Laborator 5

Diagrame de Interacțiuni - Diagrama de secvență

Diagramele de interacțiuni sunt folosite pentru a modela comportamentul unei mulțimi de obiecte dintr-un anumit *context* care interacționează în vederea îndeplinirii unui anumit *scop*.

Scopul specifică modul în care se realizează o operație sau un caz de utilizare.

Contextul unei interacțiuni (unde pot găsi interacțiuni) poate fi:

- *sistem* / *un subsistem* (uzual) - mulțimea obiectelor din sistem care colaborează între ele;
- *operație* - interacțiuni între parametri, variabile locale şi globale;
- *clasă* - interacțiuni între atributele unei clase (cum colaborează ele), interacțiuni cu obiecte globale, sau cu parametrii unei operații.

Obiectele care participă la o interacțiune pot fi lucruri concrete sau prototipuri. De obicei, într-o colaborare obiectele reprezintă prototipuri ce joacă diferite roluri, şi nu obiecte specifice din lumea reală.

Între obiectele care participă la o colaborare se pot stabili legături.

O ***legătură*** (link) reprezintă o conexiune semantică între obiecte. Ea este o instanță a unei asocieri şi poate avea toate atributele specifice asocierii (nume, roluri, navigare, agregare), dar nu şi multiplicitate.

Obiectele care interacționează comunică între ele, comunicarea făcându-se prin schimb de mesaje.

Un ***mesaj*** specifică o comunicare între obiecte. El poartă o informație şi este urmat de o activitate. Primirea unei instanțe a unui mesaj poate fi considerată o instanță a unui eveniment.

Unui mesaj îi este asociată o *acțiune* care poate avea ca efect schimbarea stării actuale a obiectului.

Forma generală a unui mesaj este

```
[cond_gardă] acțiune (lista_parametrilor)
```

unde

condiție_gardă – condiție booleană care se evaluează la fiecare apariție a mesajului specificat; acțiunea se execută doar când rezultatul evaluării este `true;`

Tipuri de acțiuni existente în UML:

- ***call***: invocă o operație a unui obiect. Un obiect îşi poate trimite lui însuşi un mesaj (invocare locală a unei operații). Este cel mai comun tip de mesaj. Operația apelată trebuie să fie definită de obiectul apelat şi vizibilă apelantului.
- ***return***: returnează o valoare apelantului.
- ***send***: trimite un semnal unui obiect.
- ***create***: creează un obiect.
- ***destroy***: distruge un obiect. Un obiect se poate autodistruge.

O diagramă de interacțiuni constă dintr-o mulțime de obiecte şi relațiile dintre ele (inclusiv mesajele care se transmit). Există două tipuri de diagrame de interacțiuni:

- *diagrama de secvență;*
- *diagrama de colaborare*.

Cele două diagrame specifică aceeaşi informație, însă pun accentul pe aspecte diferite.

Diagrama de secvență

Diagrama de secvență pune accentul pe aspectul temporal (ordonarea în timp a mesajelor).

Notația grafică este un ***tabel*** (figurile 1, 2) care are pe axa X *obiecte*, iar pe axa Y *mesaje* ordonate crescător în timp.

Axa Y arată pentru fiecare obiect:

- linia vieții - linie punctată verticală;
- perioada în care obiectul preia controlul execuției - reprezentată printr-un dreptunghi subțire pe linia vieții; în această perioadă obiectul efectuează o acțiune, direct sau prin intermediul procedurilor subordonate.

