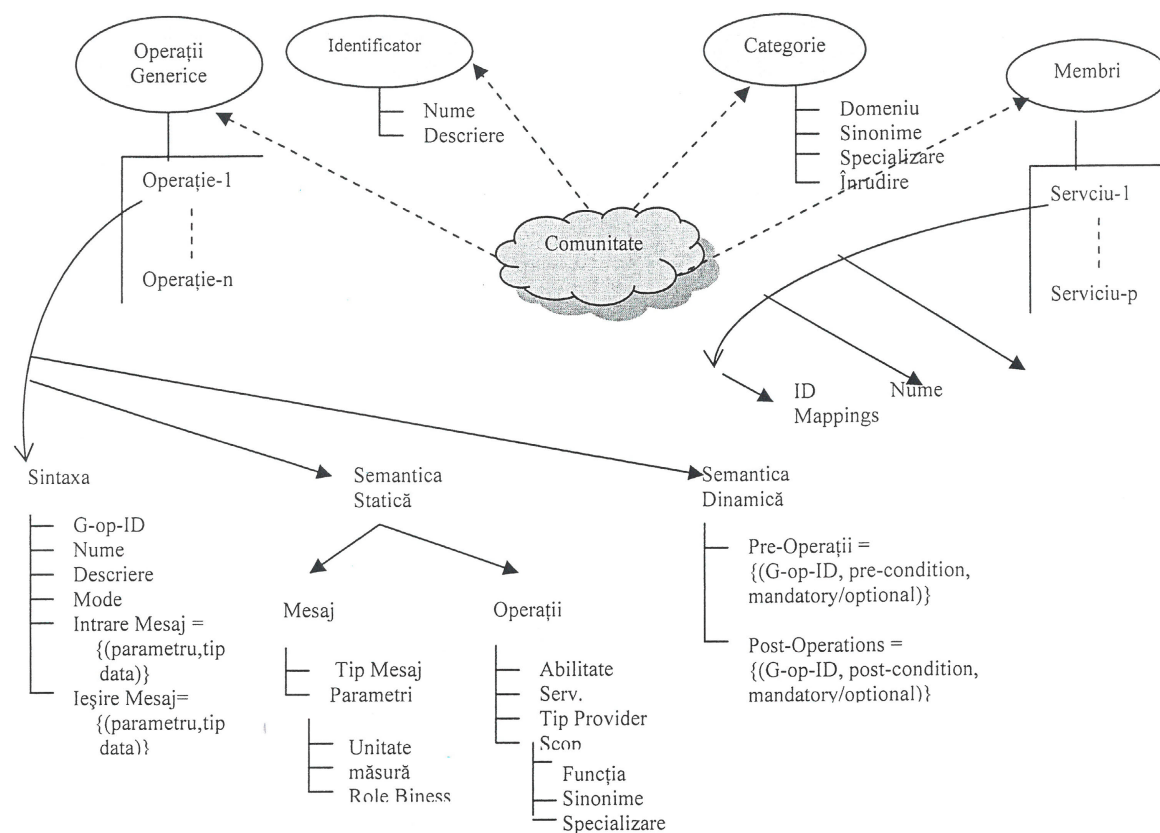


Figura 1. Structura Comunitate

### 2.1.1. Descrierea formală a comunităților și a operațiilor generice

În mod formal, o comunitate este definită prin tupletul:  $\langle \text{identifcator}, \text{categorie}, \text{Set de operații generice}, \text{Colecție de servicii web} \rangle$ .

Identificatorul reprezintă un nume unic și o descriere care prezintă sumar caracteristicile comunității. Categoria descrie aria de interes a comunității. Toate serviciile aparținând unei comunități au aceeași categorie. O categorie este accesată printr-o colecție de operații generice, care sunt operații abstracte ce sumarizează funcțiile majore ale serviciilor Web, membre ale comunității. Termenul abstract, utilizat mai sus, semnifică faptul că nu este furnizată nicio implementare a operației în corpul comunității, furnizorii de comunități oferind numai o interfață pentru fiecare operație generică. Aceste interfețe sunt utilizate și implementate prin membrii comunității. Execuția unei operații generice implică execuția unei operații actuale a unui serviciu membru ce implementează operația generică. Membrii unei comunități implementează una sau mai multe operații generice din comunitate.

Astfel, categoria unei comunități este formal definită prin tupletul:  $\langle \text{Domeniu}, \text{Sinonime}, \text{Specializări}, \text{Înrudire} \rangle$ .

Domeniul reprezintă aria de interes a comunității, atributul ia valori din taxonomiile rezultate din analiza cerințelor privind accesarea serviciilor Web. Comunitățile pot adopta diverse taxonomii pentru a specifica categoria lor. Spațiile de nume XML sunt utilizate pentru a prefixa categoriile cu taxonomiile în care sunt definite. Spațiile de nume furnizează o metodă pentru calificarea numelor elementelor și a atributelor utilizate în documentul XML, prin asocierea acestora cu referințe URI. Sinonimele conțin o colecție de nume alternative pentru domeniu, de exemplu pentru domeniul GrijăSănătate se mai poate folosi și termenul OcrotireSănătate, valorile asignate acestui atribut sunt încărcate tot din una din taxonomiile specificate. Specializarea este o colecție de caracteristici ale domeniului comunității, de exemplu, termenii asigurare și bătrâni pot fi specializări ale domeniului GrijăSănătate. Aceasta semnifică faptul că respectiva comunitate furnizează asigurări de sănătate pentru bătrâni. Comunitățile nu sunt, în general, independente, ele sunt interconectate prin diverse relații de înrudire, aceste relații fiind specificate în atributul Înrudire. Astfel, pentru o comunitate dată, atributul Înrudire specifică lista comunităților înrudite cu aceasta, în sensul că, prin combinarea acestei comunități cu oricare din lista Înrudire, vor rezulta servicii noi utile persoanelor vârstnice. Ca un exemplu de înrudire să considerăm



comunitățile Transport și Recreere. Aceste comunități sunt înrudite deoarece prin combinarea lor se pot accesa serviciile de transport în scopul deplasării persoanelor vârstnice în locuri de recreere.

Operațiile generice ale comunităților sunt definite printr-un set de atribute funcționale, care descriu caracteristicile sintactice. Atributele sintactice reprezintă structura unei operații generice, de exemplu, lista parametrilor de intrare-ieșire a unei operații care definește mesajele operației. Atributele semantice se referă la înțelesul operației și al mesajelor de intrare-ieșire ale acesteia. Există două tipuri de atribute semantice: statice și dinamice.

Atributele statice descriu aspectele necomputaționale ale operației și ele sunt independente de execuția operației. Un exemplu de atribut static este categoria unei operații. Atributele dinamice descriu caracteristicile computaționale ale operației. Aceste atribute se referă, în general, la constrângerile sub care se execută operațiile generice. Un exemplu de atribut dinamic este logica de business a operației. În momentul definirii unei comunități, furnizorul comunității asignează valori la diverse atribute ale comunității.

