Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei,

Universitatea Politehnica, Bucuresti

Tema Curs

Inginerie Software

INTRODUCERE IN UML(UNIFIED MODELING LANGUAGE)

Business Process Modeling Notation

 Studenti: Barda Alexandru Mihai

Sanduleac Cristian Nicusor

 Grupa 443A

**Cuprins**

**Barda Alexandru Mihai**

**Introducere in UML**

**Modelarea Proceului Business (Business Process Modelling)**

**PMN (Process Modeling Notation)**

**BM (Business Process)**

**Intrari, Resurse si Informatie**

**Sanduleac Cristian Nicusor**

**Evenimente**

**Rezultate**

**Obiective**

**Asamblarea elementelor**

**Trasabilitate**

**Bibliografie**

**Introducere in UML**

UML (Unified Modelling Language) este, asa cum numele sugereaza un limbaj de modelare, nu o metoda sau un process. UML are la baza o notatie specifica si regulile gramatice aferente construirii modelelor software. UML nu prescrie sau sfatuieste cum sa folosim acea notatie in procesul de creare de software sau ca metodologie de proiectare obiect-orientata.

UML suporta un set bogat de elemente de notatie grafice. El descrie notatia pentru clase, component, noduri, activitati, flux de lucru, obiecte, stari si cum sa modelam relatiie intre aceste elemente. UML suporta de asemenea si notiunea de extensii personalizate prin elemente stereotipice.

UML furnizeaza beneficii significante inginerilor software si organizatiilor prin constructia riguroasa de modele care pot fi urmarite si intretinute si care suporta intregul ciclu de viata al dezvoltarii software.

Aceasta lucrare pune accent pe extensiile personalizate ale UML-ului. Aceste extensii suporta modelarea proceselor *business*. Scopul acestor extensii, reprezentarile lor grafice si momentul in care modelarea de procese poate fi folosita in ciclul de viata al dezvoltararii de software este discutat.

**Modelarea procesului *business*(Business Process Modelling)**

Aceasta modelare este o component esentiala in orice process de dezvolatare software. Ea permite analistului sa cuprinda gama larga si procedurile care guverneaza ce face proces *business*. Acest model ofera o generealizare in ceea ce priveste unde se va integra procesul software propus in structura organizational si in activitatie de zi cu zi. Poate de aseamea sa ofere justificarea pentru crearea sistemului prin captarea manulului current si procedurile automate care vor fi integrate intr-un nou sistem si prin asocierea beneficiilor costului.

Fiind un model timpuriu, el permite analistului sa capteze evenimente importante, intrarile, resursele, si iesirile asociate cu procesul de *business*. Prin conectarea versiunilor mai noi de proiectare la procesul *business*, prin legaturi de implementare, este posibil sa cream un model ce poate fi urmarit in deplin, de la gama larga de procese pana la cerintele functionale si eventual pana la crearea efectiva a artefactelor software.

Deoarece BPM are in mod obisnuit o gama mai larga si mai cuprinzatoare decat este considerat sistemul software, acesta permite de asemenea analistului sa creeze o harta ce are ca scop modelul de implementare al sistemului si ce va fi implementata in alte modalitati.

**PMN (Process Modeling Notation)**

Un model de proces business defineste de obicei urmatoarele modele:

* Scopul sau motivul procesului;
* Intrari specifice;
* Iesiri specifice;
* Resurse consummate;
* Activitatile care sunt effectuate intr-o anumita ordine; si
* Evenimete care conduc procesul.

Procesul *business:*

* Poate afecta mai mult de o singura unitate organizational;
* Poate avea un impact organizational orizontal;
* Creeaza anumite valori pentru client. Clientii pot fi interni sau externi.

**BM (Business Process)**

Un BM este o colectie de activitati create sa produca un anumit rezultat pentru un anumit client sau pentru o anumita piata. El implica o empatie puternica asupra modului cum este efectuata munca intr-o organizatie, in contrast cu rolul unui produs. Un proces este astfel o ordonare specifica a activitatilor de munca aspura timpului si locului, cu un inceput, un sfarsit si o definire clara a intrarilor si a rezultatelor: o structura pentru o actiune anume.

