Data Processing

Github Commit Graph

19-02-2016

Universiteit van Amsterdam

Minor Programmeren

Groep B1

Julia Jansen

Shannon Bakker

Tom Schoufour

Thom Mekelenkamp

**Analyse**1. De zwarte lijn in het Githubnetwerk representeert het hoofddocument. De bolletjes in de lijnen geven weer wanneer er iets is veranderd in het bestand. In de verschillende kleuren worden de verschillende versies van het document weergegeven. Als je met de muis over de bolletjes beweegt, is te zien welke gebruiker iets heeft veranderd aan het bestand. De grafieken verschillen van elkaar van vorm, omdat bij één project meerdere owners aan het bestand werken (CS171) en bij het andere project (Caleydo) niet. Verder is aan de cijfers boven de tijdlijn te zien op welke datum er iets is veranderd aan het bestand.   
 De attributen van deze netwerken zijn dus:

- verschillende **bestandsversies**

- verschillende **owners** van het project

- **datum** van wijziging in het project

- het zou wellicht relevant zijn om ook de **grootte van de verandering** weer te geven in het netwerk

- verder kan nog de **aard van de verandering** weergegeven worden in het netwerk. Dat zou dan aangegeven moeten worden door de schrijvers van het bestand. Die zouden dan moeten kiezen uit een lijst van mogelijke veranderingen. Verschillende soorten veranderingen kunnen worden weergegeven door de bolletjes andere vormen te geven (driehoek, sterretje, vierkantje etc.).

2. Er zijn verschillende rollen die we ons kunnen inbeelden in een groepsproject op github:

- supervisor (overzicht houden)

- teamleden (schrijven mee aan onderdelen van het project)

- eindredactie (opmaak en merge-checks)

3. Het moet uit het netwerk duidelijk worden wie er heeft bijgedragen aan het project en wanneer. Ook de aard van de bijdrages van de verschillende users zou interessant zijn. Ook zou het interessant zijn om weer te geven waarom er op sommige momenten niet kan worden doorgewerkt aan het bestand als er lang niks gebeurt.

4. We zullen onze visualisatie ontwerpen voor de supervisor. Verschillende attributen die dan zichtbaar moeten zijn:

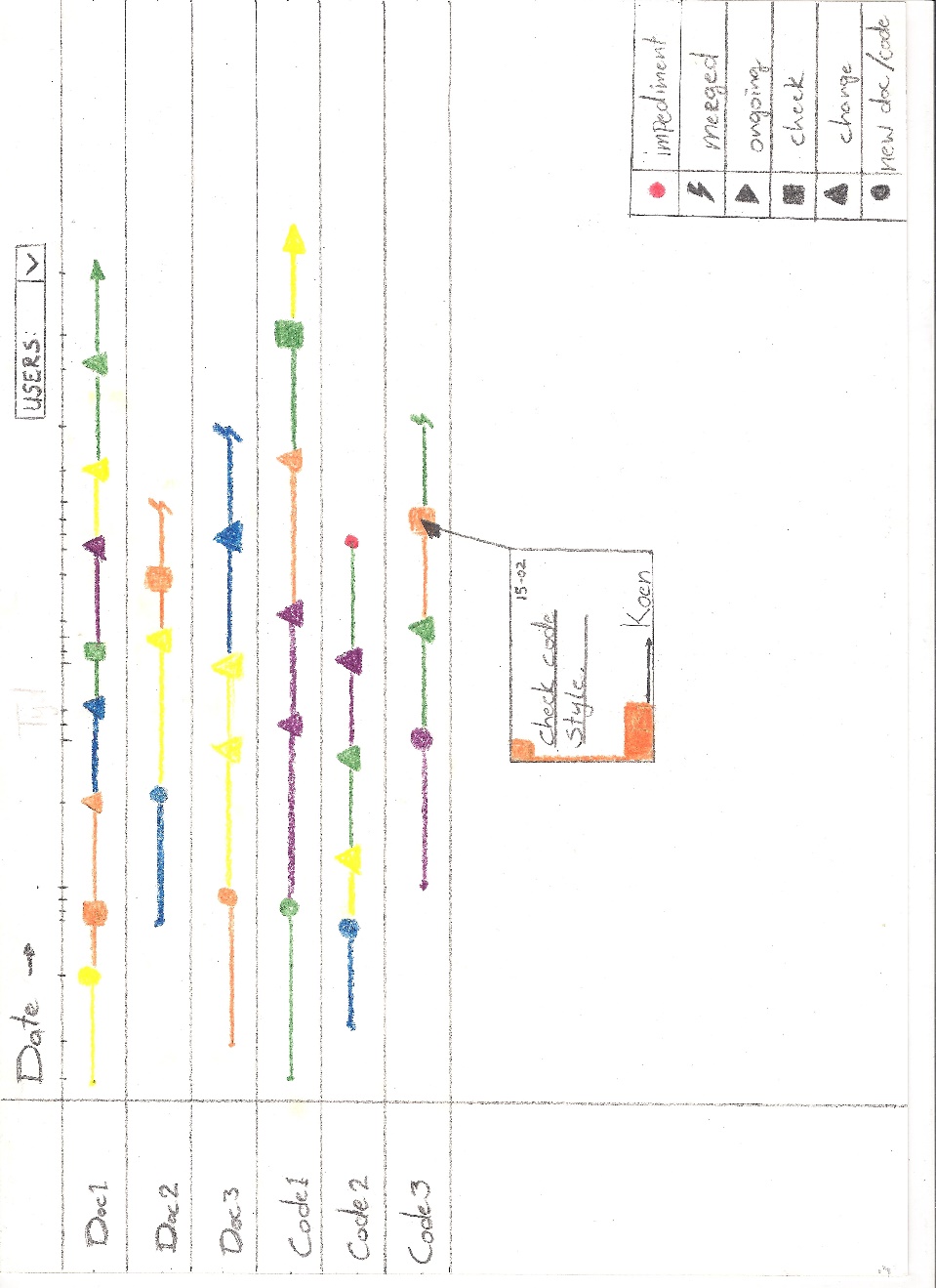
- de verschillende users die in het bestand werken

