

Inginerie Software 2024

Laboratorul 8

Analiza de robustețe.*

Mădălina Erașcu, Alexandru Munteanu, Cristian Pal, Ionica Puiu

April 2024

1 Analiza de robustețe

Ivar Jacobson, Grady Booch și James Rumbaugh au fost cei care au propus inițial o etapă de verificare cu realitatea în care obiectele dintr-un sistem propus sunt împărțite în trei categorii distincte stereotipate cu `<<interface>>`, `<<control>>` și `<<entity>>`. Fiecare obiect din sistem trebuie să aparțină unei singure categorii. În plus, există reguli care definesc interacțiunile permise.

Doug Rosenberg și Kendall Scott descriu în cartea lor “Use Case Driven Modelling – Theory and Practice” această tehnică numită analiza de robustețe.

- Ideea de bază este că se pot analiza pașii unui caz de utilizare cu scopul de a valida logica bussines a acestuia și de a ne asigura că terminologia este consistentă cu cea a altor cazuri de utilizare ce au fost analizate anterior. Cu alte cuvinte, se verifică robustețea cazurilor de utilizare în raport cu reprezentarea cerințelor de utilizare pentru sistemul ce trebuie construit.
- O altă utilizare este identificarea de noi obiecte și responsabilități ce ar putea fi necesare sprijinirii logicii cazului de utilizare.

Analiza de robustețe își are locul între analiză și proiectare. Dacă analiza (i.e. cazurile de utilizare) este “ce” și proiectarea este “cum,” atunci analiza de robustețe este proiectare preliminară. Pe parcursul acestei faze se fac presupuneri preliminare despre modelul de proiectare (design) și începem să ne gândim la arhitectura tehnică și la mai multe strategii de proiectare posibile. Deci facem parțial analiză și parțial proiectare.

Analiza de robustețe joacă mai multe roluri esențiale. Atât detaliile textuale ale cazului de utilizare cât și modelul static sunt rafinate pe parcursul analizei de robustețe. Figura 1 ilustrează bucla de reglare (feedback loop) între *modelul de robustețe* și *modelul static* [2, 1, 4, 3].

*Bazat pe resursele de laborator ale Conf. Dr. Cristina Mândruță

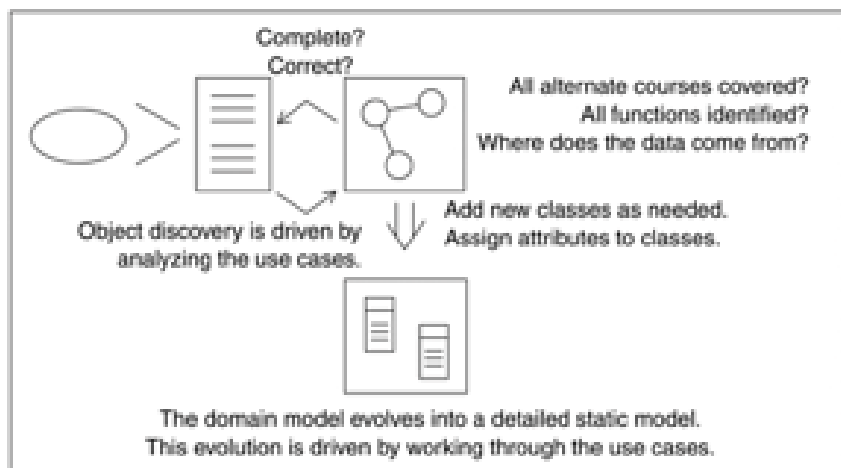


Figure 1: Bucla de reglare (feedback loop) între *modelul de robustețe* și *modelul static*

Elemente cheie ale analizei de robustețe

- Ajută să ne asigurăm că textul cazului de utilizare este corect și că nu a fost specificat comportament nerezonabil sau imposibil în contextul setului de obiecte definit. Această **rafinare a textului cazului de utilizare** schimbă natura acestui text din simplu manual de utilizare în descriere a utilizării în contextul modelului obiectual.
- Oferă o **verificare a completitudinii și corectitudinii** asigurându-ne că sunt adresate în cazurile de utilizare toate căile alternative necesare.
- Permite **descoperirea de obiecte pe parcurs**, lucru important deoarece aproape sigur unele obiecte au fost omise în timpul modelării domeniului. De asemenea, se pot adresa **discrepanțele și conflictele în numirea obiectelor** înainte ca acestea să genereze probleme serioase. Analiza de robustețe ajută, de asemenea, și la identificarea majorității claselor `<<entity>>` și `<<boundary>>` înainte de crearea diagramelor de secvențe.
- Joacă rolul de proiectare preliminară prin acoperirea zonei dintre analiză și proiectare de detaliu.

2 Diagrama de robustețe

2.1 Introducere

O **diagramă de robustețe** este un hibrid între diagrama de clase și diagrama de activitate. Este o reprezentare plastică a comportamentului descris pentru un caz de utilizare, ilustrând atât clasele participante cât și comportamentul

software-lui, deși evită în mod intenționat să arate cum este alocat comportamentul la clase. Fiecare clasă este reprezentată cu o pictogramă stereotipată. O diagramă de robustețe se citește mai curând ca o diagramă de activitate, în sensul că un obiect "comunică" cu următorul obiect. Acest flux de acțiuni este reprezentat printr-o linie între două obiecte care comunică.

Definiție (Analiza de robustețe). *Analiza de robustețe este un proces care implică analizarea textului cazului de utilizare și:*

- *identificarea unui prim set de obiecte care vor participa la fiecare caz de utilizare*
- *clasificarea acestora în obiecte <<entity>>, <<boundary>> și <<control>>.*

După terminarea analizei de robustețe trebuie făcută **revizuirea proiectului preliminar (PDR - Preliminary Design Review)**. Sesiunea PDR se asigură că diagrama de robustețe, modelul domeniului și textul cazului de utilizare sunt sincronizate. Această revizuire face trecerea de la proiectarea preliminară la etapele proiectării de detaliu, pentru fiecare pachet de cazuri de utilizare.

Remarcă. *Există o corespondență directă 1:1 între fluxul de acțiuni din diagrama de robustețe și etapele descrise în textul cazului de utilizare.*

Stereotipurile grafice pentru categoriile de clase din diagrama de robustețe sunt în Figurile 2 și 3).

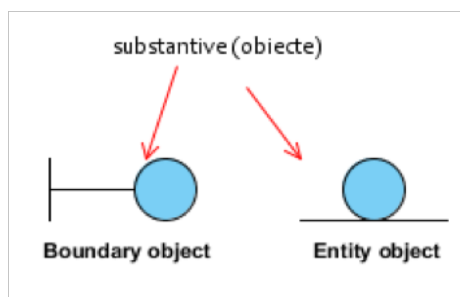


Figure 2: Stereotipurile grafice pentru obiecte de tip <<boundary>> și <<entity>>

Diagrama de robustețe este o punte între cazurile de utilizare și alte modele (vz. Figura 4).

