UNIVERSITATEA DE STAT DIN MOLDOVA

Cu titlu de manuscris
C.Z.U.: 004.434

LATUL GHEORGHE

ELABORAREA METODELOR ȘI MIJLOACELOR PENTRU CREAREA CURSURILOR DE INSTRUIRE ASISTATE DE CALCULATOR

121.03 – PROGRAMAREA CALCULATOARELOR

Rezumatul tezei de doctor în Informatică

CHIȘINĂU, 2025

Teza a fost elaborată la Departamentul Informatică al Universității de Stat din Moldova.

Conducător științific:

CĂPĂȚÂNĂ Gheorghe, doctor în științe tehnice, profesor universitar, Universitatea de Stat din Moldova.

Referenți oficiali:

MORARU Vasile, doctor în științe fizico-matematice, profesor universitar, Universitatea Tehnică din Moldova;

PERETEATCU Sergiu, doctor în științe tehnice, conferențiar universitar, Universitatea de Stat din Moldova.

Componența Consiliului Științific Specializat:

1. GUȚULEAC Emilian, doctor habilitat în informatică, profesor universitar, Universitatea Tehnică din Moldova, **președinte al C.Ș.S.**
2. **MARIN Maria**, doctor în științe pedagogice, conferențiar universitar, Universitatea de Stat din Moldova, **secretar științific al C.Ș.S.**;
3. **GAINDRIC Constantin**, doctor habilitat în informatică, profesor universitar, academician al Academiei de Științe a Moldovei, Institutul de Matematică și Informatică „Vladimir Andrunachievici", Universitatea de Stat din Moldova, **membru al C.Ș.S.**
4. **COJOCARU Igor**, doctor în informatică, Institutul de Dezvoltare a Societății Informaționale, **membru al C.Ș.S.**;
5. **TIȚCHIEV Inga**, doctor în informatică, conferențiar universitar, Institutul de Matematică și Informatică „Vladimir Andrunachievici", Universitatea de Stat din Moldova; **membru al C.Ș.S.**;

Susținerea va avea loc la **20.06.2025, ora 15^{00}**, în ședința Consiliului științific specializat în cadrul Universității de Stat din Moldova, str. A. Mateevici 60, Chișinău, MD-2009, Republica Moldova (bl.4, sala 213^{a}).

Teza de doctor și autoreferatul pot fi consultate la biblioteca Universității de Stat din Moldova și pe pagina web a CNAA (www.cnaa.acad.md).

Autoreferatul a fost expediat la **15 mai 2025.**

Secretar științific al C.Ș.S.: **MARIN Maria**, dr., conf. univ. _____

Conducător științific: **CĂPĂȚÂNĂ Gheorghe**, dr., prof. univ. _____

Autor: **LATUL Gheorghe,** lector. univ. _____

© Latul Gheorghe, 2025

CUPRINS

REPERELE CONCEPTUALE ALE CERCETĂRII 4

SINTEZA CAPITOLELOR 8

CONCLUZII GENERALE ȘI RECOMANDĂRI 23

BIBLIOGRAFIE 26

PUBLICAȚIILE AUTORULUI LA TEMA TEZEI 26

ADNOTARE 29

ABSTRACT 30

АННОТАЦИЯ 31

REPERELE CONCEPTUALE ALE CERCETĂRII

Actualitatea și importanța problemei abordate. În ultimele decenii, informatizarea a pătruns profund în toate domeniile vieții sociale, economice și științifice, generând transformări structurale și funcționale majore. În domeniul educației s-a resimțit în mod direct acest impact, prin digitalizarea resurselor, diversificarea metodelor de predare și integrarea progresivă a tehnologiilor informaționale în procesul de instruire. În prezent, procesul de formare inițială, perfecționare și reconversie profesională a cadrelor se sprijină tot mai mult pe utilizarea sistemelor informatizate de instruire, menite să sporească eficiența și accesibilitatea actului educațional [1].

Instruirea asistată de calculator a cunoscut o expansiune considerabilă în ultimele două decenii, fiind impulsionată de accesul tot mai larg la tehnologie și de necesitatea instruirii la distanță. Industria e-learning a crescut exponențial – de exemplu, între anii 2000 și 2020 s-a înregistrat o creștere[1] de 900%.

Evenimente recente, precum pandemia globală, au accelerat adoptarea instruirii online, ducând la mutarea cursurilor în spațiul virtual și la o creștere masivă a cererii de conținut educațional digital[1].

În acest context, dezvoltarea instrumentelor pentru crearea cursurilor de instruire asistată de calculator (IAC) a devenit extrem de actuală și importantă. Instituțiile educaționale și companiile caută soluții eficiente pentru a produce materiale de învățare interactive și adaptabile, fără a compromite calitatea educațională.

Progresul tehnologic a deschis noi oportunități și provocări în domeniul educațional. Apariția unor tehnologii precum aplicațiile web complexe, dispozitivele mobile, realitatea virtuală sau integrarea inteligenței artificiale în educație a ridicat nivelul de așteptare față de cursurile digitale. Adesea, cursurile moderne includ activități interactive, simulări, colaborare online și adaptare la ritmul sau nivelul fiecărui cursant. Pentru a face față acestei complexități, instrumentele de autor (*eng*. authoring tools) pentru cursuri IAC trebuie să permită proiectarea și gestionarea *scenariilor de învățare*, menținând totodată ușurința de utilizare pentru experții educaționali care nu sunt neapărat programatori. Astfel, se manifestă o nevoie acută de abordări, care să reducă decalajul dintre conceptul pedagogic și implementarea tehnică a unui curs [2, 3]. Acest lucru conferă subiectului o actualitate deosebită: cercetarea explorează modalități de

[1] Latest e-learning Statistics 2025. Available at: https://elearningstats.education. [Accesed 02.05.2025]

facilitare a creării cursurilor IAC de către practicieni, folosind abstracții de nivel înalt și automatizând detaliile tehnice.

În acest context, procesul de formare inițială, perfecționare și reconversie profesională se sprijină tot mai mult pe utilizarea sistemelor informatizate de instruire, menite să sporească eficiența și accesibilitatea actului educațional. Astfel, sistemele de instruire asistată de calculator s-au afirmat ca soluții inovatoare și viabile în sprijinul procesului educațional, oferind funcționalități de predare, învățare și evaluare într-un format digital integrat. Acestea completează sistemele tradiționale prin introducerea de componente interactive și adaptive, susținând emergența unor noi forme de predare precum învățarea bazată pe scenarii, personalizarea traseului educațional și feedback-ul automatizat.

Astfel, se conturează necesitatea dezvoltării unui instrument formal de modelare a cursurilor, care să permită transpunerea eficientă a experienței pedagogice în sisteme informatizate. Un astfel de instrument este limbajul formal de descriere a situațiilor instructive, însoțit de un analizor și un interpretator pentru sistemele IAC. Un avantaj important al acestei soluții îl constituie portabilitatea. Acest aspect este în concordanță cu cerințele actuale de interoperabilitate, promovate de standarde internaționale precum IMS (Instructional Management System) Learning Design[2].

O direcție promițătoare în dezvoltarea instrumentelor educaționale este utilizarea limbajelor declarative specifice domeniului (LSD). Un LSD este un limbaj conceput special pentru un anumit domeniu de aplicație (în cazul de față, educația asistată de calculator), oferind o sintaxă și semantică adecvate pentru a descrie *ce* trebuie realizat în acel domeniu, fără a detalia *cum* se realizează la nivel de cod. În contextul proiectării cursurilor IAC, LSD-urile permit definirea structurii unui curs – activități de învățare, resurse, secvențe de pași, roluri (student, profesor etc.) – într-o manieră declarativă, orientată pe conceptul pedagogic, urmând ca platforma software să interpreteze această specificație și să o transforme în implementare executabilă (ex. pagini web interactive, module e-learning, integrare în Sistemul de Management Educațional etc.).

Avantajele utilizării LSD-urilor în proiectarea cursurilor IAC sunt multiple. În primul rând, LSD-urile reduc complexitatea tehnică percepută de către autorii de conținut educațional. Adesea, crearea manuală a unui IAC interactiv implică sarcini repetitive, cod duplicat dificil de întreținut, adaptare pentru diferite platforme (desktop, web, mobil) și un risc mare de erori.

[2] **IEEE Learning Technology Standards Committee (LTSC)** Available at: https://www.ieeeltsc.org/ [Accesed 02.05.2025]

Un limbaj specific domeniului abstractizează aceste detalii: autorul cursului specifică logica instruirii (de exemplu, ordinea activităților, conținutul și regulile pedagogice), iar motorul din spate se ocupă de *gestionarea resurselor, generarea codului și asigurarea compatibilității tehnice* pe diverse platforme

Astfel, LSD-urile scad necesarul de cunoștințe de programare pentru designerii instrucționali, permițându-le să se concentreze pe aspectele educaționale. Studii în domeniu arată că un editor grafic bazat pe LSD poate îmbunătăți semnificativ experiența autorului comparativ cu un editor tehnic tradițional. De exemplu, un mediu de design grafic s-a dovedit mai intuitiv și mai ușor de gestionat vizual, decât notările arborescente reflexive folosite anterior în dezvoltarea model-driven [4]. Prin urmare, LSD-urile sporesc accesibilitatea instrumentelor de autor, făcând procesul de creare a cursurilor mai accesibil și mai eficient.

În al doilea rând, LSD-urile pot încorpora bune practici pedagogice și șabloane reutilizabile. Pentru că limbajul este construit în jurul conceptelor domeniului educațional, el poate oferi construcții predefinite care reflectă modele pedagogice consacrate. De exemplu, într-un LSD educațional se pot furniza șabloane pentru scenarii de învățare, inclusiv învățare adaptivă. LSD ajută profesorii să creeze design-uri de activități eficace, pornind de la modele validate în practică. Această posibilitate de reutilizare a elementelor de design didactic crește calitatea cursurilor (prin aplicarea unor strategii testate) și eficiența dezvoltării (prin economie de timp și efort în definirea de la zero a fiecărui scenariu).

În concluzie, stadiul actual al domeniului evidențiază o tendință clară de digitalizare a educației, dar și nevoia stringentă de instrumente specializate care să faciliteze crearea de conținut educațional de calitate, fără a impune autorilor de cursuri necesitatea expertizei tehnice avansate în domeniul IT și al programării.

