# Diagrama de activități (activity diagram)

Pentru modelarea procesului de executare a operațiilor în limbajul UML se utilizează așa numitele diagrame de activități. Notația grafica acceptată pentru aceste diagrame are mult comun cu notația diagramei de stări, fiindcă în diagrama de activități la fel se utilizează stările și tranzițiile. Deosebirea constă în semantica stărilor, care sunt utilizate pentru prezentarea acțiunilor, dar nu a activităților, și în aceea că tranzițiile evenimentelor nu sunt etichetate (nu conțin eveniment). Fiecare stare în diagrama de activități corespunde executării unei operații elementare, iar trecere în următoarea stare se execută numai la finalizarea operației în starea actuală. Grafic diagrama de activități se reprezintă în forma unui graf, nodurile – stări activitate și muchile – tranzițiile de la o stare la altă.

În așa fel, diagramele de activități pot fi considerate cazuri particulare ale diagramelor de stări. Anume aceste diagrame ale limbajului UML permit realizarea administrării procedurale și sincrone, care depinde de finalizarea acțiunilor interne a stării. Sensul principal al utilizării acestor diagrame constă în vizualizarea particularităților operațiilor clasei, când este necesar de a reprezenta algoritmul executării lor. Totodată fiecare stare poate fi reprezentată ca realizarea operațiilor unei clase, ceea ce permite utilizarea diagramei de activități pentru descrierea reacției ei la evenimentele interne a acestui sistem.

În contextul limbajului UML activitatea (activity) reprezintă o totalitate de operații (calcule) executate de către automat. Ca urmare operațiile (calculele) elementare pot duce la un anumit rezultat sau careva acțiune (action). În diagrama de activități se reflectă logica sau consecutivitatea tranzițiilor de la o acțiune la alta, totodată se focusează pe rezultatul activității. Rezultatul, la rândul său poate duce la schimbarea stării sistemului dat sau la returnarea unei valori.

## Starea activitate

Starea activitate (action state) este un caz particular a stării. Starea activitate nu poate avea tranziții interne fiindcă ea este elementară. Starea activitate se utilizează pentru modelarea unui pas de executarea a algoritmului (procedurii) sau a unui flux de control.

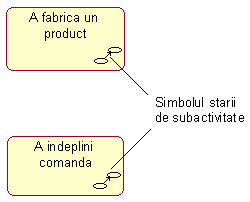
Grafic starea activitate se reprezintă printr-o figură asemănătoare cu un dreptunghi, laturile laterale ale căruia sunt substituite cu arcuri convexe (printr-un dreptunghi cu colțuri rotunjite) (fig. 1). În interiorul acestei figuri se indică expresia unei acțiuni care trebuie să fie unică în cadrul unei diagrame de activități.



**Fig. 1.** Stare activitate (a – acțiune simplă, b – expresie).

Acțiunea poate fi indicată în limbaj natural, în pseudocod sau în limbaj de programare. Nu sunt restricții suplimentare pentru indicarea acțiunilor. Se recomandă în calitate de nume al unei acțiuni simple să se utilizeze un verb cu cuvinte explicative (fig. 1, a). Dacă acțiunea nu poate fi reprezentată într-un mod formal, atunci este util a-l indica într-un limbaj de programare în care va fi realizat proiectul (fig. 1, b).

Uneori este necesar a reprezenta în diagrama de activități o acțiune complexă care la rândul său constă din mai multe acțiuni simple. În acest caz poate fi utilizată notația specială a stării de subactivitate (subactivity state). Așa fel de stare este un graf de activitate care se notează cu un semn special în colțul drept de jos (fig. 2). Această construcție poate fi aplicată oricărui element al limbajului UML care susține ”gruparea” pentru structura sa. Totodată semnul special poate fi etichetat cu tipul structurii.



**Fig. 2.** Starea subactivităţii.

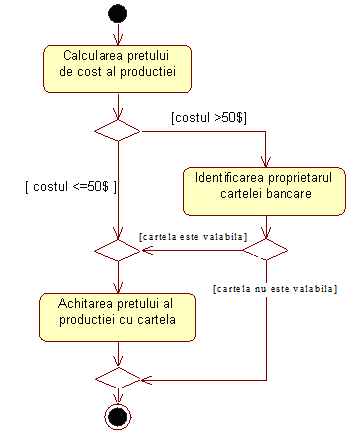
Orice diagrama de activități poate să conțină doar o singură stare inițială și o singură stare finală.

