

Seminar 1

Colectarea cerintelor. Modelul functional al unui sistem.

Diagrama UML a cazurilor de utilizare (eng. UML Use Case Diagram - UCD)

Biblio:

1. Martin Fowler , *UML Distilled* (3 ed.)
2. OMG, *UML 2.5.1 Specification*
3. Martina Seidl et al., *UML@Classroom*

Modelul functional

- produs al activitatii *Colectarea cerintelor*, alături de descrierea cerintelor nefunctionale ale sistemului
- descris în termeni de *actori și cazuri de utilizare*
- instrumente de reprezentare: *diagrama UML a cazurilor de utilizare* + *sabloane tabelare* pentru descrierea narativa a continutului cazurilor de utilizare

1) Caz de utilizare (CU)

Def.1: **Scenariu** = o secvența de pași/evenimente, care descrie o interacțiune tipică a unui „utilizator” cu sistemul.

Ex.1: scenariu/flux normal privind plasarea unei comenzi într-un magazin virtual (eng. MSS – main success scenario)

1. Clientul populeaza cosul de cumpărături, selectand produse din catalogul online.
2. Sistemul actualizeaza continutul cosului.
3. Clientul opteaza pentru vizualizarea continutului cosului.
4. Sistemul afiseaza detaliile cosului.
5. Clientul opteaza pentru finalizarea comenzii.
6. Sistemul solicita, prin afisarea unui formular, introducerea detaliilor de livrare (destinatar, adresa, tip livrare – poșta/curierat/pick-up point).
7. Clientul completeaza detaliile de livrare și submite informatiile.
8. Sistemul afiseaza prețul final al comezii (actualizat cu costurile de livrare) și solicita alegerea unei modalitati de plata (ramburs/transfer bancar/online, cu card).
9. Clientul opteaza pentru plata cu cardul.
10. Sistemul solicita detaliile cardului (nume deținător, nr. card, data expirării, CVV)
11. Clientul introduce datele cardului și opteaza pentru finalizarea platii.
12. Sistemul valideaza cardul și notifica utilizatorul referitor la plasarea cu succes a comenzii, atât printr-un mesaj în interfata, cât și printr-un email ulterior.

Ex. 2: scenariu alternativ 1 (de excepție): autorizarea cardului esueaza, datele introduse sunt incorecte

- 12^a. Sistemul afiseaza un mesaj de eroare si solicita reintroducerea datelor de card.
 - 12^{a.1} Utilizatorul reintroduce datele (corecte).
- Se revine în MSS la punctul 12.

Ex. 3: scenariu alternativ 2: clientul e unul deja înregistrat în sistem, având datele de livrare și de card salvate

- 6^a. Sistemul solicita utilizarea sau suprascrierea datelor preferentiale salvate, legate de livrare și plata cu cardul.
 - 6^{a.1} Clientul opteaza pentru utilizarea acestora.
- Se revine în MSS la punctul 12.

... samd

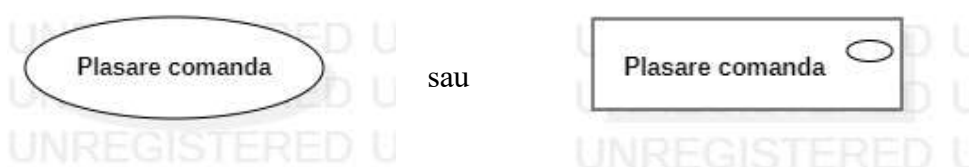
În scenariile alternative pot apărea oricâți pași, funcție de natura interacțiunii (nu doar unul, așa cum se întâmpla în scenariile descrise anterior). La fel, funcție de natura interacțiunii, se poate reveni sau nu la unul din pașii din scenariul principal.

Punctul comun al tuturor scenariilor anterior descrise = un același obiectiv final al „utilizatorului” (plasarea unei comenzi).

Def.2: Caz de utilizare = mulțimea tuturor scenariilor care au ca și punct comun un același obiectiv final (scop) al „utilizatorului” - realizarea aceleiași funcționalități.

