

# Controlul aplicațiilor ce funcționează pe sisteme Windows prin intermediul dispozitivelor Android

Cristina Șerban, Irina Grosu, Alexandra Sirițeanu, Alexandru Averescu, Adrian Iftene

Universitatea "Al. I. Cuza" din Iași, Facultatea de Informatică

General Berthelot, Nr. 16, 700483, Iași

E-mail: {cristina.serban, irina.grosu, alexandra.siriteanu, alexandru.averescu, adiftene}  
@info.uaic.ro

**Rezumat.** Articolul de față prezintă o aplicație client-server, care permite unui utilizator să controleze de la distanță prin intermediul unei componente Android aplicațiile ce funcționează pe un calculator pe care rulează sistemul de operare Microsoft Windows. Aplicația este compusă din două module importante: *clientul* (care funcționează pe telefonul utilizatorului) și *serverul* (care rulează pe calculator), adăugându-se protocoalele de comunicare dintre cele două module (bazate pe Bluetooth și pe Wi-Fi). Testele pe care le-am făcut asupra acestei soluții ne-au demonstrat robustețea aplicației la utilizare pe o perioadă mare de timp, la teste de stres și la comunicarea când apar obiecte între client și server. În articolul de față, vom face o prezentare detaliată a acestui sistem, urmată de o evaluare, bazată pe câteva studii de caz și teste relevante.

**Cuvinte cheie:** Android, servicii Windows, control la distanță.

## 1. Introducere

În ultimii ani, am asistat la o explozie a diversității dispozitivelor electronice ce ne permit să ne conectăm la Internet pentru a citi știrile sau e-mailul, pentru a socializa prin intermediul rețelelor sociale, pentru a asculta muzică sau pentru a vizualiza prezentări de tip PDF sau Power Point (laptop-uri, notebook-uri, palmtop etc.). Lucrarea de față își propune să prezinte modul în care putem adăuga acestor tipuri de dispozitive posibilitatea de a controla astfel de aplicații de la distanță. Motivația ar fi să eliminăm necesitatea atât a unor dispozitive specifice care să ne permită să derulăm prezentări PowerPoint sau documente de tip Word sau PDF la conferințe, cât și cea a unor telecomenzi atunci când ascultăm muzică de pe calculatorul personal.

În prezent, asistăm la apariția multor instrumente de acest tip, vezi, de exemplu, Vectir [8], o aplicație ce rulează pe dispozitive ce suportă Java. Aceasta poate fi folosită gratuit doar pentru 30 de zile și permite controlul prin Bluetooth și Wi-Fi a unor aplicații precum Winamp, Windows Media Player, iTunes sau PowerPoint. Avantajele folosirii conexiunii Wi-Fi în loc de Bluetooth includ un timp de răspuns mai bun și o rază mai mare de acțiune. În timpul folosirii aplicației Vectir, utilizatorii pot vizualiza monitorul, pot naviga de-a lungul ecranului, pot mări sau micșora rezoluția acestuia, și, în plus, pot trimite mesaje text către calculator. Vectir pune la dispoziție dezvoltatorilor de software pentru telefonie mobilă o bibliotecă ce conține documentație și exemple de aplicații pe baza cărora pot fi realizate plugin-uri utilizându-se platforma .NET.

PPT Remote [5] este o aplicație dezvoltată pentru telefoanele mobile ce utilizează sistemul de operare Android și care pot fi conectate la calculator prin Wi-Fi sau Bluetooth, oferind posibilitatea de a controla prezentări de tip Power Point. Utilizatorul poate vizualiza atât diapozitivul curent cât și notițele aferente acestuia pe ecranul telefonului, poate seta perioada alocată prezentării, pentru a ține evidența timpului scurs și poate sări către orice diapozitiv.

O altă aplicație, Remote Presenter for Android [6] utilizează capacitățile tehnologiei Bluetooth pentru a controla de la distanță prezentări Microsoft Power Point, Keynote, OpenOffice și fișiere PDF, fiind compatibilă cu sistemele de operare Windows, Mac și Linux. Conexiunea se realizează cu ajutorul unui scanner de bare.

Un exemplu de aplicație iOS este Tango Remote Control Media Player HD [7], disponibilă în 16 limbi și menită să controleze prin Wi-Fi sau Bluetooth aplicația iTunes (<http://www.apple.com/itunes/>) pentru Mac.