Notatia folosita este ilustrata mai jos:



 Img. 1

Notatia procesului implica un flux al activitatilo de la stanga la dreapta. In mod normal, un element eveniment este plasat la stanga procesului si rezultatul la dreapta acestuia. Pentru a nota activitatile interne, elementele activitatii UML pot fi plasate in interiorul elementului process.

**Intrari, Resurse si Informatie**

Procesele *business* folosesc infromatia pentru a crea sau a completa activitatile lor. Inforrmatia, spre deosebire de resurse, nu este consumata in process – mai degraba este folosita ca parte a procesului de transformare. Informatia poate veni din surse externe, de la clienti, din unitati organizationale interne sau poate fi chiar produsul altor procese.

O resursa este o intrare intr-un process *business* si, spre deosebire de informatie, este in mod normal consumata in timpul procesarii, De exemplu, asa cum fiecare servici de tren zilnic functioneaza si actualitatile inregistrate, serviciul de resurse este ‘epuizat’ in ceea ce priveste procesul de inregistrare a timpilor trenurilor.

Notatia ce ilustreaza informatia si resursele este afisata mai jos:



 Img. 2

O legatura de livrare (supply) arata ca informatia sau obiectul legat de proces nu sunt epuizate in faza de procesare. De exemplu, pot fi folosite sabloane de ordine mereu pentru a furniza noi ordine pentru un anumit stil – sabloanele nu sunt alterate sau epuizate in aceasta activitate.

O legatura de intrare indica ca obiectul atasat sau resursa sunt consumate in procedura de procesare. Ca de exemplu, in timp de comenzile clientului sunt procesate, ele sunt completate si trimise, si de obicei sunt folosite o singura data pe resursa unica.

**Evenimente**

Un eveniment este primirea unui obiect, un timp sau o data atinsa, o notificare sau un alt trigger care intiaza procesul *business*. Evenimentul poate fi consumat si transformat (de exemplu o comanda a unui client) sau pur si simplu ca un catalyst.



 Img. 3

**Rezultate**

Un proces  *business* va produce in mod obisnuit unul sau mai multe rezultate valoroase afacerii, fie prin folosire interna sau pentru satisfacerea cerintelor externe. Un rezultat poate fi un obiect fizic (un raport), o transformare a resurselor brute intr-un nou aranjament(un program zilnic) sau un rezultat de ansamblu precum completarea cererii unui client.



 Img. 4

O legatura a rezultatului arata ca procesul *business* produce un obiect(fie fizic sau logic) care este valoros pentru organizatie, fie ca un element vizibil extern sau ca un produs intern(posibilitate de a hrani alt process).

**Obiective**

Un process *business* are un anumit obiectiv bine definit. Acesta este motivul pentru care organizatia face acest tip de munca, si trebuie definit din punct de vedere al beneficiilor pe care le are acest process pentru organizatie.



 Img. 5

O legatura a obiectivului arata ca obiectul atasat procesului *business* descrie scopul procesului. Obiectivul este justificarea procesului pentru a face activitatea.

**Asamblarea elementelor**

Diagrama de mai jos arata cum diverse elemente de model pot fi grupate impreuna pentru a produce o imagine coerenta a unui process *business*  anume. Sunt incluse intrarile, rezulatele, evenimentele, obiectivele si alte resurse semnificative.



 Img. 6

**Trasabilitate**

Trasabilitatea reprezinta modul prin care un proces *business* va fi implementat in sistemul propus. Intr-o diagrama de implementare, pachetele si altele artefacte de model pot fi conectate cu procesul  *business* cu legaturi de implementare (<<implementation>>) ce semnifica relatia de dependenta.



Exemplu de mai sus ilustreaza cum un process *business* este implementat prin metoda ‘Use Case’ si un pachet. Pe masura ce modelul se dezvolta si componenetele software functionale sunt create si legate de Use Case, justificarea *business* pentru fiecare element poate fi provenita din acest model.

**Bibliografie**

Hans-Erik Eriksson, Magnus Penker, Business Modeling with UML

Sinan Si Alhir, UML in a NutShel

 Doug Rosenberg with Kendall Scott ,Use Case Driven Object Modeling with UML.