Ipoteză de cercetare. Elaborarea unui limbaj declarativ specific domeniului și a unei tehnologii de proiectare eficientă, flexibilă și scalabilă a cursurilor de instruire asistată de calculator, aplicabile în diverse contexte educaționale

Scopul și obiectivele cercetării. Scopul cercetării constă în crearea tehnologiei originale și instrumentelor informatice pentru proiectarea și programarea cursurilor de instruire asistată de calculator. Scopul a determinat următoarele obiective:

- Elaborarea metodelor și mijloacelor de programare a cursurilor de instruire asistată de calculator.
- Formularea cerințelor principale față de sistemele de instruire asistate de calculator.
- Formularea cerințelor față de limbajul descrierii cursurilor asistate de calculator.

- Elaborarea limbajului specific domeniului de descriere a cursurilor asistate de calculator.
- Elaborarea unei tehnologii de proiectare a cursurilor de instruire asistată de calculator.
- Dezvoltarea compilatorului și prototipului interpretorului limbajului elaborat.

Problema de cercetare constă în crearea tehnologiei de proiectare originale a programelor de instruire asistată de calculator bazată pe descrierea situațiilor instructive stereotipice, definirea scenariilor de instruire în baza scenariilor cauzale cu scopul elaborării limbajului de programare specific domeniului de creare a cursurilor de instruire asistate de calculator.

Semnificația teoretică a lucrării constă în analiza originală a procesului didactic, formalizarea unei submulțimi a acestui proces, introducerea și descrierea situațiilor instructive stereotipice. În lucrare este introdusă noțiunea de *scenariu de instruire* în baza *scenariului cauzal*.

Semnificația practică a lucrării constă în crearea limbajului descrierii cursurilor de instruire asistate de calculator. Cu ajutorul acestui limbaj a fost creat și implementat cursul de instruire asistat de calculator „*Programarea în limbajul C*", confirmat de actul de implementare în procesul de instruire a cursului elaborat.

Metodologia cercetării. Cercetările efectuate în teza au la bază analiza sistemelor de instruire asistată de calculator, a sistemelor de inteligența artificială, modelarea informatică în ingineria soft-ului, teoria gramaticelor formale și automatelor, teoria compilării. Ideile și soluțiile propuse au fost testate prin crearea prototipurilor soft-ului.

Aprobarea rezultatelor. Rezultatele obținute au fost publicate în 23 de lucrări *(a se vedea Publicațiile autorului la tema tezei)* și comunicate în cadrul a 19 conferințe:

- International Conference on Computer Technologies in Education (ICCTE'93). – (Kiev, Ucraina: 1993).
- International Conference on Computer Technologies in Education (ICCTE'94). – (Crimeea, Ucraina: 1994).
- Tehnologii avansate în pragul secolului XXI. Conferința științifico-practică (Chișinău, Moldova, 5– 7 octombrie, 2000).
- 7 conferințe ale Corpului didactico-științific al Universității de Stat din Moldova (anii 1992 - 2006).
- 5 conferințe „Mathematics & Information Technologies: Research and Education" (MITRE 2008 ,2011, 2015, 2016, 2023). Chișinău.

- În cadrul seminarelor Departamentului Informatică a Universității de Stat din Moldova.

O parte din rezultatele expuse în teză sunt obținute de autor în decursul activității la îndeplinirea unor proiecte din domeniul utilizării tehnicii de calcul în procesul de instruire:

- Elaborarea softului instrumental pentru crearea programelor de instruire asistate de calculator. Nr. înregistrării de stat 0195M00126, Nr. de inventar 0396M00141. Chișinău, USM, 1993-1995.
- Elaborarea CASE-tehnologiei de proiectare a cursurilor de instruire asistate de calculator. Nr. înregistrării de stat 0196M00877, Nr. de inventar 0298M00736. Chișinău, USM, 1997-1998. (executant responsabil).
- Proiect instituțional „Dezvoltarea sistemelor informaționale inteligente orientate spre familii de probleme decizionale cu aplicare în educație și cercetare", codul 15.817.02.38A. Chișinău, USM, 2007-2008.
- Proiectul Elaborarea și aplicarea metodelor inovaționale în instruire la distanță, codul 08.815.08.04A. Chișinău, USM, 2018-2019.
- A fost implementat cursul de instruire "Programarea în limbajul C" creat în baza soft-ului elaborat în teza.

Implementarea rezultatelor lucrării: programele instrumentale și cursul de instruire „Limbajul de programare C" au fost implementate la Facultatea de Matematică și Informatică a Universității de Stat din Moldova, confirmate prin actul de implementare

Structura tezei: Teza este scrisă în limba română cu titlu de manuscris. Lucrarea este structurată în patru capitole, concluzii generale și recomandări, adnotări în limba română, rusă și engleză, bibliografie ce cuprinde 113 de titluri. Volumul total al tezei este de 169 de pagini, dintre care 115 pagini text de bază.

SINTEZA CAPITOLELOR

În **Introducere**" este evidențiată importanța și actualitatea temei de cercetare abordate. Sunt elucidate scopul și obiectivele lucrării, contribuțiile științifice originale reflectate prin noutatea rezultatelor obținute. De asemenea, sunt expuse valoarea teoretică și aplicativă a cercetării, precum și informații privind validarea și confirmarea rezultatelor obținute.

Capitolul „1. Sistemele de instruire asistată de calculator" tratează aspectele fundamentale actuale ale utilizării tehnologiilor informatice în procesul educațional, oferind un cadru teoretic pentru înțelegerea, clasificarea și evaluarea acestor sisteme. Sunt definite conceptele esențiale, evidențiate rolul și impactul instruirii asistate de calculator asupra creșterii

eficienței procesului de predare-învățare în contextul informatizării accelerate a societății contemporane.

În continuare, sistemele IAC sunt clasificate în funcție de **destinația funcțională** a aplicațiilor: *sisteme autor (authoring systems), manuale electronice, sisteme de căutare a informației și programe de modelare*. Această clasificare subliniază faptul că dezvoltarea instruirii asistate nu se limitează la livrarea pasivă a informației, ci include și proiectarea activă a mediilor educaționale și a proceselor de învățare.

Analiza acestor sisteme este structurată în patru categorii principale: *sisteme bazate pe cadre, sisteme bazate pe limbaje de scripting, sisteme cu interfețe de tip meniu* și *sisteme multimedia și hipermedia.*

Sistemele bazate pe cadre sunt reprezentate de produse precum MICROTEXT, STAF, AOS VUZ, ASTRA și LINKWAY. Acestea organizează conținutul didactic în secvențe de cadre sau ecrane, cu legături prestabilite între ele. Predomină structuri de navigare liniară sau slab ramificată, fără suport pentru adaptarea dinamică la acțiunile cursantului. Lipsa reprezentării declarative a cunoștințelor și absența relațiilor semantice complexe constituie limitări majore ale acestor sisteme.

Sistemele bazate pe limbaje de scripting, precum PHILVAS, VISA, TOPIC și SEDF/2, oferă autorilor o mai mare libertate în proiectarea logicii de instruire. Totuși, utilizarea lor presupune cunoștințe de programare, ceea ce limitează accesibilitatea pentru cadrele didactice fără pregătire tehnică avansată. De asemenea, modelul de instruire rămâne predominant procedural și adesea greu de extins cu noi funcționalități adaptative.

Sistemele cu interfețe de tip meniu, cum sunt RACURS, MIDOS și ADONIS, simplifică procesul de creare a cursurilor, punând la dispoziția utilizatorilor șabloane predefinite și fluxuri ghidate de lucru. Prin eliminarea nevoii de programare explicită, aceste sisteme facilitează accesul larg la tehnologiile de creare a cursurilor informatizate. Cu toate acestea, ele păstrează modelul tradițional de instruire bazat pe navigare secvențială sau aleatorie, fără mecanisme solide de adaptare personalizată la nivel de scenariu educațional.

Sistemele multimedia și hipermedia (Authorware, Asymetrix ToolBook, Education ONE, TenCORE) extind considerabil posibilitățile expresive ale autorilor, integrând texte, imagini, sunet, animații și videoclipuri. Ele permit crearea de cursuri atractive și interactive din punct de vedere vizual, însă rămân, în cea mai mare parte, limitate la un model de navigare liberă sau la secvențieri statice ale conținutului. Lipsa unei reprezentări semantice formale a cunoștințelor restrânge capacitatea acestor sisteme de a sprijini adaptarea automată și inteligentă a traseului de învățare.

Analiza comparativă realizată asupra acestor sisteme evidențiază *limitările fundamentale* ale abordărilor existente:

- Dependența excesivă de scenarii rigide și de secvențialitatea prestabilită a lecțiilor.
- Absența reprezentărilor declarative de tip frame pentru cunoștințe și relații educaționale complexe.
- Dificultatea adaptării automate a instruirii în funcție de profilul și evoluția cursantului.
- Lipsa interoperabilității și a reutilizării flexibile a conținutului educațional între platforme și contexte variate.

Pornind de la aceste constatări, se argumentează necesitatea elaborării unor noi metode și instrumente pentru dezvoltarea cursurilor de instruire asistată de calculator, bazate pe modele declarative și scenarii cauzale, ce permite proiectarea cursurilor de instruire urmând *Ingineria Dirijată de Modele* (*eng*. Model-Driven Engineering). Este justificată utilizarea reprezentării cunoștințelor prin intermediul frame-urilor, inspirată din domeniul inteligenței artificiale și implementarea mecanismelor de gestionare a instruirii pe baza relațiilor cauzale între situațiile educaționale.

În cadrul acestui nou model *cunoștințele educaționale* sunt organizate sub forma unor frame-uri, fiecare descriind o situație instructivă stereotipică, cu atribute și relații semantice specifice; *scenariile cauzale* modelează dinamica procesului educațional, definind modul în care tranzițiile între situațiile instructive sunt guvernate de acțiunile și răspunsurile cursantului; *adaptabilitatea* este încorporată în structura cursului, permițând personalizarea parcursului de învățare pe baza performanțelor și nevoilor individuale ale utilizatorilor.

Capitolul formulează direcțiile fundamentale de cercetare care stau la baza tezei întreaga teză:

- Dezvoltarea tehnologiilor de proiectare a cursurilor de instruire asistată de calculator bazate pe modelul domeniului educațional și pe reprezentarea declarativă a cunoștințelor.
- Elaborarea unui limbaj evoluat de descriere, bazat pe frame-uri, destinat dezvoltării sistematice a scenariilor educaționale adaptative.
- Proiectarea și implementarea unui interpretor pentru limbajul dezvoltat, cu scopul de a asigura portabilitatea cursurilor între diverse platforme și sisteme de operare.
- Validarea experimentală a tehnologiilor propuse prin dezvoltarea și testarea prototipurilor funcționale.

