RAPPORTS

CETE du Sud-Ouest
DDAT / ESAD-ZELT

Janvier 2012

Note méthodologique
Base de données

Opération de recherche
SERRES

Ressources, territoires, habitats et logement
Énergie et climat
Développement durable
Infrastructures, transports et mer
Prévention des risques

Présent pour l'avenir

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Ministère de l'Écologie, de l'Énergie, du Développement durable et de la Mer

AFAQ ISO 9001

Centre d'Études Techniques de l'Équipement du Sud-Ouest

www-cete-sud-ouest.developpement-durable.gouv.fr

Historique des versions du document

Version	Date	Commentaire
1.0	Janvier 2012	Version 1

Affaire suivie par

Sébastien ROMON - DDAT/ESAD-ZELT
Tél. : 05 62 25 97 56 / Fax : 05 62 25 97 99
Courriel : sebastien.romon@developpement-durable.gouv.fr

Rédacteur

Sébastien ROMON - DDAT/ESAD-ZELT
Jean-Paul GARRIGOS – DDAT/ESAD-ZELT

Relecteur

Catherine BARTHE - DDAT/ESAD-ZELT

Liste des abréviations

CAN	Controller Area Network
CEESAR	Centre Européen d'Etudes de Sécurité et d'Analyse des Risques
CMS	Content Management System
CSV	Comma Separated Values
DIRCO	Dispositif de Recueil de données Comportementales
DOM	Document Objet Model
EuroFOT	European Large-Scale Field Operational Test on Active Safety Systems
FESTA	Field opErational teSt supporT Action
FOT	Field Operational Test
FCD	Floating Car Data
LAVIA	limiteur de vitesse s'adaptant à la vitesse autorisée
LAMP	Linux Apache MySQL PHP
LCR	Language de commande routier
MoCoPo	Mesure et mOdélisation de la COngestion et de la POllution
MVC	Modèle-Vue-Contrôleur
MRT	Mesure de Référence des Trajectoires
MTT	Métrologie des trajectoires et du Trafic
NGSIM	Next Generation Simulation
OdT	Observatoire de trajectoires
PREVER	Prévention et Évaluation des Risques
SERRES	Solutions pour une Exploitation Routière Respectueuse de l'Environnement et de la Sécurité
SGBDR	Système de gestion de bases de données relationnelles
SIREDO	Système informatisé de recueil de données
SQL	Structured Query Language
XML	Extensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

SOMMAIRE

1 - Avant-Propos

Cette note méthodologique sur les bases de données est rédigée dans le cadre de l'opération de recherche SERRES action 3 « Méthologies d'évaluation pluricritères du système routier ».
A travers cette note méthodologique sur les bases de données, vous trouverez des réflexions et des pistes sur la mise en forme des données et l'architecture de base de données à mettre en œuvre dans la perspective de partager des données. Nous avons repris des éléments de réflexions menées dans la cadre de l'opération MTT (Métrologie du Trafic et des Trajectoires).

Imaginons que pour tester différents modèles ou hypothèses, nous procédions à une vaste campagne de mesures utilisant un nombre varié de capteurs, d'appareils de mesures et d'expérimentateurs. Comment rendre accessible ce flot de données et d'informations à un public le plus large possible englobant des chercheurs, des statisticiens, des psychologues, des techniciens, des ingénieurs, des décideurs, des financeurs,... Des personnes aux profils divers seront intéressées par ce jeu de données, certains ne seront intéressés que par quelques indicateurs tel que l'impact du système en termes de sécurité routière, d'autres s'intéresseront à modéliser le comportements de l'usager vis à vis de l'infrastructure à partir de données fines.
Dans ce rapport, nous présentons dans un premier temps des réflexions et des pistes afin d'analyser et de préparer les données afin d'obtenir des jeux de données de qualité. Il importe de certifier les données dont nous connaissons les qualités, les défauts et le taux d'erreur. Enfin, dans un second temps, nous étudions différents types de plate-forme et de structure afin de rendre accessible la base de données à un maximum de profils d'utilisateurs.

2 - Contexte et enjeux de la base de données

L'opération de recherche SERRES héberge de nombreux projets de recherche dans lesquels de nombreuses données sont collectées. Ces projets n'ont généralement pas les mêmes finalités et objectifs mais les données sur lesquelles reposent ces projets sont similaires. Une base de données commune à l'ensemble des projets permettrait de centraliser et de partager des données entre partenaires. Pour que ces données puissent être réutilisées, il est nécessaire qu'elles soient bien documentées et traitées.

2.1 - Centralisation et partage de données

La centralisation et le partage de données ne consiste pas en une simple mise à disposition sur un serveur de données. D'une part, de simple fichiers de données brutes sans information sur le contenu, la précision et le contexte sont inutilisables. D'autre part, se posent des problèmes de confidentialité des données qui est généralement régie par les accords de consortium des projets.

2.2 - Réutiliser des données pour d'autres projets

Les campagnes de mesures sont généralement très coûteuses en moyens financiers et humains. Ces campagnes permettent de répondre à une problématique, de mesurer l'impact de systèmes ou d'infrastructure, de tester des hypothèses, de valider des modèles,... Afin de « rentabiliser » les

ressources mises en œuvre, les chefs de projet cherchent à collecter le maximum de données dans la perspective de tester ultérieurement de nouveaux modèles ou hypothèses qui exigent des données plus fines.

L'exploitation des données ne se limite pas à la date de fin d'un projet. Elles peuvent être utilisées ultérieurement pour affiner certains modèles ou tester de nouvelles hypothèses. Prenons l'exemple du projet LAVIA (projet français d'expérimentation et d'évaluation du limiteur de vitesse s'adaptant à la vitesse autorisée) initié dans les années 2000 et pour lequel les données collectées sur un panel de 90 conducteurs pendant 8 semaines sont encore utilisées par des chercheurs et des thésards.

