

*

0cm7ex plus 1ex minus .2ex1.5ex plus .2ex *

0cm4.5ex plus 1ex minus .2ex1.5ex plus .2ex *

*

*

Compte Rendu

Sonny Klotz - Jean-Didier Pailleux - Malek Zemni

*Interface de chargement, de contrôle
et d'analyse statistique des données
pour la constitution d'un graphe de flux*

22 mai 2017



Module *Projet*

Table des matières

1	Architecture de l'application	2
1.1	Organigramme et données échangées	2
1.2	Format du fichier CSV	3
1.3	Fonctionnalités des modules	3
2	Contraintes et Langages de programmation	6
3	Partie technique du projet	6
4	Points délicats	6
5	Organisation interne du groupe	6
6	Coût	6

Introduction

Ce document est le compte-rendu final de notre travail qui s'inscrit dans le cadre du module *Projet* de la licence informatique de l'UVSQ. Le sujet qu'on nous a remis a été proposé par l'entreprise DCbrain.

DCbrain développe des outils qui permettent de visualiser ce qui se passe sur les **réseaux physiques** afin de trouver les problèmes sur ces réseaux, les prédire dans le but d'optimiser les réseaux.

Les réseaux physiques peuvent être de plusieurs types. Ils peuvent être des réseaux industriels, de fluide ou de distribution, entre autres, c'est pourquoi les clients de DCbrain sont des groupes comme ERDF ou la SNCF.

Les données sont collectées à partir des réseaux physiques puis analysées grâce aux technologies du **Big Data**.

Ce projet s'inscrit dans un thème d'actualité et a pour but de fournir aux utilisateurs une application web qui se chargera de structurer les données, les analyser et les visualiser, mais aussi de fournir des API pour créer données statistiques synthétiques qui permettront d'alimenter leur base de données dans le but de mieux prévenir la présence d'anomalie sur les réseaux physiques.

L'architecture de l'application sera présentée en première partie. L'organigramme illustre le découpage de l'application en modules ainsi que les données qui transitent entre les différents modules. Dans cette première partie, des sections sont réservées pour décrire les fonctionnalités du produit ainsi que le type de données que l'application peut analyser.

On traitera dans une deuxième partie les contraintes sur le projet justifiant les langages de programmation choisis.

Enfin nous ferons dans deux parties distinctes un bilan technique et un bilan quant à l'organisation interne au sein du groupe.

Deux documents supplémentaires sont livrés avec le code source, un manuel utilisateur décrivant comment faire fonctionner notre application, et une documentation technique des modules de notre produit.