Remarcă. *Diagrama de robustețe este bază pentru crearea diagramelor de secvențe. Permite concentrarea pe interfața utilizator (prin prototipare sau codificare), deoarece include, în mod explicit, elemente de interfață cu utilizatorul (obiecte de tip <<boundary>>).*

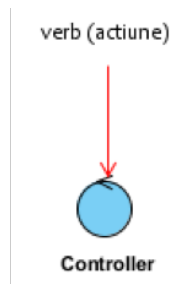


Figure 3: Stereotipurile grafice pentru obiecte de tip <<control>>



Figure 4: Rolul diagramei de robustețe între **ce** și **cum**

2.2 Elemente de modelare

- Obiectele <<boundary>> sunt obiecte cu care actorii vor interacționa (ex. ferestre, ecrane, menu-uri, dar și interfețe cu alte sisteme). Acestea se pot identifica utilizând prototipul GUI sau textul cazului de utilizare.
- Obiectele <<entity>> sunt obiecte ce reprezintă entitățile cu care va lucra sistemul. Pot fi obiecte persistente (reprezintă informații ce supraviețuiesc executării cazului de utilizare) sau obiecte tranzitorii (ex. rezultatele unor căutări). Cele persistente provin în general din modelul domeniului și deseori corespund tabelor din baza de date sau fișierelor ce păstrează aceste informații.
- Obiectele <<control>> sunt obiecte ce includ cea mai mare parte din logica aplicației. Reprezintă elementele de conectare între utilizator și datele memorate. Capturează regulile și politicile business care se schimbă frecvent, în ideea localizării acestora în aceste obiecte fără a afecta interfața cu utilizatorul sau schema bazei de date. Sunt folosite pentru a captura întreaga funcționalitate și comportament necesitate de cazul de utilizare. În aprox. 20% din cazuri, controler-ele devin obiecte de sine stătătoare, în rest, se vor transforma în operații atașate altor obiecte.

Actorii utilizează obiectele <<boundary>> pentru a comunica cu sistemul. Obiectele <<entity>> sunt derivate de obicei din modelul domeniului iar obiectele

<<control>> servesc ca elemente de legătură între obiectele <<boundary>> și obiectele <<entity>>.

2.3 Notății

Actor. Este același concept cu actorul din diagrama cazurilor de utilizare (vz. Figura 5).

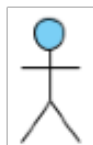


Figure 5: Actor

Boundary (interfață). Reprezintă elemente software de interfațare cu sistemul (exemple: ecrane, rapoarte, pagini HTML, alte interfețe la sistem) (vz. Figura 6).



Figure 6: Obiect de tip <<boundary>>

Controller (proces). Servește ca element de legătură între elementele <<boundary>> și elementele <<entity>>, implementând logica necesară gestionării diferitelor elemente și a interacțiunii dintre acestea. Multe controlere sunt suficient de simple pentru a fi implementate ca operații ale unor clase <<boundary>> sau <<entity>> și nu ca o clasă de sine stătătoare (vz. Figura 7).



Figure 7: Obiect de tip <<control>>

Entity (domain). Tipurile de entități găsite în mod tipic în modelul conceptual ((vz. Figura 8)).



Figure 8: Obiect de tip <<entity>>

Asociere (vz. Figura 9)

Caz de utilizare. Cazurile de utilizare pot invoca alte cazuri de utilizare (vz. Figura 10).



Figure 9: Relație de asociere



Figure 10: Invocarea unui caz de utilizare

2.4 Reguli de comunicare

- **Actorii** pot comunica doar cu clase <<boundary>>.
- Clasele <<boundary>> pot comunica doar cu clase <<control>> și cu **actori**.
- Clasele <<entity>> pot comunica doar cu clase <<control>>.
- Clasele <<control>> pot comunica cu clase <<boundary>>, cu clase <<entity>> și cu alte clase <<control>>.

Descrierea vizuală a regulilor de comunicare este în Figura 11.

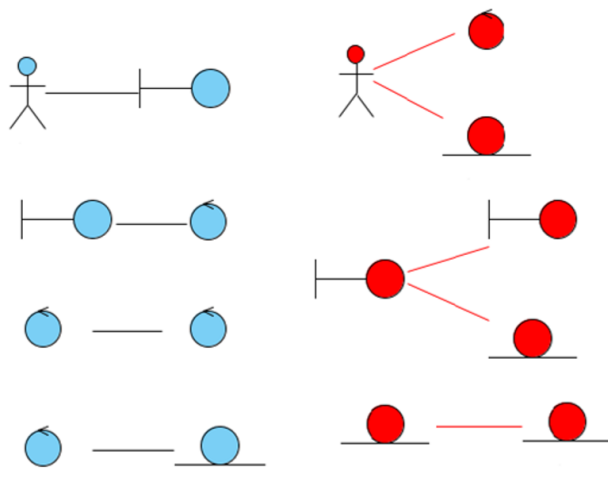


Figure 11: Reguli de comunicare în diagrama de robustețe

Remarcă (Implicarea clientului). *Diagrama de robustețe este directă și ușor de desenat, a.î. poate fi folosită ca tehnică de modelare în colaborare cu membrii echipei și cu unii clienți.*

Remarcă (Renunțarea la modelele temporare). *Diagrama de robustețe se desenează de obicei pe o tablă și este utilizată pentru a analiza logica cazului de utilizare. După ce cazul de utilizare a fost clarificat și completat, ea va fi păstrată doar până la următorul pas în dezvoltarea aplicației.*

Remarcă (Aplicarea artefactelor corespunzătoare). *Se poate renunța la utilizarea diagramei de robustețe după obținerea experienței de a face saltul logic de la cazuri de utilizare la diagrame de secvențe.*

3 Metodologie pentru dezvoltarea diagramei de robustețe

Analiza de robustețe se realizează pentru fiecare caz de utilizare prin parcurgerea textului **cazului de utilizare**, propoziție cu propoziție, și desenarea actorilor, obiectelor <<boundary>>, <<entity>> și <<control>> corespunzătoare și a conexiunilor dintre diferitele elemente ale diagramei. Se va realiza o singură diagramă ce va include atât fluxul principal cât și fluxurile alternative.

Utilizând descrierea textuală a cazului de utilizare:

- Se adaugă câte un element <<boundary>> pentru fiecare element de interfață utilizator major (ex. ecran, raport).
- Se adaugă un element <<control>> pentru fiecare activitate relevantă.
- Se adaugă un element <<entity>> pentru fiecare concept business.
- Se adaugă un caz de utilizare ori de câte ori un caz de utilizare este inclus în scenariu.