Caz de utilizare (CU) = o mulțime de secvențe de evenimente, care descriu toate interacțiunile posibile între un „utilizator” și sistem, în scopul realizării unei anumite funcționalități.

Sintaxa UML – elipsa (cu numele cazului sub/în interior) sau reprezentare de clasificator - dreptunghi (cu icon elipsa). Numele unui CU este o expresie verbală!!! (fie ea și substantivizată): *Plasează comanda* sau *Plasare comanda*, NU Comanda – ca și substantiv!!! (sa rezulte funcționalitatea oferită, o comandă se poate plasa, pune în așteptare, onora etc.)



„utilizator” = actor

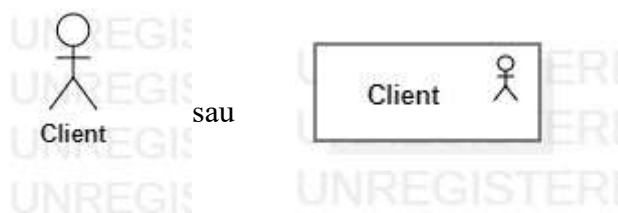
2) Actor (A) = rol jucat de o entitate *externă* sistemului, care *interacționează* cu sistemul.

Instancele actorilor pot fi persoane, alte sisteme (sistemul bancar cu care aplicația comunică pentru validarea cardurilor), echipamente hardware/dispozitive (imprimanta, senzori externi).

O instanță poate juca mai multe roluri (un manager poate juca rol de vânzător, la nevoie), un rol poate fi jucat de mai multe instanțe (orice persoană poate fi client al magazinului virtual).

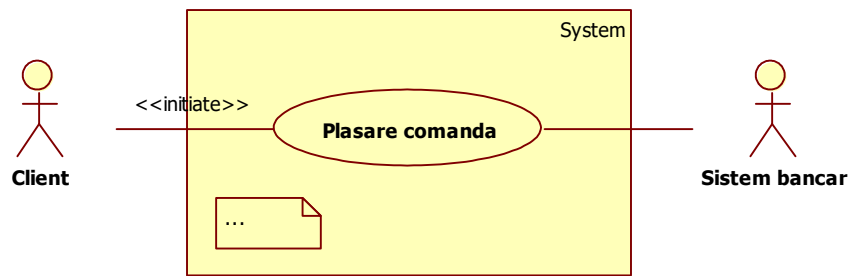
Un actor poate fi (și este, în general) implicat în mai multe cazuri de utilizare; la un caz de utilizare pot participa mai mulți actori (în CU *Plasare comanda* intervine atât clientul, cât și sistemul bancar la care se apelează pentru validarea cardului). Doar unul dintre ei inițiază interacțiunea (stereotipizare cu <<initiate>> a relației de comunicare aferente).

Sintaxa UML - eng. *stick figure* (denumirea sub/deasupra) sau reprezentare de clasificator (cu stereotip <<actor>> explicit sau icon) sau imagine (reprezentativă pentru tipul de actor, de ex. imaginea unei imprimante)



3) Frontiera sistemului (eng. *system boundary* sau *subject* - în ultimele versiuni ale specificației) – diferențiază interiorul sistemului de mediul sau: actorii sunt înafara frontierei, cazurile de utilizare înăuntru.

Sintaxa UML – dreptunghi etichetat cu numele sistemului (conform specificației UML, eticheta se plasează în colțul stânga sus)



4) Relatii

4.1. Comunicare

- între A și CU

- rel. de asociere bidirecțională – comunicarea/schimbul de informație e în ambele sensuri

Poate fi stereotipizată cu `<<initiate>>`, pentru a indica acel actor care inițiază comunicarea, în cazul în care există mai mulți actori care interacționează cu CU.

