

SISTEM INFORMATIC PENTRU PREZENTAREA UNOR CURSURI INTERACTIVE, LA DISTANȚĂ (E-LEARNING)

dr. ing. Florin Hartescu
drnd. mat. Florin Boboșatu

Institutul Național de Cercetare-Dezvoltare în Informatică – ICI București

Rezumat: Lucrarea este dedicată tehnicilor moderne de învățare cu ajutorul calculatorului, fiind elaborat un sistem informatic original, care permite prezentarea unor cursuri interactive, la distanță, exploatând facilitățile oferite de multimedia și Internet.

Cuvinte cheie: server, client, socket, e-Learning, portal.

1. Introducere

Învățarea cu ajutorul calculatorului a cunoscut o rapidă dezvoltare datorită posibilităților hardware, care au crescut spectaculos, în ultimul timp. Principalele motive care au stat la baza acestei dezvoltări au fost generate de interfața prietenoasă și de posibilitățile oferite de domenii noi, cum ar fi Internetul, Inteligența Artificială.

Lucrarea prezintă un sistem informatic Client-Server original, care permite prezentarea unor cursuri interactive, la distanță, prin Internet, folosind concepte și metodologii (e-Learning) noi de transmitere de date.

E-learning este o industrie relativ nouă și insuficient explorată, ce presupune acoperirea unui set de aplicații și de procese vaste, bazate pe învățarea cu ajutorul calculatorului. E-learning înseamnă distribuirea conținutului informației pe cale electronică (media, Internet, Intranet), iar în tehnica de specialitate este definită mult mai precis decât învățământul la distanță.

Avantajele unui sistem e-learning constau în posibilitatea cursantului de a-și gestiona singur timpul aferent studiului și în flexibilitatea sistemului ce încurajează stilul propriu de învățare.

Din această cauză, în ultima perioadă, învățământul tradițional începe să piardă teren, lăsând loc învățământului cu ajutorul calculatorului, acest lucru datorându-se, în principal, costului mic, flexibilității și posibilității de a lăsa pe student (cel care studiază) să-și împartă timpul liber așa cum dorește.

În continuare, vom prezenta cerințele care trebuie îndeplinite de orice aplicație de e-Learning, pentru a satisface cerințele celui care o utilizează:

- dinamica foarte mare a informației într-un domeniu;
- necesitatea de învățare foarte rapidă;
- găsirea de metode eficiente din punct de vedere al costurilor pentru grupuri distribuite geografic;
- niveluri de pregătire diferite (schimbările demografice cer noi metode de învățare);
- acces flexibil la procesul de învățare

Suport:

- accesul la Internet;
- tehnologiile digitale (care oferă un context multimedia și interactiv)
- crearea de servicii și produse e-learning de calitate (standardele tehnologice facilitează compatibilitatea și interoperabilitatea între produsele e-learning).

2. Arhitectura Hardware

Lucrarea are la bază o aplicație software, Client-Server care permite prezentarea unor cursuri interactive, la distanță cu ajutorul facilităților oferite de multimedia și Internet, folosind concepte și metodologii (e-Learning) noi de transmitere de date.

O configurație reală a aplicației de e-Learning este prezentată în figura 1:

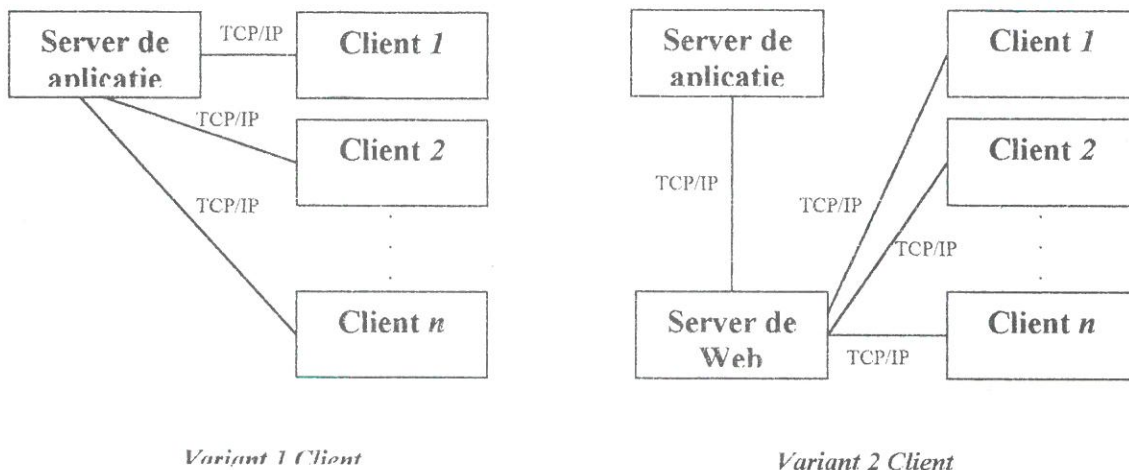


Figura 1. O configurare reală a aplicației de e-Learning

Pentru comunicația între calculatoare s-a folosit protocolul TCP/IP, care a fost implementat inițial pentru sistemul de operare UNIX deoarece:

- permite programarea relativ simplă a aplicațiilor de comunicare și transmitere de date;
- a fost folosit de rețeaua Internet ca standard pentru ca marea majoritate a firmelor, printre care și DIGITAL (care a dezvoltat DECnet), NOVELL (creatorul IPX/SPX), Microsoft (folosea NetBEUI inițial) au implementat acest protocol în aplicațiile lor.

Observație:

1. Pentru ca două calculatoare să poată comunica prin protocolul TCP/IP, este necesar ca unul să îndeplinească rolul de server, iar celălalt de client.
2. Orice calculator cu un sistem de operare, care acceptă TCP/IP, poate avea rol de server sau de client.
3. Menționăm că e posibil ca pe același calculator să ruleze simultan (într-un sistem multitasking) mai multe aplicații server și mai multe aplicații client.
4. Conectarea și schimbul de date se pot realiza și între o aplicație server și una client, care rulează pe aceeași calculator, deși protocolul TCP/IP se folosește, mai ales, pentru realizarea conexiunilor în rețea.

Comunicarea se realizează pe un port (preferabil mai mare decât 700, pentru a nu coincide cu porturile ocupate de serviciile de sistem). La un server se pot lega mai mulți clienți, iar un client poate să comunice cu un singur server. Un server este unic identificat prin adresa sa de IP (patru întregi cu valori de la 0 la 255, separați prin puncte, de exemplu "193.230.32.55"). Fiecare calculator poate avea o singură adresă de IP care se scrie în fișierul *hosts*.

Conexiunea clientului la server s-a realizat prin *socket*-uri.

Un *socket* este, de fapt, un obiect software (de tip întreg), la care se face referirea pentru a apela funcții de rețea.

Astfel, server-ul deschide un socket de ascultare a rețelei pentru a putea accepta conexiunea de la orice client care solicită acest lucru. Când un client solicită conectarea, serverul o acceptă și va deschide un nou socket de comunicație cu acel client. Pentru fiecare nou client, serverul deschide un nou socket de comunicație. La închiderea conexiunii, serverul închide socket-ul de comunicație, iar clientul, la rândul său, închide socket-ul. Clientul are numai un socket de conectare la server. Când s-a conectat, el va folosi socket-ul pentru comunicație. Serverul are un socket deschis permanent pentru acceptarea de noi clienți, cu care "ascultă" rețeaua și câte un socket pentru fiecare client conectat.

3. Structura programului software

Aplicația are două module: **server** și **client**, dezvoltate în limbajul de programare JBuilder. Acestea pot rula atât sub sistemul de operare Windows NT 4.0, cât și pe Linux RedHat 7.0. Modulul **Server** este un program care rulează în Background și stabilește conexiuni cu clienții. Rolul lui este de a prelua informații de la clienți și de a le distribui. Modulul **Client** este o aplicație care trimite informații către server și preia informații furnizate de el.

Menționăm că, pentru modulul **Client**, a fost realizată și o interfață Web, care conține aproape toate facilitățile oferite de modulul **Client**, realizat în **JBuilder**.

Pentru conectarea clientului (studentului) în varianta I la server sunt necesare următoarele:

- pornirea server-ului: acesta este un fișier *server.jar* care se lansează cu dublu clic din *Windows Explorer*, după ce în prealabil au fost setate căile către *Java Virtual Machine* pentru fișiere cu extensia *jar*;
- pornirea efectivă a clientului: dublu clic pe fișierul *client.jar* din *Windows Explorer*. Pe ecran apare fereastra următoare:

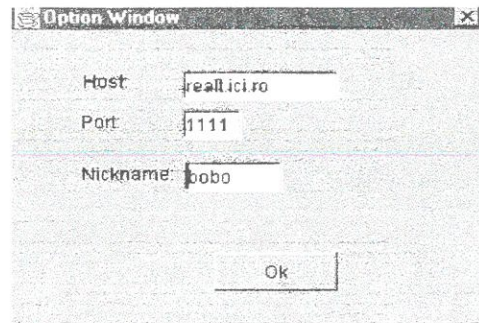


Figura 2. Fereastra cu informațiile de conectare la server

unde:

Host – este numele mașinii pe care este instalat serverul pentru soft-ul e-Learning. "real.ici.ro" este numele calculatorului pe care a fost testată aplicația;

Port – numele portului pe care se realizează conexiunea;

Nickname – numele studentului care participă la cursuri;

După setarea opțiunilor inițiale, apare pe ecran fereastra cu meniul principal (figura 3) în care avem:

- lista studenților;
- câmpul în care se pun întrebări studenților, se răspunde la întrebări studenților prin simpla editare sau prin redirectare către fișiere text, imagini, sunete etc.

Pentru trimiterea fișierelor text, acestea trebuie să se găsească într-un director setat inițial (ex. d:\f pentru sistemul de operare Windows NT, respectiv /usr/f/ pentru sistemul de operare Unix). Comanda care trebuie scrisă în câmpul de editare este de forma:

<fișier.ext

unde:

fișier – numele fișierului text transmis ca răspuns la întrebările studenților;

ext – extensia fișierului;

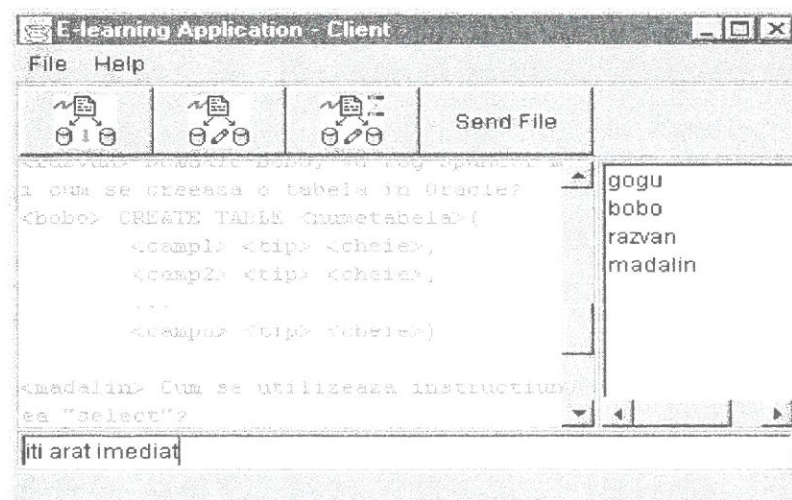


Figura 3. Meniul principal al clientului aplicației de e-Learning

În varianta II, modul de conectare la server se face folosind interfața Web. Pentru aceasta, se deschide un Browser de Web (Internet Explorer sau Netscape Communicator) și în Address Bar se lansează URL-ul <http://<adresa server>/client/AppletClass.asp> (de exemplu: <http://real.ici.ro/client/AppletClass.asp>).

