

# *Ingineria cerințelor pentru aplicații web*



**doctorand: CĂLIN EUGEN NICOLAE GAL-CHIȘ**

**conducător: Prof. Univ. BAZIL PÂRV**

Facultatea de Matematică și Informatică  
Universitatea Babeș-Bolyai, Cluj Napoca  
Kogălniceanu str. 1, Cluj Napoca, 400084, România

*Un rezumat al tezei de doctorat*

# Cuprins

---

CUVINTE CHEIE: .....	1
<b>1. Introducere .....</b>	<b>2</b>
1.1 MOTIVAȚIA .....	3
1.2 ENUNȚUL PROBLEMEI.....	4
1.3 CONTRIBUȚIA PRINCIPALĂ .....	4
<b>2. Starea actuală a domeniului de cercetare .....</b>	<b>6</b>
2.1 INGINERIA CERINȚELOR .....	6
2.2 METODOLOGII DE DEZVOLTARE ALE APLICAȚIILOR WEB.....	7
2.3 INTERESE ȘI ASPECTE ÎN DEZVOLTAREA DE SOFTWARE.....	7
<b>3. Ingineria Cerințelor prin folosirea MultiCoS .....</b>	<b>10</b>
3.1 MODELAREA SPAȚIILOR DE INTERESE FOLOSIND <i>MULTICoS</i> , O ABORDARE MULTI-DIMENSIONALĂ DE SEPARARE A INTERESESELOR.....	10
3.2 PRIMITIVELE ABORDĂRII <i>MULTICoS</i> .....	11
3.3 MAPAREA SPAȚIILOR DE ENTITĂȚI ÎN <i>MULTICoS</i> .....	12
3.5 MĂSURAREA SIMILARITĂȚILOR .....	13
3.6 MANAGEMENTUL ȘI SPECIFICAREA INTERESESELOR.....	14
3.7 PROCESUL <i>MULTICoS</i> DE INVESTIGARE A SIMILARITĂȚILOR .....	14
3.8 ABORDAREA <i>MULTICoS</i> ÎN SPRIJINUL REUTILIZĂRII ARTEFACTELOR .....	16
<b>4. Tool-ul MultiCoS .....</b>	<b>17</b>
4.1 ARHITECTURA APLICAȚIEI .....	17
4.2 STOCAREA ȘI MODELUL DE DATE .....	17
4.3 REPORTAREA ȘI GUI .....	18
4.4 UTILIZAREA TOOL-ULUI <i>MULTICoS</i> ÎN PROCESUL DE REUTILIZARE DE COD .....	19
<b>5. Utilizarea MultiCoS .....</b>	<b>21</b>
5.1 UNEALTA <i>MULTICoS</i> UTILIZATĂ PENTRU A SPRIJINI REUTILIZAREA CODULUI.....	21
5.2 UTILIZAREA <i>MULTICoS</i> ÎN INVESTIGAREA CERINȚELOR DIN ACELAȘI SPAȚIU .....	23
5.3 EVALUAREA <i>MULTICoS</i> .....	24
<b>Concluzii .....</b>	<b>26</b>
<b>Bibliografie .....</b>	<b>28</b>

## Cuvinte cheie:

Ingineria aplicațiilor, ingineria cerințelor, semantica cerințelor, aplicații web, dezvoltare de software, dezvoltare de aplicații orientată pe aspecte, ingineria cerințelor orientată pe aspect, metodologii de Dezvoltare a aplicațiilor software, instrumente de ingineria cerințelor, interese transversale, cerințe transversale, separarea multidimensională a cerințelor, cerințe transversale, interese transversale, spații de interese

# 1. Introducere

---

Această lucrare se bazează pe domeniul ingineriei cerințelor și pe domeniul aplicațiilor web folosind o abordare bazată pe *separarea intereselor / preocupărilor* (separation of concerns).

Ingineria cerințelor (IC) este definită ca “zona din ingineria aplicațiilor care se concentrează pe *procesele* ingineriei aplicațiilor (IA), care implică: cunoașterea nevoilor și așteptărilor clientului (negocierea cerințelor), analiza și specificarea cerințelor, prioritizarea cerințelor, derivarea cerințelor, partiționarea și alocarea cerințelor, depistarea cerințelor, managementul cerințelor, verificarea cerințelor, și validarea cerințelor” [1].

IC este o ramură al ingineriei aplicațiilor și este strâns legată de domeniul cerințelor aplicațiilor. IC este procesul prin care cerințele sistemelor software sunt metodic prelucrate din punctul de vedere al formulării, al documentelor în care sunt specificate sau de care sunt legate, și în modul în care sunt întreținute. IC este definită ca fiind [2] sarcina depistării, structurării, și reprezentării cu acuratețe a cerințelor clientului, astfel ca acestea să fie incluse corect în sisteme care îndeplinesc aceste cerințe. Într-o exprimare generică, putem spune că IC este [3] procesul de identificare și articulare a nevoilor unor aplicații și tehnologii noi.

Ingineria Web (IW) se ocupă de “stabilirea și utilizarea principiilor științifice, ingineresti și manageriale sănătoase și abordarea sistematică și disciplinată a dezvoltării cu succes și a instalării și întreținerii unor aplicații bazate pe tehnologii web de înaltă calitate” [4].

IW este disciplina care promovează procesul de producție al unor aplicații web de calitate folosind metodologii, abordări, tehnici și unelte adecvate. Direcțiile IW sprijină procesul de proiectare, dezvoltare, evoluție și evaluare al aplicațiilor web. Atât IW cât și IA folosesc principiile IA, IW fiind diferită prin faptul că este destinată a răspunde necesităților unor aplicații cu profil web.

Întregul conținut livrat prin intermediul tehnologiei Internet este parte din site-uri Web, aplicații Web sau sisteme bazate pe Web. Toate sistemele software care generează acest conținut sunt denumite în această teză ca *aplicații web*. Pe de altă parte, pentru aplicațiile care nu se bazează pe web este utilizată sintagma: *aplicații tradiționale*.

În această teză este prezentată o abordare a IC care îmbunătățește managementul cerințelor aplicației Web. Abordarea se bazează pe *separarea intereselor*. Gestionarea cerințelor utilizând *separarea intereselor* are la bază un proces folosit în ciclul de dezvoltare, proces care este parte a proceselor de inginerie a cerințelor.

Un *interes* este cunoscut a fi o unitate de nivel înalt al sistemul de partiționare, un container pentru localizarea cerințelor semantic legate. *Interesul* poate fi simplu (care conține numai cerințele), sau compozit (conținând alte *interese*, precum și cerințe), permițând astfel structurarea ierarhică din punctul de vedere al cerințelor [5].

Ingineria cerințelor în dezvoltarea aplicațiilor web este oarecum similară cu procesul de inginerie în dezvoltarea de software tradițional. Principalele diferențe se regăsesc în procesul de capturare al aspectelor specifice web cum ar fi aspecte care țin de navigație și aspecte tehnologice, și în modul în care cerințele sunt implementate în cod.

Pentru a răspunde necesității de a gestiona artefacte de aplicații web și pentru a facilita re folosirea de artefacte, cum ar fi cod, sau a altor documente specifice aplicației, vom introduce o metodologie de gestionare a cerințelor de web. Această metodologie poate sprijini și dezvoltarea de software tradițional, sau poate fi aplicată în alte domenii. În cele din urmă, ne vom concentra pe domeniul de aplicații web. Metodologia se bazează pe *separarea multidimensională a intereselor* (SoC) și folosește *transformări model-la-model* (model-to-model transformations) și legături între

modele pentru a surprinde specificul diferitelor aplicații web, toate acestea folosind relațiile dintre artefacte și spațiile de interese, și o metrică de similaritate a semanticii. Această abordare ne va permite, pe baza similarității semantice, să accesăm artefactele unei aplicații web existente, în scopul de a sprijini dezvoltarea unei alte aplicații.

## 1.1 Motivația

Este foarte important pentru ca activitățile de inginerie a cerințelor să fie eficiente, în scopul de a îndeplini nevoile utilizatorilor și clienților, și pentru a livra la timp aplicația, îndeplinind condițiile de livrare a produsului în termenii conveniți de timp și în buget [6]. După avântul ingineriei Web în anii nouăzeci, una dintre principalele lacune constatate a fost lipsa de tehnici adecvate de specificare a cerințelor pentru aplicații Web.

În domeniul Ingineriei Cerințelor au fost prezentate mai multe abordări în ultimele decenii. Unele abordări au fost excelente în ceea ce privește specificarea cerințelor legate de structura și comportamentul sistemelor informatice. Alte abordări s-au concentrat cu succes pe metode care descriu cerințele nefuncționale, precum fiabilitatea, reutilizarea sau performanța sistemelor informatice.

Capturarea cerințelor aplicației web are nevoie de tehnici diferite de capturare a cerințelor față de tehnicile utilizate în capturarea cerințelor pentru software-ul traditional. Acestea din urmă nu sunt adecvate în sprijinirea caracteristicilor specifice aplicațiilor web, cum ar fi cele de navigare. Astfel, cercetătorii privesc *IC pentru aplicații web* ca un nou domeniu, domeniu care urmează a fi investigat [7].

Într-un alt studiu [8] se arată că dezvoltatorii de aplicații web afirmă că proiectele sunt pe punctul de a nu respecta termenele și bugetul, principalele două motive fiind: modul de captare a cerințelor problemei și modul defectos de comunicare dintre dezvoltatori și părțile implicate. Putem concluziona că sunt necesare tehnici specifice IC pentru captarea cerințelor aplicațiilor web. Împreună cu captarea cerințelor, traducerea cerințelor în artefacte software adecvate vine să completeze procesul de utilizare a cerințelor. Aceasta este de fapt, o problemă clasică: cum să se facă avansarea din spațiul problemei (cerințelor utilizatorilor) în spațiul soluției (proiectare și implementare). Comunitatea ingineriei aplicațiilor este în căutarea unei soluții pentru a rezolva această problemă.

În practică, există anumite probleme ale aplicațiilor web care sunt deja rezolvate; există module software sau biblioteci de cod deja dezvoltate pentru anumite situații tip de utilizare. Alte probleme ar putea beneficia de soluții nepublicate, dar deja construite și disponibile în interiorul firmelor de programare. Roata nu ar trebui să fie reinventată, ci, etapa de dezvoltare a produsului software ar trebui bazată pe soluții deja existente.

În scopul de a urmări aceste soluții existente, la începutul procesului de dezvoltare, este necesară catalogarea artefactelor și gruparea lor nu doar ca artefacte unice, dar și în categorii generice, parte dintr-o gamă de soluții pentru un anumit domeniu. Acest proces ar trebui să înceapă în primele faze ale procesului de dezvoltare, cu accent deosebit pe cerințe. Mai apoi, pornind de la cerințe, procesele de stabilire a depistării / trasabilității artefactelor (“traceability”) ar trebui să fie răspândite la toate artefactele produse în timpul procesului de dezvoltare al aplicației.

Odată cu existența unor proiecte care sunt tot mai mari și mai complexe, [9] ar trebui urmate principii relevante, cum ar fi modularizarea, abstractizarea și încapsularea, deoarece oferă un sprijin important în atingerea unui proces clar de *separare de interese*. Metodele de inginerie software structurate sau orientate pe obiect furnizează metodologii bazate pe astfel de principii, dar ele nu sunt în măsură să separe complet toate relațiile complexe dintre *interese*, pentru că acestea se concentrează

pe procesul de modularizare pe criterii de *descompunere dominantă*. Această *tiranie a descompunerii dominante* [10] conduce la un proces de modularizare ineficient axat pe un singur *interes*.

