

Integrating Informatics Technologies into Oracle

Prof.dr. Manole VELICANU
Catedra de Informatică Economică, A.S.E. București

A characteristic of the actual informatics' context is the interference of the technologies, which assumes that for creating an informatics product, is necessary to use integrate many technologies. This thing is also used for database systems which had integrated, in the past few years, almost everything is new in informatics technology. The idea is that when using database management systems - DBMS the user can benefit all the necessary interfaces and instruments for developing an application with databases from the very beginning to the end, no matter the type of application and the work environment. For example, if the database application needs any Internet facilities these could be appealed from the products that the DBMS is working with offers. The concept of the interference of informatics technologies has many advantages, which all contribute to increasing the efficiency of the activities that develop and maintain complex databases applications.

Keywords: programming, database systems, human-machine interface, Java Platform, Grid Computing.

Introducere

Noțiunea de tehnologie este folosită în numeroase domenii de activitate din lumea reală, printre care și în informatică. Pe scurt, tehnologia se poate defini ca o știință a *procedurilor* (cum) și *mijloacelor* (cu ce) de prelucrare a unor resurse (materiale, informaționale, energetice etc.), având ca obiectiv obținerea unor *produse*. De exemplu, un produs software se obține prelucrând resurse informaționale (materia primă pentru un calculator) pe un suport material (hardware).

Tehnologia presupune un ansamblu de *procese* – prelucrări, *metode* - procedee de cunoaștere, *operatori* - operații efectuate asupra unor operanți, *condiții* - restricții de lucru, *mijloace* - instrumente și interfețe de prelucrare, toate utilizate în scopul obținerii unui anumit *produs*.

Informatica este știința care studiază (are ca obiectiv) culegerea, stocarea, prelucrarea și transmiterea datelor cu ajutorul mijloacelor electronice de calcul (calculatoarelor electronice). Informatica se suprapune de multe ori cu *tehnologia informației*, dar se ocupă în plus față de aceasta și de comportamentul automat al informației din natură și societate. Pentru a putea fi realizate produsele informatice, software și hardware, este necesar să se utilizează o serie de **tehnologii informatice**

adecvate.

Contextul informatic actual determină integrarea într-un produs informatic a mai multor tehnologii. Pentru produsele software **integrarea** tehnologiilor informatice este prezentă atât pentru produsele de bază cât și pentru cele aplicative. Marii producători de sisteme de gestiune a bazelor de date (SGBD) – Oracle, IBM, dar apoi și alții, au integrat în produsele lor ultimele tehnologii informatice apărute. În acest fel, SGBD actuale au devenit *infrastructuri pentru baze de date* în care se integrează numeroase tehnologii informatice: sisteme de baze de date, sisteme deschise, interfețe om-mașină, platforma Java, mediul Internet, OLAP – On Line Analytical Processing, programarea etc. În acest fel, SGBD devine un mediu complet și complex pentru dezvoltarea aplicațiilor cu baze de date.

Câteva dintre *avantajele integrării* sistemelor de baze de date (SBD) cu alte tehnologiilor informatice sunt:

- se reduc *costurile* de implementare, de exploatare (întreținere, depanare etc.), de conversie, costurile ascunse (cât te costă dacă nu ești pe piață la un moment dat);
- crește *protecția datelor* deoarece toate datele sunt supuse unor restricții uniforme;
- se învinge *complexitatea* generată de diferi-

te tehnologii utilizate, dar și de dinamica lor (diferite versiuni);

- dispar problemele de *scalabilitate*, adică nu vor fi mai multe produse cu mai multe scale;
- dispar problemele legate de *gestionarea* diferitelor produse software (cunoștințe, timp, spațiu, compatibilitate), fiind utilizat un singur produs;
- dacă sunt mai multe produse apar mai multe probleme la *întreținere*;
- utilizatorii au nevoie de *evoluție* nu de *revoluție*.

Un exemplu tipic privind integrarea bazelor de date cu alte tehnologii informatice este infrastructura pentru baze de date *Oracle 10g*. Aici sunt integrate cele mai noi tehnologii informatice pentru a se putea dezvolta, de la început – analiză, proiectare, până la sfârșit – implementare, întreținere, aplicații cu baze de date relaționale. În cele ce urmează voi prezenta, pe scurt, câteva dintre aceste tehnologii, cu exemplificare pe *Oracle 10g*.

