

Inginerie Software 2024
Laboratorul 4
Modelare comportament. Modelarea proceselor
cu Diagrama de Activitate.*

Mădălina Eraşcu, Alexandru Munteanu, Cristian Pal, Ionica Puiu

February 2024

Obiective:

1. Utilizarea diagramei de activitate în diferite contexte.

1 Introducere

O diagramă de activitate furnizează o perspectivă de nivel înalt asupra logicii de execuție a unui **proces**.

Proces = set structurat de activități care interacționează pentru realizarea unui obiectiv.

Diagrama de activitate specifică ce proces trebuie să aibă loc.

Diagrama de activitate poate fi folosită pentru a reprezenta procese de diferite nivele de abstractizare.

Nivele de abstractizare tipice ale procesului reprezentat:

1. analiza și proiectarea unui proces business
2. proiectarea fluxului logic al unui caz de utilizare
3. proiectarea fluxului logic al unei operații complexe

2 Diagrama de activitate

Diagrama de activitate este o reprezentare de tip graf format din noduri interconectate prin *arce*. Arcele reprezintă fluxul între noduri. Fluxul poate fi unul de control sau de date. Nodurile pot fi de mai multe categorii.

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

2.1 Categorii ale nodurilor

Executabile

- **Activitate:** unitatea de comportament de nivel înalt dintr-o diagramă de activitate, ce poate fi descompusă în subactivități și acțiuni.
- **Acțiune:** unitatea atomică de comportament executată în contextul unei activități.

Control – Execuția uneia sau mai multor acțiuni este direcționată printr-un **flux de control**. Nodurile de control – divizează sau unesc fluxul de control pe bază de puncte de decizie sau de posibilități de paralelizare.

Obiect – Acțiunile pot transfera valori sub formă de **fluxuri de obiecte**, adică obiectele ce conțin datele produse (ca ieșire) de către o acțiune și consumate (ca intrare) de o altă acțiune.

3 Notății de bază

3.1 Noduri executabile

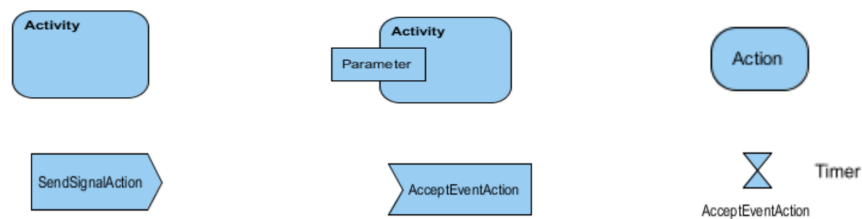


Figure 1: Noduri executabile

3.2 Noduri de control



Figure 2: Noduri de control

3.3 Noduri obiect

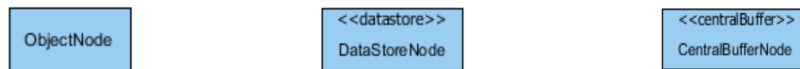


Figure 3: Noduri obiect

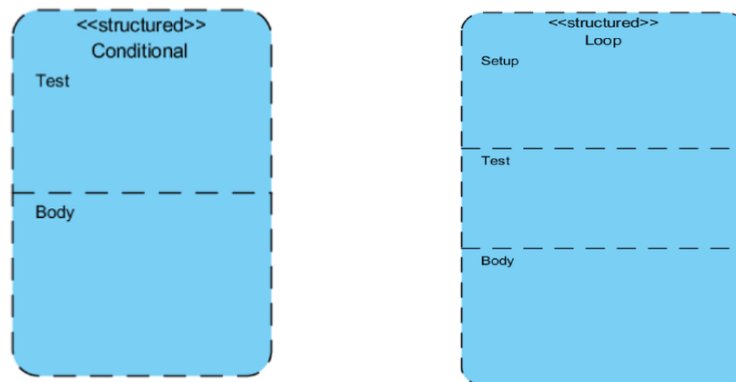
3.4 Activități structurate

Activitățile reprezintă un comportament specificat ca secvență de unități subordonate, folosind un model de flux de control și de date.

