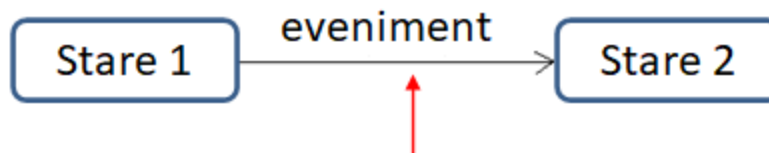


Diagrame de stări (State machine diagrams)

Prof. univ. dr. ing. Florica Moldoveanu

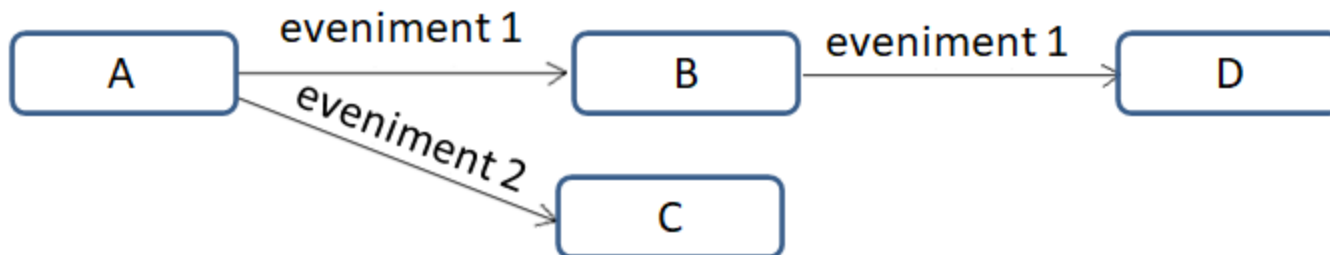
Diagrame de stari - utilizari

- ❖ Se folosesc pentru a reda aspecte dinamice ale unui sistem.
- ❖ O diagrama UML de stări redă stările unei entități într-o perioadă de timp și modul în care entitatea răspunde la diferite evenimente, prin tranziții dintr-o stare în alta.



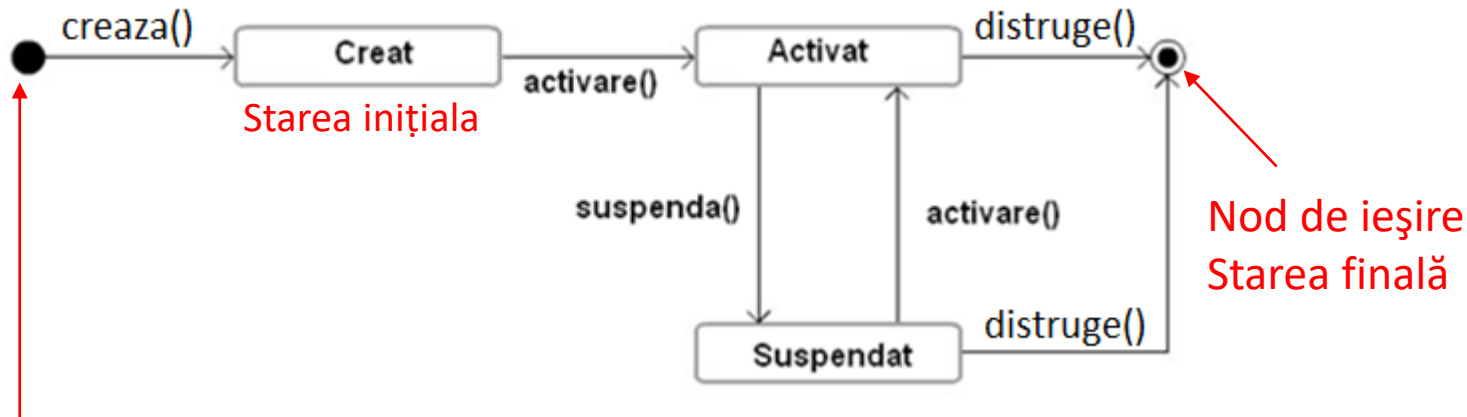
Tranziția din starea 1 în starea 2 datorată evenimentului

- ❖ **Principala utilizare:** documentarea claselor de obiecte cu un comportament reactiv.
- ❖ O diagramă de stări redă comportarea unui obiect dependentă de starea sa curentă.
Un obiect răspunde diferit la un eveniment în funcție de starea sa curentă.



Stari si tranzitii (1)

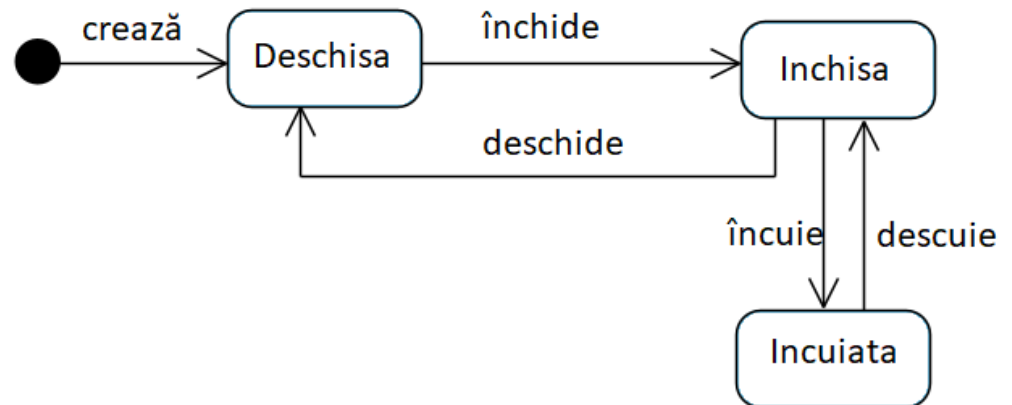
Exemplu: stările unui obiect sale și tranzițiile determinate de apelul operațiilor (clasei) sale.



