SPECIFICAȚIA SISTEMULUI DE NOTIFICARE PENTRU UTILIZATORII MOBILI

Victor Popa vpopa@ici.ro
Ileana Stănescu ileanas@ici.ro
Liliana Constantinescu lconst@ici.ro
Victoriana Popa victoriana.popa@unicredit.it
Institutul Național de Cercetare - Dezvoltare în Informatică, ICI, București
Unicredit Italia

Rezumat: Lucrarea prezintă specificația sistemului de notificare pentru utilizatorii mobili. Sistemul prezentat abordează aspecte relevante ale utilizatorilor mobili privind: specificarea serviciilor și proceselor necesare rezolvării taskurilor acestora utilizând limbajul BPEL, descoperirea sintactică și semantică a serviciilor specificate utilizând standardul UDDI extins, execuția proceselor BPEL, monitorizarea execuției proceselor, notificarea evenimentelor declanșate de serviciile componente specificate în procese, prelucrarea evenimentelor utilizând informațiile contextuale ale utilizatorilor mobili, ca de exemplu poziția geografică a acestora.

Cuvinte cheie: informații contextuale, servicii Web, procese BPEL, descoperire servicii, notificare evenimente, UDDI.

Utilizatorii mobili interacționează cu diverse servicii web în scopul rezolvării unor taskuri relevante (rezervări bilete de zbor, rezervări locuri cazare, informare condiții atmosferice, informare curs valutar, informare indice bursă etc.). Taskurile (activitățile) utilizatorilor mobili sunt specificate împreună cu structurile de control și secvențele de tratare a evenimentelor utilizând descrieri de procese în limbajul BPEL4WS [4], extins cu facilități de tip workflow. Handlerele de tratare a evenimentelor se bazează pe contextul curent al utilizatorului mobil. Astfel, dacă utilizatorului mobil i se notifică faptul că zborul său cu avionul a fost amânat cu o zi, handlerul de tratare a acestui eveniment va lua în considerare poziția geografică a utilizatorului; dacă utilizatorul a ajuns deja la destinație cu alt mijloc de transport handlerul de evenimente execută activitatea de anulare zbor, în caz contrar dacă utilizatorul nu a ajuns la destinație handlerul de evenimente dialoghează cu utilizatorul mobil, utilizând unul din protocoalele de comunicare specificate, în scopul luării deciziei potrivite.

Dacă utilizatorul mobil acceptă noua dată de zbor, handlerul de evenimente trebuie să anuleze efectul tuturor taskurilor care depind de taskul rezervare zbor (rezervare hotel, rezervare bilet auto etc.). Pentru anularea acestor efecte handlerul de evenimente va executa acțiuni compensatorii de anulare rezervare hotel, anulare rezervare bilet auto, apoi va executa din nou taskurile de rezervare hotel și rezervare bilet auto.

Sistemul de notificare pentru utilizatorii mobili include următoarele componente principale:

1. Componenta de descoperire a serviciilor Web necesare în rezolvarea taskurilor utilizatorilor mobili.
2. Componenta de specificare a proceselor (taskuri, instrucțiuni de control, model tratare evenimente, tratare excepții etc.).
3. Componenta de monitorizare a execuției taskurilor.

1. Componenta de descoperire a serviciilor Web

Componenta de descoperire a serviciilor Web are la bază descoperirea sintactică și semantică a serviciilor specificate utilizând standardul UDDI extins [10]. Componenta include două subcomponente:

- Subcomponenta de căutare în registrul master a operatorilor de registre UDDI care operează în domeniului de interes al utilizatorului mobil. Selectarea unui operator de registru se realizează în funcție de modelul domeniului de interes.
- Subcomponenta de descoperire în registrul UDDI, selectat la primul pas, a unui serviciu Web care satisface criteriile sintactice și semantice specificate de utilizator.