Prin analiza critică a stadiului actual al sistemelor de instruire asistată de calculator și prin formularea unei viziuni orientate către reprezentarea formală și adaptabilitatea procesului de

instruire, capitolul pune bazele conceptuale ale inovațiilor propuse în teză. Se argumentează necesitatea depășirii limitelor metodologiilor existente și se trasează direcțiile de evoluție spre sisteme educaționale informatizate mai flexibile, mai inteligente și mai centrate pe nevoile cursantului.

Capitolul „2. Modelul structural al procesului de instruire pentru cursurile de instruire asistate de calculator” are drept obiectiv elaborarea și fundamentarea unui model structural riguros al procesului de instruire, orientat spre realizarea cursurilor de instruire asistate de calculator prin Ingineria Dirijată de Model (IDM).

Se introduce conceptul de pas al programului de instruire, definit ca unitatea logică de interacțiune dintre student și sistem, care combină prezentarea de informații, solicitările de răspuns și procesarea feedback-ului. Această structurare permite modelarea procesului de învățare ca o succesiune de pași, fiecare cu obiective clare, reguli de tranziție și mecanisme de verificare a performanțelor cursantului.

Se subliniază importanța gestionării adaptive a procesului educațional. Gestionarea este înțeleasă nu doar ca o simplă transmitere de informații, ci ca un proces complex de monitorizare și adaptarea dinamică a traiectoriei de învățare a cursantului, în funcție de răspunsurile și comportamentele acestuia. Astfel, se pune accent pe rolul legăturii inverse în sistemele IAC, ca mecanism esențial pentru personalizarea instruirii și maximizarea eficienței educaționale.

În vederea proiectării eficiente a cursurilor asistate de calculator, autorul identifică și definește un set de situații instructive stereotipice esențiale pentru proiectarea eficientă a cursurilor de IAC. Acestea includ: situațiile instructive precum expunerea materialului didactic (P), formularea sarcinilor și solicitărilor către student (T), formularea întrebărilor (Q), analiza răspunsurilor (E), replicile de feedback (R), repetarea exercițiilor (S), controlul încercărilor (C), gestionarea răspunsurilor nerecunoscute (U) și asistența la cerere (H). Noțiunile sunt definite riguros și integrate într-un model coerent rare este utilizat pentru proiectare a cursurilor IAC.

Această modelare a situațiilor instructive stereotipice oferă o bază formală solidă pentru construirea scenariilor educaționale și pentru automatizarea procesului de instruire.

În continuare, se propune utilizarea *scenariilor cauzale* drept metodă de formalizare a succesiunii evenimentelor instructive și de modelare a dinamicii instruirii. Aceste scenarii, structurate sub forma unor frame-uri, permit reprezentarea relațiilor cauzale dintre acțiunile studentului și evoluția procesului de învățare. În acest context, scenariile cauzale constituie o bază solidă pentru dezvoltarea tehnologiilor de proiectare a cursurilor asistate de calculator.

Modelarea cursurilor prin scenarii cauzale implică utilizarea frame-urilor ca entități de bază pentru reprezentarea situațiilor instructive și a relațiilor dintre ele. Fiecare frame descrie o

situație educațională stereotipică, caracterizată printr-un set de *sloturi* care specifică datele necesare, acțiunile posibile și condițiile de tranziție. Această abordare, inspirată din inteligența artificială, permite o formalizare clară și extensibilă a procesului educațional, oferind în același timp suport pentru adaptare și personalizare.

Capitolul propune, de asemenea, utilizarea grafurilor de informații și grafurilor de dirijare pentru reprezentarea logică a structurii cursurilor. Grafurile de informații descriu relațiile dintre unitățile de cunoștințe, în timp ce grafurile de dirijare modelează succesiunea pașilor de instruire în funcție de acțiunile cursantului. Această reprezentare grafică sprijină dezvoltarea de cursuri complexe, modulare și ușor de actualizat.

O componentă importantă a capitolului este analiza modelării materialului didactic. Se introduce conceptul de *unitate de cunoștințe* ca element fundamental al structurii informaționale a cursurilor. Unitățile de cunoștințe sunt organizate ierarhic și asociate cu situații instructive concrete, facilitând adaptarea cursului în funcție de nivelul de pregătire și nevoile specifice ale fiecărui cursant.

Se abordează, de asemenea, problematica modelării cunoștințelor și comportamentului studenților. Este cercetată utilizarea de modele interne pentru reprezentarea stării de cunoaștere a cursantului și a progresului său pe parcursul instruirii. Aceste modele permit sistemului să evalueze performanța, să detecteze lacunele în înțelegere și să ajusteze în mod dinamic traseul de învățare.

În ceea ce privește evaluarea cursanților, sunt descrise metode de analiză a răspunsurilor, precum și strategii de feedback adaptativ. Se propun mecanisme pentru recunoașterea răspunsurilor incomplete sau greșite și pentru oferirea de indicații corective în timp real, contribuind la eficientizarea procesului de învățare.

Un alt aspect abordat este integrarea colectării și analizei datelor despre activitatea cursanților. Aceste date sunt utilizate atât pentru adaptarea dinamică a instruirii, cât și pentru îmbunătățirea continuă a conținutului și structurii cursurilor, în baza analizei comportamentului real al utilizatorilor.

Capitolul evidențiază că, pentru realizarea unor cursuri eficiente și adaptative, este esențială utilizarea unei metodologii de proiectare bazate pe modelul domeniului de instruire. Se argumentează că această abordare, specifică paradigmei IDM permite o mai bună structurare a cunoștințelor, o mai mare flexibilitate în actualizarea cursurilor și o adaptare mai eficientă la evoluțiile tehnologice și pedagogice.

Astfel, capitolul trasează direcțiile de dezvoltare necesare pentru elaborarea unor tehnologii avansate de creare a cursurilor asistate de calculator:

- Utilizarea reprezentării declarative a cunoștințelor prin frame-uri.
- Modelarea procesului de instruire prin scenarii cauzale adaptative.
- Crearea de limbaje declarative dedicate pentru descrierea cursurilor IAC,
- Dezvoltarea de instrumente software pentru proiectarea și interpretarea cursurilor bazate pe modele formale.

În concluzie, capitolul 2 constituie o contribuție teoretică și metodologică esențială pentru fundamentarea cercetării. Prin analiza sistematică a procesului de instruire, identificarea situațiilor instructive stereotipice, introducerea scenariilor cauzale și propunerea utilizării modelării declarative, se creează premisele pentru dezvoltarea unor programe de instruire asistată de calculator eficiente și ușor de întreținut.

Capitolul „3. Limbajul de descriere a cursurilor asistate de calculator MoCo” este dedicat dezvoltării limbajului MoCo (Modeling Courses).

Limbajul de descriere a situațiilor instructive, este destinat pentru crearea cursurilor asistate de calculator în baza scenariilor. Limbajul propus permite descrierea atât a părții informaționale, cât și a părții de control. În componența limbajului de descriere a situațiilor instructive sunt incluse mijloacele pentru structurarea materialului didactic, pentru prezentarea informației studentului, inclusiv imagini grafice și video, de analiză a răspunsurilor la întrebări, de organizare a lucrărilor de control și testare, precum și mijloace de includere a informației de îndrumare și a utilitarilor auxiliare [5]. Limbajul se bazează pe noțiunea de situație instructivă stereotipică reprezentată prin scenariul cauzal și este un limbaj declarativ (neprocedural).

Limbajul MoCo [20] este un limbaj specializat și face parte, de asemenea, din categoria *limbajelor specifice domeniului* [6, 7], limbajelor descrierilor cursurilor de instruire asistate de calculator.

Un element-cheie al acestei abordări este utilizarea de LSD-uri permite descrierea cursurilor și a situațiilor de instruire într-o formă apropiată de limbajul de specialitate educațional, facilitând astfel comunicarea între experții pedagogi și dezvoltatorii de software. De exemplu, în loc să se scrie cod general într-un limbaj de programare complex pentru a defini logica unui curs, un expert în instruire poate folosi un LSD dedicat instruirii pentru a specifica secvențe de predare-învățare folosind termeni pedagogici familiari. Prin urmare, LSD-urile contribuie la reducerea distanței semantice dintre conceptele din domeniul educației și implementarea tehnică, făcând posibilă o proiectare *mai rapidă, mai fiabilă și mai prietenoasă* a cursurilor asistate de calculator.

În a doua parte a capitolului sunt definite structurile de bază ale limbajului, incluzând regulile de formare a identificatorilor, numerelor, expresiilor și simbolurilor speciale. Sunt

prezentate tipurile de date utilizate, inclusiv parametrii legați de bazele de date educaționale, precum numele studentului, timpul activ, răspunsurile corecte și greșite și parametrii suplimentari stabiliți de autorii cursurilor.

Se introduce conceptul de curs asistat de calculator ca o colecție de lecții structurate, precedate de un frontispiciu, fiecare lecție fiind compusă dintr-o succesiune de situații instructive și de luare a deciziilor. Limbajul definește clar sintaxa și semantica acestor elemente, utilizând un sistem de delimitare clară a începutului și sfârșitului fiecărei componente.

Limbajul este descris formal utilizând notația Backus-Naur (BNF), asigurând o definiție riguroasă a sintaxei sale. Totodată, în anexa tezei este prezentată o descriere suplimentară a limbajului cu ajutorul diagramelor sintactice, ceea ce facilitează înțelegerea structurii și a regulilor sale de utilizare.

Instrucțiunile principale ale limbajului sunt: *Lesson* (Lecție), *Information* (Informație), *Question* (Întrebare), *Wrong* (Răspunsul greșit), *Correct* (Răspunsul corect), *Reaction* (Reacție la răspuns), *Diagnostics_Correct* (Punctele obținute pentru răspunsul corect), *Diagnostics_Wrong* (Punctele obținute pentru răspunsul greșit), *Help* (Ajutor oferit la o întrebare), *Edit* (Editare, pregătirea răspunsului pentru analiza), *Test* (Testare *Check (Lucrare de control)*), *Evaluate (*Evaluare a unui test sau lucrării de control*), Execute* (Apel la o funcție externă), *Branch* (Ramificare), *Manual* (Acces la un manual).

Descrierea lecțiilor include atât prezentarea materialului informativ, cât și gestionarea interacțiunii cu studentul prin întrebări de tip deschis, singular sau multiplu, testări și lucrări de control. Se detaliază mecanismele de evaluare a răspunsurilor, utilizând parametrii de diagnosticare, reacție, ajutor, editare și tratare a răspunsurilor neprevăzute.

