

Proiectarea obiectuala (detaliata). Modele dinamice ale unui sistem
Diagrame UML de secventa

Biblio:

1. Martina Seidl et al., *UML@Classroom* – pp.107-139 (atasat)

Într-un sistem orientat-obiect, obiectele comunica între ele prin schimb de mesaje. Un *mesaj* transmis unui *obiect* are ca și efect apelul unei *metode* din clasa obiectului ținta, ceea ce determină, în general, la rândul său, transmiterea de noi mesaje către obiectele din sistem.

Diagramele UML de secvență fac parte din categoria *diagramelor de interacțiune* (aici intra și *diagramele de comunicare*, abordate în seminarul următor). Diagramele de interacțiune surprind comportamentul dinamic al sistemului, prin prisma interacțiunilor dintre obiectele care îl compun, cu scopul oferirii funcționalității cerute.

Cele două tipuri de diagrame de interacțiune menționate anterior sunt echivalente; informația redata de acestea este aceeași, doar perspectiva diferă (la nivelul instrumentelor CASE, există posibilitatea generării automate a unui tip de diagramă din celălalt). În diagrama de secvență primează perspectiva temporală (secvențierea, în timp, a mesajelor transmise în cadrul interacțiunii), în timp ce, la nivelul diagramei de comunicare, accentul cade pe evidențierea comunicării/legăturilor dintre obiectele participante. Alegerea unui anumit tip de diagramă depinde de tipul interacțiunii: dacă numărul de obiecte participante este mare, dar numărul de mesaje transmise e mic, se recomandă utilizarea unei diagrame de comunicare (cea de secvență s-ar extinde mult pe orizontală, puținele mesaje transmise fiind greu de urmărit); dacă numărul de mesaje e mare și numărul de obiecte participante mai mic, se optează pentru o diagramă de secvență.

În etapa de proiectare, o diagramă de interacțiune se utilizează, de regulă, pentru a reprezenta un scenariu asociat unui caz de utilizare (= o interacțiune concretă, în care se cunoaște numărul de pași dintr-un ciclu sau alternativă aleasă). Ea poate fi utilizată însă și pentru reprezentarea unui caz de utilizare în ansamblu (există mecanisme de descriere a alternativelor și ciclurilor) sau a comportamentului asociat unei operații.

(Obs.: Pentru laborator, se cere câte o diagramă de secvență pentru fiecare dintre scenariile normale ale cazurilor de utilizare descrise la faza 1.)

Exemplu (atasat):

- Diagrama de secvență aferentă scenariului normal al cazului de utilizare *Login* al sistemului de gestiune a comenzilor din cadrul unui restaurant (*Restaurant System* - vezi Seminar 2), corespunzând logării cu succes a unui *Waiter* în sistem

- Fragmentul din diagrama de clase aferentă modelului structural de proiectare, incluzând clasele obiectelor care participă la interacțiunea descrisă prin diagrama de secvență de la punctul precedent și relațiile stabilite între acestea (asocieri și dependente).

Reguli privind realizarea diagramei de secvență:

1. Fiecarui obiect participant la interacțiune îi corespunde care o *linie de viață (lifeline)* la nivelul diagramei, reprezentată în plan vertical.

2. Timpul curge „de sus în jos”: dacă un mesaj e reprezentat deasupra altuia în diagramă, înseamnă că el a fost transmis, în timp, înaintea celui de-al doilea.

3. Dacă obiectul exista deja la începutul interacțiunii, caseta aferenta va apărea în partea de sus a diagramei, altfel, va apărea la nivelul corespunzător momentului instantierii sale (nu e o regula obligatorie, în scris se pot reprezenta toate casetele în partea de sus, dacă acest lucru mărește inteligibilitatea diagramei; momentul crearii e oricum evident, prin transmiterea unui mesaj de tip <<create>> către obiect).

4. Mesajele se reprezintă ca și săgeți etichetate cu numele metodelor/operatiilor cărora le corespund + argumente. Exista notatii diferite pentru diferite tipuri de mesaje (sincron/asincron, create/distroy, retur etc.).

