Laborator 7

Organizarea modelelor în pachete

Diagrame de implementare

Organizarea modelelor în pachete

Pachetul (package) este o grupare de elemente ale unui model (use case-uri, clase etc.) şi reprezintă baza necesară controlului configuraţiei, depozitării şi accesului. Este un container logic pentru elemente între care se stabilesc legături.

Pachetul defineşte un *spaţiu de nume*.

Toate elementele UML pot fi grupate în pachete (cel mai des pachetele sunt folosite pentru a grupa clase). Un element poate fi conţinut într-un singur pachet.

Un pachet poate conţine subpachete, deci se creează o structură arborescentă (similară cu organizarea fişierelor/directoarelor).

Notaţia grafică pentru pachet este prezentată în figura 1.

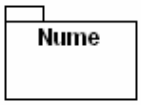


Figura 1. Notaţia grafică pentru pachet

Pachetele pot face referire la alte pachete, iar modelarea se face folosind unul din stereotipurile <<import>> (import public) şi <<acces>> (import privat) asociate unei relaţii de dependenţă (vezi figura 2).

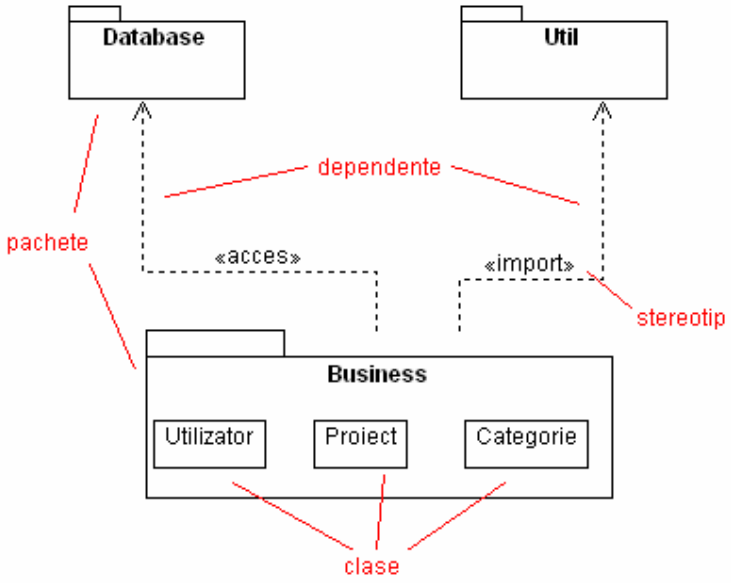


Figura 2. Exemple de relaţii care se pot stabili între pachete

Ambele tipuri de relaţii permit folosirea elementelor aflate în pachetul destinaţie de către elementele aflate în pachetul sursă fără a fi necesară calificarea numelor elementelor din pachetul destinaţie.

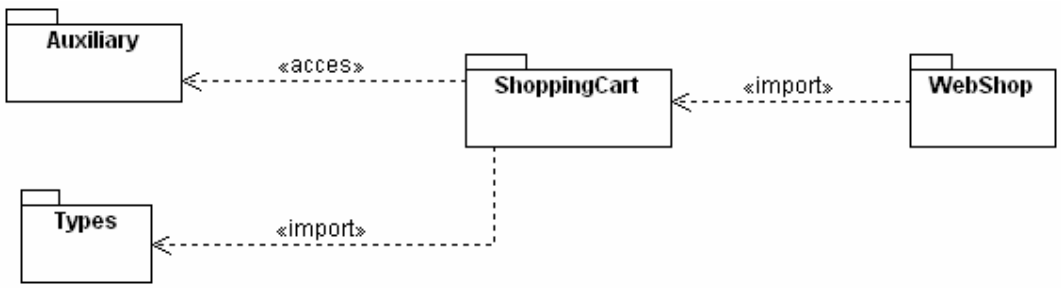


Figura 3. Exemplu de diagramă de pachete

Elementele din `Types` sunt importate în `ShoppingCart` şi apoi sunt importate mai departe de către `WebShop`. Elementele din `Auxiliary` pot fi accesate însă doar din `ShoppingCart` şi nu pot fi referite folosind nume necalificate din `WebShop`.