O componentă esențială o constituie modelarea scenariilor educaționale sub formă de rețele cauzale, în care progresul studentului este condiționat de răspunsurile și acțiunile anterioare. Limbajul permite organizarea ramificațiilor logice și adaptarea parcursului educațional în funcție de performanțele individuale.

În contextul proiectării cursurilor asistate de calculator, limbajul MoCo este conceput pentru a descrie atât *partea informațională* a procesului de instruire (prezentări de materiale, imagini, video), cât și *partea de control* (logica tranzițiilor între situațiile instructive, regulile de evaluare a răspunsurilor și mecanismele de feedback). Prin structurarea clară a acestor două componente, MoCo asigură o reprezentare completă și coerentă a cursului.

Capitolul evidențiază avantajele utilizării unui *limbaj specific domeniului* în proiectarea cursurilor de instruire asistată de calculator. Astfel de limbaje permit captarea directă a conceptelor din domeniul instruirii, reducând diferența dintre exprimarea pedagogică și

implementarea tehnică. Se argumentează că limbajele LSD oferă o expresivitate ridicată, o claritate semantică superioară și o adaptabilitate mai bună la nevoile specifice ale utilizatorilor, în comparație cu limbajele de programare generaliste.

Se analizează principalele caracteristici ale limbajelor declarative orientate pe domenii, subliniindu-se că MoCo permite descrierea situațiilor de instruire fără a impune autorilor să specifice pașii de execuție în detaliu. Astfel, utilizatorii pot concentra efortul asupra modelării pedagogice, în timp ce detaliile de implementare și de execuție sunt gestionate de interpretorul limbajului.

Utilizarea LSD-urilor în dezvoltarea de sisteme software aduce o serie de beneficii semnificative, confirmate de numeroase studii și experiențe practice. Avantajele principale sunt: expresivitate și concizie sporite, claritate și comunicare îmbunătățite, productivitate și rapiditate în dezvoltare, fiabilitate și reducerea erorilor, adaptabilitate și reutilizare multiplatformă, colaborare între experți de domeniu și programatori.

Provocările LSD-urilor sunt următoarele: costuri de dezvoltare și mentenanță a limbajului, costuri de instruire și adoptare, stabilirea echilibrului între specificitate și reutilizare/integrabilitate, acceptare de către utilizatori, limitări date de domeniu și evoluția tehnologică.

Structura limbajului MoCo este detaliată prin prezentarea principalelor concepte și constructe. Sunt definite unități precum:

- *Situațiile instructive* – farame-urile declarative care descriu un moment educațional.
- *Prezentările informaționale* – secțiuni destinate livrării conținutului didactic.
- *Activitățile de evaluare* – întrebări, teste și sarcini de control.
- *Mecanismele de feedback* – replici și reacții adaptative la acțiunile cursantului.

O componentă esențială este modelarea scenariilor cauzale. În MoCo, scenariile sunt descrise sub forma unor rețele de situații instructive, conectate prin relații cauzale condiționate de răspunsurile și comportamentele cursanților. Aceste relații permit dezvoltarea de cursuri adaptative, în care traseul de instruire variază în funcție de performanța individuală.

În final, se subliniază că limbajul MoCo, prin specificitatea sa orientată spre domeniul educațional și prin suportul său pentru modelarea declarativă a proceselor de instruire, reprezintă un pas important către realizarea unor sisteme de instruire asistată de calculator mai flexibile, mai adaptative și mai eficiente.

Rezultatele prezentate în capitol constituie fundamentul tehnologic pentru dezvoltarea ulterioară a platformei de creare a cursurilor asistate de calculator, care utilizează limbajul MoCo ca mediu de descriere standardizat.

Capitolul „4. Proiectarea platformei pentru crearea cursurilor de instruire asistate de calculator” ~~se~~ ia în considerare problemele creării translatorului și unei platforme de dezvoltare a aplicațiilor în limbajul MoCo.

O parte a capitolului este dedicată analizei lexicale, sintactice și semantice a limbajului MoCo de descriere a cursurilor asistate de calculator, în contextul dezvoltării unui translator pentru acest limbaj. Se pornește de la definirea formală a gramaticii limbajului MoCo, exprimată ca o gramatică independentă de context, unde sunt specificate mulțimile de simboluri terminale și neterminale, axioma și regulile de producție.

Pentru a asigura eficiența procesului de analiză, gramatica inițială a fost transformată într-o gramatică LL(1) (*Gramatica cu parsare de la stânga și derivare stângă – eng. Left Leftmost grammar*), utilizând metode standard de eliminare a recursiei stângi și de factorizare stângă. Această transformare permite aplicarea algoritmului predictiv de analiză sintactică, ceea ce reduce complexitatea analizei la timp liniar în raport cu lungimea șirului de intrare.

Etapele procesului de translare sunt detaliate sistematic: analiza lexicală, analiza sintactică și analiza semantică (generarea codului intermediar). Programul sursă scris în limbajul MoCo este prelucrat inițial ca o succesiune de simboluri, transformate ulterior într-un flux de lexeme. Acestea constituie baza pentru construcția arborelui sintactic și pentru generarea codului intermediar interpretabil.

Analiza lexicală este abordată ca procesul de grupare a simbolurilor de intrare în unități lexicale (lexeme) distincte, fiecare lexem fiind definit printr-o pereche (tip, date). Sunt detaliate regulile de recunoaștere a cuvintelor-cheie, identificatorilor, constantelor numerice și textelor, precum și tratamentul simbolurilor speciale. Se subliniază rolul tabelelor de simboluri și utilizarea memoriei dinamice sub formă de stivă pentru gestiunea informației despre lexeme.

În continuare, se detaliază procesul de analiză sintactică utilizând algoritmul 1-predictiv stâng, bazat pe conceptul de „avan-simbol” și pe utilizarea unei stive pentru derivarea stângă a șirului de intrare. Se descrie construcția tabelului de dirijare MTP (Mașina de Traducere Predictivă), utilizat pentru determinarea acțiunilor de parsare: expansiune, avans sau semnalarea erorii.

Configurarea algoritmului predictiv este formalizată prin tripletul *(șir de intrare, stivă, bandă de ieșire),* iar funcționarea acestuia este explicată prin reguli precise: *extinderea simbolului neterminal, avansul pe șirul de intrare* sau *recunoașterea erorilor de sintaxă.*

Se prezintă un exemplu detaliat de analiză a unei secvențe lexicale corespunzătoare descrierii unei lecții, demonstrând aplicarea regulilor și funcționarea tabelului MTP.

În ceea ce privește analiza semantică, aceasta constă în interpretarea structurii sintactice obținute și în generarea codului intermediar. Programul sursă este astfel transformat într-o reprezentare internă structurată, destinată interpretării de către motorul de execuție. Arborele sintactic, rezultat al analizei, reflectă atât organizarea logică a cursului, cât și relațiile dintre elementele instructive și mecanismele de control.

Se detaliază funcționarea translatorului limbajului MoCo, alcătuit din două faze principale: analiza lexicală și faza combinată de analiză sintactică și semantică. Se subliniază că interpretarea cursurilor are loc prin intermediul codului intermediar generat, ceea ce sporește portabilitatea și flexibilitatea sistemului dezvoltat.

În cadrul analizei, este explicat modul în care sunt utilizate datele asociate lexemelor: în timp ce prima componentă *(tipul lexemului)* este folosită în procesul de analiză sintactică, cea de-a doua componentă *(datele asociate)* este utilizată la generarea codului pentru interpretare.

Procesul de generare a codului intermediar este o mapare a arborelui sintactic într-o secvență de instrucțiuni interne, care păstrează semantica programului sursă. Această abordare asigură separarea clară între nivelul de descriere declarativă a cursului și nivelul de execuție interpretativă.

Sunt evidențiate avantajele majore ale utilizării unei gramatici LL(1) și ale analizei predictive: simplitatea algoritmilor de analiză, detecția rapidă a erorilor de sintaxă și suportul eficient pentru implementarea interpretorului de cod intermediar.

Trebuie remarcat faptul că prin construcția unei gramatici adecvate, a analizoarelor lexicali și sintactici corespunzători, și a unui mecanism robust de generare a codului intermediar, se asigură o bază tehnologică solidă pentru dezvoltarea platformei de creare a cursurilor de instruire asistată de calculator. Această platformă, bazată pe limbajul MoCo, oferă atât flexibilitate în definirea scenariilor educaționale, cât și eficiență în execuția cursurilor pe diferite platforme hardware și software.

În capitol se analizează strategiile posibile de generare a codului pentru aplicațiile dezvoltate în limbajul MoCo;. se compară două opțiuni fundamentale *(generarea de cod executabil nativ versus generarea de cod intermediar interpretabil),* urmărindu-se evidențierea avantajelor, limitărilor și impactului fiecărei abordări asupra portabilității, flexibilității și întreținerii aplicațiilor.

Prima abordare analizată este *compilarea în cod executabil nativ*. Aceasta presupune traducerea directă a programelor MoCo în cod binar specific unei platforme țintă, oferind performanțe de execuție superioare și posibilitatea optimizării statice. Se subliniază că un astfel de compilator ar putea detecta erori înainte de execuție și ar proteja mai bine proprietatea

intelectuală. Totuși, se argumentează că dezavantajele sunt majore: lipsa portabilității, necesitatea menținerii multiplelor versiuni pentru diferite platforme, complexitatea dezvoltării unui compilator optimizator pentru un limbaj declarativ și dificultatea majoră de depanare în termeni de concepte educaționale.

În contrast, a doua opțiune, *generarea de cod intermediar și interpretarea acestuia,* este prezentată ca o soluție mai adecvată pentru MoCo. Codul intermediar, similar bytecode-ului din Java, permite rularea pe orice platformă pentru care există un interpretor, fără recompilare.

Se argumentează că interpretarea aduce beneficii esențiale în contextul unui limbaj declarativ: ușurința extinderii limbajului, capacitatea de adaptare rapidă la noi cerințe și posibilitatea unei depanări naturale, directe în termeni de modele educaționale..

Capitolul prezintă în detaliu și *problematica performanței*. Deși interpretarea introduce un cost suplimentar de procesare față de execuția nativă, acesta este considerat acceptabil în domeniul instruirii asistate, unde aplicațiile sunt predominant orientate pe interacțiuni și afișare de conținut, și nu pe calcule intensive.

În ceea ce privește *întreținerea și evoluția sistemului,* interpretarea codului intermediar este considerată net superioară. Se demonstrează că orice modificare în limbaj sau extindere a funcționalităților poate fi implementată rapid în interpretor, fără a necesita recompilarea cursurilor existente.