Furnizorii de servicii pot selecta o comunitate de interes (bazându-se pe categoria acesteia) și înregistra serviciile lor cu acea comunitate. Procesul de înregistrare presupune construirea unei mapări între unele operații generice și unele operații actuale ale serviciilor Web. Maparea conține perechi formate dintr-o operație generică și o operație concretă. De asemenea, pentru fiecare pereche se specifică corespondența între parametrii de intrare ai operației generice și parametrii de intrare ai operației actuale și corespondența între parametrii de ieșire ai operației generice și parametrii de ieșire ai operației actuale. De remarcat faptul că, într-o mapare, operația generică poate să conțină mai mulți parametri de intrare decât operația actuală, deoarece parametrii existenți în plus în operația generică nu sunt utilizați în operația concretă. Reciproc, nu se poate ca într-o mapare operația concretă să conțină mai multe argumente de intrare decât operația generică, deoarece operația concretă nu ar putea fi executată. De asemenea, numărul parametrilor de ieșire din operația generică poate fi mai mic decât numărul parametrilor de ieșire din operația concretă. Reciproc nu este valabil. Într-o pereche formată dintr-o operație generică și una abstractă este necesar ca operațiile să fie compatibile din punct de vedere semantic (algoritmul de matching va verifica acest lucru și va calcula gradul de compatibilitate). De asemenea, algoritmul de matching va verifica în ce măsură corespondențele între parametrii operațiilor generice și cei ai operațiilor concrete includ parametrii compatibili din punct de vedere sintactic și semantic. [5]

După cum s-a specificat anterior, operațiile generice sunt descrise prin atribute sintactice, semantice statice și dinamice.

Atributele sintactice sunt utilizate la două niveluri: al operațiilor și al mesajelor. Nivelul operațiilor conține atribute care descriu operațiile, iar nivelul mesajelor utilizează atribute sintactice pentru descrierea mesajelor (nume, tip etc.).

Fiecare operație generică are un *identificator* care este unic în spațiul operațiilor generice, un *nume* și un text *descriptiv*. De asemenea, operațiile generice conțin atributul *mode* care descrie ordinea de trimitere și primire a mesajelor. Acesta arată dacă operația inițiază interacțiuni ori răspunde simplu la invocările din alte servicii și poate avea două valori posibile Inp/Out și Out/Inp. Una dintre aceste valori este asignată atributului de către furnizorul de comunitate. Operațiile de tip Inp/Out mai întâi primesc un mesaj de intrare de la un client, prelucrează mesajul și apoi returnează ieșirea către client. Operațiile de tipul Out/Inp mai întâi trimit un mesaj de ieșire către server și apoi primesc un mesaj de intrare ca rezultat. Un exemplu de operație de tip Inp/Out apelată de persoanele vârstnice este operația „înregistrare-vizită-club”. Un exemplu de operație de tip Out/Inp este operația „expirareCard” care notifică unei persoane vârstnice expirarea cardului financiar.

Fiecare mesaj de intrare sau ieșire conține unul sau mai mulți parametri, fiecare parametru având un *nume* și un *tip de date* care specifică domeniul valorilor ce poate fi asignat parametrului. Pentru specificarea tipului de date, pentru fiecare parametru, se utilizează formalismul XML schema, acestea fiind definite, de asemenea, de furnizorii de comunități.

În afară de atributele sintactice, utilizate în descrierea operațiilor generice, se folosesc atribute semantice statice, al căror conținut este independent de execuția operației. Atributele semantice specifică atât semantica operației (ce face operația), cât și semantica mesajelor de intrare-ieșire.

Semantica statică a unei operații generice este definită prin următoarele atribute: *serviabilitate*, *tipul furnizorului comunității*, *tipul consumatorului*, *scopul*, *categoria*.

Atributul *serviabilitate* dă tipul de asistență furnizat de operație („informațional”, „social”, „cash” etc.). O operație poate fi suportată prin unul sau mai multe tipuri de furnizori („guvernamentali”, „neguvernamentali”). Tipul consumatorului specifică grupul de bătrâni care consumă această operație.



Fiecare operație îndeplinește o anumită funcționalitate pentru o anumită arie de interes. Acest lucru este specificat utilizând atributele scop și categorie. Toate operațiile generice ale unei comunități moștenesc categoria comunității în care sunt definite. Atributul scop specifică scopul operației în cadrul comunității. El este definit prin trei atribute: funcția, sinonimele și specializarea. Funcția descrie funcționalitatea business oferită de operație (înregistrare, eligibilitate etc.). Sinonimele și specializările sunt definite la fel ca în cazul comunităților.

Mesajele operațiilor trebuie să fie, de asemenea, descrise din punct de vedere semantic pentru a putea fi înțelese atât de furnizorii de servicii, cât și de consumatori. De asemenea, mesajele trebuie înțelese de algoritmul de matching în momentul înregistrării serviciilor cu comunitățile de interes. Fiecare mesaj are asociat un tip care dă descrierea generală a conținutului reprezentat prin mesaj (Comanda, Factura etc.). Ontologiile verticale pot fi utilizate pentru descrierea tipurilor mesajelor. Tipul mesajelor nu captează întreaga semantică a mesajului, fiind necesară descrierea semantică a fiecărui parametru component. Parametrii mesajelor sunt reprezentați prin atributele nume și data type, dar aceste atribute nu captează semantică parametrilor. În scopul modelării semanticii parametrilor au fost introduse atributele „role business” și „unit”. Atributul „role business” dă tipul informației conținută în parametru. De exemplu, parametrul adresa poate însemna o stradă și un număr sau poate însemna un oraș și un cod zip. Parametrul preț poate însemna prețul total sau prețul fără taxe. Atributul „role” ia valori dintr-o taxonomie predefinită, astfel fiecare parametru are un înțeles bine stabilit conform acestei taxonomii. Un exemplu de taxonomie ce poate fi utilizată este dicționarul de termeni de business RossetaNet. Conform acestui dicționar, dacă parametrul preț semnifică un preț total, atunci rolul său în dicționarul de business va fi „preTotal”. Pentru o bună flexibilitate, diferite comunități pot adopta diferite dicționare de business. Din această cauză, atributul role este prefixat cu un spațiu de nume xml. Atributul „unit” se referă la unitatea de măsură în care conținutul parametrului este furnizat. Astfel, acest parametru poate să ia valori ca: euro, dolar, kilogram etc. WebAgeing utilizează unitățile de măsură standard pentru a asigura valori acestui parametru, care sunt stabilite de furnizorii de comunități.

### 2.1.2. Descrierea semanticii dinamice

Semantica dinamică permite descrierea atributelor legate de execuția operațiilor generice, care trece prin următoarele stări: gata de execuție, start, activ și sfârșit execuție.

O operație se găsește în „starea gata de execuție” dacă nu a fost emisă o cerere pentru execuția ei. O operație se găsește în „starea start” dacă un mesaj este trimis acestei operații, iar operația are modul Inp/Out sau dacă un mesaj de ieșire a fost trimis de operație și aceasta are modul Out/Inp. O operație se găsește în „starea activ” dacă este în curs de execuție pentru a prelucra cererea dată. După prelucrarea unei cereri, operația trece în starea finală când se returnează rezultatul cererii către client dacă operația este în mod Inp/Out sau când mesajul de intrare este primit de la server dacă operația este în modul Out/Inp. Execuția unei operații poate implica trecerea printr-un întreg proces (dictat de legile în vigoare) ce implică execuția unor operații generice, numite preoperații. O operație generică poate avea zero sau mai multe preoperații generice, ceea ce înseamnă că invocarea operației respective va putea să fie făcută numai după ce preoperațiile sale au fost executate. De asemenea, o operație generică poate fi o preoperație pentru mai multe operații generice.

Dacă o condiție este specificată pentru o operație generică, atunci operația respectivă se poate executa numai dacă toate preoperațiile sale au fost executate și condiția este adevărată. De remarcat faptul că această condiție este un predicat ale cărui intrări sunt parametrii de intrare-ieșire din preoperații și/sau din profilul utilizator. Dacă nu este specificată o condiție, atunci se consideră una implicită, care are valoarea true întotdeauna. Execuția anumitor preoperații pentru o operație generică dată poate fi obligatorie sau facultativă. Acest lucru se specifică utilizând predicatul „mandatory” atașat fiecărei preoperații. Acest predicat este adevărat când execuția preoperației respective este obligatorie. În cazul în care execuția este facultativă, predicatul are valoarea false.