Nod de intrare (pseudo-starea initiala), din care se trece automat în starea inițială

❖ **Nodul de ieșire poate lipsi dintr-o diagramă de stări.**

Exemplu: stările unei uși
pe parcursul existenței sale.

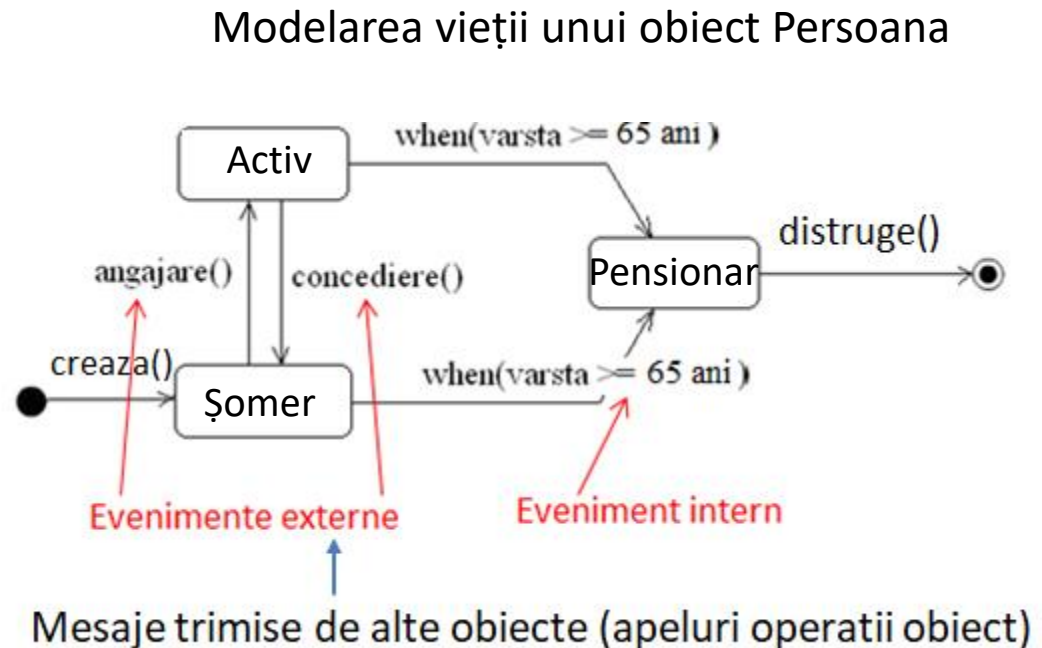
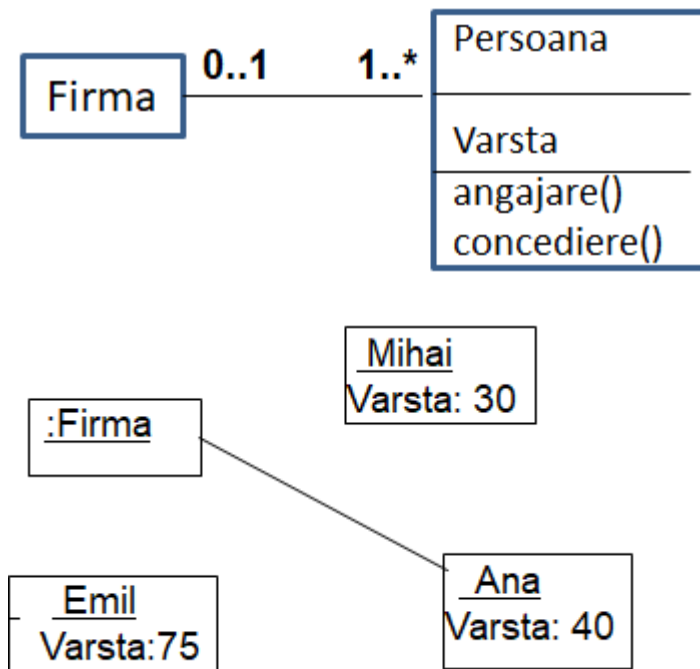


Stari si tranzitii (2)

- Starea curenta a unui obiect este determinata de valorile atributelor sale, inclusiv de legaturile existente intre obiect si alte obiecte.

De exemplu, starea curenta a unei persoane poate fi: în activitate, în somaj sau la pensie.

Este determinata de varsta persoanei si de prezența unei legaturi catre o firma:

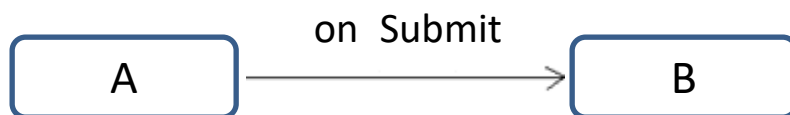


Evenimente (1)

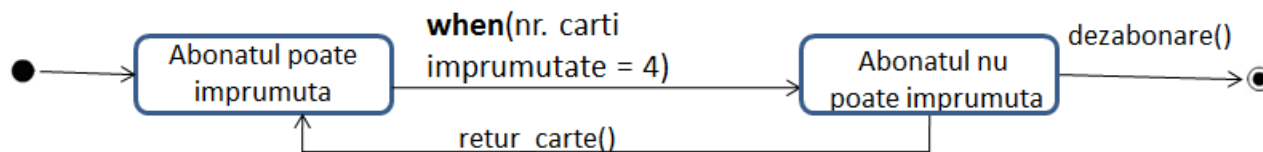
Un eveniment poate fi:

- Receptionarea unui semnal, cum ar fi o exceptie, o notificare, un eveniment generat de interactiunea cu utilizatorul.
- Receptionarea unui apel, adica invocarea unei operatii a clasei obiectului.
- Recunoasterea unei conditii in mediul extern sau in obiectul insusi:
 - conditie predefinita, care este indeplinita la un moment dat: “*change event*”
 - ***when***(*expresie booleana*).
 - trecerea unei perioade de timp desemnate: “*elapsed-time event*”
 - ***after***(*expresie al carei rezultat este o perioada de timp*).
- Un eveniment se reprezinta printr-o eticheta amplasata deasupra tranzitiei pe care o declanseaza.

Evenimente (2)



Eveniment generat de
interactiunea cu utilizatorul



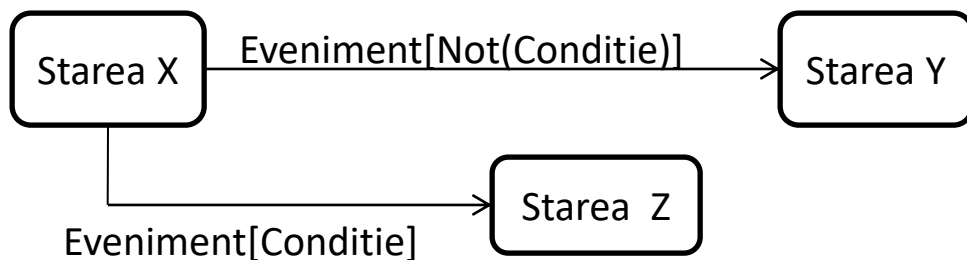
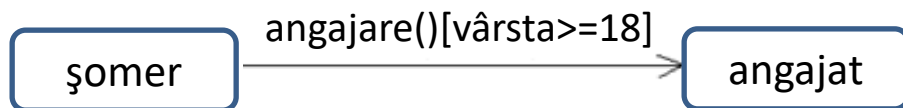
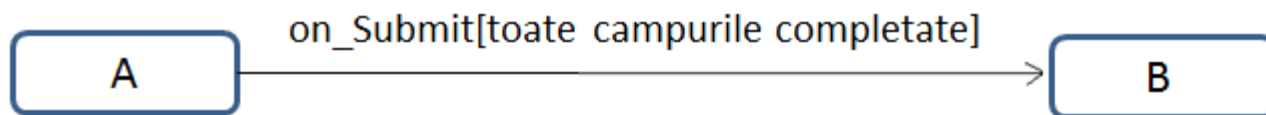
*“change
event”*



“elapsed-time event”

Tranzitii conditionate

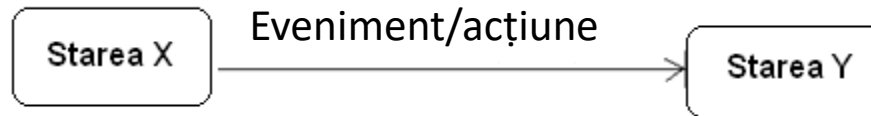
- Tranzițiile pot fi controlate prin garzi. O garda este o expresie booleana care validează declansarea unei tranziții în cazul apariției unui eveniment.



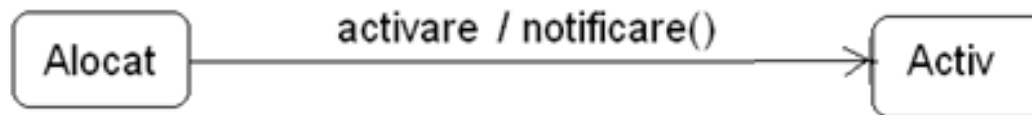
- Garzile asociate unui eveniment care poate produce tranziții diferite din aceeași stare trebuie să fie mutual exclusive.

Actiuni si activitati(1)

- Operatiile unei clase apar in diagramele de stari ca **actiuni si activitati**.
- O actiune este considerata ca instantanee, adica are un timp de executie neglijabil in raport cu dinamica sistemului, și **nu poate fi întrerupta de un eveniment**.
- In diagramele de stari, actiunile sunt atasate tranzitiilor:

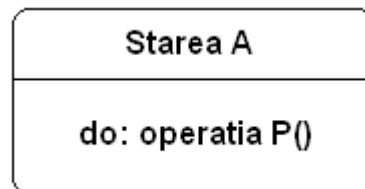


- O actiune defineste modul in care obiectul care receptioneaza evenimentul trebuie sa raspunda la eveniment. Timpul de executie al unei actiuni este nesemnificativ.
- Exemple de actiuni: apelul unei operatii a clasei, crearea sau distrugerea unui alt obiect, trimiterea unui semnal catre un alt obiect.



