

Tehnologii pentru aplicatii distribuite pe Internet

Asist. Carmen STANCIU

Catedra de Informatica Economica, A.S.E. Bucuresti

Crearea aplicatiilor distribuite reprezinta o preocupare veche si importanta a proiectantilor de aplicatii, iar paradigmele client/server si obiectele distribuite sunt cunoscute de relativ multa vreme. Azi, pe Internet se utilizeaza o serie de tehnologii pentru aplicatii distribuite care vor fi prezentate succint în continuare.

Cuvinte cheie: CORBA, DCOM, ASP, JavaRMI, JavaIDL, Web de obiecte.

De-a lungul celor câtiva ani de existenta, Web-ul a avut o evolutie spectaculoasa, trecând de la simplu la complex, stadiul spre care se îndreapta astazi.

Initial sa pornit de la **Web-ul hypertext**, care avea la baza *serverul de fisiere bazat pe URL*, urmat apoi de **Web-ul cu raspunsuri simple**, format *din formulare, CGI, tabele, ISAPI, NSAPI*, si pe urma *script-uri, push, WebObjects, DHTML, cookies, plug-in CORBA, ASP*, iar în prezent, în formare, cel cu interactivitate avansata, **Web-ul de obiecte** care contine aplicatii client/server, create cu ajutorul *Java Beans, applet-uri, ActiveX, coordonatori de componente, inter-actiuni ORB via CORBA sau DCOM*.

Aceasta evolutie spectaculoasa a WWW-ului a condus, fireste, la dezvoltarea tehnologiilor de aplicatii distribuite, care au devenit din ce în ce mai complexe.

Directiile de dezvoltare a tehnologiilor de aplicatii distribuite au fost orientate spre Java, care ofera mecanisme puternice în acest sens, precum si spre CORBA, care permite în plus independenta de limbaj a implementarii clientului si serverului care comunica între ele.

În continuare se vor prezenta succint cele mai importante caracteristici ale principalelor tehnologii pentru aplicatii distribuite.

1. CORBA

Premiza de baza a programarii orientate obiect consta în faptul ca aplicatiile si functiile pot fi divizate în clase de obiecte care pot fi reutilizate. Un program orientat

OOP este organizat în jurul *datelor* ce trebuie procesate si nu în jurul *actiunilor* ce trebuiesc luate, rezultând astfel sisteme mult mai stabile.

OOP a impus noi concepte software si noi structuri lexicografice, ce includ obiecte, attribute, clase, abstractii, mosteniri, încapsulari si polimorfism.

Implementarea cu succes în lumea comerciala a tehnologiei OOP este strâns legata de standardele emise de grupurile specifice, cum ar fi *OSF (Open Software Foundation)* si *OMG (Object Management Group)*.

OMG (Object Management Group) a dezvoltat un standard care s-a concretizat în arhitectura **OMA (Object Management Architecture)**.

Aceasta arhitectura furnizeaza o infrastructura ce permite obiectelor software definite sa comunice si sa interopereze între ele, indiferent de specificul platformei sau limbajului.

Elementul central al arhitecturii îl constituie **ORB (Object Request Broker)**. Acest element actioneaza ca un broker (agent de schimb) si este responsabil cu obiectele clientilor, care pun întrebări, si cu implementarea obiectelor, care furnizeaza raspunsuri.

OMG a elaborat o specificatie pentru ORB, numita **CORBA (Common Object Request Broker Architecture)**.

CORBA, gândit initial pentru clienti puternici, cum ar fi întreprinderi mari care detin sisteme si aplicatii realizate în arhi-

tecturile traditionale pe care doresc sa le modernizeze fara a renunta complet la ele, standardizeaza arhitectura generala a unui model obiect, lasând deschise toate aspectele particulare si cerând compatibilitate pentru codul sursa al aplicatiilor si pentru formatul mesajelor de retea.