Capitolul evidențiază și *avantajele în domeniul depanării și validării cursurilor*. Prin interpretare, execuția poate fi urmărită la nivel de concepte MoCo, iar erorile pot fi raportate într-o formă direct accesibilă experților educaționali. Această transparență sprijină adoptarea limbajului MoCo de către un public larg de creatori de conținut educațional.

În concluzie, analiza comparativă conduce la o argumentare în favoarea *generării codului intermediar și interpretării* acestuia în cadrul limbajului MoCo. Se arată că această alegere oferă portabilitate maximă, flexibilitate sporită, un ciclu de dezvoltare mai scurt.

Platforma no-code pentru dezvoltarea cursurilor de instruire asistate de calculator.

Limbajul MoCo permite crearea cursurilor asistate de calculator specialiștilor-experți din domeniile de studii fără a recurge la ajutorul programatorilor. Totuși, programarea cere eforturi considerabile. Existența mediului de dezvoltare specializat al programelor favorizează crearea mai rapidă produselor program.

Problema poate fi rezolvată pe două căi: crearea unei platforme de dezvoltare cu programare redusă – LCDP (*eng*. Low-code Development Platform); crearea unei platforme de dezvoltare fără programare – NCDP (*eng.* No-code Development Platform).

No-code platformele permit crearea de aplicații software prin intermediul interfețelor grafice cu utilizatorul și al configurării aplicațiilor, în locul programării tradiționale pe calculator. Platformele de dezvoltare *no-code* sunt strâns legate de platformele de dezvoltare *low-code*, deoarece ambele sunt concepute pentru a accelera procesul de dezvoltare a aplicațiilor. Cu toate acestea, spre deosebire de *low-code*, platformele de dezvoltare *no-code* nu necesită deloc scrierea de cod, oferind în general șabloane predefinite cu ajutorul cărora întreprinderile și utilizatorii finali pot crea aplicații. În ultimii ani aceste platforme devin tot mai populare, dat fiind că permit crearea programelor de către utilizatori care nu sunt specialiști în programare, de exemplu, profesori din diferite domenii de educație.

Principala diferență între platformele de dezvoltare *low-code* și *no-code* este cantitatea de cunoștințe de programare necesară utilizatorului. Platformele de dezvoltare *low-code* necesită anumite cunoștințe de bază de programare pentru a dezvolta și integra aplicații complexe, în timp ce platformele de dezvoltare *no-code* nu necesită deloc abilitatea de a scrie coduri [8, 9].

Având în vedere că autorii cursurilor de instruire asistată de calculator sunt de cele mai multe ori specialiști în domeniul lor de activitate și nu sunt programatori, platformele NCDP sunt mai potrivite pentru ca aceștia să dezvolte cu succes cursuri asistate de calculator.

Vom prezenta un prototip al unei astfel de platforme NCDP, dezvoltat de autor pe baza limbajului MoCo.

Ideile principale. Succesul multor limbaje de programare și limbaje de descrieri se bazează pe diferiți factori. Unul dintre cei mai importanți factori prezintă posibilitatea creării rapide și comode a produsului final. Platformele NCDP de programare de obicei rezolvă problema construirii interfețelor utilizator și problema simplificării elaborări aplicațiilor. Aceasta se realizează prin înlocuirea metodei de scriere a programei cu metoda de construire a acestora. Platformele NCDP propun utilizarea programării vizuale. Aceasta presupune prezentarea elementelor constructive ale programei în mod intuitiv. A fost creată platforma NCDP care conține mediul vizual ce permite crearea cursurilor asistate de calculator fără cunoașterea limbajului MoCo.

Pe parcursul elaborării programului, platforma generează codul programului. În mediul MoCo se realizează conceptul de creare a cursului din situații instructive. Interfața situației instructive definește felul de comportare și modul de legătură cu alte situații în curs. Interfața constă din datele următoare:

- *Atributele* – datele la care au acces alte situații instructive. Aceste date reprezintă conținutul și stările situației. De exemplu, acestea pot fi textul întrebării, denumirea testului, denumirea unui fișier etc.

- *Evenimentele* – semnale transmise de la situația instructivă cu scopul semnalării schimbării stării sale, de exemplu, despre răspunsul incorect la o întrebare.
- *Acțiunile* – operații sau funcții ce pot fi executate de o situație. De exemplu, reacția la depășirea numărului de încercări de răspunsuri la întrebare sau oferirea ajutorului.

În *(Fig. 1 și 2)* este prezentată interfața mediului NCDP al platformei de creare a cursurilor de instruire asistate de calculator: crearea situației „*Information*" și *Crearea întrebărilor* pentru un test. Codul generat în urma acestui dialog a se vedea *(Fig. 3)*.

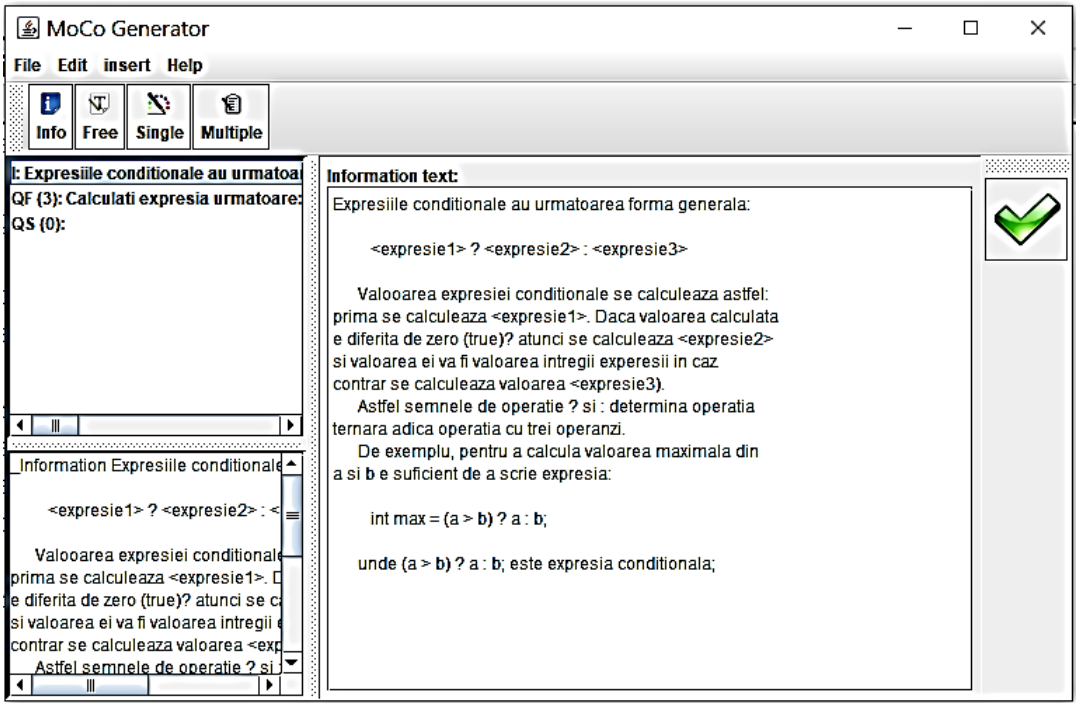


Figura 1. Interfața generatorului. Crearea situației „Information"

Utilizatorul poate să aleagă întrebările pentru un curs de diferite tipuri: *Free* – de tip deschis, *Single* – cu alegerea unui singur răspuns corect din mulțimea răspunsurilor, *Multiple* – selectarea mai multor răspunsuri corecte dintr-un set de răspunsuri. Rezultatele se prezintă în Figura. 2.

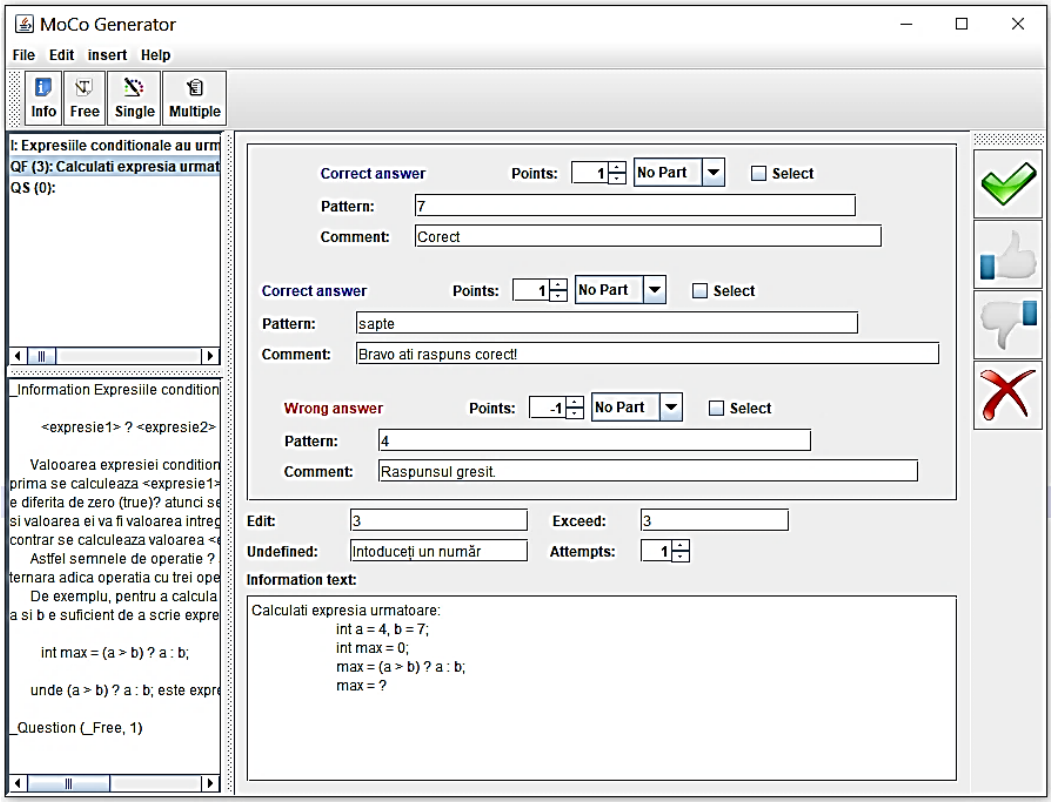


Figura 2. Interfața generatorului. Crearea întrebărilor pentru un test

În *(Fig. 3)* este prezentată o secvență de cod generată de platforma *no-code*.

```
_Information
        Expresiile conditionale au urmatoarea forma generala:
         <expresie1> ? <expresie2> : <expresie3>
...
      De exemplu, pentru a calcula valoarea maximala din
a si b e suficient de a scrie expresia:
         int max = (a > b) ? a : b;
         unde (a > b) ? a : b; este expresia conditionala;
_Question (_Free)
 Calculati expresia urmatoare:
      int a = 4, b = 7;
      int max = 0;
      max = (a > b) ? a : b;
      max = ?
$(_Correct, 1)7
$(_Reaction) Bravo, ati raspuns corect!
$(_Wrong, -1) 4
$(_Reaction) Raspunsul e grasit
$(_eXceed) Ati depasit numarul de incercari
_End_of_Question
_Question (_Single, 3) Calculati expresia
     10 % 3  (aveti trei incercari)
      $(_Wrong, 0) 3
      $(_Wrong, -1) 10
      $(_Reaction) Raspunsul e grasit
      $(_Correct, 1)1
      $(_Reaction) Bravo, ati raspuns corect la intrebarea
    $(_eXceed) Ati depasit numarul de incercari
    $(_Help) Incercati sa divizati numere
```

Figura 3. Exemplu de cod generat

Codul generat este trecut la intrarea interpretorului, care lucrează folosind structura de *frame*-uri a limbajului. În Figurile 4 și 5 sunt prezentate exemple de lucru al interpretorului programelor de instruire asistată de calculator.

