Generisk Programmering: Assignment 2
Visualisering

Jens Peter Rosenkvist[1]

Kevin Hejn[2]

Datalogisk Institut, Københavns Universitet
Universitetsparken 1, DK-2100 København Ø, Danmark

[1] di060191@diku.dk
[2] di060190@diku.dk

Resumé. Denne rapport handler om visualisering af kode i store udviklingsprojekter, evt. med flere millioner linjer kode. Visualiseringen skal give et let overblik over forskellige aspekter af koden så som alder, antal rettelser og kodekompleksitet. Der tages ikke udgangspunkt i et konkret værktøj men i generelle principper for visualisering.

1. Introduktion

Samtidig med at projekter bliver større og kodemængden stiger, vil koden ofte også blive mere og mere uoverskuelig. Uden ordentlig dokumentation kan det være en umulig opgave at sætte sig ind i et stort projekt, hvilket kan være nødvendigt ved eksempelvis fejlretning. Det kan derfor være nødvendigt at bruge værktøjer til at hjælpe med at præsentere koden, så programmøren hurtigst muligt kan forstå koden og finde frem til de relevante dele. Et sådant værktøj kan være *visualisering*.
Vi vil beskrive hvilke principper et visualiseringsprogram kan benytte og hvilket problemer det kan løse. Vi har taget udgangspunkt i [1], der beskriver hvordan de hos Bell Laboratories har designet et sådant program. Metoderne vi beskriver, er illustreret i artiklen.

2. Visualiseringsformer

En meget anvendt form for visualisering er indrykning af koden, kombineret med syntaksfarvning. Dette tjener til at give et hurtigt overblik over simple metoders struktur, men giver ikke nok information om den overordnede struktur når der er mange linjers kode. F.eks. vil et funktionskald sagtens kunne være et kald til en funktion på 100 linjer, hvor syntaksfarvning ikke vil hjælpe på forståelsen af kaldet. Det er derfor nødvendigt med visualiseringer der viser en mere generel repræsentation af koden. Vi vil nu

beskrive 4 mulige repræsentationsformer hvor den første er kodenær og de efterfølgende gradvist abstrahere mere.

2.1 Linjerepræsentation

En metode der minder om den almindelige tilgang er linjerepræsentation. I denne vises et udsnit af koden til venstre på skærmen. I midten vises et større udsnit med en mindre font. Det fremgår tydeligt hvilken del af denne kode der vises helt til venstre. Endelig vises der til højre et endnu større udsnit hvor fonten er så lille at den ikke kan læses, men indrykning er beholdt så koden kan genkendes og overskues. Ydermere er koden farvet så det fremgår hvor gammel de enkelte dele af koden er.
Denne metode kan bruges til at give et større overblik over koden generelt, såvel som sammenhængen, i forhold til den normale repræsentation. Metoden er dog stadig forholdsvis kodenær og giver kun overblik over et begrænset udsnit af projektet.

2.2 Pixelrepræsentation

I pixelrepræsentation forsøges pladsen på skærmen at bruges mere optimalt. Hver kodelinje reduceres til en enkelt eller få farvede pixels. Farven kan blandt andet indikere alderen på koden. Denne repræsentationsform kan f.eks. bruges til at vise indholdet af en lang række filer. Hver enkelt fil er repræsenteret ved et rektangel hvor størrelsen af filen er proportional med størrelsen af rektanglet. Dette gør det let at sammenligne filstørrelser og se hvilke filer der senest er blevet rettet i. En forholdsvis lille del af skærmen kan også bruges til at vise et udsnit af koden som normalt hvorved det er let at hoppe mellem de forskellige dele af koden ved hjælp af de nævnte rektangler. Denne del af koden kan eventuelt også vises med linjerepræsentation.
En ulempe ved repræsentationen er at den fortæller meget lidt om hvad koden gør. Desuden er det svært at vise relationer mellem metoderne. Ydermere kan hele projektet ikke ses her, men kun et begrænset udsnit. Da rektanglerne er proportionale med filstørrelserne, kan nogen filer muligvis ikke ses. Det vil eksempelvis ske, hvis der er en meget stor fil, der optager en stor del af skærmen.

2.3 Opsummeret filrepræsentation

I denne repræsentationsform forsøges en større del af projektet at medtages end ved de tidligere former. Hver enkelt fil repræsenteres med et rektangel. Der er dog kun fire forskellige størrelser. Filerne inddeles i kvartiler og herved bliver alle filer, inklusiv de meget små, vist. Farvningen i rektanglerne kan angive forskellige ting så som kodealder, antal fejlrettelser eller tilføjelser. Denne repræsentation kan bruges til at give et hurtigt overblik over en meget

stor del af koden. Som ved pixelrepræsentation gives der dog ingen detaljer om funktionaliteten af koden.

2.4 Hierarkisk repræsentation

I hierarkisk repræsentation forsøges der at give et overblik over hele projektet på én gang. Dette sker ved at dele informationerne op i en række niveauer. På det øverste niveau vises hver underdel af projektet. I hver underdel vises mapperne som er knyttet til den pågældende del. Her kan man via farver f.eks. illustrere hvor stor en del af koden i mapperne der er kommentarer. Hver underdel kan vises i en højere detaljegrad hvor det fremgår hvilke filer, hver mappe indeholder. Igen kan farvning bruges til at vise mængden af kommentarer, blot på filniveau.
Repræsentationen giver langt det største overblik over hele projektet, men der gives på ingen måde information om hvad koden gør.

2.5 Farvning

Der er i alle repræsentationsformer nævnt at man kan bruge farvning til at illustrere f.eks. alder, forfatter og antal rettelser ved koden. Fejl vil ofte være i kode der er blevet lavet for nylig da den ikke er blevet gennemtestet, så når en fejl skal findes, kan det ofte betale sig at undersøge den senest rettede kode. Ligeledes er der ofte fejl i kode der er blevet rettet mange gange da en rettelse let kan introducere en ny fejl. Mange rettelser kan derfor være en indikation på at koden bør omskrives. Endelig kan information om forfatteren bruges til at se hvem der har skrevet størstedelen af en fil så man ved hvem man skal opsøge for yderligere information.

3. Konklusion

Vi har beskrevet fire forskellige visualiseringsformer som alle hjælper med at overskue store mængder af kode lettere. Som det fremgår af beskrivelserne, kan meget store mængder kode repræsenteres på et begrænset område.
De fire metoder repræsenterer koden på forskellige niveauer og har enten høj detaljegrad eller stort overblik. Som repræsentationsformerne har vist, kan begge dele generelt ikke opnås samtidigt. Man må derfor vælge den form og dermed detaljegrad der passer til det aktuelle behov.

Litteratur

[1] T. Ball and S. Eick, Software visualization in the large, **Computer** **29**,4 (1996), 33–43.