Caracteristicile CORBA sunt:

- ofera o interfata de programare de aplicatii simpla, disponibila tuturor obiectelor;
- obiectul are o singura interfata care este specificata folosind un limbaj independent de implementarea obiectelor, adica limbajul de descriere a interfetelor IDL;
- nu are pointeri, folosind în locul lor referinte;
- nu are mostenire de implementare;
- nu specifica nimic despre gestiunea automata a memorie;
- defineste interfete pentru depozitele de interfete si depozitele de implementari;
- serviciile sunt definite sub forma de bibliotecii cu functionalitate diversa, cum ar fi persistente obiectelor, relatiile între obiecte, structurile de date generice; serviciile sunt de doua tipuri:
 - proprii mediilor desktop – pentru documente compuse, accesul la baze de date si servicii de retea;
 - la nivel de întreprindere – pentru gestiunea tranzactiilor, securitate, mesagerie asincrona, directoare.
- are ca parte componenta protocolul **IOP** (Internet Inter ORB) , coloana vertebrala a comunicatiei în întreprindere, care asigura interoperabilitatea între ORB-uri.

Modul de lucru al ORB-ului

ORG este un mecanism *transparent* de a pune întrebări si de a primi raspunsuri de la obiecte locale sau la distanta, fara ca, clientii sa cunoasca mecanismele utilizate pentru reprezentarea, activarea, stocarea, comunicarea cu obiectele.

Aceasta permite ORB-ului sa fundamenteze construirea aplicatiilor formate din obiecte distribuite si realizarea interoperabilitatii între aplicatii din toate mediile.

Pasii de lucru al ORB-ului sunt explicati în continuare si ilustrati în figura 1:

- 1- clientul emite o cerere ORB-ului pentru un serviciu de obiect, printr-o interfata API (Application Programming Interface);
- 2- ORB-ul utilizeaza un index (directory) pentru localizarea obiectului celui mai potrivit si se consulta (cu un “repository”) pentru a determina cum sa comunice cu acel obiect;
- 3- cererea este trecuta interfetei adaptorului de obiect, situata pe server;
- 4- obiectul efectueaza serviciul sau cere ajutor altor obiecte prin ORB;
- 5- raspunsul este trimis înapoi clientului, prin ORB.

Tipurile de servicii de obiecte, sunt: persistenta (se refera la standardizarea obiectelor stocare), *ciclu de viata* (crearea, stergerea, copierea, mutarea obiectelor), *notificarea evenimentelor*, *numirea*, *externalizarea*, *controlul concurent*, *relatiile*, *tranzactile*, *cererile*.

Pentru viitor, se pune problema cine va câstiga lupta dintre COM si CORBA si raspunsul poate veni din zona Java. CORBA a fost în topul preferintelor câtiva ani buni, Microsoft vine acum puternic din urma (prin extinderea lui COM cu COM+), iar între ei se infiltreaza un competitor serios, Java Virtual Machine. Se pare ca, CORBA va ramâne o infrastructura importanta de comunicatie între aplicatii la nivel de întreprindere prin IOP, protocol de comunicatie pentru obiecte Java distribuite care utilizeaza RMI.

2. JavaRMI, JavaIDL

Java RMI (Remote Method Invocation) este un mecanism puternic, care permite comunicarea distribuita si scalabila între programele Java.

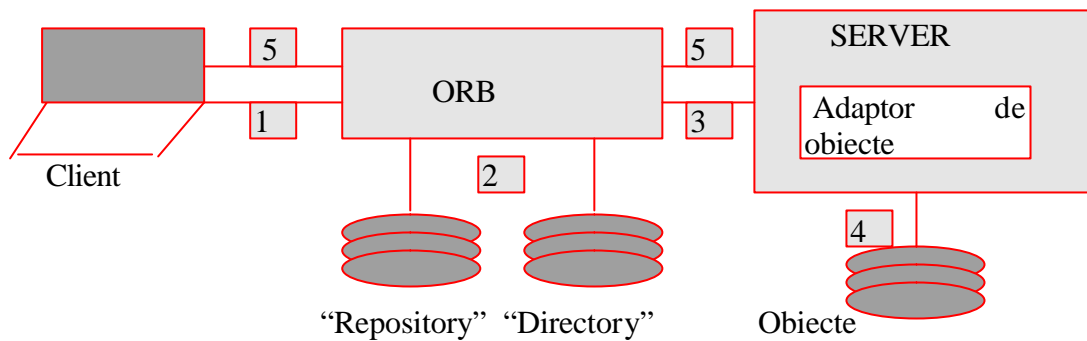


Figura 1

A fost introdus în JDK1.1, este destinat exclusiv Java și este creat de JavaSoft în scopul facilitării apelului metodelor de la distanță de către programele Java, precum și a transmiterii obiectelor între sisteme.

