

Descrierea proiectelor software folosind diagrame UML

Cuprins

- 1. Utilitatea limbajului UML**
- 2. Versiuni UML**
- 3. Principalele părți ale UML**
 - View-uri
 - Diagrame
- 4. Generarea diagramelor UML**

Utilitatea limbajului UML

- **UML (Unified Modelling Language) este un limbaj vizual de modelare.**
- **Limbajul este destinat vizualizării, specificării, construirii și documentării sistemelor de aplicații, dar are limitări în ceea ce privește generarea codului.**
- **UML nu este dependent de tehnologia sau de limbajul folosit.**
- **UML reunește cele mai bune tehnici și practici din domeniul ingineriei programării.**

Versiuni UML

- **Versiunea 1.4 - [09.2001]**

Cea mai importantă schimbare adusă de UML 1.4 constă în adăugarea profilurilor.

- **Versiunea 2.0 – [08.2005]**

Diagramele de obiecte si diagramele de pachete devin diagrame oficiale.

Diagramele de colaborare devin diagrame de comunicare.

Versiuni UML

Sunt introduse, de asemenea, noi tipuri de ansamblu a interacțiunilor, timing și structuri compozite.

- **Versiunea 2.1 – [08.2005]**

Sunt făcute modificări minore față de versiunea UML 2.0, în principal au loc corecții și îmbunătățiri de consistență.

- **Versiunea 2.4.1 – [08.2011],**

- **2.5 – [03.2015] (modif. minore)**

<http://www.omg.org/spec/UML/2.5/PDF/>

Este versiunea curentă a UML, aduce diferite îmbunătățiri minore față de UML 2.3 claselor și pachetelor.

Principalele părți ale UML

- **Un view este format din mai multe diagrame, iar o diagramă poate face parte din mai multe view-uri.**
- **View-uri**
 - Design view
 - Deployment view
 - Implementation view
 - Process view
 - Use case view

Principalele părți ale UML

- O diagrama ofera utilizatorului un mijloc de vizualizare si de manevrare a elementelor de modelare. Majoritatea diagramelor sunt grafuri, compuse din elemente si arce.