1 Architecture de l'application

1.1 Organigramme et données échangées

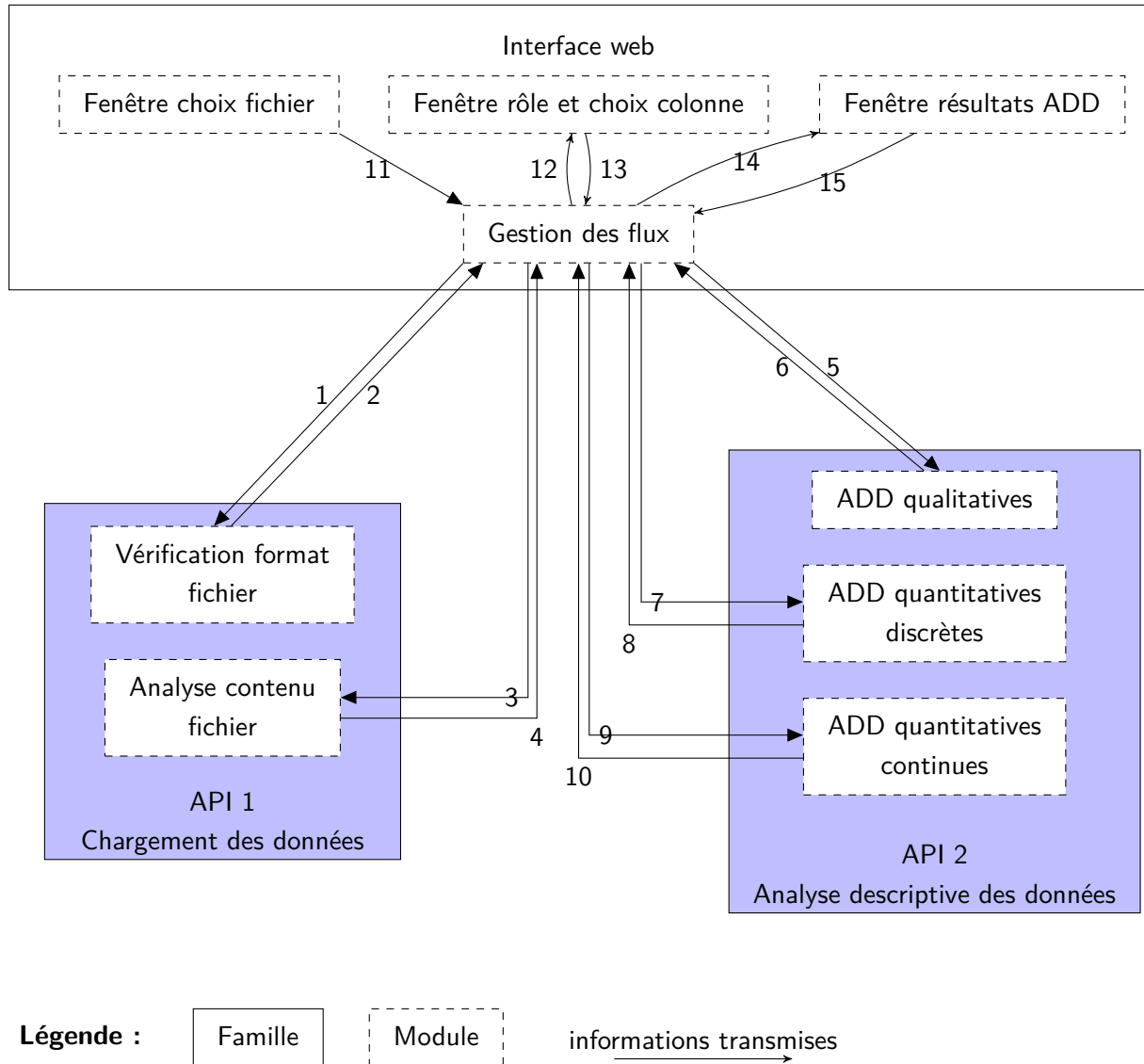


FIGURE 1 – Organigramme des différents modules du logiciel

Notes :

- (1) Fichier CSV : lancement de la vérification de son format
- (2) Code d'erreur : fichier OK ou ERREUR
- (3) Fichier CSV : lancement de l'analyse de son contenu
- (4) **structure 1** : contenant les données du fichier, le nombre de lignes et le nombre de colonnes (connus à partir de la taille de la structure)

structure 2 : contenant 3 informations sur chaque colonne : le type, le rôle et les positions des données erronées ou manquantes

(5) Ensemble de données de type qualitatif

(6) Erreur ou effectifs, effectifs cumulés, fréquences, fréquences cumulées, diagramme en secteur, histogramme

(7) Ensemble de données de type quantitatif discret

(8) Erreur ou indicateurs de tendance central, de dispersion et de forme, les anomalies, la distribution des données, un diagramme à moustaches

(9) Ensemble de données de type quantitatif continu

(10) Même données que (8)

(11) Chemin du fichier CSV importé

(12) Informations du (4) et ensemble de données contenu dans le fichier CSV

(13) Signal de validation du choix de la colonne, et noms des colonnes

(14) Envoi des résultats d'analyses de (6), (8) et (10)

(15) Signal de contrôle : demande d'exportation des résultats de l'ADD, analyse d'une autre colonne, ou importation d'un autre fichier

légende (non modifiée, on la met en accord avec le dernier produit quand on aura finit)

1.2 Format du fichier CSV

Le format du fichier est établi par DCbrain. Son contenu est décrit par des colonnes aux types prédéfinies :

Timestamp	Père	Enfant	Mesure 1	Mesure 2
January 1st 2017, 15 :00 :00.000	102	95	26644.235	176.253

Remarques : le graphe de flux utilisé par DCbrain pour analyser les réseaux de ses clients est orienté, d'où l'utilisation des noeuds *Père* - *Enfant* numérotés. Ceux-ci permettent de repérer de quelle connexion on parle.

Les colonnes *Mesure* renseignent sur les données mesurées sur une connexion à un temps donné, le nombre de colonne n'est pas limité et en pratique il ne dépassera pas la dizaine.