În unele situații, terminarea execuției unei operații generice implică execuția altor operații numite postoperații ale operației generice. O operație poate fi „postoperație” pentru una sau mai multe operații generice. Postoperația se execută când cel puțin una din operațiile generice a fost realizată și condiția corespunzătoare este adevărată. La fel ca în cazul unei preoperații, execuția unei postoperații poate fi obligatorie sau facultativă, așa cum specifică valoarea de adevăr a predicatului mandatory. Dacă valoarea de adevăr este true, atunci execuția condiției este obligatorie în condițiile în care cel puțin o operație generică a fost executată, iar condiția atașată este adevărată. Preoperațiile și postoperațiile furnizează mijloace de a defini procese intra- și intercomunități. În cazul e-government, aceste procese sunt dictate



de regularizările și legile date de guvern. Pentru reprezentarea acestor procese, se pot utiliza formalisme ca Petri-nets pi-calcul, diagrame de stare etc.

Reprezentarea utilizată în cadrul proiectului se bazează pe diagrame grafice de activități pentru că acestea sunt ușor de utilizat și destul de simple pentru a modela procese de business. Diagramele de activități vizualizează grafic ordinea de execuție a activităților în cadrul procesului. În abordarea noastră, activitățile unei diagrame sunt operații generice din una sau mai multe comunități. Dacă diagrama include numai operații generice dintr-o singură comunitate, ea reprezintă imaginea grafului compus din preoperațiile și postoperațiile existente în comunitatea respectivă. Dacă diagrama cuprinde operații din mai multe comunități, aceste operații vor fi prefixate cu Id-ul comunității de care aparțin. Post/Pre între două comunități se definesc în cadrul relației *Înrudire*. Dacă între două comunități nu există o relație de înrudire, atunci între operațiile lor nu pot exista relații post/preoperații. În afară de operațiile generice ale comunităților, o diagramă de activități include noduri de decizie unde sunt memorate condițiile atașate relațiilor pre/postoperații. O diagramă de activități în afară de nodurile de activități și cele de decizie conține un nod de start și unul sau mai multe noduri de stop. Figura următoare ilustrează un exemplu de diagramă de activități, utilizată pentru frecventarea unui club de recreere. Diagrama include trei operații generice, două noduri de decizie, un nod de start și două noduri de stop. Din diagramă se observă faptul că invocarea de către o persoană vârstnică a operației generice TransportClub implică execuția preoperațiilor de înregistrare a persoanei ca membru al clubului și a operației generice de planificare a vizitei la club.

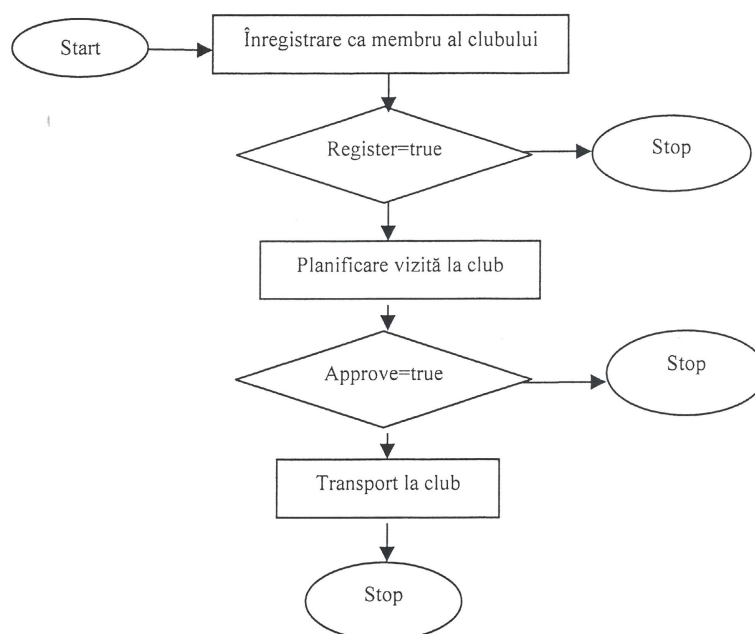


Figura 2. Exemplificare diagramă de activități

## 2.2. Analiza cerințelor privind modelarea serviciilor Web. Ontologia de modelare a serviciilor Web

Serviciile Web necesită descrieri astfel încât consumatorii (persoane vârstnice, programe) să le poată înțelege trăsăturile și să învețe cum să interacționeze cu acestea. Un limbaj în curs de dezvoltare pentru descrierea trăsăturilor operaționale ale serviciilor Web este limbajul WSDL (Web Service Description Language), limbaj în curs de standardizare de către consorțiul W3C. Pentru înregistrarea serviciilor Web cu comunitățile de interes este necesar ca pe lângă atributele ce descriu sintaxa să existe și atribute care descriu semantica. Aceste atribute sunt folosite în algoritmii de matching în timpul înregistrării serviciilor cu comunitățile de interes.

Ontologia propusă este redată în continuare folosind un grafic direcționat. Nodurile reprezintă conceptele ontologiei și includ conceptele WSDL (exemplu: Binding, input, output etc.) și sunt cele cu contur desenat prin linie discontinuă. Nodurile desenate cu linie continuă se referă la atributele semantice ce extind limbajul WSDL. Arcele reprezintă relațiile între conceptele ontologiei. Ele sunt adnotate cu cardinalitatea relației corespunzătoare. Astfel, un serviciu are una sau mai multe operații. O operație are cel mult un mesaj de intrare/ieșire. Un serviciu Web este definit prin instanțierea fiecărui concept din



ontologia specificată. Conceptele semantice Categorie, Scop au același înțeles ca pentru ontologia Comunitate, descrisă anterior. [2]

Un serviciu Web este identificat printr-un nume și o descriere textuală ce rezumă trăsăturile serviciului. Interacțiunile cu serviciul respectiv trebuie să fie conforme cu valoarea atributului Binding. Binding-ul definește formatul mesajului și detaliile de protocol pentru mesaje și operațiunile de serviciu. Binding-ul ia valori ca: SOAP (Simple Object Access Protocol), HTTP Get/Post și MIME (Multipurpose Internet Mail Extensions). Un serviciu poate avea mai multe binding-uri asociate lui. Pentru fiecare serviciu Web avem asociate un scop și o categorie.

Fiecare atribut Scop se referă la funcționalitatea de business, oferită de o operație a serviciului. Categoria descrie domeniul de interes al serviciului și include categoriile tuturor operațiilor furnizate de serviciu, cât și o categorie proprie, iar domeniul de interes al serviciului compus poate fi diferit de domeniile componente.

De remarcă faptul că, în ontologia de descriere a serviciilor Web, descrisă în continuare, fiecare operație are categoria sa, ceea ce nu se întâmplă pentru ontologia Comunitate, unde toate operațiile moștenesc categoria comunității. Acest lucru este explicabil prin faptul că un serviciu Web poate fi un serviciu compus, caz în care el implementează operații din mai multe domenii de interes. Acest lucru implică posibilitatea ca un serviciu Web să se înregistreze cu una sau mai multe comunități de interes, dacă operațiile sale au diferite categorii.