Principalele părți ale UML

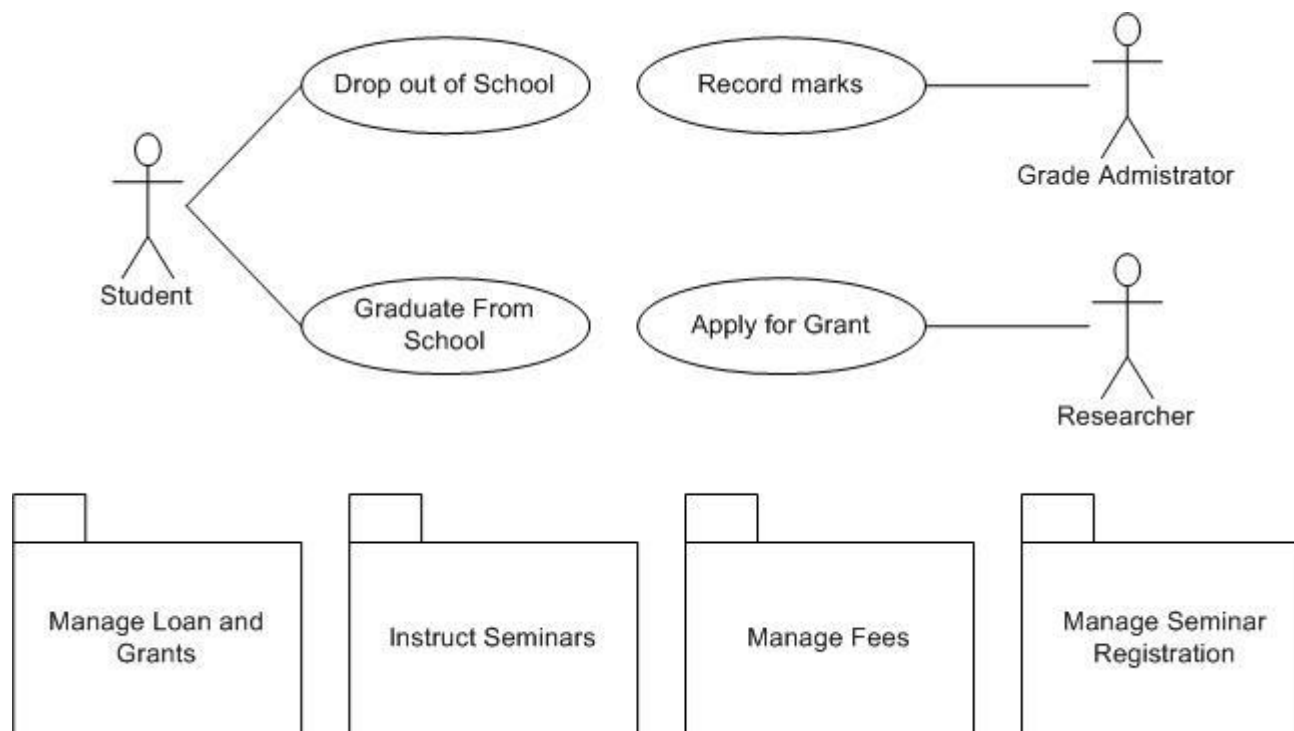


Figura 1 Diagrama cazurilor de utilizare (Use Case Diagram)