Pentru reprezentarea structurilor de control se pot folosi activități structurate în cadrul cărora se vor amplasa noduri și arce conform logicii activității modelată.

Conditional = activitate structurată reprezentând variante din care se alege cea care îndeplinește condiția.

Loop = activitate structurată ce reprezintă o buclă iterativă constând din 3 părți : inițializare (setup), test și conținut (body).



(a) Conditional

(b) Loop

Figure 4: Notății UML pentru conditional si loop

3.5 Detalii și exemple

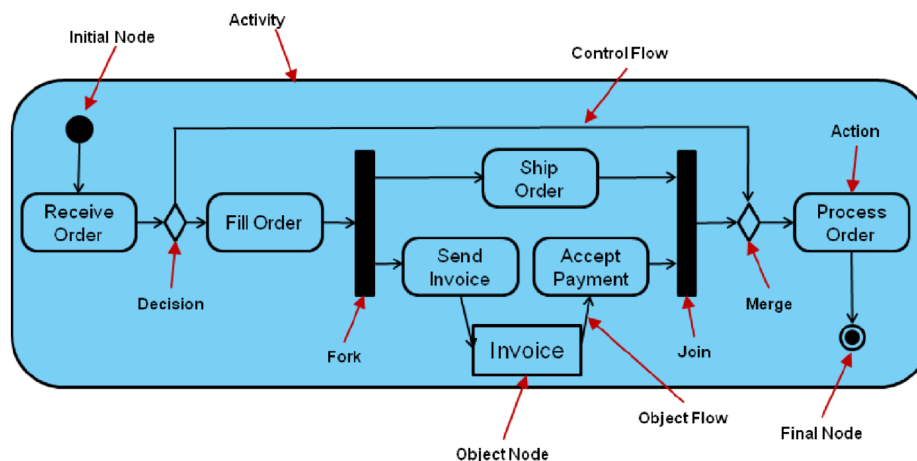


Figure 5: Exemplu - diagrama de activitate

Nod de decizie – specifică un punct în care logica trebuie să facă o alegere între elementele unui set de variante reciproc exclusive.

- un arc de intrare.
- două sau mai multe arce de ieșire, dar orice cale prin nodul de decizie trebuie să urmeze numai una din căile de ieșire.
- condiție gardă – (în general expresie booleană plasată între paranteze pătrate) - pe fiecare arc de ieșire, reprezentând condiția în care este aleasă calea respectivă.

Nod de unificare – punct din care mai multe căi alternative încep să urmeze o cale unică.

- mai multe arce de intrare;
- un singur arc de ieșire.

Remarcă. *UML 2.0 permite combinarea nodurilor de unificare și de decizie pentru a simplifica diagrama; astfel se pot folosi noduri cu mai multe arce de intrare și mai multe arce de ieșire.*

3.6 Noduri fork și join pentru modelarea concurenței

Diagrama de activitate utilizează notația **fork** și **join** pentru a modela procese sau fire de execuție paralele sau concurente.

Un nod **fork** denotă începutul unei **procesări paralele**. Este reprezentat ca bară cu un arc de intrare și două sau mai multe arce de ieșire. Arcele de ieșire se pot executa în orice ordine, inclusiv simultan. Date transferate prin arcul de intrare sunt multiplicare și trimise pe fiecare arc de ieșire.

Un nod **join** denotă terminarea unei procesări paralele; procesarea va continua doar după ce toate fluxurile de intrare ajung la nodul join. Este reprezentat ca bară cu două sau mai multe arce de intrare și un arc de ieșire.

Nodurile fork și join pot fi combinate pentru a modela situații în care sincronizarea unui set de fire de execuție sau de procese este urmată imediat de divizarea în noi fire de execuție sau procese multiple.

