

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL AUTOMATICĂ

# Mediu de învățare 3D

Lucrare de disertație

Absolvent  
Mureșan Cătălin-Gabriel

Iulie, 2014



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE

DECAN  
Prof.Dr.Ing. Liviu MICLEA

DIRECTOR DEPARTAMENT  
Prof.Dr.Ing. Rodica POTOLEA

# Mediu de învățare 3D

## Lucrare de disertație

1. Absolvent: Mureșan Cătălin-Gabriel
2. Coordonator științific (1): Conf.dr.ing. Mihai DAMIAN
3. Coordonator științific (2): Prof.Dr.Ing. Liviu MICLEA
4. Conținutul lucrării: Pagina de prezentare, aprecierile coordonatorului, titlul capitolului 1, titlul capitolului 2, ..., titlul capitolului n, bibliografie, anexe, CD.
5. Locul documentării: UTCN, Cluj-Napoca
6. Consultanți: .....
7. Data emiterii temei: .....
8. Data predării: .....

Semnătură coordonator  
Conf.dr.ing. Mihai DAMIAN  
Prof.Dr.Ing. Liviu MICLEA

Semnătură absolvent  
Mureșan Cătălin-Gabriel

Iulie, 2014



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE

**Declarație pe proprie răspundere privind  
autenticitatea lucrării de disertație**

Subsemnatul *Mureșan Cătălin-Gabriel*, legitimat cu *CI* seria *KT* numărul *771623*, *CNP 1820513013911*, autorul lucrării *Mediu de învățare 3D* elaborată în vederea susținerii examenului de finalizare a studiilor de masterat la Facultatea de Automatică și Calculatoare, Departamentul Automatica, Specializarea *Informatică Aplicată* din cadrul Universității Tehnice din Cluj-Napoca, sesiunea *Iulie* a anului universitar *2013/2014*, declar pe proprie răspundere, că această lucrare este rezultatul propriei mele activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate în textul lucrării și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență sau disertație.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de disertație*.

Cluj-Napoca  
data

PRENUME NUME  
Semnătură absolvent



## Rezumat

Progresul rapid în domeniul I.T.& C și scăderea prețurilor componentelor hardware performante a făcut fezabilă aplicarea metodei de învățare în medii virtuale 3D în oricare treaptă a sistemului educațional. Într-un mediu virtual tridimensional, obiectele studiate sunt reprezentate prin coordonate ce descriu forma și poziționarea lor în spațiu, apropiind astfel modul de reprezentare de cel al obiectelor din lumea reală. Utilizatorii se pot poziționa în orice punct al spațiului virtual, fapt care le va permite să studieze obiectele din orice unghi, lucru dificil de realizat cu programele informatice cu redare 2D. De asemenea o parte dintre obiectele virtuale pot fi programate să răspundă la acțiunea utilizatorului, conducând la sporirea gradului de implicare a utilizatorului, cu rezultate benefice în procesul de învățare.

Foarte importantă este siguranța și costul redus. Utilizatorii au șansa de a efectua activități care ar fi altfel foarte costisitoare pentru instituțiile implicate în procesul de educare, sau în medii cu grad mare de risc. În mediile de învățare 3D se pot repeta în siguranță proceduri care nu sunt tolerante la erori, precum operațiile chirurgicale sau controlul proceselor într-o centrala nucleară.

În această lucrare sunt identificate elementele necesare pentru implementarea unui program informatic reprezentând un asemenea mediu de învățare 3D, care să ruleze în sistemul de operare Linux, și se va face descriere a arhitecturii aplicației. Această lucrare va conține și codul sursă cu comentarii pentru a putea fi folosită ca exemplu pentru programatorii și designerii de medii virtuale aplicate în educație.





# Cuprins

<b>Listă de tabele</b>	<b>iii</b>
<b>Listă de figuri</b>	<b>iv</b>
<b>Capitolul 1 Introducere</b>	<b>1</b>
1.1 Introducere . . . . .	1
1.2 Problema adresată . . . . .	1
1.3 Motivație . . . . .	1
1.4 Descrierea studiului pe capitole . . . . .	2
1.5 Învățarea și mediile 3D. Concepte relevante. . . . .	2
<b>Capitolul 2 Obiectivele cercetării</b>	<b>4</b>
2.1 Introducere . . . . .	4
2.2 Obiective . . . . .	4
2.3 Argumente . . . . .	5
2.4 Mini studiu empiric privind raportul aplicațiilor web - aplicații clasice cu destinație educativă . . . . .	6
<b>Capitolul 3 Stadiul actual în domeniu</b>	<b>8</b>
3.1 Introducere . . . . .	8
3.2 Abordări similare . . . . .	8
3.3 Second Life . . . . .	8
3.4 Uther Academy . . . . .	9
3.5 iSocial . . . . .	10
3.6 Tehnici/Tehnologii folosite . . . . .	11
<b>Capitolul 4 Planul aplicației</b>	<b>12</b>
4.1 Introducere . . . . .	12
4.2 Funcțiile sistemului . . . . .	12
4.2.1 Minimul de funcționalitate de cercetat și implementat . . . . .	12
4.2.2 Funcționalitate extinsă . . . . .	13
4.3 Decizie privind design-ul sistemului . . . . .	13

4.4	Test . . . . .	14
4.5	Argumentarea deciziei pe baza rezultatelor testării . . . . .	15
<b>Capitolul 5 Design-ul aplicației server</b>		<b>16</b>
5.1	Introducere . . . . .	16
5.2	Funcțiile aplicației server . . . . .	16
5.2.1	Funcții de bază ale serverului . . . . .	16
5.2.2	Aspecte tehnice și principii generale de design . . . . .	17
5.3	Mecanism pentru obținerea scalabilității aplicației server . . . . .	18
5.3.1	Interfața PLUG-IN-ului . . . . .	18
5.3.2	Tipuri de plug-in-uri dezvoltate pentru serverul mediului de învățare 3D . . . . .	19
5.3.3	Descrierea mecanismului de încărcare a plug-in-urilor . . . . .	20
5.4	Modul de reprezentare a datelor . . . . .	22
5.4.1	Scurtă introducere în XML (CITE ...) . . . . .	22
5.4.2	Structura XML a datelor transmise dinspre server spre aplicația client	23
5.5	Schema serverului. Diagrama claselor . . . . .	24
5.5.1	Modelarea comenzilor primite de server . . . . .	24
<b>Capitolul 6 Conluzii</b>		<b>26</b>
6.1	Conținut . . . . .	26
6.2	Detalii tehnice . . . . .	26
6.2.1	Dimensiune . . . . .	26
<b>Anexa A Testarea fiabilității modelului CLIENT-SERVER</b>		<b>27</b>
A.1	Script . . . . .	27
A.2	Server de test . . . . .	28
A.2.1	Rezultate . . . . .	31