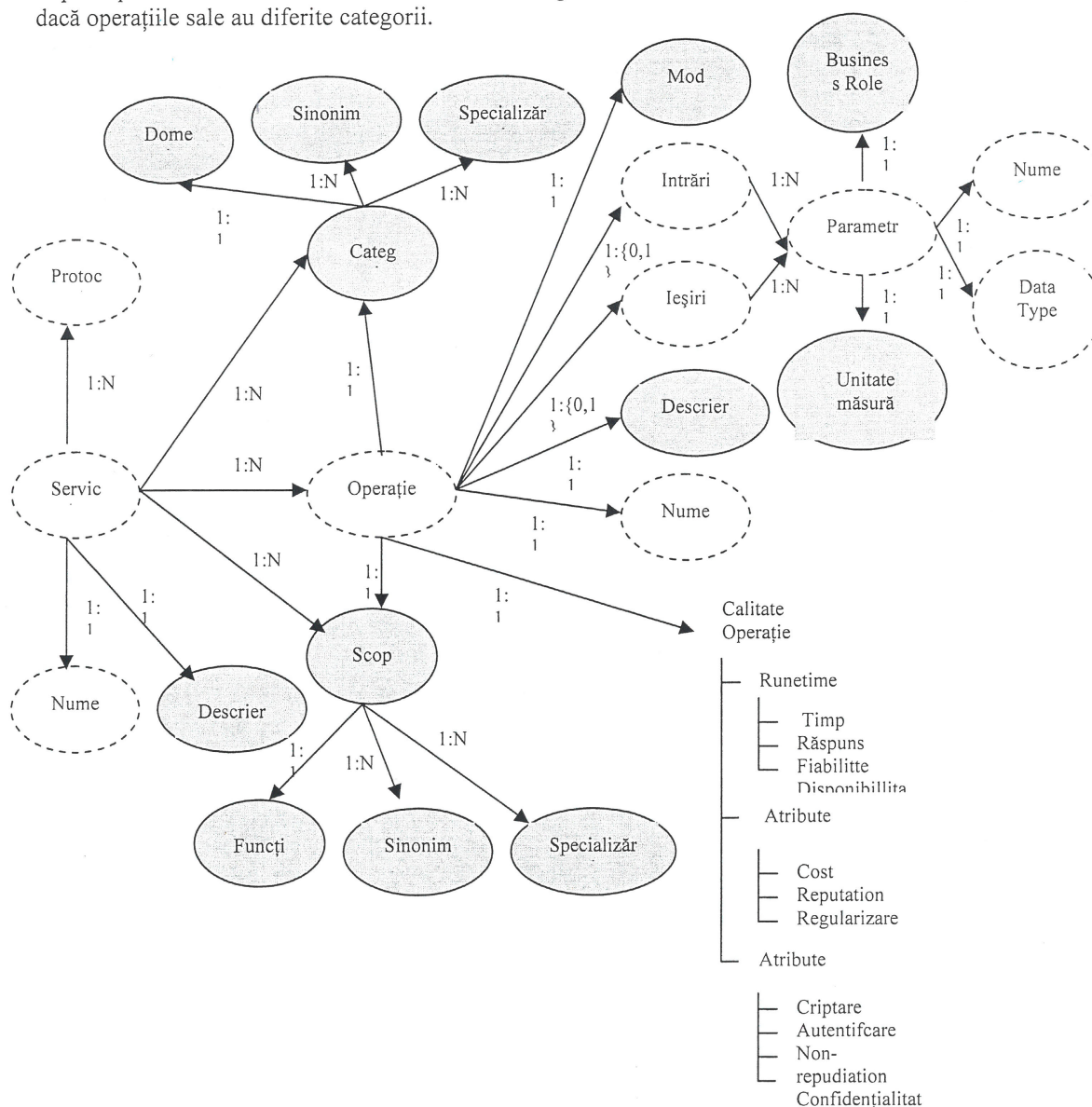


Figura 3. Ontologia de descriere a serviciilor Web



La fel ca în cazul ontologiei Comunitate, fiecare operațiune are cel mult un mesaj de intrare/ieșire. Operațiile de notificare nu au decât mesaje de ieșire. Fiecare mesaj conține unul sau mai mulți parametri (denumiți părți în WSDL). Un parametru are un nume și un tip de informație. Tipul parametrului îi conferă marja de valori, care îi poate fi atribuită. S-a utilizat schema XML pentru specificarea tipurilor mesajelor. Tipurile „built-in” sunt predefinite în schema de specificare XML. Ele pot fi ori primitive, ori derivate. Spre deosebire de tipurile primitive (de exemplu: string, decimal), tipurile derivate sunt definite în termeni de alte tipuri, de exemplu, integer este derivat din tipul primitiv decimal.

În timp ce tipurile de date sunt importante pentru matching-ul sintactic al parametrilor mesajelor, ele nu cuprind semnificativele acelor parametri. De exemplu, prețul parametrului poate fi în dolari americani, yen sau euro. În plus, poate reprezenta un preț total sau un preț fără taxe. Pentru a modela asemenea constrângeri, ontologia de mai sus utilizează două atribute semantice: unitate de măsură și rolul de business al fiecărui parametru. Rolul de business oferă semantica parametrului corespunzător și ia valori din taxonomia definită pentru rolul de business. Fiecare parametru trebuie să aibă o semantică bine definită conform taxonomiei. Un exemplu pentru o asemenea taxonomie este dicționarul de business RosettaNet, ce conține un vocabular comun ce poate fi utilizat pentru a descrie proprietățile de business. De exemplu, dacă parametrul „preț” are rolul „prețExtins” (definit în dicționarul RosettaNet), atunci parametrul preț reprezintă un „preț total pentru cantitatea produsului”. Din motive de flexibilitate, serviciile Web pot adopta diferite taxonomii pentru a specifica rolul parametrilor lor. Prefixarea numelui taxonomiei cu un spațiu de nume XML este un artificiu des folosit în practică.

Serviciile Web pot furniza operațiuni „similare” în ceea ce privește atributele „mode”, „mesaj”, „scop” și „categorie”. Astfel, la selectarea serviciilor, sunt luate în considerare proprietățile calitative ale acestora, care ajută utilizatorii la selectarea serviciilor potrivite.

În mod formal, un serviciu este definit prin tupletul: (Descriere, Opi, BindingSi, Scop, Categorie), unde:

- *Descriere*, reprezintă un text ce rezumă trăsăturile serviciului;
- *Opi* este un set de operații furnizate de WSJ;
- *BindingSi* reprezintă setul de protocoale suportate de WSJ;
- $Scop = \{Scopul_i(op_i) \mid op_i \in Opi\}$  este setul tuturor scopurilor operațiilor componente;
- $Categorie = \{Categorie(Op_i) \mid op_i \in Opi\} \cup \{categoria\ serviciului\}$ .

Din definiția serviciului Web, rezultă că un serviciu Web poate avea mai multe domenii, de exemplu, serviciile compuse conțin grupe de operații care pot aparține de diverse domenii de interes, diferite de domeniul de interes al serviciului. În cadrul serviciilor simple, toate operațiile au același domeniu de interes identic cu cel al serviciului.