Caracteristicile RMI sunt următoarele:

- suportă colector de deseuri distribuit;
- păstrează elemente de securitate din Java;
- permite cereri de prelucrare multiple datorită naturii multifilare a limbajului Java;
- folosește serializarea obiectelor, ca și în Java, pentru transmiterea obiectelor între sisteme fără a folosi scheme de codare complexe ca și în CORBA.

RMI extinde puterea limbajului Java în domeniul sistemelor distribuite, realizând și o interoperabilitate cu CORBA prin implementarea unui subset din IIOP. În viitor, RMI poate oferi replicare la invocarea metodei și poate chiar alte protocoale de transport diferite de TCP/IP, putând deveni viitorul comunicațiilor orientate pe obiecte.

Java IDL este un ORB care adaugă la CORBA capabilități Java, oferind interoperabilitate și conectivitate bazate pe standarde.

IDL (Interface Definition Language) permite ca toate obiectele CORBA să fie descrise în aceeași manieră, realizând astfel independența față de limbajul nativ, care poate fi C, C++, Java, COBOL etc.

Java IDL face posibilă dezvoltarea de aplicații distribuite Java care invocă în mod transparent operații de pe noduri diferite de rețea, folosind standardul industrial OMG IDL și protocolul IIOP.

Java IDL este de fapt un nucleu ORB care respectă standardul CORBA 2.0, și care folosește limbajul de descriere a interfețelor, IDL. Acesta vine împreună cu un serviciu de nume compatibil Java, numit *Java IDL naming service*, care permite accesarea obiectelor printr-un set de standarde de interfețe.

3. DCOM (MFC COM/OLE)

COM, apărut în 1993, este o tehnologie de împachetare, un grup de convenții și biblioteci de suport care permit interacțiuni între diferitele părți soft, într-un mod consistent, orientat obiect. Obiectele COM pot fi scrise în diferite limbaje, cum ar fi C++, Java, Visual Basic, putând fi implementate în DLL-uri sau în executabile proprii care se execută ca procese distincte.

COM este un standard elaborat de Microsoft și specifică modul de interacțiune a componentelor unei aplicații distribuite, aplicația fiind constituită din obiecte care comunică între ele prin intermediul "interfețelor" și "metodelor". O interfață poate conține o sau mai multe metode și nu se poate schimba o dată ce a fost specificată.

Fiecare obiect trebuie să suporte obligatoriu o singură interfață, numită "Unknown", care conține trei metode, celelalte interfețe fiind la îndemâna proiectantului. Obiectul își oferă serviciile unui client, prin intermediul interfețelor, clientul trebuind să instanțieze obiectul din clasa respectivă pentru a beneficia de serviciile lui și apoi să obțină pointeri către interfețele suportate și metodele lor. Obiectele pot rula în același proces cu clientul, sub forma de bibliotecă

dinamica (DLL) sau printr-un proces separat.

COM asigura suport pentru mai multe **mecanisme de persistenta**, dintre care cele mai utilizate sunt:

- *persistenta bazata pe fisiere* – este cea mai simpla si la care un obiect încarca doar datele persistente dintr-un fisier;

- *memorarea structurata* – este o solutie bazata pe COM în care fiecare fisier contine un fel de sistem de fisiere, permitând unor obiecte COM sa partajeze acelasi fisier.

Obiectele COM, ca si alte tipuri de obiecte, ofera metode ce pot fi apelate de catre clienti, prin intermediul a doua tipuri de interfete:

- *vtable interfaces* – în special pentru clienti realizati în C++;

- *dispach interfaces* – în special pentru clienti realizati în limbaje mai simple, ca Visual Basic.

În 1996, o data cu lansarea lui Windows NT 4.0, a aparut **DCOM** (Distributed COM), un membru important al familiei ActiveX.