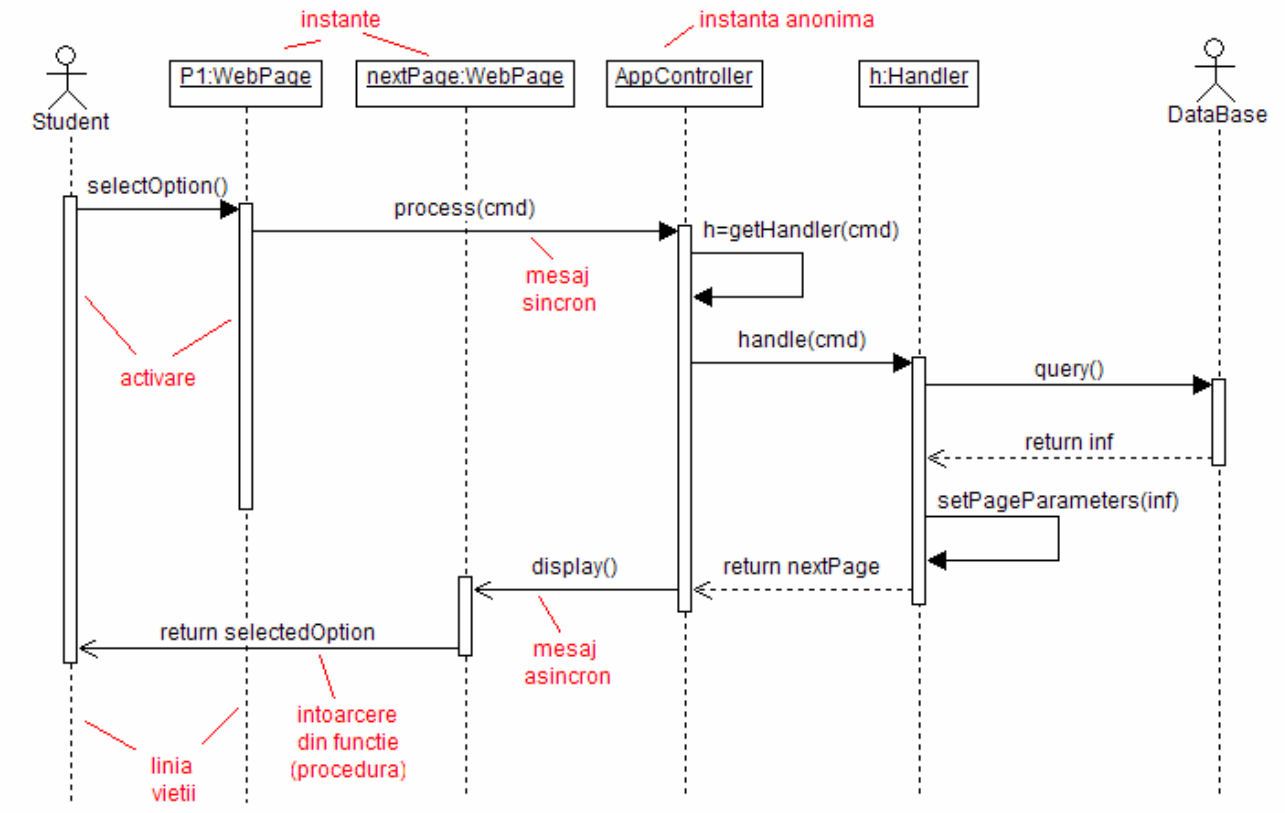

Figura 1. Exemplu de diagramă de secvență

Notația grafică pentru mesaje

A ⟶ **B** **Comunicare sincronă.** Controlul execuției trece de la A la B şi revine la A după ce B îşi termină execuția. *Exemplu*: apel de funcție.

A ⟶ B **Comunicare asincronă**. A trimite un semnal după care își continuă execuția mai departe. *Exemplu*: aruncarea unei excepții.

⟵ **Întoarcere dintr-o funcție** (**procedură**). În cazul în care este omisă se consideră implicit că revenirea se face la sfârșitul activării.

În figura 2 este prezentat un exemplu de diagramă de secvență. Mesajul `extrageNume()` este primul mesaj recepționat de `Client` și corespunde cererii `Managerului de Campanie` de a furniza numele clientului selectat. Obiectul Client recepționează apoi mesajul `listeazăcampanii()` și începe a doua perioadă de activare. Oiectul `Client` trimite apoi mesajul `extrageDetaliiCampanie()` fiecărui obiect `Campanie` pe rând pentru a construi o listă a campaniilor. Această operațiune repetată se numește ***iterație*** și se marchează prin caracterul „*” înaintea numelui mesajului. Condiția de continuare sau de încetare poate fi precizată în interiorul numelui mesajului. Condiția de continuare poate fi scrisă sub forma:

```
[pentru toți clienții campaniilor] *extrageDetaliiCampanie()
```

`Manager Campanie` trimite apoi un mesaj unui obiect particular `Campanie` pentru a obține o listă a reclamelor. Obiectul `Campanie` deleagă responsabilitatea pentru a extrage titlul reclamei fiecărui obiect `Reclamă` deși obiectul `Campanie` păstrează responsabilitatea pentru lista reclamelor (fapt indicat de păstrarea activării și după ce este trimis mesajul).