4 Exemple

4.1 Exemplul 1

Text CAZ DE UTILIZARE
<p>Nume: Afișează detalii carte</p> <p>Identificator: UC 19</p> <p>Descriere: Permite explorarea listei de cărți și afișarea detaliilor pentru cea selectată.</p> <p>Precondiții: Clientul este logat în sistem.</p> <p>Postcondiții: Niciuna</p> <p>Flux principal de acțiuni: Clientul tipărește URL-ul pentru pagina Home a librăriei. Sistemul afișează o listă de cărți din catalog pe pagina Home, sub formă de link-uri selectabile. Clientul selectează un link de pe pagina Home. Sistemul extrage detaliile cărții selectate și le afișează pe pagina Detalii Carte.</p> <p>Flux alternativ A: CarteNegasita A.1. Sistemul afișează pagina CarteNegasita.</p>

Flux principal de acțiuni:

1. Clientul tipărește URL-ul pentru pagina Home a librăriei (Figura 12).



Figure 12: Exemplul 1, flux principal, pas 1

2. Sistemul afișează o listă de cărți din catalog pe pagina Home, sub formă de link-uri selectabile (Figura 13). Sistemul:
 - inspectează catalog,
 - extrage (posibil conform unui criteriu) o listă de cărți
 - afișează linkurile cărților din această listă.

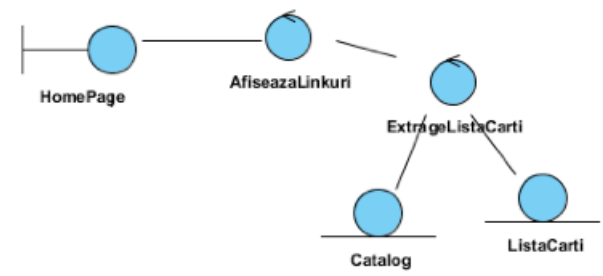


Figure 13: Exemplul 1, flux principal, pas 2

3. Clientul selectează un link de pe pagina Home (Figura 14). **selectează link** este un *eveniment* la care sistemul răspunde; *nu* este o acțiune de sistem.
4. Sistemul extrage detaliile cărții pentru cartea selectată și le afișează pe pagina DetaliiCarte (Figura 15).

Flux alternativ A: Carte negăsită

A.1. Sistemul afișează o pagină CarteNegasita.

În Figura 17 este prezentată diagrama de robustețe completă.

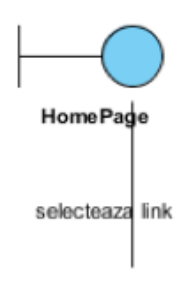


Figure 14: Exemplul 1, flux principal, pas 3

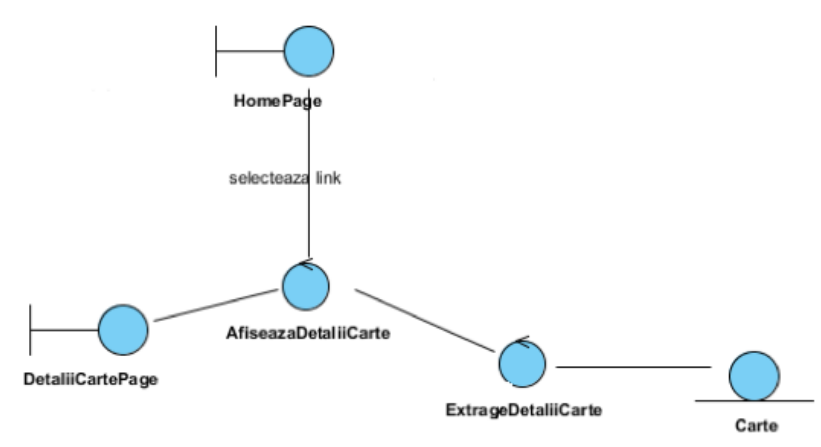


Figure 15: Exemplul 1 - Pasul 4

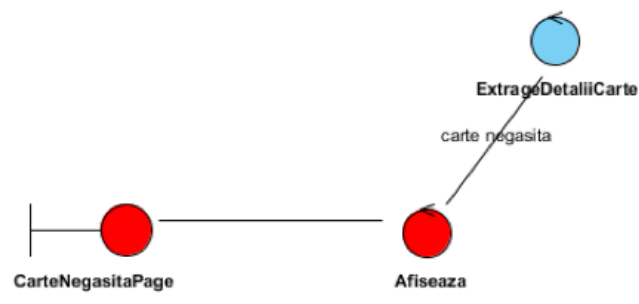


Figure 16: Exemplul 1, flux alternativ

4.2 Exemplul 2

Lista cerințelor pentru o librărie accesibilă pe Internet:

- Librăria va accepta ordine prin Internet.

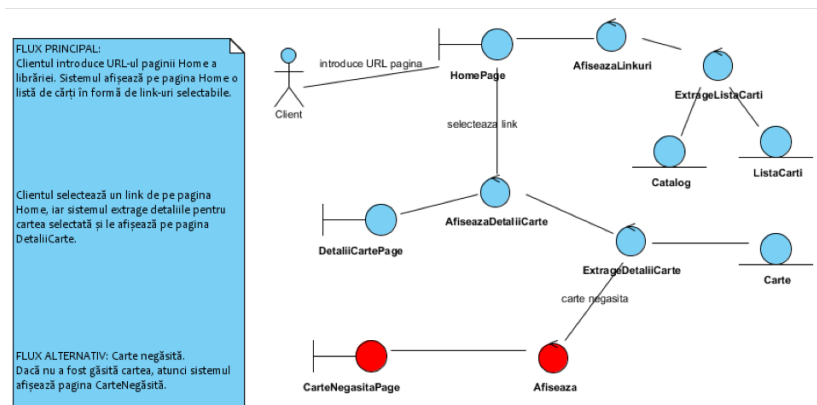


Figure 17: Exemplul 1 - Diagrama de robustețe

- Librăria va menține o listă de conturi pentru max. 1,000,000 clienți.
- Librăria va oferi protecție pe bază de parolă pentru fiecare cont.
- Librăria va oferi posibilitatea de a căuta într-un catalog central de cărți.
- Librăria va oferi mai multe metode de căutare în catalog, incluzând căutare după autor, după titlu, după ISBN, și după cuvânt cheie.
- Librăria va oferi un mijloc securizat de realizare a plăților cu card de credit de către clienți.
- Librăria va oferi un mijloc securizat de plată via ordin de plată.
- Librăria va oferi link-uri electronice între Web și baza de date și sistemul de livrare.
- Librăria va oferi link-uri electronice între Web și baza de date și sistemul de gestiune a stocurilor.
- Librăria va păstra recenzii ale cărților și va permite oricui să încarce recenzii pe site.
- Librăria va menține clasamente ale cărților, bazate pe intrările de la clienți.

Diagrama cazurilor de utilizare pentru sistemul librărie online

Remarcă. În diagrama de cazuri de utilizare (Figura 18) s-a omis intenționat reprezentarea grafică a sistemului.

Diagrama claselor de domeniu este prezentată în laboratorul corespunzător.