1.1. Componenta de căutare în registrul master

Pentru ca serviciile Web ale furnizorilor să poată fi accesibile, fiecare furnizor își publică în registrul său privat serviciile Web proprii. Pentru ca registrele private ale furnizorilor să poată fi găsite, ele sunt înregistrate în registrul master care clasifică registrele în funcție de domeniile de interes utilizând ontologia de registre. Figura de mai jos ilustrează un fragment din ontologia registrelor memorată în registrul master.

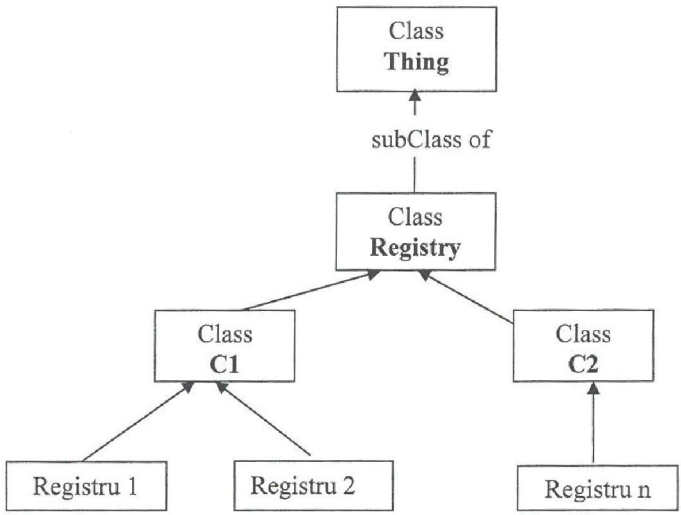

Figura 1. Registrul Master

În figură se observă faptul că dacă un utilizator este interesat de domeniu definit prin clasa C1, atunci el va găsi doi furnizori de registre UDDI care operează în acel domeniu. Selectarea, în final, a registrului 1 sau 2 depinde de ontologiile domeniului furnizate de cei doi furnizori.

1.2. Componenta de descoperire a serviciilor Web în UDDI

1.2.1. Structura registrelor UDDI

În scopul descoperirii serviciilor relevante publicate în UDDI se prezintă mai întâi structura UDDI utilizată în publicarea serviciilor Web. Principalele structuri ale registrului UDDI sunt:

- BusinessEntity
- BusinessService
- BindingTemplate
- Tmodel
- TmodelInstanceInfo
- InstanceDetail

BusinessEntity este structura utilizată în scopul memorării informațiilor care descrie furnizorii care își publică serviciile în UDDI.

BusinessService este structura utilizată în scopul memorării informațiilor care descrie serviciile de business publicate în UDDI.

BindingTemplate este structura care memorează informații ce descriu felul cum serviciile vor fi invocate. Un BusinessService poate fi invocat în mai multe feluri.

Tmodel este structura care memorează informații referitoare la numele și cheia sa, referința către o specificație dată (overviewdoc), criterii de clasificare (categoryBag), informații de identificare (identifierBag) etc.

Structura categoryBag utilizează informația keyedReference în scopul referirii criteriilor de clasificare. Această informație este descrisă mai amănunțit în continuare.