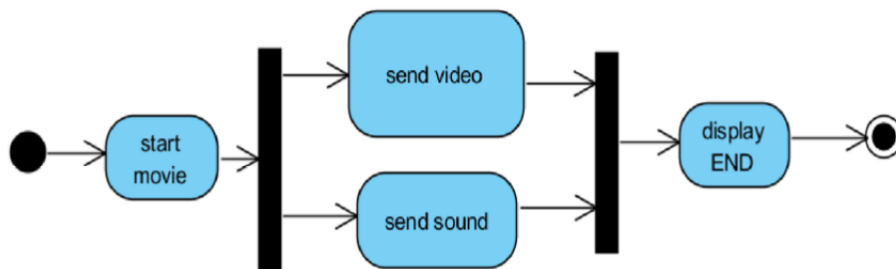


Figure 6: Exemplu: nodurile fork și join.

3.7 Noduri obiect și fluxuri de obiecte

Fluxul de obiecte modelează un **transfer de date** de la o acțiune la alta. Datele sunt conținute într-un **nod obiect**.

Un flux de obiecte este reprezentat sub formă de săgeată. Un nod obiect este un dreptunghi cu numele obiectului.

Un nod obiect poate identifica un singur obiect sau un set de obiecte (ex. DataStoreNode, CentralBufferNode).



Figure 7: Exemplu: nod obiect.

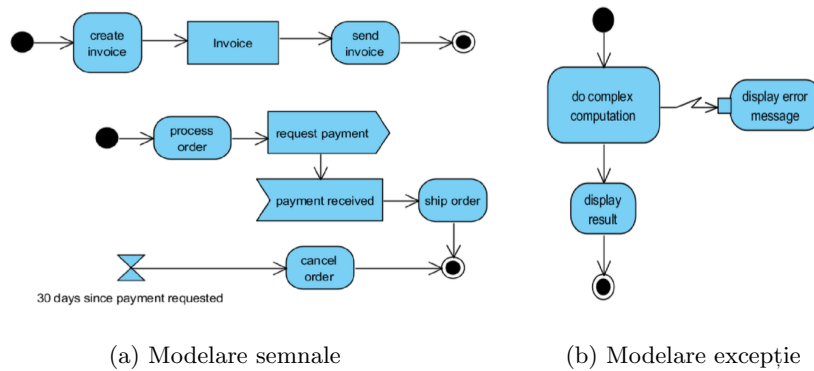


Figure 8: Modelarea semnalelor si a exceptiilor

4 Partiționarea unei activități

Diagrama swimlane - partițiile activității:

Oferă o perspectivă organizațională a nodurilor și arcelor din cadrul unei activități.

1. Utilizată pentru a evidenția responsabilitățile diferiților actori ai unui caz de utilizare. Arcele pot ilustra modul în care responsabilitățile trec de la o partiție la alta.
2. Utilizată pentru a evidenția fluxul principal și fluxurile alternative într-un caz de utilizare.

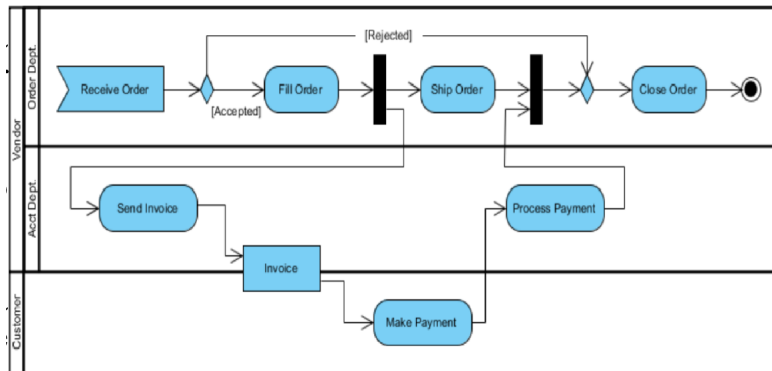


Figure 9: Diagrama Swimlane

5 Modelare scenarii ale cazurilor de utilizare

Considerăm următoarea descriere inițială a unui caz de utilizare.