### 2.3. Analiza cerințelor privind înregistrarea serviciilor cu comunitățile de interes

Algoritmii utilizați, atunci când un serviciu simplu sau compus trebuie înregistrat cu anumite comunități de interes, iau în considerare atât proprietățile sintactice, cât și cele semantice. De exemplu, ar fi dificil să se stabilească un mapping între o operație a unei comunități și o operație a unui serviciu Web dacă nu există nici o potrivire între parametrii celor două operațiuni (tipuri de informație, număr de parametri). În acest context, se identifică două seturi de reguli privind înregistrarea serviciilor Web cu anumite comunități: sintactice și semantice. [1]

Regulile sintactice compară atributele „mode” pentru fiecare pereche (P) formată dintr-o operație generică și o operație concretă, numărul de parametri ai mesajelor de intrare/ieșire și tipul de date pentru fiecare parametru. Regulile semantice compară pentru fiecare pereche (P) de operații domeniile de interes, sinonimele domeniilor, specializările domeniilor, funcțiile operațiilor, sinonimele funcțiilor, specializările funcțiilor, tipul mesajelor, tipul de date al parametrilor, rolul business al parametrilor, unitățile de măsură ale parametrilor.

Interacțiunile între serviciile Web presupun schimburi de mesaje. Un mesaj este compus din unul sau mai mulți parametri, fiecare conținând un anumit tip de date. Astfel, este important de verificat dacă tipurile de date pentru parametrii de intrare/ieșire ai serviciilor sunt compatibile cu tipurile de date ai parametrilor de intrare/ieșire ai comunității cu care se înregistrează. Ca metode primare de compatibilitate a tipurilor de informații, se consideră compatibilitatea directă și indirectă. Doi parametri sunt direct compatibili dacă au același tip de date. Doi parametri sunt indirect compatibili dacă tipul de date al unuia este derivat din tipul de date al celuilalt.



Compatibilitatea tipului de date se extinde și la mesaje, după cum urmează: un mesaj de intrare  $M$  al unei operații aparținând unui serviciu Web este compatibil din punct de vedere al tipului de date cu un mesaj  $M'$  al unei operații generice, aparținând unei comunități, dacă fiecare parametru din  $M$  este direct sau indirect compatibil cu anumiți parametri din  $M'$ . Nu toți parametrii lui  $M'$  necesită să fie mapați la parametrii din  $M$ . Modalitatea rațională care explică acest fapt, este că un mesaj de intrare pentru o operație a unui serviciu poate folosi subset de parametri dintr-un mesaj al unei operațiuni generice. De asemenea, pentru ca doi parametri  $p$  și  $p'$  aparținând mesajelor  $M$  respectiv  $M'$ , să fie compatibili, trebuie să aibă semantici compatibile. De exemplu, un preț total nu este compatibil cu un preț parțial. În mod similar, un preț exprimat în dolari americani nu este compatibil cu un preț exprimat în yen japonezi. În final, se procedează analog cu mesajele de ieșire, singura diferență în acest caz este că mesajul de ieșire al unei operații concrete poate avea mai mulți parametri decât mesajul de ieșire al operației generice cu care se înregistrează.

Pentru înregistrarea unei operații  $Ow$  a unui serviciu Web cu o operație generică  $Og$  aparținând unei comunități date, trebuie verificată compatibilitatea domeniilor și compatibilitatea scopurilor. Verificarea compatibilității domeniilor celor două operații presupune verificarea următoarelor condiții:

- Condiția 1: dacă aceste operații au același domeniu (reamintim că toate operațiile unei comunități moștenesc domeniul comunității de care aparțin) sau dacă domeniului operației concrete aparține sinonimelor domeniului operației generice sau dacă domeniul operației generice  $Og$  aparține domeniului sinonimelor operației concrete  $Ow$  sau dacă sinonimele domeniului operației  $Og$  nu sunt disjuncte de sinonimele domeniului operației  $Ow$ ;
- Condiția 2: specializările domeniului operației generice  $Og$  sunt incluse în specializările domeniului operației concrete  $Ow$ ;

Verificarea compatibilității scopurilor celor două operații presupune verificarea următoarelor condiții:

- Condiția 1: dacă aceste operații au aceeași funcție sau dacă funcția operației concrete aparține sinonimelor funcției operației generice sau dacă funcția operației generice  $Og$  aparține sinonimelor funcției operației concrete  $Ow$  sau dacă sinonimele funcției operației  $Og$  nu sunt disjuncte de sinonimele funcției operației  $Ow$ ;
- Condiția 2: specializările funcției operației generice  $Og$  sunt incluse în specializările funcției operației concrete  $Ow$ .

Vom spune că un serviciu Web  $W$  poate înregistra o operație  $a$  sau  $Ow$  cu o operație generică  $Og$  a comunității  $C$  dacă cele două operații sunt compatibile din punct de vedere sintactic și semantic, conform definițiilor de mai sus.

Ca un prim exemplu, operația generică „aplicaPentruAsigurare” nu este semantic compatibilă cu operația concretă „calculeazaRata” deoarece prima operație aparține domeniului ‘asigurări’, iar cea de a doua operație aparține domeniului „împrumut bancar”.

Ca un al doilea exemplu, operația generică ‘sendPrice’, având funcția „cerere preț bilet tratament”, nu este compatibilă cu operația concretă ‘sendInformation’, având funcția „solicitare informații centre balneare”, deoarece, deși aparțin aceluiași domeniu „grija pentru sănătate”, ele au funcții de business diferite.

Procesul de înregistrare a serviciilor Web cu comunitățile de interes cuprinde următoarele activități:

- definirea comunităților de către furnizorii de comunități utilizând ontologia Comunitate descrisă mai sus și înregistrarea acestora în registrul UDDI;
- definirea serviciilor Web de către furnizorii de servicii pentru persoanele vârstnice și înregistrarea acestora în registrul UDDI;
- înregistrarea serviciilor cu comunitățile de interes, utilizând regulile de verificare compatibilitate, descrise mai sus; dacă descrierea unui serviciu este modificată sau ștearsă în UDDI, atunci înregistrările serviciului cu comunitățile de interes trebuie actualizate.

Figura următoare ilustrează arhitectura utilizată de sistemul WebAgeing în procesul de înregistrare a serviciilor cu comunitățile de interes:



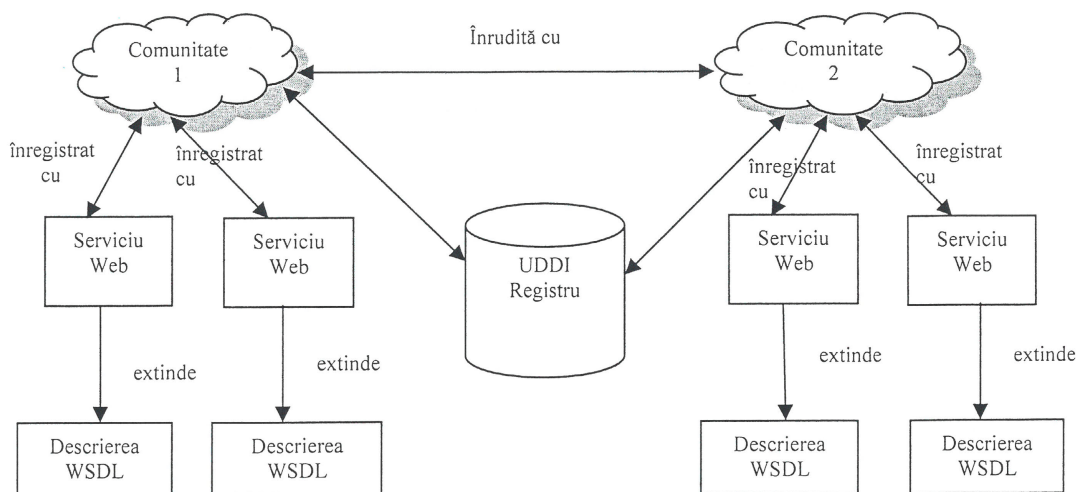


Figura 4. Arhitectura privind înregistrarea serviciilor cu comunitățile de interes

### 2.3.1. Utilizarea standardului WSDL extins pentru reprezentarea cunoștințelor

Serviciile Web cât și comunitățile sunt descrise utilizând structurile sintactice din WSDL la care se adaugă atribute semantice pentru:

- identificarea serviciului (categorie),
- identificarea operațiilor (scop),
- atașate mesajelor și parametrilor,
- calitative (securitate, preț, timp de răspuns etc.) atașate serviciilor.