Actiuni si activitati(2)

- O activitate este o operatie care necesita un anumit timp de executie. Este asociata unei stari.
- Anumite **activitati** sunt **ciclice**, ca afisarea unei imagini pe ecranul unui monitor sau ca soneria telefonului care persista pana cand un eveniment o intrerupe declansand o tranzitie.
- Alte **activitati** sunt **secventiale**, cu o durata finita, ca de exemplu executia unui calcul.
- Activitatile sunt indicate prin cuvantul cheie "do":

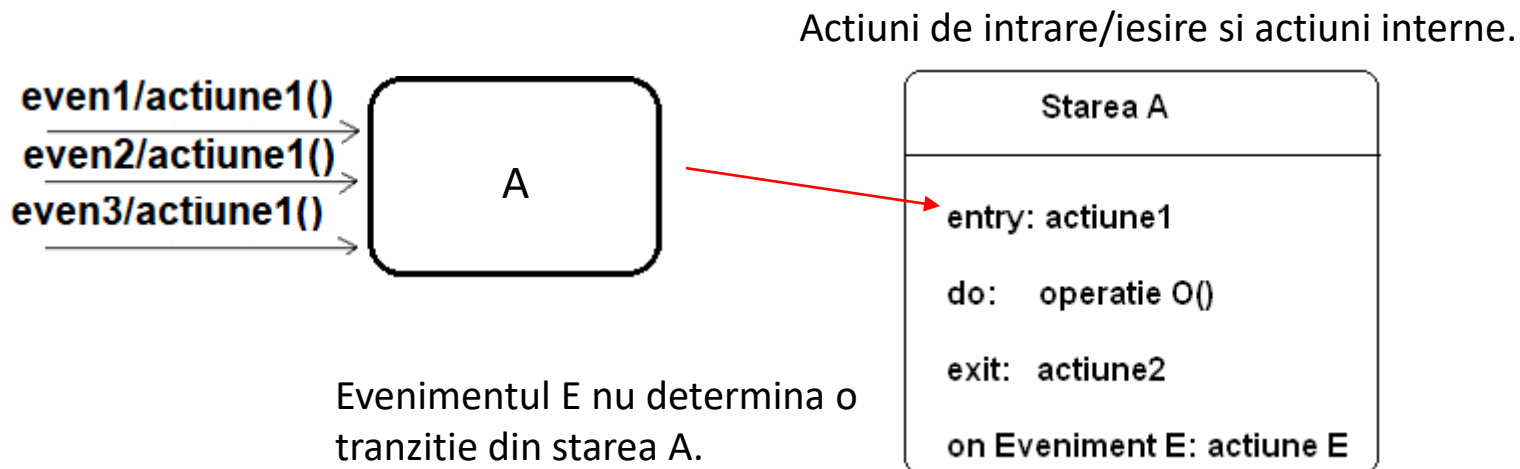


In starea A se execută operația P

- O **activitate** poate fi **întreruptă oricând**, atunci cand are loc un eveniment care determina o tranzitie de iesire din starea respectiva.

Actiuni si activitati(3)

- Mai multe evenimente pot determina tranzitia in aceeaasi stare.
- Fiecare eveniment poate declansa o anumita actiune.
- Atunci cand toate evenimentele care conduc in aceeaasi stare declanseaza aceeaasi actiune, actiunea poate fi modelata ca **actiune de intrare** in starea respectiva.
- Iesirea dintr-o stare poate fi determinata de mai multe evenimente.
- Atunci cand toate evenimentele care declanseaza tranzitii din aceeaasi stare specifica o aceeaasi actiune, actiunea poate fi modelata ca o **actiune de iesire** din starea respectiva.

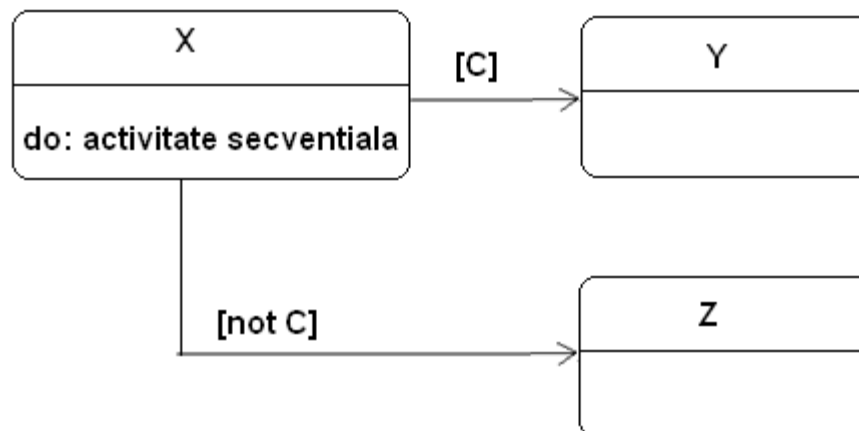


Tranzitii automate

- La terminarea unei activitati secventiale are loc o tranzitie automata din starea în care s-a executat.
- O tranzitie care nu este marcata printr-un eveniment este numita **tranzitie automata**.



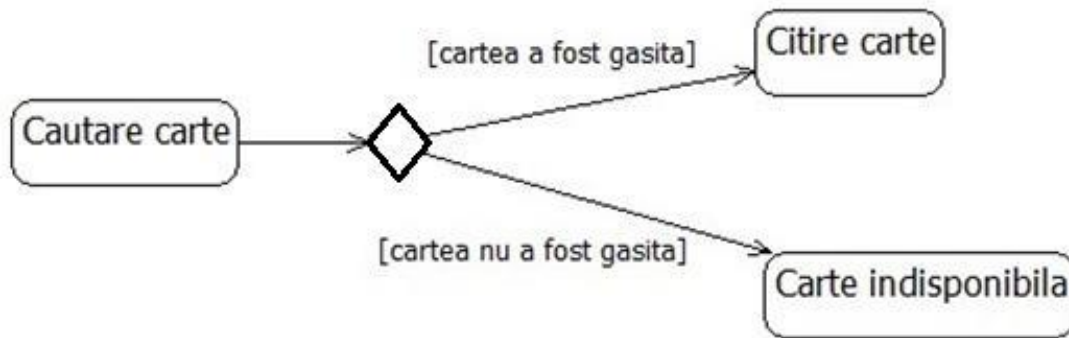
- Tranzitiile automate pot fi controlate prin garzi:



Pseudostări (1)

Alegerea dinamică (Choice)

- ❖ Pseudostare în care se efectuează o ramificare condiționată dinamică.