Când o reclamă este adăugată la campanie este creat un obiect `Reclama`. Această creare este indicată de mesajul `Reclama()` (care invocă un constructor).

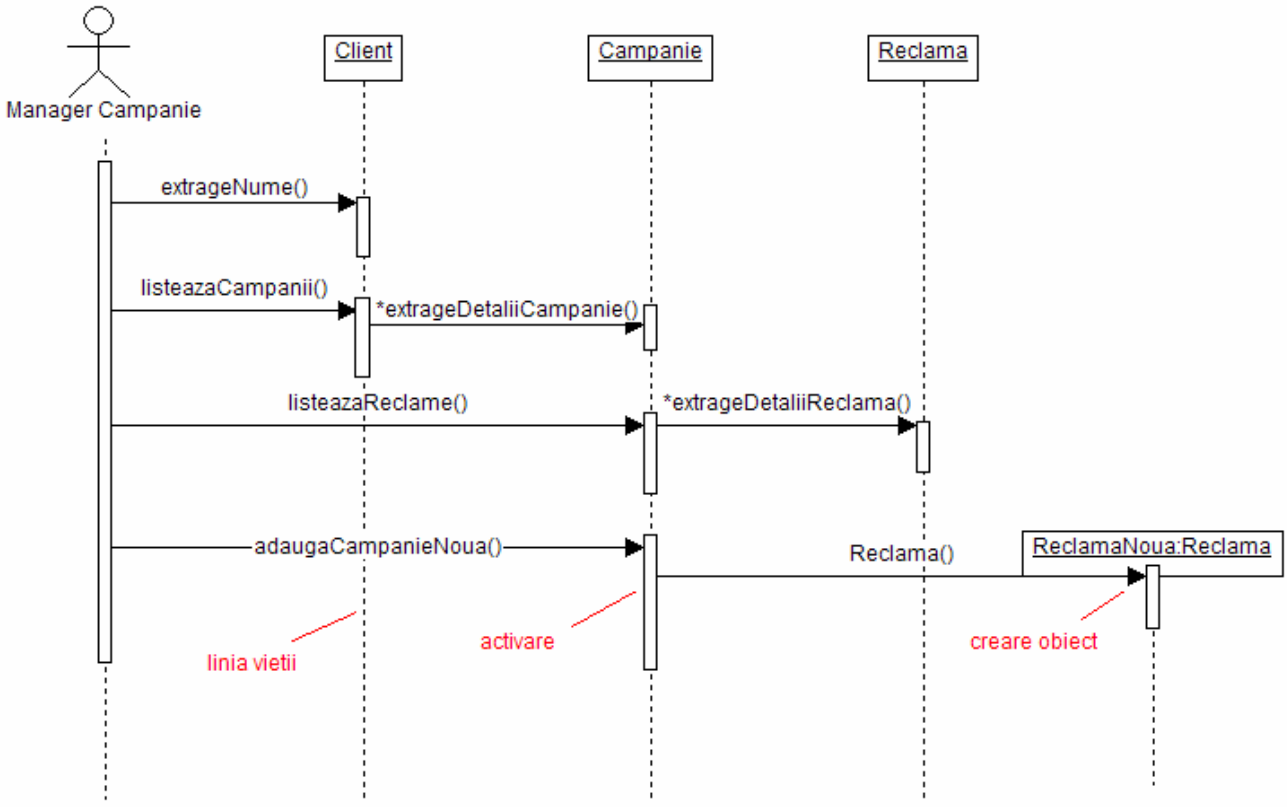

Figura 2. exemplu de diagramă de secvență care modelează adăugarea unei noi reclame unei campanii

Obiectele pot fi create sau distruse la diferite momente ale interacțiunii. Distrugerea unui obiect este marcată printr-un "X" pe linia vietii. Un obiect poate fi distrus când primeşte un mesaj (ca în figura 3) sau se poate distruge singur la sfârşitul unei activări.