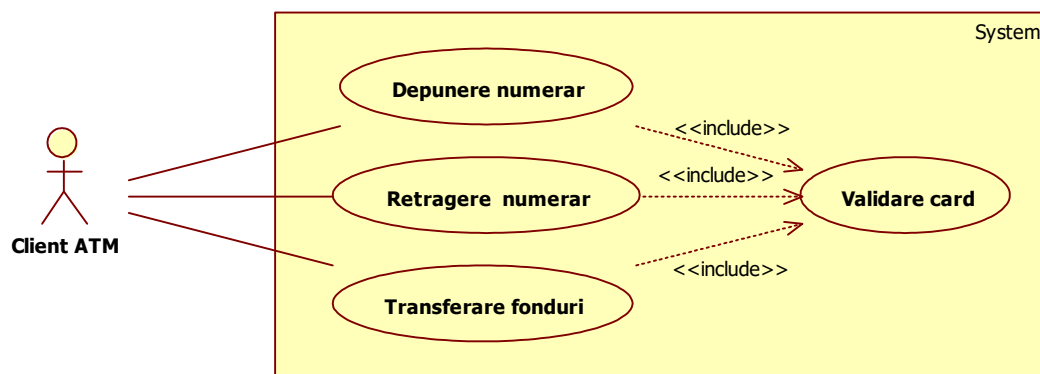
4.2 Incluziune

- între două CU

- rel. de dependență, stereotipizată cu `<<include>>`

Sensul dependenței e de la CU care include (de bază) la CU inclus. Dpdv semantic, comportamentul cazului de utilizare inclus va fi inserat în cadrul comportamentului cazului de utilizare care include. Comportamentul cazului de utilizare care include nu este complet definit (nu are sens), decât în prezența comportamentului cazului de utilizare inclus (nu se poate face retragere de numerar fără validarea cardului).

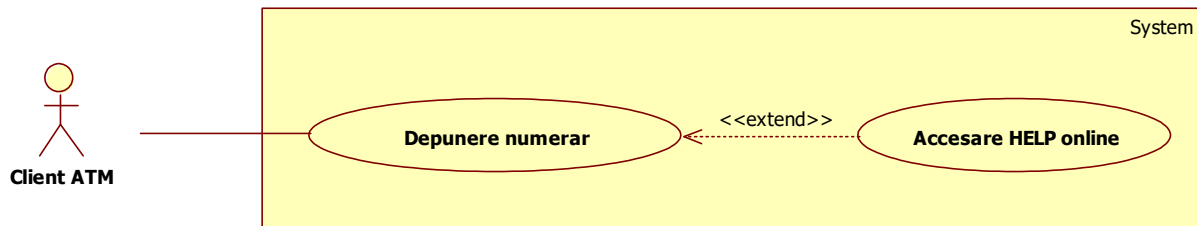
Relația de incluziune se recomandă a fi folosită atunci când două sau mai multe cazuri de utilizare partajează o aceeași secvență de comportament. Secvența comună poate fi izolată într-un nou caz de utilizare, ce va fi inclus în cele inițiale. Scopul este evitarea redundanțelor și simplificarea descrierii cazurilor de utilizare (maximizarea inteligibilității descrierilor textuale). Similitudine cu apelul de procedură din limbajele de programare. În descrierea textuală a cazurilor de utilizare care includ, secvența comportamentală factorizată (succesiunea de evenimente/pasi) poate fi înlocuită cu un singur pas, care referă numele cazului de utilizare inclus (hiperlink).