1. Interfața om-mașină

Pentru toate categoriile de utilizatori în informatică, interfața om-mașină atașată produselor software are o mare importanță. În ultima perioadă, din totalul codului sursă al unei aplicații informatice peste 50% este destinat interfeței om-mașină. Acest lucru a fost necesar și posibil în același timp datorită lucrului masiv cu microcalculatoarele, ca mașini interactive, ceea ce presupune că inițiativa aparține concomitent calculatorului și utilizatorului. Stilurile de lucru mai vechi: pe loturi – *batch* și conversațional au fost abandonate. Atât forma cât și conținutul interfeței om-mașină au evoluat în strânsă legătură cu sistemele de operare care oferă interfețe grafice prietenoase. De aici, produsele software care rulează sub astfel de sisteme de operare au adoptat același stil.

Toate interfețele și instrumentele unui SGBD lucrează acum în același stil interactiv prietenos și permit dezvoltarea unor aplicații cu baze de date care vor avea interfețe om-mașină de asemenea interactive. În acest scop, SGBD oferă asistenți tip *Wizard* și generatoare care permit dezvoltarea aplicațiilor ușor și rapid. Productivitatea activității de dezvoltare a aplicațiilor cu baze de date a crescut

astfel foarte mult, codul sursă scris de dezvoltator fiind foarte puțin sau chiar deloc. Dezavantajul acestei soluții este că portabilitatea aplicației rezultate va avea de suferit.

Necesitatea unei interfețe om-mașină cât mai bune apare în special la dezvoltarea machetelor de intrare pentru aplicațiile cu baze de date. În acest sens, se pot construi videoformate – ecrane formate, forme - în mod eficient cu ajutorul generatorului de forme. Toate SGBD actuale au astfel de generatoare.

Oracle 10g are pachetul *Oracle Developer Suite* din care face parte *Forms Builder*, care permite generarea de videoformate în interacțiune cu baza de date relațională. Astfel, se pot construi interactiv ecrane care să conțină tot felul de obiecte vizuale – texte, butoane, ferestre imagini etc. - aranjate astfel încât să corespundă documentelor primare pe care le folosește utilizatorul în mod curent. *Oracle Forms* respectă tendința de integrare a sistemului *Oracle* și permite utilizarea diferitelor tehnologii informatice: programarea vizuală, programarea orientată obiect, acces la baza de date relațională, interacțiunea cu celelalte componente *Oracle*, utilizarea unor componente din platforma *Java*, servicii *Web*. Cu ajutorul videoformatelor date sunt preluate, în mod interactiv, de la utilizator și încărcate în baza de date *Oracle* respectându-se restricțiile impuse la descrierea datelor conform modelului relațional.

2. Limbajele de programare

Rolul limbajelor de programare a fost, și rămâne și acum, de mare importanță pentru specialiștii în informatică. Pentru dezvoltarea unei aplicații cu baze de date se folosesc de obicei mai multe limbaje de programare. Acestea fac parte toate dintr-un SGBD, cazul cel mai întâlnit, sau doar o parte din ele aparțin unui SGBD.

Un *limbaj de programare* este o mulțime de cuvinte rezervate, cu o anumită semnificație, împreună cu gramatica aferentă - reguli de sintaxă și semantică. Cu alte cuvinte, el semnifică orice limbaj folosit pentru descrierea datelor și a fenomenelor din lumea reală, care este înțeles de un calculator. Descrierea fenomenelor poate fi făcută algoritmic și/sau declarativ, rezultând limbaje de programare

corespunzătoare. Programatorul pleacă de la lumea reală, identifică fenomenele și datele corespunzătoare, realizează o descriere a lor prin specificații de programare, pe care le folosește pentru întocmirea schemelor logice sau a pseudo-codurilor aferente. În continuare, își alege un limbaj de programare în care scrie *programele sursă* aferente. Acestea sunt apoi "rulate" pe calculator parcurgând *fazele*: compilare - pleacă de la programul sursă și produce programul obiect, editare de legături (link-edit) - pleacă de la programul obiect și produce programul executabil, execuție - pleacă de la programul executabil și produce rezultate.

