

Livrable 4.2

Ce document décrit les travaux poursuivis au sein du projet HUBBLE sur un langage permettant de formuler des requêtes sur les traces, et sur des opérateurs d'analyse des informations contenues dans les traces.

Le document est structuré en 3 parties :

- Un langage naturel pour interroger des traces .v2
- Un outil permettant de définir une méthode de transformation pour changer facilement de point de vue sur une trace
- La construction de processus d'analyse par programmation visuelle

1. Un langage naturel pour interroger des traces .v2

Ce document décrit les travaux réalisés par le LIRIS dans la suite du livrable 4.1. L'objectif a été d'ajouter des outils de visualisations aux résultats des requêtes faites par un utilisateur sur une trace d'un kTBS. Pour rappel, le langage naturel proposé en L4.1 permettait à un utilisateur non informaticien d'interroger un kTBS sans connaissance du langage de requête SPARQL.

Dans un premier temps, une fonctionnalité permettant l'auto-complétion a été ajoutée à l'éditeur existant. Cette fonctionnalité a pour objectif de faciliter la formulation de requêtes respectant la grammaire du langage naturel contrôlé. De plus, l'auto-complétion permet de choisir les type/attributs des *obsels* sans avoir à consulter le modèle de trace.

Pour déterminer le teneur de la requête d'un utilisateur, nous avons revu la méthode de génération de la requête SPARQL. Concrètement, les noms des variables SPARQL générées depuis le langage naturel possèdent un suffixe qui détermine s'il s'agit d'*obsels* ou d'indicateurs calculés. Ceci permet de proposer l'outil d'affichage des résultats adapté à la requête.

Nous proposons trois types d'affichages des résultats d'une requête :

- Un affichage tabulaire qui liste les résultats bruts.
- Un affichage sous forme de timeline, lorsqu'un utilisateur récupère des ensembles d'*obsels*.
- Un affichage sous forme de graphique, lorsque l'utilisateur récupère des indicateurs calculés.

Ces fonctionnalités ont été développées au sein de kTBS4LA :

<https://github.com/TaaabsElements/ktbs-for-la>

2. Un outil permettant de définir une méthode de transformation pour changer facilement de point de vue sur une trace

Parallèlement au langage naturel, un autre outil a été développé par le LIRIS pour permettre à un utilisateur de générer une méthode de transformation de manière graphique. Cette méthode de transformation permet de changer de point de vue sur une trace.

Concrètement, dans *KTBS4LA*, une trace est affichée sous forme de timeline (Figure 1), où chaque groupe d'*obsels* est représenté par un symbole.

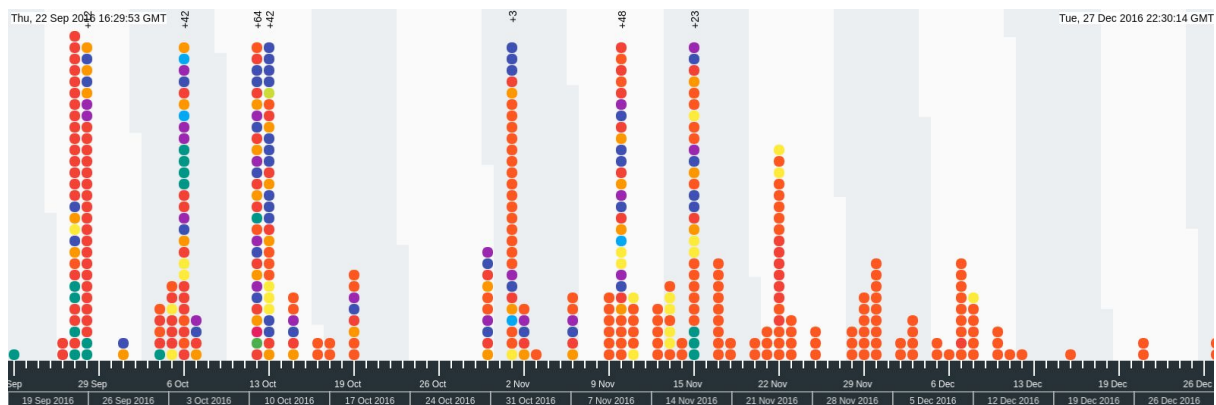


Figure 1. Exemple de timeline.