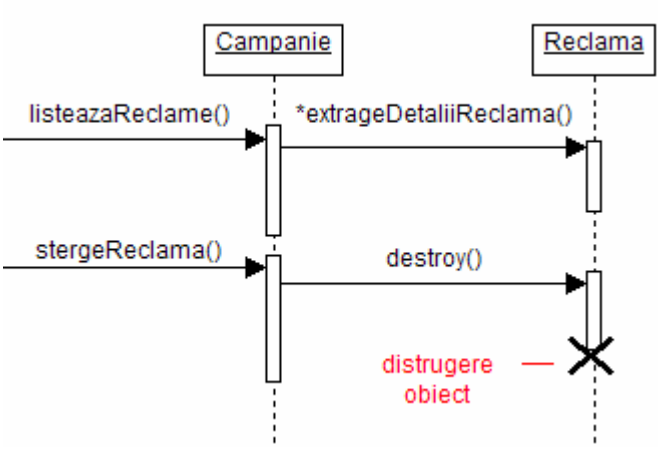

Figura 3. Distrugerea unui obiect

Un obiect îşi poate trimite un mesaj lui însuşi. Acest mesaj este cunoscut sub numele de mesaj reflexiv şi este reprezentat de o săgeată care pleacă şi se termină pe o activare a aceluiaşi obiect. Diagram de secvență din figura 4 include mesajul reflexiv `calculeazăCheltuieliregie`() trimis de obiectul `Campanie` lui însuşi.

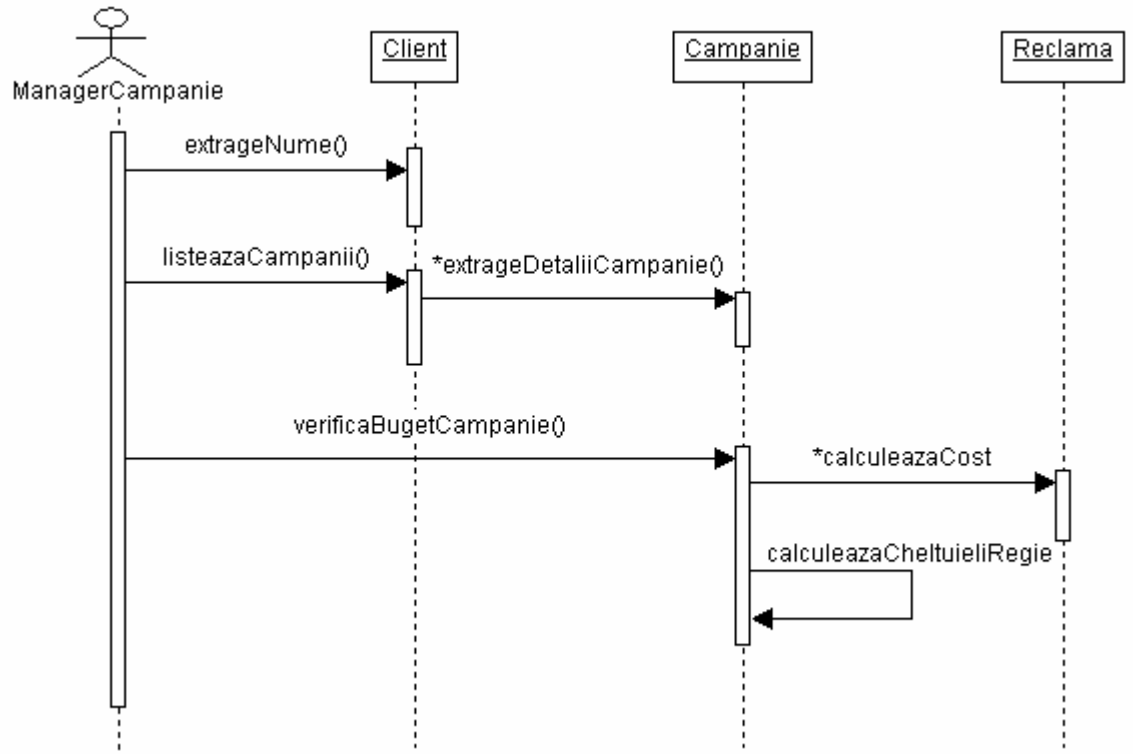

Figura 4. Diagramă de secvență care modelează `Verifică bugetul campaniei`

După cum am mai precizat, revenirea controlului la obiectul care a trimis un mesaj se poate marca explicit în diagrama de secvență printr-o săgeată trasată cu linie întreruptă (figura 5). Valoarea de revenire de obicei nu se prezintă într-o diagramă de secvență.