Dezvoltarea de software orientat pe aspecte (AOSD) oferă un suport mai bun în gestionarea procesului de *separarea a intereselor* și de a ajunge la o mai bună modularitate prin descompunerea sistemului în părți care au cât mai puține funcționalități suprapuse cu puțință. De asemenea, scopul AOSD este de a gestiona *interesele transversale* (crosscutting concerns), sugerând o metodologie bazată pe identificarea, separarea, reprezentarea și compoziția sistematică a *intereselor*. Obiectivele primare AOSD sunt modularizarea și încapsularea *intereselor transversale*. Termenul de *interesele transversale* înseamnă *interesele legate*, și sunt de fapt attribute software comune. În ultimul timp, managementul *intereselor transversale* se face, nu doar în faza de programare, dar și în etapele anterioare dezvoltării produsului software, cum ar fi designul, arhitectura și cerințele [11, 12].

Pentru a atinge un astfel de obiectiv, este necesară o metodologie de grupare și abordarea acestor artefacte pe baza intereselor, iar apoi este necesar un proces de creare a relațiilor între artefactele nou cerute și alte soluții existente, candidate pentru a fi reutilizate. O astfel de metodologie va sprijini reutilizarea de artefacte și a codului, oferind soluții complete / parțiale sau indicând soluții existente în domenii identice sau similare.

## 1.2 Enunțul problemei

După cum am văzut, ingineria cerințelor pentru aplicații web este încă o temă de cercetare deschisă, și există domenii care pot fi îmbunătățite.

Organizarea cerințele și a artefactelor în categorii și crearea de legături între ele poate duce la reutilizarea de artefacte și cod. Următoarele direcții de cercetare au fost propuse:

**Direcția de cercetare 1:** Cum ar trebui cerințele aplicațiilor web să fie specificate și organizate în scopul de a captura cât mai eficient attributele și semantica lor și de a le grupa în categorii?

**Direcția de cercetare 2:** Cum ar trebui să se realizeze metodologic legături între modele pentru a avea acces la și pentru a putea reutilizarea soluții sau resurse existente?

**Enunțul problemei.** Cu alte cuvinte, problema va afirma următoarele: Să se furnizeze o metodă pentru a descrie relațiile dintre orice resurse implicate în sau legate de procesul de dezvoltare a produsului software (cum ar fi cerințele, artefacte, cod, părțile interesate, sau orice alte entități logice sau fizice), pentru a defini relațiile de mapare a resurselor la o varietate de *interese* comune. Măsurarea similarității resurselor va fi bazată pe importanța relațiilor de similaritate dintre resurse.

Pentru că similaritatea resurselor este direct legată de *interesele transversale*, putem, nu numai să oferim sprijin pentru reutilizarea resurselor similare, dar, de asemenea, putem să urmărim *interesele transversale* bazate pe astfel de relații cu resursele.

## 1.3 Contribuția principală

În această teză cercetările efectuate conduc la dezvoltarea unei metodologii numite MultiCoS (Multiple Concern Spaces), susținute de un instrument software care oferă o soluție la această problemă.

În primul rând, o abordare proprie este folosită pentru a organiza cerințele și artefactele. Abordarea se bazează pe separarea multidimensional de intereselor (multidimensional separation of concerns) [12] și a fost adaptată pentru a fi aplicată pe diferite artefacte și resurse, ne fiind limitată la un singur tip, cum ar fi cerințele [13].

După ce cerințele și alte artefacte sunt organizate în conformitate cu abordarea MultiCoS, am introdus o metodă de a mapa cerințele și resursele din spații de probleme diferite la diferite *spații de interes* (concern spaces) reprezentative. Odată maparea realizată, putem calcula un coeficient de similaritate semantică a diferitelor perechi de resurse, bazat pe indicii de similaritate Gower [14]. În funcție de importanța relației dată de coeficienții de similaritate, putem detecta artefacte similare semantic din alte spații de probleme sau în același spațiu problemă și putem decide să le reutilizăm. Transformări model-la-model sunt efectuate în timpul procesului menționat pentru a permite metodei să producă rezultatele așteptate.

Pentru a obține aceste rezultate, un studiu a fost efectuat în capitolul al doilea, pe mai multe domenii, aceste domenii fiind prezentate pe scurt în continuare.

Diferitele metodologii disponibile pentru dezvoltarea de aplicații web au fost investigate, o atenție deosebită fiind dată modului în care sunt gestionate cerințele.

Starea actuală a domeniului de cercetare în ceea ce privește ingineria cerințelor, cu precădere cel al reprezentării cerințelor pentru aplicațiile web, este cercetat cu scopul de a surprinde cât mai bine toate aspectele cerințelor și de a le integra în procesul de gestionare a cerințelor, utilizând abordarea propusă aici.

Deoarece această abordare se bazează pe paradigma *separării intereselor* (Separation of Concerns), vom investiga metodologiile de dezvoltare ale aplicațiilor existente care folosesc *separarea intereselor*, pentru a observa diverse moduri în care interacționează artefactele aplicațiilor în timpul procesului de dezvoltare al acestora, cu un accent pe procesul de management al cerințelor și pe specificul aplicațiilor web.

Având în vedere că, pentru a sprijini fluxul procesului pentru această abordare, este folosit un tool bazat pe tehnologie web, o altă investigație este făcută în ceea ce privește reprezentarea primitivelor și a relațiilor între modelele introduse de metodologia proprie, și cu referire la transformările model-la-model necesare și relațiile existente între modele. De asemenea, o investigație se face cu privire la tooluri similare existente.

Teza este organizată după cum urmează: în capitolul 1, scopul tezei este explicat, motivația tezei și declarația problemei sunt, deasemenea, prezentate. În capitolul 2 sunt prezentate în trei secțiuni starea cercetării actuale pentru domeniul ingineriei cerințelor, pentru metodologiile de dezvoltare ale aplicațiilor web și pentru metodologiile orientate pe aspect, utilizând *interese*, toate acestea având în vedere că soluția problemei de cercetare care este rezolvată în această teză se bazează pe aceste domenii. Capitolul 3 se axează pe contribuția tezei, MultiCoS, o abordare care susține reutilizarea cerințelor și a altor artefacte, pe baza similarităților semantice ale acestora. În capitolul 4 se prezintă instrumentul software MultiCoS, bazat pe abordarea MultiCoS. Fluxul de lucru al acestei aplicații, MultiCoS, este testat într-un studiu de caz în capitolul 5, oferind un exemplu al modului în care abordarea MultiCoS poate oferi sprijin în reutilizarea cerințelor și a modulelor de cod, având la bază similaritatea cerințelor.

## 2. Starea actuală a domeniului de cercetare

---

În timpul procesului de cercetare, au fost investigate cele mai importante rezultate și studii legate de paradigmele care fac parte din domeniul tezei de față, subliniind pentru fiecare, specificul și conceptele lor de bază și, de asemenea, subliniind legăturile relevante cu tema studiată.

Cu scopul de a surprinde cât mai bine imaginea de ansamblu, cercetarea s-a concentrat pe următoarele domenii: aplicații web, metodologii Web, dar și IC, cerințe pentru aplicații web, tool-uri de IC pentru aplicații Web, *separarea intereselor* și metodologiile lor.

### 2.1 Ingineria Cerințelor

Pentru a avea o mai bună înțelegere a IC în această secțiune vor fi trecute în revistă aspecte legate de metodologii centrate pe IC, managementul cerințelor și metode de reprezentare a cerințelor.

*Cerințele* specifică ce ar trebui să facă sau ce condiții ar trebui să îndeplinească un produs software. Având în vedere acest lucru, cerințele sunt declarații valoroase care ar trebui să fie foarte specifice, cât mai clare posibil, și lipsite de ambiguitate.

Există mai multe abordări notabile pentru ingineria cerințelor:

- Ingineria cerințelor orientate pe agent
- Ingineria cerințelor orientată pe obiective (KAOS)
- Ingineria cerințelor acționată de model (Model Driven)
- Ingineria cerințelor orientată pe aspecte (AORE)
- Abordare orientată pe puncte de vedere (ViewPoint Oriented Approach - VOA)
- Ingineria cerințelor acționată de procese business: o abordare bazat pe obiective.

Printre cele mai avansate framework-uri care sprijină aceste abordări ale ingineriei cerințelor sunt i \*, fUML, KAOS.

Prelucrarea cerințelor înseamnă parcurgerea următoarelor activități specifice cu privire la cerințe, cum ar fi: suscitarea (elicitation), analiza/negocierea, specificarea (documentarea), verificarea (validarea).

Cerințele pot fi cerințe funcționale sau cerințe non-funcționale. Specificul cerințelor pentru aplicații web includ în cadrul cerințelor funcționale, printre altele, cerințele referitoare la date, (utilizator), cerințe externe de interfață utilizator, cerințe de navigație, cerințe de personalizare, cerințe tranzacționale. Cerințele non-funcționale sunt inclusiv constrângeri de proiectare, constrângeri de natură tehnică, cerințe determinate de respectarea standardelor și cerințe legate de performanță.

Managementul cerințelor înseamnă o abordare sistematică a determinării, organizarea, și documentarea cerințelor sistemului. Unele dintre cele mai frecvente probleme ale cerințelor sunt scrierea lor și urmărirea schimbărilor acestora.

Potrivit Friedental et al [15]: "o *cerință* specifică o capacitate sau condiție care trebuie să fie îndeplinită. O cerință poate defini o funcție pe care un sistem trebuie să o efectueze, sau o condiție de performanță pe care un sistem trebuie să o obțină".

Pentru a identifica mai bine cerințele unei probleme este mai bine să fie folosite metodologii existente testate și rezultatele cercetărilor disponibile în domeniu. Reutilizarea cataloagelor cu cerințe este sugerată pentru a ușura procesul de identificare al cerințelor și de a folosi

același vocabular pentru cerințe similare. Potrivit template-ului Volere de specificare a cerințelor [16], avem trei tipuri de cerințe: (1) constrângeri, (2) cerințe funcționale și (3) cerințe non-funcționale (NFR).

Menirea unui sistem este de a produce anumite funcționalități pentru un client, în scopul de a-l ajuta sau de a-i oferi servicii, făcându-i acestuia viața mai ușoară și mai eficientă. De obicei, persoana care crează sistemul nu este persoana pentru care este destinat sistemul, astfel, existând riscul unor probleme de comunicare și realizare, și diferențe dintre cerințele preconizate și produsul livrat.

Pentru sistemele de dimensiuni mici riscul este redus, pentru că prototipul și versiunile inițiale pot fi ușor ajustate, iar comunicarea între utilizatori și dezvoltatori nu este finalizată, reducând neînțelegerile.

Pe de altă parte, pentru sistemele mari această abordare nu este adecvată, ceea ce duce, de obicei, la proiecte pentru care nu vor fi respectate termenele de execuție sau cerințele de buget. Pentru aceste proiecte mari, majoritatea programatorilor nu sunt în aceeași locație, unii fiind subcontractați de la alte companii, și cu siguranță, nu sunt în contact cu utilizatorul final. Ei au nivele diferite de acces la proiect și relații restricționate profesional cu antreprenorii. Acest lucru elimină orice șansă de comunicare directă între extremele acestui proces: programator și utilizator.

## 2.2 Metodologii de dezvoltare ale aplicațiilor web

Am investigat aici metodologiile disponibile pentru dezvoltarea de aplicații web, precum și tool-urile specific IC, în cazul în care acestea există și se potrivesc cu aceste metodologii de dezvoltare ale aplicațiilor web .

Table 1 Tool-uri de IC dezvoltate pentru a sprijini metodologii web existente

Methodologie	Aplicația care include tool-ul	Tool-ul	Open Source / Comercial
OOWS	Eclipse	Graph2OOWS & AGG	Open Source
UWA	Eclipse/WebDSL	WebDSL	Commercial
UWE	MagicDraw	MagicDraw	Commercial
UWE	MockupDD	MockupDD	Commercial
WebML	Eclipse	WebRatio	Open Source
NDT	NDT application.	NDT-Tool	Commercial

## 2.3 Interese și aspecte în dezvoltarea de software

În această secțiune vom face o prezentare generală a metodologiilor de dezvoltare software orientate pe aspecte (AOSD) și a abordărilor IC orientate spre aspecte (AORE).