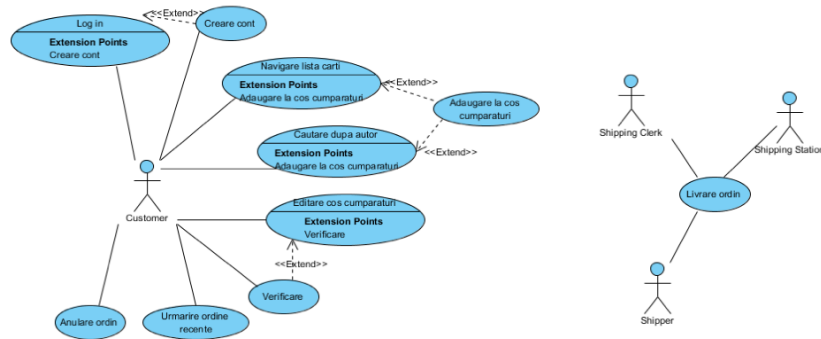


Figure 18: Diagrama cazurilor de utilizare

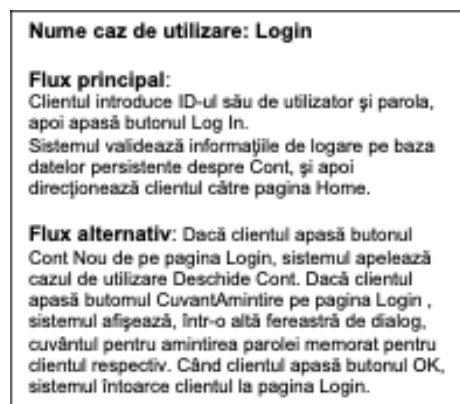


Figure 19: Descriere Login

Diagramele de robustețe sunt prezentate în Figurile 20, 22 pornind de la descrierile cazurilor de utilizare 19, 21.

5 Recomandări pentru analiza de robustețe

1. Respectați regulile substantiv/verb pentru diagrama de robustețe : substantivele (`<<boundary>>` și `<<entity>>`) pot comunica doar cu verbe, iar verbele (`<<control>>`) pot comunica și cu substantive și cu alte verbe. Aceste reguli impun textului un format de tip *substantiv-verb-substantiv* a.î. să nu alocați comportament obiectelor înainte de a avea suficiente informații pentru o decizie bună de proiectare. Regulile pentru obiectele `<<boundary>>` au rolul de a asigura că vor fi specificate explicit granițele dintre sistem și actorii din contextul lui.
2. Utilizați analiza de robustețe pentru a obține un format consistent al textului cazului de utilizare. Șablonul *boundary-control-entity* apare în ma-

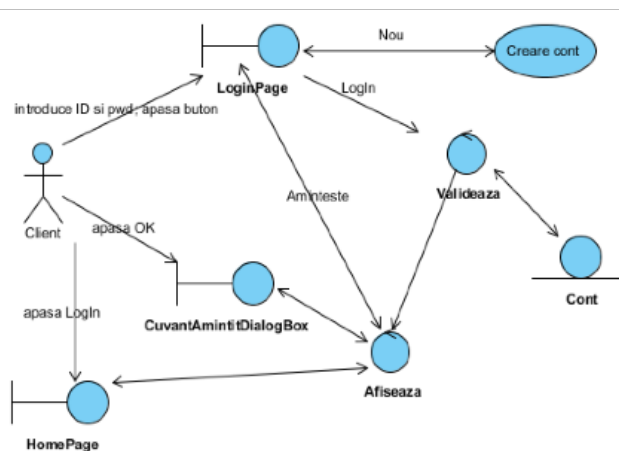


Figure 20: Diagrama de robustețe Login

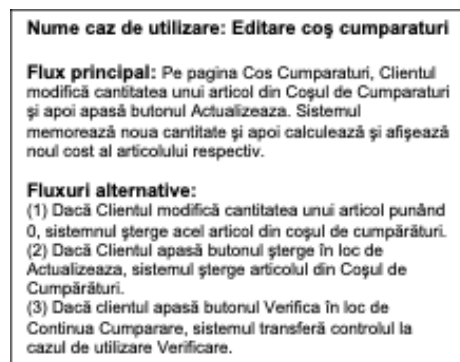


Figure 21: Descriere Editare coș cumpărături

joritatea diagramelor de robustețe. Acesta este strâns corelat cu șablonul *subiect-verb-obiect* al propozițiilor din limbajul natural. Trebuie să folosiți diagrama de robustețe pentru a obține un ansamblu cât mai consistent al textelor care descriu cazurile de utilizare, ceea ce va duce la sporirea lizibilității și mentenabilității acestora.

3. Includeți fluxurile alternative pe diagrama de robustețe. Analiza de robustețe trebuie realizată pe întreg textul cazului de utilizare, nu doar pe fluxul principal. Mare parte din comportamentul interesant al sistemului are loc în contextul fluxurilor alternative, deci este foarte importantă analizarea acestui comportament în timpul efortului de modelare. Analiza de robustețe poate ajuta, de asemenea, la descoperirea de noi fluxuri alternative, în special când etichetați controlerele cu *verifica* sau *valideaza*.
4. Utilizați analiza de robustețe pentru a asigura consistență între numele

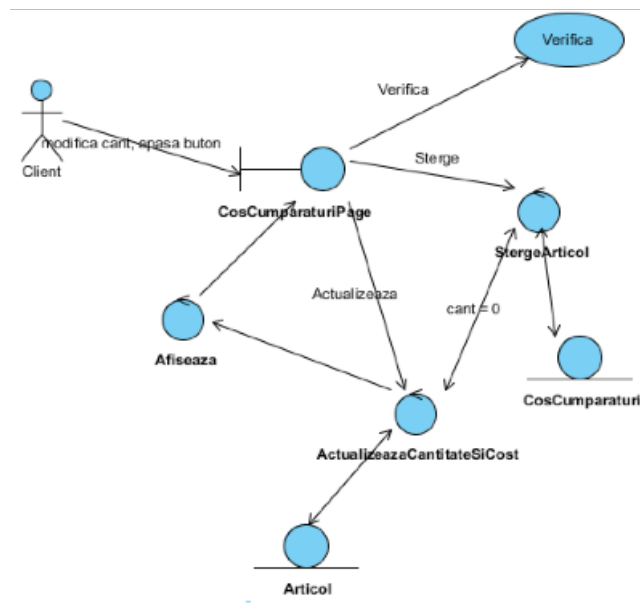


Figure 22: Diagrama de robustețe Editare coș cumpărături

claselor din diagrama de clase și substantivele din textul cazului de utilizare. Specificarea utilizării sistemului în contextul modelului obiect stă la baza construirii diagramelor de secvențe.