# Listă de tabele

2.1	Rezultate . . . . .	7
3.1	Tehnologii utilizabile pentru dezvoltarea de medii 3D sub Linux . . . . .	11

# Listă de figuri

2.1	Căutare - tehnologii web . . . . .	6
2.2	Căutare - tehnologii 3D și c++ . . . . .	6
3.1	Eveniment în Second Life . . . . .	9
3.2	U.A. - aulă virtuală . . . . .	10
3.3	iSocial - panou informativ . . . . .	10
4.1	Modelul P2P . . . . .	14
4.2	Modelul Client-Server . . . . .	14
5.1	Arhitectura Plug-in . . . . .	18
5.2	Schema Plug-in Hibrid . . . . .	19
5.3	Structură arborescentă de date descrisă cu XML . . . . .	23
5.4	UML - COMENZI . . . . .	25

# Capitolul 1

## Introducere

### 1.1 Introducere

Acest capitol definește în secțiunea 1.2 problema adresată, fiind subliniați principalii piloni ai problemei: costul serviciilor educaționale și în special complexitatea și costul ridicat al soluțiilor existente ce implică tehnologia mediilor de învățare 3D. Secțiunea 1.3 este un enunț al soluției ce se dorește a fi oferită în această lucrare. Secțiunea 1.4 subliniază etapele studiului iar secțiunea 1.5 tratează în linii mari teorii referitoare la mediile de învățare 3D și la învățare în general, teorii ce vor avea influență asupra produsului final (aplicația informatică).

### 1.2 Problema adresată

Asigurarea unui învățământ de calitate poate implica costuri semnificative pentru diversele instituții sau organizații care oferă asemenea servicii, în special când mediul optim de învățare presupune colaborarea, întreprinderea de experimente multiple sau lucrul în medii cu risc ridicat. De asemenea, în unele cazuri, instruirea poate avea loc în medii simulate al căror cost de construire și folosire ar fi mult prea mare.

### 1.3 Motivație

Ca răspuns la complexitatea și costul soluțiilor existente, se dorește realizarea unei aplicații reprezentând o un mediu virtual 3D; o aplicație de dimensiuni reduse dar care poate fi cu ușurință extinsă și completată cu noi elemente reprezentând: experimente de laborator virtual, diverse forme de reprezentare în forma grafică 3D a cunoștințelor, spații virtuale pentru desfășurarea de activități educative (muzee, săli de conferință virtuală) etc. O asemenea aplicație va avea se va dezvolta în două module: client și server, ambele dinamice, ușor de extins și care să se constituie o mini platformă pe care alți programatori

să poată construi cu minim efort lumi virtuale orientate spre livrarea conținutului educativ într-o formă cât mai atragătoare pentru utilizatori.

Aplicația informatică va rula sub GNU/Linux, pentru implementarea acesteia se vor folosi doar unelte și biblioteci software dezvoltate de comunitățile free software / open source. Codul sursă va fi eliberat cu o licență liberă.

## 1.4 Descrierea studiului pe capitole

În **Capitolul 1** (capitolul curent) se pune accentul pe descrierea nevoii instituțiilor furnizoare de servicii educațional de unelte moderne pentru îndeplinirea obiectivelor lor specifice și pe lipsa de alternative care să îndeplinească **simultan** următoarele aspecte: libertatea codului (open source / free software), dimensiune redusă și modularitate ridicată, costuri minime. Se afirmă obiectivul creării unui program liber (free/open source) ușor de menținut și extins reprezentând o lume virtuală 3D. Definițiile și conceptele teoriei învățării în medii virtuale încheie capitolul 1.

În **Capitolul 2** sunt enumerate obiectivele cercetării.

În **Capitolul 3** sunt prezentate specificațiile generale ale lucrării de cercetare, atât obiectivele minimale care vor trebui atinse până la definitivarea studiului cât și obiectivele potențiale de atins în cazul extinderii lucrării.

În **Capitolul 4** se analizează sistemul în întregime prin descrierea ansamblului *client-server*. Se decide design-ul sistemului și se argumentează alegerea făcută prin testele anterior stabilite.

În **Capitolul 5** se realizează un studiu amănunțit al aplicației *server* din sistem. Se decide design-ul aplicației, se efectuează teste și se decid aspectele tehnice referitoare la implementarea aplicației: limbajul folosit pentru implementare, librării software folosite, algoritmi și metode de implementare.

În **Capitolul 6** se realizează un studiu amănunțit al aplicației *client* din sistem. Se decide design-ul aplicației, se efectuează teste și se decid aspectele tehnice referitoare la implementarea aplicației: limbajul folosit pentru implementare, librării software folosite, algoritmi și metode de implementare.

În **Capitolul 7** sunt prezentate concluziile.

## 1.5 Învățarea și mediile 3D. Concepte relevante.

În mare măsură, soluția tuturor acestor probleme se găsește în mediile de învățare virtuale cu redare 3D, datorită unor caracteristici care le califică ca și cadru (în unele cazuri ideal) de învățare. Unele dintre cele mai importante caracteristici ale mediilor de învățare cu redare tridimensională sunt: capacitatea de a simula orice spațiu fizic, de a intermedia interacțiunea dintre diverse persoane aflate în zone geografice diferite și oferta de unelte

de observare și măsurare a performanțelor sau a progresului participanților la procesul de învățare și faptul ca orice resursă virtuală poate fi refolosită fără costuri suplimentare.

O analiză a autorilor Wann și Mon Williams oferă o descriere a mediilor tridimensionale ca medii ce "valorifică aspectele naturale ale percepției umane prin extinderea informațiilor vizuale în trei dimensiuni spațiale și care poate suplimenta aceasta informație cu alți stimuli și modificari temporale"[?] și care "permit interacțiunea utilizatorului cu obiectele redată"[?]. Se pot astfel deduce trei elemente care disting mediile de învățare 3D de alte medii de învățare virtuale. Mai detaliat, mediile virtuale tridimensionale sunt medii grafice ce crează impresia de spațiu 3D, în care utilizatorul controlează caractere generate de computer (avatare), caractere care îi reprezintă în timp ce interacționează cu mediul sau cu alți utilizatori. Acestea pot contribui la sistemul educațional prin facilitarea colaborării, comunicării și experimentării. Prin oferirea unui surogat al realității, mediile de învățare pot crea percepția de existență a utilizatorului în mediul simulat.

În psihologie și educație, învățarea este definită ca fiind procesul care aduce împreună experiența cognitivă, emoțională și influența de mediu, pentru acumularea, îmbunătățirea sau schimbarea cunoștințelor, abilităților sau a concepției despre lume a unui individ.[C02]

Clasificarea metodelor de învățare, stabilită de Frederic Vester[c03] : învățarea auditivă, învățarea vizuală, învățarea tactilă, învățarea cognitivă ( prin intelect ). Prin această metodă de clasificare F. Vester, neagă efortul intelectual pentru primele trei tipuri, acest efort fiind atribuit învățării cognitive. Deși nu poate fi în totalitate adevărat, fiecare dintre noi am putut experimenta reducerea 'consumului' intelectual atunci când am învățat folosindu-ne de materiale didactice cu vizuale sau auditive. Personele de toate vârstele învață cel mai bine atunci când sunt implicate în experiențe semnificative. Învățarea are loc atunci când mintea este capabilă să pună la un loc informațiile primite de la toate simțurile și să le coreleze cu experiențele trecute. Prin folosirea mai multor simțuri pentru a învăța se poate da mai mult sens procesului de acumulare. Copii în mod natural învață folosindu-se de toate simțurile în cel mai eficient mod posibil.

