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

22/05/2017

Module *Projet*

Table des matières

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 ce passe sur les **réseaux physiques** afin de trouver les problèmes sur ces réseaux, les prédire dans le but d'optimiser les réssraux.

Les réseaux phyiques 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 physique.

L'architecture de l'application sera présentée en première partie. L'organigramme illustre le découpage de lapplication 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'apllication 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 distintes 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.

========== Peut-être parler un peu plus du big data, aussi des techniques de graphes mining dans le travail de DC brain, pour présenter le fait que le réseau est assimilé à un graphe ==========

1 Architecture de l'application

1.1 Organigramme et données échangées

Ci-dessous se trouve l'organigramme du produit représentant la décomposition en modules de l'application, ainsi que les informations circulant entre ces modules :

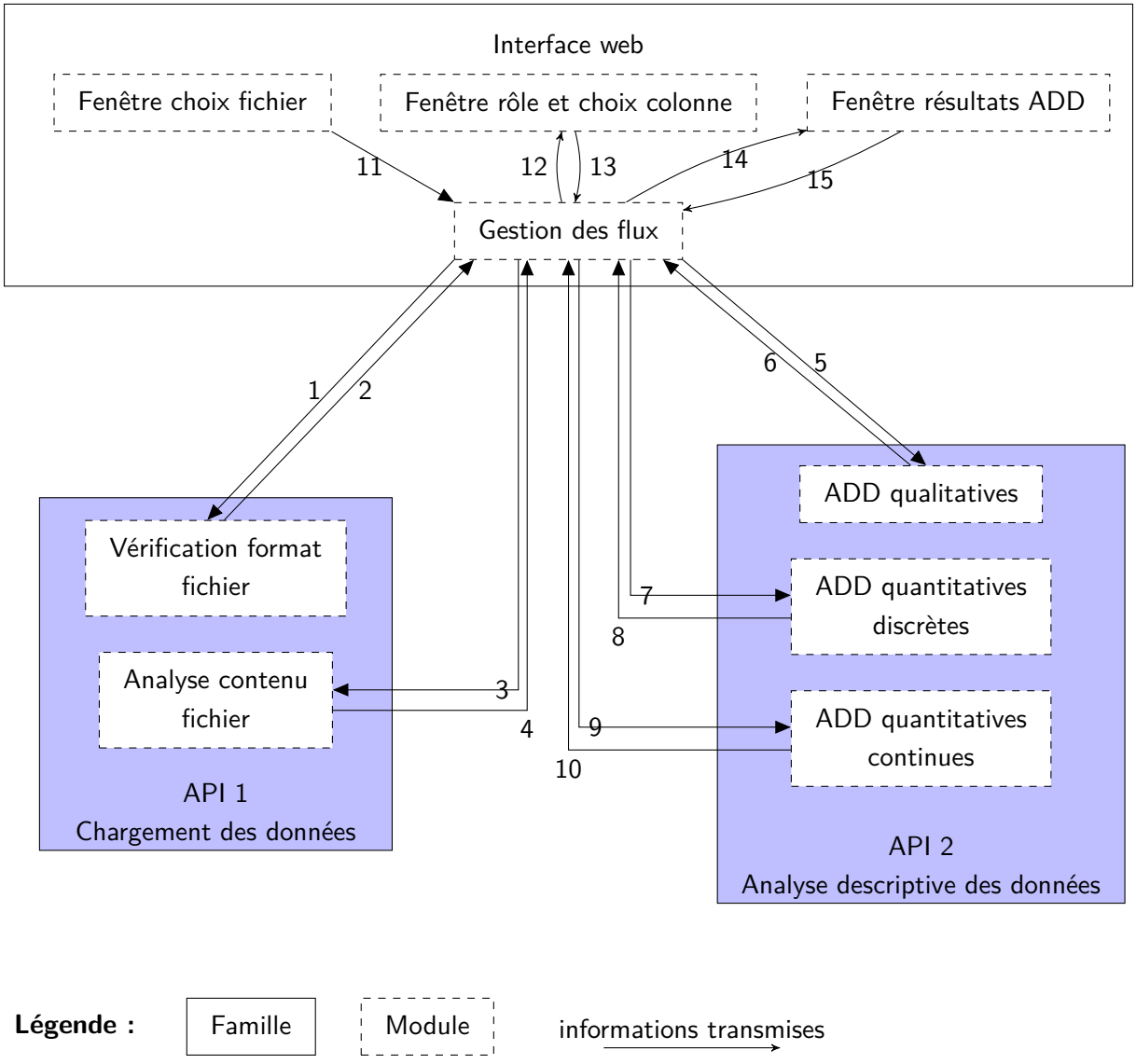

FIGURE 1 – Organigramme des différents modules du logiciel

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

========== Faites un tri dans vos fonctionnalités : réduire le plus possible, pour que tout rentre en 10 pages. Et mettre que les fonctionnalités du produit final.