Stare în care utilizatorul poate primi un mesaj sau i se dă opțiuni de continuare, ca de ex. căutarea unei alte cărți.

- Se evaluează gărzile asociate tranzițiilor de ieșire din pseudostare și se execută tranziția a cărei gardă este adevărată.
- Cel puțin una dintre gărzi trebuie să fie adevărată.
- Dacă mai mult de o gardă este adevărată, se alege pentru execuție, în mod arbitrar, una dintre tranziții.

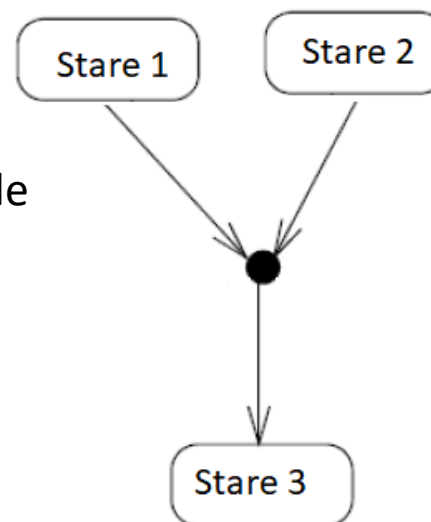
Pseudostări (2)



Toate garzile sunt expresii binare cu același operand în stânga.

Joncțiune (Junction)

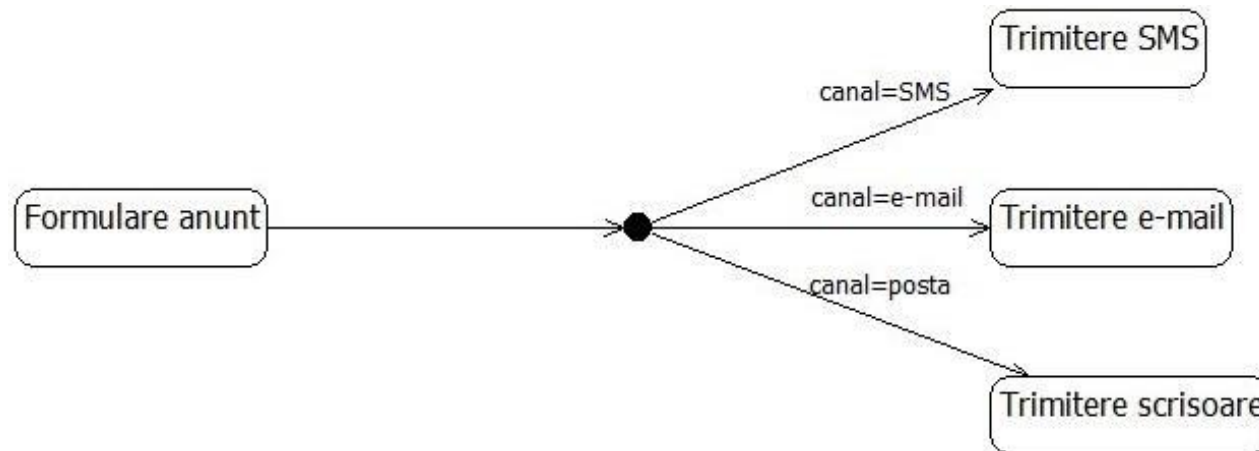
- Se poate folosi pentru a combina mai multe tranzitii de intrare într-o singură tranziție de iesire.



Pseudostări (3)

Joncțiune –ramificare condiționată statică

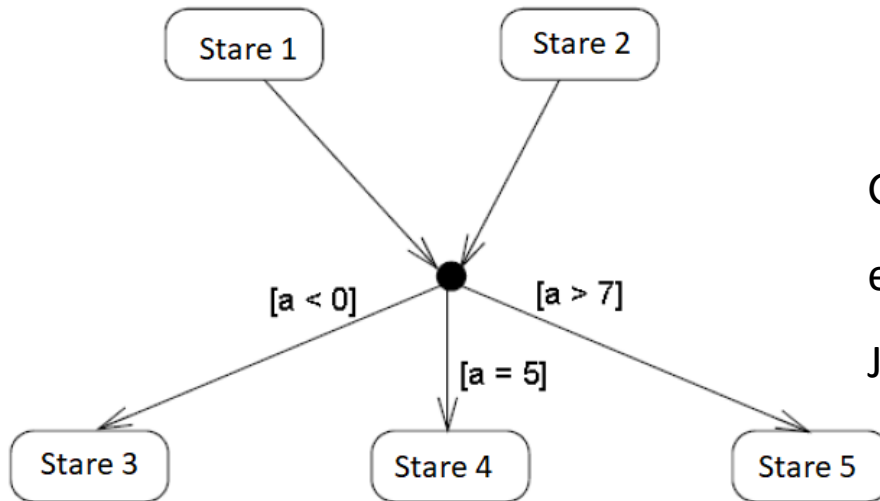
- Pseudostarea se poate folosi pentru a ramifica o tranziție de intrare în mai multe tranziții de ieșire cu gărzi (constrangeri) asociate diferite.