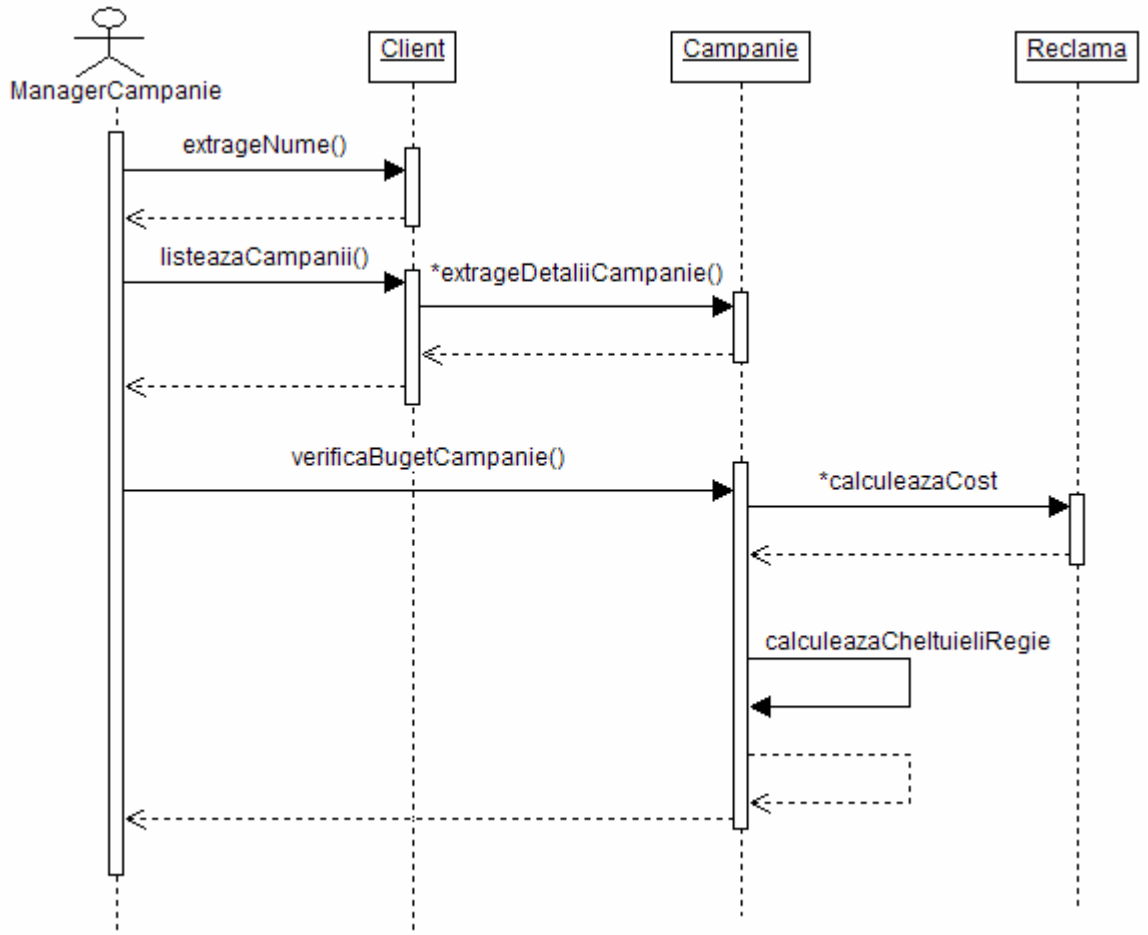

Figura 5. Diagramă de secvență care modelează `Verifică bugetul campaniei` cu marcarea explicită a revenirilor

Dacă *mesajele sincrone* (care invocă o operație) determină suspendarea execuției obiectului sursă până când destinatarul își termină execuția, *mesajele asincrone* nu necesită suspendarea execuției obiectului ce trimite mesajul. Mesajele asincrone sunt des folosite în aplicațiile de timp real în care operațiile diferitelor obiecte se execută în paralel, fie din motive de eficiență, fie deoarece sistemul simulează procese concurente. Este necesar ca o operație invocată să notifice obiectul care a invocat-o în momentul când își termină execuția. Această notificare se face printr-un mesaj explicit (numit *callback*).