WIN – Remote Service [9] este realizat exclusiv pentru sistemele Windows și poate fi accesată atât prin intermediul unei tastaturi, cât și prin intermediul unui mouse. Acesta permite vizualizarea ecranului pe telefon, răsfoirea fișierelor din computer, navigarea printre programele deschise cu ajutorul unui Task Manager, controlul volumului, vizualizarea de imagini și prezentări, utilizarea unui navigator web și schimbarea stării calculatorului și a monitorului. Conexiunea poate fi creată atât prin Wi-Fi, 3G sau 4G pe baza unei adrese IP, cât și prin Bluetooth, cu condiția ca dispozitivul mobil și calculatorul să fie deja interconectate.

Alte aplicații precum Xtralogic Remote Desktop Client [10] pun la dispoziție doar conectarea la ecranul calculatorului și navigarea cu ajutorul mouse-ului și/sau a tastaturii, nefiind definite acțiuni pentru accesarea anumitor aplicații.

Soluția pe care am realizat-o permite controlarea prezentărilor PowerPoint și a aplicației Windows Media Player atât prin Bluetooth cât și prin Wi-Fi. Prin intermediul ecranului tactil al telefonului dumneavoastră puteți porni o prezentare Power Point sau puteți trece la diapozitivele următoare sau la cele anterioare. Similar, în cazul aplicației Media Player, puteți asculta lista favorită de melodii, puteți schimba melodiile care sunt redade sau puteți modifica nivelul volumului. Pentru toate acestea nu mai este nevoie să fiți în fața calculatorului, ci o puteți face mult mai ușor de oriunde v-ați afla, folosind ecranul tactil al telefonului dumneavoastră.

Spre deosebire de aplicațiile prezentate mai sus, programul nostru nu permite vizualizarea ecranului calculatorului la care este conectat dispozitivul mobil, iar navigarea prin diapozitive sau prin lista de melodii nu se realizează cu ajutorul unui cursor, ci prin intermediul unor butoane virtuale. De asemenea, aplicația noastră nu permite posibilitatea editării prezentărilor Power Point. La fel ca Vectir [8], aplicația este realizată pe platforma .NET, ceea ce înseamnă că nu este portabilă pe alte sisteme – Linux, Mac.

În cele ce urmează vom prezenta pe scurt soluția pe care am construit-o, oprindu-ne asupra elementelor sale principale, componenta client și componenta server. În partea de final, facem o evaluare a acestui sistem, prezentând principalele avantaje și dezavantaje ale folosirii acestuia.

## 2. Prezentarea soluției

### 2.1 Arhitectura

Aplicația este compusă din două părți principale: *componenta client* (aplicația Android ce funcționează pe telefon) și *componenta server* (aplicația Desktop ce rulează pe sistemul de operare Windows). Comunicarea dintre acestea poate fi realizată atât prin Bluetooth cât și prin Wi-Fi.

Figura 1 oferă o imagine tehnică de ansamblu asupra arhitecturii aplicației noastre, concentrându-se mai mult pe maniera de funcționare a serverului. Săgețile indică direcția de interacțiune între submodule. Toate mesajele care vin sau sunt trimise la client trec printr-o fațadă de comunicare (*Connection Façade*), care compune mesajele (în cazul trimiterii) sau le interpretează (în cazul recepționării). Această fațadă primește informații legate de aplicațiile accesibile clientului (*Accessible applications*) și cele care rulează (*Running Applications*) pentru a-i trimite notificări, iar comenzile sunt transmise unui profil personalizat pentru fiecare tip de aplicație (*Profile*), care va ști ce acțiune să întreprindă pentru a realiza comanda respectivă. Aceste profile sunt create de către un *Profile Manager* atunci când modulul ce se ocupă cu monitorizarea proceselor ce rulează pe calculator îi indică faptul că a fost lansată în execuție o aplicație de tip Power Point sau de tip Windows Media Player.