Utilitatea pachetelor. Pachetele împart sistemele mari în subsisteme mai mici şi mai uşor de gestionat. De asemenea permit dezvoltare paralelă iterativă şi definirea unor interfețe clare între pachete promovează refolosirea codului (ex. pachet care oferă funcții grafice).

Diagrama de componente

Diagrama de componente este o diagramă de implementare care modelează dependențele dintre componentele software ale sistemului şi entitățile care le implementează (fişiere cod sursă, cod binar, executabile, scripturi etc.).

Într-un proiect de dimensiune mare, vor exista multe fişiere care realizează sistemul. Aceste fişiere depind unele de altele. Natura acestor dependențe e dată de limbajul (limbajelor) folosite pentru dezvoltarea proiectului. Dependențele pot exista în momentul compilării, link-editării sau rulării. Există de asemenea dependențe între fişiere sursă şi cele executabile sau obiect, rezultate din primele prin compilare.

În figura 4 este prezentată o diagramă de componente care descrie dependențele dintre o sursă scrisă în C++ şi fişierul header asociat, dependența fişierului obiect de cele două fişiere anterioare şi dependența fişierului executabil de fişierul obiect. Se pot adăuga şi stereotipuri care se folosesc pentru a arăta tipurile diferitelor componente.

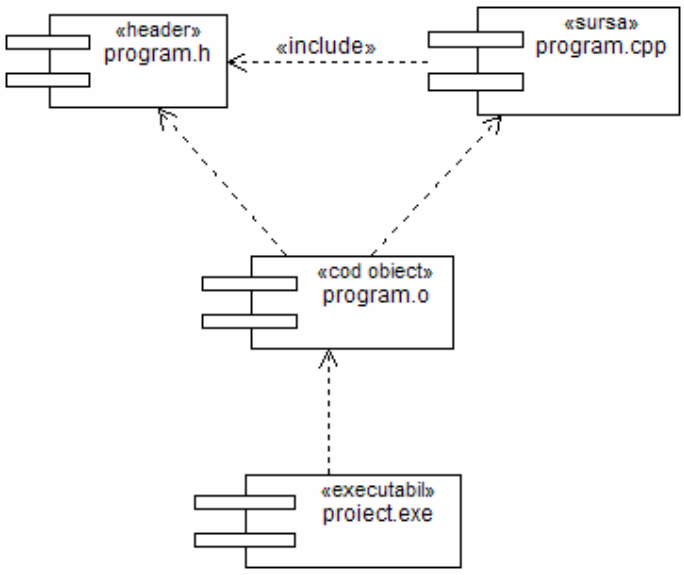


Figura 4. Diagramă de componente care arată dependențele în C++

O alternativă de reprezentare a unei părți a diagramei de componente este de a utiliza notația pentru interfață în UML pentru a arăta specificația unei clase (fişierul header în C++) ca o interfață şi corpul ca o componentă (vezi figura 5).

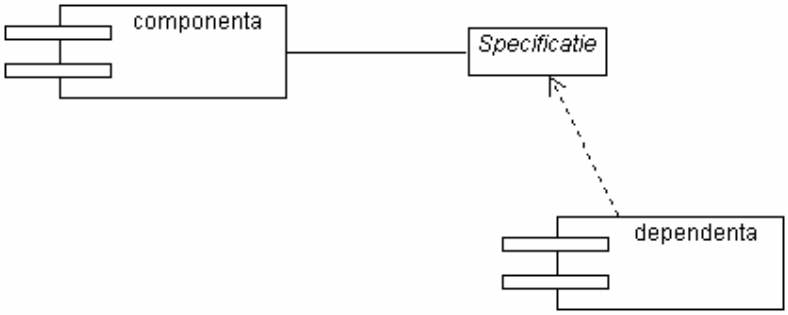


Figura 5. Dependența unei componente printr-o interfață cu o altă componentă

Observații.

- Componentele unei diagrame de componente pot fi componente fizice ale sistemului.
- Diagramele de componente sunt utilizate în general pentru a marca dependențele la scară mare între componentele unui sistem (vezi figura 6).