## Tranziţii

Tranziția ca element al limbajului UML a fost studiată în capitolul despre diagrama de stări. La construirea diagramei de activități se utilizează doar tranziții nontrigere care acționează deodată după finalizarea activității sau după executarea acțiunii corespunzătoare. Această tranziție transmite activitatea în următoarea stare imediat după ce se termină acțiunea din starea precedentă. În diagrama această tranziția se reprezintă printr-o linie continuă cu o săgeată.

Dacă din starea dată iese numai o tranziție atunci ea poate să nu fie marcată (indicată), iar dacă tranzițiile de ieșire sunt mai multe atunci poate avea loc doar una din ele. Anume în acest caz pentru fiecare din tranziții trebuie să fie indicată în paranteze pătrate o condiție gardă (sau mai multe, atunci ele se separă prin ”,”). Totodată pentru toate tranzițiile de ieșire poate să fie executată numai una singură, acea care va conține valoare de adevăr. Acest caz are loc când consecutivitatea activităților executate trebuie să fie divizată în ramuri alternative independente în dependență de valoarea rezultatului intermediar. Această situație are denumirea de ramificație. Grafic ramificația se reprezintă printr-un romb gol (fig. 3). În acest romb poate intra numai o singură săgeată de la starea activitate precedentă, după care fluxul de control poate să continue doar pe o singură ramură (tranziție, care va avea valoarea de adevăr). Săgeata de intrare se unește la vârful de sus sau la cel din stingă al simbolului de ramificație. Pot fi mai multe săgeți de ieșire, dar pentru fiecare din ele trebuie să fie indicată condiția gardă sub formă de expresie booleană.

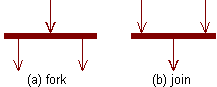
Ca exemplu vom cerceta calcularea costului total al produselor procurate cu o cartelă bancară într-un market (fig. 3). Dacă costul depășește 50$ atunci se va trece la identificarea proprietarului cartelei bancare. În caz dacă cartela este valabilă și contul nu depășește suma în 50$ atunci suma se scoate de pe cont și se achită produsele procurate. În caz contrar (dacă cartela nu este valabilă) achitarea nu se execută și produsele rămân la vânzător.



**Fig. 3.** Variante ale ramificațiilor în diagrama de activități.

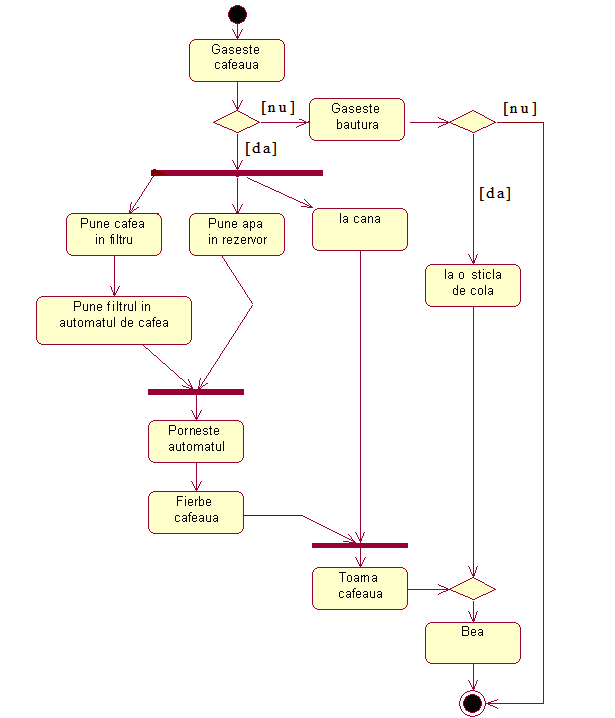
Unul din cele mai semnificative neajunsuri ale schemelor-block sau ale schemelor ce reprezintă algoritmi este legat cu problema reprezentării ramurilor paralele ale unor calcule. Din motiv că divizarea calculelor în ramuri separate mărește viteza produsului soft, sunt necesare primitive grafice pentru reprezentarea proceselor paralele. În limbajul UML pentru acest scop se utilizează simboluri pentru diviziune și unire a calculelor paralele sau a fluxurilor de control. Acest simbol este o linie dreaptă analogic notației unei tranziții în formalismul rețelelor Petri.