SGBD actuale dețin mai multe tipuri de limbaje de programare astfel încât să se poată dezvolta tot felul de aplicații cu baze de date. În funcție de domeniul de utilizare, limbajele de programare pot fi: **de asamblare** - specific fiecărui tip de calculator - ASM pentru PC; **procedurale** - imperative / algoritmice, care pot fi *universale* (se folosesc pentru orice domeniu din lumea reală - Fortran, Cobol, Basic, Pascal, C etc.) sau *specifice* / speciale (se folosesc doar în anumite domenii - baze de date: PL/SQL, limbajul din Visual Foxpro); **neprocedurale** - descriptive / declarative: *limbaje relaționale* (SQL, QUEL, QBE), *limbaje orientate obiect* (Smalltalk, Delphi, C++, Java), *limbaje pentru inteligența artificială* (LISP, PROLOG), *limbaje de simulare* (Simula, Ada); *limbaje de scenarii* - script (HTML, JavaScript, BasicScript, PHP).

Orice SGBD are cel puțin un limbaj de programare cu instrucțiuni pentru *descrierea* datelor - LDD și instrucțiuni pentru *manipularea* datelor - LMD. Aceste limbaje permit implementarea unui model logic de date pentru baze de date, deci sunt specifice. Pe lângă acestea, majoritatea SGBD dețin sau au interfețe cu alte limbaje de diferite tipuri. Scopul este ca SGBD să ofere toate facilitățile de programare necesare pentru dezvoltatorul de aplicații cu baze de date. Acest lucru este necesar deoarece majoritatea problemelor din lumea reală implică, pentru informatizare, utilizarea mai multor tipuri de limbaje de programare, deoarece fiecare oferă anumite

facilități și avantaje.

Oracle deține, în acest moment, în nucleul sistemului mai multe limbaje de programare: *SQL* ca limbaj relațional pentru descrierea și manipularea datelor în BD relațională, *PL/SQL* ca limbaj procedural propriu, *Java* ca limbaj orientat obiect și de comunicație, *extensie* de programare orientată obiect în PL/SQL, *precompilatoarele* - Pro* ca interfețe de programare cu limbajele universale - Fortran, Cobol, Pascal, C, Ada, PL1. Rezultă așadar că Oracle oferă dezvoltatorului posibilitatea de a scrie cod sursă în limbajele de programare pe care le știe sau de care are nevoie, practic în orice tip de limbaj.

3. Platforma Java

O tehnologie de actualitate utilizată pentru dezvoltarea aplicațiilor cu baze de date este platforma Java. Aceasta a fost realizată și promovată de firma Sun Microsystems USA. În construcția platformei s-a plecat de la realizarea limbajului de programare Java în anul 1995, ca o soluție de adaptare a limbajelor de programare universale la noul context informatic, dominat în principal de mediul Internet.

În jurul limbajului de programare Java, ulterior, au fost create noi produse, bazate pe tehnologia Java, toate împreună formând *platforma Java* actuală. Aceste produse sunt:

- *limbajul de programare Java* - limbaj universal orientat obiect și pe comunicație;
- limbajul de scenarii *JavaScript* - limbaj specific destinat construcției de pagini *Web* interactive;

- *Servlets* - clase (module) Java, necesare pentru construirea unor pagini *Web* dinamice, încărcate și menținute rezident pe server;

- *Java Server Pages* (JSP) - interfață utilizată pentru construirea unor aplicații care să conțină pagini *Web* cu un conținut dinamic, utilă pentru construirea aspectelor estetice ale paginilor *Web* (amplasare, aspect etc.) și mai puțin utilă în ceea ce privește sursa și manipularea datelor dinamice;

- *Java DataBase Connectivity* (JDBC) - standard pentru conectarea unei baze de date cu alte sisteme. El este o dezvoltare a tehnologiei ODBC - Open DataBase Connectivity, ținând cont de cerințele mediului Internet: in-

teroperabilitate, integrare, deschidere, comunicare, portabilitate. JDBC este constituită dintr-un set de clase și interfețe scrise în Java, furnizând mecanisme standard de comunicare pentru proiectanții aplicațiilor de baze de date;