- de verschillende versies van het bestand

- datum van wijzigingen

- momenten wanneer document-versies gemerged worden

- de aard van de veranderingen die gedaan zijn

- momenten waarop schrijvers een probleem hebben waardoor ze niet verder kunnen werken

**Schetsen**

Schets 1:

* Onder elkaar worden de verschillende versies van het bestand weergegeven. Elk van deze documenten wordt weergegeven met een lijn die loopt vanaf de datum wanner die bestandsversie wordt aangemaakt totdat er (1) wordt gemerged met het hoofdbestand of (2) er een probleem is waardoor er niet verder gewerkt kan worden.
* De verschillende kleuren geven in deze visualisatie de verschillende schrijvers weer.
* De datum staat bovenaan (zoals ook in het netwerk op Github).
* De vorm van de nodes geeft weer wat de aard van de wijziging was.

Voordelen van deze schets:  
- In deze schets wordt voor het belangrijkste attribuut (welke document het betreft) de sterkste visuele variabele gebruikt: positie. De verschillende documenten zijn door hun positie onder elkaar van elkaar te onderscheiden.

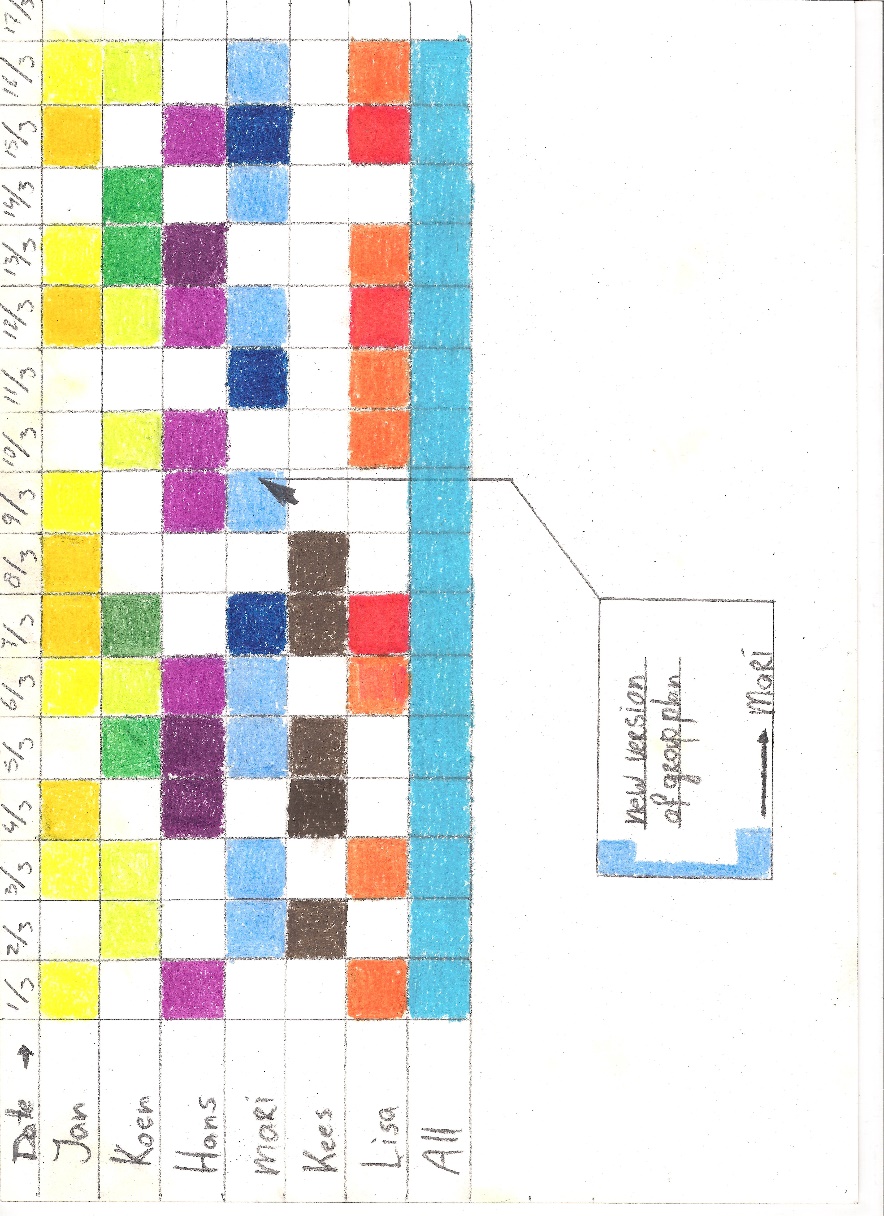
- Verder gebruiken we kleur en vorm van de nodes om respectievelijk de verschillende auteurs en verschillende soorten wijzigingen aan te geven. Er wordt geen ordening (van groot naar klein of andersom) in de data gebracht, dus deze visuele variabelen worden voor selectieve attributen gebruikt (Carpendale, 9-10, 12-13).