AORE se concentrează atât pe importanța *intereselor* funcționale cât și pe cea a celor non-funcționale ca o nevoie de a satisface trei factori importanți ai intereselor [17]: necesitatea de compoziție, necesitatea de urmărire și dezvoltarea de noi tehnologii. La nivelul cerințelor, un *aspect* este o proprietate a unei cerințe sau a unui set de cerințe care afectează un sistem, deoarece acestea pot restricționa un anumit comportament sau pot modifica un comportament specific cerințelor.

În această secțiune este realizată o trecere în revistă a celor mai importante abordări din Analiza Cerințelor Orientată pe Aspecte (AORA). De asemenea, este evidențiată importanța metodelor de dezvoltare a produselor software bazate pe aceste tipuri de abordări. Motivul folosirii metodologiilor de dezvoltare orientată pe aspecte a produselor software este legată de înțelegerea



conceptului de separare a intereselor, care are scopul de a găsi și stabili într-o formă modulară a componentelor software prin utilizarea unor concepte, obiective sau scopuri specifice.

Abordările specifice ingineriei cerințelor orientate pe aspecte sunt investigate în această secțiune, fiind prezentate în detaliu metodologiile lor și specificul acestora. Fiecare dintre ele utilizează *interese* în procesele interne, interesele fiind de obicei parte din definițiile cerințelor.

Separarea intereselor este introdusă de Dijkstra. Principalul motiv pentru folosirea separării intereselor este gestionarea unei probleme într-un mod mai ușor, prin divizarea în module bazate pe astfel de interese, module care pot fi combinate ulterior pentru a oferi un răspuns complet la o problemă. Acest concept este important în dezvoltarea aplicațiilor software, deoarece oferă anumite beneficii, cum ar fi: reducerea complexității, sprijinirea trasabilității, oferirea unei mai bune înțelegeri a problemei. Dar, unul dintre cele mai importante beneficii este acela că oferă mijloace pentru efectuarea de schimbări cu impact mic în cod, pentru că *interesele* sunt stocate în module diferite. Interesele transversale (crosscutting concerns) sunt un domeniu major care urmează a fi abordat, deoarece unele interese nu pot fi complet separate unul de altul din punctul de vedere al modularizării, ceea ce va conduce la abordări diferite în rezolvarea problemelor care apar.

O definiție generală provenind din domeniul dezvoltării de software orientat spre aspecte de afirmă că "un *interes* este orice preocupare a unui sistem software" [18]. O definiție mai specifică este asigurată de IEEE care prevede: "un *interes* este acea preocupare care se referă la dezvoltarea sistemului, funcționarea acestuia sau orice alte caracteristici care sunt critice sau altfel importante unuia sau mai multor părți interesate" [19].

În sensul acestei teze, un *interes* se va referi la o proprietate care abordează o anumită problemă, care este de importanță pentru una sau mai multe părți implicate și care pot fi definite de un set de cerințe coerente [20]. Având în vedere acest lucru, un *interes* se referă la proprietăți pe care un sistem trebuie să posede, în scopul de a satisface funcționalități necesare.

Scopul dezvoltării de software orientat pe aspecte (AOSD) este de a oferi un sprijin cât mai bun în gestionarea procesului de separare a intereselor și de a atinge o mai bună modularitate prin descompunerea sistemului în părți care au cât mai puține funcționalități suprapuse.

Când cerințele transversale, funcționale sau nefuncționale, sunt dificil de separat în module, impactul acestora în sistem nu poate fi cuantificat cu precizie. Spre deosebire de alte abordări tradiționale, care nu gestionează interesele transversale cu succes, abordările AORE încearcă să gestioneze cerințele transversale compunându-le printr-un mecanism de abstractizare de nivel ridicat.

Vom arăta că, în timp ce abordările existente pentru modelarea sistemelor adresează interesele în contexte specifice pentru scopuri specifice, o modelare generică care să trateze interesele este încă necesară. Scopul acestei secțiuni este de a exprima nevoia unui cadru unitar de modelare a spațiilor de interese și nevoia de a specifica relațiile acestora cu diferite tipuri de entități, de a defini primitivele necesare pentru a reprezenta astfel de relații, și nevoia de a propune modele care pot fi utilizate în diferite scenarii în timpul folosirii unei anumite metodologii.

Diverse metodologii de tip AORE, de gestionare a separării intereselor sau modele de separare a intereselor au fost investigate, printre ele enumerăm: Hyperspaces, Requirements Modeling Using Separation of Concerns, ModelSoC, Aspect-oriented software development with use cases, Cosmos, Tropos, Multi-Dimensional Separation of Concerns, Aspect-oriented requirements engineering with ARCADE, Aspect-oriented scenario modeling, Theme, RDL – Requirements description language, AOV-Graph.

Interesele sunt responsabile în a oferi sens artefactelor, dar lucrând cu acestea se adaugă un volum de muncă semnificativ etapelor de dezvoltare a aplicațiilor. Având în vedere că interesele pot fi ușor relaționate cerințelor, este necesar să prelucrăm în același timp și interesele și cerințele. Metodologiile investigate utilizează într-un mod diferit interesele și cerințele. În dorința de a folosi

interesele în organizarea cerințelor și de a aplica un proces controlat pentru gestionarea cerințelor aplicațiilor web, unele dintre metodologiile sunt mai potrivite din acest punct de vedere decât altele.

Două dintre aceste metodologii oferă un suport solid în această abordare a gestionării cerințelor: abordarea AORE, și abordarea Separarea multidimensională a intereselor (Multidimensional Separation of Concern). O îmbinare a acestor două abordări, cu o nouă viziune de gestionare a cerințelor funcționale și non-funcționale, de gestionare a cerințelor specifice aplicațiilor web și de gestionare a altor artefacte este metodologia MultiCoS care va fi prezentată în capitolele următoare.

În ceea ce privește interesele transversale, care sunt cu precădere non-funcționale, ar trebui dezvoltate metodologii pentru identificarea intereselor transversale funcționale. Tehnici specifice pentru a identifica cerințele aspectuale ar trebui să fie propuse, iar instrumente pentru a gestiona cerințele aspectuale ar trebui să fie eficientizate. Metode pentru rezolvarea conflictelor la nivel de cerințe ar trebui, deasemenea, propuse.

## 3. Ingineria Cerințelor prin folosirea MultiCoS

---

O caracteristică comună a tuturor abordărilor prezentate în capitolul anterior este că acestea se concentrează pe un singur tip de interese grupate într-un spațiu. În cea mai mare parte, în aceste abordări, din necesitatea de a rezolva o anumită problemă, au apărut reprezentări structurate grupurilor de interese sub forma unor spații, fapt ce a dus la definirea *spațiului de interese*. Astfel, același concept de *spații de interese* a fost introdus în moduri diferite în fiecare dintre abordări pentru a se potrivi necesităților abordării respective.

### 3.1 Modelarea Spațiilor de Interese folosind *MultiCoS*, o abordare Multi-Dimensională de separare a intereselor

În produsele software, specificațiile, cerințele, chiar și codul sau modulele software sunt etichetate cu cuvinte cheie, sau descrise folosind atribute sau valori specifice. Scopul folosirii acestor etichetate este legat de semantica obiectului etichetat, și este folosit ca o formă de indexare pentru acea categorie specifică. Un meta-model pentru separarea intereselor este propus aici pentru a oferi sprijin în folosirea un tip unitar de notație în etichetarea diferitelor tipuri de resurse utilizate în procesul de dezvoltare a software-ului, de la cerințe și specificații până la cod sau module software.

Utilizarea unui standard, a unei notații unitare poate avea mai multe beneficii. Vom arăta că domeniile care pot beneficia de acest standard variază de la reutilizarea codului, sau reverse-engineering, la atribuirea tehnologiei potrivite pentru dezvoltarea aplicațiilor, sau la dezvoltarea de software orientat pe aspecte (AOSD), și, nu în ultimul rând, în ingineria cerințelor (gruparea cerințelor pe categorii, cum ar fi: tehnologie, importanță, actori, volatilitate, funcționalitate).

Se știe că separarea intereselor în ingineria software oferă multe avantaje, cum ar fi reducerea complexității produsului software, îmbunătățirea procesului de reutilizare a codului, și un proces de evoluție simplificat în dezvoltarea aplicațiilor. Utilizarea intereselor ajută programatorii să vizualizeze mai bine sarcinile de lucru prestate și locul acestora în procesul de dezvoltare al aplicațiilor. Pe de altă parte, clienții și cei interesați în produsul software pot să se relaționeze mai ușor la specificațiile aplicației software. Inginerie inversă (reverse engineering), analiza impactului schimbărilor și reutilizarea de cod sunt, de asemenea, sprijinite de utilizarea intereselor. Practic, sistemele software ar trebui să beneficieze în cât mai multe moduri atunci când se utilizează separarea intereselor în timpul ciclului de dezvoltare al produsului.

Din punct de vedere al generalității terminologiei, *un interes* poate fi considerat a fi *orice problemă de interes într-un sistem software*, iar *un spațiu de interese* ca *o reprezentare organizată a intereselor și a relațiilor dintre acestea*. Aceste definiții simple și versatile, vor da flexibilitatea necesară abordării de față, fără a restricționa sau contrazice în vre-un fel definiții deja existente. Definirea spațiilor de interese și legarea acestora de codul sursă corespunzător în timpul stadiului de dezvoltare are puterea de a adăuga semantică componentelor unei aplicații. Valoarea semantică poate fi adăugată diferitelor spații de entități, cum ar fi: spațiul modelului aplicației, spațiul colecției de date, spațiul de relații, spațiul de vizualizări, spațiul de design, dar și spațiul de cerințe.

Abordarea prezentată în această teză este numită MultiCoS (Multiple Concern Spaces), o abordare bazată pe separarea multi-dimensională a intereselor, și a fost introdusă în [13], extinzând separarea multi-dimensională a intereselor definită în [12]. O noutate introdusă de abordarea

MultiCoS este că modelul poate face referire la mai multe *spații de sisteme* (Figura 1) și, deasemenea, se poate relaționa cu mai multe spații. Relațiile formate între *spațiile de sistem* și *spațiile de interes* pot adăuga valoare în domeniul ingineriei software, oferind un mod integrat care să reprezinte relații, dependențe, calități/atribute ale diferitelor concepte sau entități software.

De exemplu, două *spații de sistem* diferite (două aplicații web diferite) pot fi relaționate prin folosirea unui *spațiu de interes* comun. Acest lucru poate duce la reutilizare de artefacte și cod.

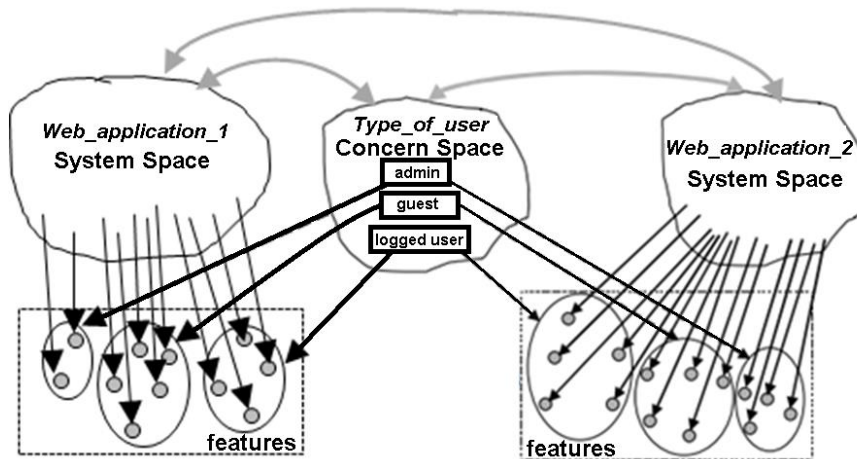


Figure 1 Spații de sistem diferite relaționate printr-un spațiu de interes comun în *MultiCoS*

### 3.2 Primitivele abordării *MultiCoS*

Vor fi definite următoarele primitive ale *MultiCoS*: ConcernSpace (Spațiu de interes), Multispace (multispațiu), Concern (dimensiune de interes), entity (entitate), entity space, concern matching value, așa cum au fost adoptate din [21,22] și utilizate în contextul [12].