5. Nu alocați comportament claselor din diagrama de robustețe. Controlerele sunt substituenți pentru funcționalitatea și comportamentul sistemului. În această etapă nu există suficiente informații disponibile pentru a asigura operații la clase. Deciziile de alocare a comportamentului se vor face utilizând diagrame de secvențe.
6. Nu includeți prea puține sau prea multe controlere. Se recomandă să existe între 2 și 5 controlere pe o diagramă de robustețe. Câte un controler pe caz de utilizare înseamnă multe cazuri de utilizare fiecare cu comportament redus. La peste 10 controlere pe o diagramă se recomandă divizarea cazului de utilizare în mai multe cazuri de utilizare mai ușor de manipulat.
7. Nu alocați prea mult timp încercării de a perfecționa diagrama de robustețe. Analiza de robustețe este un instrument intermediar care ajută la descoperirea de obiecte, alocare de atribute și verificarea completitudinii și corectitudinii textului cazului de utilizare. După îndeplinirea acestei misiuni diagrama nu trebuie păstrată.
8. Nu faceți proiectare de detaliu pe diagrama de robustețe. Proiectarea preliminară produce diagrame volatile, ceea ce nu se întâmplă în cazul proiectării de detaliu. Diagramele de secvențe sunt potrivite pentru proiectare

de detaliu. Analiza de robustețe trebuie să fie o trecere rapidă prin toate scenariile.

9. Urmăriți în paralel textul cazului de utilizare și diagrama de robustețe corespunzătoare.
10. Actualizați modelul static. Analiza de robustețe se va încheia cu actualizarea modelului domeniului.

6 Exemple

Următoarele exemple, provenite din diagramele de robustețe ale modelului aplicației pentru librărie accesibilă pe Internet, au rolul de a ilustra aplicarea recomandărilor anterioare.

6.1 Exemplul 3

Descrierea cazului de utilizare Login (Figura 23) este aceeași cu descrierea din Figura 19 și este adăugată și aici pentru conveniență. Diagrama de robustețe incorectă pentru Login este în Figura 24, iar cea corectată este în Figura 25 care este aceeași cu Figura 20 dar a fost adăugată aici pentru conveniență.

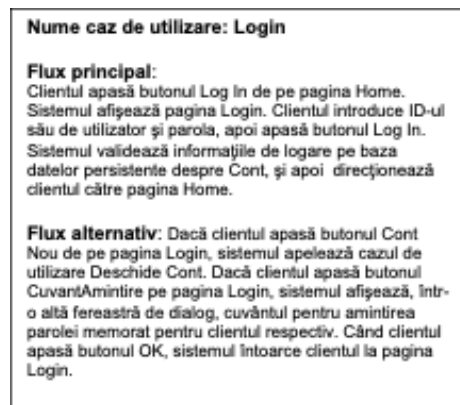


Figure 23: Descriere Login

Explicație. *Obiectul boundary HomePage comunică cu obiectul boundary LoginPage și cu obiectul entity Cont. Obiectul Cont are asignată o operație. Nu este reprezentat nici un flux alternativ. Paginile Web trebuie afișate utilizând un controler standard, Afiseaza.*

6.2 Exemplul 4

S-a realizat diagrama de robustețe din Figura 27 pe baza descrierii din Figura 26. Urmând indicațiile de realizare (Figura 28), s-a obținut varianta corectată

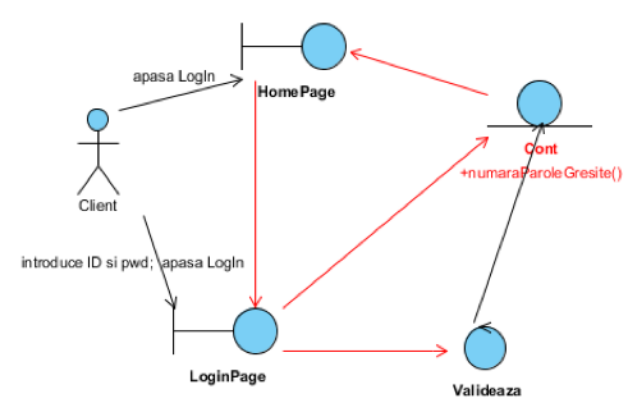


Figure 24: Diagrama de robustețe Login (greșeli)

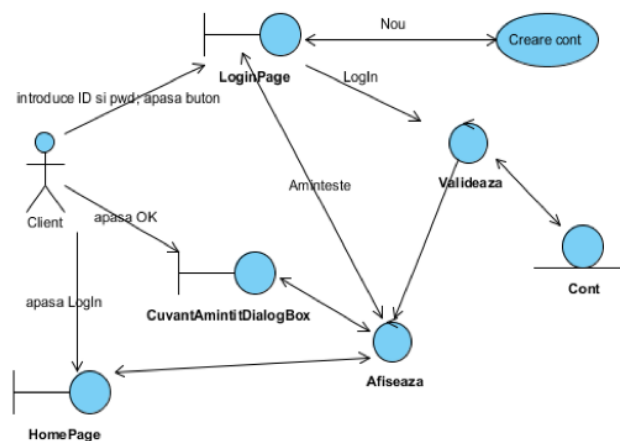


Figure 25: Diagrama de robustețe Login (corectată)

din Figura 29.

Explicație. *Sunt prea puține controlere. VerificareFrazaCautare permite sistemului să evite o căutare fără a avea o frază de căutare, pe când Afisare este un controler standard asociat cu paginile Web. (Aceste controlere reflectă și fluxuri alternative ce nu apar pe diagrama precedentă). Obiectul entity RezultateCautare comunică cu obiectul boundary RezultateCautarePage. Textul cazului de utilizare nu reflectă crearea obiectului RezultateCautare.*

6.3 Exemplul 5

Descrierea cazului de utilizare *Editare coș* (Figura 30). Modelarea domeniului și Diagrama de robustețe cu greșeli (Figura 31). Modelarea domeniului și

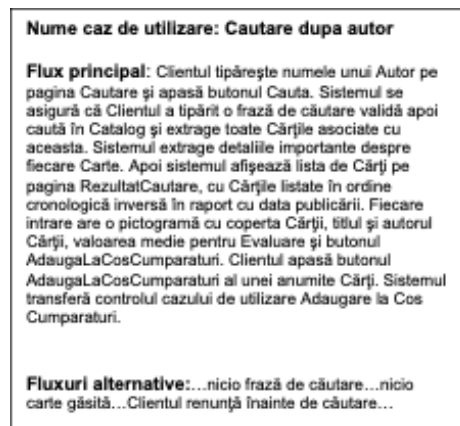


Figure 26: Descrierea cazului de utilizare Căutare după autor

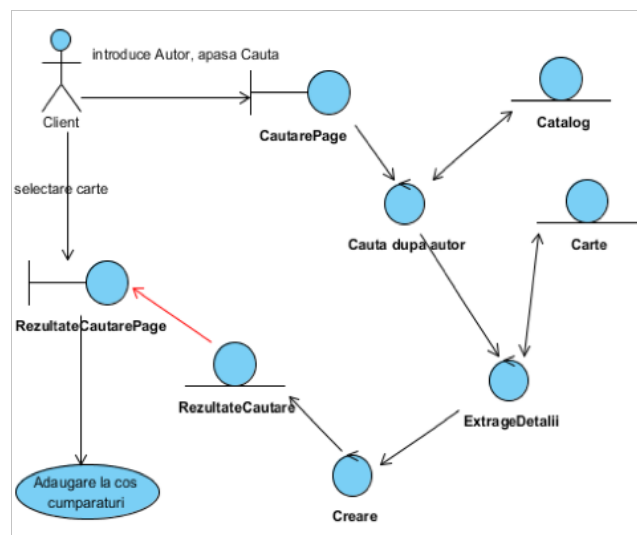


Figure 27: Diagrama de robustețe Căutare după autor (greșeli)

Diagrama de robustețe corectată (Figura 33) pe baza recomandărilor (Figura 32).