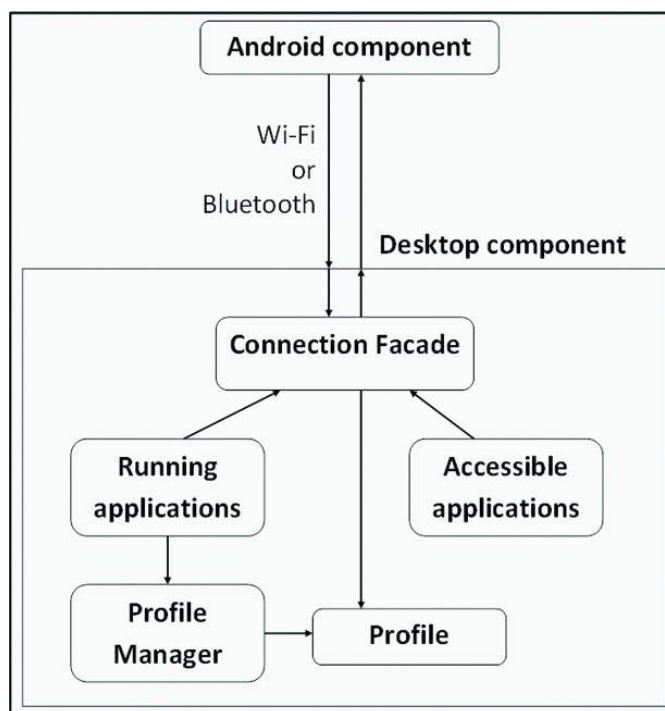


Figura 1. Arhitectura sistemului

## 2.2 Componenta Client

Am dezvoltat o aplicație Android care se poate conecta la orice calculator pe care rulează serverul. Clientul este construit urmând indicațiile și principiile de design pentru Android UI [1], asigurând o experiență plăcută și o interacțiune ergonomică între utilizator și dispozitivul mobil.

Aplicația Android constă din două ecrane, organizate în tab-uri pentru a oferi simplitate (vezi *Figura 2* de mai jos).

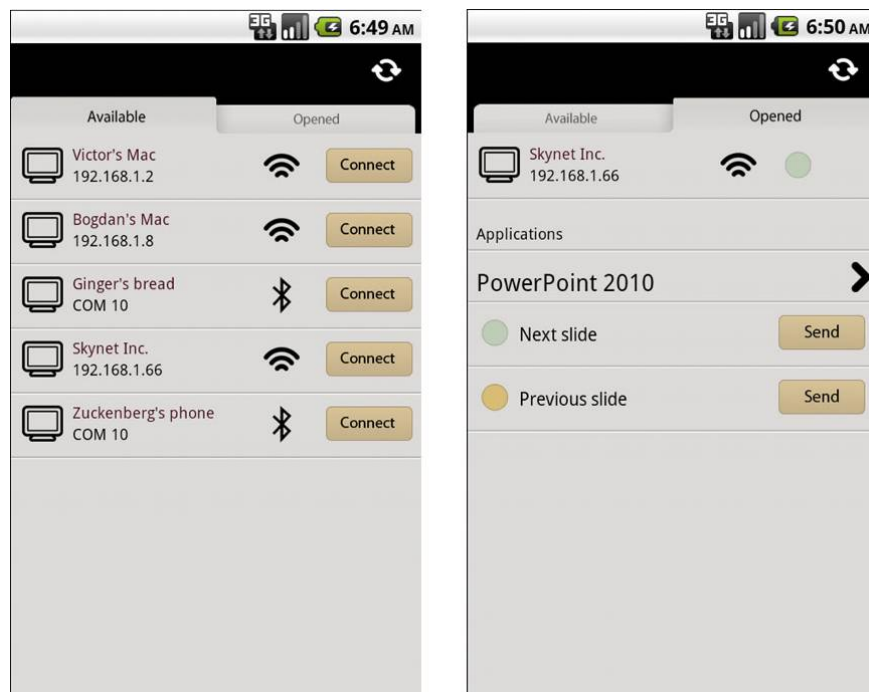


Figura 2. Ecranele aplicației Android

Primul ecran afișează o listă a dispozitivelor la care se poate conecta clientul, lucru care se poate realiza apăsând butonul de conectare (*Figura 2* în stânga). Mai mult, sunt oferite reprezentări grafice ale tipului de conexiune ce se poate realiza (Bluetooth sau Wi-Fi), alături de adresa IP și numele calculatorului. Odată conectat, al doilea ecran devine disponibil și oferă informații asupra aplicațiilor care pot fi controlate (*Figura 2* în dreapta). Este afișată o listă conținând toate comenzile ce pot fi trimise la

server prin apăsarea butoanelor corespunzătoare. Trecerea de la o aplicație la alta se poate realiza prin apăsarea unei săgeți din dreptul numelui aplicației curente.

În partea de sus, este disponibil un buton de reîmprospătare pentru ca, atunci când utilizatorul dorește, să se poată actualiza lista de dispozitive cu care se poate stabili o conexiune.

