

Extraction de la variabilité dans des familles de produits interconnectées

Cédric DURAND
Johann GOLMARD
Clément DESGENETEZ
Thomas DI GIOVANNI

Sommaire

1 - Jeu de données utilisé	1
2 - Sélection des données	1
3 - Nettoyage des données	1
4 - Extraction d'information	3
5 - Extraction de variabilité	3

1 - Jeu de données utilisé

Nous avons sélectionné comme jeu de donnée :
Comparison_of_WebSocket_implementations

Cela correspond à un PCMs () avec 19 features :

[frame_size_limit, protocol_test_report, version_compared, server, implementation_language/environment, automatic_pongs_for_pings, manual_pings/pongs, binary_message_support, automatic_heartbeat_pings, streaming_api_input/output, text_message_support, frame_based_api, license, protocol, message_based_api, self_hosted_server, flow_control_framework, client, api_language/environment]

2 - Sélection des données

Nous n'avons pas gardé les colonnes suivantes :

- Protocol test report, par manque de données.
- Frame Size Limit, il existait trop de variables non pertinentes.
- Version compared, jugé non nécessaire.

3 - Nettoyage des données

Colonnes modifiées :

- Streaming API input/output -> No/No en No

Colonnes divisées :

- Protocol (spec) version support en 2 :
 - RFC 6455 (Protocole normalisé par l'IETF dans la RFC 6455)

- Autres. Comprend les protocoles 0,7,8,10,13,17. Nous avons jugé les données trop disparates pour les séparer en différentes colonnes.
- Implementation language/environnement en 3 :
 - C++
 - Java
 - .NET
- API language/environnement en 4 :
 - C++
 - Java
 - .NET
 - JavaScript
- License en 10 colonnes, soit une par licence disponible.

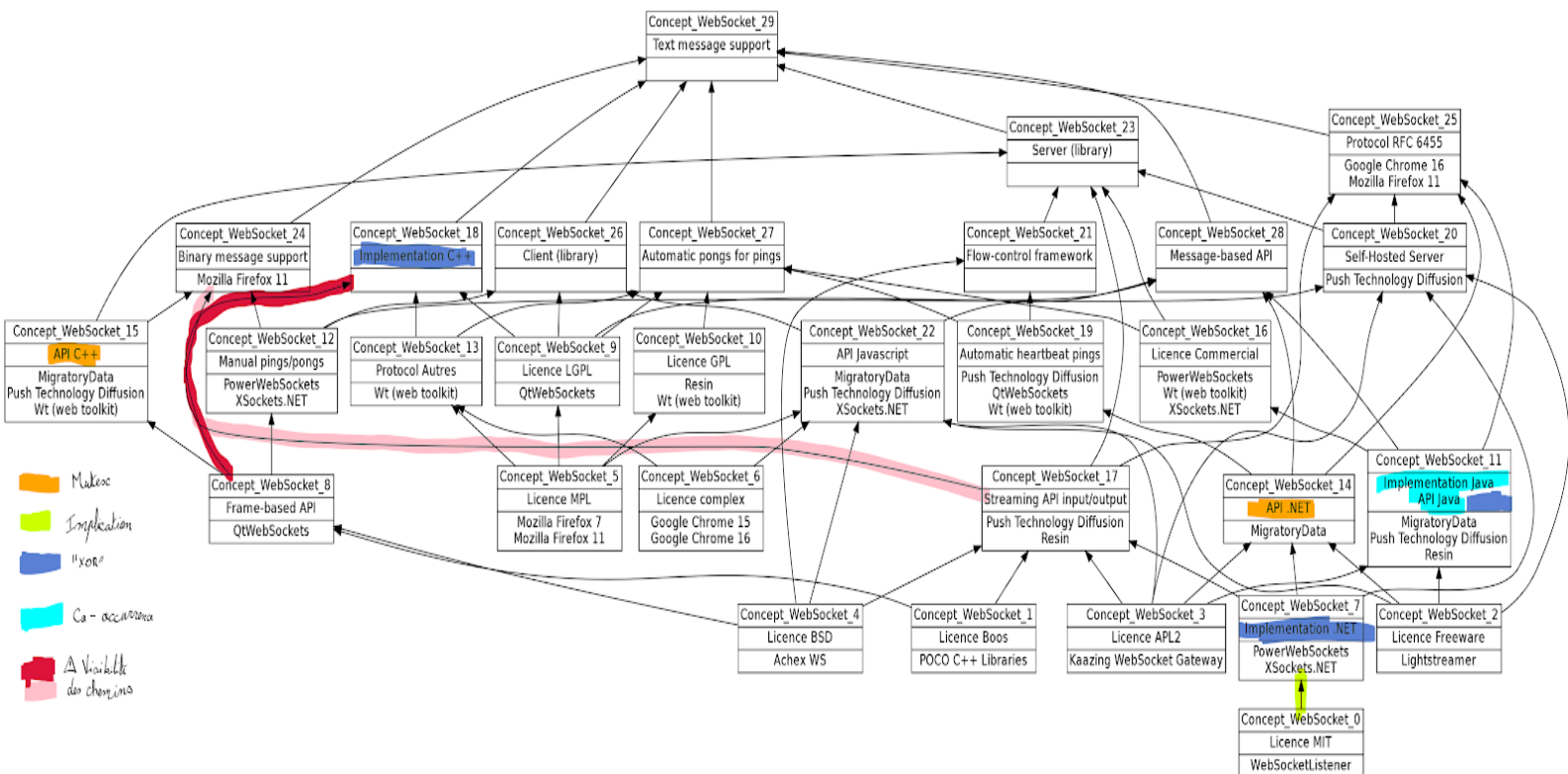
Quelques choix :

Qt est compté comme C++ (car très proche et basé sur du C++)

Les Java J2ME / J2SE sont comptés comme Java.

Au vu du nombres de choix possibles pour les APIs, nous avons décidé de prendre les plus représentés d'entre elles.

4 - Extraction d'informations



5 - Extraction de variabilité

1. Co-occurrence :

L'on peut voir dans le Concept_WebSocket_11 qu'il introduit la co-occurrence Implementation Java \Leftrightarrow API Java.

C'est bien le résultat que l'on attendait, une implémentation Java est tout le temps couplée avec une API Java.

2. Implication :

Sur les concepts Concept_WebSocket_14 et Concept_WebSocket_7 qu'il y a une implication du dernier Implementation .NET \Rightarrow API .NET vers le premier. (7 \Rightarrow 14). L'algorithme a trouvé une implication logique.

3. Mutex :

Un exemple de mutex en se basant sur les
Concept_WebSocket_15 et Concept_WebSocket_14 est que l'API
C++ => API .NET.

4. Groupes :

a. - XOR ->

Aucun exemple de XOR ne couvre entièrement le treillis,
cependant un exemple simple de XOR qui aurait pu convenir
ici serait entre les implémentations des différents langages
présents :

Text message support XOR {Implémentation C++,
Implémentation Java, Implémentation .NET}.

b. - OR ->

Aucun exemple de OR ne couvre entièrement le treillis,
cependant un exemple de OR qui aurait pu convenir ici serait
entre les différentes licences disponibles.