Valoarea unui mediu 3D bine construit constă în faptul ca poate antrena simțul vizual al utilizatorului, cu îmbunătățirea rezultatelor.

# Capitolul 2

## Obiectivele cercetării

### 2.1 Introducere

În secțiunea 2.2 se definește obiectivul lucrării și se enumeră etapele necesare îndeplinirii obiectivului. Motivația alegerii acestui obiectiv este prezentată, în secțiunea 2.3, cu argumente de ordin estetic referitoare la calitatea prezentării materialelor educative în redare tridimensională cât și cu argumente practice referitoare la utilitatea aplicației software rezultate. Dorința autorului de a dezvolta o aplicație din categoria free software / open source este evidentă și este în sine un argument.

### 2.2 Obiective

Pornind de la observația empirică privind existența unui număr mult mai mare de aplicațiilor educative bazate pe tehnologii web comparativ cu numărul aplicațiilor native bazate pe tehnologie ce implică grafică 3D (în special aplicații non proprietare pentru Linux), și luând în calcul conceptul înrădăcinat în rândul programatorilor privind gradul de dificultate redus pentru realizarea de aplicații web comparativ cu aplicațiile "standalone", **se dorește prin această lucrare a se demonstra faptul că se poate dezvolta o aplicație non-web care să faciliteze publicarea de materiale educative aproape la fel de ușor ca în cazul folosirii tehnologiilor web dar de o calitate mai ridicată și într-o formă mult mai atractivă.**

În acest sens se vor urmări:

- Identificarea unei distribuții Linux care să înlesnească instalarea componentelor necesare pentru dezvoltarea aplicației.
- Identificarea librăriilor software necesare pentru comunicarea în rețele web, grafică 3D, interfețe grafice etc.



- Crearea unei platforme software care să preia în mare măsură complexitatea obișnuită în cazul aplicațiilor cu grafică 3D.
- Crearea unei librării software care să permită altor programatori să extindă funcționalitatea aplicației (sistemului client-server).
- Elaborarea documentatiei minime referitoare la instalare mediului de dezvoltarea a aplicațiilor.
- Dezvoltarea unui site web informativ pentru mediul de învățare 3D obținut.
- Realizarea unei mini-studii comparative pentru aplicația dezvoltată și soluțiile existente (inclusiv soluțiile web)

## 2.3 Argumente

Într-o mică se tratează problema practică a reducerii decalajului de proliferare dintre mediile de învățare bazate pe tehnologii web prin realizarea unei aplicații software cât mai flexibile care să permită crearea și publicarea de conținut educativ interactiv într-un mediu 3D adecvat, de către persoane ce posedă cunoștințe minime de programare. Astfel, se poate echilibra proporția acestor sisteme în totalul produselor informatice destinate învățării și se pot valorifica progresele recente din domeniul hardware.

**Argumentele generale** în favoarea metodei de învățare în medii 3D sunt : modul de reprezentare a obiectelor studiate și apropierea lor de obiectele din lumea reală prin formă; faptul că utilizatorii se pot poziționa în orice punct al spațiului virtual, fapt care îi permite utilizatorului să studieze obiectele din orice unghi, lucru greu de realizat cu materialele didactice tradiționale sau cu programele informatice cu redare 2D; faptul că obiectele virtuale pot fi programate să răspundă la acțiunea utilizatorului, fapt ce poate conduce la sporirea gradului de implicare a utilizatorului, cu rezultate benefice în procesul de învățare; siguranța și costul redus, utilizatorii având șansa de a efectua activități care ar fi altfel foarte costisitoare pentru instituțiile implicate în procesul de educare, sau în medii cu grad mare de risc. În mediile de învățare 3D se pot repeta în siguranță proceduri care nu sunt tolerante la erori, precum operațiile chirurgicale sau controlul proceselor într-o centrală nucleară. De asemenea, un argument puternic în favoarea sistemelor informatice de învățare este eliminarea necesității prezenței studentului într-o clasă sau în o anumită zonă geografică.

**Argumentele speciale** ale acestui studiu sunt de natură practică. Se pune accentul atât pe flexibilitate și extensibilitate cât și pe crearea unui mediu cât mai apropiat de cel oferit de un browser web obișnuit, în care să se poată publica conținut educativ în format tridimensional cu aproximativ aceeași ușurință cu care se publică orice alt tip de conținut media pe internet. Se urmărește crearea unui model de aplicație software care să permită persoanelor cu minime cunoștințe de programare să participe cu extensii și

cu materiale educative. *Simplitatea sistemului* este se asemenea un argument în favoarea aplicației informatice practice, cu un cod sursa ce va putea fi ușor de analizat. Eliminarea gradului sporit de tehnicitate ce 'acompaniază' în general mediile și sistemele de învățare 3D, ar putea capta interesul diverselor persoane implicate sau implicabile în realizarea de software educativ, a persoanelor implicate în sistemul de învățământ și nu în ultimul rând, al utilizatorilor finali.

## 2.4 Mini studiu empiric privind raportul aplicațiilor web - aplicații clasice cu destinație educativă

O metodă rapidă pentru cuantificarea interesului public pentru orice domeniu este metoda "motorului de căutare". Astfel, se poate beneficia de efortul uriaș depus de anumite companii pentru colectarea și clădirea datelor. Rezultatele nu sunt la fel de precise ca și studiile direcționate pe fenomenul proliferării tehnologiilor diverse dar sunt destul de credibile.

Pentru compararea proliferării metodologiilor web și a celor clasice cu destinație educativă vom considera numărul de pagini returnate precum și viteza de returnare. Primul indice este concludent. Al doilea indice poate oferi informații suplimentare, considerând mecanismul de depozitare (cache-ing) folosit pentru stocarea datelor cu număr mai mare de accesări. Pentru cuvinte cheie de cautare:

- web based learning environments 2.1
- 3D learning environment c++ 2.2

rezultatele sunt :

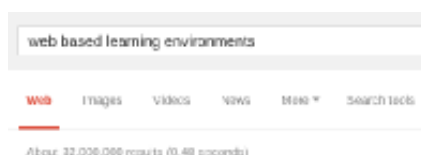


Figura 2.1: Căutare - tehnologii web

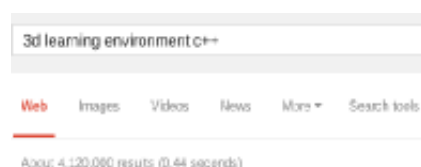


Figura 2.2: Căutare - tehnologii 3D și c++

#### 2.4. MINI STUDIU EMPIRIC PRIVIND RAPORTUL APLICAȚIILOR WEB - APLICAȚII CLASICE

	<i>tehnologie</i>	<i>pagini returnate</i>
<i>web</i>	32000000	0,48
<i>3D - C++</i>	4120000	0.44

Tabela 2.1: Rezultate

Datele returnate de către motorul de căutare indică faptul că tehnologiile web sunt cu mult mai apreciate decât tehnologiile clasice cu grafică 3D, numărul de rezultate returnate fiind de 8 ori mai mare în favoarea tehnologiei web pentru timp de răspuns comparabil.