- *Java Beans* - produs construit pentru a permite realizarea unor componente software independente, din care utilizatorii să poată asambla apoi aplicații complexe;

- *Enterprise Java Beans (EJB)* - arhitectură pentru dezvoltarea aplicațiilor distribuite, tranzacționale și bazate pe componente. Aplicațiile EJB pot fi dezvoltate complet în Java, prin obiecte tip container. Specificațiile EJB detaliază nu numai formatul unei componente ci și un set de servicii ce trebuie furnizate de container, reprezentând totodată o metodologie pentru construirea aplicațiilor distribuite;

- *Business Components for Java (BC4J)* - module Java prefabricate care pot fi utilizate într-o aplicație multi-strat pentru a realiza: o interfață utilizator pe partea de client scrisă în Java, Java Script și/sau HTML; una sau mai multe componente pe stratul de mijloc care se referă la "logica afacerii" și la obiectele de tip "afacere"; tabelele din baza de date ce conțin datele necesare aplicației;

- *Java to Enterprise Edition Edition (J2EE)* - mediu de dezvoltare care furnizează o platformă de lucru, bazat pe componente și presupune realizarea de aplicații multistrat. În acest fel, aplicația poate fi segmentată pentru a rula pe mai multe dispozitive, aceste dispozitive fiind alese în funcție de posibilitatea aplicației de a realiza o anumită sarcină;

- *SQL Java (SQLJ)* – interfață care permite programatorilor să includă comenzi SQL în codul sursă Java, oferind însă și extensii pentru suportul operațiilor SQL dinamice.

Oracle 10g are inclusă întreaga platformă Java, cu toate componentele de mai sus în interfața JDeveloper. Aceasta permite dezvoltarea completă a unei aplicații cu baze de date folosind tehnologia Java, în mediul Internet.

4. Arhitectura Grid Computing

Ultima arhitectură de lucru în mediul de rețea de calculatoare, la nivelul unei firme, este cea denumită Grid Computing - GC. Ea a dezvoltat

alte tehnologii informatice existente deja – Internet, Intranet, baze de date distribuite etc. printr-o organizare eficientă și optimizare a resurselor de calcul la nivelul firmei. *Grid Computing* reprezintă utilizarea coordonată a mai multor servere mici care acționează ca un singur sistem foarte puternic - Grilă de Calculatoare. Pentru utilizatori care lucrează în tehnologia GC nu mai contează: unde sunt stocate datele, unde sunt stocate aplicațiile, ce calculatoare procesează cererea de regăsire, ce resurse sunt folosite în rețea.

Enterprise Grid Computing - EGC înseamnă procesul de punere la lucru împreună a mai multor calculatoare existente la nivelul unei organizații și funcționarea ca un sistem integrat.

Arhitectura GC evidențiază grafic diferitele componente și legăturile dintre ele. Structurarea se face pe patru niveluri (aplicații, server de aplicații, baze de date, echipamente) conectate prin mecanisme de interfață de servicii și monitorizare (control).

Oracle este prima companie care a promovat tehnologia GC, într-un SGBD – Oracle 10g, pentru întreprinderi. Astfel, Oracle oferă o infrastructură software completă și comercială, proiectată special pentru GC cu baze de date. Oracle 10g permite adaptarea sistemelor informatice la modelul EGC, care folosește puterea de prelucrare a unui mare număr de calculatoare, puțin costisitoare, ce acționează ca un tot unitar.

Bibliografie

- [1] Velicanu M. – *Dicționar explicativ al sistemelor de baze de date*, Ed. Economică, 2005.
- [2] Date J. – *An introduction to Database Systems*, ed. Addison Wesley, 2004.
- [3] Velicanu M., Lungu I. s.a - *Sisteme de baze de date – Teorie și practică*, Ed. Petron, București, 2003.
- [4] Velicanu M., Lungu I., Muntean M., Iorga M., Ionescu S. - *Oracle – platformă pentru baze de date*, Ed. Petron, București, 2002.
- [5] *** Oracle – Documentație de firmă
- [6] G. Riccardo – *Principles of Database Systems - with Internet and Java Applications*, ed. Addison Wesley, 2001.
- [7] *** Oracle Magazin, 2005