Sinteza etapei a II-a a contractului de cercetare ştiinţifică 1223,
"Studii avansate de elasticitate aplicată din perspectivă multidisciplinară, asistate de calculator"

Obiectivul 6 – "Stabilirea tehnologiilor informatice care vor fi folosite pentru implementări: limbaje de programare, biblioteci de programe, editoare specializate, etc."
Activităţi asociate – "Pe baza facilităţilor oferite de limbajele de programare propuse (domenii de utilizare, posibilitatea de a crea aplicaţii cross-platform, etc), se va stabili o ierarhie a acestora. În final se vor alege maxim două limbaje de bază care vor fi folosite pentru integrarea aplicaţiilor specializate."

Au fost evaluate mai multe limbaje de programare. S-a urmărit ca limbajul ales să poată fi folosit pentru o serie de cerinţe complexe, cum ar fi: programare numerică, aplicaţii grafice, interfaţă prietenoasă cu utilizatorul, comunicaţii, aplicaţii cross-platform, etc. Limbajele de programare şi mediile de dezvoltare studiate sunt: Fortran, Pascal, VisualBasic, Visual FoxPro, Lisp, C, C++, Java, ECMAscript, etc.
Datorită 'ofertei' generoase de limbaje de programare, am decis să studiem soluţiile posibile din perspectiva limbajelor clasice de programare.
Au fost identificate funcţionalităţile aplicaţiei finale şi au fost asignate fiecărei funcţionalităţi limbajele de programare compatibile cu aceste constrângeri.
S-a urmărit imaginarea unui mod de interfaţare a ramurilor de program scrise în diferite limbaje de programare, fiind identificate 3 aspecte:
1. folosirea unui mediu de programare/dezvoltare din care sunt lansate aplicaţii scrise în limbaje diferite pe baza unei semaforizări;
2. folosirea sistemului de operare ca nivel de bază în lansarea diferitelor aplicaţii scrise în limbaje diferite, cu utilizarea fişierelor lot de comenzi;
3. folosirea unor limbaje care să poată îngloba codul obiect rezultat din compilarea ramurilor de program scrise în diferite limbaje de programare;
4. folosirea unui limbaj de programare care să conţină biblioteci pentru implementarea tuturor funcţionalităţilor dorite, aplicaţiile respectiv ramurile de program urmând a fi scrise în acest limbaj.
Au fost evidenţiate avantajele şi dezavantajele fiecărei soluţii strategice.

Obiectivul 7 – "Documentare referitoare la limbajele de programare alese."
Activităţi asociate – "Sunt identificate documentaţii de ultimă oră iar titularii de problemă susţin cursuri / prezentări cu informaţiile esenţiale necesare începerii activităţii de implementare. Este de dorit ca prezentările să aibe conotaţii practice (gen laborator) pentru ca noţiunile predate să poată fi aplicate imediat. Vor fi prezentate mediile de editare / dezvoltare, probleme efective, utilizarea unor programe utilitare. Sunt prezentate probleme nerezolvate, chestiunile care necesită documentare pentru a fi în atenţia tuturor membrilor echipei."

Membrii echipei au lucrat frenetic pentru identificarea tuturor aspectelor care pot prezenta avantaje sau dezavantaje în dezvoltarea ulterioară a proiectului.
Sunt avute în vedere atât aspectele legate de:
1. funcţionalităţi specifice;
2. flexibilitate în arhitecturarea aplicaţiilor;
3. interfaţare;
4. portabilitate cross-platform;
5. posibilitatea de a testa şi ulterior de a îndeplini toate obiectivele autoimpuse.
Dat fiind caracterul explorator al proiectului, ca măsură de siguranţă, sunt alese două direcţii de dezvoltare. Prima direcţie reprezintă un nivel fundamental şi se referă la alegerea unor limbaje cunoscute de programare în care membrii echipei de

cercetare au experiența necesară în dezvoltarea aplicațiilor specifice. Cea de a doua direcție reprezintă o direcție fundamental inovativă în care membrii echipei lucrează cu riscurile specifice muncii de cercetare. Experiența din prima direcție de dezvoltare va fi folosită pentru realizarea obiectivelor folosind cea de a doua direcție strategică.
Sunt identificate principalele funcționalități (aplicații utilitare, editoare, atribute specifice) ale instrumentelor software care vor fi folosite. Se pune problema rațională a asimilării acestor funcționalități prin dezvoltarea de module originale specifice, într-un limbaj de programare, sau folosirea din exterior a utilitarelor existente.

