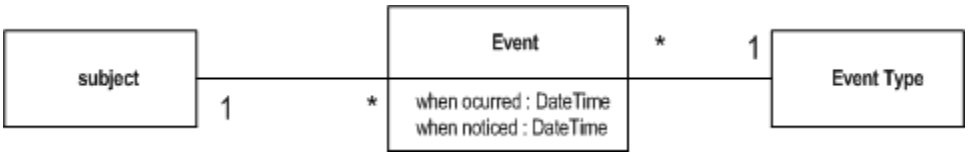
Bortica Dumitru-Daniel
An IV, Grupa 1.1

Tema 2
Domain Driven Design - Domain Event

Un event este ceva ce s-a întâmplat în trecut. Un domain event este ceva semnificativ ce a avut loc într-un domeniu, pentru care, anumite părți din acel domeniu trebuie să fie conștiente. Părțile notificate reacționează, într-un anumit fel, la evenimente.



Ideea este în felul următor: dacă se dorește indicarea unui eveniment semnificativ pentru un anumit domeniu, acel eveniment trebuie să fie marcat, iar celelalte clase din modelul de domeniu trebuie să fie lăsate să se aboneze și să reacționeze la acesta. Un beneficiu important al domain event este faptul că efectele secundare pot fi exprimate în mod explicit.

Esența unui domain event constă în faptul că este utilizat pentru a capta lucruri care pot declanșa o schimbare a stării aplicației în curs de dezvoltare. Aceste obiecte eveniment sunt apoi procesate pentru a provoca modificări ale sistemului și sunt stocate pentru a furniza un jurnal de audit.

Dacă se utilizează Entity Framework și trebuie să existe o reacție la un anumit eveniment, probabil că trebuie să se codeze aproape tot ce este nevoie ca să declanșeze evenimentul, deci regula este cuplată, implicit la cod și trebuie analizat codul pentru a spori faptul că regula este pusă în aplicare.

Pe de altă parte, folosirea domain events face acest concept să fie explicit, deoarece există un DomainEvent și cel puțin un DomainEventHandler implicat.

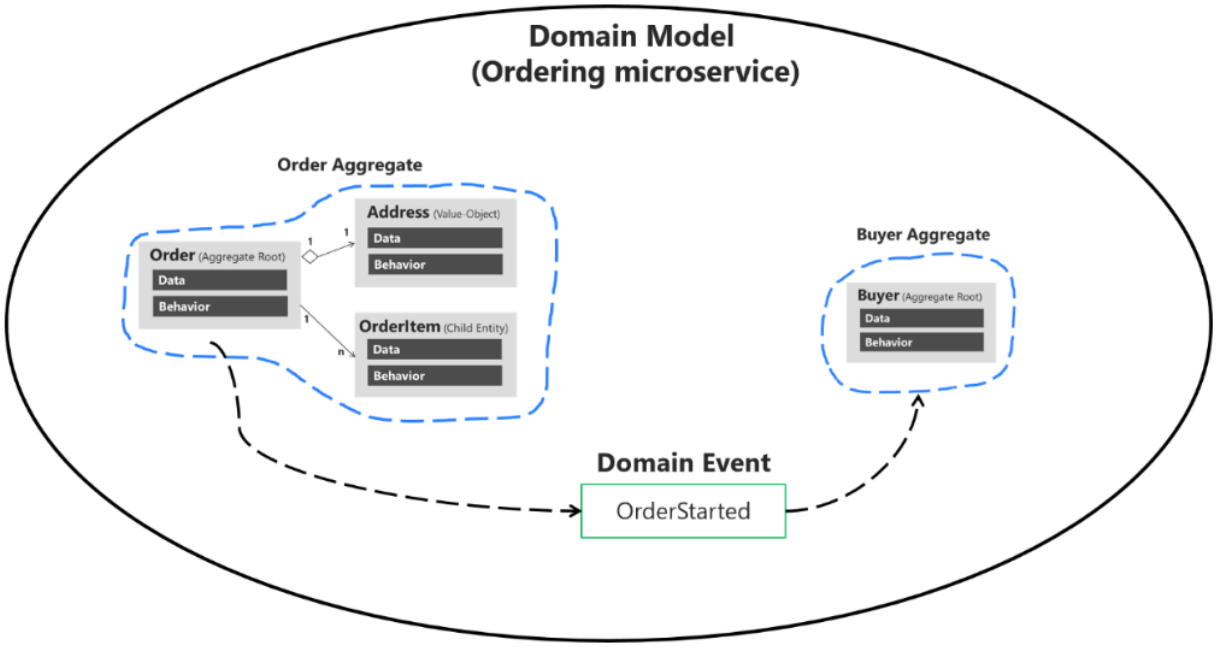
Domain events ajută la exprimarea în mod explicit a regulilor de domeniu, bazate pe limbajul omniprezent oferit de experții domeniului. Domain events permit, de asemenea, o mai bună separare a preocupărilor între clasele din același domeniu.

Este important să se asigure că, la fel ca o tranzacție de baze de date, fie toate operațiunile legate de un domain event se termină cu succes, fie nici una dintre ele nu o face.