- Garzile asociate tranzițiilor de ieșire sunt evaluate înainte de intrarea în pseudostarea
Joncțiune: pseudostarea mai este numită “ramificare condiționată statică”

Pseudostări (4)

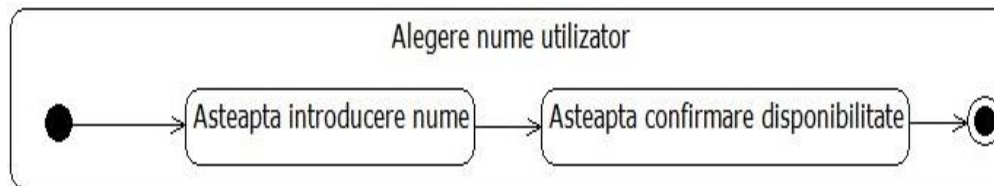
Joncțiune – continuare



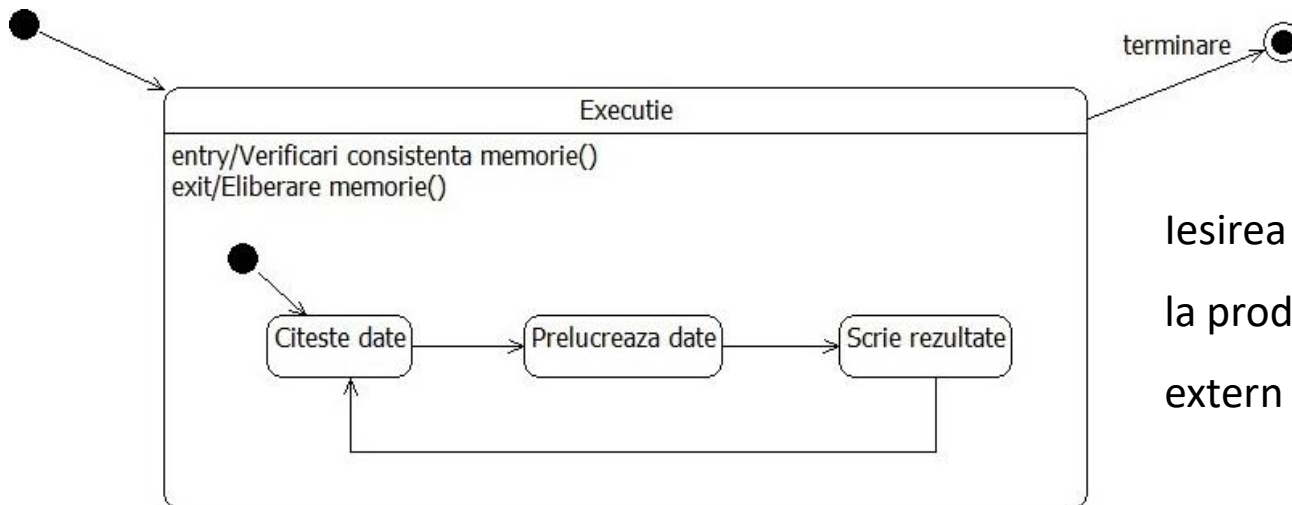
Garzile asociate tranzițiilor de ieseire sunt evaluate înainte de intrarea în pseudostarea Joncțiune

Stări compuse (1)

- O stare care conține substări se numește *stare compusă*. O stare care nu conține substări se numește *stare simplă*.
- Substările unei stări compuse pot fi secvențiale (disjuncte) sau concurente (ortogonale).

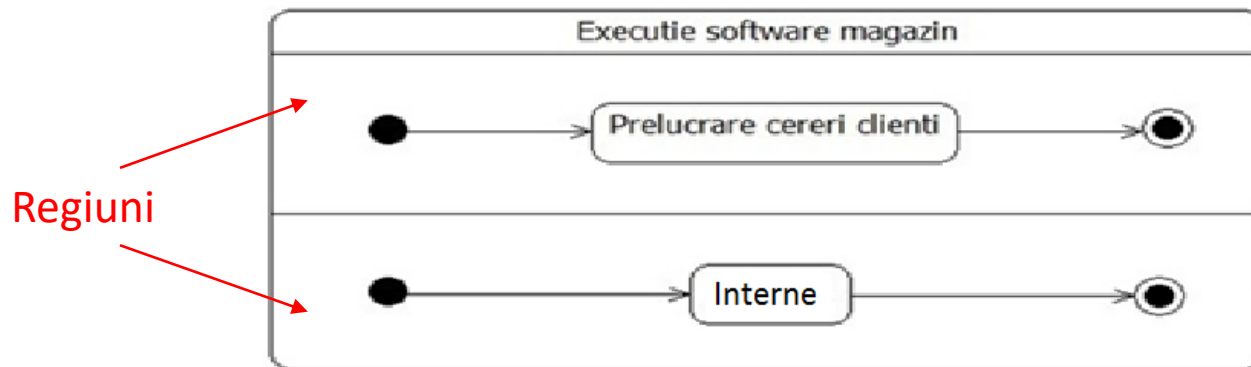


Stare compusa cu 2
substari secventiale



Iesirea din starea Executie are loc
la producerea unui eveniment
extern (terminare)