Obiectivul 8 – "Inițiere a implementării de module software care folosesc algoritmii generali şi metodele numerice identificate anterior."
Activități asociate – "Este începută etapa de dezvoltare de aplicații software prin care sunt realizate module (subprograme, proceduri/funcții, obiecte, clase) care implemetează algoritmii generali sau cei specifici metodelor numerice anterior identificate. Activitatea este începută prin conceperea unui sistem de tratare a erorilor care are în vedere următoarele aspecte: prevederea unor tipuri de erori care pot apare (numitori zero, expresii negative sub radical, etc), asocierea unor coduri de eroare, stabilirea unui mecanism de semnalare a erorii pe baza unui flag, stabilirea unui mod unitar de prezentare a erorii generale dar şi a unui mesaj de eroare specific aplicației de nivel superior care foloseşte bibliotecile de bază, etc. Fiecare modul realizat este verificat cu o serie de programe de test iar programul sursă este comentat în detaliu. Modulele testate împreună cu programele de test sunt introduse într-un manual de metode numerice. Primele rezultate sunt predate pe data de 04 Aprilie 2008 şi ar fi de dorit să se refere la tratarea erorilor."

Ca măsură de siguranță pentru prevenirea erorilor de execuție care sunt dificil de depanat, se doreşte proiectarea unui cadru general de tratare a erorilor. Astfel, sunt identificate principalele tipuri de erori fatale care pot apare şi sunt dezvoltate studii privind erorile din programarea numerică. Aceste studii vor fi folosite în etapele următoare pentru identificarea gradului de precizie dorit şi corelarea acestuia cu tipurile de variabile curent folosire şi cu reprezentarea acestora ca cifre semnificative şi zecimale în fişierele text folosite pentru eventuale transferuri de date.
Odată identificate categoriile generale de erori care pot conduce la blocarea aplicației, sunt imaginate situațiile în care pot apare şi rezultă faptul că sunt necesare:
1. identificarea erorii;
2. identificarea instrucțiunii care o generează;
3. identificarea contextului în care apare această eroare.
Având în minte aceste obiective, sunt efectuate următoarele:
1. sunt catalogate erorile ca tip şi li se atribuie un cod de eroare şi un mesaj general de eroare;
2. sunt elaborate metodologii de codare a aplicației care să conțină verificări pentru toate sursele de erori fatale;
3. sunt identificate metodele flexibile prin care analistul poate trasa execuția şi poate releva contextul apariției unei erori fatale;
4. sunt identificate nivele de ierarhizare în trasarea execuției prin activarea sau dezactivarea unui parametru flag implicit care are o vizibilitate dată de context sau folosirea unor parametri flag specifici la nivel de aplicație, bibliotecă, modul, secvență, respectiv context efectiv. Această ultimă metodă prevede şi posibilitatea de a relaționa aceşti parametri flag.
Sunt proiectate primele module de program.

Politica de asigurare a unui nivel înalt de încredere în aplicațiile deja scrise impune dezvoltarea de scurte aplicații de testare care să verifice comportamentul modulelor generale pe toate ramurile.
Definirea acestor module trebuie făcută într-un mod cât mai flexibil, care să permită adăugarea ulterioară de noi erori şi tratarea specifică a acestora.
Sunt identificate facilitățile oferite de limbajele moderne de programare care rezolvă deja o serie de probleme, însă nu respectă cadrul strategic general definit anterior. Sunt căutate şi găsite soluții specifice, care ar trebui să trateze unitar această problemă. Datorită diferențelor între conceptele de bază a limbajelor de programare avute în vedere, acest lucru este posibil numai parțial, fapt care impune identificarea cadrului problematicii comune.