## 2.3 Componenta Server

Componenta desktop este formată din trei module principale: Accesible Applications, Running Applications, Profile Manager.

**Accesible Applications** conține o listă de aplicații ce pot fi accesate de programul nostru. Informațiile despre aplicațiile suportate sunt preluate dintr-un fișier de configurare *.xml* (vezi Figura 3).

```
<applications update="16 may 2011">
  <application>
    <id>1</id>
    <name>Microsoft Office PowerPoint</name>
    <path>C:\Program                               Files\Microsoft
Office\Office12\POWERPNT</path>
    <action>
      <name>View show</name>
      <shortcut>F5</shortcut>
      <codification>1004</codification>
    </action>
    <action>
      <name>Next slide</name>
      <shortcut>CTRL+Enter</shortcut>
      <codification>1006</codification>
    </action>
    ...
  </application>
  ...
</applications>
```

Figura 3. Structura fișierului de configurare *.xml*

**Running Applications** monitorizează procesele ce rulează pe un calculator. La intervale de timp regulate, serverul verifică dacă aplicațiile Windows Media Player și Microsoft Power Point se află în memorie prin intermediul Windows Management Instrumentation (<http://msdn.microsoft.com/en-us/library/windows/desktop/aa394582%28v=vs.85%29.aspx>). În momentul în care una dintre cele două aplicații își schimbă starea (este deschisă sau este închisă) este trimis un mesaj la dispozitivul Android pentru a-l înștiința de modificarea survenită. De asemenea, acest modul se ocupă și de pornirea/oprirea aplicațiilor precizate, atunci când comanda corespunzătoare este primită de la aplicația mobilă.

**Profile Manager** creează profile specifice pentru o aplicație, prin intermediul cărora poate realiza diferite acțiuni (e.g. *run*, *exit*, *close presentation*, *view show*, *exit show*, *next slide* și *previous slide* pentru aplicația PowerPoint). Putem reprezenta orice tip de aplicație pe care sistemul o suportă. Asignându-i un anumit profil, creăm un „bridge” și, astfel, orice comandă primită de aplicație ajunge în profil, iar acesta știe exact ce acțiune să întreprindă. Acest modul a fost scris în C# și este programat să pornească odată cu calculatorul. Aplicația rulează în background, putând fi accesată de către utilizator din system tray și, odată pornită, așteaptă ca un telefon să se conecteze prin Wi-Fi sau Bluetooth. La un moment dat, doar un singur dispozitiv mobil poate fi conectat la această aplicație. Când dispozitivul Android este deconectat, programul desktop începe să aștepte noi cereri de conectare.

Componenta server primește comenzi de la componenta Android și, astfel, putem controla aplicații precum Microsoft Power Point și Windows Media Player. Lista aplicațiilor accesibile și acțiunile suportate de către fiecare dintre acestea este stabilită la rulare, prin citirea informațiilor aferente din fișierul de configurare *.xml*. Solicitarea execuției unei acțiuni înseamnă recepționarea unui șir de caractere codificat de la client. Validitatea este verificată prin utilizarea expresiilor regulate. În cazul în care comanda nu se potrivește cu formatul existent, este afișat un mesaj de eroare. Dacă mesajul trece de verificare, dar nu este găsit în lista de comenzi disponibile, cererea este ignorată și este afișat un mesaj de eroare specific. Dacă niciuna dintre aceste situații nu are loc, atunci comanda este corectă și acțiunea corespunzătoare este executată.

Pentru controlul aplicațiilor am folosit două abordări diferite. *Prima abordare*, a fost folosită în cazul aplicațiilor Microsoft Power Point, unde am ales ca modalitatea de control să se facă prin intermediul obiectului aferent aplicației active aflate în memorie (sau prin crearea acestui obiect). Aplicația noastră poate identifica și acțiunile inițiate de un utilizator direct pe calculator (e.g. deschiderea manuală a unei prezentări). Pentru a realiza acest lucru, folosim biblioteca Microsoft.Office.Interop [4], oferită de platforma .NET. Astfel, dispunem de control total asupra aplicației putând întreprinde acțiuni dintre cele mai diverse. Această variantă oferă o flexibilitate foarte mare și o gamă variată de întrebuițări, dar dezavantajul este că, pentru fiecare comandă ce se dorește a fi realizată, trebuie scris cod explicit.