DCOM este o extensie a standardului COM pentru mediul distribuit. Serviciile unui obiect în retea sunt transparente pentru utilizator, mecanismul de comunicare dintre ele fiind **RPC** (Remote Procedure Calls). Mecanismul RPC permite unui proces care ruleaza pe un calculator sa apeleze o procedura care ruleaza în spatiul altui proces, de pe alt calculator, transmitând parametrii si preluând rezultatul returnat.

DCE (Distributed Computing Environment)

DCOM a fost construit pe un standard deja existent, numit **DCE (Distributed Computing Environment)**, comparativ cu celelalte tehnologii Java RMI sau CORBA IIOP care au pornit de la zero. Se mai numeste si ORPC (Object RPC) pentru a specifica faptul ca este o extensie orientata pe obiecte DCE RPC.

OSF (Open Software Foundation), format în 1988, o organizatie de cercetare non-profit, cu circa 350 membri, având ca scop dezvoltarea de sisteme deschise bazate pe

standarde a realizat o aplicatie pentru medii distribuite, numita **DCE** (Distributed Computing Environment). Acesta permite construirea, utilizarea si întreținerea aplicatiilor distribuite. *Serviciile distribuite* ale lui DCE, includ: *RPC* (Remote procedure Call), *SS* (Security Service), *DDS* (Distributed Directory Service), *TS* (Time Service) si Thread Service. *Serviciile de partajare a datelor* contin: *DFS* (Distributed File System), *DWSS* (Diskless Workstation Support Service), *PCIS* (Personal Computer Integration Service).

DCE furnizeaza un set comun pentru de interfete de aplicatie API, neutre din punct de vedere al producatorilor, ce permite unor module software diferite sa comunice si sa interactioneze între ele.

Din punct de vedere al tehnologiei orientate obiect, OSF si-a elaborat:

- propriul sau agent de schimb pentru management, numit **MRB (Management Request Broker)** ;
- propriul sau limbaj de definire a interfeței **IDL (Interface Definition Language)**, ambele facând parte din DCE.

DCE RPC este un mecanism RPC complex care permite serializarea informatiilor, codificarea datelor prin introducerea de nivele de autentificare, autorizare si integritate a mesajelor, precum si utilizarea de protocoale fara conexiune sau orientate pe conexiune. DCE RPC defineste 20 de mesaje diferite, numite unitati de date de protocol (PDU – Protocol Data Units) care sunt schimbate între clienti si servere. Fiecare PDU este format dintr-un antet care contine informatii de control, date si vericator de autenticitate.

DCOM nu aduce schimbari majore în modul în care clientul creeaza si interactioneaza cu obiecte COM, îmbunatateste mecanismul de securitate distribuita prin criptarea si autentificarea datelor, utilizeaza servicii ca DNS (Domain Name System) pentru localizarea obiectelor COM, permite neutralizarea fata de limbaje, independenta fata de

locatie, colectarea deseurilor, pastrarea versiunii componentelor.

Arhitectura DCOM cu diferitele moduri de invocare este prezentata în figura 2. Clientul apeleaza direct la metodele unei componente, daca aceasta este interna, iar daca este externa (la distanta) foloseste un proxy. Fiecare proxy are un "stub" corespunzator pe partea de server prin care comunica pentru apelarea metodei unei componente. Pentru scrierea de servere DCOM scalabile, Microsoft ofera MTS (Microsoft Transaction Server), care are suport pentru tranzactii si ofera servicii de urmarire automata si re folosire inteligenta a obiectelor. Aplicatiile MTS sunt recomandate a fi scrise în Visual Basic si reprezinta implementari DLL a unor obiecte COM. Pentru client, MTS este transparent, clientul utilizând obiectele COM ca si pâna acum.

În prezent Microsoft se ocupa cu portarea lui DCOM si pe alte platforme, cum ar fi Sun cu Solaris, IBM cu OS/390, cu mutarea lui DCOM în masina lor virtuala Java si cu extinderi ale DCE RPC continute în Windows NT 5.0, pentru a permite clientilor apeluri asincrone RPC.