Observați că fluxul principal și fluxurile alternative nu sunt încă diferențiate.

Nume UC: SelectPerformance (selectare spectacol al unui eveniment artistic)

Actor : Client

Precondiții: Niciuna.

Declanșator: Cazul de utilizare se lansează la cerere.

Descriere: Clientului trebuie să i se dea un set implicit de informații despre spectacolele planificate în următoarele 20 de zile. Trebuie, de asemenea, să i se dea lista tuturor evenimentelor (artistice) planificate la sală (de spectacole). Când utilizatorul selectează un eveniment (artistic), sistemul trebuie să-i ofere un set de spectacole planificate cu acel eveniment (evenimentul afișat trebuie să rămână neschimbat). Când utilizatorul selectează un spectacol, sistemul trebuie să-i solicite confirmarea selecției, pentru a evita greșelile. Utilizatorul trebuie să poată cere o listă a spectacolelor dintr-o perioadă dată, obținând la această solicitare o nouă listă de spectacole (evenimentul rămâne afișat neschimbat). Utilizatorul poate încheia cazul de utilizare fără a face o selecție.

Postcondiții: Spectacolul selectat trebuie salvat pentru a fi transferat etapei următoare din fluxul de activități. Ieșirea netă a cazului de utilizare este un spectacol selectat.

Probleme nerezolvate: Trebuie să permitem utilizatorilor să-și stabilească informațiile implicite? Câte elemente să permitem a fi văzute la un moment dat?

Obiectivul dezvoltării **scenariilor** este identificarea tuturor posibilităților logice în fluxul cazului de utilizare. Fiecare segment identifică o singură cale logică în totalul cazului de utilizare.

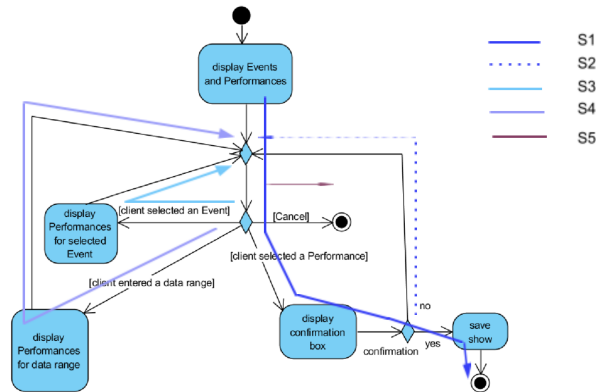


Figure 10: Diagrama de activitate pentru ilustrarea dialogului și scenariilor cazului de utilizare.

Scenariile din exemplu:

- S1: Calea scenariului principal (de succes).
- S2: Scenariul în care se selectează un spectacol dar nu se acordă confirmarea. Este combinat cu S1 pentru completarea definirii: pentru specificarea completă a S2, sunt incluși primii doi pași și decizia din S1 plus unicul pas al S2.
- S3: Clientul a selectat un eveniment. Din lista prezentată de sistem clientul va selecta un spectacol. Astfel S3 este combinat cu S1 pentru finalizare.
- S4: Scenariul în care clientul alege să primească o nouă listă de spectacole bazată pe indicarea unei perioade de timp.
- S5: Scenariul în care utilizatorul alege să anuleze folosirea cazului de utilizare fără a selecta un spectacol.

6 Fluxul datelor

În cadrul diagramei de activitate se pot reprezenta atât **fluxul de control** cât și **fluxul de date**. Figura următoare conține exemplul anterior care a fost documentat suplimentar prin ilustrarea fluxului de obiecte între acțiuni.