# Capitolul 3

## Stadiul actual în domeniu

### 3.1 Introducere

Dezvoltarea tehnologică accelerată din ultimele decenii a afectat și domeniul educației asistate de calculator. Câteva exemple de medii accesibile publicului sunt: Second Life, Uther Academy și iSocial (<http://isocial.missouri.edu/iSocial/>). Fiecare dintre aceste implementări abordează diferit modul de livrare a materialelor educative. Deși codul sursă pentru majoritatea mediilor de învățare 3D este cod proprietar, se va încerca descrierea lor din perspectiva utilizatorului în următoarele subcapitole. Second Life este open source și poate fi studiat. Secțiunea 3.2 se azează pe similitudini în încercarea de a determina majoritatea opțiunilor pe care un utilizator le așteaptă de la mediile de învățare 3D.

### 3.2 Abordări similare

Toate exemplele notabile prezentate ulterior fac uz de avatare pentru a induce utilizatorului sentimentul de participare activă și imersiune în lumea virtuală. Utilizatorii sunt încurajați astfel să comunice între ei atât verbal cât și prin modificarea posturii și aparenței grafice a avatarului ce îi reprezintă în lumea virtuală.

Această tehnică este luată în considerare în implementarea aplicației descrise în acest studiu.

### 3.3 Second Life

Percepția generală este că Second Life (SL) ar fi un joc pe Internet. Nu este însă un joc organizat, cu reguli impuse și unde să fie urmărit un anumit scop. Pe site-ul web oficial, Second Life este descris ca fiind „o lume virtuală imaginată și creată de rezidenții ei”; într-adevăr, SL este o lume diversificată, în care poți întâlni oameni din toate colțurile

lumii reale. Este o rețea de tip social, care face parte din fenomenul din Internet numit Web 2.0.



Figura 3.1: Eveniment în Second Life

Cele mai importante și interesante activități de educative sunt derulate în Second Life:

- Vizite asistate în muzee și teatre virtuale.
- Cursuri în săli de clasa virtuale.
- Jocuri de tip "orientare turistică" cu puncte intermediare și indicii cu subiect educațional.
- Proiecte cu colaborare în echipă.
- Cursuri online la diverse universități
- Panouri informative.
- Grupuri educaționale

Majoritatea facilităților oferite de SL ar trebui să se regasească în orice platformă de e-learning cu avatare.

## 3.4 Uther Academy

UtherAcademy este o aplicație de e-learning care facilitează participarea studenților din toată lumea la cursuri în medii imersive 3D. Liniile educative propuse sunt din categoria dezvoltării profesionale. La data redactării acestei lucrări UtherAcademy avea deschise trei departamente :

- Academia de afaceri online.
- Cursuri pentru decoratori.
- Body arts.



Figura 3.2: U.A. - aulă virtuală

Modul de prezentare generală nu diferă foarte mult de Secon Life. Studenții participă online la cursuri în clase virtuale. Procesul de învățare este supravegheat de instructori.

### 3.5 iSocial

iSocial este un mediu de învățare 3D, dezvoltat pe baza toolkit-ului pentru crearea lumilor virtuale OpenWonderland, creat pentru predarea de competențe sociale tinerilor diagnosticați cu autism (ASD). În acest scop, iSocial facilitează interacțiunea socială și oferă suport pentru dezvoltarea de competențe sociale într-un mediu sigur și complet controlat.

Localizarea geografică poate restricționa accesul la tratamentele și exercițiile necesare recuperării sociale a tinerilor afectați de ASD. iSocial este una dintre soluțiile rezolvării pozitive a acestei probleme.



Figura 3.3: iSocial - panou informativ

<i>Denumire</i>	<i>Bibliotec/Ubuntu</i>	<i>Utilizare</i>
<i>OpenGL</i>	libGLU.so	API pentru grafica 2D și 3D
<i>Zlib</i>	libz.so	Comprimare date
<i>OpenSSL</i>	libssl.so	Protocoale de comunicare în rețea SSL și TLS
<i>OGG</i>	libogg.so	Format media audio-video
<i>PNG</i>	libpng12.so	Imagini PNG
<i>GLib</i>	libdbus-glib-1.so	Sistem de transmitere a mesajelor între procese
<i>GTK</i>	libgtk2.0-dev	GIMP toolkit
<i>OpenAL</i>	libopenal-dev;libalut-dev	OpenAL - Bibliotecă pt. redarea sunetului (audio)
<i>Vorbis</i>	libvorbis-dev	Codec audio-video (API)
<i>APACHE</i>	libapr1-dev	Apache portabile runtime
<i>JPEG</i>	libopenjpeg.so;libjpeg.so	Codec JPEG
<i>SDL</i>	libsdl1.2-dev	Media Layer - faciliteaza accesul la periferice
<i>Boost</i>	libboost-dev	Alternativa pentru C++ STL (+ comunicare în rețea)
<i>JsonCpp</i>	libjsoncpp-dev	Interpretarea fișierelor Json (c++)

Tabela 3.1: Tehnologii utilizabile pentru dezvoltarea de medii 3D sub Linux

## 3.6 Tehnici/Tehnologii folosite

Second Life (SL) este cea mai cunoscută și populară implementare a unei lumi virtuale. Deși nu este dezvoltată strict ca aplicație destinată instruirii în medii 3D, o parte însemnată a activităților desfășurate în Second Life sunt activități educative.

Un studiu al tehnologiilor folosite pentru dezvoltarea mediului Second Life este suficient pentru a identifica majoritatea bibliotecilor software folosite pentru dezvoltarea sub Linux a unei aplicații similare. S-a alcătuit o listă a tehnologiilor folosite pentru dezvoltarea SL.3.1

Pentru dezvoltarea aplicației client SL se folosesc următoarele tehnologii: OpenGL pentru redarea graficii 3D, GTK pentru interfețele grafice, Boost și APACHE pentru schimbul de date în rețea între aplicația client și serverul lumii virtuale, Formatul de date JSON pentru structurarea datelor interschimbate în rețea între aplicația client și server, OpenAL pentru redarea sunetului și OGG/Vorbis ca și format/codec media și PNG/JPEG pentru redarea stocarea imaginilor. Codul sursă pentru serverul SL nu este open source.

O parte dintre aceste tehnologii sunt folosite pentru realizarea mediului de învățare 3D descris în această lucrare.