5. Mesajele pot fi numerotate (în stil imbricat, de preferinta), însă nu e obligatoriu; perspectiva temporala e implicita într-o diagrama de secvență. Numerotarea imbricata este însă foarte utila în momentul generarii diagramei de comunicare echivalente (pe diagrama de comunicare lipseste perspectiva temporala, numerotarea imbricata e obligatorie pentru a asigura echivalenta celor doua).

6. Cand un mesaj atinge linia de viața a obiectului ținta, începe așa-numitul „dreptunghi de activare”, care corespunde intervalului de timp în care se executa metoda aferenta mesajului. Dacă în cadrul acelei metode se trimit mesaje către alte obiecte, acestea vor avea originea în acel dreptunghi de activare și, în numerotarea imbricata, se numereaza relativ la mesajul initial (dacă mesajul initial e numerotat 2.5 și, din dreptunghiul sau de activare, pleacă alte 3 mesaje, acestea vor fi numerotate 2.5.1, 2.5.2 si 2.5.3).

7. ! Atenție la dimensiunile dreptunghiurilor de activare, revenirea dintr-un apel (finalul dreptunghiului de activare aferent) ar trebui să fie înaintea revenirii din apelul din cadrul caruia a fost efectuat.

Reguli privind corespondenta dintre cele doua diagrame:

1. Fiecarui *obiect* care participa la interacțiunea descrisa prin diagrama de secvență (are un *lifeline* asociat) ii corespunde o *clasa* la nivelul diagramei de clase.

2. Fiecarui *mesaj* reprezentat la nivelul diagramei de secvență ii corespunde, la nivelul diagramei de clase, o *operație/metoda în clasa obiectului ținta*.

3. În cadrul diagramei de clase, reprezentarea relatiilor intre acestea are la baza urmatoarele reguli:

a. Dacă un obiect *o1* (instanța a clasei *C1*) trimite un mesaj unui alt obiect *o2* (instanța a clasei *C2*), atunci *o1* are o referinta spre *o2*; poate fi vorba de o relație de *asociere* intre clasele corespuzatoare sau doar de o *dependentă* (*asocierea* se traduce prin date membru/atribute de timpul respectiv, *dependentă* prin utilizare de parametri, variabile locale; asocierea este „mai puternica” decât dependentă).

b. Dacă cele doua obiecte mai comunica și în afara apelului de metoda în cadrul caruia a fost transmis mesajul în cauza, e necesara o asociere intre clasele aferente; altfel, e suficienta o dependentă. Navigabilitatea asocierii e dictata de sesnsul în care se transmit mesajele intre cele doua obiecte: dacă mesajele sunt transmise într-un singur sens (de la *o1* către *o2*, spre *exemplu*), e suficienta doar o asociere unidirectionala (navigabila dinspre *C1* înspre *C2*), altfel, e necesara o asociere bidirectionala (navigabila în ambele sensuri)

! Atenție, informațiile privind comunicarea dintre obiecte se extrag analizand interactiunile la nivelul sistemului în ansamblu (prin consultarea tuturor diagramelor de secvență

realizate); o singura diagrama poate dicta doar necesitatea unei simple dependente, în timp ce analiza mai multora poate să impună o asociere.

c. Numele de rol utilizate în diagrama de secvență corespund celor din diagrama de clase.

d. Particularizarea asocierilor ca și agregari/compuneri se va face pe baza semanticii cunoscute a celor din urmă.

Observatii:

1. Exemplul prezentat are la bază o arhitectură în care obiectele *control* corespund cazurilor de utilizare; un obiect *control* e creat de obiectul *boundary* care inițiază cazul de utilizare, urmând ca, ulterior, acesta să fie responsabil de crearea de noi obiecte *boundary* (vezi curs).