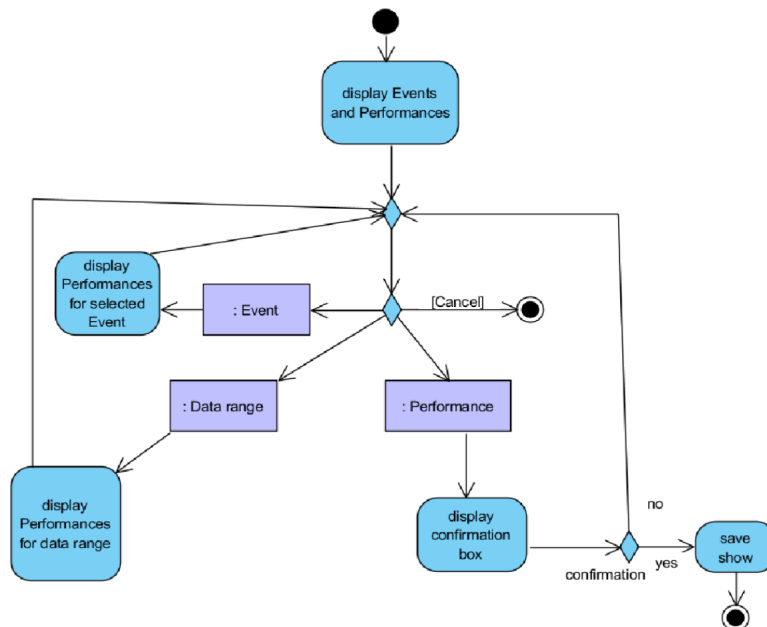


Figure 11: Modelarea scenariilor si fluxul de obiecte între acțiuni.

7 Descrierea procesului business

Modelul procesului business (Diagrame de activitate la nivel de sistem) vs. Modelul funcțiilor business (Diagrama cazurilor de utilizare).

Un **model business** se poate concentra pe **funcțiile** business (Diagrama UC si detalii) sau pe **procese** business (Diagrama de activitate la nivel de sistem).

O diagrama de activitate la **nivel de sistem** (care reprezintă procesul business) reprezintă ordinea si condițiile în care sunt înseriate cazurile de utilizare.

Fiecare caz de utilizare capturează un mod de utilizare a sistemului, adică o funcție business. Deoarece cazurile de utilizare se concentrează mai curând pe funcționalitatea pe care sistemul o va oferi decât pe procesul business pe care trebuie să îl suporte, sistemul rezultat va fi centrat pe funcții. În consecință, această abordare este potrivită pentru dezvoltarea de aplicații verticale, pe când modelul procesului business este potrivit pentru aplicațiile enterprise.

Diferențele cheie dintre modelul funcțional si modelul procesului sunt:

- Activitățile din modelul procesului devin cazuri de utilizare in modelul funcțional.
- Unele acțiuni devin cazuri de utilizare de sine stătătoare.
- În modelul funcțional nu există o noțiune clară de flux de control sau condiții

8 Studiu

- <http://www.agilemodeling.com/style/activityDiagram.htm>
- <https://www.visual-paradigm.com/VPGallery/diagrams/Activity.html>
- https://www.visualparadigm.com/support/documents/vpuserguide/94/2580_activitydiag.html

Secțiunile Drawing activity diagrams și Splitting control flow in activity diagram

9 Temă

1. Desenați diagrama de activitate pentru fiecare caz de utilizare definit pentru SISTEM ÎNCHIRIERE MAȘINI (descrie în cadrul laboratorului 2, ex. 6).
2. Construiți o diagramă **swimlane** care să reprezinte următorul proces:
 - Un client solicită un produs de la un vânzător.
 - Vânzătorul consultă agentul de inventar pentru a vedea dacă produsul este disponibil.

- Vânzătorul răspunde clientului.
- Dacă produsul este disponibil clientul plasează un ordin către vânzător, altfel procesul se încheie.
- Vânzătorul trimite ordinul către agentul de inventar.
- Agentul de inventar pune produsul în așteptare și informează vânzătorul.
- Vânzătorul emite factura.
- La recepția plății pentru factură vânzătorul solicită agentului de inventar să facă livrarea.
- Agentul de inventar trimite clientului o notă de livrare.