Constrângeri de timp

O diagramă de secvență poate fi etichetată cu constrângeri de timp în moduri diferite. În figura 6 se asociază expresiile de timp cu numele mesajului astfel încât constrângerile de timp pot fi specificate pentru execuția unei operații sau transmisia unui mesaj. Spre exemplu funcția `a.sendTime` furnizează timpul la care mesajul `a` este trimis şi `d.receiveTime` furnizează timpul la care o instanță a clasei A primeşte mesajul `d`. Există construcții care pot fi utilizate pentru

a marca un interval de timp – în figura 6 este marcat intervalul de timp scurs între recepția mesajului `b` şi trimiterea mesajului `c`. Constrângerile de timp sunt utilizate frecvent în modelarea sistemelor de timp real. Pentru alte tipuri de sisteme constrângerile de timp nu sunt semnificative.

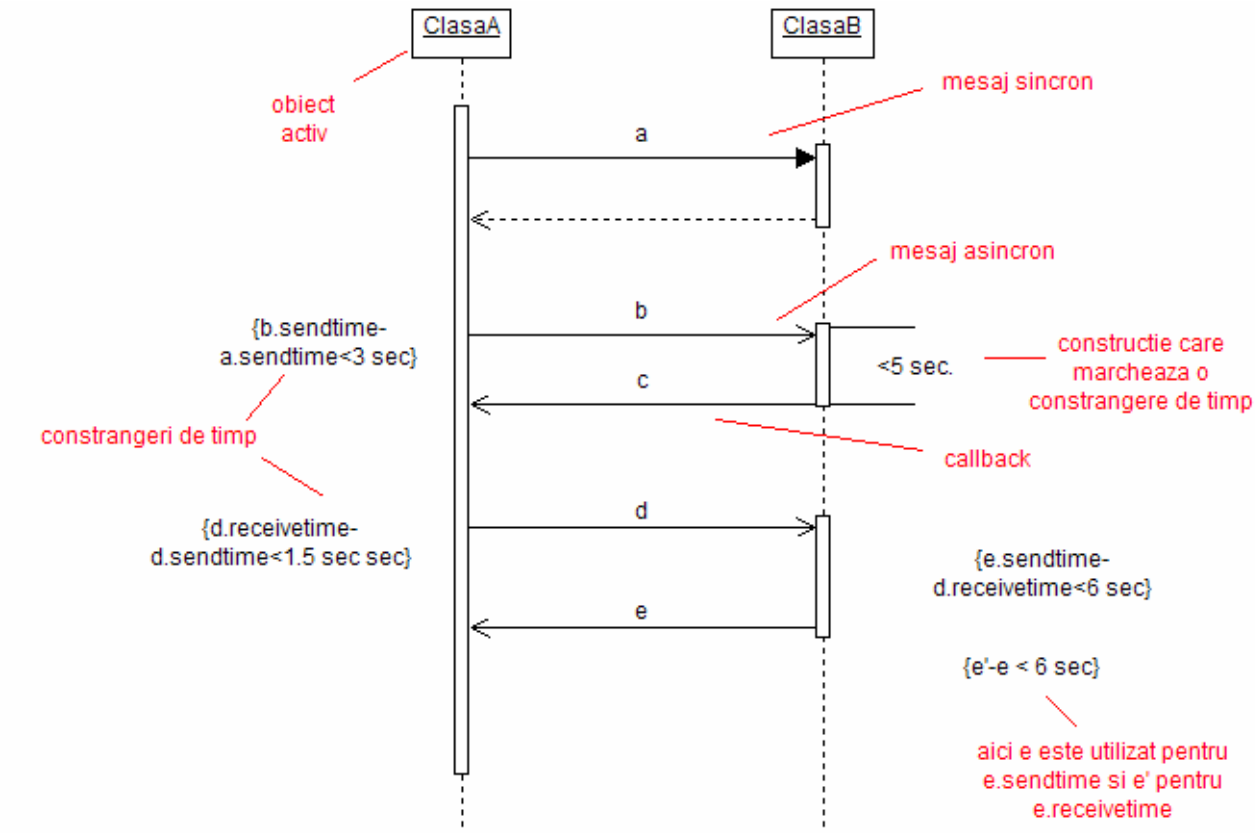

Figura 6. Diagramă de secvență cu tipuri diferite de mesaje şi constrângeri de timp