Comunitățile sunt consemnate în registrul Comunităților utilizând atributele sintactice (ID comunități) cât și atributele semantice (Categorie, Sinonimele, Specializările) în scopul descoperirii eficiente în momentul construirii serviciilor compuse personalizate, utilizate pentru a furniza răspunsuri la cererile utilizator. Comunitățile înrudite sunt conectate între ele, iar serviciile Web descrise folosind WSDL extins se înregistrează cu comunitățile de interes. Figura următoare ilustrează modul de utilizare a fișierelor WSDL extinse cu atribute semantice pentru descrierea comunităților, serviciilor și a informațiilor privind înregistrarea serviciilor cu comunitățile de interes. Exemplul din figura de mai jos descrie comunitatea „GrijaPentruSănătate” având o singură operație generică „aplicaPentruAsigurare” care are ca intrare țara de destinație și perioada de deplasare, iar ca ieșire documentul „Asiguarare-Medicală”.

```

*****      Descrierea spațiilor de nume      *****
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link">
*****      Descrierea tipurilor de date      *****
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">

```



```

        <s:element name="InputMessag">
            <s:complexType>
                <s:sequence>
                    <s:element minOccurs="1" maxOccurs="1" name="tara" type="s:string"/>
                    <s:element minOccurs="1" maxOccurs="1" name="perioada"
type="s:integer"/>
                </s:sequence>
            </s:complexType>
        </s:element>
        <s:element name="OutputMessage">
            <s:complexType/>
        </s:element>
    </s:schema>
</wsdl:types>
***** Descrierea mesajelor *****
<wsdl:message name="intrare" Type="formular informații">
    <wsdl:part name="part1" element="tns:InputMessge" role="formular informații"/>
</wsdl:message>
<wsdl:message name="iesire" Type="Asigurare medicală">
    <wsdl:part name="part1" element="tns:OutputMessage" />
</wsdl:message>
***** Descrierea operațiilor generice *****
<wsdl:portType name="FileServiceSoap">
    <wsdl:operation name="operatia1" documentation="operație solicitare asigurare medicală">
        <wsdl:purpose function="aplicaPentruAsigurare" specializarea="solicitare adeverință"
sinonime=""/>
        <wsdl:input message="tns:intrare"/>
        <wsdl:output message="tns:iesire"/>
    </wsdl:operation>
</wsdl:portType>
***** Descriere binding-uri utilizate în serviciile Web *****
<wsdl:binding name="binding1" type="tns:FileServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http">
        <wsdl:operation name="operatia1" price="20" time="30" security="yes">
            <soap:operation soapAction="http://tempuri.org/operatie1" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
<wsdl:binding name="binding2" type="tns:FileServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="writeLine" price="10" time="40" security="yes">
            <soap:operation soapAction="http://tempuri.org/operatie1" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>

```



```

        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
***** Descriere comunitate și servicii înregistrate *****
<wsdl:community name="Community-1" documentation="utilizată în">
    <wsdl:category domain="GrijaSanatate" specializare="asigurari" sinonime="" />
    <wsdl:port name="port1" binding="tns:binding1">
        <soap:address location="http://localhost:8080/HelloWorldService/File1.asmx"/>
    </wsdl:port>
    <wsdl:port name="port2" binding="tns:binding2">
        <soap:address location="http://localhost:8080/HelloWorldService/File2.asmx"/>
    </wsdl:port>
</wsdl:community>
***** Descriere relație înrudire cu alte comunități *****
<plnk:Inrudire>
    <plnk:partner taxonomy="external"/>
    <plnk:partner taxonomy="taxonomy1"/>
    <plnk:partner taxonomy="taxonomy2"/>
</plnk:Inrudire>

```

Figura 5. Exemplu de utilizare a fișierelor WSDL extinse pentru reprezentare comunități și înregistrare servicii Web

### 2.3.2. Utilizarea abordării relaționale pentru reprezentarea cunoștințelor

Deși utilizarea fișierelor WSDL extinse pentru descrierea comunităților, serviciilor și informațiilor de înregistrare a serviciilor cu comunitățile de interes oferă un mijloc compact de reprezentare a cunoștințelor, este posibilă utilizarea modelului relațional ca formalism de reprezentare a informației, așa cum este ilustrat și în figura următoare. Comparând cele două formalisme de reprezentare, reprezentarea bazată pe standardul WSDL pare cea mai potrivită pentru scopul sistemului, unde este necesar ca informația să fie afișată într-un mod compact.

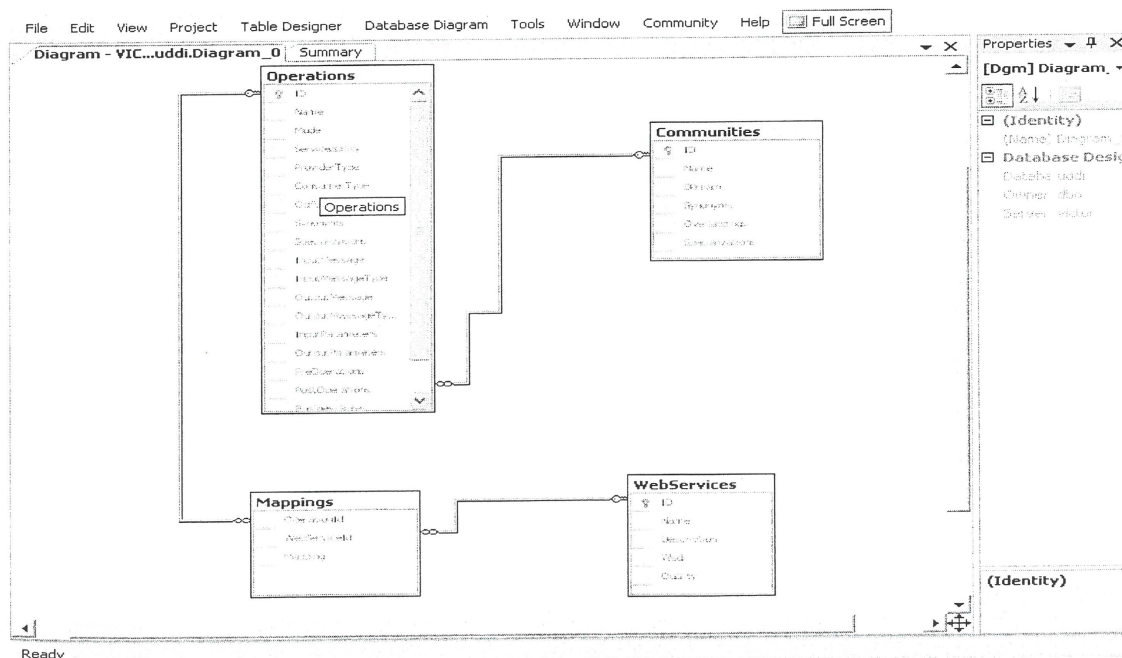


Figura 6. Utilizarea modelului relațional pentru descrierea comunităților, serviciilor și înregistrării acestora



## 2.4. Analiza cerințelor privind specificarea cererilor utilizator

Cetățenii pot accesa sistemul WebAgeing prin specificarea unui workflow (figura următoare), ale cărui activități sunt operații generice selectate din comunitățile de interes în mod grafic. În afară de nodurile de tip activitate, workflow-ul mai conține noduri de decizie, care asigură controlul dorit în execuția activităților acestuia. Fereastra de proiectare a workflow-lui conține un control de navigare prin colecția de comunități, iar în cadrul unei comunități sunt furnizate mijloace de navigare pe operațiile generice ale comunității și se permite tragerea cu ajutorul mouse-ului a operației generice selectate. În momentul tragerii unei operații generice pe workflow-ul curent, utilizatorul specifică valorile parametrilor de intrare, cât și atributele de calitate așteptate de la operația concretă a serviciului Web, ce implementează operația generică.