- Tot slot gebruikten we de lengte van de lijnen ten opzichte van elkaar om aan te geven hoe lang de documenten bestaan. Dit tijdsaspect wordt dus aangegeven met de visuele variabele lengte.

Nadelen van deze schets:

- Wanneer er heel veel schrijvers aan een bestand meewerken wordt de visualisatie onoverzichtelijk en onrustig voor het oog vanwege de vele kleurtjes. Het is echter wel goed te doen voor de lezer om op één kleur te focussen voor een inschatting van de hoeveelheid bijdrages van die auteur (Carpendale, 12-13).

- Als er heel veel verschillende documenten en documentversies zijn dan staan er heel veel lijnen onder elkaar, terwijl die er niet meer allemaal toe doen wanneer de supervisor het netwerk bekijkt.



Schets 2:

- In deze schets zijn alle schrijvers die aan een project hebben meegewerkt onder elkaar weergegeven. Deze hebben allemaal een eigen kleur. Het is voor een supervisor makkelijk te zien of alle schrijvers meewerken aan het project.

- De hokjes geven weer wanneer er een wijziging is gemaakt door die user. Wanneer je met je muis over een hokje beweegt is de samenvatting van de wijziging te zien.

- De hokjes hebben verschillende waardes op basis van de grootte van de wijziging ten opzichte van de totale omvang van het bestand op dat moment.

Voordelen van deze schets:

- Het belangrijkste attribuut in deze visualisatie wordt weergeven door de visuele variabelen positie (onder elkaar) en kleur. Alle meewerkende auteurs staan onder elkaar en hebben een eigen kleur. Kleur en positie worden dus als selectieve variabelen gebruikte.

- Voor de supervisor waar we ons bij deze schets op gericht hebben is het ook belangrijk om te kunnen zien hoe groot de bijdrages van de verschillende auteurs zijn. De waarde van de kleur in de hokjes geeft aan hoe groot de verandering aan het bestand op die datum was.

Nadelen van deze schets:

- De grootte van de bijdrages van de verschillende auteurs zijn moeilijk te vergelijken omdat het moeilijk is waardes van verschillende kleuren met elkaar te vergelijken (Carpendale).

- Deze visualisatie geeft weinig informatie over de algemene progressie van het hoofdbestand.

- Het aantal kleuren geeft een max aan hoeveel mensen er op de lijst gezet kunnen worden, omdat in deze schets alle waardes van een bepaalde kleur voor één auteur gereserveerd zijn.

- Je kan niet makkelijk in één oogopslag zien wat er wordt verteld, omdat het vooral een hoop kleurtjes is.

**Groepsreflectie**

We waren het er al snel over eens dat de visualisatie van Github aan veel vereisten voor het weergeven van een documentnetwerk voldeed. De eerste schets van onze alternatieve visualisaties heeft uiteindelijk de voorkeur uit onze twee opties. Echter zouden we nog wat interactieve elementen toe kunnen voegen om te voorkomen dat