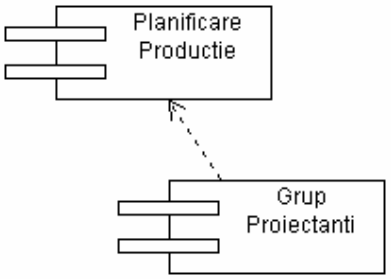


Figura 6. Exemplu de dependență la scară mare între componente unui sistem

- Obiectele active care rulează pe fire de execuție separate pot fi prezentate în diagrama de componente (vezi figura 7)

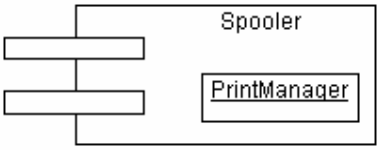
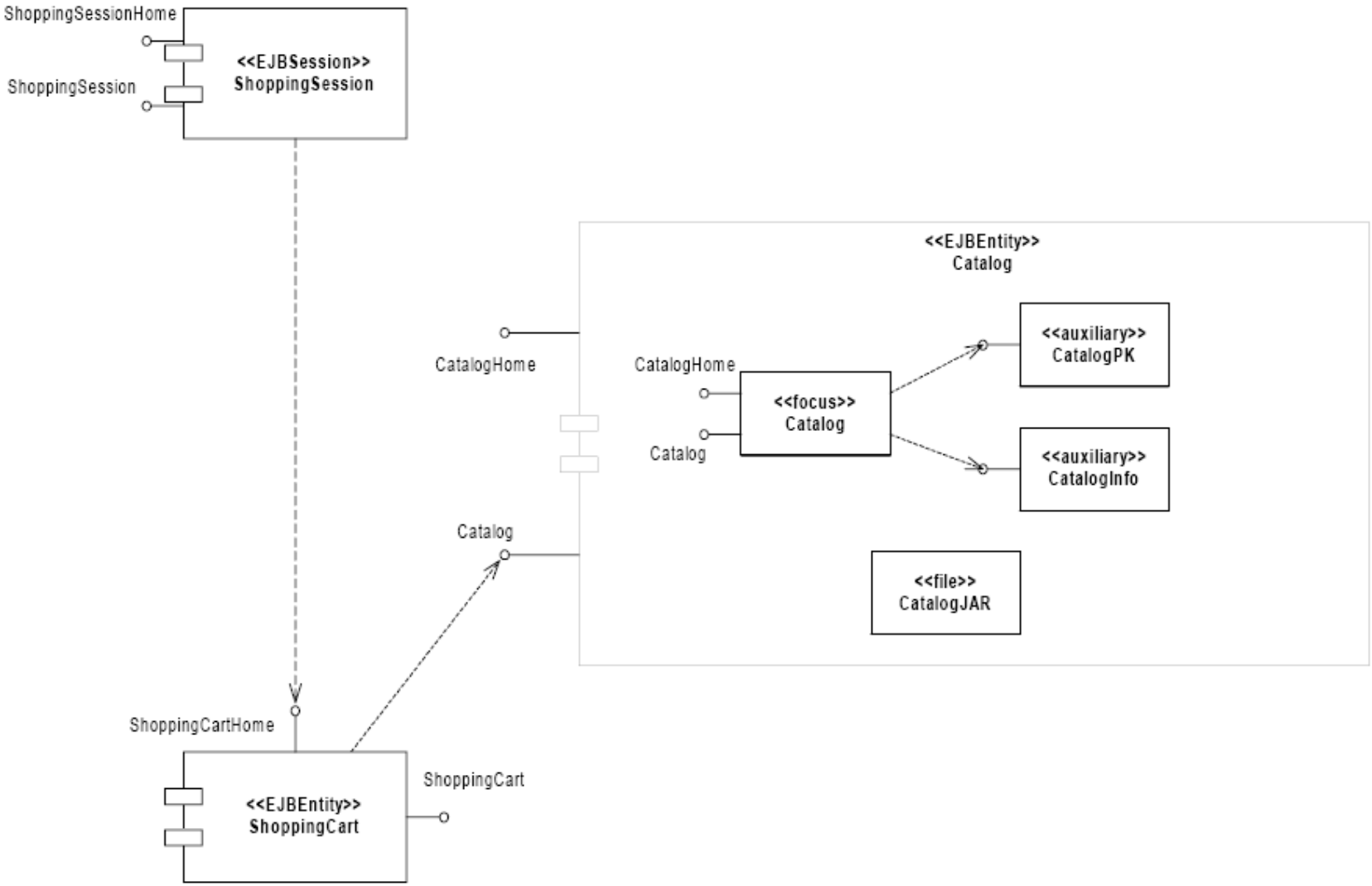