# Capitolul 4

## Planul aplicației

### 4.1 Introducere

În acest capitol, în secțiunea 4.2, se stabilesc funcțiile sistemului. În secțiunea 4.3 se identifică numitorul comun al funcțiilor sistemului ca fiind facilitarea funcției de **comunicare** și se enumeră metodele prin care comunicarea la distanță se poate implementa. După analiza punctelor slabe și a punctelor forte ale acestor metode se afirmă alegerea metodei **CLIENT-SERVER** ca fiind decizia de design de urmat și implementat. În secțiunea 4.4 se proiectează un test ca suport pentru argumentarea deciziei de design (din secțiunea 4.5).

### 4.2 Funcțiile sistemului

#### 4.2.1 Minimul de funcționalitate de cercetat și implementat

Prin analiza funcțională, bazată pe observație liberă a câtorva dintre mediile 3D existente și accesibile pe internet, am putut determina o parte din funcțiile absolut necesare ale sistemului.

Aceste funcții sunt parte integrantă a cerințelor sistemului. La acestea se mai adaugă cerințele de ordin general precum: implementarea unei interfețe grafice ușor de folosit, mentenabilitatea și extensivitatea sistemului, etc..

#### **Funcții ale sistemului:**

- Utilizatorii sistemului sunt conștienți de prezența altor utilizatori (persoane reale) în lumea virtuală. Sistemul îndeplinește funcția de liant prin facilitarea schimbului de idei între utilizatori.



- Utilizatorii pot interacționa cu o parte dintre obiectele din lumea virtuală. În unele situații, rezultatul acestei interacțiuni conduce la modificări permanente în reprezentarea lumii virtuale simulate. Sistemul îndeplinește funcția de stocare a modelului tridimensional al lumii virtuale, inclusiv a stării acestui model, stare ce poate fi modificată de către utilizatori.
- Sistemul va stoca informațiile referitoare la identificarea utilizatorilor. Această identificare nu presupune stabilirea identității persoanei reale ci are funcția de a servi ca identificator pentru un catalog al progresului utilizatorului în procesul de învățare.
- Utilizatorii vor putea contribui la extinderea lumii virtuale prin publicarea de materiale educative dezvoltate de către aceștia. Sistemul va permite contribuția utilizatorilor va avea funcția de stocare și prezentare a noilor materiale educative.

Cea mai mare parte dintre aceste funcții se vor implementa în aplicația software demonstrativă atașată acestui proiect.

#### 4.2.2 Funcționalitate extinsă

Următoarele funcții nu vor fi incluse în software-ul demonstrativ, dar sunt recunoscute ca necesare pentru orice sistem de învățare.

- Funcția de asigurare a accesibilității pentru persoanele cu deficiențe de vedere sau deficiențe motorii.
- Funcția de redare a tuturor formelor de conținut media. Din motive tehnice și pentru complexității aplicației demonstrative, în mare măsură se vor exclude conținuturile media video și audio.

### 4.3 Decizie privind design-ul sistemului

Toate funcțiile primare ale sistemului fac referire la facilitarea comunicării între utilizatori pentru asigurarea schimbului liber de informații, sau între sistem și utilizatori cu scopul stocării și/sau accesării de date cu pentru diverse scopuri legate de funcționarea sistemului sau menținerea unui catalog al progresului utilizatorilor. Aceste schimburi de date se fac la distanță, pentru a nu condiționa geografic participarea la procesul de învățare.

Pentru implementarea comunicării în rețea se va alege unul dintre cele două modele general utilizate în practică: modelul client-server 4.2 și modelul p2p (point to point) 4.1.

**Modelul P2P** este un model descentralizat în care fiecare nod este atât client cât și server pentru celelalte noduri din rețea. Modelul este mai dificil de implementat, consistența datelor în rețea este mai greu de menținut și administrarea este destul de dificilă. Deși modelul este mai robust prin faptul că pierderea unui nod din rețea nu conduce la sistarea serviciului deoarece funcția de server este preluată de restul nodurilor;

acest model nu este cel mai bun pentru implementarea unei lumi virtuale, unde consistența datelor este foarte importantă. Aceasta consistență a datelor permite tuturor utilizatorilor să perceapă aceeași lume virtuală.

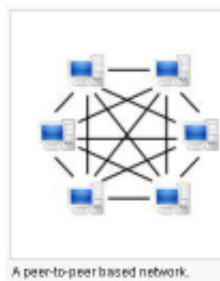


Figura 4.1: Modelul P2P

**Modelul Client-Server** este un model centralizat în care fiecare nod client comunică și depinde de serviciile nodului server. Acest model este relativ ușor de implementat, ușor de administrat, consistența datelor poate fi verificată și înpusă la nivelul serverului și stocarea datelor poate fi făcută centralizat la același nivel. Cel mai mare dezavantaj al acestui model este faptul că, pierderea accesului la server prin oprirea acestuia face tot sistemul inoperabil. Acest model este mult mai potrivit pentru implementarea unei lumi virtuale și a unui mediu de învățare 3D.



Figura 4.2: Modelul Client-Server

## 4.4 Test

Testul constă în simularea comunicării dinspre server către client. Un server va primi mesaje de la o aplicație client și va încerca să expedieze mesaje către un număr de 10 alte aplicații client, care salvează în câte un fișier mesajele primite de la server, cu o etichetă pentru momentul primirii mesajului. Mesajele vor avea o lungime cuprinsă între 1 și 1024 caractere iar timpul din etichetă va fi exprimat în secunde trecute de la 1 ianuarie 1970 (Unix epoch).

Se consideră **testul reușit** dacă toate mesajele ajung la toate aplicațiile client în aceeași ordine și într-un timp copmarabil, sub 1 secundă distanță.

Pentru **rularea testului** se codifică aplicația server demonstrativă și se utilizează utilitarul **telnet** ca program client. Un “shell script” va fi programat pentru **rularea automată** a testului. **Evaluarea** se va realiza cu ajutorul comenzii **diff**.

Codul sursă al aplicației server, scriptul pentru rularea automată a testului și rezultatele returnte de utilitarul diff sun atașat în Anexa A.

## 4.5 Argumentarea deciziei pe baza rezultatelor testării

Testul a fost executat cu succes. Datele obținute se află în ANEXA A.

Deoarece comunicarea între aplicațiile client și server sunt livrate consistent și la timp, se concluzionează faptul că se poate folosi schema client-server pentu implementarea aplicației.

# Capitolul 5

## Design-ul aplicației server

### 5.1 Introducere

În acest capitol, în secțiunea 5.2, se stabilesc funcțiile serverului și se prezintă tehnologiile folosite la producerea aplicației server. În secțiunea 5.3 se explică mecanismul adoptat pentru atingerea unui scop important, ”scalabilitatea aplicației”. În secțiunea 5.4 se stabilește modul și formatul de reprezentare a datelor transmise între aplicația client și aplicația server. Secțiunea 5.5 este dedicată prezentării părții celei mai relevante din diagrama claselor. În secțiunea 5.6 sunt stabiliți parametrii testului privind scalabilitatea aplicației, codificarea testului și rezultatele fiind expuse în Anexa B. Testul modului de comunicare și a eficienței comunicării între client și server s-a efectuat pentru capitolul anterior, rezultatele fiind publicate în Anexa A.