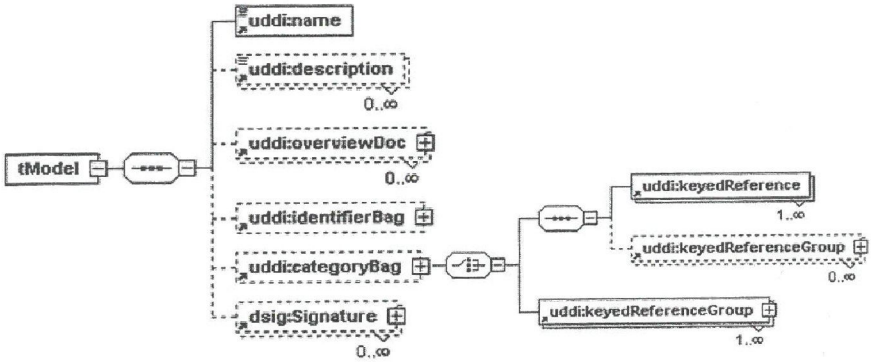

Figura 2. Structura unui tmodel

CategoryBag este cel mai important element care permite referirea structurilor ce trebuie clasificate conform sistemelor de clasificare cunoscute (NAICS, ISO etc.). CategoryBag conține în mod direct unul sau mai multe elemente keyedReference care fac referință la tmodelele utilizate în clasificare. Fiecare tModel utilizat în clasificare este referit prin cheia sa unică.

Structura registrelor UDDI utilizate în publicarea serviciilor Web este ilustrată mai clar în figura următoare. Se observă că elementele businessService, businessEntity cât și tModel pot conține o colecție de elemente de tipul categoryBag, utilizate în principal pentru clasificări. Fiecare element de tipul businessEntity include o colecție de elemente de tipul businessService, iar fiecare element de tipul businessService include o colecție de elemente de tipul bindingTemplate.

Numărul de elemente de tipul bindingTemplate coincide cu numărul de endpoints (puncte finale) pe care un element de tipul businessService îl furnizează. Fiecare endpoint al unui serviciu e specifică protocolul de accesare a serviciului, specificația calitativă a serviciului accesat prin acest endpoint, modelul de business al serviciului accesat etc.

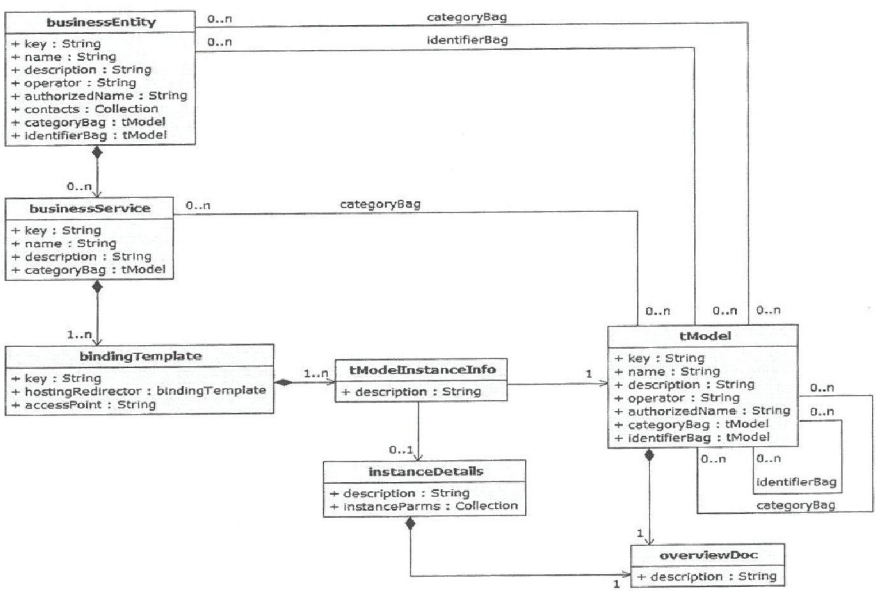

Figura 3. Structura registrelor UDDI

Elementele de bază utilizate în structura UDDI sunt realizate în limbajul XML, pentru următoarele elementele:

- BusinessService
- BindingTemplate
- tModel

1.2.2. Publicarea interfețelor WSDL în UDDI

Furnizorii de interfețe utilizează tagul <overviewDoc> al unui tModel pentru a referii interfața de publicat. Clasificarea interfețelor în funcție de anumite standarde este realizată cu ajutorul unui set de taguri de tipul <KeyedReference>. Interfața mobil.wsdl este publicată în UDDI utilizând un tModel cu cheia "uuid:5DD52389-B1A4-4fe7-B131-0F8EF73DD175". Interfața este clasificată ca o specificație de tip wsdl ("wsdlSpec"). De asemenea, interfața este clasificată în raport cu standardul NAICS ca aparținând grupei 514191, grupă pentru servicii online.