- **Concern Space** = un grup de interese referitoare la, sau care descriu capacități (probleme / comportament) similare pentru cel puțin un tip de entități. Un *Concern Space* are un nume, o descriere, și un set de dimensiuni (stocate ca un vector - o dimensiune pentru fiecare interes din spațiu). Opțional, parametrii pot fi declarați, pentru a configura un *Concern Space* (un codomeniu de valori pentru toate dimensiunile, sub forma unui interval de forma  $[0, maxvalue]$ , care, în mod implicit este  $[0, 1]$ ).

- **Multispace** = un spațiu de *Concern Spaces*—mai multe *Concern Spaces* grupate în funcție de aceeași categorie de meta date.

- **Concern** = Interes - unul dintre elementele din setul de elemente ale unui *spațiu de interes*. Un *interes* are scopul de a descrie un atribut al unei entități.

- **Entity** (Entitate) - un obiect care poate fi asociat cu unul sau mai multe interese dintr-un spațiu interes. Entitățile pot fi cerințe, artefacte, utilizatorii, module software, tehnologii, specificații, și se poate referi chiar și la spații.

- **Entity Space** (Spațiu de entități) - o colecție de entități cu coeziune. Devreme ce entitățile pot fi de diferite categorii, ele pot fi grupate în spații de entități distincte.

- **Concern Matching Value** (Valoarea de potrivire a interesului) - o valoare care descrie numeric (scalar) *importanța* pe care o are relația dintre o entitate și un interes (o dimensiunea de interes), prin o valoare din intervalul  $[0, maxvalue]$ . Valoarea interesului reflectă greutatea atributului entității în ceea ce privește intervalul de valori la care se referă spațiul de interes, și la compararea/relaționarea contrastului interesului cu alte interese din spațiul de interes. *Concern Space* ar trebui să includă cel puțin o dimensiune (un interes).

Pentru fiecare entitate există un vector din fiecare spațiu de interese care este asociat cu entitatea, el exprimând amprenta entității în spațiul de interese. Această asociere poate fi exprimată ca o funcție având ca și argumente entitatea și spațiul de interese și ca și rezultat, vectorul asociat cu valorile intereselor pentru fiecare dimensiune a spațiului de interese.

Fie  $e$  o entitate a spațiului sistem  $E$ ,  $S$  un spațiu de interese cu  $n$  dimensiuni și o funcție de mapare  $f$  care descrie relația (1) dintre entitatea  $e$  și spațiul de interese  $S$ .

$$f(e,S) = v, \text{unde } e \in E, v \in \mathbf{R}^n \quad (1)$$

Spațiul de interese  $S$  este descris folosind următorul format:

$$S = (id, name, description, list\ of\ concern\ dimensions, max\_value)$$

Unei entități îi poate fi asociată o valoare pentru fiecare dimensiune, valoarea trebuind să fie în intervalul  $[0, maxvalue]$ .  $Maxvalue$  este valoarea maximă pe care o poate lua o entitate în fiecare dimensiune a spațiului de interese, și este o valoare pozitivă, nenulă.

### 3.3 Maparea Spațiilor de Entități în *MultiCoS*

Așa cum se vede și din imaginea următoare (Figure 2), o entitate poate fi asociată cu mai multe spații de interese. Vom considera următoarele spații de interese:

```
S1={1, "MVC", "the weight in the Model-View-Controller", (Model, View, Controller), 1},
S2={2, "CRUDS", "Create Read Update Delete Static functionalities of an entity", (read, create, update, delete, static), 1},
S3={3, "nonFA", "non-functional aspects of entities", (sleekdesign, loadspeed, volatility), 10},
S4={4, "Priority", "importance of the entity", (priority), 10}.
```

Entitatea *Entity1* este o cerință în spațiul de cerințe cu numele "productpage" având descrierea: „Display a professional looking Product Page with product details”. Mai jos sunt reprezentate valorile funcției de mapare  $f$ , având *Entity1* ca primul argument și fiecare dintre spațiile de interese prezentate ca și al doilea.

```
f(Entity1, S1)={0,1,0}
f(Entity1, S2)={1,0,0,0,0}
f(Entity1, S3)={8,4,0}
f(Entity1, S4)={7}
```

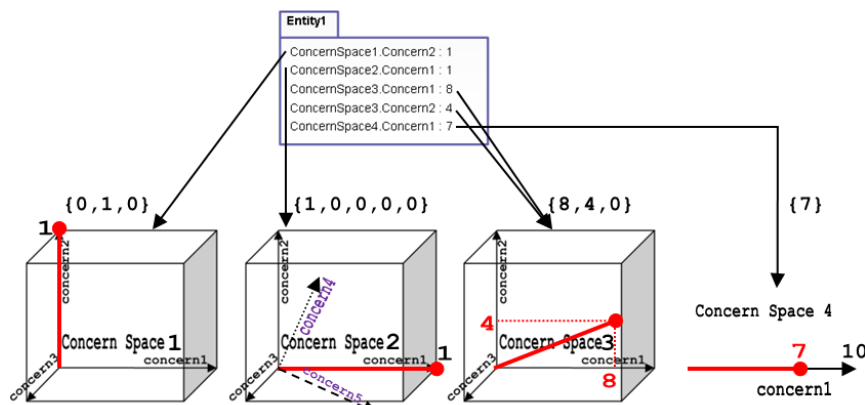


Figure 2 Entitate mapată în mai multe spații de interese și valorile de mapare corespunzătoare

### 3.5 Măsurarea similarităților

Putem considera că entitățile sunt similare, în cazul în care funcția  $f$  oferă vectori de valori apropiate pentru entitățile din spațiile de interes. O metrică va fi utilizată pentru a măsura similaritatea a doi astfel de vectori în abordarea MultiCoS [23]. Similaritatea și distanța sunt valori opuse, indicele de similaritate dintre două puncte fixe într-un spațiu  $n$ -dimensional este cu atât mai mare cu cât distanța dintre cele două puncte este mai mică, iar indicele de similaritate este cu atât mai scăzut cu cât distanța este mare. Relația dintre distanța  $d$  și coeficientul de similaritate  $s$  sunt prezentate în principal în două formule, în mare, alegerea uneia sau a alteia se face în funcție de faptul că distanța este sau nu normalizată.

Când distanța este normalizată la o valoare în intervalul  $[0, 1]$ , indicele normalizat al distanței se mai numește și coeficient de disimilaritate. Formula utilizată pentru a raporta disimilaritatea (distanța) la similaritate este, în cele mai multe cazuri, inversul probabilistic (2.1). Acesta este mai ales utilizat atunci când valorile sunt normalizate în intervalul  $[0, 1]$ . În caz contrar, se utilizează inversul (2.2).

$$s = 1 - d \quad (2.1)$$

$$s = 1 / d \quad (2.2)$$

Pentru a calcula similaritatea, am considerat un model cunoscut de reprezentare al similarității, și anume *funcția de densitate a probabilității* (*probability density function*) sau, pe scurt *pdf* [5]. Fie  $A$ , o mulțime de  $n$  elemente a cărei valori sunt discrete și finite în intervalul  $[0, k]$ . O histogramă  $H(A)$  a unei mulțimi  $A$  reprezintă frecvența fiecărei valori. Frecvența valorii corespunzătoare elementului cu numărul de ordine  $i$  este notată cu  $H_i(A)$ .

Un *pdf* este determinat efectuând împărțirea fiecărui nivel la  $k$ :  $P = H(A)/k$ .

Fie două entități  $X, Y$  din spațiul de entități  $E$ ,  $X, Y \in E$ , și un spațiu de interese  $S$ , cu  $n$  dimensiuni. Să considerăm că avem doi vectori pentru cele două entități, vectorul  $A=\{a_i\}$  pentru entitatea  $X$ , și vectorul  $B=\{b_i\}$  pentru entitatea  $Y$ , reprezentând funcția de mapare pentru cele două entități, în spațiul de interese  $S$ , și  $P = H(A)/k$ , iar  $Q = H(B)/k$ . Vom măsura  $ss$ , indicele de similaritate al spațiului pentru două entități (4), folosind *pdf*-urile corespunzătoare și formula lui Gower (3) pentru măsurarea distanței convertită la index de similaritate (2).

$$d = \frac{1}{n} \sum_{i=1}^n |P_i - Q_i| \quad (3)$$

$$ss = 1 - d = 1 - \frac{1}{n} \sum_{i=1}^n |P_i - Q_i| \quad (4)$$

Să considerăm  $d_i = |P_i - Q_i|$ , distanța corespunzătoare dimensiunii  $i$  și  $s_i = 1 - d_i$ , coeficientul de similaritate corespunzător dimensiunii. Pentru doi vectori egali,  $P_i = Q_i$ , obținem  $d_i = 0$ , ceea ce duce la  $s_i = 1$ , însemnând similaritate maximă. Cunoscând acest lucru, pentru două valori nule,  $x_i = y_i = 0$ , obținem de asemenea  $s_i = 1$ , similaritate maximă, deci, vom exclude astfel de dimensiuni din calculul coeficienților de similaritate, având ca raționament faptul că lipsa unei anumite proprietăți nu poate genera o similaritate puternică. Această presupunere este făcută cu aceleași considerente și în calcularea similarității Tanimoto sau a indexului Jaccard [5], bazați pe raționamentul că absența unui caracteristici în două obiecte nu furnizează informații valoroase pentru stabilirea similarității [6].

Coeficientul de similaritate pentru multispații (*multispace similarity coefficient*) (6),  $ms$ , a două entități  $X, Y$  cu referire la un set de spații de interese  $S_1, \dots, S_w$  este calculat ca fiind media coeficienților de similaritate pentru toate cele  $w$  spații de interese selectate (Figura 3).

$$ms = \frac{\sum_{j=1}^w ss_j}{w} \quad (6)$$

Similaritatea este calculată folosind formula distanței Gower care normalizează coeficienții la intervalul  $[0, 1]$ , și, de asemenea, folosind principiul calculului indicelui Jaccard, care exclude din

calcul valorilor  $P_i$ ,  $Q_i$  atunci când acestea sunt ambele valori nule, ca în caz contrar, lipsa unei anumite proprietăți pentru două entități ar genera creșterea valorii coeficientului de similaritate.

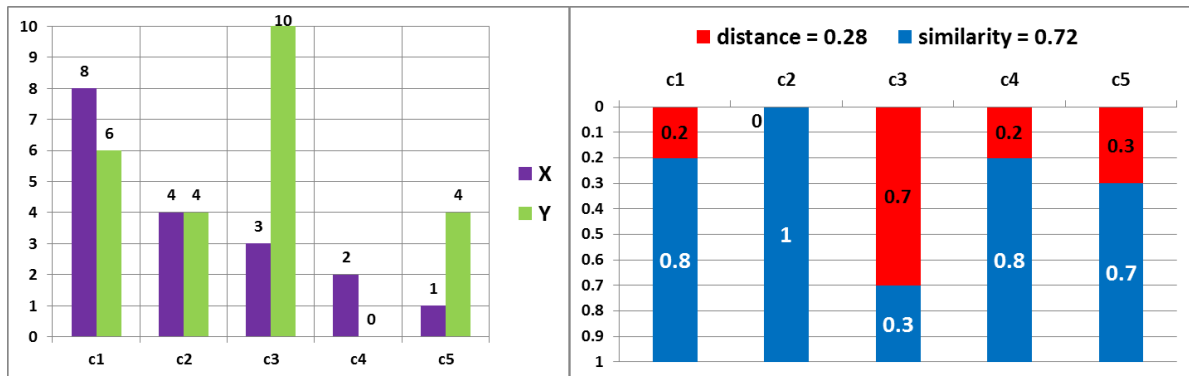


Figure 3 Maparea și similaritatea

Pentru a încheia prezentarea metricei, principalul avantaj al abordării MultiCoS este urmărirea entităților cu interese similare, precum și posibilitatea de a căuta entități care sunt potrivite ca și profilele unor spații de interese specifice într-un anumit nivel măsurabil.