1.3 Fonctionnalités des modules

[style=unboxed,leftmargin=0.2cm]1. API 1 - Chargement des données

1. Module Vérification format fichier :

- Vérification de la possibilité d'ouverture du fichier.
 - Vérification du contenu : le fichier est un CSV contenant du texte brut non formaté. (pas de mise en forme avec des balises ou autres marqueurs)
 - Vérification de l'accessibilité du fichier en lecture.
2. Module Analyse contenu fichier :
- Ce module comprend deux fonctionnalités principales :
- Lecture du contenu du fichier CSV :
 On initialise une première structure (**structure 1**) pour y sauvegarder le contenu du fichier.
 On lit ligne par ligne des caractères du fichier. A chaque fois qu'on détecte un caractère de séparation (une virgule, un point-virgule ou une tabulation), on stocke les caractères lus (la donnée) dans la structure.
 Cette fonctionnalité fournit la **structure 1** contenant les données du fichier, le nombre de lignes et le nombre de colonnes (connus à partir de la taille de la structure).
 - Descriptions préliminaires des données de chaque colonne du fichier CSV :
 On initialise une deuxième structure (**structure 2**) pour y stocker des informations sur chaque colonne.
 On lit une par une les données de chaque colonne. On vérifie le type de la donnée en le comparant au type attendu. Si le type ne correspond pas, on signale dans la structure que la colonne contient une donnée erronée ou manquante (repérée par sa position dans la colonne). A la fin du parcours d'une colonne, on pourra lui attribuer un rôle/nom.
 Cette fonctionnalité fournit donc la **structure 2** contenant 3 informations sur chaque colonne : le type, le rôle et les positions des données erronées ou manquantes.
- Ce module va donc fournir les deux structures **structure 1** et **structure 2** décrites ci-dessus.

2. API 2 - Analyse descriptive de données

1. Module ADD qualitatives :
 - Calcul des effectifs, effectifs cumulés, fréquences et fréquences cumulées des données
 - Préparation des représentations graphiques : diagramme en secteur et histogramme
2. Module ADD quantitatives discrètes :
 - Statistiques : moyenne, quantiles, variance, écart-type, coefficients de symétrie et d'aplatissement de Fisher
 - Anomalies de Tukey : toute donnée hors de l'intervalle $[Q1 - 1.5*IQ; Q3 + 1.5*IQ]$, où Q1 et Q3 sont les quartiles, et IQ l'écart inter-quartiles.

- Préparation des représentations graphiques : fonction de distribution, fonction de distribution cumulative, boîte à moustaches
3. Module ADD quantitatives continues :
- Discrétisation de l'étendue (découpage en classe d'intervalles)
 - Calcul des quantiles pour le cas de données continues
 - Préparation des représentations graphiques : fonction de distribution cumulative

3. Interface web

1. Module Gestion des flux :
- Lancement de l'application et exécution continue.
 - Gestion des branchements : exécution normale ou arrêts pour cause d'erreur.
 - Interface entre les différentes fonctionnalités : communique les données nécessaires entre les modules.
2. Module Fenêtre choix fichier :
- Récupération un fichier CSV.
 - Validation du choix pour passer à la prochaine fenêtre (En renseignant son chemin dans le système de fichier, ou de la manière d'un Drag & Drop).
3. Module Fenêtre rôle et choix colonne :
- Affichera le nombre de lignes/colonnes contenu dans le CSV.
 - Affichera le titre du fichier.
 - Affichera un échantillon du contenu du CSV (environ les 1000 premières lignes) avec un système de scroll.
 - Affichage des lignes erronées (numéro de la ligne + contenu + type d'erreur).
 - Mise en place d'un système de navigation sous forme d'onglet (Onglet erreurs, onglet échantillon,...). Cela permettra d'éviter que la fenêtre contienne trop d'informations.
 - L'utilisateur devra sélectionner la colonne avec un clic, puis pourra lancer l'analyse sur celle-ci.
4. Module Fenêtre résultats ADD :
- Affichage des résultats d'analyse descriptive : informations statistiques de l'API 2 et représentations graphiques pour visualiser les données dans leur ensemble.
 - Une fonctionnalité de retour en arrière permet de sélectionner une nouvelle colonne sans relancer l'applet.

2 Contraintes et Langages de programmation

contraintes : -voir cahier des charges : applet, add, ... choix langages et frameworks : -python : justification + flask (voir cahier + ton expérience) -> intègre en plus un module de tests -html / css : easy on fait du web -javascript - jquery et AJAX : UX design tu peux parler vite fait, js donne interactivité côté client, ajax pour communiquer infos côté serveur entre autres et jquery qui permet de faire tout ça -Sphinx, génération documentation automatique (la doc flask est générée grâce à sphinx par ex), si tu sais pas on le fera plus tard, mais faut pas l'oublier
-> 2 pages (J-D)

3 Partie technique du projet

Quoi marche (comment ça marche - à l'aide de la partie avant) QUoi marche pas (pb et et resolution si trouvee)
1 page, plus tard

4 Points délicats

Justification des points qui divergent par rapport aux specs et organigramme de départ
1 page, plus tard à moins que t'ai déjà des idées

5 Organisation interne du groupe

Insérer la répartition des tâches (cahier des charges)
-> (J-D)
1 page
On expliquera plus tard comment ça s'est déroulé concrètement par rapport à ce qui était prévu.

6 Coût

Insérer le tableau des couts, ajouter une colonne (vide) "coût" le final
-> J-D 1 demi - page

Conclusion

1 demi-page, plus tard