Obiectivul 9 – "Proiectare structuri de date care vor fi folosite ca platformă de bază în dezvoltarea aplicațiilor de metode numerice şi grafică."
Activități asociate – "Sunt analizate structurile de date care pot fi definite în limbajele de programare alese. Obiectivul este de a defini o structură de date care să ofere viteză maximă de procesare, folosire maximă a spațiului RAM, să aibă flexibilitate, deci să poată fi adaptată la condiții noi, să poată fi folosită ca interfață între aplicațiile scrise. Este inițiată analiza pe baza căreia va fi creată o bază de date SQL care să folosească tabele care au structura identică cu cea a stucturilor de date definite. Este proiectată efectiv baza de date folosind o tehnologie de tip SQL Server."

Problema structurilor de date este fundamentală pentru dezvoltarea ulterioară a aplicației. Astfel, în fiecare limbaj de programare sunt imaginate modalitățile optime de rulare a aplicației din perspectiva următoarelor criterii:
1. ***folosirea unui volum maxim de date;***
2. ***atingerea unei viteze maxime de execuție;***
3. ***oferirea unui nivel de control maxim, de dorit nemijlocit, al informațiilor în timpul executării aplicației;***
4. ***oferirea de flexibilitate maximă la nivel matematic;***
5. ***oferirea de flexibilitate maximă la nivelul proiectării algoritmilor;***
6. ***flexibilitate maximă în dezvoltarea aplicației, care să permită, de exemplu, modificări la nivelul cel mai de sus prin modificări minime la nivelul de bază;***
7. ***flexibilitate maximă la nivelul folosirii resurselor sistemului gazdă (RAM/HDD);***
8. ***posibilitatea de interfațare optimă a aplicației.***

Dintre structurile de date folosite este aleasă structura optimă şi se reține principiile de bază ale acesteia cu scopul de a o putea implementa, cel puțin parțial la nivelul funcționalităților esențiale, în limbajele de programare folosite.
Sunt implementate efectiv structurile de date şi biblioteca de bază.
Sunt imaginate o serie de optimizări care sunt de asemenea implementate.
Mai mult, este decisă implementarea unui editor specializat care să permită controlul datelor chiar în formatul folosit pentru implementări ulterioare.
Pe baza structurii de date anterior proiectate este dedusă structura tabelei de bază, care va conține informațiile portabile între modulele scrise în limbaje diferite. Această tabelă este creată într-un sistem de gestiune a bazelor de date folosind instrucțiuni SQL. Operarea şi interogarea tabelei se realizează tot cu instrucțiuni SQL, acestea urmând a fi folosite în tehnologia SQL Server care urmează a fi dezvoltată odată cu achiziția server-ului.

Obiectivul 10 – "Folosind structurile de date proiectate sunt implementați şi testați algoritmii pentru metode numerice şi pentru primitivele grafice de bază."
Activități asociate – "Pe baza structurilor de date proiectare sunt implementate module software care folosesc algoritmii generali şi metodele numerice identificate anterior, similar activității 8. Astfel, după ce este implementat şi testat un algoritm în cadrul activității 8,

acesta este implementat şi testat folosind structura de date proiectată în cadrul activității 9. Rezultatele testelor trebuie să fie apropiate de cele ale programelor 'de firmă'."