### 3.6 Managementul și specificarea intereselor

Spațiile de interese utilizate în secțiunea anterioară dovedesc diversitatea intereselor care pot fi legate de diferite entități. Putem menționa aici mai multe dintre ele, adăugând, de asemenea, multe altele.

Adaptând template-ul de document AORA la formatul abordare MultiCoS vom obține un meta-spațiu de interese ca un document șablon care înregistrează informațiile.

### 3.7 Procesul MultiCoS de investigare a similarităților

În această secțiune este prezentat un proces care relaționează entități din același spațiu problemă și în cadrul căruia se va defini un proces de constatare a similarităților. După ce astfel de conexiuni sunt definite, stabilite și ponderate, printr-un proces similar, vom crea legături între diversele tipuri de entități (cum ar fi cerințele sau modulele de cod) și vom putea să reutilizăm cod, să accesăm artefacte de la o aplicație existentă pentru dezvoltarea unei alte aplicații. Procesul prezentat în această secțiune descrie abordarea MultiCoS pentru analiza intereselor, în timpul ciclului de dezvoltare a unui produs software, cu o atenție deosebită dată ingineriei cerințelor.

Procesul MultiCoS va include activități ca: , managementul intereselor, managementul cerințelor, transformarea spațiului de cerințe în spațiu de entități, maparea entităților la interese, maparea entităților la entități din același spațiu, calcularea coeficienților de similaritate pentru entitățile în spații diferite.

Artefactele generate sunt documente și modele bazate pe XML. Pentru a realiza investigarea similarității cerințelor, va trebui să se urmeze procesul din Figura 4. Procesul de mapare al cerințelor sau a altor tipuri de entități la spațiile de interese este reprezentat în Figura 6.

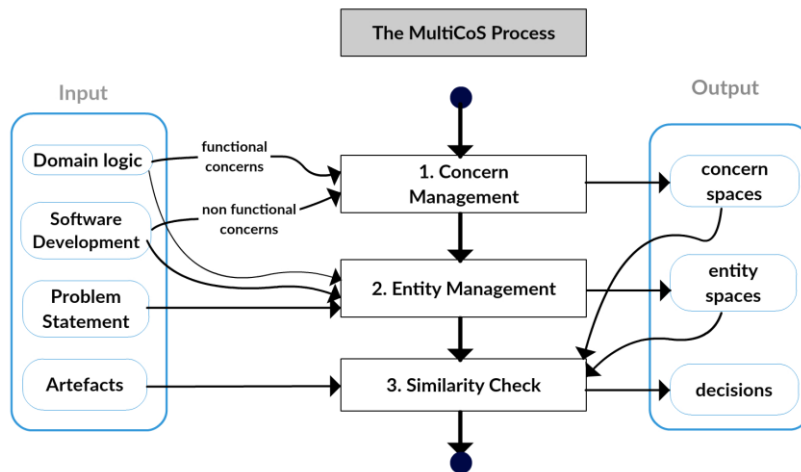


Figure 4 Procesul de mapare al cerințelor în spațiile de interese în MultiCoS

Având în vedere că șablonul folosit pentru cerințe este implicit *requirement* class de la SysML, o cerință va avea următoarele atribute: *name*, *text*, *id*, *derived*, *derivedFrom*, *satisfiedBy*, *refinedBy*, *tracedTo*, and *verifiedBy*.

Clasa *Entity* va fi definită cu următoarele atribute: *id*, *name*, *description*, *sourceId*, *sourceType*, *derivedFromId*.

Pentru fiecare cerință este creat un singur element entitate, și nu vor fi entități create, care să fie lăsate neatribuite unor elemente, cum ar fi cerințele. Relația dintre clasa *requirements* și clasa *entity* este "unu-la-unu". Clasa *entity* conține un număr limitat de câmpuri, dar suficient pentru a capta specificul entității reprezentate. În acest fel, informații pertinente despre elementul inițial vor fi înregistrate în entitate. În cazul cerințelor, următoarele atribute ale clasei *requirements* vor fi copiate (Figura 7): *name*, *text*, *id*, și *derivedFrom*. Aceste atribute vor fi stocate în câmpurile *name*, *description*, *sourceId* și respectiv *derivedFromId* ale clasei *Entity*. De asemenea, atributul *sourceType* al entității va trebui să aibă valoarea "requirement", pentru că entitatea înregistrată este o *cerință*. Astfel, informații cu privire la *spațiu problema* vor fi înregistrate.

Maparea va fi stocată într-un model diferit, în clasa *Relation*, și va înregistra informații cu privire la două obiecte mapate: *Entity* și *Concern*. Clasa *Relation* este proiectată să rețină rezultatul mapării pentru fiecare element rezultat din funcția de mapare  $f$  introdusă în capitolul anterior, pentru modelarea relațiilor dintre entități și interese.

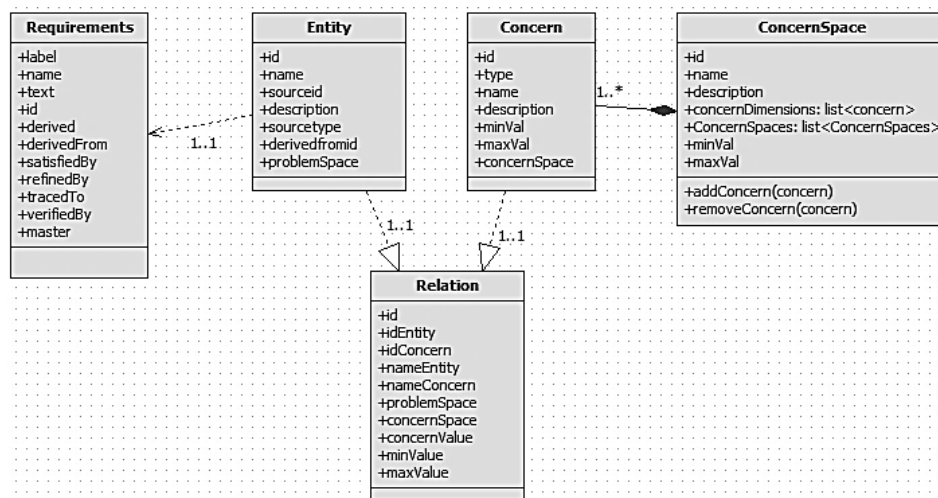


Figure 5 Dependența dintre clase în timpul procesului de transformare



Cerințele și interesele sunt legate și relații sunt stabilite între cele două. Rezultatul mapării este stocat într-un fișier XML .

Cu cât coeficientul de similaritate al multi-spațiului este mai mare cu atât este mai probabil ca să se poată reutiliza artefacte de la o entitate la o entitate similară; sau putem crea artefacte care au în vedere similaritatea, astfel încât mai târziu să putem adapta și personaliza aceste artefacte la specificul fiecărei entități.

### 3.8 Abordarea MultiCoS în sprijinul reutilizării artefactelor

Pentru a putea reutiliza artefacte, cum ar fi module de cod, mai întâi trebuie să relaționăm entități din două probleme diferite. De asemenea, trebuie să avem disponibile artefacte, module de cod pentru a putea crea o relație între entități cum ar fi cerințe și artefacte sau module de cod, ale uneia din problemele existente.

Procesul de relaționare a entităților, cum ar fi cerințele, la artefacte cum ar fi module de cod dintr-un alt spațiu problemă, este descris mai jos:

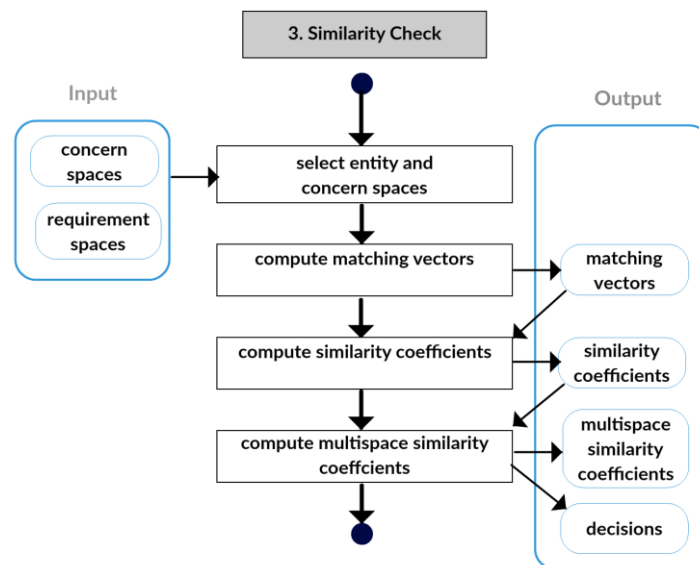


Figure 6 Verificarea Similarității pentru reutilizarea codului și a artefactelor în MultiCoS

Aplicând o tranzitivitate asupra rezultatelor din Etapa 1 și Etapa 2, putem obține în Etapa 3, maparea de la cerințe la module cod din spații de probleme diferite. Investigarea acestor mapări și a coeficienților de similaritate multi-spațiu găsiți în Pasul 1, poate ajuta în a decide cu privire la reutilizarea artefactelor în timpul etapei 4.

Ca parte a procesului, vom crea o mapare de la cerințe la modulele de cod din același spațiu de problemă.

În scopul de a remapa entități de la un spațiu problema altă problemă vom folosi maparea dintre spațiu, unde trebuie să investigăm coeficienții de similaritate multispațiu dintre cele două spații. Rapoartele disponibile propun astfel de translații de la entități dintr-un spațiu problemă la entități dintr-un alt spațiu problemă. În procesul de remapare, diferite spații entitate selectate pot fi selectate și calculul similarității poate fi realizat.

Procesul produce artefacte, cum ar fi rapoarte, care conține toate detaliile cu privire la mapare și este util în procesul decizional al etapei următoare, când este cercetată posibilitatea de reutilizare a artefactelor existente în implementarea unei noi aplicații software.

## 4. Tool-ul MultiCoS

Această secțiune prezintă tool-ul MultiCoS, destinat gestionării cerințelor similare în dezvoltarea de aplicații web. MultiCoS este o abordare de tip AORE derivată din abordarea Multi Concern Space (MCSA), și care permite calculul coeficienților de similaritate pentru diferite perechi de cerințe.

### 4.1 Arhitectura aplicației

Instrumentul MultiCoS este o aplicație web cu arhitectura MVC. Datele sunt stocate în fișiere XML. Logica aplicației are cinci etape principale (Figura 7): (1) utilizatorul trimite cererea, (2) Cererea este direcționată către controlerul adecvat, (3) controlerul interacționează cu modelul de date, (4) controlerul invocă vizualizarea (View) pentru a afișa rezultatul, și (5), vizualizarea este redată în browser-ul web.

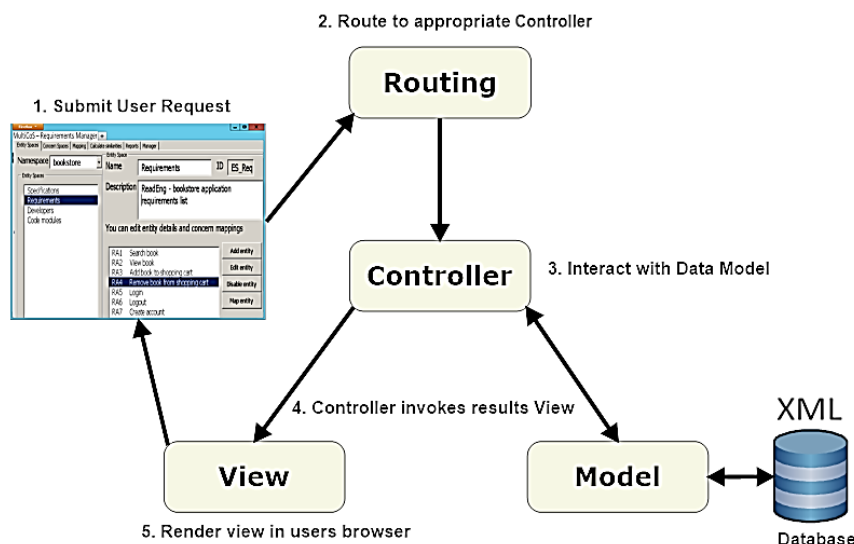


Figure 7 Arhitectura tool-ului MultiCoS

### 4.2 Stocarea și modelul de date

Instrumentul MultiCoS poate importa cerințele de la alte instrumente, atât timp cât datele sunt înregistrate în format XML și definiția schemă este mapată pentru a facilita importul fișierului XML. Instrumentul MultiCoS surprinde cerințele într-o formă simplificată a Templateului Volere. Toate datele sunt stocate în fișiere XML.