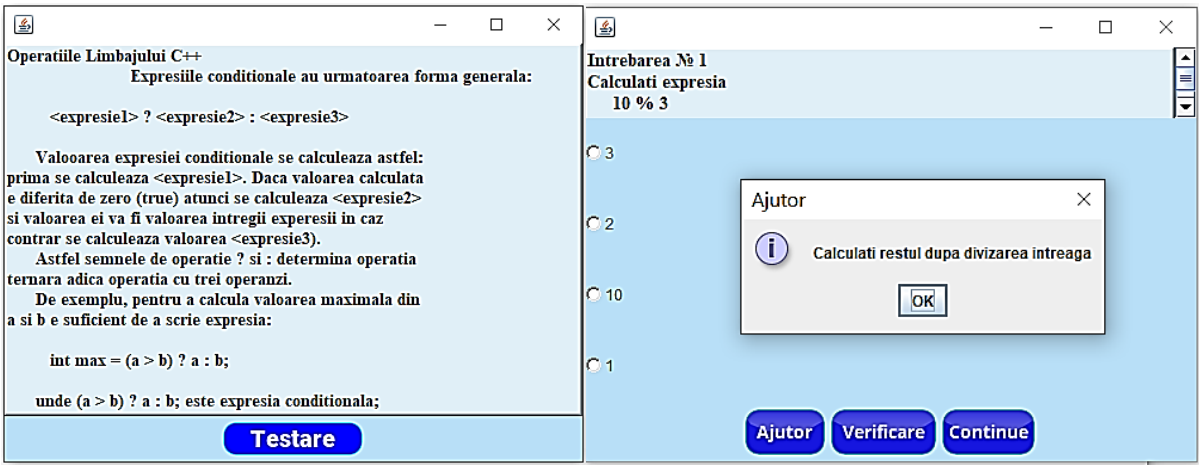


Figura 4. Afișarea informației și ajutorului interpretorului limbajului MoCo

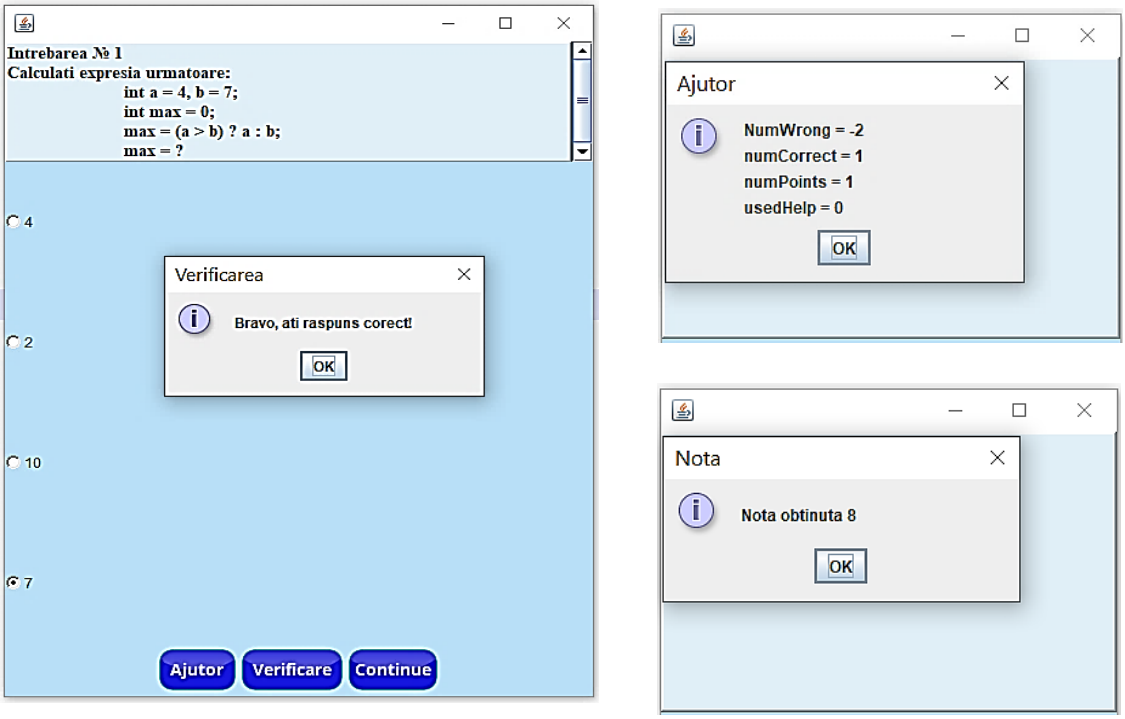


Figura 5. Verificarea răspunsului de către interpretorul limbajului MoCo: Nota și statistica

CONCLUZII GENERALE ȘI RECOMANDĂRI

Crearea instrumentelor pentru dezvoltarea cursurilor de instruire asistată de calculator împreună cu alte mijloace, cunoscute sub denumirea generică de e-Learning, constituie o direcție actuală și prioritară în cadrul domeniului dezvoltării moderne de software. Avansul rapid al tehnologiilor informaționale, împreună cu cerințele tot mai accentuate de digitalizare a proceselor educaționale, subliniază necesitatea elaborării unor soluții specializate, care să sprijine proiectarea, implementarea și livrarea eficientă a conținutului educațional interactiv.

În acest context, cercetarea desfășurată confirmă importanța dezvoltării unor tehnologii și limbaje specifice domeniului educațional, care să permită nu doar automatizarea procesului de creare a cursurilor, ci și adaptarea acestora la diverse scenarii educaționale și tehnologice. Rezultatele obținute reflectă contribuția lucrării la consolidarea unei baze metodologice și instrumentale în domeniul instruirii asistate de calculator.

Studiul rezultatelor obținute permite formularea următoarelor concluzii generale:

- Analiza instrumentelor existente pentru crearea cursurilor de instruire asistată de calculator, precum și a limbajelor de programare utilizate în cadrul acestor sisteme, evidențiază existența unui număr semnificativ de resurse și soluții tehnice disponibile. Cu toate acestea, pe fondul progresului continuu al tehnologiilor informaționale și al diversificării cerințelor, se conturează o nevoie tot mai accentuată pentru instrumente mai flexibile, mai accesibile și adaptate specificului domeniului educațional. În acest context, dezvoltarea de limbaje declarative specializate devine o direcție importantă pentru a susține proiectarea eficientă și personalizată a cursurilor de instruire asistată de calculator.
- Tehnologia dezvoltată pentru proiectarea programelor cursurilor de instruire asistată de calculator, care se bazează pe situații instructive stereotipe, a constituit un cadru metodologic solid pentru elaborarea unui limbaj de programare dedicat acestui domeniu. Această abordare a permis formalizarea structurii cursurilor în termeni pedagogici standardizați și a condus la definirea unor constructe specifice, relevante pentru descrierea logicii instruirii. Ca rezultat, a fost posibilă dezvoltarea unui mediu de programare destinat autorilor de cursuri, care facilitează definirea și implementarea scenariilor educaționale într-un mod eficient și intuitiv. Acest mediu oferă suport pentru generarea automată a conținutului, reducând complexitatea tehnică și contribuind la creșterea calității procesului de instruire asistată de calculator.
- În cadrul cercetării a fost elaborat limbajul de descriere a cursurilor de instruire asistată de calculator, denumit MoCo, care asigură o reprezentare adecvată,

structurată și expresivă a materialului didactic în format digital. Printre caracteristicile definitorii ale acestui limbaj se numără apropierea semantică maximă de situațiile reale din domeniul educațional, ceea ce facilitează utilizarea sa de către specialiștii în instruire. Totodată, MoCo se remarcă prin caracterul său intuitiv, care reduce bariera de acces tehnologic pentru autori, prin suportul pentru dezvoltarea cursurilor adaptive și prin aplicarea unor constrângeri minime asupra conținutului didactic. Aceste proprietăți contribuie la crearea de cursuri flexibile, accesibile și relevante din punct de vedere didactic.

- Descrierile cursurilor elaborate cu ajutorul limbajului MoCo constituie baze de cunoștințe, exprimate într-un limbaj orientat pe frame-uri, special conceput pentru reprezentarea structurilor specifice instruirii asistate de calculator. Limbajul MoCo permite modelarea logicii de instruire într-un mod expresiv și formalizat, facilitând astfel procesul de programare a cursurilor. Interpretorul MoCo acționează ca un motor inferențial, având rolul de a interpreta aceste baze de cunoștințe și de a genera comportamentul corespunzător al aplicației în baza regulilor și relațiilor definite în timpul descrierii cursului.
- Pe baza translatorului limbajului MoCo, a fost elaborat în premieră un prototip funcțional al unei platforme de tip *no-code* pentru dezvoltarea cursurilor de instruire asistată de calculator. Această platformă permite autorilor să creeze conținut educațional complex fără a apela la programare, facilitând astfel accesul larg la tehnologii educaționale moderne.