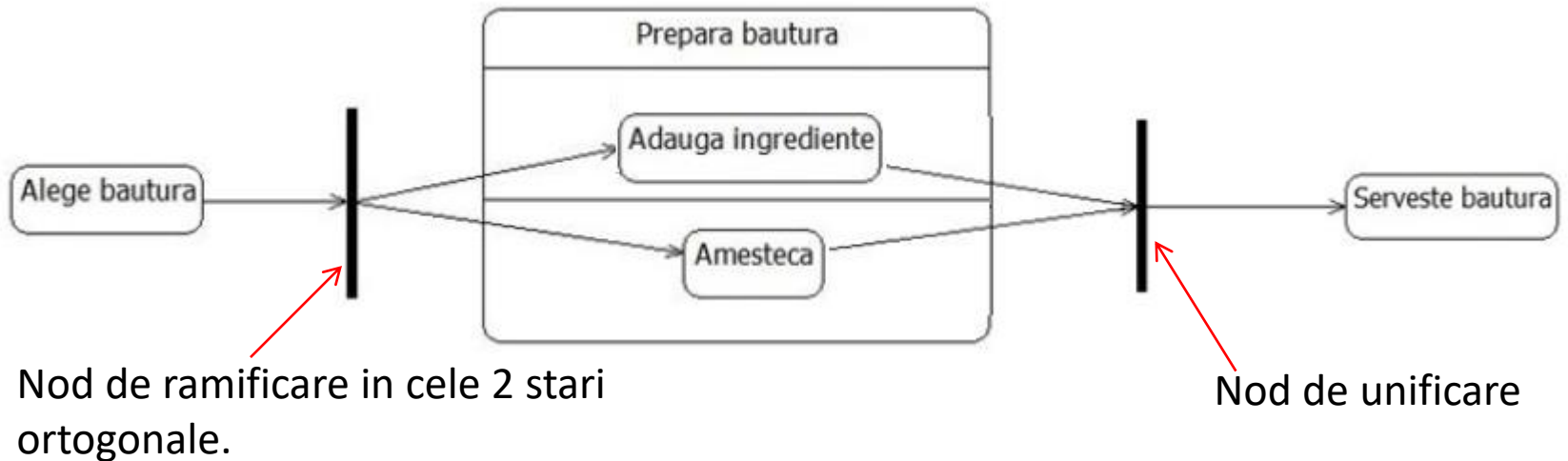
Stari compuse (2)



Stare compusa cu 2 substari concurente(ortogonale).

- Fiecare regiune contine un set de stari si tranzitii care se excuta secvential.
- Regiunile unei stări compuse ortogonale se execută în paralel.
- Regiunile descriu stari si tranzitii ale unor fire de executie paralele.

Stari compuse (3)



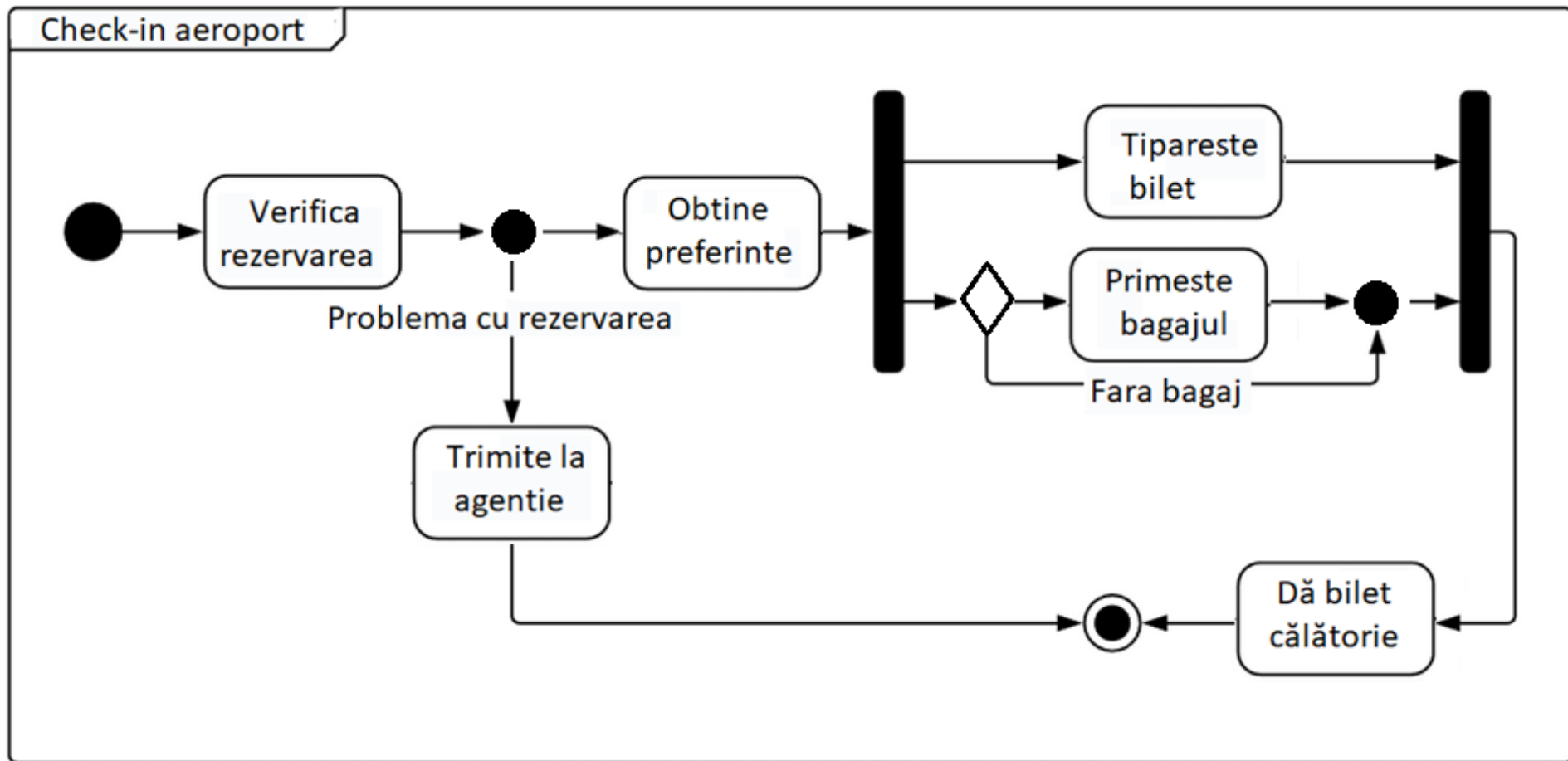
Ramificarea (Fork)