Principalele părți ale UML

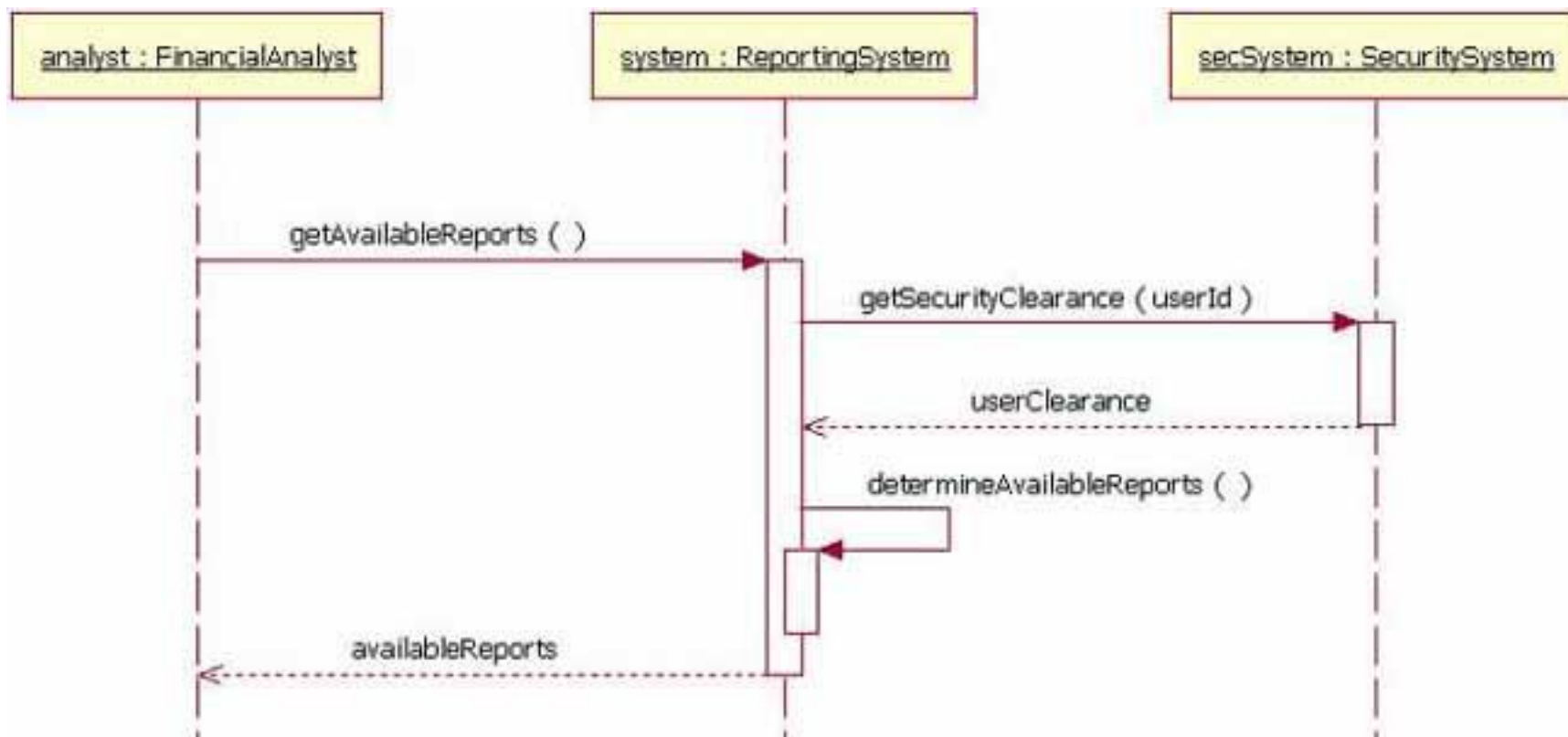


Figura 2 Diagrama de secvență (Sequence Diagram)

Principalele părți ale UML

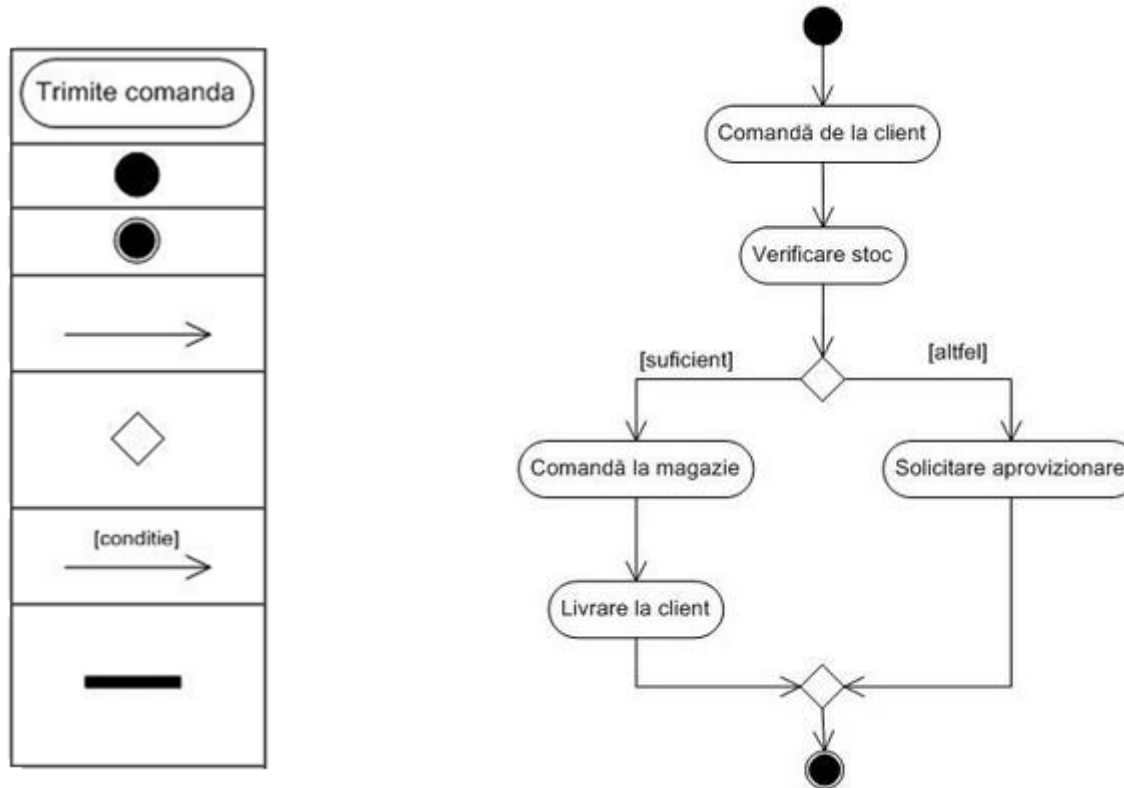


Figura 3 Diagrama de activitate (Activity Diagram)