4. ASP (Active Server Pages)

ASP (Active Server Pages) este un mediu de rulare a script-urilor pe partea de server, care poate fi folosit pentru crearea si rulare a aplicatiilor de tip Web interactive.

Tehnologia ASP a fost introdusa de Microsoft în 1996, sub numele de Denali.

ASP permite *includerea de scripturi executabile direct în continutul HTML, aplicatiile ASP fiind usor de creat, orientate obiect si extensibile cu alte componente ActiveX Server.*

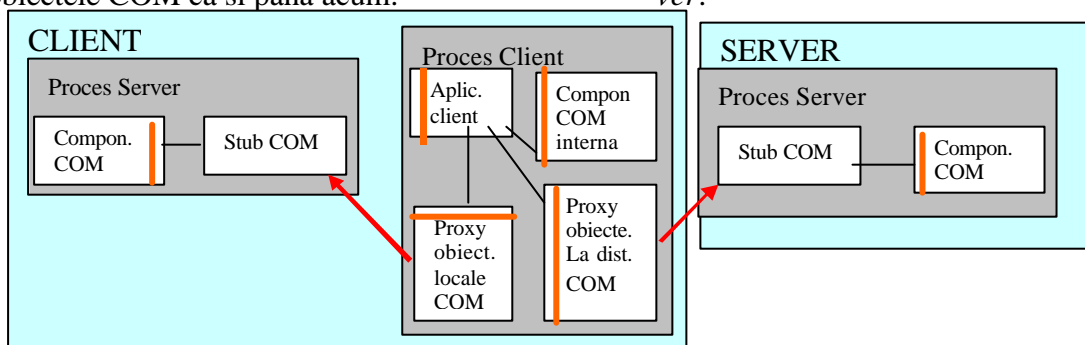


Figura 2

unde: → = comunica tii interne;
 → = comunica tii client/server;
 — =interfa ta

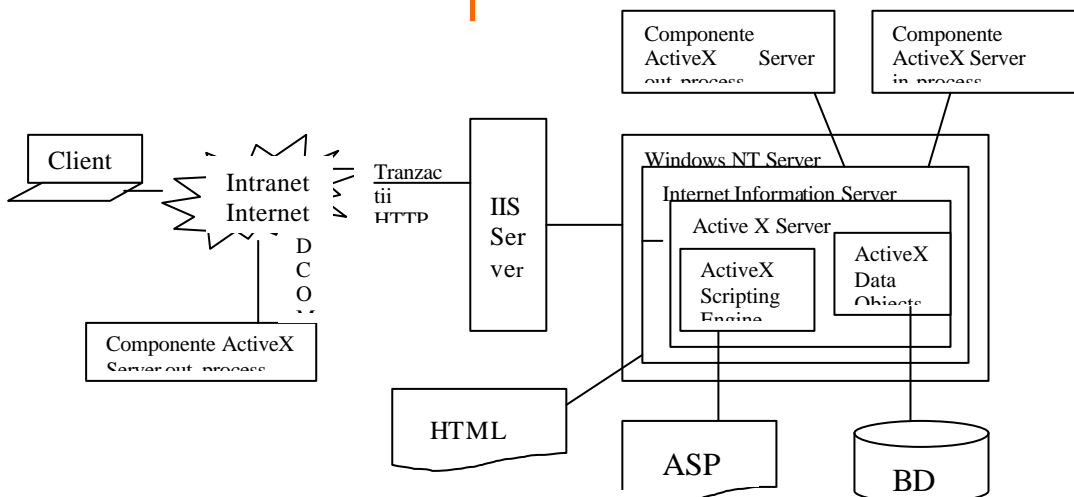


Figura 3

Aplicatiile ASP se realizeaza folosind limbaje script ASP, cum ar fi VBScript, JScript. Scriptul ASP presupune integrarea procesului de scripting cu cel de aranjare a informatiei în pagina ce va fi afisata utilizatorului.

ASP functioneaza cu urmatoarele servere Web:

- Microsoft Internet Information Server – (IIS) minim versiunea 3.0, care ruleaza sub Windows NT Server 4.0;
- Microsoft Peer Web Services 3.0, care ruleaza sub Windows NT Workstation;
- Microsoft Personal Web Server, care ruleaza sub Windows 95.