Nodurile de decizie sunt trase de pe toolbox-ul ferestrei de proiectare workflow, condițiile din interiorul nodurilor de decizie sunt condiții compuse utilizând operatorii logici cunoscuți, iar variabilele utilizate în cadrul acestor condiții sunt parametri de intrare/ieșire ai operațiilor generice din workflow, sau variabile care sunt definite în profilul utilizator. În afară de nodurile de tip activitate și decizie, workflow-ul conține un nod de start și unul sau mai multe noduri de stop. În momentul proiectării workflow-lui, nu se verifică existența ciclurilor deoarece nu se cunosc valorile variabilelor din nodurile de decizie. Ciclurile vor fi depistate în momentul execuției workflow-lui.

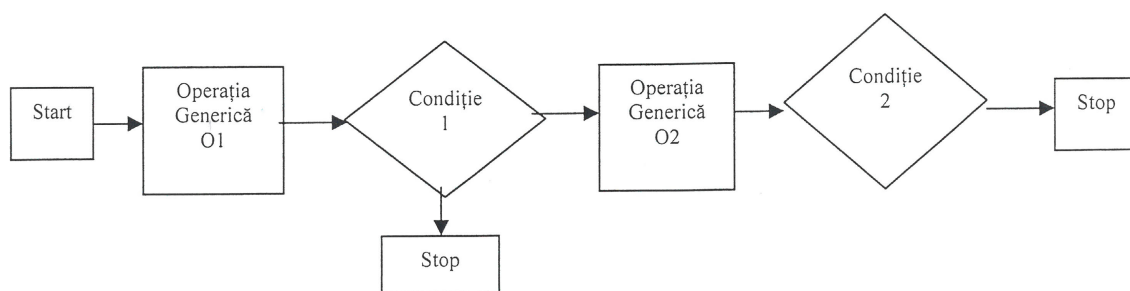


Figura 7. Specificarea cererilor utilizator sub forma unui workflow

Evident, cea mai simplă cerere pe care un utilizator poate să o specifice este reprezentată printr-un workflow care conține o singură operație generică ca în figura de mai jos:

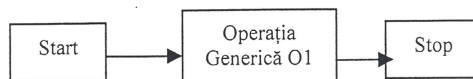


Figura 8. Specificarea unei cereri elementare

Execuția acestei cereri va fi făcută după ce, în prealabil, au fost executate toate preoperațiile sale. Evident, algoritmul de execuție a operațiilor are un caracter recursiv, deoarece, atunci când se execută preoperațiile unei operații generice date, aceste preoperații pot avea ele înseși preoperații și așa mai departe. Algoritmii ce vor fi descriși în continuare ilustrează, în structura lor, natura recursivă a procesului de execuție a operațiilor generice.

Fereastra de proiectare a workflow-lui are și posibilitatea de ștergere și editare a nodurilor conținute în workflow. De asemenea, fereastra conține instrumente pentru construirea în mod vizual a condițiilor compuse utilizând operatorii logici: and, or, not. În workflow-l atașat unei cereri utilizator este posibilă introducerea și altor tipuri de noduri ca: Join de tip AND, Join de tip OR, paralelism.

Cererile admise în sistemul WebAgeing, utilizează numai cele două tipuri de noduri: de tip operație generică (poate avea un singur succesor) și de tip decizie (au două noduri succesoare). Nodurile de tip Stop sunt singurele noduri care nu au succesor.

## 2.5. Analiza cerințelor privind execuția serviciilor Web compuse

Satisfacerea cererilor persoanelor implică execuția workflow-lor atașate acestor cereri utilizator. Execuția fiecărui workflow se face în mod personalizat, în sensul că toate condițiile care apar atât în nodurile de decizie ale workflow-lui, cât și cele atașate preoperațiilor conțin variabile ale căror valori se găsesc fie în profilul utilizator, fie în parametri de intrare/ieșire ai operațiilor componente. Astfel, aceeași cerere adresată sistemului de doi utilizatori diferiți, va fi satisfăcută în mod diferit, în funcție de profilul fiecăruia.

Algoritmul de execuție personalizată a workflow-lor atașate cererilor utilizator include următorii pași:



1. setează ca nod curent de execuție nodul "Start" al workflow-lui;
2. dacă nodul curent este de tip "Stop" se oprește algoritmul;
3. execută nodul curent;
4. stabilește nodul succesor și mergi la 1.

Execuția unui nod de tip operație generică se face parcurgând următorii pași:

1. se caută preoperațiile acestei operații;
2. se execută fiecare preoperație, dacă una din preoperații cade, atunci cade și execuția operației părinte;
3. se testează condiția atașată preoperației, dacă aceasta este falsă atunci execuția operației părinte cade.

Execuția unui nod de decizie se face testând condiția atașată și, dacă este adevărată, atunci se execută ramura corespunzătoare, iar dacă este falsă, se execută cealaltă ramură.

Execuția efectivă a unei operații generice după ce toate preoperațiile sale au fost efectuate și toate condițiile sunt adevărate, presupune următorii pași:

1. selectarea operațiilor concrete înregistrate cu operația generică;
2. reținerea, dintre aceste operații, a acelor care satisfac criteriile calitative.

Figura următoare ilustrează o descriere a algoritmului precedent, utilizând un pseudocod:

```

ExecuteRequest(OperationsList)
{
    Map map=new Map();
    foreach operation O in OperationsList
        ExecuteOperation(O.Id_Operation)
}

ExecuteOperation(string Id_Operation)
{
    if not(verifyMap(Id_Operation))
    {
        if (Preoperations(Id_Operation)==null)
        {
            WebService ws=selectService(Id_Operation);
            Call(ws);
            Map[Id_Operation]=true;
            Return ;
        }
        foreach Preoperations(Id_Preoperation) P
        {
            executeOperation (P.Id_Preoperation)
            if not(P. Condition) throw new Exception("failure")
        }
    }
}

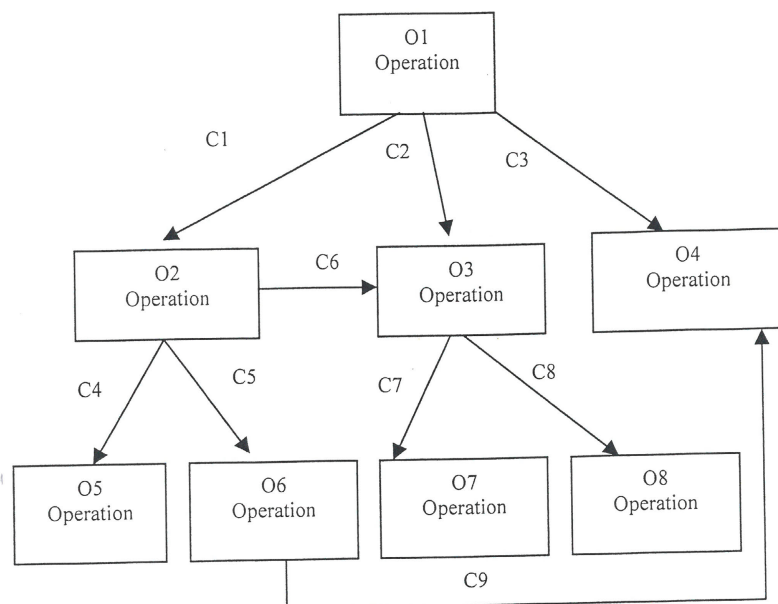
bool verifyMap(sting Id-Preoperation)
{
    if (map.exist(Id-Operation))
    {
        if ( map[Id_Operation]) return true
        else
            throw new Exception(" s-au depistat cicluri in graph");
    }
    else
        return false
}