*A doua abordare* este folosirea scurtăturilor de tastatură, în cazul aplicației Windows Media Player. Aceasta este mult mai eficientă și necesită doar scrierea combinației de taste specifice. Totuși, deși eficientă, această soluție nu este flexibilă, acțiunile ce pot fi întreprinse fiind limitate la scurtăturile definite deja pentru aplicația respectivă.

Pentru o implementare eficientă și cât mai flexibilă a componentei server, am folosit diverse șabloane de proiectare (Gamma et al., 1995). Unul dintre acestea este *Abstract Factory* [2], care ne ajută la crearea de profile specializate pentru fiecare aplicație în parte. Din acest motiv, este foarte ușor de adăugat funcționalități pentru aplicații noi, nefiind nevoie decât înregistrarea acestora în fișierul de configurare *.xml* și implementarea unui profil specific prin extinderea clasei abstracte *Profile*. La acea fațadă de comunicare despre care am amintit mai sus, în contextul prezentării arhitecturii sistemului, am folosit șablonul *Façade* [2].

Interpretarea mesajelor primite de la client și compunerea celor trimise către acesta se realizează în cadrul unei clase specializate, care cunoaște specificul limbajului folosit de cele două entități care comunică, implementată folosind șablonul *Interpreter* [2]. Mai amintim *Singleton* [2], care ne asigură că există câte o singură instanță a claselor care se ocupă de monitorizarea proceselor, administrarea aplicațiilor accesibile, respectiv crearea profilelor asociate acestor aplicații.



## 2.4 Comunicarea

Modulul pentru comunicarea server-client este o componentă foarte importantă și oferă posibilitatea trimiterii și primirii de mesaje între cele două entități, aceste mesaje fiind reprezentate de șiruri de caractere cu un format prestabilit. Comunicarea poate fi realizată în două moduri, în funcție de preferințele utilizatorului: folosind Bluetooth (<http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>) sau folosind Wi-Fi (<http://en.wikipedia.org/wiki/Wi-Fi>).

Oricum, aplicația oferă aceleași facilități indiferent de metoda aleasă pentru comunicare.

### Comunicarea folosind Bluetooth

Prima modalitate de comunicare ce poate fi realizată se folosește de conexiunea Bluetooth. Acest modul reprezintă o alternativă pentru trimiterea mesajelor și comenzilor pentru aplicația pe care am construit-o. Este foarte important să avem o altă metodă de comunicare acolo unde nu există acces la Internet sau când utilizatorul nu dispune de conexiune Wi-Fi.

### Identificarea dispozitivelor și stabilirea conexiunii

Pentru a crea o conexiune între cele două dispozitive, serverul trebuie să verifice dacă telefonul Android are un adaptor Bluetooth încorporat. Dacă adaptorul este pornit, utilizatorul poate căuta componente desktop la care să se conecteze. Pentru a realiza acest lucru, trebuie să introducă codul de securitate dat de PC într-un timp limită. Dacă nu se stabilește o conexiune în acest timp, codul este resetat și se poate încerca din nou.

### Trimiterea de mesaje și comenzi

Dacă s-a stabilit cu succes o conexiune, funcțiile de *read* și *write* pot fi folosite pentru a trimite mesaje. Pentru a evita blocarea interfeței grafice, funcția de citire rulează într-un fir de execuție separat, fiind întreruptă atunci când conexiunea prin socket aruncă o excepție. Acest lucru face ca serverul să termine bucla de ascultare, oferind posibilitatea de conectare și altor dispozitive mobile. Un singur client poate fi conectat la un moment dat la server, făcând aplicația mai sigură.

### Comunicarea folosind Wi-Fi

A doua modalitate de comunicare se realizează prin intermediul conexiunii wireless. Modulul „Wireless Connectivity” oferă două funcționalități celorlalte module:

1. *Clientul* va avea posibilitatea de a identifica calculatoarele din apropiere, oferind posibilitatea conectării la acestea;
2. *Clientul* și *serverul* pot trimite și primi mesaje după ce conexiunea dintre ei a fost stabilită.

Pentru a putea realiza descoperirea dispozitivelor, serverul trebuie să trimită un mesaj care conține adresa IP a acestuia. Orice client care se află în raza de acțiune a serverului va aștepta mesajul și va reține o listă cu toate serverele disponibile. Este stabilită o perioadă de așteptare, iar după ce aceasta s-a terminat, dacă serverul nu mai este identificat, nu va mai fi vizibil nici în lista oferită clientului.