Cette partie a aussi besoin d'un peu de mise en page, il y a trop de marge à droite a cause des enum + itemize + description) ==========

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 :
   — Gestion des branchements : exécution normale ou arrêts pour cause d'erreur.
   — Interface entres les différentes fonctionnalités : communique les données nécessaires entre les modules.
2. Module Fenêtre choix fichier :
   — Récupération un fichier CSV en renseignant son chemin en parcourant l'arborescence de fichiers, 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 erronés (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

2.1 Contraintes

Le développement de l'application s'accompagne de contraintes imposées. Celles-ci établie par DCbrain au commencement du projet. On peut en énumérer trois, qui sont les suivantes :

Contrainte	Fiche
1. Le produit doit fournir une application web	**Description :** notre produit sera une application fonctionnant sur un navigateur web, appelée ***applet***. **Justification :** assure une très grande portabilité et fournit à l'utilisateur une interface interactive. **Critère de satisfaction :** on peut lancer l'application sur un navigateur web.
2. Le produit doit être développé avec un langage de programmation compatible avec l'analyse de données	**Description :** le langage de programmation choisi doit inclure des bibliothèques qui permettent d'analyser les données (taches de data mining et de machine Learning). **Justification :** permettre une analyse de données efficace. **Critère de satisfaction :** on peut effectuer une analyse descriptive de données.
3. Le produit doit fournir une API d'analyse de données en sortie	**Description :** l'application doit intégrer des API d'analyse descriptive de données qui pourront être livrées en sortie au client. **Justification :** permettre une réutilisabilité des fonctionnalités majeures du produit. **Critère de satisfaction :** on peut exporter une API d'analyse de données en sortie.

2.2 Choix langages et frameworks

Les contraintes appliquées au projet ainsi que les exigences définies nous ont permis de fixer notre choix sur un langage de programmation : ***Python***.

Ce choix est justifié par le fait que ***Python*** soit compatible pour l'Analyse Descriptives de Données, puisqu'il est doté de nombreux modules permettant d'effectuer du calcul scientifique sur des données, donc l'analyse de données. De même, il est compatible avec le développement d'applications web, grâce aux nombreux frameworks disponibles.
Pour ce dernier, notre choix s'est porté sur ***Flask***, qui s'efforce d'être le plus simple possible pour une prise en main et le plus petit possible. À l'opposé de Django ou autres, il est donc plus abordable pour une première approche sur le développement d'applications web.

L'application devant être une applet intégrée dans une page web pour s'exécuter, le recours à des langages de balisage comme ***HTML*** pour la présentation et ***CSS*** pour la mise en forme de ces pages web étaient nécessaires.

Ensuite, pour permettre à l'application de faire communiquer les pages web avec un serveur web sans occasionner le rechargement de la page, nous avons eu recours à ***Ajax***. Il s'agit d'un concept de programmation web reposant sur plusieurs technologies comme le ***Javascript***, qui lui se charge d'établir la connexion entre la page web et serveur. ***Javascript*** est également employé pour rendre les pages web interactives. ***Jquery*** est quant à lui utilisé pour faire le lien entre ***HTML*** et ***Ajax***.

Enfin, pour la génération de la documentation des fonctionnalités du programme, on utilisera le logiciel ***Sphinx*** qui se charge de généré la docstring à partir des commentaires présents dans le code python.

3 Partie technique du projet

Quoi marche (comment ca 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 - (Peut-être rajouté les points délicats de flask/application web sur l'approche et tout ce qui va avec ?

5 Organisation interne du groupe

Ci-dessous se trouve le tableau de la répartition des modules pour chaque membre du groupe établie avant le commencement du développement :

Module	Malek	Sonny	Jean-Didier	Total
Gestion des flux			x	1
Fenêtre choix fenêtre			x	1
Fenêtre rôle et choix colonne	x			1
Fenêtre résultats ADD		x		1
ADD qualitatives			x	1
ADD quantitatives discrètes		x		1
ADD quantitative continues		x		1
Vérification format fichier	x			1
Analyse contenu fichier	x			1

1 page On expliquera plus tard comment ça s'est déroulé concrètement par rapport à ce qui était prévu.

6 Coût

Tableau indiquant l'estimation et le coût final en nombre de lignes de code, pour chaque module :

Module	Nombre de lignes	Justification	Coût final
Gestion des flux	15	Mise en forme du main et appel de l'application	
Fenêtre choix fichier	10 + 20	Fonctions pour : Drag& Drop + Système de fichiers	
Fenêtre rôle et choix colonne	5 + 20 + 10 + 10 + 20	Communication avec le module application + Affichage de l'interface+ lecteurs des valeurs + affichage des valeurs	
Fenêtre résultats ADD	10 + 3* 30	Envoie d'informations au module application + Construction des graphe pour l'ADD pour les 3 types d'analyse	
ADD qualitatives	20 + 20*3	Application des formules pour les calculs de fréquences et d'effectifs + calcul des valeurs pour la construction de 3 graphes	

ADD quantitatives discrètes	60 + 20*2	Application des formules attaché à l'analyse quantitative discret + calcul des valeurs pour la construction de 2 graphes	
ADD quantitatives continu	20 + 10 + 10 + 5 + 20*2	Parcours + choix précision classe d'intervalle + écriture + communication avec les modules+ calcul des valeurs pour la construction de 2 graphes	
Vérification format fichier	30	Ouverture fichier + vérification si ouverture en lecture + présence de texte formaté ou non	
Analyse contenu fichier	20 + 5 + 25 + 10	Recopie et vérification + initialisation de la structure+ Parcours du fichiers avec condition + Fonction pour donner nom et type de colonne	
Coût Total	**565**	**Estimation totale du coût**	

Conclusion

1 demi-page, plus tard