Odată proiectate structurile de date, acestea au fost implementate într-o serie de biblioteci de programe care au rolul de a permite accesul optim la informații.
Aceste biblioteci sunt folosite în etapa actuală ca fundament pentru dezvoltarea unor biblioteci de complexitate superioară care să rezolve o serie de probleme concrete legate de metodele numerice care urmează a fi folosite în aplicațiile dedicate domeniilor specifice.
Astfel, sunt proiectate şi implementate module de program pentru rezolvarea sistemelor de ecuații pentru domenii particulare (MEF), sisteme de ecuații diferențiale, etc.
Apar o serie de aspecte care necesită ajustări ale tipurilor de bază, respectiv ale bibliotecilor de nivel inferior, fapt care este realizat cu uşurință datorită flexibilității asigurate în etapele anterioare.
Sunt dorite o serie de optimizări ale timpului de acces prin folosirea optimă a memoriei.
Pentru a verifica aceste module de program sunt create o serie de programe de testare cu ajutorul cărora sunt parcurse toate ramurile modulelor, inclusiv cele care trebuie să genereze mesaje de eroare.
Rezultatele acestor programe de testare sunt comparate cu rezultatele rulării aplicațiilor de firmă. Se constată cvasiidentitatea acestora.
Sunt identificate ca extrem de folosite reprezentările grafice de tip grid care pot fi întâlnite şi într-o serie de aplicații 'de firmă'. Se analizează oportunitatea implementării acestui tip de grafică şi se decide folosirea aplicațiilor externe care sunt suficient de deschise pentru a accepta date din fişiere text în diferite formate (CSV, SDF, etc), fişiere care pot fi create în aplicațiile deja scrise. În plus, aceste aplicații externe pot fi folosite în diferite sisteme de operare, fapt care asigură funcționalitatea cross-platform, în conformitate cu abordarea generală.

Obiectivul 11 – "Folosind structurile de date proiectate şi algoritmii deja implementați, sunt concepute, proiectate, implementate şi testate aplicații dedicate fiecărui domeniu de specializare."
Activități asociate – "Pe baza structurilor de date proiectate şi a algoritmilor care folosesc aceste structuri şi care sunt implementați, sunt dezvoltate aplicații din fiecare domeniu de specializare. Primele aplicații sunt simple, gradul de dificultate crescând cu fiecare studiu de caz care se dezvoltă. Rezultatele aplicațiilor originale sunt verificate cu ajutorul programelor "de firmă". Testele trebuie să conducă la rezultate convergente."

Activitatea pentru acest obiectiv este în desfăşurare, finalizarea fiind prognozată în jurul datei de 06 Martie 2009.
În perioada alocată acestei faze au fost rezolvate următoarele probleme:
- ***analiză probleme pe specialități;***
- ***documentare privind ultimele tendințe în domeniu;***
- ***dezvoltare biblioteci de bază;***
- ***inițiere a primelor module de program din fiecare specialitate în parte.***

Concluzii

Etapa a doua a proiectului de cercetare ştiințifică "Studii avansate de elasticitate aplicată din perspectivă multidisciplinară, asistate de calculator" pune bazele dezvoltării de instrumente software pentru toate fazele următoare.

Am urmărit ca arhitectura filozofiei acestei abordări să fie cât mai flexibilă, pentru a putea modifica cât mai uşor bibliotecile de bază, cu reutilizarea aplicațiilor de nivel superior fără transformări majore.
Figura de mai jos prezintă structura ierarhică a bibliotecilor dezvoltate până în acest moment.