Figura 7. Exemplu de obiect activ în interiorul unei componente

- În timpul analizei şi la începutul proiectării, se folosesc diagramele de pachete pentru a arăta gruparea logică a diagramelor de clase (sau a modelelor care utilizează alte tipuri de diagrame) în pachete referitoare la subsisteme.
- În timpul implementării, diagramele de pachete pot fi folosite a arăta gruparea componentelor fizice în subsisteme.
- Diagrama de componente poate fi combinată cu diagrama de plasare pentru a arăta localizarea fizică a componentelor sistemului. Clasele dintr-un pachet logic pot fi distribuite peste locațiile fizice dintr-un sistem fizic, iar diagramele de componente şi plasare arată tocmai acest lucru.



Exemplu de diagramă de componente

Diagrama de plasare

Diagrama de plasare arată configurația procesării elementelor care execută direct activități specifice şi componentele de program, procesele, obiectele care determină funcționarea acestor componente.

Componentele care nu au rol direct în execuție nu sunt arătate în această diagramă.

Diagrama de plasare este alcătuită din:

- noduri;
- Asocieri.

Nodurile arată computerele iar asocierile marchează rețeaua şi protocoalele care sunt folosite pentru a comunica între noduri (se modelează sistemele client / server din punct de vedere al topologiei).

Nodurile mai pot fi utilizate pentru a modela şi alte resurse cum ar fi personalul uman şi resursele mecanice.

Diagramele de plasare modelează arhitectura fizică a sistemului.

Notația grafică pentru noduri este prezentată în figura 8.

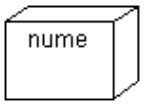


Figura 8. Notația grafică pentru nodurile unei diagrame de plasare

Diagramele de plasare pot arăta fie tipuri de maşini fie instanțe particulare ca în figura 9.

În figura 10 se prezintă locul `bazei de date Vanzari` (pe server) şi câteva componente ale calculatoarelor clienților.