Direcțiile prioritare recomandate pentru viitoarele cercetări și dezvoltări în domeniul instruirii asistate de calculator ar putea include:

- Dezvoltarea viitoare a instrumentarului propus, mai concret elaborarea unei interfețe performante, atât pentru utilizator, cât și pentru autorul cursurilor (*eng*. authoring tools), care să asigure o interacțiune intuitivă, eficientă și adaptabilă cu sistemul. O astfel de interfață ar permite proiectarea ușoară a conținutului educațional, fără a necesita cunoștințe tehnice avansate, facilitând astfel implicarea directă a cadrelor didactice în procesul de creare a cursurilor de instruire asistată de calculator și sporind aplicabilitatea practică a soluției dezvoltate.
- O altă direcție importantă pentru dezvoltarea ulterioară constă în extinderea limbajului de programare propus cu mijloace specifice de integrare a facilităților multimedia în cadrul programelor cursurilor de instruire asistată de calculator.

Includerea nativă a suportului pentru elemente precum audio, video, animații și interacțiuni vizuale ar permite crearea unor materiale educaționale mai dinamice, atractive și eficiente din punct de vedere pedagogic, contribuind la sporirea gradului de implicare a cursanților și la îmbunătățirea procesului de învățare asistat de calculator.

- Pentru a îmbunătăți instrumentele de creare a cursurilor de instruire asistată de calculator, se recomandă integrarea facilităților oferite de sistemele de inteligență artificială, în special a modelelor lingvistice de mari dimensiuni (Large Language Model). Aceste modele pot asista autorii în generarea și personalizarea conținutului educațional, optimizând astfel procesul de dezvoltare a cursurilor. Implementarea se poate realiza prin utilizarea unor interfețe de programare a aplicațiilor (API) specializate, care permit integrarea eficientă a LLM-urilor în platformele de e-learning. Un exemplu relevant este utilizarea API-urilor oferite de platforme precum OpenAI, care permit dezvoltatorilor să încorporeze modele lingvistice avansate în aplicațiile lor. Aceste API-uri facilitează generarea automată de texte, răspunsuri la întrebări și alte funcționalități utile în context educațional.

BIBLIOGRAFIE

[1] GROS, B., GARCÍA-PEÑALVO, F.J. Future Trends in the Design Strategies and Technological Affordances of E-Learning. In: „*Learning, Design, and Technology*". Springer, Cham. 2016. pp. 1-23. Online ISBN: 978-3-319-17727-4 [citat 16.12.2024]; Accesibil la adresa: https://doi.org/10.1007/978-3-319-17727-4_67-1

[2] MIAO, Y., SODHI, T., BROUNS, F., SLOEP, P., KOPER, R. Bridging the Gap between Practitioners and E-Learning Standards: A Domain-Specific Modeling Approach. În: „*Technologies Across Learning Contexts"*. EC-TEL 2008. Lecture Notes in Computer Science, vol 5192. Springer, Berlin, Heidelberg. pp 284–289. ISBN978-3-540-87604-5. [citat 16.12.2024]; Accesibil la adresa: https://doi.org/10.1007/978-3-540-87605-2_32

[3] MICHAL, Śmiałek, NOWAKOWSKI, Wiktor. From Requirements to Java in a Snap: Model-Driven Requirements Engineering in Practice. Springer.2015. 602 p. ISBN-13: 978-3319128382.

[4] BRAMBILLA, Marco, CABOT, Jordi, WIMMER, Manuel. Model-Driven Software Engineering in Practice (Synthesis Lectures on Software Engineering). Morgan & Claypool. 2017. 191 p. ISBN-10: 1627057080. ISBN-13: 978-1627057080.

[5] JEFFERY, Clinton, L. Build Your Own Programming Language: A programmer's guide to designing compilers, interpreters, and DSLs for modern computing problems. Packt Publishing. 2024. 556 p. ISBN-10: 1804618020. ISBN-13: 978-1804618028

[6] LUCAS, Oliver, JR. Domain-Specific Languages Mastery : The Power Of Custom Language. Kindle Edition. 2024. 84 p. ISBN: B0DL3CSWWK

[7] WĄSOWSKI Andrzej, BERGER, Thorsten. Domain-Specific Languages: Effective Modeling, Automation, and Reuse. Springer.2024 . 504 p. ISBN-10: 3031236688. ISBN-13: 978-3031236686.

[8] CABALLAR, Rina Diane. Programming Without Code: The Rise of No-Code Software Development. IEEE Spectrum magazine. 11 March 2020. [citat 1.10.2024]; Accesibil la adresa:https://spectrum.ieee.org/programming-without-code-no-code-software-development

[9] ROKIS, K., KIRIKOVA, M. Challenges of Low-Code/No-Code Software Development: A Literature Review. In: Nazaruka, Ē., Sandkuhl, K., Seigerroth, U. (eds) Perspectives in Business Informatics Research - BIR 2022. Lecture Notes in Business Information Processing, vol 462. Springer, Cham. pp. 3-17. [citat 15.11.2024]; Accesibil la adresa: https://doi.org/10.1007/978-3-031-16947-2_1

PUBLICAȚIILE AUTORULUI LA TEMA TEZEI

[10] LATUL, GH. The media for automatic design of computer assisted courses. În: *Computer Technologies in Education*. *Proceedings of the International Conference on Computer Technologies in Education. Part 2. (ICCTE'94).* Ucraina, Crimea, 1994, pp. 159-160.

[11] ЛАТУЛ, Г.В. Интерпретация обучающих курсов в системе автоматизированной генерации программ. În: *Materialele conferinței corpului didactico-științific. Bilanțul activității științifice a U.S.M. pe anii 1993-1994. științe naturale*. Chișinău. Ch.: Centrul ed. al USM, 1995, p. 60.

[12] ЛАТУЛ, Г. Модели обучения для компьютерных учебных курсов În: *Conf. corpului didactico-științific Bilanțul activității științifice a USM pe anii* 1996/97.

Rezumatele comunicărilor. Ştiinţe naturale. Chişinău. Ch.: Centrul ed. al USM, 1998. pp. 70.

[13] LATUL, GH. Modelul aprecierii cunoştinţelor în cursuri asistate de calculator. În: *Anale ştiinţifice ale Universităţii de Stat din Moldova. Seria „ştiinţe fizico-matematice”*. Chişinău. Ch.: Centrul ed. al USM, 1998, pp. 36-41. ISBN 9975-917-04-6

[14] ЛАТУЛ, Г. Модели урока при проектировании компьютерных учебных курсов. În: *Anale ştiinţifice ale Universităţii de Stat din Moldova. Seria „ştiinţe fizico-matematice”*. Chişinău. Ch.: Centrul ed. al USM, 1999, pp. 252-256. ISBN 9975-917-26-7

[15] ЛАТУЛ, Г. Инструментарий для создания компьютерных учебных курсов. În: *Tehnologii avansate în pragul secolului XXI. Materialele conferinţei ştiinţifico-practice*. Chişinău, Î.E.P. Ştiinţa, 2000, pp. 22-24.

[16] LATUL, GH., CARELOVA, L. Limbajul de descriere a situaţiilor instructive şi analizorul lui. În: *Conferinţa corpului didactico-ştiinţific. Bilanţul activităţii ştiinţifice a USM pe anii 1998/99. Rezumatele comunicărilor. Ştiinţe fizico-matematice*. Chişinău. Ch.: Centrul ed. al USM, 2000, pp. 117-118.

[17] LATUL, GH. Generarea paginilor Web după descrierea scenariilor de instruire. În: *Conferinţa corpului didactico-ştiinţific. Bilanţul activităţii ştiinţifice a USM pe anii 1998/99. Rezumatele comunicărilor. Ştiinţe fizico-matematice*. Chişinău. Ch.: Centrul ed. al USM, 2000, pp 119-120.

[18] ЛАТУЛ, Г. В. Использование шаблонов при создании компьютерных учебных курсов. În: *Conferinţa corpului didactico-ştiinţific. Bilanţul activităţii ştiinţifice a USM în anii 2000 2003. Rezumatele comunicărilor. Ştiinţe fizico-matematice*. Chişinău. Ch.: Centrul ed. al USM, 2003, pp. 188-189.

[19] ЛАТУЛ, Г. Проблемы организации анализа ответов обучаемых в компьютерных учебных курсах. În: *Analele ştiinţifice ale Universităţii de Stat din Moldova. Seria „Ştiinţe fizico-matematice”*. Chişinău. Ch.: Centrul ed. al USM, 2004. pp. 132-136. ISSN 1811-2641

[20] ЛАТУЛ, Г. Язык описания компьютерных учебных курсов. În: *Analele ştiinţifice ale Universităţii de Stat din Moldova. Seria „Ştiinţe fizico-matematice”*. Chişinău. Ch.: Centrul ed. al USM, 2004.pp. 129-131. ISSN 1811-2641

[21] ЛАТУЛ, Г. Проблемы проектирования компьютерных учебных курсов при помощи шаблонов. În: *Conferinţa Internaţională Tendinţele de Dezvoltare a Societăţii Informaţionale Academiei de Studii Economice*. Chişinău, 2004. pp. 180-181. ISBN 9975-75-265-9

[22] LATUL, GH. Perspectivele instruirii la distanţă În: Republica Moldova. În: *Învăţământul superior şi cercetarea – piloni ai societăţii bazate pe cunoaştere. Conferinţa ştiinţifică internaţională. Rezumatele comunicărilor Ştiinţe Reale*. Chişinău. Ch.: Centrul ed. al USM, 2006. pp. 53-54.

[23] BRAGARU, T., LATUL, GH. Dezvoltarea cursurilor de instruire la distanţă. In: *Proceedings of the International Conferences „Mathematics & Information Technologies: Research and Education” (MITRE 2008 – 2009.* Chişinău, Ch.: Centrul ed. al USM, 2010, pp. 149 -154, ISBN 978-9975-70-995-8

[24] CĂPĂŢÂNĂ, GH., LATUL, GH. *Sisteme inteligente de e-learning. In Mathematics & Information Technologies: Research and Education (MITRE 2011)*. Abstracts. Chişinău, Ch.: Centrul ed. al USM, 2011, pp. 157 -158, ISBN 978-9975-71-144-9.

[25] LATUL, GH. Learning material presentation in computer assisted courses interpretation with XML aid. In *Mathematics & Information Technologies: Research and Education (MITRE-2013). International Conference.* Abstracts. Chișinău, Ch.: Centrul ed. al USM, 2013, pp. 110-111, ISBN 978-9975-71-411-2.

[26] ЛАТУЛ, Г. Построение генератора компьютерных учебных курсов. În: *Integrare prin cercetare și inovare, conferință științifică națională cu participare internațională Rezumatele ale comunicărilor. Științe ale naturii. Științe exacte*. Chișinău, Ch.: Centrul ed. al USM, 2014, pp. 150 -151. ISBN 978-9975-71-573-7.