### 5.2 Funcțiile aplicației server

#### 5.2.1 Funcții de bază ale serverului

- Asigurarea comunicării înspre una sau mai multe aplicații client.
- Asigurarea comunicării între clienți prin intermediul serverului.
- Stocarea datelor referitoare la activitățile întreprinse de către utilizatori într-o bază de date.
- Eliberarea datelor din baza de date, la cererea utilizatorului.
- Stocarea și publicarea informațiilor ce descriu geometria lumii virtuale.
- Stocarea și publicarea acțiunilor utilizatorilor ce au efect asupra lumii virtuale.

Serverul are scop demonstrativ. Funcția de securitate a datelor la transfer și la stocare este ignorată. De asemenea, implementarea funcțiilor se realizează în cel mai simplu mod posibil, pentru reducerea complexității și dimensiunii aplicației.

### 5.2.2 Aspecte tehnice și principii generale de design

#### Limbaje de programare

Codificarea aplicației server se va executa în limbajele de programare Java și JavaScript.

Java este probabil cel mai potrivit limbaj pentru implementarea aplicațiilor server datorită numărului mare de biblioteci software utilizate pentru realizarea schimbului de date în rețelele de calculatoare. Acestea pun la dispoziția programatorului o multitudine de funcții de nivel înalt pentru comunicarea la distanță folosind diverse protocoale de comunicație de nivel înalt (HTTP/FTP/SSH/etc), cât și funcții de nivel mediu (bazate pe protocoalele TCP/UDP/etc..). Implementarea unei aplicații scalabile prin utilizarea "plug-in"-urilor este relativ ușor de realizat. Existența unui interpretor JavaScript pentru platforma Java este încă un argument în favoarea folosirii acestui limbaj de programare.

#### Sistemul de operare

Aplicațiile java sunt extrem de portabile, acestea rulând pe virtual orice sistem de operare și pe orice platformă hardware. Serverul mediului de învățare 3D va rula în varianta pentru GNU/Linux a "Mașinii Virtuale Java" (JVM). Fiabilitatea kernelului sistemului de operare Linux va afecta pozitiv viteza de rulare a mașinii virtuale java cu efect direct asupra performanței aplicației server dezvoltate pentru mediul virtual 3D.

#### Principii generale de design

**Pentru intermedierea comunicării între utilizatori** sau pentru orice fel de notificări trimise utilizatorilor s-a ales un tipar cunoscut în domeniul informatic sub numele "Observer Pattern". Această metodă definește și utilizează o dependență  $1 \rightarrow n$  între obiecte astfel încât un obiect își modifică starea, toate obiectele dependente sunt notificate. Aplicat, în cazul serverului (1), când un set de date este prelucrat, utilizatorii (n) sunt notificați.

**Pentru a realiza o aplicație scalabilă** se utilizează mecanismul de extindere dinamică a funcționalității cu "plug-in"-uri.

## 5.3 Mecanism pentru obținerea scalabilității aplicației server

Se va urmări o cât mai mare flexibilizare a sistemului, astfel încât dezvoltarea ulterioară să fie cât mai facilă. Pentru realizarea acestui obiectiv se va urmări integrarea limbajului de scriptare JavaScript, în serverul sistemului. Se are în vedere folosirea mecanismului de extindere dinamică a funcționalității sistemului prin "plug-in"-uri.

Utilizatorul, prin intermediul programului client al mediului virtual 3D, expediază **comenzi** și **informații** către server cu **scopul** de a fi prelucrate de către server și returnate sub forma de comenzi ce urmează a fi executate de către aplicația client, cu **efecte** asupra reprezentării mediului virtual 3D.

---

**DEF:** Un **plug-in** 5.1 este o componentă software care poate fi integrată de către aplicația gazdă la momentul execuției sau imediat la începutul procesului de execuție. De cele mai multe ori "plug-in"-urile sunt salvate în formă binară.

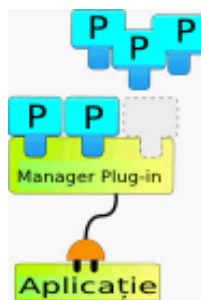


Figura 5.1: Arhitectura Plug-in

---

### 5.3.1 Interfața PLUG-IN-ului

"Plug-in"-urile aplicației server implementează următoarea interfață:

```
public interface CommandTypeDependentInterpreter {

    public static final int XML_FORMAT = 1;
    public static final int PALIN_TEXT_FORMAT = 2;
    public void prepareResponse(ServerCommand execCmd, int format);
    String getInterpretedResult();

}
```

### 5.3. MECANISM PENTRU OBTINEREA SCALABILITĂȚII APLICAȚIEI SERVER 19

Un plug-in extinde funcționalitatea aplicației server prin introducerea unei noi modalități prin care serverul răspunde la o cerere din partea aplicației client. În prima etapă, plug-in-ul preia comanda expediată de aplicația client prin intermediul metodei:

– *prepareResponse(ServerCommand execCmd, int format)* –

În a doua etapă, rezultatul interpretării și procesării datelor de către plug-in se va returna în vederea expedierii către utilizator.

#### 5.3.2 Tipuri de plug-in-uri dezvoltate pentru serverul mediului de învățare 3D

Două **tipuri de plug-in-uri** au fost dezvoltate pentru aplicația server. Primul tip de plug-in este **"clasic"**, implementat folosind 100% limbajul Java. Al doilea tip de plug-in este **"hibrid"**. La implementarea acestuia sunt utilizate două limbaje de programare: **Java și JavaScript**. Acest tip de plug-in este o noutate.

În principiu, un plug-in hibrid are rolul de a executa la nivelul serverului instrucțiuni codificate în limbaj JavaScript în numele aplicației client. Rezultatele sunt încapsulate în format XML și sunt returnate programului client pentru interpretare. Executarea comenzilor primite de la aplicația client este precedată de următorii doi pași pregătitori: inițializarea unui motor de prelucrare JavaScript (clasa `ScriptEngineManager`) și concatenarea la comanda utilizatorului a unei biblioteci care conține obiecte JavaScript utile.