Instrumentul MultiCoS poate înregistra spații de interes în formatul stabilit și oferă suport de bază pentru activitățile spațiilor de interes și ale dimensiunilor acestora, cum ar fi: *create*, *edit*, *duplicate*, *remove*, *import* pentru spații de interes, și *add*, *edit*, *copy*, *disable*, and *duplicate* pentru interese.

În procesul de mapare, funcția de mapare  $f$  are două argumente, o entitate din spațiul entitate și un spațiu de interes. Pentru fiecare dimensiune o valoare poate fi atribuită pentru mapare ca să descrie corelația existentă între entitate și dimensiunea la care se referă.

## 4.3 Reportarea și GUI

Interfața cu utilizatorul a instrumentului gestionează procesul de MultiCoS, de la înregistrarea și încărcarea de date legate de diferite proiecte (de exemplu cerințe) în modelul de date al aplicației, la prelucrarea și furnizarea de rapoarte, tabele de indici de similaritate și exporturi XML cu datele existente.

Interfața cu utilizatorul este alcătuit din șase vizualizări (views), care gestionează diferite părți ale modelului de date și procesul MultiCoS: a) spații de entități, b) spații de interese, c) mapare, d) calculul similarității, e) generarea de rapoarte, și f) managerul de tranziție.

*Entity Spaces View* gestionează cerințe și entități și înregistrarea lor în format XML. La fel ca și în proiecte de programare, spațiile de entități sunt parte dintr-un *namespace*, și pentru fiecare spațiu entitate pot fi efectuate operațiunile de CRUD. Entitățile nu sunt șterse, fiind doar cu dezactivate.

Cerințele pot fi înregistrate folosind orice instrument RE, ca, apoi cerințele să poată fi importate în instrumentul MultiCoS, în urma unei proceduri de import. De asemenea, cerințele pot fi înregistrate direct în instrumentul MultiCoS.

*Concern Spaces View* gestionează interesele și spațiile de interese, într-un mod similar cu *Entity Spaces View*. Un mecanism de înregistrare al versiunilor este implementat în spațiile de interese: versiuni diferite sunt înregistrate, cum ar fi, în cazul în care sunt efectuate actualizări cu privire la spații, maparea trebuie refăcută cu noile valori ale spațiului de interese.

*Mapping View* creează mapările de la cerințe la interese. Pentru fiecare cerință în spațiul entități selectat și pentru fiecare interes din spațiul de interese selectat, o valoare de relaționare este alocată. Entitățile și interesele sunt legate și relații sunt stabilite între acestea. Rezultatul mapării este stocat într-un fișier XML. Fișierul este actualizat dacă procesul este repetat, sau alte mapări sunt efectuate.

*Calculate Similarities View* este fereastra în care utilizatorii pot configura și calcula similarități personalizate pentru cerințele din diferite aplicații. Prin selectarea celor două spații de entități, o matrice de adiacență este generată pe baza legăturile de mapare stabilite dintre acestea și spațiile de interes. Procesul de calculare a coeficienților de similaritate se efectuează în conformitate cu metoda prezentată în capitolul 3. O matrice de adiacență este creată pentru a stoca coeficienții de similaritate multispații dintre cele două spații de entități.

Instrumentul MultiCoS oferă o vizualizare a matricei de adiacență și o altă matrice cu calculele detaliate ale coeficienților spațiu de similaritate pentru fiecare entitate.

Instrumentul MultiCoS oferă posibilitatea de creare de rapoarte, ca parte din *Reports View* care ajută utilizatorul în investigarea similarității, decizia de reutilizare și procesul de reutilizare al artefactelor. *Reports View* afișează raportul de similaritate al celor două spații de cerințe, pentru setul de spații de interese mapate care au fost selectate pentru raportare. Raportul de similaritate este prezentat în trei moduri vizuale: sub forma unei matrici de adiacență, ca un graf bipartit, și, de asemenea, ca o listă sortată. Un prag poate fi setat, astfel încât toți coeficienții cu valori mai mari decât pragul respectiv sunt evidențiați. Ambele spații de entități investigate sunt mapate în același

multispațiu de interese. Pentru entitățile cu un coeficient ridicat de similaritate, putem decide să reutilizăm cod și să adaptăm mai târziu unele dintre artefacte.

*Manager View* mapează trecerea de la cerințele dintr-o aplicație la module de cod dintr-o altă aplicație. Un raport tranzitiv de mapare poate fi generat aici și furnizează legături către modulele de cod care sunt sugerate a fi reutilizate în implementarea cerințelor noii aplicații.

Funcționalitatea MultiCoS include, de asemenea facilități de export pentru fișierele XML utilizate pentru a stoca modelul de date, precum și mai multe rapoarte de mapare și coeficienții de similaritate.

## 4.4 Utilizarea tool-ului MultiCoS în procesul de reutilizare de cod

Instrumentul MultiCoS urmează procesul MultiCoS pentru reutilizarea codului și a artefactelor. Fluxul de lucru al toolului va fi prezentat în această secțiune, împreună cu un exemplu realizată cu ajutorul toolului MultiCoS.

Procesului de corelare a cerințelor din diferite spații problemă constă în 4 pași: (1) Verificarea similarității dintre spații de entități (cerințe) și spații de probleme diferite, (2) Maparea spațiului de entități (cerințe) pe un spațiu de interese (module de cod) în același spațiu problemă, (3) remaparea spațiului de entități (cerințe) la un spațiu de interese (module de cod) dintr-un spațiu problemă diferit folosind coeficienții de similaritate, (4) investigarea coeficienților de similaritate (decizia de a reutiliza module de cod sau alte artefacte).

Modul în care instrumentul sprijină aceste sub-procese va fi detaliate în următoarele sub-secțiuni. Sub-procesele care fac parte din fluxul instrumentului MultiCoS sunt: (1) transformarea specificațiilor cerințelor în entități prin transformări model-la-model, (2) stabilirea de legături între cerințele în diferite spații de probleme, (3) specificarea intereselor - transformarea modulelor de cod în interese prin transformări model-la-model, (4) stabilirea unor legături de la cerințele la module de cod în același spațiu problemă, (5) stabilirea de legături conceptuale între spațiu de cerințe dintr-o problemă și module cod din alt spațiu problemă folosind coeficienți similaritate și mapări existente, și, (6) investigarea coeficienților de similaritate între entități pentru a decide reutilizarea de artefacte sau de module de cod.

Fluxul de proces pentru determinarea coeficienților de similaritate între două spații de cerințe folosind instrumentul MultiCoS este prezentat în continuare. Când două cerințe din diferite spații de cerințe au o valoare mare a coeficientului de similaritate pe un spațiu sau o valoare mare a coeficientului de similaritate multi-spațiu, putem accesa artefacte din alte probleme și să le reutilizăm. Devreme ce au fost stabiliți coeficienți de similaritate între spațiile celor două cerințe, o greutate este atribuită relației. Fluxul poate fi aplicat pentru fiecare cerință în fiecare spațiu cerință în relație cu un anumit multi-spațiu.

În procesul de remapare, în tabul *Manager*, putem selecta cele două spații entitate pentru care a fost efectuat calculul similarității. Vom alege spațiul de interese pentru care se face remaparea, și, de asemenea, din lista de similaritate, perechile pentru care se efectuează tranzitivitatea mapării. Pentru selecția făcută, este generată o mapare tranzitivă, și în acest fel, cerințele unei aplicații sunt mapate la modulele de cod ale unei alte aplicații.

Raportul creat în această etapă, conține toate detaliile cu privire la mapare și poate fi utilizat în procesul de decizie de la pasul următor, atunci când este cercetată reutilizarea artefactelor existente pentru a dezvolta o nouă aplicație.

Folosind raportul generat în tabul *Manager*, o investigație mai atentă poate fi desfășurată pe similaritatea dintre cele două spații de entități și posibilitatea reutilizării de artefacte pe baza mapării tranzitive create. În plus față de informațiile afișate în tabul *Manager*, instrumentul MultiCoS, poate genera un raport complet. Detaliile de mapare și investigarea coeficienților de similaritate pot ajuta utilizatorul în a decide asupra reutilizării de artefacte de la o aplicație existentă la una nouă.

## 5. Utilizarea MultiCoS

Un studiu de caz de utilizare al abordării MultiCoS a fost realizat. Scenariul este necesitatea dezvoltării unei aplicații web. Pe lângă resursele obișnuite în scopul de a dezvolta produsul, o aplicație web deja dezvoltată este disponibilă cu toate artefactele sale mapate folosind abordarea MultiCoS. Scopul este de a reutiliza documente și codul din aplicația web existentă, pentru a sprijini dezvoltarea noii aplicații.

Fiind stabilit acest lucru; vom lua în considerare două aplicații web diferite, Aplicația A, o aplicație Web E-commerce de vânzarea de cărți, și, Aplicația B, o aplicație web multimedia pentru ascultarea de muzică on-line.

În aplicația A, utilizatorul poate căuta și vizualiza cărți în magazin, el poate adăuga / elimina cărți în coșul de cumpărături, și el poate începe procesul de cumpărare cărți dacă el este conectat cu contul utilizator. Să considerăm că numele aplicației este *ReadEng*.

În aplicația B, utilizatorul poate căuta și asculta un cântec, el poate adăuga / elimina melodii într-o listă de redare, și el poate asculta lista de redare numai în cazul în care este conectat cu contul de utilizator. Să considerăm numele acestei aplicații existente, deja implementate, *MyTuneCaster*.

Am preluat și organizat cerințele din cele două probleme, o parte fiind redte mai jos:

Table 2 Spațiile de cerințe din cele două aplicații *ReadEng* și *MyTuneCaster*

Problem P-A - <i>ReadEng</i>		Problem P-B – <i>MyTuneCaster</i>	
Id	Requirement	Id	Requirement
RA1	Search book	RB1	Search song
RA2	View book	RB2	Listen song
RA3	Add book to shopping cart	RB3	Add song to playlist
RA4	Remove book from shopping cart	RB4	Remove song from playlist
RA5	Login	RB5	Login
RA6	Logout	RB6	Logout
RA7	Create account	RB7	Create account
RA8	Logged user can buy products in shopping cart	RB8	Logged user can listen songs in playlist

### 5.1 Toolul MultiCoS utilizat pentru a sprijini reutilizarea codului

Aplicațiile sunt destul de diferite, de la servicii furnizate, la potențialii clienți. Totuși, există anumite funcționalități comune în cele două aplicații. Utilizatorii ambelor aplicații pot căuta conținut, pot deschide / vizualiza un articole (carte / melodie), pot gestiona o listă de articole, și amândoi pot folosi articole din lista în scopul specific propus, în cazul în care utilizatorii sunt înregistrați pe site. Vom verifica dacă cerințele sunt similare, folosind abordarea MultiCoS.

Unele spații de interese luate în considerare pentru acest studiu de caz sunt menite să descrie proprietățile standard ale diferitelor tipuri de entități, pentru orice tip de sisteme software; altele sunt specifice aplicațiilor web. În acest studiu de caz, vom utiliza de spații de interese adecvate cerințelor.