Principalele părți ale UML (6/13)

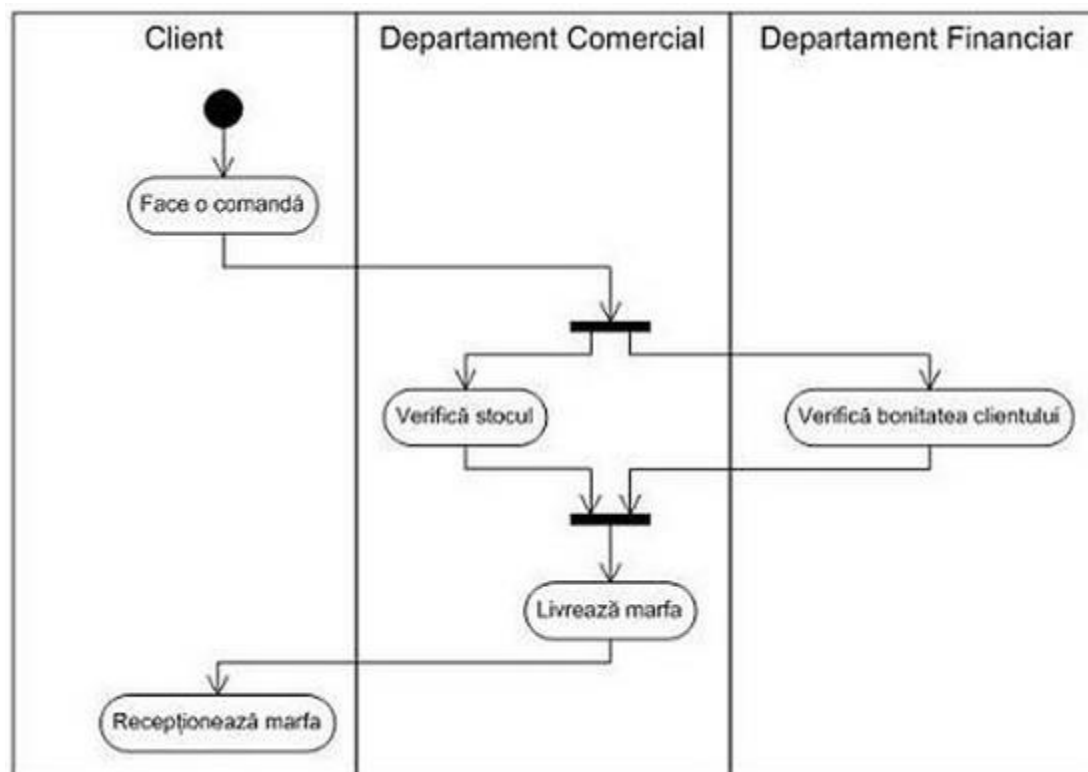


Figura 4 Diagrama de activitate (Activity Diagram)

Principalele părți ale UML

- La modelarea unei componente, se poate utiliza o vizualizare de tip black-box sau white-box, împreună cu anumite interfețe oferite sau generate de componentă
- Componentele se pot lega între ele prin așa numiții “assembly connectors” și pot conduce la construirea unui sistem deosebit de complex

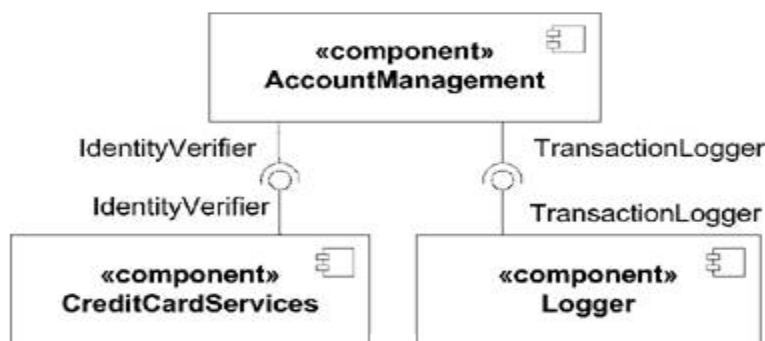


Figura 5 Diagrama componentelor (Component Diagram)

