

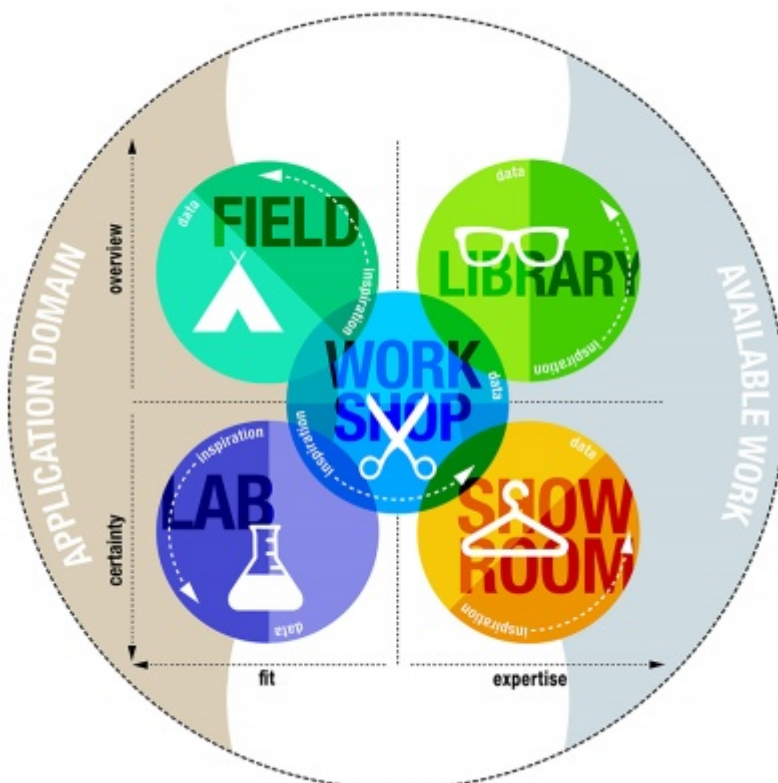
Design patterns



Introductie

Een ontwerppatroon of patroon (Engels: design pattern) in de informatica is een generiek opgezette softwarestructuur, die een bepaald veelvoorkomend type software-ontwerpprobleem op lost. Het patroon geeft geen concrete oplossing, maar biedt een soort sjabloon, waarmee het ontwerpprobleem kan worden aangepakt.

Aanpak



DOT Framework (Design Oriented Triangulation Framework) helpt bij het uitvoeren van praktijk gericht

onderzoek. Voor het onderzoek zal dit framework gebruikt worden. Het geeft structuur en biedt opties om zelf te bepalen hoe het onderzoek in te vullen voor de software ontwikkeling cyclus.

Bron: [DOT Framework](#)

Wat, Waarom, Hoe?

DOT Framework bestaat uit 3 verschillende niveau's die moeten helpen met de structuur van dit onderzoek. Hierin worden de 'Wat' 'Waarom' en 'Hoe' in beschreven.

Wat (de vraag van het onderzoek)

Er wordt in het programma van Beer.Cool informatie verzonden tussen klassen. De verzender en ontvanger mogen niet van elkaar weten hoe of waar deze informatie vandaan komt. De klassen mogen geen afhankelijkheid van elkaar hebben.

De hoofdvraag:

Kan het command-pattern gebruikt worden zodat de klassen(zender/ontvanger) geen afhankelijkheid van elkaar hebben en wanneer nodig ,een methode aan te roepen die (later) nodig is om de communicatie te voltooien?

Waarom (de reden van het onderzoek)

Om meer te weten te komen over bepaalde concepten heb ik besloten een proof-of-concept te maken die onderwerpen in het klein laat terugkomen. POC's dienen niet alleen als bewijslast, maar ook als geheugensteun om eventueel terug te kijken hoe iets precies werkt.

Wat lost het onderzoek op?

Het scheiden van klassen die anders afhankelijk zijn van elkaar om met elkaar te kunnen communiceren. Je kan het zien als een ontvangen en een zender. Als de zender iets wil doorgeven dan roept deze de ontvanger aan. Hierdoor ontstaat een afhankelijkheid.

Door het implementeren van het command-pattern wordt het mogelijk dat de ontvangen en de zender niet van elkaar af weten maar dat de communicatie tussen beiden wel goed verloopt. Het bericht bevat de juiste informatie en instructies zodat deze weet wat er moet gebeuren zonder contact gehad te hebben met de zender.

Subvraag

- a. [Hoe wordt encapsulatie bereikt?](#)
- b. [Hoe implementeer ik het command pattern](#)
- c. [Hoe complex wordt de code?](#)

Dit zijn de deelvragen die uiteindelijk moeten helpen om de hoofdvraag te kunnen beantwoorden.

Bronnen

Om informatie te krijgen over dit onderwerp en een beter begrip wordt er gebruik gemaakt van YouTube, Udemy, StackOverflow, Wikipedia en Google. Er is veel te vinden over het command-pattern en valt onder gedragspatronen. Een gedragspatroon indentificeerd de algemene communicatie patronen tussen objecten.

Hoe (onderzoek strategieën en methoden)

DOT Framework heeft 5 verschillende strategieën over de aanpak van onderzoek:



Bieb: Onderzoek naar wat al bestaat of al ooit gedaan is. Je gaat na welke theorieën er zijn over je onderwerp die in de context van je eigen project passen. Je zoekt informatie op of gebruikt bijvoorbeeld documentatie die door andere mensen geschreven is.



Veld: Je gebruikt een veld-strategie om een beter inzicht te krijgen wat de eindgebruikers van jouw product verwachten en wat ze hier mee willen bereiken, hoe ze het gebruiken etc.



Lab: Test je product door gebruik te maken van verschillende test methodieken die aangegeven worden in de lab strategie. Security, System, Unit testen maar ook statistische testen zijn allemaal methoden die je kunt gebruiken hiervoor.



Showroom: Laat de "positie" van jouw product zien tegenover die van andere. Je kunt je product laten zien aan experts of beschrijven wat er anders is aan jouw product ten opzichte van andere.



Workshop: Je gaat na wat je mogelijkheden zijn door middel van prototyping (POC), design en eventueel tutorials volgen. Je doet dit om inzicht te krijgen in de mogelijkheden in context van jouw product.

Voor de methoden die bij elke strategie horen zie bron: [DOT Framework Methods](#)

Gekozen strategieën



3 voorbeelden van hoe een command-pattern geïmplementeerd kan worden.

<https://github.com/Koffiemolen/Pattern-Command>

Conclusie

Gebaseerd op het onderzoek, ontwerp patroon Command is vrij complex om te implementeren en brengt een groot aantal kleine klassen met zich mee. Deze is uitermate geschikt voor het encapsuleren van de communicatie tussen objecten.