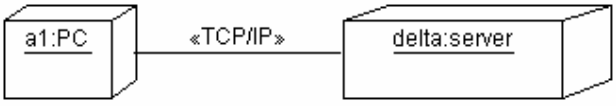


Figura 9. Exemplu de diagramă de plasare

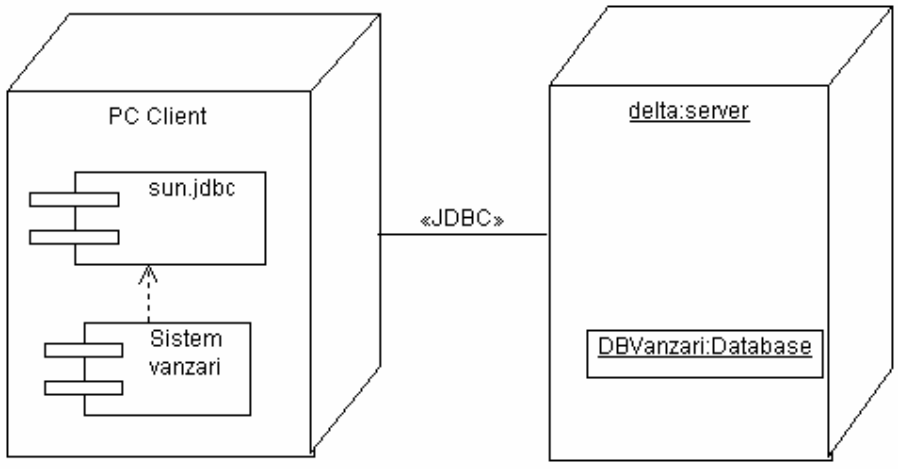
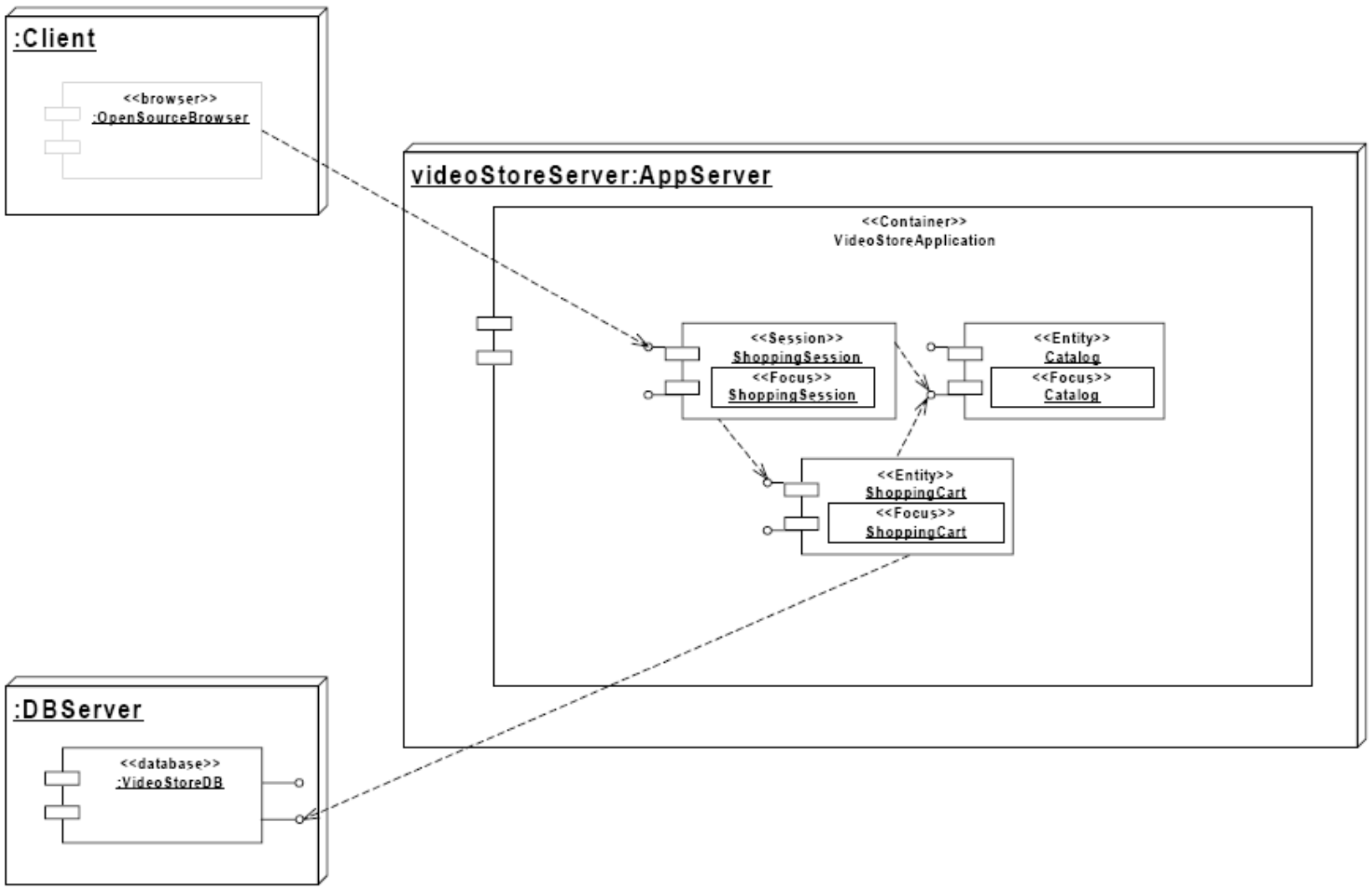


Figura 10. Diagramă de plasare cu componente ale `PC Client`
şi cu un obiect activ pe server



Exemplu de diagramă de plasare