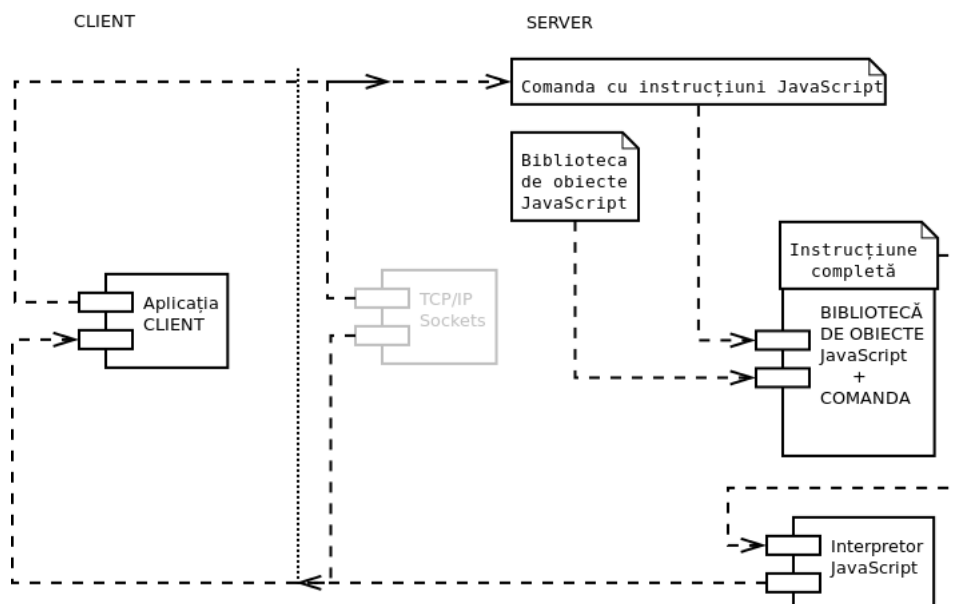


Figura 5.2: Schema Plug-in Hibrid

Avantajele plug-in-urilor hibride:

- Se pot descrie și implementa mult mai ușor și mai rapid.
- Biblioteca de obiecte JavaScript se concatenează la fiecare comandă trimisă de utilizator. Astfel, orice utilizator poate beneficia de obiectele conținute de această bibliotecă de obiecte.
- Codul se execută la nivelul seerverului, cu efect în creșterea vitezei de lucru a programului client.

Dezavantajele plug-in-urilor hibride:

- Comenzile JavaScript invalide pot destabiliza aplicația server.
- Codul se execută la nivelul serverului, cu efect în descreșterea vitezei de lucru a acestuia.
- Folosirea limbajului JavaScript introduce diverse riscuri de securitate a datelor.

### 5.3.3 Descrierea mecanismului de încărcare a plug-in-urilor

Serverul va ține evidența plugin-urilor disponibile într-un fișier de configurare, organizat în format XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- numele comenzii trimise de client -->
<!-- tipul comenzii (CLASIFICARE) -->
<!-- text ajutător (documentatie) -->
<!-- numărul paramerilor .. si expersia (reg-exp) pt diverse prelucrari -->
<!-- plug-in client delegat sa execute acțiuni în aplicația client -->
<!-- sursa de unde poate fi descărcat plug-in-ul client -->
<!-- executorul .. este plug-in-ul in aplicația server delegat pentru procesarea date
<!-- .. de la utilizator (CLIENT) -->
<commands>
.....
<command>
<name>@add</name>
<type>math</type>
<help>addition of 1..n numbers</help>
<param_count>-1</param_count>
<param_regexp>n</param_regexp>
<client_plugin>
    libL3dConsole.so
</client_plugin>
```



### 5.3. MECANISM PENTRU OBTINEREA SCALABILITĂȚII APLICAȚIEI SERVER 21

```
<client_plugin_source>
    http://aSite.com/plugins/libL3dConsole.so.1.0
</client_plugin_source>
<executor>
    ro.utcluj.learning3d.server.TEST_COMMAND_Math_Addition
</executor>
</command>
....
</commands>
```

Încărcarea dinamică a claselor este un mecanism foarte important pus la dispoziție programatorilor Java. Modul în care clasele sunt efectiv încărcate dinamic este foarte puțin cunoscut deoarece procesul este ascuns de către dezvoltatorii limbajului de programare. Mașina virtuală Java (JVM) folosește mai multe rutine de încărcare dinamică a claselor, rutine organizate într-o ierarhie complicată.

În principiu, pentru a încărca dinamic o clasă Java, programatorul trebuie să obțină o referință către un obiect **ClassLoader**. Un obiect "ClassLoader" poate încărca dinamic o clasă Java prin intermediul funcției **loadClass(..)** ce primește ca parametru numele clasei de încărcat și returnează o instanță a clasei încărcate în caz de success sau NULL în caz contrar.

Porțiunea de cod folosită pentru a încărca dinamic comenzi pentru serverul mediului de învățare 3D este un exemplu mai potrivit pentru explicarea mecanismului de încărcare a plug-in-urilor în Java.

...

```
try {
```

```
    ClassLoader myClassLoader = L3DServerManager.class.getClassLoader();
    String classNameToBeLoaded = execCmd.executor;
    Class<?> myClass = myClassLoader.loadClass(classNameToBeLoaded);
    Object executor = myClass.newInstance();
    ((CommandTypeDependentInterpreter)executor).prepareResponse
    (execCmd, CommandTypeDependentInterpreter.XML_FORMAT);
```

```
    return ((CommandTypeDependentInterpreter)executor).getInterpretedResult();
```

```
} catch(..)
```

..

## 5.4 Modul de reprezentare a datelor

În prezent, pentru transmiterea datelor între server și aplicații client se pot folosi mai multe scheme de structurare a informațiilor: informații nestructurate (compuse din șiruri de simboluri și caractere fără o structură impusă) sau informații structurate folosind una dintre schemele cele mai des utilizate precum XML (extended markup language) sau JSON (JavaScript Object Notation).

Informațiile transmise între aplicația server și aplicația client a mediului de învățare 3D sunt structurate și transmise în format XML.

### 5.4.1 Scurtă introducere în XML (CITE ...)

#### XML : Limbaj Extensibil de Marcare

- XML este un limbaj de marcare deoarece reprezintă un set de adnotări folosit pentru descrierea unor structuri de date de tip text.
- XML este extensibil deoarece adnotările pot fi definite de către utilizator.
- Nu depinde de nici un limbaj de programare.
- Conține blocuri asemănătoare cu elementele HTML.
- Elementele XML se auto documentează prin utilizarea numelor sugestive pentru transmiterea datelor.

#### Exemplu:

```
<books>
```

```
<book category="computer">  
  <author> D Knuth </author>  
  <title> The art of computer programming. VOL1 </title>  
  <edition>1</edition>  
</book>
```

```
<book category='Computer Programming'>  
  <author> D Knuth </author>  
  <title> The art of computer programming. VOL2 </title>  
  <edition>1</edition>  
</book>
```

```
</books>
```

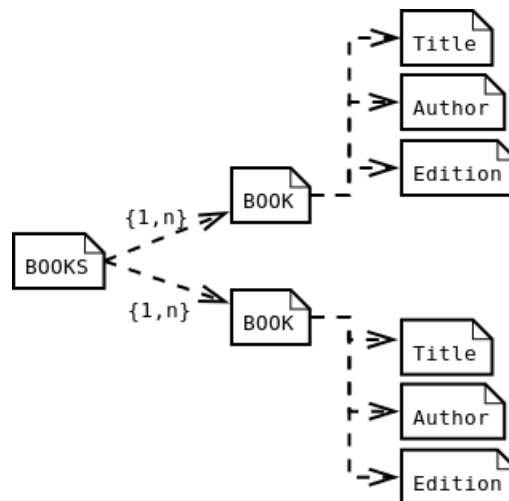


Figura 5.3: Structură arborescentă de date descrisă cu XML

### 5.4.2 Structura XML a datelor transmise dinspre server spre aplicația client

Atât aplicația server cât și aplicația client au cea mai mare parte din funcționalitate implementată în extensii (sau plug-in-uri). Din acest motiv, doar o porțiune a mesajelor transmise între client și server este standardizată. Restul mesajului trebuie să fie structurat conform nevoilor fiecărei extensii de funcționalitate în parte.

Mesajele transmise către clientul mediului de învățare 3D se clasifică în două categorii:

- Mesaje de eroare (în cazul în care serverul nu reușește să execute cererea clientului).
- Răspunsuri ale serverului la cererea aplicației client.

#### Structura XML a mesajelor de eroare transmise către aplicația client