A doua proprietate este implementată folosind o clasă „wrapper” pentru clasele Socket și ServerSocket din limbajul Java. Ele oferă funcționalitățile necesare pentru a trimite și primi date, cât și pentru adresa IP și starea conexiunii. Acest lucru permite serverului și clientului să nu interacționeze direct cu conceptele de nivel jos.

Chiar dacă serverul este implementat în C#, modulul conexiunii wireless este implementat în Java. Folosind IKVM [3], am reușit să transformăm fișierul *.jar* într-un fișier *.dll* valid care poate fi folosit de către server pentru a transmite și primi diversele mesaje de la client.

### 3. Evaluarea soluției

Spre deosebire de celelalte aplicații similare disponibile pe Internet, ne-am decis să ne concentrăm pe a realiza un produs ușor de utilizat. Deși o întreagă gamă de instrucțiuni sunt disponibile, acestea pot fi personalizate prin intermediul fișierului de configurare *.xml*, clientul Android având acces doar la cele existente în fișier.

Alternarea aplicațiilor poate fi realizată prin simpla apăsare a unui buton de pe ecran. Acest lucru este foarte util dacă se dorește combinarea unor fișiere video cu prezentarea existentă, tranziția dintre ele fiind atât rapidă, cât și ușor de manevrat, ceea ce oferă o facilitare suplimentară utilizatorului.

### 3.1 Studii de caz și teste

În scopul evaluării soluției noastre și al monitorizării comportamentului ei în diverse situații, am realizat o serie de teste și studii de caz.

#### Utilizarea aplicației timp îndelungat

Pe parcursul mai multor teste, am utilizat programul timp de aproximativ 40 de minute, susținând o prezentare Power Point, și nu am întâmpinat probleme majore. Aplicația s-a comportat cum era de așteptat, permițându-ne să ne deplasăm înainte și înapoi prin documentul Power Point. Legătura dintre client și server, atât prin intermediul conexiunii Bluetooth, cât și a conexiunii Wi-Fi, a putut fi deschisă la orice moment și nu au existat situații neprevăzute, cum ar fi întreruperea bruscă sau probleme de conectare, atât pe partea clientului, cât și a serverului.

#### Variația distanței

În cazul acestor teste am variat în timpul prezentărilor distanța dintre calculator și telefonul mobil pe care aveam aplicația Android. În timp ce distanța dintre telefon și calculator nu este o problemă atunci când este implicată conexiunea Wi-Fi, pentru Bluetooth aceasta reprezintă un element important, de care trebuie să ținem cont. Pentru a vedea care este aria de acoperire, am testat pe o distanță cuprinsă între 0 și 25 de metri. Am ajuns la concluzia că aplicația rulează fără dificultăți pe o distanță de maxim 20 de metri, peste acest prag existând posibilitatea deconectării (cam în 60% din cazurile testate conexiunea s-a întrerupt).

#### Obstacole

De această dată, am luat în considerare situațiile când între client și server există diverse obiecte (am considerat chiar și situația când cele două se află în camere diferite, despărțite printr-un perete) și am verificat cum influențează acestea comunicarea dintre ele. □i în cazul acestor scenarii, legătura prin Wi-Fi se realizează cu succes. Atunci când conexiunea se realizează prin Bluetooth, am observat următoarele: atât timp cât distanța nu depășește 20 de metri (care am dedus anterior că ar fi pragul de acceptare), programul funcționează fără probleme indiferent de micile obstacolele care se interpun. Acest lucru este valabil și atunci când se încearcă stabilirea unei

conexiuni din două camere diferite despărțite de un perete. Însă, la interpunerea a doi sau mai mulți pereți, conexiunea nu mai poate fi stabilă.

### **Teste de stres**

Cu scopul de a detecta cât de multe instrucțiuni pot fi executate într-un timp redus, am realizat o testare manuală, cu ajutorul unui client ce apăsă foarte rapid butoanele din aplicația Android. Totodată, am creat un fișier jurnal cu fiecare instrucțiune executată de server alături de momentul când a fost apelată și timpul necesar executării ei. În urma analizării fișierului jurnal am observat că fiecare apăsare detectată de ecran a fost trimisă către server și fiecare comandă corespunzătoare acesteia a fost îndeplinită cu succes. Într-o secundă am efectuat 7-8 instrucțiuni, un număr mai mare neputând fi posibil, întrucât testarea a fost realizată manual. Cu toate acestea, nu credem că acest tip de situație ar reprezenta o problemă pentru server, singura piedică putând fi constituită de ecranul telefonului în sine, în cazul în care nu ar detecta fiecare atingere. Acest lucru are loc atunci când frecvența mișcărilor este mai mare de 10 instrucțiuni pe secundă. Întrucât niciun utilizator nu ar fi capabil să execute un flux de comenzi atât de rapid, considerăm că aplicația funcționează corect în acest tip de situații.

