haute école neuchâtel berne jura — arc gestion neuchâtel delémont

Hes·so Haute Ecole Spécialisée de Suisse occidentale

Génération de graphiques à partir de données en graphe

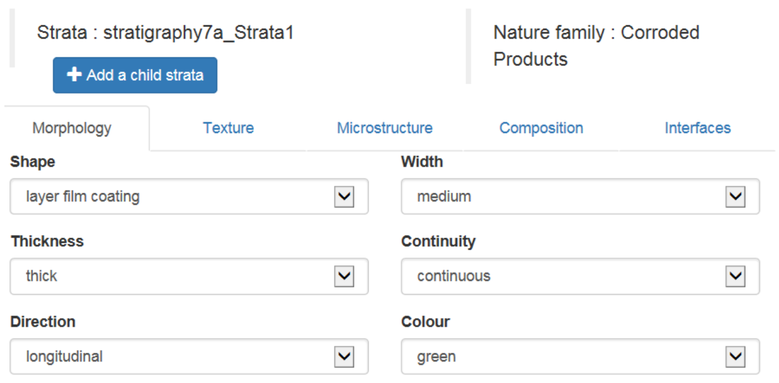
Thierry Hubmann

Contexte

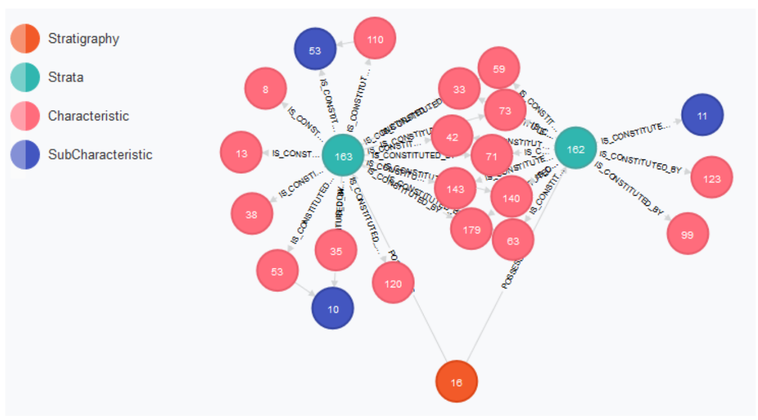
MiCorr est une base de données d'objets métalliques patrimoniaux qui permet d'en analyser les faciès de corrosion.

Exemple d'artefact métallique

Les conservateurs-restaurateurs saisissent les caractéristiques des couches de corrosion à l'aide d'un formulaire web.



Les caractéristiques des couches de corrosion ainsi que leurs propriétés sont enregistrées dans une **base de données en graphe**.



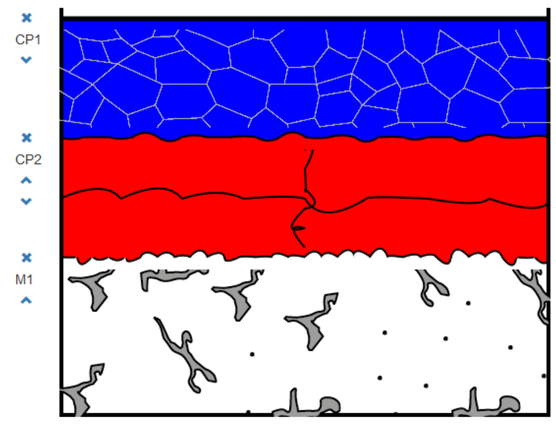
Base de données en graphe neo4j

Ces donnés sont ensuite utilisées pour rechercher des artefacts similaires à l'aide de requêtes Cipher ou pour générer des représentations visuelles des faciès de corrosion (stratigraphies).