```

**Figura 9. Algoritmul de execuție a cererilor utilizator**

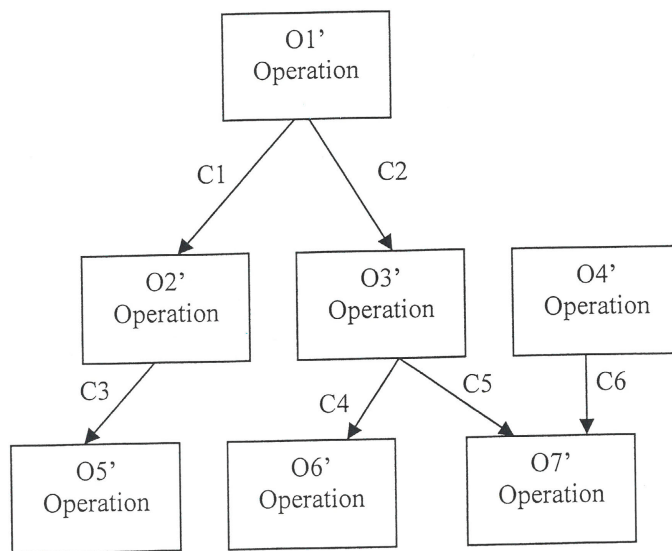


Pentru a exemplifica modul de lucru al algoritmului de execuție al cererilor utilizatorului, se consideră că baza de cunoștințe conține două comunități de interes pentru persoanele vârstnice. Prima comunitate include 8 operații generice cu relațiile „Preoperations” ca în figura de mai jos. Astfel, operația O1 se poate executa numai dacă operațiile O2, O3, O4 au fost executate, iar condițiile C1, C2, C3 (care sunt predicate ce au ca argumente atât atribute din profilul utilizator, cât și intrări/ieșiri ale operațiilor O2, respectiv O3, respectiv O4) sunt adevărate.



**Figura 10. Comunitatea 1 de interes pentru persoanele vârstnice**

Cea de a doua comunitate de interes include șapte operații generice cu relațiile „Preoperation” ca în figura următoare. Pe arcele grafului sunt trecute condițiile de execuție ale operațiilor generice.

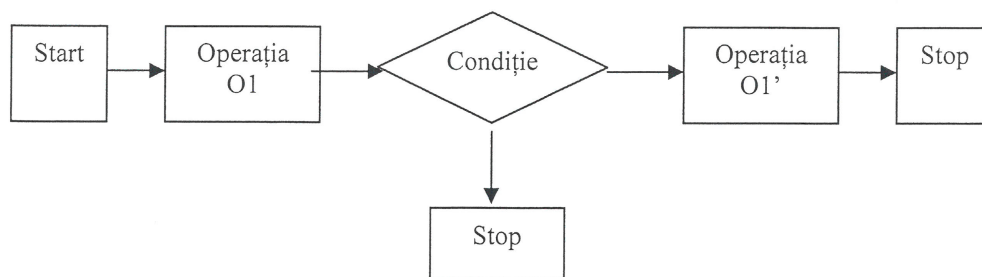


**Figura 11. Comunitatea 2 de interes pentru persoanele vârstnice**

Utilizatorul, care este o persoană vârstnică sau un funcționar public ce lucrează cu persoane vârstnice, își construiește cererea ca o listă de operații generice aparținând celor două comunități. Lista poate include, pe lângă operațiile generice, și blocuri de decizie.

Să presupunem că cererea utilizatorului a fost construită în mod grafic ca mai jos:





**Figura 12. Cerere utilizator construită în mod grafic, sub formă unui workflow**

Aceasta presupune execuția a două operații generice din comunități diferite. Astfel, înainte de execuția cererii sistemul trebuie să consulte baza de cunoștințe pentru a valida dacă cele două comunități pot fi combinate între ele pentru a produce servicii utile persoanelor vârstnice. Cererea validată este transmisă funcției `ExecuteRequest`, funcție care apelează, pentru fiecare operație generică din listă, funcția `ExecuteOperation`.

O alternativă a algoritmului de execuție a unei operații se bazează pe utilizarea unui registru de stivă SP pentru parcurgerea în preordine a grafului asociat operației, graf care este aciclic și care memorează în nodurile sale diverse informații utile algoritmului de execuție, pe care algoritmul precedent le memorează într-o structură de tip Map.

```

ExecuteOperation(string Id_Operation)
{
    Node node=verifymarcaj(Id_Operation, null, true);
    Node SP=node;
    1:
    if (Sp=null) Stop
    else
    {
        Operation preperation =nextPreoperation(SP.Id_Operation);
        if (preoperation !=null )
        {
            (SP.nrcopil)++;
            SP= verifymarcaj( SP.Id_Operation, SP, preoperation.Cndition),
            Goto1
        }
        else {execute(SP) ,
            mark(SP.Id_operation),
            if eval(SP.condition)==false , throw new Exception("conditie falsa")
            else
            {
                SP=SP.father
                Go to 1
            }
        }
    }
}

Node verifymarcaj(string Id_Operation , Node father, Conditon condition)
{
    [ Node m , bool mark]=map(Id_Operation);
    if(m==null) reurn new Node(Id_Operation, condition,0,father)
    else if(mark) {if(eval(condition) then return father else throw new Exception("conditie falsa")
        throw new Excepion("depistat ciclului)
  
```

**Figura 13. Varianta 2 a algoritmului de execuție workflow**



## Bibliografie

1. **BENATALLAH, B., M. DUMAS, NGU AHH SHENM:** Declarative Composition and peer-to-peer Provisioning of Dynamic Web Services. În: Proc. of the ICDE conference, San Jose, CA, February 2002, pp. 297-308.
2. **CHAKRABORTY, D, F. PERICH, A. JOSHI, T. FININ, Y. YESHA:** A Reactive Service Composition Architecture for Pervasive Computing Environments. În: Proc. of the 7th Personal Wireless Communications Conference, Singapore, October 2002, pp. 53-62.
3. **LI, L., I. HORROCKS:** A Software Framework for Matchmaking Based on Semantic Web Technology. În: Proc. of the WWW 2003 Conf., Budapest, May 2003, pp. 331-339.
4. **MEDJAHED, B., A. REZGUI, A. BOUGUETTAYA, M. OUZZANI:** Infrastructure for e-government Web Services. IEEE Internet Comput 7(1), 2003, pp. 58-65.
5. **PAOLUCCI, M., T. KAWAMURA, TR. PAYNE, K. SYCARA:** Semantic Matching of Web Services Capabilities. În: Proc. of the 1st international Semantic Web Conf., Sardinia, Italy, June 2002, pp. 318-332.
6. **SHETH, A., J. MILLER:** Web Services: Technical Evolution Yet Practical Revolution. IEEE Intell Sys 18(1), 2003, pp. 78-80.