Utilizând instrumentul MultiCoS, putem obține o mapare de la spațiul de cerințe la spațiul de module de cod din același spațiu al problemă. Procesul este similar cu cel de mapare al cerințelor dintr-un spațiu la cerințele într-un alt spațiu diferit.

Modulele de cod pot fi înregistrate direct sub forma spațiilor de interese.

Apoi, vom crea o mapare de la cerințe și module de cod în din același spațiu problemă. Utilizând instrumentul MultiCoS, putem efectua procesul de mapare.

Raportul disponibil în tabul Rapoarte propune o traducere de la spațiu de cerințe al unei probleme la spațiu de cerințe al unei alte probleme și ne ajută să investigăm coeficienți de similaritate multispațiu din cele două spații de cerințe. O investigație va ajuta remaparea de la spațiul de cerințe al unei probleme la spațiul modulelor de cod al celeilalte aplicații.

În procesul de remapare, în tabul *Manager*, vom selecta cele două spații de cerințe pentru care a fost efectuat calculul similarității. Aici, vom alege spațiul de interes pentru care se face remaparea, și, de asemenea, din lista cu similarități și perechile pentru care se efectuează maparea tranzitivă. Pentru selecția făcută maparea tranzitivă este realizată (Figura 8).

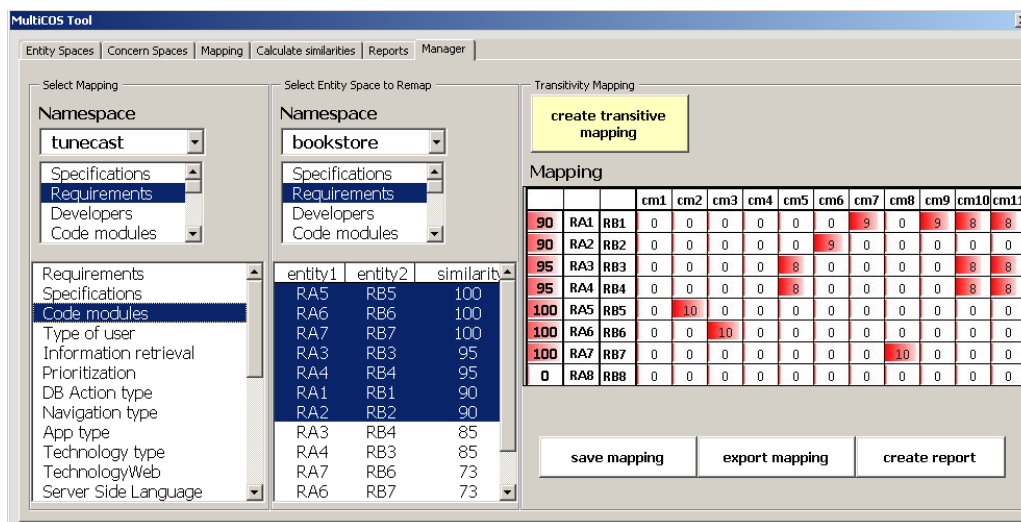


Figure 8 Maparea și crearea de rapoarte în tool-ul MultiCoS tool

Raportul generat de aici conține toate detaliile cu privire la mapare și este util în investigarea reutilizării de module de cod existente în dezvoltarea unei noi aplicații.

Odată generat raportul în tabul *Manager*, putem investiga mai îndeaproape similaritatea dintre cele două spații de cerințe și posibilitatea reutilizării de module de cod bazate pe maparea tranzitivă creată. Detaliile disponibile în raportul generat ajută în procesul de decizie. Raportul prezentat în tabelul 3 are mai multe informații cu privire la maparea modulelor de cod.

Table 3 Detalii ale raportului de mapare tranzitivă generat de tool-ul MultiCoS

Requirement	Code module	Details
RA5: Login		
Mapping value: 10	cm2: doLogin()	user login file: <a href="#">Login.php</a> , <a href="#">index.php</a> , <a href="#">User.pho</a>
RA6: Logout		
Mapping value: 10	cm3: doLogout()	user logout file: <a href="#">Login.php</a> , <a href="#">index.php</a> , <a href="#">User.pho</a>
RA7: Create account		
Mapping value: 10	cm8: createAccount()	creates an user accountfile: <a href="#">register.blade.php</a> , <a href="#">main.blade.php</a> , <a href="#">User.pho</a>
...	...	...

Detaliile furnizate de mapare pot ajuta utilizatorul în a decide asupra reutilizării modulelor de cod în dezvoltarea noii aplicații (Figura 9).

Transitivity Mapping Report			
Mapping Process performed by MultiCOS Requirements tool at 12-04-2014 at 16:43:22			
Source Entity Space	Namespace: tunecast   ID: RB   Name: Requirements		
Concern space map	Namespace: tunecast   ID: CS_03   Name: Code modules		
Destination Entity Space	Namespace: bookstore   ID: ES_Req   Name: Requirements		
Concern Multispace Selected	[CS_04 : Type of user] [CS_05 : Information retrieval] [CS_07 : DBAction type] [CS_15 : Data format] [CS_16 : MVC] [CS_17 : NonFunctional req]		
Multispace Similarity Coefficient			
Destination Entity Space	ID	Similarity Coefficient	Source Entity Space
Search book	RA1	90	RB1 Search song
View book	RA2	90	RB2 Listen song
Add book to shopping cart	RA3	95	RB3 Add song to playlist
Remove book from shopping cart	RA4	95	RB4 Remove song from playlist
Login	RA5	100	RB5 Login
Logout	RA6	100	RB6 Logout
Create account	RA7	100	RB7 Create account
Logged user can buy products in shopping cart	RA8	0	RB8 Logged user can listen songs in playlist
Transitive Mapping generated			
			cm1 cm2 cm3 cm4 cm5 cm6 cm7 cm8 cm9 cm10 cm11
90	RA1	RB1	0 0 0 0 0 0 9 0 0 8 8
90	RA2	RB2	0 0 0 0 0 0 9 0 0 0 0
95	RA3	RB3	0 0 0 0 8 0 0 0 0 8 8

Figure 9 Raportul de mapare generat de tool-ul MultiCoS

Putem vedea din acest raport că, evident, cerințele de *logare* din cele două aplicații sunt identice și deasemenea cerințele *logout* din cele două aplicații sunt identice, având în vedere că acestea au un coeficient multispațial de similaritate 100, astfel încât să putem reutiliza codul referitor la aceste procese. Cu toate acestea, codul va trebui să fie ajustate pentru a se potrivi în profilul noii aplicații, dar atâta timp cât folosim doar aplicația existentă ca singura sursă de cod similar, și același framework, toate celelalte biblioteci ar trebui să fie comune, și ajustările decod vor fi minore.

De asemenea, evident, cerința *create account* are un coeficient multispațiu de similaritate de 100. În cazul în care cerințele nu prevăd alte detalii, reutilizarea codului pentru *model*, *view* și *controller* se va face cu foarte puține ajustări. În cazul în care există și alte cerințe pentru *create account*, cum ar fi, *folosirea de conturi din rețele sociale*, acest lucru va implica, de asemenea, unele ajustări majore în partea logică de *înregistrare* și *logare*, pentru ca aplicații create de părți terțe să poată fi utilizate și încorporate în noua aplicație. Alte cerințe au similaritate 90-95. Pentru acestea reutilizarea va fi făcută la nivel abstract, cu referire la conceptul care a oferit similaritate și cu modificarea codului care ține de diferența de specific a celor două aplicații (RA3 „add book to shopping cart”, RB3 „add song to playlist”), (RA4, RB4), (RA2, RB2), sau (RA1, RB1).

## 5.2 Utilizarea MultiCoS în investigarea cerințelor din același spațiu

Atunci când se utilizează instrumentul MultiCoS pentru a investiga doar cerințele problemei 1 și de a vedea în ce măsură aceste sunt similare între ele, se observă că cerințele RA3: *Add book to shopping cart* și RA4: *Remove book from shopping cart* sunt similare cu un coeficient de similaritate de 90. Pornind de aici, putem investiga subtaskuri ce pot fi utilizate în ambele.

Astfel de subactivități ce pot fi reutilizate sunt *updating the shopping cart* și *refreshing the view*. Practic, aceste sub-sarcini sunt cele care furnizează similaritatea cerințelor RA3 și RA4.

Table 4 Similarități între cerințele aceleiași probleme

Similarity coefficient	Entity Space 1		Entity Space 1	
	Id	Requirement	Id	Requirement
93	RA5	Login	RA6	Logout
90	RA3	Add book to shopping cart	RA4	Remove book from shopping cart



Singurul lucru care rămâne acum de făcut, este de a oferi o formă cât mai abstractă ca posibil celor două sub-sarcini secundare, astfel încât să poată fi identic incluse în oricare dintre cele două sarcini care le includ.

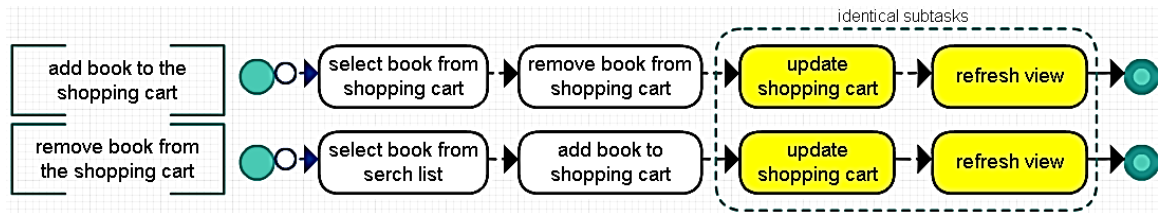


Figure 10 Subtaskuri care pot fi folosite identic ca parte a unor taskuri diferite

În acest fel, abordarea MultiCoS duce la crearea de sarcini abstracte, sprijină reutilizarea resurselor, cum ar fi codul, și documentele proiectului. Deasemenea cerințele transversale pot fi gestionate în acest fel.

### 5.3 Evaluarea MultiCoS

Acest capitol evaluează abordarea MultiCoS în două dimensiuni diferite: cantitative și calitative.

Evaluarea cantitativă este realizată prin aplicarea unui set de metrici orientate pe aspect, care investighează cerințe și specificații din mai multe abordări AORE diferite, inclusiv MultiCoS. Rezultatele obținute pentru abordările alese sunt apoi comparate.

Table 5 Rezultatele aplicării metricii pe abordări AORE

Attributes	Metrics	MultiCoS	EA-Miner	Multi-Dimensional CORE	AOVGraph	AORA
<b>SoC</b>	Concern Diffusion over Artefacts (CDA)	2	3	1	2	1
	Concern Diffusion over Operations (CDO)	5	23	11	12	7
	Concern Diffusions over LOC (CDLOC)	8	11	14	10	9
<b>Coupling</b>	Coupling Between Artefacts (CBA)	4	3	10	3	6
<b>Cohesion</b>	Lack of Cohesion in Operations (LCOO)	0	23	0	24	0
<b>Size</b>	Artefacts (Artefacts)	20	15	18	22	22
	Vocabulary Size (VS)	9	3	2	6	9
	<b>Weighted Operations per Artefact (WOA)</b>	<b>Average: 3.2</b>	<b>Average: 6</b>	<b>Average: 4.4</b>	<b>Average: 8.3</b>	<b>Average: 3</b>

Instrumentele MultiCoS și Aora au cea mai mică valoare a răspândirii intereselor în artefacte în comparație cu celelalte abordări, pentru că artefactele primare ale celor două sunt *interesese*. Deasemenea, abordările MultiCoS și AORA, au o valoare zero pentru metrica de coeziune.

Această situație apare tot deoarece artefactele principale ale acestor două abordări sunt interesese. Evaluarea calitativă este realizată prin selectarea unui set de criterii pentru a compara abordările AORE utilizate în evaluarea cantitativă. În plus, evaluarea este, de asemenea, realizată prin aplicarea abordării MultiCoS la un studiu de caz.