După ce o interfață a fost publicată utilizând un tModel, obiectul businessService, care implementează interfața respectivă, trebuie să creeze o referință către acest tModel utilizând cheia acestuia.

1.2.3. Publicarea modelului calitativ al unui serviciu Web(QoS)

Primul pas este construirea unui tModel care să refere modelul QoS, în al doilea pas se construiește un obiect de tipul bindingTemplate, care referă tModelul construit la primul pas. Modelul calitativ poate fi specificat în mod simplu sau poate fi specificat utilizând un model matematic de calcul plecând de la indicatorii calitativi ai fiecărui serviciu partener, atunci când procesul este un proces colaborativ.

Având modelul QoS construit, fișierul model.qos este publicat în tModelul cu cheia "uuid:5DD52389-B1A4-4fe7-B131-0F8EF73DD200", tModelul respectiv este apoi clasificat ca un tModel conținând o specificație de tip QoS (keyValue="qos").

După ce modelul calitativ a fost publicat într-un tModel, este necesar ca acele implementări de servicii Web (bindingTemplate) care satisfac cerințele specificate în modelul calitativ să facă referiri la tModelul respectiv.

1.2.4. Publicarea semantică a unui serviciu Web

Pentru fiecare operație se specifică în obiectul businessService, tModelul care referă ontologia ce conține concepte înrudite cu conceptul atașat operației respective, se specifică un tModel ce referă ontologia parametrilor de intrare, se specifică un tModel ce referă ontologia parametrilor de ieșire, de asemenea se specifică un tModel ce referă ontologia Precondiții, în final se specifică un tModel ce referă ontologia Efecte.

1.2.5. Publicarea modelului de business

Modelul de business este un model abstract incluzând activitățile colaborative între parteneri (invoke, receive, replay etc.).

Modelul de business este referit în tModel utilizând linia: <overviewURL> http://ici.ro/schema_files/abstractmodel.bpel </overviewDoc>.

Odată ce modelul de business a fost publicat într-un tModel, acest tModel trebuie să fie referit în unul din obiectele de tipul <bindingTemplate> conținute în obiectul <businessService>.

1.2.6. Descoperirea serviciilor prin interfețe și semantică

În scopul descoperirii serviciilor Web care implementează o interfață dată, utilizatorul trebuie să specifice, în faza de proiectare a procesului pentru fiecare serviciu ce trebuie descoperit pentru a fi apelat, următoarele informații:

- Domeniul la care aparține serviciul sau registrul UDDI.
- Detalii de descoperire, dacă sunt cunoscute (furnizor, categorie etc.).
- Restricțiile privind calitatea serviciilor ce trebuie descoperite.

- Specificarea modelului de business.

Editorul de proiectare a proceselor utilizează UDDI API pentru a căuta toate obiectele businessService de tip wsdl care satisfac detaliile de descoperire. Utilizatorul selectează acel businessService care implementează interfața dorită. De asemenea, utilizatorul selectează operația dorită din interfața fixată și asignează variabile din proces parametrilor de intrare - ieșire pentru operația selectată.

În timpul execuției procesului, pentru descoperirea finală a serviciului care implementează interfața, se parcurg toate obiectele bindingTemplate aparținând obiectului businessService fixat în faza de proiectare și se reține numai acele obiecte bindingTemplate care satisfac specificația modelului de business.

Pentru fiecare obiect bindingTemplate reținut se calculează măsura de potrivire a acestuia cu specificația privind calitatea serviciului descoperit. În final se reține cel cu măsura cea mai mare și apelează utilizând URL acestuia.