Principalele părți ale UML

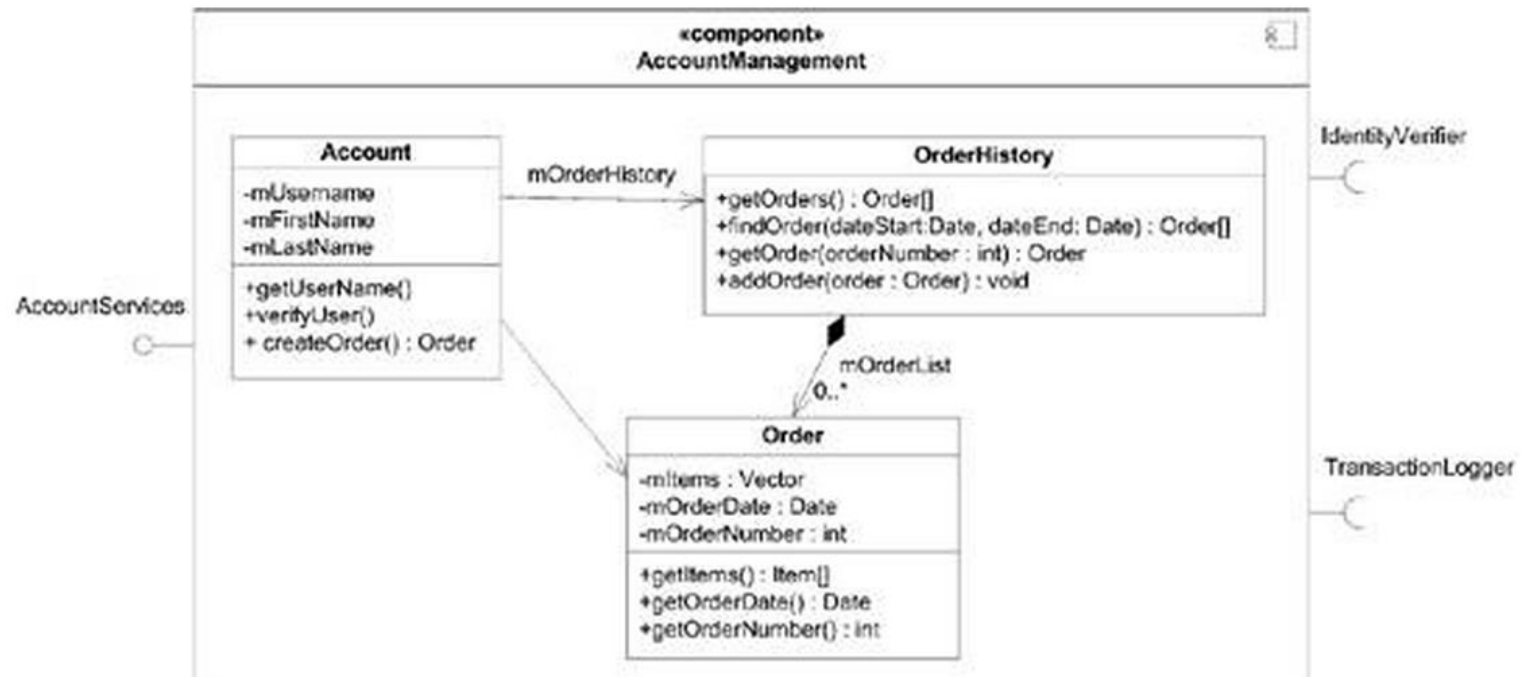


Figura 6 Diagrama componentelor (Component Diagram)