- ❖ Pseudostare care descompune tranziția de intrare în două sau mai multe tranziții de ieșire în stări ortogonale ale unei stări compuse.
- Tranzițiile de ieșire dintr-o pseudostare de tip *ramificare* nu pot avea asociate gărzi sau evenimente.

Unificarea (Join)

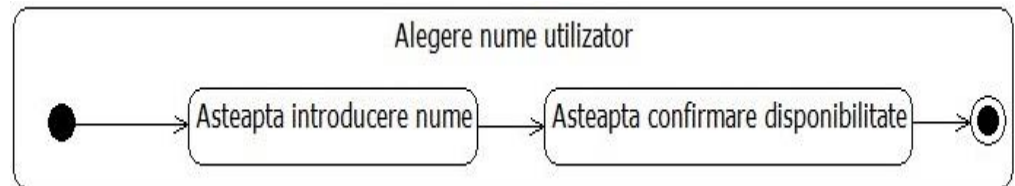
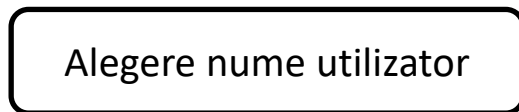
- ❖ Pseudostare care unifica tranziții din stări ortogonale.
- Tranzițiile de intrare într-o pseudostare Join nu pot avea asociate garzi sau evenimente.

Stari compuse si pseudostari - exemplu



Reprezentare ierarhica

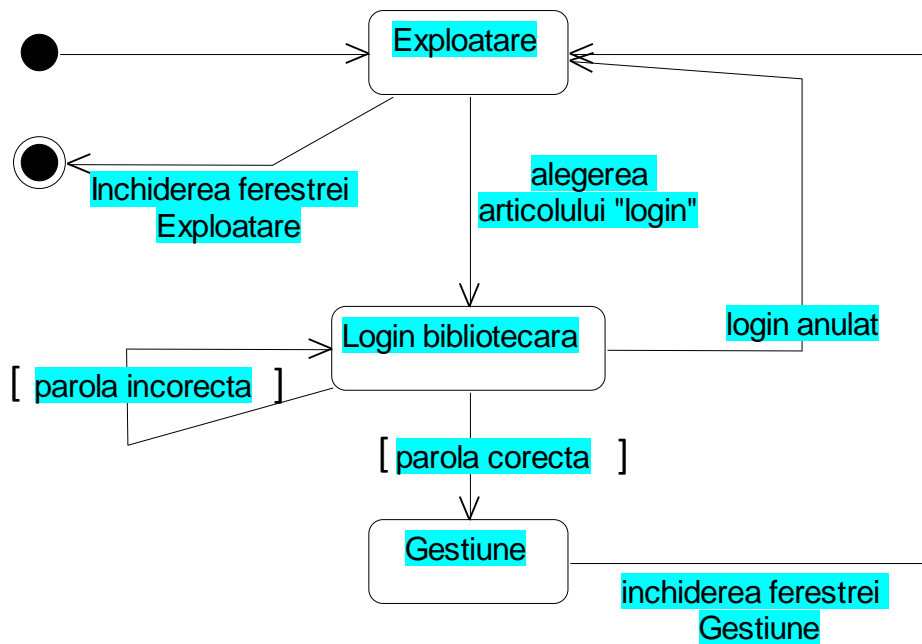
- Mai multe stari pot fi abstractizate intr-o singura stare, care corespunde unei reprezentari de nivel ierarhic mai inalt.
- O stare compusa corespunde unui nivel de abstractizare mai coborat.
- De exemplu, cele 2 reprezentari corespund la 2 niveluri de abstractizare diferite



- Pentru fiecare nivel de abstractizare exista o singura stare initiala.
- Este posibil sa existe mai multe stari finale, fiecare corespunzand unei conditii de sfarsit diferite.

Concluzii

- Diagramele de stari se folosesc pentru **modelarea comportamentala** a **entităților reactive** → pot fi identificate analizand diagramele de interactiune.
- Permit o documentare mai buna a claselor complexe.
- Pot fi folosite în alegerea cazurilor de test.
- **Pot fi folosite si pentru redarea comportamentului in timp al unui sistem** sau pentru modelarea comportamentala a interfetei utilizator. Exemplu:



Lecturi suplimentare

1. Alin Moldoveanu, Florica Moldoveanu, Maria-Iuliana Dascălu, Anca Ioniță, Oana-Maria Ferche, Victor Asavei, Anca Morar, UML practic, Ed. MatrixRom, 2014
2. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/>
3. <https://warren2lynch.medium.com/state-diagram-comprehensive-guide-with-examples-e08b6d1c70fe>
4. <https://sparxsystems.com/resources/tutorials/uml2/state-diagram.html>
5. <https://www.lucidchart.com/pages/uml-state-machine-diagram>