[27] CĂPĂȚÂNĂ, GH., LATUL, GH., PERETEATCU, S. Analiza răspunsurilor grafice în cursuri de instruire asistate de calculator. În: *International Conference „Mathematics & Information Technologies: Research and Education (MITRE - 2015)*. Chișinău, Ch.: Centrul ed. al USM, 2015, pp. 109, ISBN 978-9975-71-678-9.

[28] CĂPĂȚÂNĂ, GH., LATUL GH. Frame using in the language for computer assisted courses programming. În: *Mathematics & Information Technologies: Research and Education (MITRE 2016). International conference.* Abstracts. Chișinău, Ch.: Centrul ed. al USM, 2016, pp. 77, ISBN 978-9975-71-794-6.

[29] LATUL, GHEORGHE, BRAGARU TUDOR, CROITOR MIHAIL. Objective items development system. In *Mathematics & Information Technologies: Research and Education" (MITRE 2019). International Conference*. Chișinău, Ch.: Centrul ed. al USM, 2019. 104p, P. 77-78, ISBN 978-9975-149-17-4.

[30] LATUL, GH. Model of authoring system of computer assisted courses generator. In *Mathematics & Information Technologies: Research and Education (MITRE 2023). International conference*. Chișinău. Ch.: Centrul ed. al USM, 2023, pp. 81-82, ISBN 978-9975-62-535-7

[31] LATUL, GH. Reprezentarea cunoștințelor în cursurile de instruire asistată de calculator. În: *Acta et commentationes, seria Științe ale Educației*, nr. 1(35), 2024, p. 103-108. ISSN 1857-062, E-ISSN 2587-3636, Tip B. [citat 16.01.2025]; Accesibil la adresa: https://revistaust.upsc.md/index.php/acta_educatie/article/view/979/958

[32] LATUL, GH. Modelul conceptual și arhitectura platformei no-code de generare automată a cursurilor de instruire asistată de calculator. În: *Intellectus*, nr. 1, 2024, p. 147 – 156. ISSN 1810-7087, Tip B. [citat 16.01.2025]; Accesibil la adresa: https://agepi.gov.md/ro/intellectus/intellectus-1-2024/modelul-conceptual-%C8%99i-arhitectura-platformei-no-code-de-generare

ADNOTARE

Gheorghe Latul. „Elaborarea metodelor și mijloacelor pentru crearea cursurilor de instruire asistate de calculator”

Teză de doctor în informatică, Chișinău, 2025.

Structura tezei: Teza a fost elaborată în Universitatea de Stat din Moldova, este scrisă în limba română și constă din introducere, patru capitole, concluzii generale și recomandări. Teza conține 115 pagini text de bază, 17 figuri și un tabel, bibliografie din 113 titluri și 8 anexe. Rezultatele obținute sunt publicate în 23 lucrări științifice.

Cuvinte cheie: instruire, instruire asistată de calculator, e-Learning, no-code, low-code, frame, scenariul, limbajul de programare, interpretorul limbajului, tehnologii de proiectare a softului.

Scopul cercetării: crearea tehnologiei originale și instrumentelor informatice pentru proiectarea și programarea cursurilor de instruire asistată de calculator.

Obiectivele cercetării: elaborarea metodelor și mijloacelor de programare a cursurilor de instruire asistată de calculator, formularea cerințelor principale față de sistemele de instruire asistate de calculator; formularea cerințelor față de limbajul descrierii cursurilor asistate de calculator, elaborarea limbajului formal de descriere a cursurilor asistate de calculator; elaborarea compilatorului și prototipului interpretorului limbajului elaborat.

Noutatea și originalitatea științifică a lucrării constă în noi abordări ale proiectării programelor de cursuri de instruire asistată de calculator bazate pe situații stereotipice de instruire, conceptul cărora este formulat în teză. A fost dezvoltat un limbaj de programare declarativ specific domeniului, destinat pentru crearea cursurilor de instruire asistată de calculator bazat pe scenarii cauzale, care permite păstrarea cunoștințelor în frame-uri.

Problema științifică importantă rezolvată în domeniul de cercetare constă în dezvoltarea tehnologiei de proiectare a programelor de instruire asistată de calculator bazată pe descrierea situațiilor instructive stereotipice, definirea scenariilor instructive în baza scenariilor cauzale, elaborarea limbajului de programare specific domeniului de creare a cursurilor de instruire asistate de calculator.

Rezultatul obținut, care contribuie la soluționarea unei probleme științifice importante constă în prezentarea cunoștințelor în cadrul cursurilor de instruire asistată de calculator cu ajutorul scenariilor cauzale – unei varietăți de frame-uri.

Semnificația teoretică constă în elaborarea unei tehnologii inteligente, orientate *frame*, de proiectare a cursurilor de instruire asistată de calculator în baza *scenariilor cauzale*.

Valoarea aplicativă a lucrării: constă în elaborarea unui limbaj original de descriere a cursurilor de instruire asistată de calculator MoCo, a translatorului acestui limbaj și a unei tehnologii informaționale inteligente de elaborare efectivă a cursurilor de instruire asistate de calculator.

Implementarea rezultatelor lucrării. Tehnologii de proiectare a cursurilor asistate de calculator, limbajul MoCo și interpretorul lui elaborate în teza a fost utilizate la crearea cursului de instruire asistat de calculator „Programarea în limbajul C”. Cursul a fost implementat în procesul de studii la Facultatea de Matematică și Informatică la disciplina Fundamentele de Programare.

ABSTRACT

Gheorghe Latul: „Methods and technique development of computer assisted courses creation”

Doctoral thesis in Computer Science, Chisinau, 2025.

Structure of the thesis: the work on the thesis was carried out at the State University of Moldova, it is written in Romanian and consists of an introduction, 4 chapters, general conclusions and recommendations, a bibliography of 113 titles, as well as eight appendices. The dissertation contains 115 pages of main text, 17 figures and 1 table. The main results of the thesis have been published in 23 scientific papers.

Keywords: instruction, computer assisted instruction, e-Learning, no-code, low-code, scenario, programming language, language interpreter, software design technologies.

The purpose of the research: to create original technology and IT tools for the design and programming of computer-assisted instruction courses.

Research objectives: development of methods and tools for programming computer-assisted courses, formulation of the main requirements for to computer-assisted instructional systems; formulation of requirements for the language of the language of computer-assisted course description, development of a formal language for describing computer-assisted courses; development of a compiler and a prototype of the interpreter of the language under development.

Scientific novelty and originality of the thesis consists in new approaches to designing programmes of computer assisted instruction courses based on stereotypical instructional situations, the concept which is formulated in the thesis. A domain specific declarative programming language for computer assisted instruction courses based on causal scenarios has been developed, it makes possible to store domain knowledge in frames.

The obtained result, which contributes to the solution of an important scientific problem, consists in the representation of knowledge in computer assisted instruction courses by means of causal scenarios – a type of frames.

The theoretical significance consists in the development of an intelligent, *frame-oriented* technology for designing computer-assisted instruction courses based on *causal scenarios*.

The applied value of the work: consists in the development of an original language MoCo for describing MoCo computer-assisted training courses, the translator of this language and an intelligent information technology for the effective development of computer-assisted training courses.

Implementation of the results of the work. Computer-aided course design technologies, the MoCo language and its interpreter developed in the thesis were used in the creation of the computer-aided training course "Programming in C language". The course was implemented in the study process at the Faculty of Mathematics and Computer Science in the discipline Fundamentals of Programming.

АННОТАЦИЯ

Георге Латул: „Разработка методов и средств создания компьютерных учебных курсов”.

Докторская диссертация по информатике, Кишинёв, 2025.

Структура диссертации: работа над диссертацией велась в Государственном Университете Молдовы, написана на румынском языке и состоит из введения, 4 глав, общих выводов и рекомендаций, библиографии из 113 наименований, а также восьми приложений. Диссертация содержит 115 страниц основного текста, 17 иллюстраций и 1 таблицу. Основные результаты диссертации опубликованы в 23 научных работах.

Ключевые слова: обучение, автоматизированное обучение, *e-Learning, no-code, low-code*, фрейм, сценарий, язык программирования, интерпретатор языка, технологии проектирования программного обеспечения.

Цель исследования: создание оригинальной технологии и ИТ-инструментов для проектирования и программирования компьютерных учебных курсов.

Задачи исследования: разработка методов и средств программирования автоматизированных учебных курсов, формулировка основных требований к автоматизированным обучающим системам; формулирование требований к языку описания компьютерных курсов, разработка формального языка описания компьютерных курсов; разработка компилятора и прототипа интерпретатора разрабатываемого языка.

Научная новизна и оригинальность работы состоит в новых подходах к проектированию программ компьютерных учебных курсов, основанных на стереотипных учебных ситуациях, понятие о которых сформулировано в диссертации. Разработан предметно ориентированный декларативный язык программирования компьютерных учебных курсов на основе каузальных сценариев, позволяющий хранить знания предметной области в фреймах.

Полученный результат, который вносит вклад в решение важной научной проблемы состоит в представлении знаний в компьютерных обучающих курсах с помощью каузальных сценариев – разновидности фреймов**.**

Теоретическая значимость заключается в разработке интеллектуальной, фрейм-ориентированной технологии для проектирования автоматизированных учебных курсов на основе каузальных сценариев.

Практическое значение работы: заключается в разработке оригинального языка описания компьютерных учебных курсов MoCo, транслятора этого языка и интеллектуальной информационной технологии для эффективной разработки компьютерных учебных курсов.

Внедрение результатов работы. Технологии проектирования компьютерных курсов, язык MoCo и его интерпретатор, разработанные в данной работе, были использованы при создании автоматизированного учебного курса «Программирование на языке Си». Курс был внедрен в учебный процесс на факультете Математики и Информатики по дисциплине Основы программирования**.**

LATUL GHEORGHE

ELABORAREA METODELOR ȘI MIJLOACELOR PENTRU CREAREA CURSURILOR DE INSTRUIRE ASISTATE DE CALCULATOR

121.03 – PROGRAMAREA CALCULATOARELOR

Rezumatul tezei de doctor în informatică

Aprobat pentru tipar: 5.05.20255	Formatul hârtiei 60x84 1/16
Hârtie ofset. Tipar ofset.	Tirajul 30 ex.
Coli de tipar: 2,0	Comanda nr. 74

CEP Universitatea de Stat din Moldova, str. A. Mateevici 60, Chișinău, MD-2009, Republica Moldova