Principalele părți ale UML

- Clasa reprezentată printr-un dreptunghi cu 3 compartimente ce conține numele, attributele și metodele acesteia
- Notăția pentru moștenire, indica clasa moștenita
- Notăția pentru asociere (de tip unu la unu sau unu la mai mulți)
- Dependența: apare când o clasă este utilizată ca și atribut de altă clasă
- Relațiile de agregare și compoziție, se utilizează când o clasă părinte are clase copil

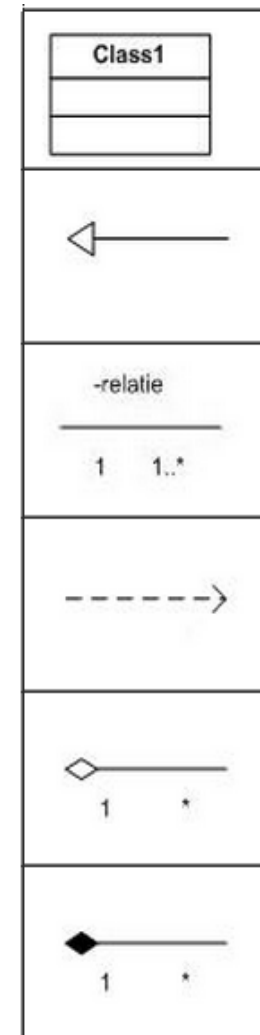
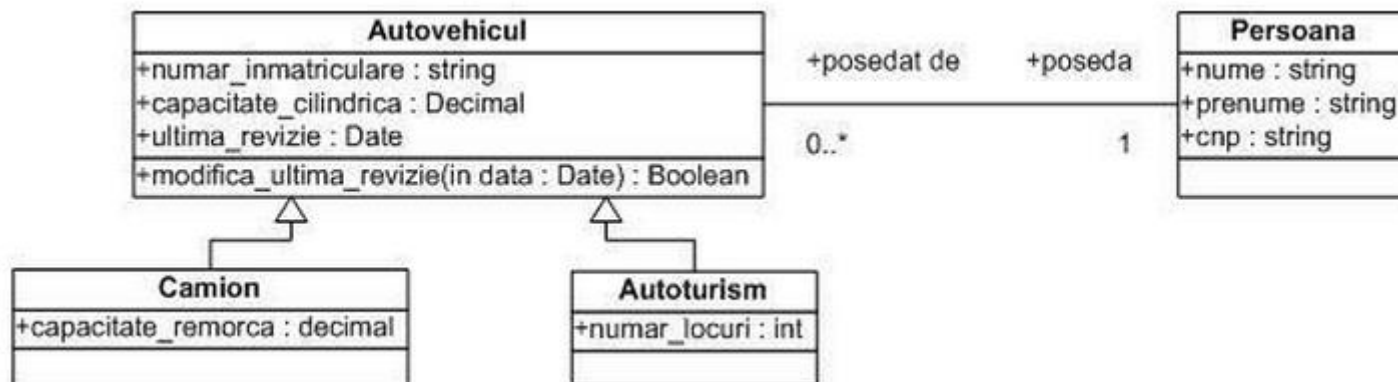


Figura 7 Diagrama de clase (Class Diagram)

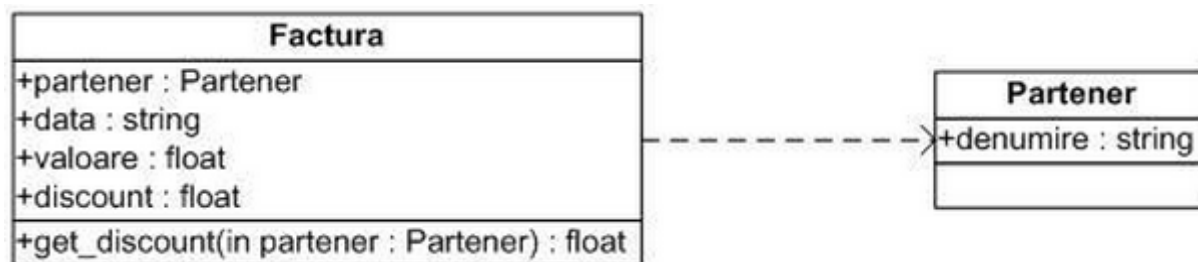
Principalele părți ale UML



- In figură este reprezentată clasa Autovehicul, moștenită de cele două clase Camion și Autoturism și care este în relație cu clasa Persoana. Se observă câteva attribute specificate