Pentru descoperirea semantică a unui serviciu, utilizatorul trebuie să furnizeze, în faza de proiectare a procesului, următoarele informații:

- Domeniul la care aparține serviciul sau registrul UDDI.
- Detalii de descoperire, dacă sunt cunoscute (furnizor, categorie etc.).
- Restricțiile privind calitatea serviciilor ce trebuie descoperite.
- Specificarea modelului de business.
- Conceptul atașat operației de executat.
- Conceptele atașate parametrilor de intrare.
- Conceptele atașate parametrilor de ieșire.
- Precondițiile ce trebuie satisfăcute.
- Efectele așteptate după execuția serviciului.

Editorul de proiectare utilizează UDDI API pentru a fixa colecția de obiecte businessService care satisface detaliile de descoperire. Pentru fiecare element din colecția stabilită anterior, editorul calculează măsura de potrivire (Ms) a elementului cu specificațiile semantice (operație, intrări, ieșiri, precondiții, efecte).

În timpul execuției procesului, descoperirea semantică continuă cu următorii pași:

- Pentru fiecare businessService (bS) din colecția anterioară, pentru fiecare obiect bindingTemplate (bT) aparținând obiectului (bS) se calculează măsura de potrivire (Mq) utilizând specificația calitativă și cea a modelului de business, măsura de potrivire (Mc) a specificației semantice cu modelul semantic.
- În final, utilizând cele trei măsuri (Ms), (Mq) și (Mc) se calculează măsura finală de potrivire pentru fiecare obiect bindingTemplate. Se reține acel obiect bindingTemplate care are măsura de potrivire (M maximă) și se apelează serviciul reținut utilizând URL-ul acestuia.

2. Componenta de specificare a taskurilor

Componenta permite utilizatorului mobil să-și definească activitățile și controlul execuției acestora. Limbajul utilizat în specificarea activităților și a controlului execuției lor este limbajul BPEL4WS, limbaj ce permite execuția proceselor de business în mod colaborativ de către mai mulți parteneri, fiecare partener având un rol bine definit în cadrul procesului.

Modelul BPEL4WS permite atât specificarea abstractă a serviciilor partenerilor și a relațiilor dintre aceștia cât și implementarea concretă a proceselor (variabile de stare, control etc.).

Trăsăturile esențiale ale limbajului, evidențiate în continuare (colaborarea mai multor servicii partenere, posibilitatea apariției în orice punct al procesului a evenimentelor declanșate de serviciile partenere, posibilitatea răzgândirii utilizatorului mobil în orice moment; prin efectuarea unor activități compensatorii, utilizatorul mobil poate anula efectul unor activități deja executate) fac limbajul BPEL potrivit pentru implementarea sistemului de notificare.

Specificarea unei secvențe de activități în BPEL se face utilizând tagurile <sequence> și </sequence>, așa cum este ilustrat în figura de mai jos.

Figura 4. Diagrama de proces pentru două activități secvențiale

Maparea diagramei de mai sus, în limbajul BPEL, se face folosind activitatea compusă <sequence> așa cum este ilustrat în figura de mai jos.

Specificarea secvențelor de activități, ce se vor executata în paralel, se face utilizând tagurile <flow> și </flow>, așa cum este ilustrat în figura de mai jos. De remarcat faptul că deși secvențele sunt executate în paralel ele pot fi sincronizate în diverse puncte definind relații de precedență între activitățiile secvențelor.

