Ministerul Educaţiei al Republicii Modova

Universitatea de Stat din Moldova

Facultatea de Fizica si Inginerie

Catedra Fiziсă Aplicată şi Informatică

Lucrare de laborator Nr. 1

Tema: **“ *Diagrama Use Case*”**

Lucrarea a fost îndeplinită de studentul grupei 3.2:

Ciobanasu Ion

Lucrarea a fost controlată de

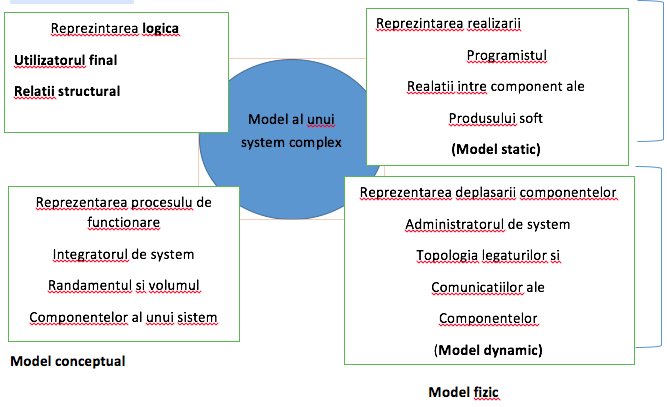
lector univ: Diaconu Elena

Chisinau 2016

Limbajul UML reprezinta limbajul de destinatie generala al modelarii vizuale , care este elaborate pentru specificarea ,vizualizarea ,construirea si documentarea componentelor produsului soft ,bussines-proceselor si a altor sisteme. Acest limbaj contine cele mai bune calitati ale metodelor ingineriei de program care au fost utilizate cu success pe parcursul ultimilor ani la modelarea sistemelor complexe .

Limbajul UML se imparte in 2 modele :   
 **1.Model static**

**2.Model dynamic**



**Diagrame UML**

1. Diagrame cazurilor de utilizare (use case diagram)
2. Diagrame de clase (class diagram)
3. Diagrame de comportament (behavior diagrams)

1. Diagrame de stari (statechart diagrams)
2. Diagrame de activitati (activity diagrams)
3. Diagrame de interactiune (interaction diagrams)

1. Diagrame de secventa (sequence diagram)
2. Diagrame de colaborare (collaboration diagram)
3. Diagrame de realizare (implementation diagram)
4. Diagrame de component (component diagram)
5. Diagrame de plasare (deployment diagram)

In continuare vom vorbi despre diagrama **Use case.**

O diagrama a cazurilor de utilizare (use case diagram) prezinta o colectie de cazuri de utilizare si actori care:

• ofera o descriere generala a modului in care va fi utilizat sistemul

• furnizeaza o privire de ansamblu a functionalitatilor ce se doresc a fi oferite de sistem

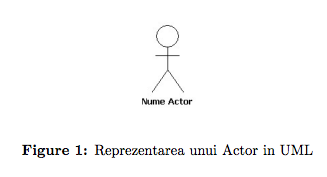
• arata cum interactioneaza sistemului cu unul sau mai multi actori

• asigura faptul ca sistemul va produce ceea ce s-a dorit.

**Actorii**

Un **actor** este un stereotip al unei clase. Actorii sunt reprezentati de utilizatori sau entitati care pot interactiona cu sistemul. Ei nu fac parte din sistem si definesc multimi de roluri in comunicarea cu acesta.

Un actor se reprezinta sub forma unui ”omulet” sub care se trece numele acestuia (vezi Figura 1).



**Identificarea Actorilor**

Identificarea actorilor se face raspunzand la urmatoarele intrebari:

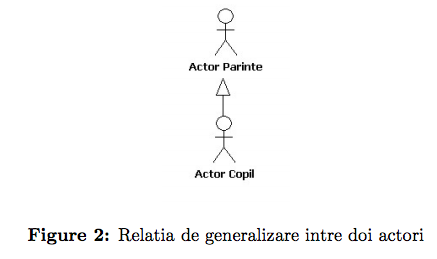
• Cine doreste sau este interesat de informatiile aflate in sistem?

• Cine modifica date?

• Cine interactioneaza cu sistemul?

**Relatii intre actori**

Intre actori poate exista relatia de generalizare. Daca un actor mosteneste un alt actor, atunci el poate sa comunice cu aceleasi cazuri de utilizare ale sistemului ca si parintele sau. Notatia UML este o sageata cu linie continua, avand la capat un triunghi gol, care indica spre actorul parinte(vezi Figura 2).