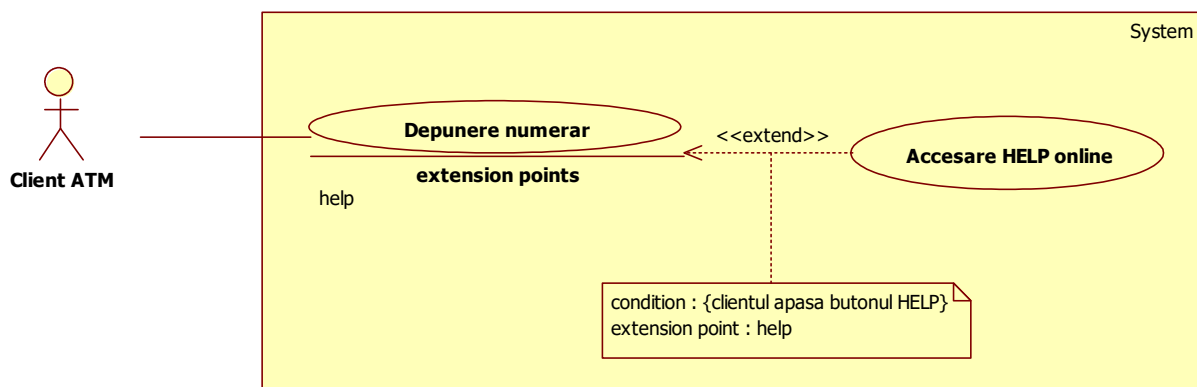
4.3 Extindere

- între două CU
- rel. de dependență, stereotipizată cu <<extend>>

Sensul dependenței e de la CU care extinde la CU extins (de baza). Dp dv semantic, relația precizează *când* (în ce condiții) și *unde* în comportamentul cazului de utilizare de baza se poate insera comportamentul descris de cazul de utilizare care extinde. Comportamentul cazului de utilizare extins este însă complet definit în absența celui care extinde (se poate face depunere fără accesare de help online). În schimb, cazul de utilizare care extinde poate fi doar un increment comportamental, fără a avea semnificație de sine statatoare (în absența cazului de care depinde).



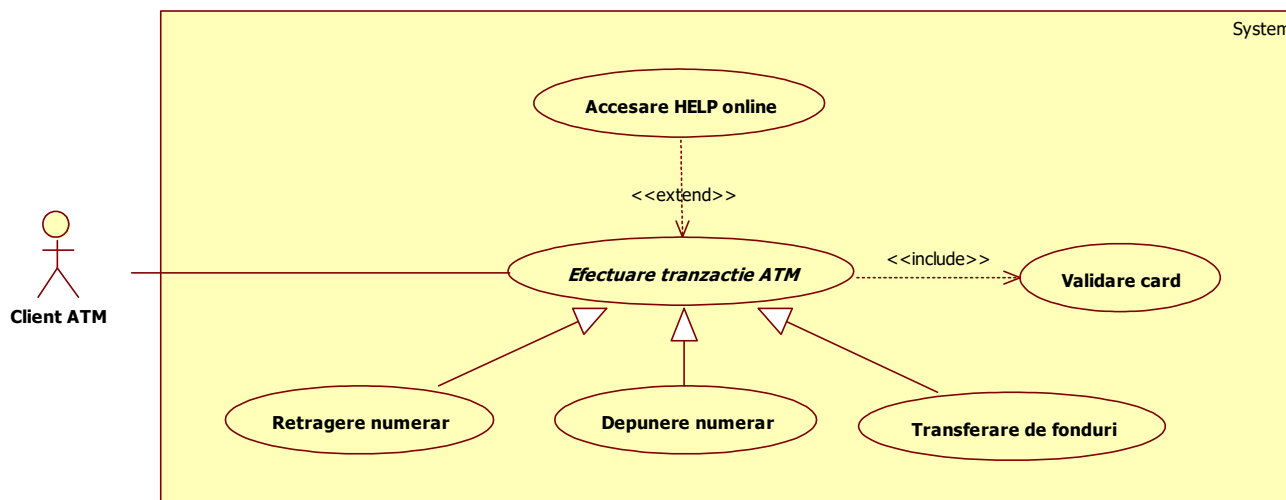
Pentru specificarea completă a relației de extindere este necesară precizarea condițiilor în care se face extinderea și localizarea acesteia (folosind o notă UML atașată relației de extindere). Localizarea se face cu ajutorul așa-numitor *puncte de extindere* (eng. *extension points*), care se definesc la nivelul cazului de utilizare de baza. Pentru fiecare punct de extindere se precizează numele (ales de utilizator) și locația (raportată la modalitatea de descriere a conținutului cazului de utilizare). În cazul descrierii tabelare/narative a cazului de utilizare, localizarea unui punct de extindere se poate face prin prezizarea numerică a pasului după care se aplică. Un același punct de extindere poate avea mai multe locații posibile (un același increment se poate aplica în mai multe locuri la nivelul unui caz de utilizare). În cazul în care incrementul descris de cazul de utilizare care extinde se poate aplica oriunde în comportamentul de baza, oțim precizarea locației punctului de extindere aferent.