În general, indicele de calitate al MultiCoS este mare (a se vedea tabelul 6), recomandând-o din punct de vedere calitativ. Conform cu analiza cantitativă și calitativă, MultiCoS este o metodă care merită utilizată, îmbunătățind separarea intereselor, modularizarea și înțelegerea în procesul de reutilizare al modulelor de cod și al altor resurse.

Table 6 Controlul calității MultiCoS

<b>Calitate</b>	<b>Indeplinit</b>	<b>Motiv</b>
Traceability	Da	Supported by mapping all system spaces to the same set of concern spaces (multispace)
Modularization of crosscutting requirements	Da	The crosscutting requirements can be encapsulated in custom modules, depending on the multispace considered
Identification of crosscutting requirements	Da	By calculating similarity coefficients
Composition of crosscutting requirements	Da	Using formalized decision trees based on similarity coefficients of requirements
Conflict management	Nu	Similarity coefficients can provide means to solve conflict situations
Tool support	Da	Web based Eclipse Tool
Validation	Partial	Tested on study cases
Mapping crosscutting requirements to later development stages	Da	Performed by mapping requirements to entity spaces defined in the later stages
Maturity	Nu	Not tested in large projects

De asemenea, abordarea MultiCoS ajută gestionarea problemelor transversale, îmbunătățește înțelegerea, promovează trasabilitatea, precum și mentenanța codului și a resurselor aplicației. Totuși, există o nevoie de a testa abordarea MultiCoS pe proiecte mari; o astfel de testare fiind obligatorie pentru a valida MultiCoS.

## Concluzii

---

În această teză a fost prezentată o abordare metodologică pentru gestionarea cerințelor și, de asemenea, un proces specific de aplicațiilor web pentru a lega cerințele de artefacte software. Motivația, declarația problemei au fost detaliate și contribuția principală a fost introdusă pe scurt în prima parte.

În al doilea capitol a fost trecut în revistă stadiul actual al domeniului cercetării în ceea ce privește zona de interes a lucrării. În prima secțiune au fost investigate cerințele; în a doua secțiune metodologiile web au fost comparate și, de asemenea, instrumente de sprijin ale metodologiilor și gestionarea cerințelor. În a treia secțiune au fost analizate metodologiile care implică separarea intereselor și aspectelor cu o concentrare asupra separării multidimensionale a intereselor și metodelor AORE, care au fost la baza acestei lucrări.

În al treilea capitol au fost prezentate primitivele și meta-modelele pentru spațiile multiple de interese utilizate în abordarea MultiCoS. Sunt prezentate de asemenea procesele care duc la identificarea cerințelor și a artefactelor similare. Aceste două procese sprijină decizia de reutilizarea cerințelor, a modulelor de cod sau a altor artefacte și resurse. În primul proces, cerințele din același spațiu problemă au fost relaționate, în timp ce în al doilea proces, cerințele au fost legate de codul compilat al unei aplicații web și un proces de investigare a fost realizat cu scopul de a decide dacă anumite module de cod pot fi refolosite.

În al patrulea capitol, după ce sunt prezentate tooluri folosite în IC, instrumentul MultiCoS este prezentat, iar funcționalitățile acestuia sunt prezentate ca sprijinind procesul MultiCoS prezentat în capitolul al treilea. Caracteristicile de arhitectura ale instrumentului, modelul de date, GUI și modulul de raportare sunt, de asemenea, prezentate în acest capitol.

În capitolul al cincilea, un studiu de caz a fost realizat, folosind abordarea MultiCoS susținută de instrumentul MultiCoS. Cerințele unei noi aplicații a fost verificate dacă sunt similare cu cerințele unei aplicații existente. Pentru că într-o aplicație existentă, a fost efectuată o mapare de la cerințele aplicației la modulele de cod pe care aplicația le implementează, este efectuată o activitate legată de tranziția necesară legării cerințelor unei noi aplicații de modulele de cod ale aplicației web existente. Bazat pe investigarea coeficienților de similaritate, poate fi luată o decizie de reutilizare a anumitor module de cod de la o aplicație deja existentă la o aplicație nouă. La finalul capitolului este făcută o evaluare a abordării MultiCoS.

Pentru a sprijini procesele de dezvoltare bazate pe metodologii de separarea intereselor ar trebui să fie create baze de date cu spații de interese grupate în multispații de interese și spații de entități (spații de sistem).

Pentru a dovedi generalitatea modelului și a sugera acceptarea metodologiei sunt necesare modele de conversie de la tipuri de spații introduse în alte abordări la modelul introdus aici.

Maparea manuală a modulelor de cod și a spațiilor de interese este dificilă și cere timp, așa cum s-au dovedit în [25]. Aceste activități ar trebui să fie asistată de către instrumente specializate și, dacă este posibil, parțial automatizat cu acel instrument specializat. Instrumentul MultiCoS este prezentat ca sprijinind și astfel de activități.

Instrumentul MultiCoS introdus aici sprijină abordarea cu același nume, prin efectuarea eficientă a unor calcule de similaritate complexe și greoaie și prin menținerea unui model de date destul de complex. Procesul sugerat de abordare oferă o soluție formalizată care conduce la reutilizarea de cod. Instrumentul gestionează atât cerințe, grupate în spații de cerințe, cât și interese, grupate în spații de interese. Maparea celor două spații de cerințe diferite asupra același set de spații

de interese este fundamentul calculului coeficienților de similaritate pentru cerințele din cele două spații cerințe.

Instrumentul produce rapoarte cu cerințe similare care se găsesc în diferite aplicații (spații de cerințe) și, de asemenea, oferă o mapare de la un set de cerințe al unei aplicații la module de cod și cerințe dintr-o aplicație deja implementată. Studiul de caz a demonstrat că instrumentul poate ajuta procesul de reutilizarea codului, prin corelarea un set de cerințe cu module de cod dintr-o aplicație deja implementată prin intermediul legăturilor cu cerințe similare.

Idei de dezvoltare ulterioară vor include utilizarea instrumentului în sprijinirea dezvoltării de aplicații mai mari, și, de asemenea, îmbunătățirea funcțiilor de import / export de la formatul intern în diferite formate XML existente utilizate pentru înregistrare de cerințele în UML și Eclipse SysML.

Metodologia MultiCoS pot fi foarte utile pentru a determina similaritate în diferite entități din domenii cum ar fi mediul, medicina, psihologia, în cazul în care modelele și similaritățile sunt investigate din mai multe puncte de vedere, acestea putând fi tratate ca și spații de interese. Modelul poate fi exprimat vizual ca un grafic, pot sprijinii scenarii complexe și este flexibil, de încredere, în ciclul de viață al dezvoltării aplicațiilor.

Impactul acestei abordări pe cerințele aplicației web și de dezvoltare a produsului va fi de asemenea un subiect de cercetare viitoare, având în vedere posibilitatea plasării de informații în codul sursă al aplicațiilor web în scopuri de reverse engineering.

Abordarea prezentată aici este menită să introducă o generalizare a spațiilor de interese și de modelare a separării intereselor. Modelul prezentat are multiple aplicații posibile, cum ar fi ingineria cerințelor, sisteme informaționale din domenii diverse, și reverse engineering.

## Bibliografie

---

- [1] R. Young, *The Requirements Engineering Handbook*, Norwood, MA, USA: Artech House, Incorporated, 2003.
- [2] D. Howe, "Free On-line Dictionary of Computing," 1995. [Online]. Available: [http://dictionary.reference.com/browse/requirements engineering](http://dictionary.reference.com/browse/requirements%20engineering). [Accessed 10 10 2013].
- [3] S. P. Parker, *McGraw-Hill Dictionary of Scientific & Technical Terms*, 6E, The McGraw-Hill Companies, Inc., 2003.
- [4] S. Murugesan, Y. Deshpande, S. Hansen and A. Ginige, "Web Engineering: A New Discipline for Development of Web-based Systems," in *Proceedings of the First ICSE Workshop on Web Engineering, International Conference on Software Engineering*, 1999.
- [5] A. R. P. R. W. Ruzanna Chitchyan, "Semantics-based Composition for Aspect-Oriented Requirements Engineering," in *AOSD IEEE Conference*, 2007.
- [6] A. & E. S. Al-Rawas, "Communication problems in requirements engineering: a field study," in *In Proceedings of the First Westminster Conference on Professional Awareness in Software Engineering*, London, 1996.
- [7] A. K. N. C. C. S. F. P. G. I. G. J. K. G. K. A. M. M. M. S. M. N. B. R. T. R. W. R. J. S. A. S. W. W. M. G. Vallecillo, "MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods. A position paper," in *MDWEnet workshop*, Como, Italy, 2007.
- [8] A. a. W. R. McDonald, "Web Engineering in Practice," in *Proceedings of the 4th Workshop on Web Engineering (in conjunction with 10th Int. Conf. on WWW)*, Hong Kong, 2001.
- [9] D. G. J. I. C. Ross, "Software Engineering: Processes, Principles, and Goals," in *IEEE Computer*, IEEE Computer Society, 1975, pp. 17-27.
- [10] P. O. H. H. S. J. S. Tarr, "N Degrees of Separation: Multi-Dimensional Separation of Concerns," in *21th International Conference on Software Engineering (ICSE'99)*, pp. 107-119, 1999.
- [11] A. M. A. A. J. T. B. B. E. C. P. Rashid, "Early Aspects," 2006. [Online]. Available: <http://www.early-aspects.net/>.
- [12] A. R. A. A. J. Moreira, "Multi-dimensional Separation of Concerns in Requirements Engineering," in *In 13th Requirements Engineering Conference (RE'05)*, Paris, 2005.
- [13] G.-C. C.E.N., "A Multi-Dimensional Separation of Concerns of the Web Application

- Requirements," *Studia Universitatis Babes-Bolyai, Series Informatica*, vol. LVIII, pp. 29-40, 2013.
- [14] J. C. Gower, "Measures of similarity, dissimilarity, and distance," in *Encyclopedia of statistical sciences*, New York, S. Kotz. , Wiley, 1985, pp. 397-405.
- [15] A. M. R. S. Sanford Friedenthal, *A Practical Guide to SysML: The Systems Modeling Language*, 3rd edition, Waltham: OMG, 2015.
- [16] S. R. James Robertson, "Volere Requirements Specification Template," Atlantic Systems Guild, London, UK, 2012.
- [17] L. H. E. T. T. Silva, "Comparing approaches in AORE through ISO/IEC 9126," in *New Trends in Software Methodologies, Tools and Techniques*, Québec, Canada, 2006.
- [18] R. E. T. C. S. A. M. Filman, *Aspect-Oriented Software Development*, Addison-Wesley, 2005.
- [19] IEEE, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE, 2000.
- [20] I. S. S. Brito, *Aspect-Oriented Requirements Analysis - PHd Thesis*, Lisboa: Universidade Nova de Lisboa, Faculdade de Ciência e Tecnologia, 2008.
- [21] G.-C. C.E.N., "Modeling Concern Spaces Using Multi Dimensional Separation of Concern," *International Journal of Computers and Techology*, vol. 11, no. 2, pp. 2302-2313, 2013.
- [22] C. Z. Angela Hantelmann, "Adding Aspect-Oriented Programming Features to C#.NET by using Multidimensional Separation of Concerns (MDSOC) Approach," *Journal of Object Technology*, vol. 5, no. 4, pp. 59-89, 2006.
- [23] G.-C. Calin Eugen Nicolae, "Using Concern Spaces to Measure Requirements Similarities," *Studia Informatica*, vol. LX, no. 1, pp. 35-46, 2015.
- [24] S.-H. Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," *International Journal Of Mathematical Models And Methods In Applied Sciences*, vol. 1, no. 4, pp. 300-307, 2007.
- [25] M. G. C. Lai A., "The Structure of Features in Java Code: An Exploratory Investigation," in *Workshop on Multi-Dimensional Separation of Concerns in Object-Oriented System*, Oopsla, 1999.