Indreica Claudia-Nicoleta
AC - Ingineria Sistemelor, anul 4, semigrupa 3.1

Domain Driven Design – Domain Events

Domain events:

- ofera o modalitate de a descrie activitati importante sau schimbari de stare ce au loc in sistem, dupa care alte parti ale domeniului pot raspunde la aceste evenimente.
- reprezinta o parte foarte importanta a unui context limitat(bounded context). Contextul limitat este o descriere a unei limite (de obicei, un subsistem sau o activitate a unei anumite echipe) în care un anumit model este definit și aplicabil.)
- se refera la ceva ce s-a intamplat in trecut, iar cum nu se poate modifica trecutul, domain events sunt imuabile. Drept urmare, in denumirea evenimentului se folosesc verbe la timpul trecut, pentru a arata ca o actiune deja s-a petrecut.

Astfel, obiectele ce genereaza evenimentele nu vor fi 'preocupate' de comportamentul ce trebuie sa apara la producerea evenimentului si de asemenea, obiectele ce manipuleaza evenimentele nu au nevoie sa stie de unde a provenit evenimentul.

Spre exemplu, interfata cu utilizatorul din Windows Forms (.Net Framework) ne familiarizeaza cu ideea de evenimente, folosind handle-uri pentru acestea. In acest exemplu avem o singura pagina cu un singur buton, iar in codul din spate putem vedea ca avem 2 handle-uri pentru fiecare eveniment: unul pentru situatia in care pagina este incarcata, iar celalalt pentru situatia in care butonul este apasat.

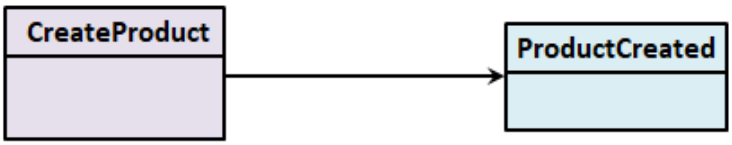
Evenimentele sunt utile pentru ca ne lasa sa evitam logica conditionata, putandu-se semnaliza cand un anumit eveniment a avut loc si astfel, un listener va putea actiona mai departe, in mod cocrespunzator.

```
namespace SalesApplication
{
    public partial class Page : Form
    {
        private void Page_Load(object sender, EventArgs e)
        {
            //Handle Page_Load event
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            //Handle Button1 click event
        }

        public Page()
        {
            InitializeComponent();
        }
    }
}
```

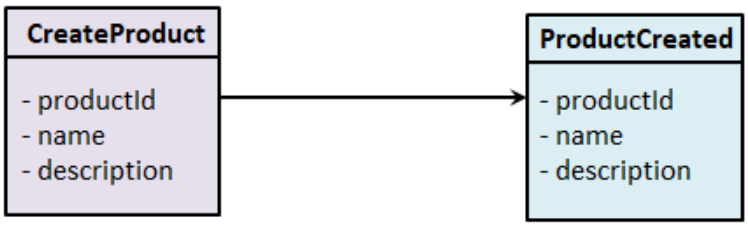
Domain events ofera aceleasi avantaje modelului ca si evenimentele din interfata cu utilizatorul, in sensul ca: in loc sa fim obligati a include intregul comportament de care s-ar putea sa avem nevoie ori de cate ori se schimbă starea obiectului nostru, putem genera un eveniment in acest sens, apoi putem scrie cod separat cu privire la eveniment, pastrand simplitatea designului modelului nostru si ajutandu-ne sa ne asiguram ca fiecare dintre clasele noastre are o singura responsabilitate.

Regula importanta: a nu se crea domain events decat in cazul in care chiar avem nevoie de acestea, adica daca avem un comportament ce trebuie să fie declansat atunci când un eveniment are loc si se doreste a se desparti comportamentul de declansarea acestuia.
Aceasta regula se aplica atunci cand comportamentul nu apartine clasei care o declanșeaza.
Fiecare eveniment de domeniu ar trebui sa aiba clasa proprie.

In ceea ce priveste numele domain event-ului si proprietatile sale, acestea ar trebui sa fie indeajuns de sugestive, incat sa transmita pe deplin ce s-a intamplat in domain model. Drept urmare, trebuie sa ne punem intrebarea: *"Care este factorul declansator ce determina publicarea domain event-ului?"*



In acest exemplu, "*ProductCreated*" apare in urma comenzii "*CreateProduct*". Deci putem spune ca "*ProductCreated*" este rezultatul comenzii "*CreateProduct*".



Comanda "*CreateProduct*" are cateva proprietati:

- *productId* - care identifica produsul ce se creaza,
- *name* – numele produsului
- *description* – descrierea produsului

Fiecare dintre aceste proprietati sunt esentiale in crearea unui produs, drept urmare, domain event-ul "*ProductCreated*" ar trebui sa contina toate aceste proprietati care au fost

furnizate cu comanda ce a generat produsul. Aceste proprietati vor informa toti subscriberii despre ce s-a intamplat in model: un produs a fost creat, a fost identificat unic prin *productId*, avand atribuite *name* si *description*.

In concluzie, cand se proiecteaza un eveniment, trebuie sa ne gandim la detaliile specifice evenimentului pe care dorim sa le captam. Raspunzand la intrebarea: "*Ce informații ar trebui să declanșeze din nou evenimentul?"* obtinem un set de informatii care sunt importante pentru acest eveniment. In mod similar, este posibil sa avem nevoie sa cunoastem identitatile tuturor agregatelor implicate in eveniment, chiar daca nu includem intregul agregat in sine. Acest lucru va permite operatorilor de evenimente sa retraga informatiile din sistemul care le-ar putea solicita atunci cand gestioneaza evenimentul. Insa trebuie sa tinem cont de faptul ca obiectele domain event-urile ar trebui sa contina suficiente informatii pentru a gestiona evenimentul, dar nu intr-atat de multe incat evenimentul sa fie ingreunat.