De regulă această linie se reprezintă printr-un segment al unei linii orizontale sau verticale, grosimea căreia este mai mare decât grosimea liniilor în diagrama de activități. Totodată fork (divizarea – concurrent fork) are o tranziție de intrare și mai multe de ieșire (fig. 4, a). Join (unirea – concurrent join) invers are mai multe tranziții de intrare și numai o tranziție de ieșire (fig. 4, b).



**Fig. 4.** Fork şi join a mai multor fluxurilor paralele de control.

Pentru ilustrarea particularităților proceselor paralele de executare a acțiunilor este util de a cerceta exemplul devenit clasic: pregătirea unei băuturi. Avantajul acestui exemplu constă în faptul că exemplul practic nu cere explicații adăugătoare, fiindcă este considerat evident (fig. 5).

****

**Fig. 5.** Diagrama de activitate pentru un exemplu de preparare a băuturii.

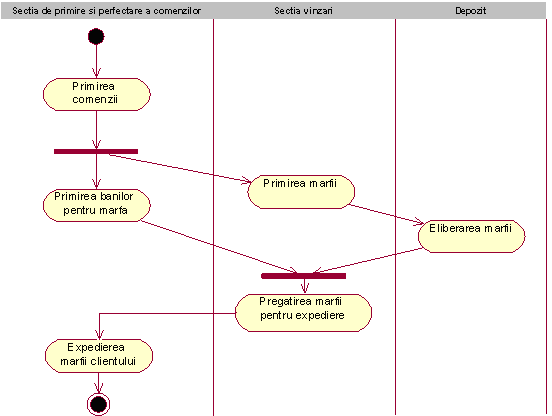
## Partiţii

Diagramele de stări pot fi utilizate nu numai pentru specificarea algoritmelor de calculare sau fluxurilor de control în sistemele de programare. Un domeniu de utilizare este legat cu modelarea business-proceselor. Întra-adevăr, activitatea oricărei companii reprezintă totalitatea acțiunilor independente îndreptate spre atingerea unui anumit rezultat. Totuși, în cazul business-procese este de dorit ca îndeplinirea fiecărei acțiuni (activități) să fie asociată cu o subdiviziune a unei companii concrete. În acest caz aceste subdiviziuni au responsabilitatea de realizare a acestor acțiuni (activități), iar business-procesele reprezintă trecerea acțiunilor (activităților) de la o subdivizare la alta.

Pentru modelarea acestor particularități în limbajul UML se utilizează construcția specială, care are denumire de partiții (swimlanes), care sunt divizate unul de celălalt prin linii verticale. Două linii vecine formează o partiție, iar un grup de stări între aceste linii sunt executate de subdiviziunea separată (secție, filial, departament) a companiei (fig. 6).

Denumirile subdiviziunilor sunt indicate în partea de sus a partiției. A întretăia linia partiției pot numai tranzacțiile, care în acest caz indică ieșirea sau intrarea fluxului de control în subdiviziunea respectivă a companiei. Ordinea trecerii partițiilor nu are care-va informație semantică și este definită după motivele de confortabilitate.

Ca exemplu vom analiza fragmentul diagramei de activitate a companiei de vânzări online, care deservesc clienții la distanță. Subdiviziunile companiei sunt: secția de primire și perfectare a comenzilor, secția vânzări și depozitul. Acestor subdiviziuni vor corespunde trei partiții în diagrama de activitate, fiecare din care specifică zona de responsabilitate a subdiviziunii. În cazul dat diagrama de activitate conține nu numai informația despre consecutivitatea executării acțiunii (activității) lucrătorilor, dar și care subdiviziuni a companiei de vânzări online trebuie să execute o acțiune (activitate) sau alta (fig. 6).