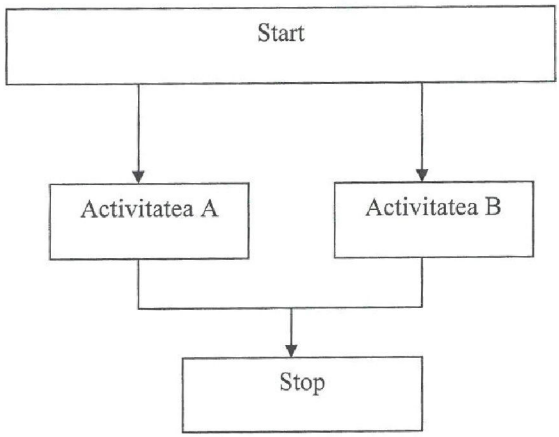

Figura 5. Diagrama de proces a două secvențe paralele

Maparea acestei diagrame de proces, în limbajul BPEL4WS, se face utilizând activitatea <flow>. Specificarea sincronizării execuției a două ramuri ale unui proces se face utilizând link-uri între activitățile care trebuiesc sincronizate.

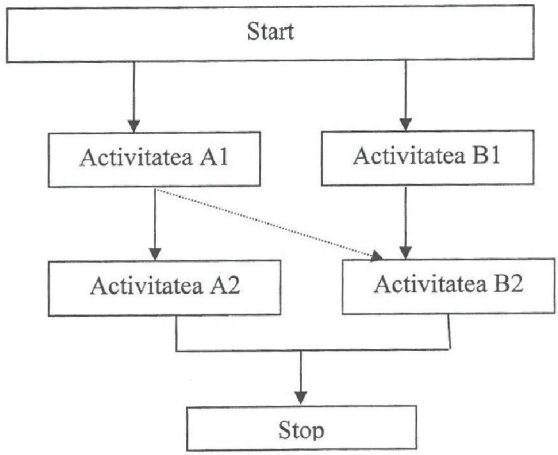

Figura 6. Diagrama proces de sincronizare a două ramuri paralele

Specificarea „listenerilor" din proces, pentru ascultarea producerii evenimentelor, se face folosind tag-urile <pick> și </pick>. În interiorul acestor tag-uri se specifică, folosind activitățile onMessage, handlerele utilizate pentru prelucrarea fiecărui eveniment așteptat.

După ce utilizatorul mobil a startat procesul conținând taskurile ce trebuiesc executate, el are

posibilitatea să se răzgândească în orice moment, dacă procesul nu a executat deja anumite acțiuni critice.

Comunicarea între procesele care conțin taskurile utilizator și procesele partenere se face în mod sincron sau asincron. În mod sincron, procesul ce conține taskurile utilizator utilizează activitatea „invoke" cu parametrii de intrare și ieșire, procesele partenere utilizează activitățile „receive" și „replay".

Comunicarea asincronă este utilizată frecvent în cazul când serviciul apelat durează un timp lung, iar procesul poate executa în acest timp alte activități ale sale, urmând ca atunci când serviciul apelat se termină, acesta să notifice procesului acest fapt. Pentru a apela în mod asincron un serviciu, procesul utilizează activitatea „invoke serviciu partener" dar fără a specifica parametrii de ieșire. Notificarea se face de către serviciul apelat prin execuția uneia dintre acțiunile „replay" sau „invoke".

Controlul execuției taskurilor este realizat prin acțiuni compuse switch, while, foreach etc. Tratarea excepțiilor este realizată prin utilizarea handlerelor de tratare a excepțiilor care sunt incluse în interiorul activităților compuse de tip scope. Dacă o activitate scop nu are handlere de tratare a excepțiilor, excepțiile sunt trimise nivelului imediat superior acestei activități (activitatea părinte).

Contextul utilizator este memorat în variabilele procesului, iar aceste variabile sunt ulterior utilizate în activitățile de tratare a evenimentelor.

Serviciile partenere ale procesului sunt specificate în mod abstract ca dublete (rol, portType) urmând ca endpoint-ul fiecărui serviciu partener să fie descoperit prin una din metodele descrise mai sus. Implementările actuale ale limbajului BPEL4WS nu au integrate metodele de descoperire a serviciilor descrise mai sus, acestea urmând a fi încorporate în implementările viitoare ale limbajul BPEL.