### **Versiunile software**

Programul a fost testat pe versiunea 12 a aplicației Windows Media Player. În ceea ce privește Microsoft Power Point, aplicația a fost folosită pe versiunile 2003, 2007 și 2010 ale pachetului Microsoft Office. Trebuie să precizăm faptul că pe durata acestor teste nu am întâmpinat nicio problemă, nefiind necesare ajustări pentru niciuna dintre aceste versiuni.

## **3.2 Avantaje**

În această secțiune vrem să enumerăm avantajele folosirii aplicației pe care am realizat-o, urmând ca în secțiunea următoare să prezentăm dezavantajele. Avantajele identificate de cei care au folosit aplicația noastră au fost:

- Soluție utilă pentru a accesa computerul de la distanță și pentru a executa comenzi pe acesta prin intermediul telefonului mobil;
- Rapidă și ușor de folosit, poate înlocui cu succes o telecomanda clasică pentru controlul la distanță a prezentărilor;

- Poate fi extinsă ușor pentru a controla și alte aplicații ce funcționează pe sistemul de operare Windows, în acest caz având loc o creștere a versatilității și a cazurilor de utilizare;
- Este o soluție de încredere în ceea ce privește stabilirea unei conexiuni și recepționarea de comenzi.

### 3.3 Dezavantaje

Dezavantajele identificate au fost:

- Securitatea nu constituie un punct forte, neexistând o componentă ce verifică identitatea utilizatorului;
- Serverul poate rula doar pe calculatoarele pe care rulează Microsoft Windows, alte sisteme de operare nefiind suportate;
- Un alt dezavantaj este datorat ariei limitate de programe pe care aplicația le poate accesa pentru moment.

## 4. Concluzii și direcții viitoare de dezvoltare

În acest articol am prezentat modul în care am creat o aplicație robustă, eficientă și ușor de utilizat care poate fi folosită pentru a controla de la distanță programele Windows Media Player și Microsoft Power Point.

Aplicația are două componente principale, o componentă server și o componentă client, care comunică între ele prin intermediul conexiunilor Bluetooth sau Wi-Fi. Testele pe care le-am realizat ne-au demonstrat robustețea aplicației la utilizarea îndelungată, la testele de stres sau la situațiile în care interpunem obiecte între client și server.

Pentru viitor, modalitatea de controlare a aplicațiilor prin utilizarea scurtăturilor (cum se procedează în cazul aplicației Windows Media Player) poate fi folosită cu succes pentru orice aplicație ce dispune de acest mod de control rapid. Serverul ar putea dispune de un panou de control, o fereastră în care utilizatorul ar putea selecta aplicația pe care dorește să o controleze sau să precizeze numele acesteia. De asemenea, dorim ca pe viitor să oferim posibilitatea utilizatorilor să adauge și să editeze scurtăturile ce se doresc a fi folosite, dându-le un nume cât mai sugestiv. Denumirea va fi vizibilă pe ecranul dispozitivului mobil pe care rulează aplicația Android, iar toate informațiile vor fi salvate în fișierul de configurare *.xml*. Această abordare

ar oferi flexibilitate, scurtăturile memorate putând fi modificate ușor în cazul în care combinația de taste folosită se modifică.

Prin această abordare, aria aplicațiilor suportate poate ajunge să conțină aplicații din orice domeniu. Spre exemplu, am putea controla Skype, care să ne permită să realizăm convorbiri telefonice prin intermediul internetului. Am putea configura serverul nostru, astfel încât utilizatorul să poată realiza o convorbire telefonică, să o pună în așteptare, să întrerupă sunetul sau să o încheie. În același mod, ar putea fi controlate și navigatoarele web și am putea defini acțiuni de modificare a poziției paginii curente, sau de zoom in/out.

