TEMA 2
TEUSDEA ADRIAN-PAUL

# Domain driven design - domain event

Un eveniment de domeniu este o anumita actiune ce s-a petrecut in domeniul in care alte parti ale aceluiasi domeniu sa fie constiente de aparitia acelui eveniment. Partile care au fost notificate de aceasta aparitie vor reactiona intr-un anume fel la acele evenimente, iar efectele secundare se pot exprima intr-un mod implicit.

Sunt foarte importante aceste evenimente de domeniu, deoarece se pot exprima in mod explicit regulile de domeniu, bazandu-se pe un limbaj omniprezent.Considerand de exemplu clasele din acelasi domeniu putem spune faptul ca este util acest domeniu de evenimente pentru separarea preocuparilor dintre clase.

Un alt aspect important al DDD in domeniul evenimentelor il reprezinta faptul ca reactile aparute in urma aparitiei unui eveniment, duce la schimbarea sistemului, iar aceste modificari pot furniza un jurnal de audit, in care se remarca transabilitatea evenimentelor. Fiecare eveniment de domeniu va capta informatii din exterior, inregistrand-ule cu ajutorul jurnalelor.

Folosirea unui domeniu de evenimente presupune un stil de arhitectura mai distinct asupra aplicatiei, dar si asupra regulilor de programare care construiesc sistemul. Un avantaj al utilizarii acestui domeniu ar fi faptul ca avem la dispozitie un jurnal de audit care ofera o inregistrare complete, care poate fi folosit si in procesul de depanare. Datorita acestui jurnal implementarea unui domeniu de evenimente poate fi folositoare, deoarece pot aparea cazuri in care sistemul intra in stari la care programatorul poate nu s-a gandit, dar avand acel jurnal, sansele de a rezolva problema cresc. Tot prin faptul ca se poate vizualiza sursa unui eveniment poate duce la o intelegere mai bine a sistemului, iar datorita acestui aspect modificarile sunt facute prin intermediul domeniului de evenimente.

Se precizeaza faptul ca evenimente fac referinta si la momentul in care acestea s-au intamplat, deci un aspect foarte important este ca vor exista informatii legate de timpul producerii. Se traseaza 2 repere, primul fiind momentul de timp in care evenimentul a aparut, iar al 2-lea ar fi momentul in care evenimentul ce a avut loc a fost observat, rezultand astfel notiunea de timp real si de inregistrare, recomandandu-se a fi trecute pentru a nu exista confuzie. Este posibil sa fie nevoie de mai multe puncte de inregistrare in timp, deoarece pot fi observate evenimente de la sisteme diferite.

Evenimentele de domeniu ar trebui sa apara aproape instant, in cele mai multe dintre cazuri acestea apar in timpul procesului si in acelasi domeniu. Deci, evenimentele de domeniu pot fi sincrone sau asincrone, in timp de evenimentele de inregistrare vor fi intodeauna asincrone.

TEMA 2
TEUSDEA ADRIAN-PAUL

Diferenta dintre domeniul de evenimente si domeniul de integrare, reprezinta in parte acelasi lucru, in sensul ca raporteaza notificarile despre ceva ce s-a intamplat, insa implementarea acestor doua concepte trebuie sa fie una diferita, in sensul in care evenimentele de domeniu sunt doar mesaje impinse catre un dispecer de evenimente, care poate fi implementat ca si un arbitru in memorie. Rolul domeniului de integrare este acela de a propaga actualizari sistemelor aditionale, indiferent de marimea acestor servicii.

Se pune problema maririi domeniului de evenimente, astfel incat sa fie gestionate si evenimentele conexe. Ca si mijloace de extensie s-au propus ultilizarea unei clase statice pentru gestionarea evenimentelor, ca de exemplu clasa DomainEvents prin care se genereaza evenimente de domeniu imediat cand este apleata, insa aceasta modalitate face procesul de testare si depanare unul mai greoi, deoarece imediat apare executia handler-ului.

Pentru rezolvarea problemei de testare exprimata mai sus, se recomanda inregistrarea evenimentelor de domeniu inainte de a realiza tranzactia, practice se defineste un container pentru evenimentele din domeniul nostru de obiecte.

Receptorul poate sa fie indiferent fata de expeditor, in sensul ca nu trebuie sa inteleaga ceva despre el, insa ceea ce este important este intelegerea mesajului pe care emitatorul il propaga, pe scurt acestia sunt constransi de mesaj. Acest aspect de indiferenta intre receptor si emitaor este foarte folositor in sisteme mari, sisteme care sunt separate prin module.

Mesajele sunt imutabile se remarca prin faptul ca avem un mesaj trimis catre mai multe destinatii. In momentul in care mesajul ajunge la un anumit nod, iar acel nod ar schimba mesajul inainte sa ajunga la un alt nod, atunci mesajul nu va fi afectat nici la sursa, nici la nodurile care au receptionat mesajul initial de la emitator.

Principiul de plicuri reprezinta un aspect interesant, deoarece se poate proteja mesajul in in interiorul unui “plic” de la trimitere pana la receptie, iar toate entitatile care nu sunt implicate in procesul de transport vor astepta pana cand un plic cu mesaje le sunt atribuite.

Numele evenimentelor reprezinta alt aspect important pentru transabilitiate. Se recomanda utilizarea unor propozitii mici, simple de inteles, in care se exprima exact evenimentul aparut, ca si exemplu putem spune: “STOC EPUIZAT”, “TRANZACTIE ESUATA”.

Asadar, gandirea unui domeniu de evenimente si construirea sa difera foarte mult de la sistem la altul, deoarece sistemele comunica intre ele si este foarte important sa se respecte anumite reguli, dar de asemenea este important urmarirea procesului de aparitie al evenimentelor si modul in care acestea sunt gestionate si salvate. Prin faptul ca sistemele comunica prin mesaje este de remarcat faptul ca de la un sistem la altul nu se stie nimic de structura interna a sistemului

cu care comunica, ceea ce poate duce la o limitare, insa daca granitele sunt alese bine din start, atunci problema se poate rezolva. Un alt considerent important este existenta protocoalelor sub care mesajul este trimis, ceea ce da sens unui mesaj, altfel fara aceste protocoale nu ar exista nicio comunicatie, iar domeniul evenimentelor ar fi trebuit gandit altfel fata de cum este in ziua de astazi.