3. Componenta de monitorizare a execuției proceselor

Mașina BPEL execută taskurile din procese conform controlului specificat în cadrul acestora, utilizatorul mobil intervenind în una din următoarele situații:

- Execuția taskurilor manuale din proces.
- Luarea deciziilor când sunt notificate diverse evenimente.
- Monitorizarea execuției proceselor.

Mașina BPEL, când întâlnește taskuri ce trebuie executate de către utilizatorul mobil, le memorează pe agenda de lucrări a utilizatorului, trimițând SMS-uri utilizatorului mobil pentru atenționarea acestuia. Utilizatorul mobil se conectează la agenda sa de lucrări și completează taskurile de pe agendă.

Există situații când handlerele de tratarea erorilor și a evenimentelor trebuie să notifice utilizatorului mobil aceste erori sau evenimente pentru luarea deciziilor potrivite. Notificarea se face utilizând protocoalele de comunicare definite pentru fiecare utilizator mobil. Monitorizarea execuției proceselor este la fel de importantă ca și execuția acestora, aceasta presupunând:

- facilități pentru utilizatorul mobil de a interveni în execuția proceselor executând comenzi ca anularea efectelor activităților deja executate, aparținând unui scop dat, utilizând functii oferite de BPEL (activități compensatorii). De remarcat faptul că nu se poate utiliza mecanismul Rollback din tranzacțiile clasice deoarece o activitate BPEL poate dura o perioadă lungă de timp și resursele nu pot fi blocate pe tot parcursul execuției acelei activități.
- Terminarea forțată a execuției unei instanțe de proces, chiar dacă nu s-au executat toate activitățile sale.
- Aruncarea excepțiilor potrivite atunci când execuția procesului nu concordă cu scopul dorit.

4. Concluzii

Sistemul de notificare pentru utilizatorii mobili prezentat în lucrare, permite utilizatorilor mobili:

- specificarea serviciilor necesare rezolvării taskurilor lor utilizând procese BPEL4WS,
- descoperirea sintactică și semantică a serviciilor specificate utilizând standardul UDDI,
- execuția proceselor,
- monitorizarea execuției proceselor,

- notificarea evenimentelor declanşate de serviciile componente specificate în procese,
- prelucrarea evenimentelor utilizând informaţiile contextuale ale utilizatorilor mobili (poziţie geografică etc.).

Bibliografie

1. **NAGANUMA, T.**: A task oriented approach to service retrieval in mobile computing environment. În: *Proc. of IASTED* International Conference on Artificial Intelligence and Applications *(AIA 2005)*, 2005.
2. **PALLICKARA, S., G. FOX**: On the matching of events in distributed brokering systems. În: Proc. of IEEE ITCC Conference on Information Technology, Vol. II, April 2004, pp. 68–76.
3. * * *: Get an overview of the processes defined in the UML that generates corresponding BPEL and WSDL files in the article "FROM UML to BPEL" (developerWorks, September 2003).
4. * * *: Read the Reference guide for creating BPEL4WS documents (developerWorks, November 2002).
5. * * *: Learn the concepts involved in implementing executable business processes in the column series *Business Process with BPEL4WS*.
6. * * *: Get an explanation of business processes and BPEL4WS in "Business processes in a Web services world" (developerWorks, August 2002).
7. * * *: Explore a basic introduction to the concepts of workflow in "Business processes and workflow in a Web services world" (developerWorks, January 2003).
8. * * *: Learn more about the Web Services Security model in "Security in a Web Services World: A proposed architecture and roadmap" (developerWorks, April 2002).
9. * * *: Understand the impact and importance of standards and specifications for the development of SOA and Web services with this standards roadmap.
10. * * *: OASIS Consortium Members Approve UDDI Version 3 as an OASIS Standard CoverPages, 2 Feb 2005.
11. * * *: UDDI v2 Ratified as OASIS Open Standard, OASIS News, 20 May 2003.