Explicație. Controlerul *ModificareCost* nu este necesar din moment ce atât el cât și controlerul *ActualizareCantitate* operează pe obiectul *Articol*. Textul cazului de utilizare se referă la *Rând*, dar reiese clar din extrasul din diagrama de clase și din diagrama de robustețe ca textul trebuie să se refere la *Articol*. (Acest tip de inconsistență în utilizarea numelor este foarte periculoasă.) Extrasul din diagrama de clase nu reflectă atributele menționate în textul cazului de utilizare.

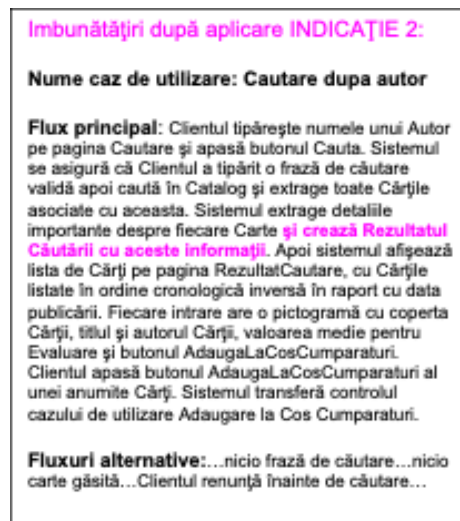


Figure 28: Îmbunătățiri realizare diagrame de robustețe

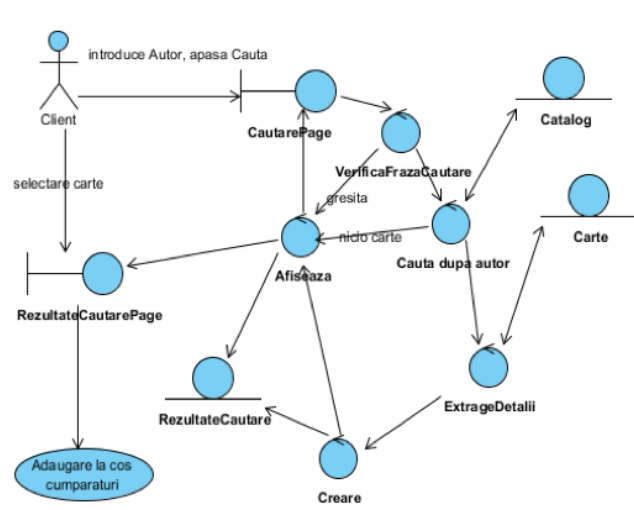


Figure 29: Diagrama de robustețe Căutare după autor (corectată)

6.4 Exemplul 6

Descrierea cazului de utilizare *Livrare Ordin* (Figura 34). Diagrama de robustețe cu greșeli (Figura 35). Diagrama de robustețe corectată (Figura 37) pe baza recomandărilor (Figura 36).

Explicație. *Obiectul Întrerupere este o construcție care ține de proiectarea de detaliu. Obiectul Ordin are asignată o metodă. Textul cazului de utilizare*

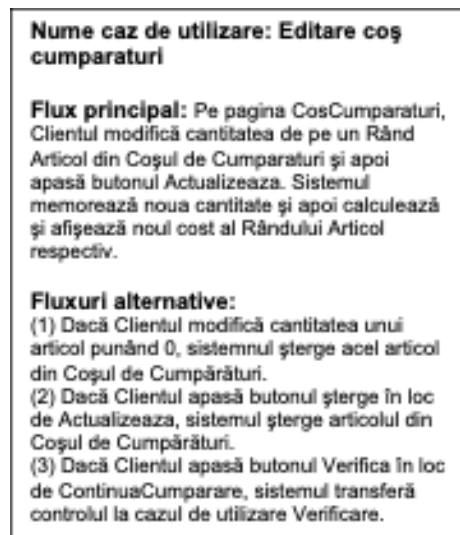


Figure 30: Descriere caz de utilizare Editare Coș

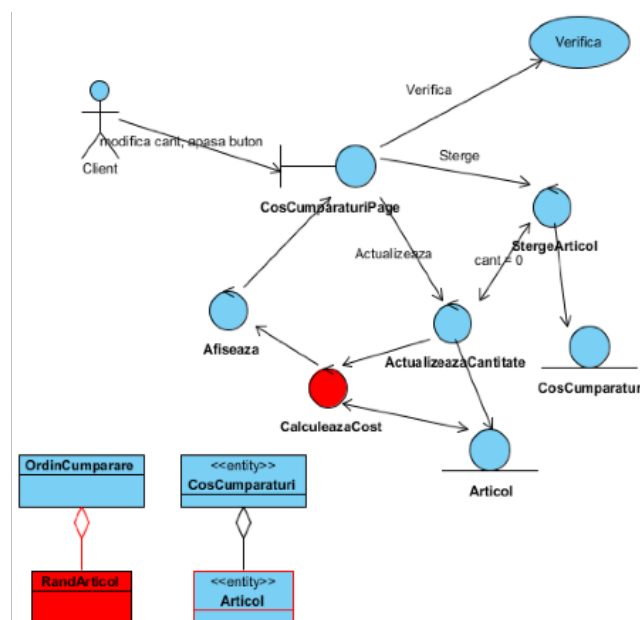


Figure 31: Modelarea domeniului și Diagrama de robustețe Editare Coș (greșeli)

conține o construcție pasivă și nu este suficient de precis în descrierea diagramei de robustețe.

Îmbunătățiri după aplicare INDICAȚIE 2:

Nume caz de utilizare: Editare coș cumparaturi

Flux principal: Pe pagina CosCumparaturi, Clientul modifică cantitatea **de pe un Rând unui Articol** din Coșul de Cumparaturi și apoi apasă butonul Actualizeaza. Sistemul memorează noua cantitate și apoi calculează și afișează noul cost al **Rândului Articolului** respectiv.

Fluxuri alternative:

- (1) Dacă Clientul modifică cantitatea unui articol punând 0, sistemul șterge acel articol din Coșul de Cumpărături.
- (2) Dacă Clientul apasă butonul șterge în loc de Actualizeaza, sistemul șterge articolul din Coșul de Cumpărături.
- (3) Dacă Clientul apasă butonul Verifica în loc de ContinuaCumparare, sistemul transferă controlul la cazul de utilizare Verificare.