2. ! Pentru activitatea de laborator, diagramele de secvență realizate vor reflecta arhitectura aleasă pentru dezvoltarea sistemului, cu sabloanele de comunicare aferente; exemplul de seminar are doar caracter orientativ, pentru a înțelege regulile pe baza cărora (în etapa de proiectare detaliată) se elaborează diagramele de interacțiune și se rafinează diagrama de clase aferentă modelului conceptual (corespunzătoare etapei de analiză).

3. În evaluarea activității de laborator, se va urmări sincronizarea dintre interacțiunile descrise prin diagrame de secvență în etapa de proiectare (obiecte participante, schimb de mesaje) și codul aferent. Ca urmare, în cazul în care, în etapa de implementare, se decide schimbarea unora dintre deciziile luate în proiectare, se va reveni asupra diagramelor de interacțiune, cu modificarea acestora, pentru a reflecta aceste schimbări (codul și modelele UML trebuie pastrate sincronizate!).

4. În proiectul UML de laborator, NU se va realiza câte o diagramă de clase pentru fiecare diagramă de secvență descrisă, ci informațiile colectate din toate diagramele de secvență se vor integra la nivelul unui singur model structural de proiectare. În cazul în care dimensiunea diagramei de clase aferente e mare, cu efecte negative asupra inteligibilității acesteia, se vor crea view-uri/diagrame diferite, segmentând astfel informația (ex.: o diagramă de clase incluzând toate clasele și relațiile dintre acestea, adnotate corespunzător, dar fără detalii privind atributele/operatiile claselor + diagrame separate ilustrând structura claselor SAU partitionarea modelului structural în mai multe diagrame de clase, fiecare acoperind o parte a sa, cu toate atributele/relațiile/operatiile corespunzătoare).

Relativ la realizarea diagramelor aferente fazei de proiectare cu ajutorul instrumentului StarUML (se consideră o structură a proiectului conforma template-ului UML Conventional):

- tot ceea ce ține de proiectare (model structural - diagramă de clase rafinată + model dinamic - diagrame de secvență și comunicare) trebuie inclus în proiect în partea de Design Model (anterior, modelul funcțional a fost inclus la secțiunea UseCase Model, iar cel conceptual la secțiunea Analysis Model).

- trebuie lucrat în paralel pe partea de model structural și model dinamic, după cum urmează:

a. Se dorește crearea unei diagrame de secvență și e nevoie de un lifeline care să reprezinte un actor => se adaugă un lifeline pe diagramă și lui i se va asocia un rol – Role1, de ex. Se accesează Role1 - acesta are ca și proprietate un type. Pentru acel type, se navighează în model și se selectează actorul din modelul funcțional (Use Case Model) de care e nevoie. După care, dacă acesta nu e reprezentat implicit ca și actor, click dreapta > Format > Stereotype Display > Icon.

b. E nevoie de un lifeline care sa reprezinte un obiect din aplicatie => se adauga pe diagrama acel lifeline, lui i se asociaza un rol - Role2. Se adauga mai intai pe diagrama de clase aferenta modelului structural de proiectare clasa a carui instanta e respectivul obiect. Dupa care se seteaza atributul type al lui Role2 la acea clasa (selectata din Design Model). Se adauga din nou si clasele din modelul conceptual, fiindca va fi nevoie ca acestea să fie rafinate aici (prin adaugare de operatii, attribute, relatii, etc.).

c. Numele de roluri setate pentru lifeline-urile din diagrama de secvența vor corespunde numelor de rol folosite în adnotarea relațiilor de asociere din diagrama de clase aferenta modelului structural de proiectare. Ele pot fi omise, dacă nu sunt relevante (pentru actori, de ex.).

d. E nevoie de un mesaj intre doua lifeline-uri => se adauga obiectul mesaj din toolbar pe diagrama, dupa care acesta trebuie etichetat cu un nume de metoda. Se acceseaza mai intai clasa obiectului tinta, se adauga acea operatie, dupa care se revine la mesaj si i se seteaza atributul signature, alegand din model operatia nou adaugata. Ulterior, pot fi adaugate și argumente (daca e cazul), pentru ca logica interactiunii sa poată fi urmarita în mod natural.