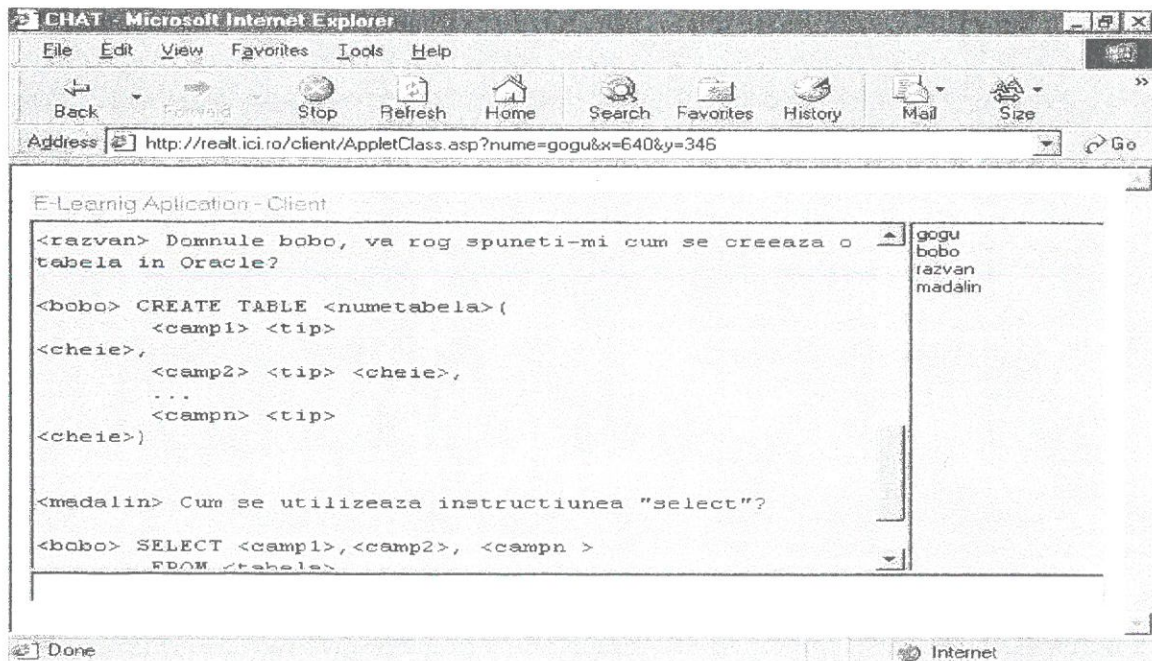


Figura 4. Modul de conectare la server folosind interfața Web

Concluzii

Creșterea continuă a volumului de informații și utilizarea tehnologiilor moderne necesită acumularea și înțelegerea unui număr tot mai mare de cunoștințe și de informații, de către tot mai mulți oameni. Tehnicile moderne de învățare, pentru a fi eficiente, trebuie să aibă un anumit grad de interactivitate cu subiectul uman și să transmită informația pe mai multe canale (text, sunet și imagine) într-o manieră asociată. Forma nu trebuie să fie excesiv de variată, dar suficient de adaptată conținutului, pentru a transmite ideile cât mai firesc.

Dezvoltarea explozivă a Internetului și a tehnologiilor Web a creat premisele unor oportunități și unor noi provocări pentru relațiile dintre oameni. În această nouă economie digitală, există pericolul ca o instituție să constate că nu poate face față avalanșei de informații și că propria rețea Intranet a ajuns la un nivel de saturare, care nu îi mai poate asigura gestionarea eficientă a datelor. Pentru a rămâne în "business", o astfel de organizație nu trebuie decât să implementeze un portal care să îi permită optimizarea sigură și scalabilă a accesului la informație, precum și o administrare centralizată a tuturor serviciilor prin intermediul unui simplu Browser. Într-un studiu IDC (International Data Corporation), se estimează că, până la sfârșitul anului 2001, peste 50% din companiile mari și de mărime medie își vor construi portaluri de corporație. Astfel, aplicațiile e-Learning vor deveni mai viabile și mai utile.

Bibliografie

1. ROSENBERG, M.J.: E-Learning: Strategies for Delivering Knowledge in the Digital Age, McGraw-Hill, Professional Publishing, October, 2000.
2. HALL, B.: Web-Based Training Cookbook, John Wiley & Sons, 1997.
3. HORTON, W.K.: Designing Web-Based Training, John Wiley & Sons, September, 2000.
4. KRUSE, K., KEIL, J.: Technology-Based Training : The Art and Science of Design, Development, and Delivery, Jossey-Bass Publishers, June 2000.

5. **KRUSE, K., KEIL, J.:** Web-Based Training: Using Technology to Design Adult Learning Experiences, May 1999.
6. **HARTESCU, FL. ș.a.:** Software Package for an Automatic Data Monitoring System, ICI, Bucharest, 1999.
7. **HAUSLEIN, A., PAGE, B.:** Knowledge-Based Approaches to Modelling and Simulation Support. În: System Analysis, Modelling Simulation 8, 1991, 4/5, pp. 257-272.
8. **LUSARDI, F.:** The Database Experts'Guide to SQL, McGraw - Hill, 1990.
9. **NUSSBAUMER, H.:** Computer Communication Systems, John Wiley & Sons, 1990.
10. **OLSHEVSKY, V., PONOMAREV, A.:** OOP Using C++, Wrox Press Ltd. 14. Win32 Programming Using Visual C++, Myke Blaszczak, Wrox Press Ltd.
11. **PAGE, B.:** Environmental Computing - Status and Research Perspective. În: P. Zanetti (ed.), Computer Technique in Environmental Studies. Proc. ENVIROSOFT'88, 2nd International Conference, Porto Carracs Greece, September 1988, Computational Mechanics, Southampton - Boston, 1988, pp. 597-606.
12. **PETZOLD, C.:** Programming in Windows, 1990.
13. **SELIC, B., WARD, P.T.:** The Challenges of Real Time Software Design, Embedded Systems Programming, Miller Freemans Inc., October, 1996.
14. **SELIC, B., GULLIKSONS, G., WARD, P.T.:** Real Time Object Oriented Modelling, John Wiley & Sons, 1994.
15. **TANENBAUM, S.A.:** Computer Networks, Prentice Hall, 1989.