Figure 32: Îmbunătățiri realizare diagrame de robustețe

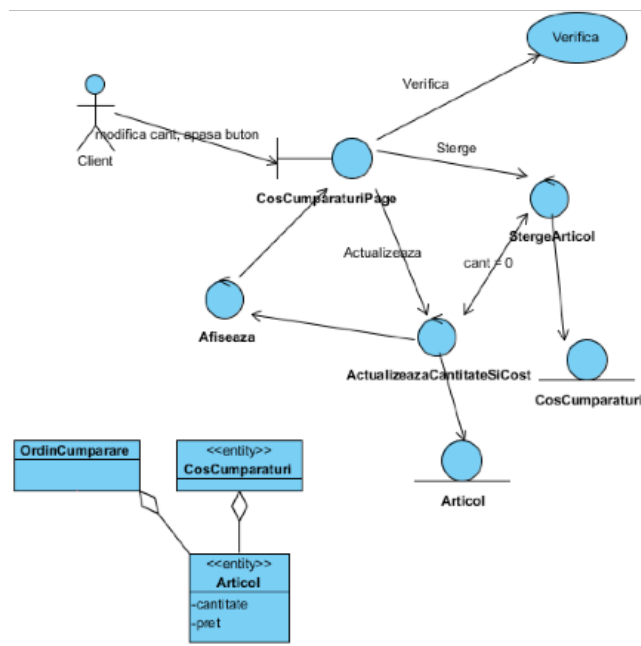


Figure 33: Modelarea domeniului și Diagrama de robustețe Editare Coș (corectată)

Nume caz de utilizare: Livrare ordin
Flux principal: Funcționarul de livrare se asigură că Articolele listate pe eticheta pachetului pentru Ordin corespund elementelor fizice. Apoi trece codul de bare de pe eticheta pachetului prin fața senzorului stației de livrare. Starea ordinului se schimbă în "livrare". Metoda de livrare este afișată pe Consola Stației de Livrare. Funcționarul cântărește setul elementelor fizice. Funcționarul împachetează Articolele. Funcționarul atașează o etichetă corespunzătoare metodei de livrare. Funcționarul trece codul de bare al acestei etichete prin fața senzorului. Funcționarul trimite pachetul prin furnizorul asociat.
Flux alternativ: Dacă Funcționarul de livrare identifică o nepotrivire între Ordin și elementele fizice, el întrerupe procesarea Ordinului până ce va rezolva această problemă.

Figure 34: Descriere Livrare Ordin

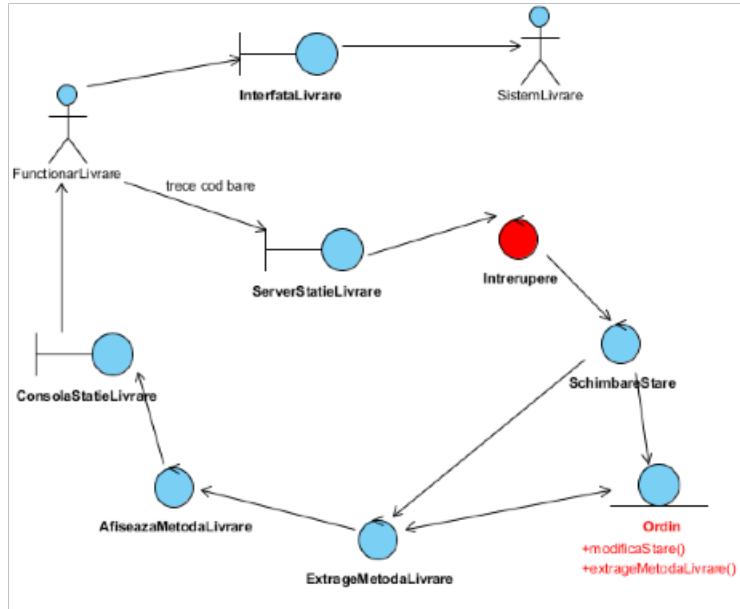


Figure 35: Diagrama de robustețe Livrare Ordin (greșeli)

6.5 Exemplul 7

Descrierea cazului de utilizare *Urmărire Ordin* (Figura 38). Diagrama de robustețe cu greșeli (Figura 39). Diagrama de robustețe corectată (Figura 40).

Explicație. Obiectul boundary *DetaliiOrdinePage* comunică cu obiectul boundary *UrmărireOrdinePage*. Nu se indică ce se întâmplă dacă *Clientul* nu a plasat niciun ordin în ultimele 30 zile. Extrasul din diagrama de clase nu reflectă clasa *TabelaOrdine* nou descoperită.

Nume caz de utilizare: Livrare ordin

Nume caz de utilizare: Livrare ordin

Flux principal: Funcționarul de livrare se asigură că Articolele listate pe eticheta pachetului pentru Ordin corespund elementelor fizice. Apoi trece codul de bare de pe eticheta pachetului în fața senzorului stației de livrare. **Starea Ordinului se schimbă în "În curs de livrare". Metoda de livrare este afișată pe Consola Stației de Livrare.** Sistemul schimbă starea Ordinului în "În curs de livrare". Apoi sistemul extrage Metoda de Livrare specificată de Client pentru acest Ordin și o afișează pe Consola Stației de Livrare. Funcționarul cântărește setul elementelor fizice. Funcționarul împachetează Articolele. Funcționarul atașează o etichetă corespunzătoare metodei de livrare. Funcționarul trece codul de bare al acestei etichete în fața senzorului. Funcționarul trimite pachetul prin furnizorul asociat.

Flux alternativ: Dacă Funcționarul de livrare identifică o nepotrivire între Ordin și elementele fizice, el întrerupe procesarea Ordinului până ce va rezolva această problemă.

Figure 36: Îmbunătățiri realizare diagrame de robustețe

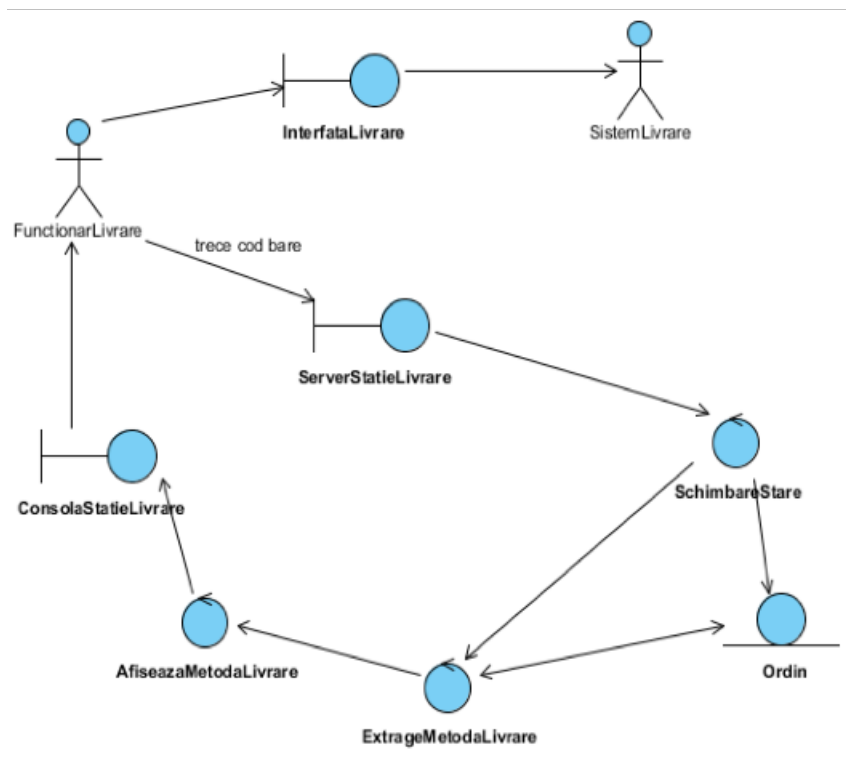


Figure 37: Diagrama de robustețe Livrare Ordin (corectată)