Domain events sunt similare cu evenimentele de tip mesagerie, dar cu o diferență importantă. În mesageria reală, un mesaj este trimis întotdeauna asincron și comunicat între procese și mașini. Acest lucru este util pentru integrarea mai multor Bounded Contexts, microservicii sau chiar aplicații diferite. Cu toate acestea, cu domain events, se dorește ridicarea unui eveniment din operația de domeniu care se execută în prezent și se dorește ca orice efecte secundare să apară în același domeniu.

Domain events și efectele lor secundare (acțiunile declanșate ulterior care sunt gestionate de agenții de procesare a evenimentelor) ar trebui să apară aproape imediat, de obicei în timpul procesului și în același domeniu. Astfel, domain events ar putea fi sincrone sau asincrone. Evenimentele de integrare ar trebui să fie întotdeauna asincrone.

Dacă se execută o comandă legată de o instanță agregată, necesită rularea unor reguli de domenii suplimentare pe unul sau mai multe agregate suplimentare și trebuie să se realizeze proiectarea și implementarea acelor efecte secundare care vor fi declanșate de domain events. După cum se arată în figura de mai jos, fiind unul dintre cele mai importante cazuri de utilizare, un domain event ar trebui să fie utilizat pentru a propaga modificările de stare pe mai multe agregate în cadrul aceluiași model de domeniu.



În figură, atunci când utilizatorul inițiază o comandă, domain eventul OrderStarted declanșează crearea unui obiect Buyer în microserviciul de comandă, pe baza informațiilor originale ale utilizatorului din microserviciul de identitate (cu informațiile furnizate în comanda CreateOrder). Domain eventul este generat de agregatul comenzii, în primul rând atunci când este creat.

Alternativ, rădăcina agregată poate fi abonată pentru evenimentele ridicate de membrii agregatelor sale (entități copil). De exemplu, fiecare entitate copil al OrderItem poate ridica un eveniment atunci când prețul elementului este mai mare decât o anumită sumă sau atunci când suma elementului de produs este prea mare. Rădacina agregată poate primi apoi acele evenimente și poate efectua un calcul global sau o agregare globală.

Manipularea domain events este o preocupare a aplicației. Stratul de model de domeniu ar trebui să se concentreze doar pe logica domeniului - lucruri pe care un expert de domeniu le-ar înțelege, nu pe infrastructură de aplicații precum manipulatorii și acțiuni de persistență a efectelor secundare care utilizează repositories. Prin urmare, nivelul stratului de aplicație este locul în care ar trebui să fie operatorii de evenimente care declanșează acțiuni, când un domain event este ridicat.

Domain events pot fi, de asemenea, folosite pentru a declanșa orice număr de acțiuni de aplicare și, ceea ce este mai important, trebuie să fie deschis pentru a crește acest număr în viitor într-un mod decuplat. De exemplu, atunci când ordinea este pornită, este posibilă dorința de a publica un domain event pentru a propaga acele informații în alte agregate sau chiar pentru a ridica acțiuni de aplicație cum ar fi notificările.

Punctul cheie este numărul deschis de acțiuni care trebuie executate atunci când apare un domain event. În cele din urmă, acțiunile și regulile din domeniu și aplicație vor crește. Complexitatea sau numărul de acțiuni ale efectului secundar atunci când se întâmplă ceva, se va dezvolta, iar dacă codul a fost cuplat cu "lipici" (adică crearea de obiecte specifice cu "new"), atunci de fiecare dată când este nevoie să se adauge o acțiune nouă, va fi nevoie să se modifice codul de lucru și cel testat.

Această schimbare ar putea avea ca rezultat noi bug-uri și această abordare contravine principiului Deschis-Închis de la SOLID. De asemenea, clasa originală care a orchestrat operațiunile ar crește încontinuu, ceea ce contravine Principiului Singurei Responsabilități (SRP). În ciuda naturii sale neobișnuite, există câteva avantaje semnificative pentru utilizarea acestei abordări:

- jurnalul de audit al evenimentelor oferă o înregistrare completă, care este valoroasă atât pentru audit, cât și pentru depanare. Dacă sistemul nu mai funcționează corespunzător, există un jurnal complet al tuturor inputurilor. Prin stocarea evenimentelor care au fost efectiv procesate se diminuează șansele de a neglija scrierea unor informații importante în jurnalul de audit.

- fluxurile de evenimente clare facilitează ca alt sistem să înlocuiască parțial sau total o aplicație prin adăugarea unui router de mesaje în viitor, pentru a redirecționa evenimentele către un nou sistem. Deși nu este la modă proiectarea unui sistem într-un mod care să faciliteze eventualul lui deces, frecvența redusă a proiectelor de înlocuire a sistemelor ar trebui să însemne că ar trebui să acordăm mai multă atenție acesteia.

- domain event este deosebit de important ca șablon necesar pentru Event Sourcing, care organizează un sistem astfel încât toate actualizările să se facă prin Domain Event.

În concluzie, domain events se folosesc pentru a implementa în mod explicit efecte secundare ale modificărilor dintr-un domeniu. Pentru a utiliza terminologia DDD, se utilizează domain events pentru a implementa în mod explicit efectele secundare pe unul sau mai multe agregate. În plus, pentru o scalabilitate mai bună și un impact mai mic asupra blocărilor bazei de date, se recomandă utilizarea eventualei coerențe între agregatele din același domeniu.