Figura 8 Diagrama de clase (Class Diagram)

Principalele părți ale UML



Dependență,



Agregare,



Compoziție

Figura 9 Diagrama de clase (Class Diagram)

Principalele părți ale UML

- Diagrama de desfașurare este utilizată pentru a modela amplasarea fizică a artefactelor pe noduri.
- Astfel, pentru a descrie un web site, o diagramă de desfașurare indică ce componente hardware (noduri) există: web server, application server, database server si ce componente software (artefacte) rulează pe fiecare nod: web application, database.
- Prin artefact se înțelege un fișier binar, un fișier sursa, un script, word, o parte fizică de informație.

Principalele părți ale UML

- Un nod reprezintă o resursă de calcul care permite amplasarea unui artefact pentru execuție. Exemple de noduri de tip device: calculator, telefon, alte dispozitive. Noduri de tip execution environments: Operating System, JVM.

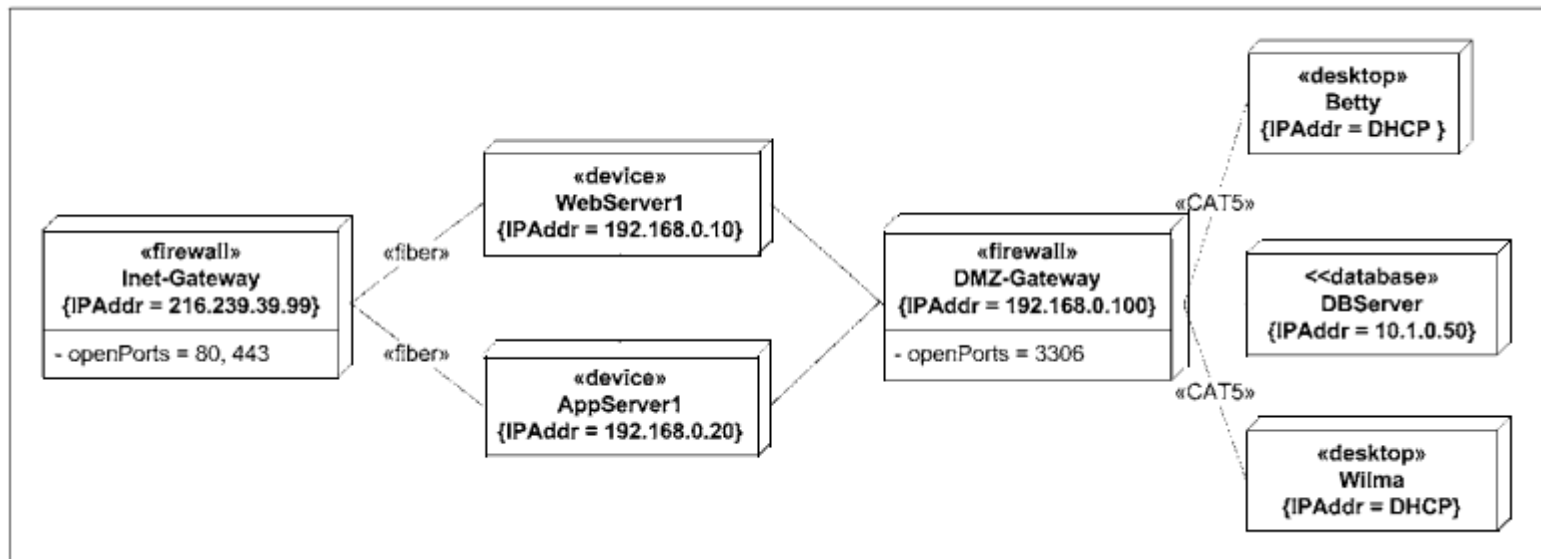


Figura 10 Diagrama de desfășurare (Deployment Diagram)

Generarea diagramelor UML

- Două dintre cele mai cunoscute unelte pentru generarea diagramelor UML sunt StarUML si Altova. Se pot genera diagrame si utilizând un plugin pentru Eclipse.