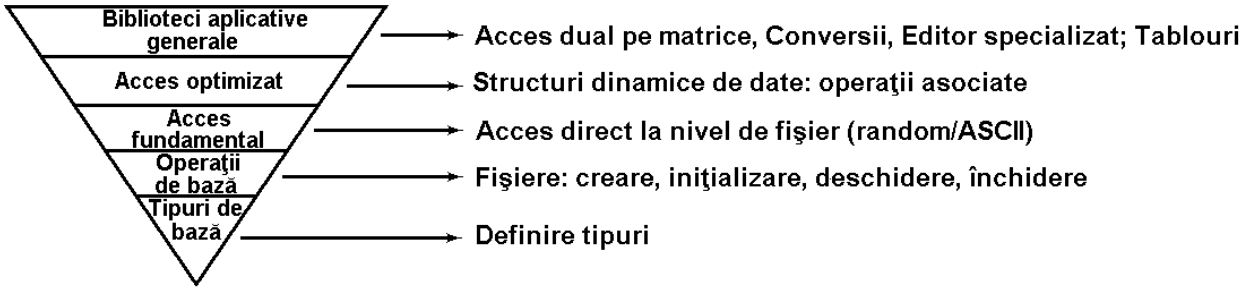

Figura 1 – Ierarhia bibliotecilor generale

Ideea fundamentală este de a trata unitar aplicațiile scrise în limbaje de programare diferite. Ca atare, a fost adoptată o politică de transfer de informații bazată pe folosirea unei interfețe constituită dintr-un fişier ASCII cu structură fixă. Figura de mai jos ilustrează această idee.

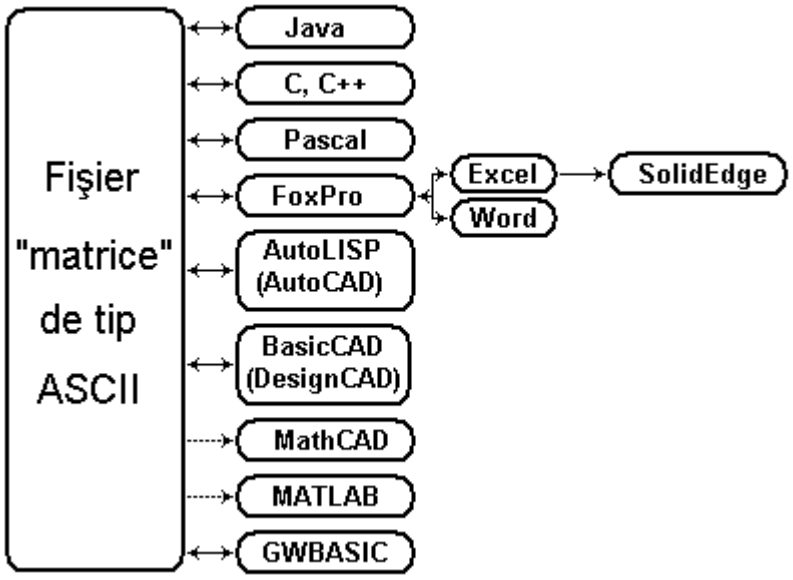

Figura 2 – Interfață între limbajele de programare

Accesul pe disk duce la timpi mari de execuție, deci s-a căutat optimizarea acestuia prin folosirea memoriei. În acest scop au fost folosite structurile dinamice de date. Figura de mai jos prezintă tipurile de liste studiate.

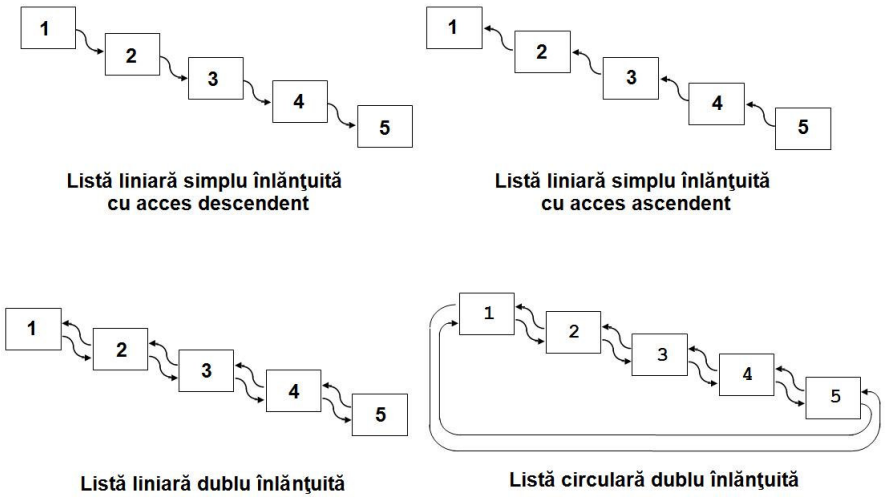

Figura 3 – Structuri dinamice de date

În continuare a fost definit un tip de date de nivel superior care poate folosi optim listele anterior definite. În continuare este prezentat modul în care este folosit acet tip pentru definirea blocurilor, în cadrul unei matrice.