La génération des représentations des stratigraphies est entièrement réalisée dans le browser en utilisant une librairie **JavaScript** pour générer le **code SVG** nécessaire à son affichage.

```
<svg xmlns:xlink="http://www.w3.org/1999/xlink" height="50" width="500"
    version="1.1" xmlns="http://www.w3.org/2000/svg" id="SvgjsSvg1013">
<rect fill="red" height="50" width="500" id="SvgjsRect1015"></rect>
<path stroke-width="2" stroke="black" fill="none" d="M0 25Q28.25 5 62.5 25 " id="SvgjsPath1016"></path>
<path stroke-width="2" stroke="black" fill="none" d="M62.5 25Q105.75 22 125 25 " id="SvgjsPath1017"></path>
<path stroke-width="2" stroke="black" fill="none" d="M125 25Q141.25 21 187.5 25 " id="SvgjsPath1018"></path>
<path stroke-width="2" stroke="black" fill="none" d="M187.5 25Q217.75 26 250 25 " id="SvgjsPath1019"></path>
<path stroke-width="2" stroke="black" fill="none" d="M250 25Q260.25 34 312.5 25 " id="SvgjsPath1020"></path>
<path stroke-width="2" stroke="black" fill="none" d="M312.5 25Q346.75 3 375 25 " id="SvgjsPath1021"></path>
<path stroke-width="2" stroke="black" fill="none" d="M375 25Q399.25 18 437.5 25 " id="SvgjsPath1022"></path>
<path stroke-width="2" stroke="black" fill="none" d="M437.5 25Q458.75 18 500 25 " id="SvgjsPath1023"></path>
<path stroke-width="8" stroke="black" fill="none" d="M0 0L0 50 " id="SvgjsPath1024"></path>
<path stroke-width="8" stroke="black" fill="none" d="M500 0L500 50 " id="SvgjsPath1025">
</path><defs id="SvgjsDefs1014"></defs></svg>
```

Code SVG généré en JavaScript dans le browser



Représentation graphique d'une stratigraphie

Problématique et contraintes

Problématique: L'implémentation actuelle génère les graphiques en JavaScript **du côté client (browser).**

Limites: Cette solution ne permet pas l'utilisation de ces graphiques en dehors de leur contexte (liste de graphiques ou export dans des formats PDF ou PNG).

Objectif: Ce travail vise à implémenter la génération des stratigraphies côté serveur tout en conservant la possibilité de les générer par le navigateur. On conservera ainsi les avantages des deux solutions selon le cas d'utilisation (réactivité vs flexibilité)

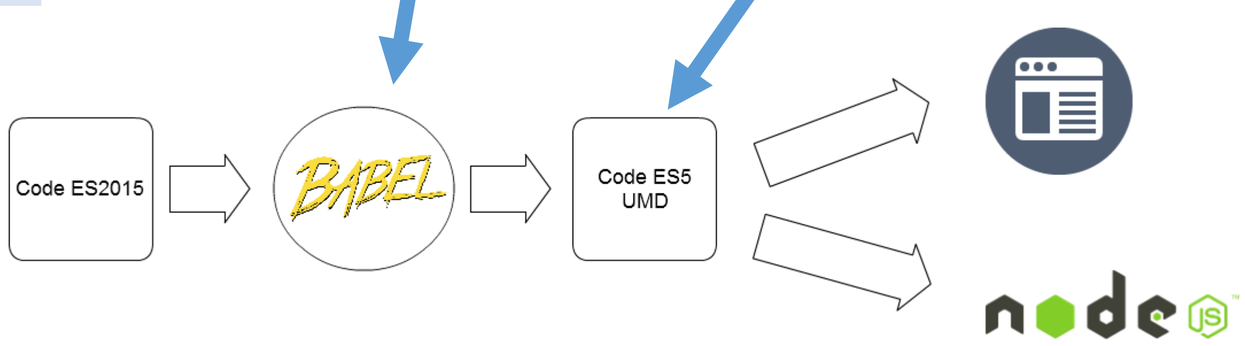
Contraintes: Pour éviter d'avoir de la redondance de code et pour faciliter la maintenance on cherche à utiliser **la même base de code** pour le client et pour le serveur. Cela, malgré leurs différences (DOM, accès aux fichiers, chargement des dépendances…)

Solution

Etape 1: Ecriture du code en EcmaScript 2015 afin de profiter d'un système de chargement des modules compatible avec le browser et un serveur.

Etape 2: Compilation du code avec Babel pour assurer la compatibilité avec les browser actuels

Etape 3: Déploiement du code compilé sur Node.js et dans le navigateur



Avantages:

- Solution pérenne
- Réutilisation de l'existant
- Une source de code utilisable directement par les deux environnements

Travail de Bachelor de la filière Informatique de gestion. Session 3IG PT, semestre de printemps 2016