? Gestionarea datelor unui client deja înregistrat (flux alternativ al cazului de utilizare plasare comanda în magazin virtual) poate fi izolată la nivelul unui caz de utilizare care extinde plasarea comenzii? Dar fluxurile de excepție?

4.4 Generalizare

- între A sau între CU
- semantica uzuala



Practic: Diagrama cazurilor de utilizare pentru sistemul de preluare a comenzilor din cadrul unui restaurant (vezi modelul din Seminar 2).

5) Descrierea conținutului cazurilor de utilizare: Relativ la realizarea modelului functional, UML ofera doar diagrama cazurilor de utilizare, însă nu și o modalitate standard de a descrie conținutul acestora (motiv pentru care nu este formalizata nici modalitatea de localizare a punctelor de extensie discutate anterior). De regula, în acest scop se folosesc sabloane tabelare, în care un caz de utilizare este descris textual, în limbaj natural, precizand cel puțin: numele cazului de utilizare, actorii participanti, fluxul normal de evenimente și fluxuri alternative sau excepții. Mai pot apărea pre/post-conditii și cerinte de calitate.

Ex. 4: Descrierea cazului de utilizare Plasare comanda al sistemului MagazinVirtual

Nume	<i>Plasare comanda</i>
Actori participanti	Initial de <i>Client</i> Comunica cu <i>Sistem bancar</i>
Flux de evenimente (scenariu normal)	<ol style="list-style-type: none">1. Clientul populeaza cosul de cumpărături, selectand produse din catalogul online.2. Sistemul actualizeaza conținutul cosului.3. Clientul opteaza pentru vizualizarea conținutului cosului.4. Sistemul afiseaza detaliile cosului.5. Clientul opteaza pentru finalizarea comenzii.6. Sistemul solicita, prin afisarea unui formular, introducerea detaliilor de livrare (destinatar, adresa, tip livrare – poșta/curierat/pick-up point).7. Clientul completeaza detaliile de livrare și submite informatiile.8. Sistemul afiseaza prețul final al comezii (actualizat cu costurile de livrare) și solicita alegerea unei modalitati de plata (ramburs/transfer bancar/online, cu card).9. Clientul opteaza pentru plata cu cardul.10. Sistemul solicita detaliile cardului (nume deținător, nr. card, data expirării, CVV)11. Clientul introduce datele cardului și opteaza pentru finalizarea platii.12. Sistemul valideaza cardul și notifica utilizatorul referitor la plasarea cu succes a comenzii, atât printr-un mesaj în interfata, cât și printr-un email ulterior.

Scenarii alternative	<p><u>scenariu alternativ 1</u> (de excepție): autorizarea cardului esueaza, datele introduse sunt incorecte</p> <p>12^a. Sistemul afiseaza un mesaj de eroare si solicita reintroducerea datelor de card.</p> <p>12^a.1 Utilizatorul reintroduce datele (corecte).</p> <p>Se revine în MSS la punctul 12.</p> <p><u>scenariu alternativ 2</u>: clientul e unul deja înregistrat în sistem, având datele de livrare și de card salvate</p> <p>6^a. Sistemul solicita utilizarea sau suprascrierea datelor preferentiale salvate, legate de livrare și plata cu cardul.</p> <p>6^a.1 Clientul opteaza pentru utilizarea acestora.</p> <p>Se revine în MSS la punctul 12.</p>
----------------------	---