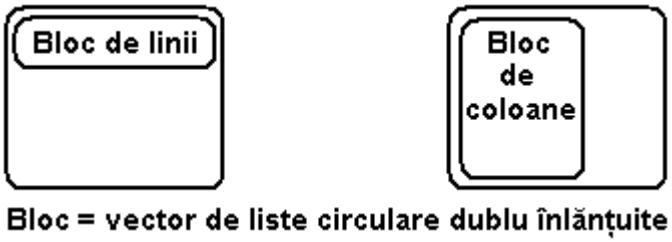

Figura 4 – Definire de blocuri a unei matrice, folosind structurile dinamice de date

Folosind acesată filozofie, utilizatorul poate folosi trei metode de acces folosind procesarea pe blocuri a unei matrice. Figura următoare prezintă aceste modalități de acces.

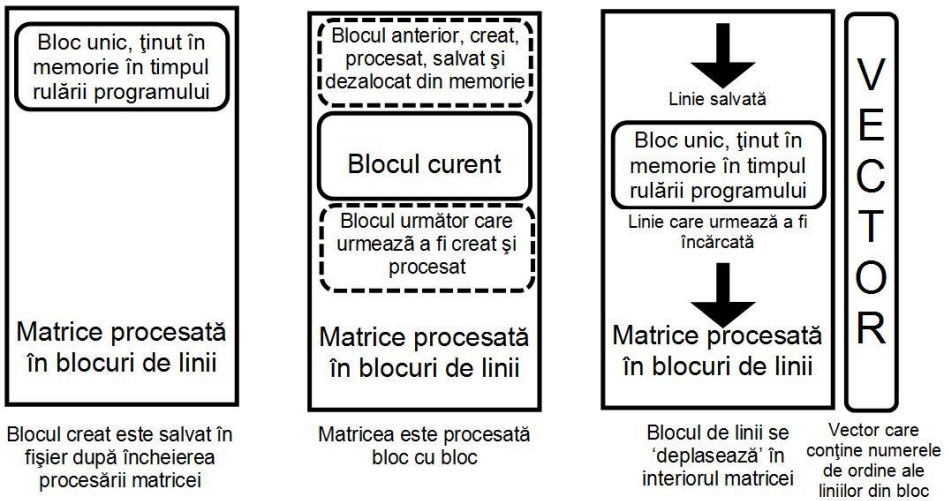

Figura 5 – Modalități de implementare a procesării pe blocuri a matricelor

În paralel cu folosirea limbajelor clasice de programare este avut în vedere şi limbajul Java care oferă facilități pentru dezvoltarea tuturor ramurilor unei aplicații industriale. Figura următoare prezintă câteva dintre domeniile în care poate fi folosit acest limbaj.

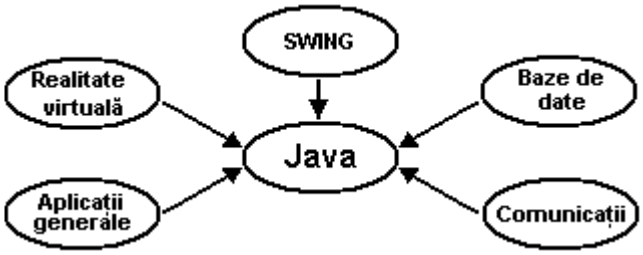

Figura 6 – Domenii de utilizare ale limbajului Java

În plus, datorită încadrării dezvoltării în Java în curentul 'open source', există posibilitatea de a reutiliza aplicațiile scrise de alți dezvoltatori.

CONCLUZIE

Faza actuală se încheie cu realizarea tuturor obiectivelor propuse. Dificultățile neprevăzute în etapele anterioare au fost depăşite. În prezent munca echipei este orientată pentru dezvoltarea fundamentelor pentru aplicațiile specifice fiecărui discipline de bază.