Les groupes d'obsels sont définis par une feuille de style qui correspond à une liste de règles (Figure 2). Cette liste de règles est une forme normale disjonctive de règles primaires. Une règle primaire correspond à une comparaison de valeurs sur un attribut d'*obsel* ou correspond à un type d'*obsel*. Chaque *obsel* de la trace sera affiché via le symbole de la règle qui lui correspond le plus.



Figure 2. Exemple de feuille de style.

On va pouvoir, par exemple, récupérer une vue de la trace en fonction des règles suivantes :

- Un tri par type d'*obsel*. Chaque règle a pour seule valeur discriminante le type des obsels.

- Un tri par intervalles de valeurs. Par exemple, on pourra grouper les *obse/s* par intervalles de scores obtenus.

À partir d'une feuille de style, on va pouvoir créer une méthode de transformation qui va matérialiser chaque règle en type d'obsel. Puis, dans un second temps, on pourra appliquer cette méthode sur une trace pour créer sa trace transformée. Cette trace correspond à une nouvelle vue sur la trace d'origine ou à une trace indicateur.

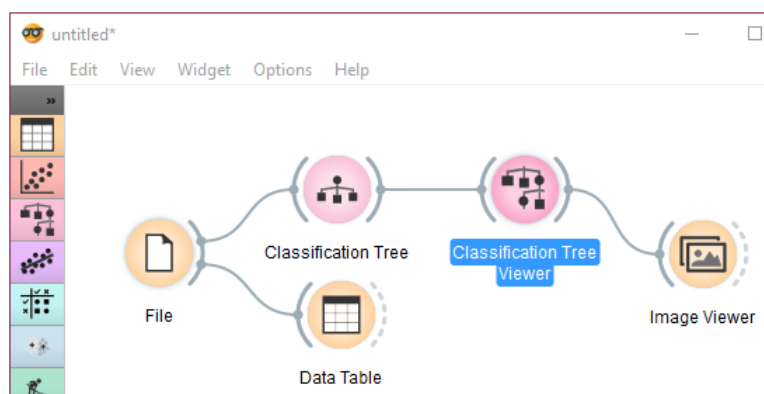
Ces fonctionnalités ont été développées au sein de ktBS4LA :

<https://github.com/TaaabsElements/ktbs-for-la>

3. Construction de processus d'analyse par programmation visuelle

La modélisation d'un processus d'analyse par un diagramme de flux de données et l'exploration des données à l'aide de la programmation visuelle sont des pratiques courantes dans le domaine de la fouille de données. Ces deux outils (diagramme de flux de données, programmation visuelle) permettent à des utilisateurs non spécialistes de travailler sur les données, ils sont adaptées à une utilisation par des enseignants et des décideurs.

Le logiciel Orange - Data Mining (<https://orange.biolab.si/>) est conçu ainsi : comme une boîte à outils d'opérateurs de traitement des données à associer en diagramme de flux de données par programmation visuelle. C'est la base sur laquelle UnderTracks s'est organisé.



Exemple (Orange - Data Mining) : à gauche les opérateurs, à droite le processus d'analyse

Le travail du projet d'UnderTracks (<https://undertracks.imag.fr>) a consisté à :

- faire un travail d'intégration de l'outil Orange - Data Mining avec la plateforme UnderTracks pour permettre l'accès via l'outil aux données sauvegardées sur le site UnderTracks et la sauvegarde des processus construits à l'aide de cet outil sur le site d'UnderTracks
- enrichir et structurer la base d'opérateurs disponibles dans l'outil pour une utilisation adaptée aux données visées (log temporelles d'interaction Machine-Apprenant). Plus de 50 opérateurs ont ainsi été définis, répartis en 3 groupes : Gestion, Analyse, Visualisation.

- structurer et améliorer le prototype des opérateurs définis pour permettre une utilisation des opérateurs plus ergonomique et une sauvegarde des paramètres d'application des opérateurs plus solide.

À ce jour (Juin 2017), plus de 100 processus d'analyse ont été construits sur la plateforme UnderTracks.

Exemples d'analyses :

- <https://undertracks.imag.fr/php/studies/study.php/qcml1allege2016>
- <https://undertracks.imag.fr/php/studies/study.php/copexchimie2010>