**Fig. 6.** Diagrama de activități a companiei de vânzări online.

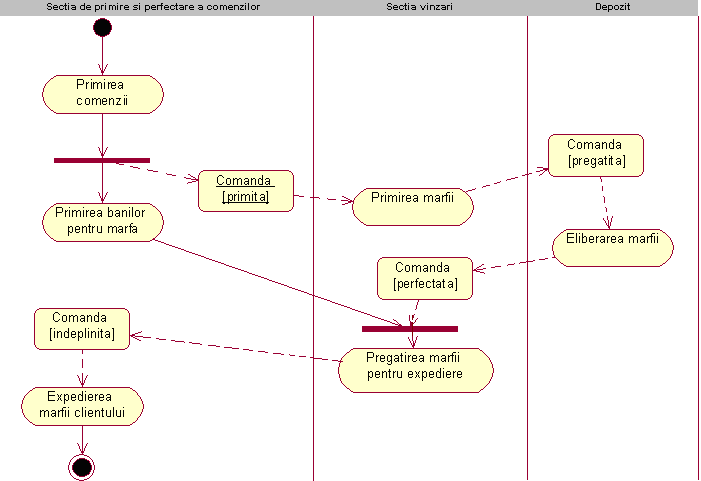
Din această diagramă de activitate este evident că după acceptarea comenzii de la client a secției de primire și perfectare a comenzilor se realizează divizarea activității în două fluxuri (divizarea – concurrent fork). Primul flux rămâne în aceeași subdiviziune și este legat cu achitarea pentru produsele (marfa) selectată. Al doilea flux inițiază procesul de selectare a produsului dorit de către secția vânzări (modelul produsului, dimensiunea, culoarea, anul de editare etc.). După finisarea lucrului acesta inițializează acțiunea de eliberare a produsului de la depozit. Totuși pregătirea produsului pentru expediere de către secția vânzări va avea loc doar după achitarea produsului de către client și eliberarea acestuia de la depozit (unirea – concurrent join). Numai după aceasta produsul va fi expediat clientului și devine proprietatea lui.

## Obiecte

În cazul general acțiunile în diagrama de activitate sunt efectuate asupra unor anumite obiecte. Aceste obiecte sau inițializează executarea acțiunilor sau definesc un anumit rezultat a acestor acțiuni. În urma căruia acțiunile specifică careva apeluri, care sunt transmise de la un obiect a grafului de activitate la altul. Întrucât în așa cazuri obiectele joacă un anumit rol în înțelegerea procesului de activitate, uneori apare necesitatea indicării lor în diagrama de activități. Pentru reprezentarea grafică a obiectelor sunt utilizate dreptunghiurile clasei cu o deosebire: numele obiectului se subliniază. După nume poate fi indicată caracteristica stării obiectului între paranteze pătrate. Aceste dreptunghiuri a obiectelor sunt unite cu stare activitate printr-o relație de dependență (linia întreruptă cu săgeată în capăt). Dependența respectivă definește starea concretă a obiectului după efectuarea acțiunii precedente.

În diagrama de activitate cu utilizarea partițiilor plasarea obiectelor are un sens adăugător. Și anume, dacă obiectul este plasat la hotarul ambilor partiții (pe linia partiției), acest lucru înseamnă că trecerea la starea activitate următoare în partiția vecină este asociată cu un document finit (obiectul în care-va stare) sau cu alte cuvinte de acest obiect ”răspund” ambele partiții. Iar dacă obiectul este amplasat în cadrul partiției, atunci starea acestui obiect este definită de acțiunile partiției date.

Revenind la exemplul precedent cu compania de vânzări online, se poate observa că obiectul central a procesului de vânzări online este comanda, anume starea ei de executare. La început, până la primirea comenzii de la client, comanda ca obiect lipsește și apare numai după ce a avut loc primirea comenzii (prima stare activitate). Totuși, această comandă nu este îndeplinită până la urmă, deoarece este necesar de alege produsul prin intermediul secției vânzări. După pregătirea lui el este transmis la depozit, unde are loc eliberarea produsului de la depozit. Și în sfârșit, după primirea confirmării despre achitarea produsului de către client și perfectarea acestuia este expediat clientului de către secția de primire și perfectare a comenzilor (fig. 7).



**Fig. 7**. Diagrama de activități a companiei de vânzări online cu obiect – comandă.