Ramificații

Diagramele de secvență permit reprezentările ramificațiilor prin mai multe săgeți care pleacă din acelaşi punct şi eventual sunt etichetate cu condiții.

În figura 7 este prezentat un exemplu de diagramă de secvență cu ramificații.

Probleme propuse

Pentru fiecare din problemele de mai jos să se realizeze diagramele de secvență:

1. Automat cafea (alegere tip cafea, introducere moneda, eliberare rest, preluare produs, etc)
2. ATM (verificare PIN, vizualizare suma din contul personal, extragere, tipărire chitanța etc.)
3. Ceas electronic (afişare ora curentă / data curentă, modificare oră / dată, cronometru etc.)

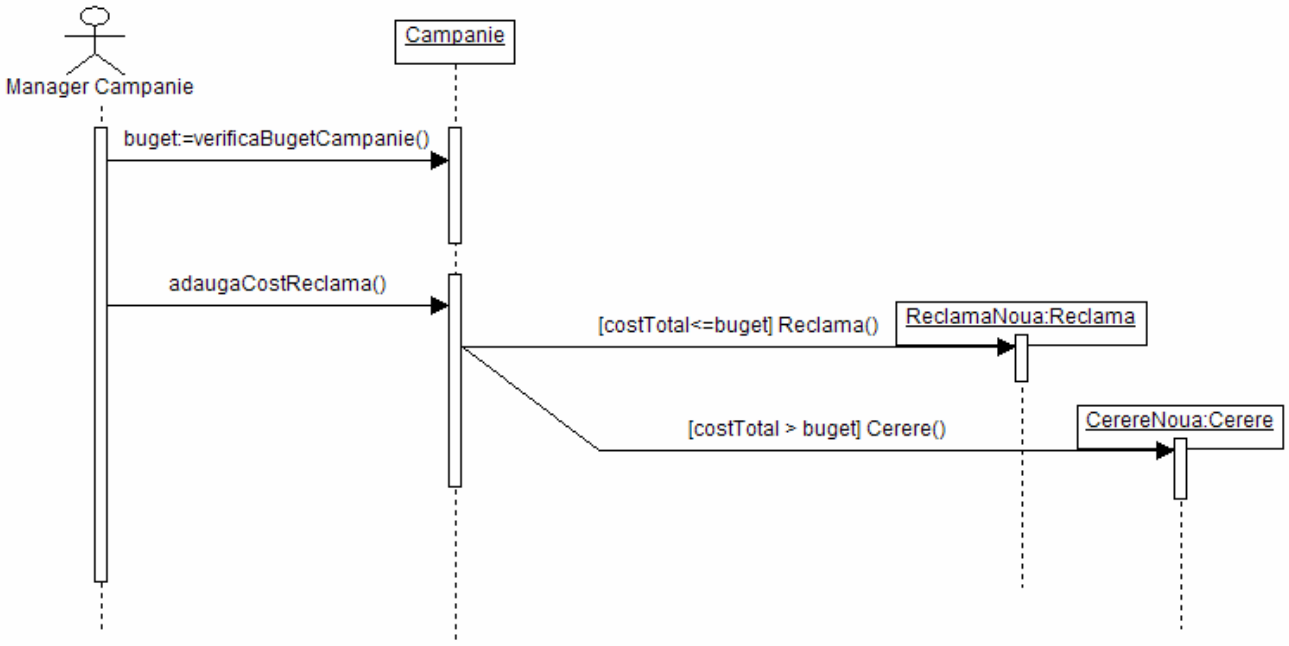

Figura 7. Exemplu de diagramă de secvențe cu ramificații.