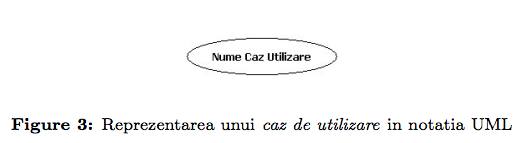
****

**Cazuri de utilizare**

**Un caz de utilizare** reprezinta o colectie de scenarii posibile, referitoare la comunicarea intre sistem si actorii externi, caracterizate de anumite scopuri. Aceste scenarii sunt definite ca secvente de pasi carora le pot corespunde cazuri de utilizare de nivel inferior. Cazurile de utilizare arata ce trebuie sa faca sistemul si nu cum.

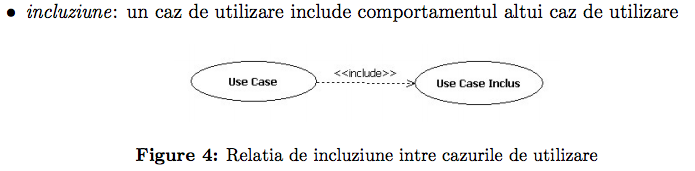
**Reprezentare**

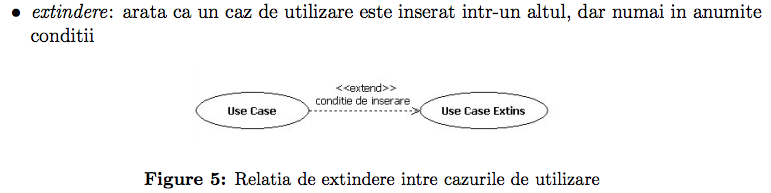
Un caz de utilizare se reprezinta sub forma unui oval in care se trece numele acestuia (vezi Figura 3).

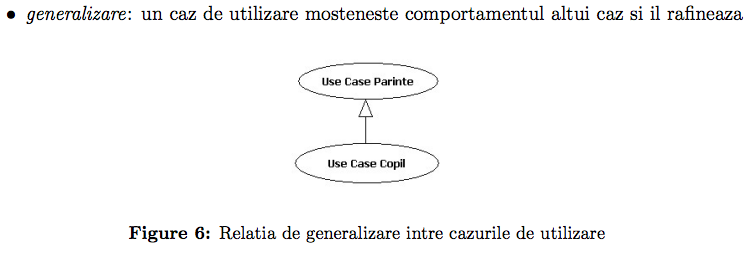


**Relatii**

Intre cazurile de utilizare pot exista urmatoarele relatii:



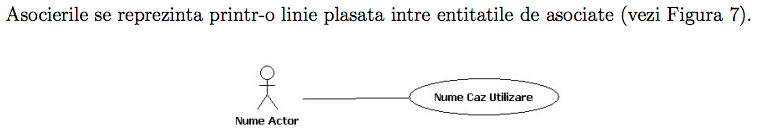




**Asocieri**

O asociere reprezinta o conexiune semantica intre cazurile de utilizare si actori.

**Reprezentare**

****

Lucru practic :

Sa se realizeze diagrama cazurilor de utilizare pentru un produs software ce urmeaza sa fie vandut din cadrul unui supermagazin.

1. Clientul solicita produsul.

2. Vanzatorul realizeaza facturarea acestuia. Pentru acesta:

(a) se scaneaza codul de bare

(b) se cauta in baza de date codul produsului

(c) daca produsul este gasit se adauga la factura

(d) daca prosusul nu este gasit este respins

**Diagrama use case**

Din enuntul anterior se identifica urmatorii actori:

• Client

• Vanzator

Cazurile de utilizare corespunzatoare functionalitatii descrise sunt urmatoarele:

• Solicitarea produsului

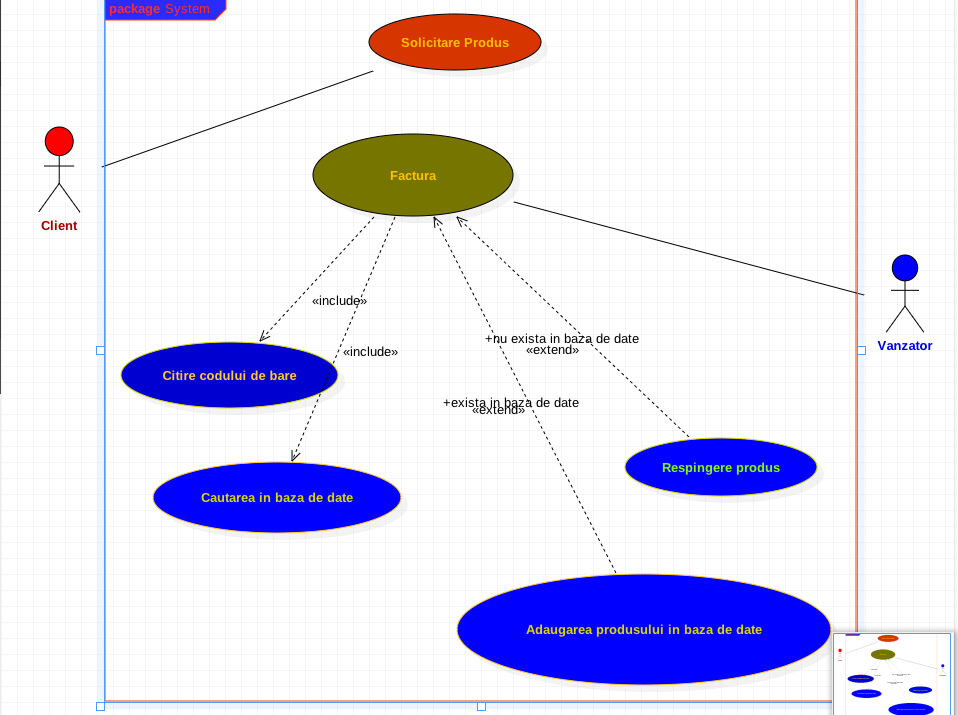
• Facturarea

• Citirea codului de bare

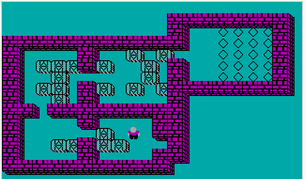
• Cautarea in baza de date

• Adaugarea produsului la factura

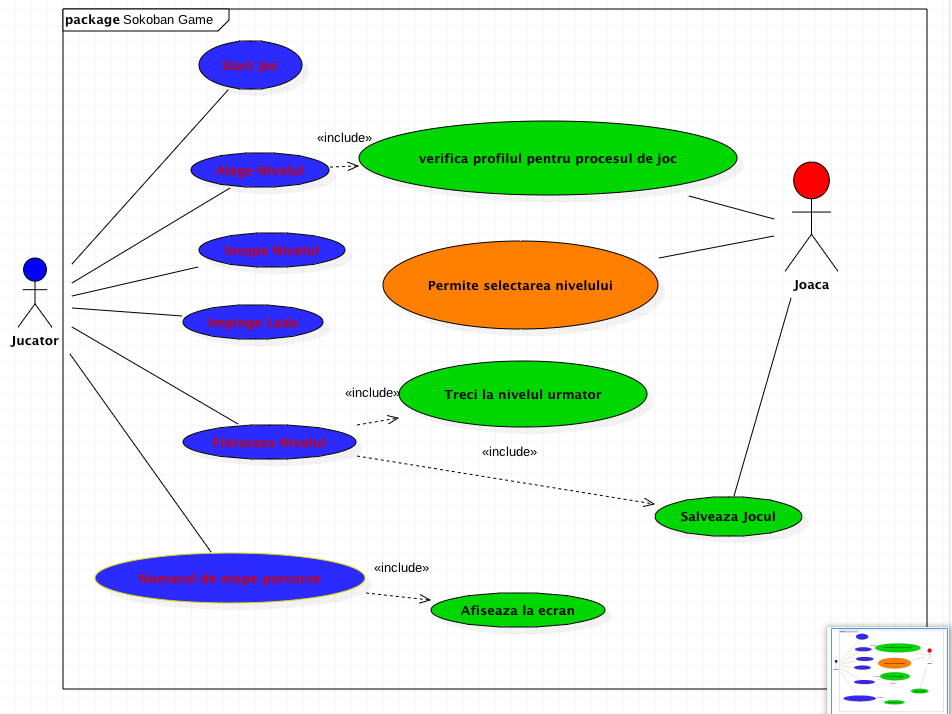
• Respingerea produsului



2-lea exemplu de diagram Use Case este joaca de logica Sokoban .



Sokoban - un joc de logica în care jucătorul se mișcă prin labirint de cutii, așa cum se arată în planul pentru a pune toate casetele pe poziția finală specificată. Doar o singură cutie poate fi mutata la un moment dat, cu eroul jocului care poate împinge doar cutii, dar nu le trage. Având în vedere că jocul este destul de dificil de a recrea în mod fizic, este de obicei pus în aplicare ca un joc pe calculator.



Concluzie :

In acest laborator am rezlizat doua diagrame Use Case in care am descris un bussines process (prima diagrama) si un un model de vizualizre a unui soft (a doua diagrama).

Use case diagram ne ajuta sa facem schema (prototip) al unui soft , in general a unei idei pe care dorim so realizam. Facand aceasta diagrama ne va fi usor sa scrim codu pentru un soft, sa rezlizam mai repede bussinnes idea.

In ziua de azi cred ca majoritatea se folosesc de use case, deoarece simplifica mai apoi mersul lucrarii.La orice lucru se face schema,(prototip) ca mai apoi sa se conduca dupa acest prototip sis a se observe unde sunt erori.