- StarUML: freeware, independent de mediul de dezvoltare, rulează in Windows.
- ALTOVA : comercialware, dispune de mai multe opțiuni, independent de mediul de dezvoltare, rulează in Windows.

Generarea diagramelor UML

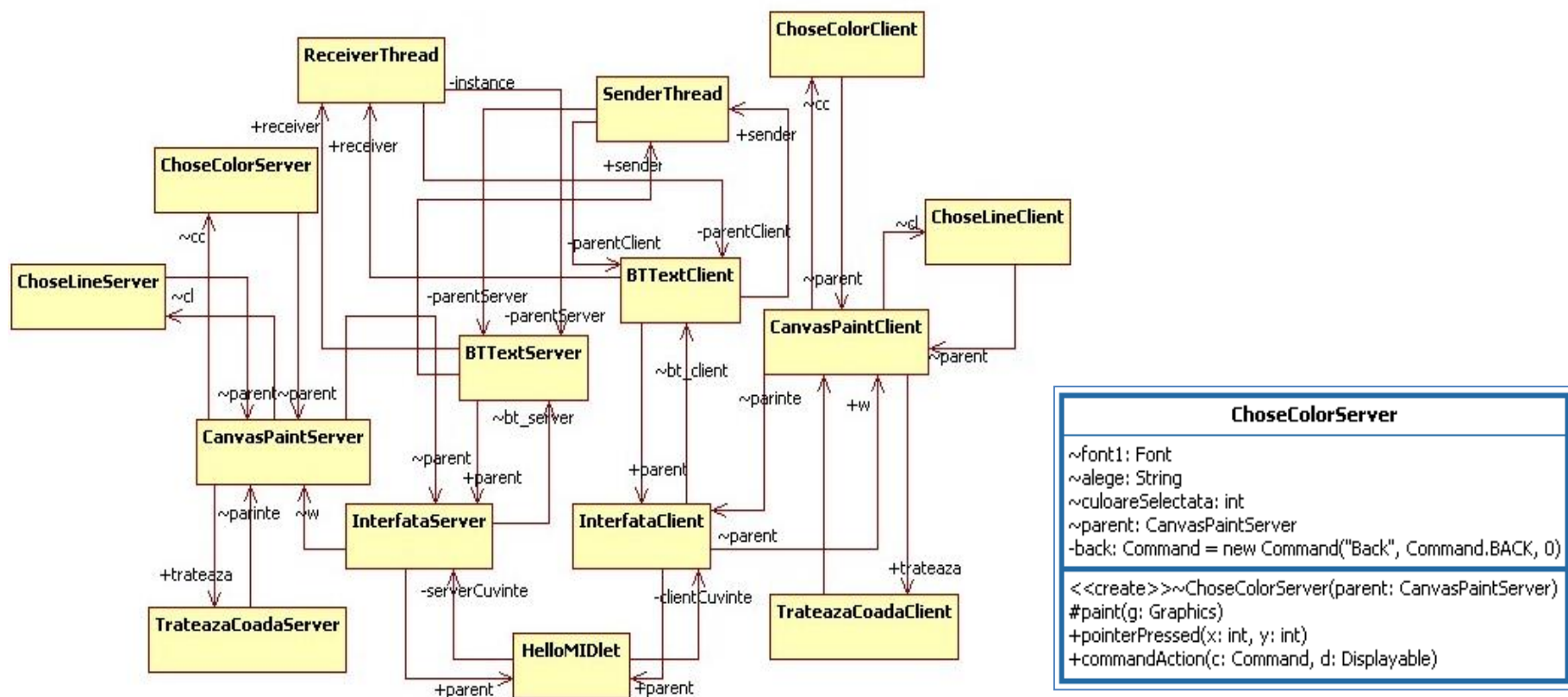


Figura 11 Diagrama de clase (Class Diagram)