```

<?xml version="1.0" encoding="UTF-8"?>
<response>
  <type></type>
  <exception>
    <message> .... </message>
  </exception>
</response>

```

#### Structura XML a răspunsurilor transmise către aplicația client

```

<?xml version="1.0" encoding="UTF-8"?>
<response>

```

```

<type> TIPUL COMENZII</type>
<name> NUMELE COMENZII </name>
<client_plugin>
    ----- NUMELE PLUG-INULUI CLIENT -----
</client_plugin>
<client_plugin_source>
    ----- SURSA ONLINE PT DESCĂRCAREA PLUG-IN-ULUI CLIENT -----
</client_plugin_source>
<raw>
    ----- FRAGMENT XML CONFORM CERINȚELE FIECĂRUI PLUG-IN -----
</raw>
</response>"

```

## 5.5 Schema serverului. Diagrama claselor

### 5.5.1 Modelarea comenzilor primite de server

Clasa **ServerCommand** este imaginea câmpurilor XML ce descriu registrl comenzilor și plug-in-urilor apelabile dinamic de către server. Această structură descrie o comandă.

Rolul clasei **CommandProcessor\_step1** este de a despărți textul primit de la aplicația client în subcomponente și de a stoca aceste componente în structura internă a unei instanțe a clasei **ServerCommand**. Clasa **CommandProcessor** apelează funcțiile celor două clase amintite anterior pentru a obține textul răspunsului de trimis către aplicația client.

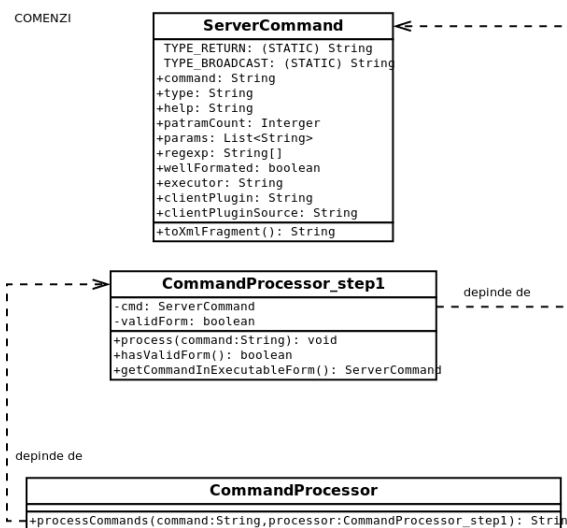


Figura 5.4: UML - COMENZI

# Capitolul 6

## Conluzii

### 6.1 Conținut

- un rezumat al contribuțiilor aduse
- a analiză critică a rezultatelor obținute: avantaje, dezavantaje, limitări
- o descriere a posibilelor dezvoltări și îmbunătățiri ulterioare

### 6.2 Detalii tehnice

#### 6.2.1 Dimensiune

Cca 3–5% din total.

# Anexa A

## Testarea fiabilității modelului CLIENT-SERVER

### A.1 Script

```
#!/bin/bash

# seconds_EPOCH.sh

while read line

do
    echo $(date +%s) ":" $line;
done
```

---

```
#!/bin/bash

# RUN_TESTS.sh

#!/bin/bash

telnetClients=10
rm ./*.out

echo "INCHIDEM SERVERUL ..."
kill -9 'pidof java'
echo "START SERVER .... in 3 secunde "
java server &
```

```

sleep 3s
echo "OK"

for ((1; i<=$telnetClients; ++i )) ;
do
    fileName="./file_${i}.out"
    telnet localhost 8080 | ./seconds_EPOCH.sh | tee -a $fileName &
done

```

## A.2 Server de test

```

import java.net.ServerSocket;
import java.net.Socket;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.util.ArrayList;

class SClient extends Thread {

    private Socket          socket;
    private Integer         serverId;
    private BufferedReader  istream;
    private PrintStream     ostream;
    private boolean         keepRunning;
    private static ArrayList<SClient> allServers = new ArrayList<SClient>();

    @SuppressWarnings("unused")
    private SClient() {
        // blocat
    }

    public SClient(Socket socket, Integer id) throws IOException {

        this.socket = socket;
        this.serverId = id;
        this.istream = new BufferedReader(new InputStreamReader(this.socket.getInputStream()));
        this ostream = new PrintStream(socket.getOutputStream());
        this.keepRunning = true;
    }

```

```

        allServers.add( this );
    }

    public void sendResponse(String response) {
        this.ostream.println(response);
    }

    private String getRequest() throws IOException {
        String line = "";

        try{
            int c = 0;
            while(((char)c)!='\n'){
                c= this.istream.read();
                line = line+(char)c;
                if(c==-1){
                    this.keepRunning = false;
                    break;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
            this.keepRunning = false;
        }

        return line;
    }

    @Override
    public void run() {
        super.run();

        while(keepRunning) {

            String text = "";
            try {
                text = this.getRequest();
            } catch (IOException e) {
                e.printStackTrace();
                text = "EROARE";
            }
        }
    }

```



```
        if ( text.contains("quit") || text.contains("exit") )
        {
            System.exit(0);
        }

        try{

            if ( allServers.size() >= 1 ) {

                for(SClient user : allServers) {
                    if(user!=null) {
                        user.sendResponse(text);
                    }
                }

            }

        } catch (Exception e){
            e.printStackTrace();
            this.sendResponse("\nEROARE");
        }
    }

    try {
        this.socket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

}

/*

    serverul de test

*/

public class server {

    public static void main(String[] args){
```

```

/*
 *  in asteptarea clientilor
 */
int serverId=1;
ServerSocket serverSocket;
try {
    serverSocket = new ServerSocket(8080);
    for(;;){
        Socket s = serverSocket.accept();
        System.out.print("\nNew□server□[id="+serverId+"]");
        /*
         *  client nou
         */
        Thread t = new SClient(s, serverId);
        serverId++;
        t.start();
    }
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

### A.2.1 Rezultate

La verificarea diferențelor de timp la primirea datelor de către programele client cu comanda **diff <fisier1> <fisier2>** . Testul este disponibil pentru verificare directorul "Teste/AppendixA" în arhiva GIT a lucrării. Descărcarea se poate face cu comanda **git clone <https://github.com/gabrielchmod777/virtualWorldVRML-CLIENT.git>**.

Testul este efectuat cu succes.