ASP lucreaza cu obiecte, si pentru a putea fi utilizate, obiectele trebuiesc instantiate de un program. ASP furnizeaza cinci obiecte care nu necesita instantiere, si anume:

- obiectul *Request* – furnizeaza informatia venita de la clientul HTTP care initiaza cererea; contine cinci tipuri de colectii: *QueryString*, *Form*, *Cookies*, *Server Variables*, *ClientCertificate*; accesul la acest obiect se face printr-o sintaxa de forma: *Reques.NumeColectie(variabila)*;
- obiectul *Response* – care trimite informatia catre client; contine metodele: *Write* (trimite informatia direct la browser), *Redirect* (trimite spre un URL), *ContentType* (controleaza tipul de continut trimis catre client), *Cookies* (asigneaza valori unice de identificare a clientilor);
- obiectul *Server* – care controleaza mediul de executie ASP;
- obiectul *Session* – care contine informatii legate de sesiunea de lucru deschisa de client cu serverul Web;
- obiectul *Application* – care contine informatii globale despre aplicatie, ce pot fi accesate de mai multi clienti.

Active Server Pages permite un înalt nivel de interactivitate pe Web, tranzactii comerciale sigure, creare de continut dinamic, toate acestea independent de software-ul clientului, acesta primind cod HTML pur.

Schema de ansamblu a functionarii Active Server-ului este prezentata în figura 3.

Codul ASP contine:

- *comenzi ASP native* – care acceseaza obiectele si componentele motorului Active Server; se utilizeaza între tag-urile `<% ... %>` sau `<SCRIPT>`;
- *comenzi de scripting* – au o sintaxa bine precizata în care trebuie specificat ce limbaj de script se foloseste, implicit fiind VBScript; pentru partea de client se utilizeaza comenzile între tag-urile `<SCRIPT> ...</SCRIPT>`, iar pe partea de server (fiind transparent pentru client) se foloseste tag-ul `<SCRIPT RUNAT=SERVER> ... </SCRIPT>`.

Scripturile sunt de doua feluri:

- scripturi client-side – sunt scripturile care se executa si sunt interpretate de browser si care acceseaza obiectele disponibile în browser; se pot crea de exemplu cu ActiveX Control Pad;
- scripturi server side – înainte ca o pagina sa fie transmisa clientului sunt executate anumite actiuni pe server, serverul fiind responsabil cu generarea codului HTML pentru client; astfel de scripturi sunt oferite de ASP pentru IIS, si permit includerea si de componente ActiveX.

Script-ul este alcatuit din comenzi care se executa pe client sau pe server si realizeaza o serie de actiuni cum ar fi: crearea de variabile, asignare de valori, operatii cu variabile, executare de proceduri, creare dinamica a script-urilor- client de catre server; Scripturile dintr-o pagina ASP sunt transmise motorului de scripting (scripting-engine), un obiect COM care este apelat pentru interpretarea scriptului si ale carui rezultate sunt executate de mediul gazda.

Facilitatile cele mai spectaculoase ale ASP-ului constau în **interactiunea cu bazele de date**, care devine foarte accesibila prin utilizarea *Active Data Objects*, **dezvoltarea de aplicatii complexe la nivel de server** în care se programeaza ASP la nivel de sesiune (*Session Object*) sau aplicatie (*Application*)

Object), precum si exploatarea Active Server si a componentelor de server IIS.

5. Cadrul San Francisco (IBM)

IBM si alti peste 100 de furnizori independenti de soft au realizat proiectul San Francisco, un cadrul de lucru multinivel si multiplatforma, bazat pe Java, care permite legarea componentelor în vederea obtinerii de aplicatii pentru afaceri.

San Francisco este un model alcatuit din 3 parti, si anume:

- **nivel de baza** – care include facilitati de tiparire, controlul conflictelor, persistenta, tranzactii si clase ale modelului de obiecte;
- **obiecte comune afacerii** – cum ar fi conversie de valuta, adrese, calendare;
- **procese esentiale ale afacerii** – cum ar fi contabilitatea, administrarea depozitelor si comenzilor.