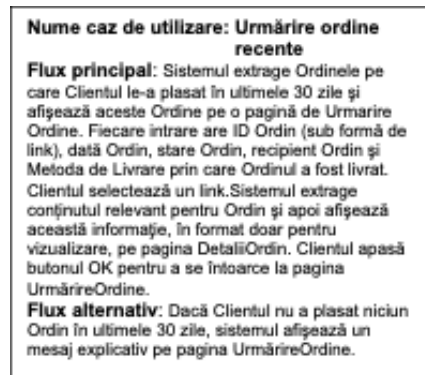


Figure 38: Descriere Urmărire Oridin

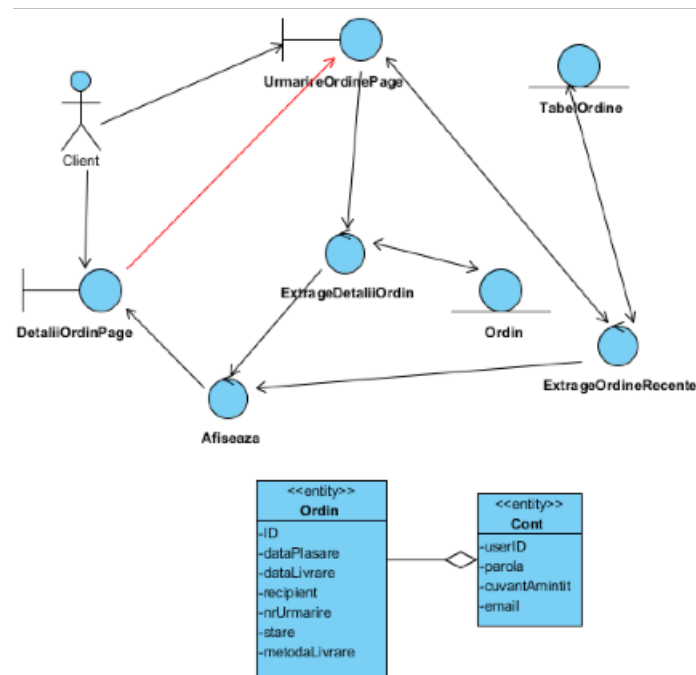


Figure 39: Diagrama de robustețe Urmărire ordin (greșeli)

Figura 41 reprezintă diagrama de clase (modelul domeniului) care include unele atribute pentru clasele aplicației Librărie Online.

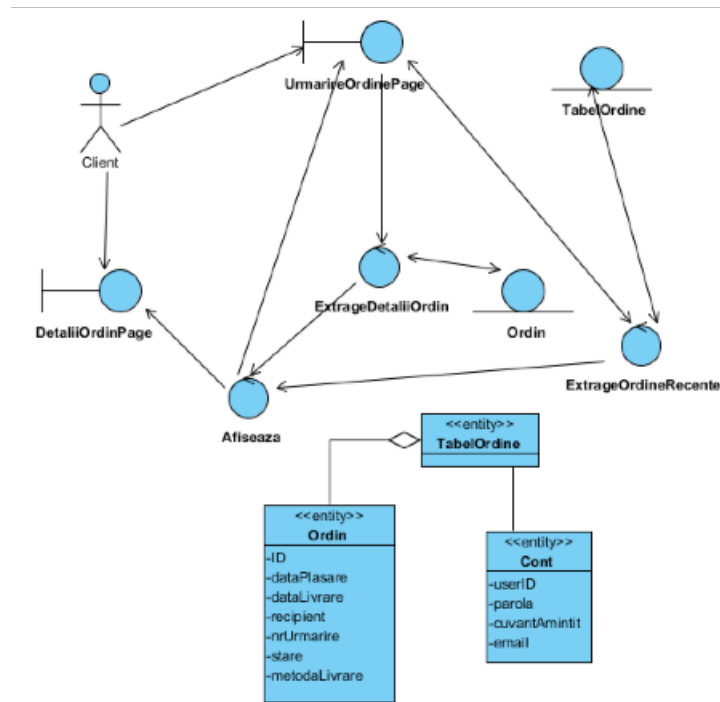


Figure 40: Diagrama de robustețe Urmărire ordin corectată

7 Temă

1. Studiați cum se adaugă stereotipurile care sunt utile în reprezentarea diagramelor de robustețe.
2. Realizați analiza de robustețe pentru toate cazurile de utilizare ale aplicației *Sistem închiriere mașini*.

References

- [1] *Agile Modeling*. <https://agilemodeling.com/>. Accessed: April 14th, 2024.
- [2] *Iconix Process*. <https://iconixprocess.wordpress.com/iconix-process/>. Accessed: April 14th, 2024.
- [3] Tom Pender. *UML 2 Bible*. John Wiley & Sons, 2003.
- [4] Doug Rosenberg and Kendall Scott. *Use case driven object modeling with UML*. Springer, 1999.

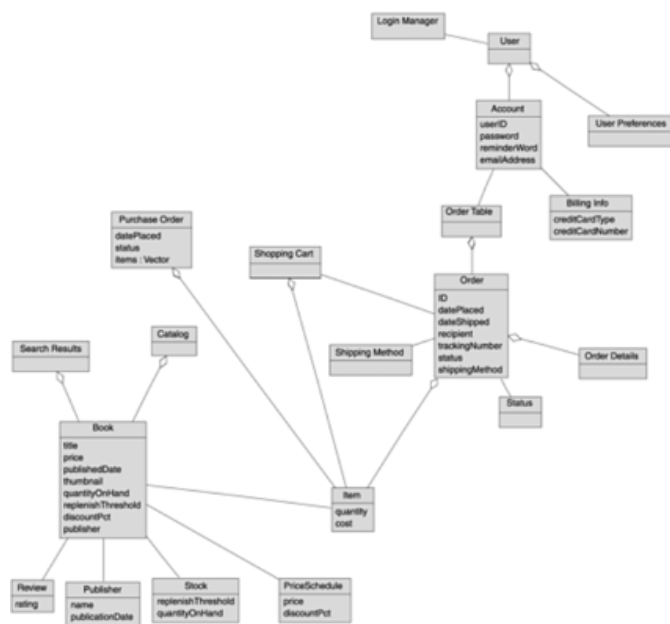


Figure 41: Modelul domeniului pentru aplicația Librărie Online