În ceea ce privește clientul, dorim ca pe viitor aplicația mobilă să dispună și de un buton de stingere a laptopului pe care rulează serverul. Această facilitate ar fi folositoare la finalul susținerii unei prezentări. Prin apăsarea acestui buton, laptopul s-ar închide, iar utilizatorul ar putea evita timpii morți rezultați în urma utilizării metodei clasice de închidere.

Legat de server, dorim ca pe viitor acesta să fie portabil pe orice platformă. Până în prezent, aplicațiile pe care le-am avut în vedere sunt specifice sistemului de operare Microsoft Windows, astfel încât o direcție importantă pentru dezvoltarea produsului ar fi să considerăm aplicații care sunt independente de sistemul de operare (e.g. Open Office). Acest lucru ar face să crească numărul de situații în care aplicația noastră ar putea fi folosită.

O altă idee care vine tot în sprijinul portabilității ar fi realizarea unor versiuni diferite ale serverului pentru fiecare tip de sistem de operare (Windows, Mac sau Linux). Fiecare dintre aceste versiuni ar oferi posibilitatea controlării aplicațiilor specifice sistemului de operare aferent. Crearea lor nu ar constitui o problemă majoră, întrucât structura de bază a serverului rămâne aceeași, modificările apărând doar în cadrul modulului care se ocupă de monitorizarea proceselor de pe mașina gazdă (aceasta fiind specifică fiecărui sistem de operare în parte). De asemenea, este necesară crearea unor profile specializate pentru fiecare aplicație suportată, extinzând clasa abstractă *Profile*.

Alte direcții de care vom ține cont în viitor ar putea fi:

- Adăugarea de elemente care să ne asigure securitatea soluției (e.g. autentificarea și verificarea identității), dând aplicației o notă de

siguranță atunci când este vorba de prezentări oficiale și nu se dorește ca oricine să poată avea acces la aplicația desktop.

- Posibilitatea de a conecta simultan mai multe telefoane Android la server. Acest lucru ar putea fi posibil, dacă modul de gestionare al cererilor ar putea fi ajustat, pentru a nu exista conflicte între ele.
- Studiarea posibilității de a avea acces la toate aplicațiile deschise și oferirea utilizatorului a unei modalități de a controla aplicația dorită. Această cerere vine în contextul în care, chiar dacă serverul poate gestiona mai mult de o prezentare Power Point în același timp, această funcționalitate nu este implementată și la nivel de client. Momentan, comanda pe care un client o trimite către aplicația desktop este direcționată către prezentarea Power Point activă.

## Mulțumiri

Mulțumim colegilor de la Facultatea de Informatică care ne-au ajutat la construirea anumitor componente ale sistemului.

Cercetarea prezentată în această lucrare a fost finanțată de către Programul Operațional Sectorial Dezvoltarea Resurselor Umane prin proiectul „Dezvoltarea capacității de inovare și creșterea impactului cercetării prin programe post-doctorale POSDRU/89/1.5/S/49944”.

## Referințe

- [1] Android User Interface Guidelines. [http://developer.android.com/guide/practices/ui\\_guidelines/index.html](http://developer.android.com/guide/practices/ui_guidelines/index.html), 2012.
- [2] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Co., 1995.
- [3] IKVM.NET. Implementation of Java for Mono and the Microsoft .NET Framework, <http://www.ikvm.net>, 2011.
- [4] Microsoft.Office.Interop.PowerPoint Namespace, 2012, <http://msdn.microsoft.com/en-us/library/microsoft.office.interop.powerpoint.aspx>
- [5] PPT Remote. *A free Android remote control for Power Point*, <http://www.pptremotecontrol.com>, 2012.
- [6] *Remote Presenter for Android*, <http://www.free-power-point-templates.com/articles/remote-presenter-for-android>, 2012.
- [7] Tango Remote Control Media Player HD, [http://www.appsmenow.com/app\\_page/16978-Tango\\_Remote\\_Control\\_Media\\_Player\\_HD](http://www.appsmenow.com/app_page/16978-Tango_Remote_Control_Media_Player_HD), 2012.

- [8] Vectir. Remote control your computer from your phone, [www.vectir.com](http://www.vectir.com), 2012.
- [9] WIN – Remote Service, [http://www.banamalon.net/android/wiki/index.php?title=WIN\\_-\\_Remote](http://www.banamalon.net/android/wiki/index.php?title=WIN_-_Remote), 2012.
- [10] Xtralagic Remote Desktop Client for Android, <http://www.xtralagic.com/rdpclient.shtml>, 2012.