Aceste nivele se afla intercalate între masina virtuala Java si aplicatiile de comert. Masina virtuala Java se poate afla pe platforme Windows NT, OS/2, OS/400, CISC (Customer Information Control System), MVS/ESA.

Dezvoltarea unei aplicatii folosind cadrul San Francisco depinde de nivelul pentru care se face si va avea nevoie de un mediu de dezvoltare Java, cum ar fi Symantec Visual Cafe sau Visual Age for Java, precum si de un instrument de modelare, cum ar fi Rational Rose.

În prezent, problema care se pune în legatura cu aplicatiile de afaceri scrise în Java este legata de viteza de executie, care este prea lenta. În acest sens, IBM lucreaza la îmbunatatirea performantelor masinii virtuale Java în propriile sisteme de operare, precum si la crearea unui compilator nativ pentru aplicatii Java pe server.

6. Web de obiecte

Paradigma curenta HTTP/CGI este în prezent depasita, datorita faptului ca nu poate satisface cerintele actuale, iar extensiile CGI, cum ar fi MSAPI (Microsoft Internet Services API), NSAPI (Netscape API), ASP

(Active Server Pages) sunt doar niste solutii temporare.

Cercetatorii Robert Orfali, Dan Harkey si Jery Edwards considera ca pentru nivelul urmator de dezvoltare a Web-ului, acesta are nevoie de obiecte distribuite, iar noul Web se va numi “Web de obiecte” (object Web). Acest lucru se poate realiza azi în doua moduri:

- **varianta CORBA –Java** – care încearca sa contrabalanseze solutia Microsoft;
- **varianta Active X - DOCM** – solutia proprie a Microsoft care își contruieste astfel propriul Web de obiecte.

Specialistii considera ca, **CORBA** este mai mult decât un ORB, si anume o *platforma foarte complexa de obiecte distribuite, care extinde domeniul aplicatiilor peste granitele retelelor, limbajelor, componentelor si sistemelor de operare*, iar **Java** este mai mult decât un limbaj cu legaturi CORBA, fiind *un sistem mobil de obiecte, un sistem portabil pentru rularea obiectelor*. Cele doua infrastructuri de obiecte se completeaza reciproc, Java începând acolo unde se termina CORBA, CORBA ocupându-se de transparenta retelei, iar Java cu transparenta implementarii.

Largimea infrastructurii Web-ului cu CORBA/Java ofera doua beneficii imediate, si anume:

- **eliminarea gâtuirilor din CGI**, clientul transmitând direct parametrii folosind stubul precompilat, serverul receptionând apelul prin intermediul unui skeleton precompilat;
- **infrastructura scalabila pe servere**, depozitele de obiecte server putând comunica printr-un ORB CORBA si putând actiona sincron folosind tranzactii si servicii CORBA.

Web-ul de obiecte CORBA/Java implica un amalgam de produse si protocoale, fiind alcatuit din trei treimi, si anume (figura 1.6):

- **client Web** – reprezinta browsere Web traditionale care suporta pagini HTML dinamice care contin ansambluri de Java Beans; Bean-urile server vor apela metodele Bean-ului client prin metode CORBA; pe

aceeasi retea vor putea rula **IOP** (pentru comunicarea client-server CORBA) si **HTTP** (pentru descarcarea paginilor Web, arhivelor Jar si a imaginilor);

- **server Web** – ruleaza pe orice server care poate deservi clienti CORBA si HTTP, fiind suportata de aproape orice platforma (Unixx, NT, OS/2, NetWare, MacOS, OS/400, MVS etc); obiectele CORBA pot fi împachetate ca Enterprise Java Beans, încapsuleaza logica afacerii si interactioneaza

- cu Java Bean-urile client prin CORBA/IOP; obiectele CORBA de pe server interactio-neaza între ele prin ORB CORBA; tot aici se pot crea magazine de titluri de obiecte, pa-gini HTML care sunt pastrate în arhive Jar, utilizare de ODBMS, DBMS si transmise clientilor;

- partea care contine **tot ceea ce un obiect CORBA poate accesa: monitoare TP procedurale, intermediarul orientat pe mesaje, DBMS-uri, ODBMS-uri, posta electronica.**

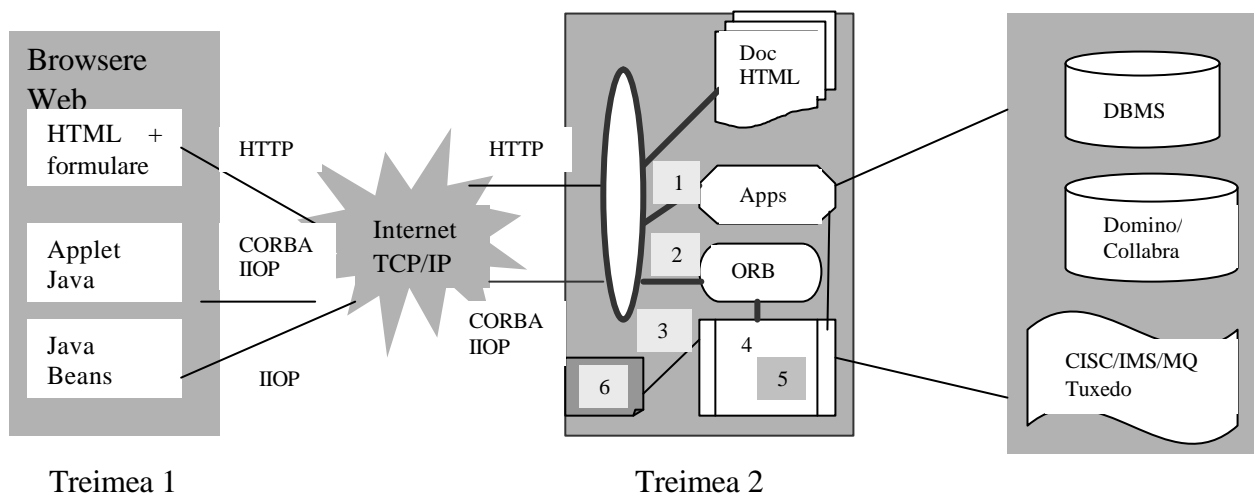


Figura 5, unde:

- 1- CGI; 2-CORBA; 3- IBM Component Browker, BEA Iceberg, Oracle NCA;
4- Coodonator componente; 5- Obiecte si Bean-uri; 6- JAR si ODBMS.

În concluzie, în paralel cu Microsoft care își construiește Web-ul propriu de obiecte bazat pe DCOM si ActiveX, CORBA si Java ofera o legatura arhitecturala care leaga produsele de pe Web de obiecte, fiind prima încercare a industriei actuale de a oferi plug-and-play la nivelul produselor soft, adica sistemul ideal deschis.

Bibliografie

- *** www.webhub.com, 1998
 *** www.aaa.org/sld012.htm, 1997
 *** www.omg.org/corba/cichpter.htm;
 *** CORBA 2.0/IOP Specification,
 www.omg.org/corba/corbaiiop.html
 *** java.sun.com/products/jdbc, 1997;
 *** Java Development Kit 1.2 beta 4, doc.;

- CLIP98 CLIP,P.,DCOM: Microsoft îmbunătățeste DCE, BYTE România, martie,1998;
 CRIS98 CRISTESCU, S., Comparatie JavaRMI-Java IDL, PC Report, aug.,1998
 DREY98 DREYFUS, P., Corba:Theory and Practice, home.netscape.com/, 1998;
 FOTA97 FOTACHE, C., Activarea Web-ului cu tehnologia ASP si Internet Information Server, PC Report, iunie 1997;
 MACE98 MACE,S.,s.a.,Pânza mai buna pentru Web, BYTE România, mart.,1998
 ORFA97 ORFALI, R., s.a., Corba, Java si Web-ul de obiecte, BYTE România, oct.1997;
 PANO98 PANONE, J., Noi lianti pentru aplicatii Web, BYTE România, iul.,1998
 PÂRV98 PÂRV, B., Modele obiect pentru POC, PC Report, iunie, 1998;
 RADD98 RADDING, A., Corba se desprinde, BYTE România, oct.,1998.