Enfin, on peut recourir à des techniques de fouilles de données (data-mining) afin de chercher des modèles cachés, des liens entre variables, des comportements induits par le système.

La constitution de bases de données est une opération transversale à différents projets. C'est d'ailleurs l'un des objectifs d'une opération de recherche regroupant différents projets.

2.3 - Base de données documentée et certifiée

Pour qu'une base de données soit accessible par tous, elle doit être documentée. Il doit être spécifié la nature des différentes variables mesurées, l'unité de mesure, la précision des données, le taux estimé d'erreurs, les conditions d'expérimentation,...
Les données devront être nettoyées et être directement exploitables. Les méthodes, les algorithmes utilisés pour la transformation des données brutes en données de qualité devront être décrits et validés.

3 - Typologie des données de trafic

Nous distinguons deux principales catégories de données :

- les données recueillies par un appareil de mesure « bord de route » ;
- les données collectées par des systèmes embarqués à bord de véhicules.

Dans la première catégorie, les données concernent un flot de véhicules ou un flot de trajectoires. Les données enregistrées sont celles de véhicules dont les paramètres cinématiques sont en partie régis par des lois de poursuite des véhicules.

Concernant les données collectées à bord d'un véhicule sonde, les variables collectées sont des séries temporelles liées aux paramètres cinématiques du véhicule et au comportement de l'usager qui évolue en fonction du temps et de l'infrastructure.

Les finalités, l'exploitation de ces données, diffèrent selon la catégorie des données.

Il existe aussi d'autres types de données. Par exemple, les temps de parcours qui sont calculés entre deux points du réseau. Pour obtenir des temps de parcours de manière exhaustive, il est nécessaire d'identifier les véhicules en entrée et en sortie de réseau.

3.1 - Données recueillies par un appareil de mesure « bord de route »

L'utilisation d'un appareil de mesure « bord de route » permet de collecter de manière exhaustive des données de trafic en un point ou une section d'un réseau.

Nous pouvons distinguer deux types d'appareils de mesure :

- l'appareil de mesure qui va enregistrer des données en un point déterminé du réseau, par exemple une station SIREDO ;
- l'appareil qui va enregistrer des données sur une portion de voirie, par exemple une caméra vidéo ou un observatoire de trajectoires.

3.1.1 - Données mesurées en un point du réseau

Parmi ces données, nous retrouvons les données de trafic provenant de boucles électromagnétiques, un des capteurs les plus répandus sur le réseau français.
Les données collectées se rapportent à un flot de véhicules. Les véhicules ne sont pas indépendants les uns des autres. Ils peuvent rouler à « vitesse libre » lorsque le trafic est faible ou être contraints par la circulation et doivent adapter leur vitesse en fonction de l'environnement.
Les données individuelles peuvent être utilisées pour caractériser des flux de véhicules. Par exemple, lorsque nous interdisons aux PL de doubler, nous observons le formation de « trains de poids-lourds ». On cherchera alors à caractériser le train de poids-lourds en fonction des paramètres des PL qui le constituent.

3.1.2 - Données mesurées sur une section du réseau

Les données mesurées sur une section du réseau permettent de suivre des véhicules sur une section telle qu'une section de rocade avec une voie d'entrecroisement ou d'insertion, ou un virage.
On peut par exemple recourir à des campagnes de mesures vidéo que l'on va analyser et exploiter afin d'extraire des trajectoires de véhicules.
Les méthodes d'analyse pour ces données diffèrent de celles utilisés pour les données individuelles mesurées en un point du réseau. Ici, chaque véhicule est représenté par sa trajectoire constituée de points (x,y,t) éventuellement associés à des vitesses et accélérations.

3.2 - Données collectées par des systèmes embarqués

Les systèmes embarqués enregistrent les paramètres cinématiques et le comportement de l'usager. Les paramètres cinématiques tels que la vitesse, l'accélération sont des séries temporelles. Chaque observation ou enregistrement correspond à un état d'un système composé du véhicule et du conducteur, qui interagit avec l'environnement extérieur. Ce sont des données très riches en informations. Avant l'exploitation, il est nécessaire de les analyser et d'extraire les informations dont on a besoin. On peut pour cela recourir à des techniques dites de « fouilles de données ».

4 - Analyse des données

Le but de l'analyse des données décrite dans ce chapitre est de nettoyer, d'enrichir un jeu de données afin d'obtenir au final une base de données de qualité, certifiée et documentée.
Les données brutes de capteurs ou d'appareils de mesure ne sont généralement pas exemptes d'erreurs de mesure. Avant l'exploitation des données, il est nécessaire d'analyser les données afin de repérer les éventuelles erreurs de mesure et les données manquantes. De nos jours, le volume de données collectées lors d'une campagne de mesure peut être considérable et il est nécessaire d'utiliser des outils d'analyse de données.

Cette partie à été élaborée à partir de la méthodologie FESTA et du livre Data Mining et statistique décisionnelle [Tufféry,2007].

4.1 - Mise en forme des données

La première étape lorsque nous disposons d'un jeu de données est de les mettre en forme afin qu'il soit lisible par un logiciel de traitement de données. Pour un maximum de comptabilité, il suffit généralement de mettre les données sous la forme de tableau, par exemple au format CSV (Comma-Separated Values). La plupart des logiciels de statistique permettent d'importer des fichiers de données tabulaires.
On peut aussi choisir d'importer directement les données dans des systèmes de gestion de bases de données.

Il arrive fréquemment que la mise en forme des données soit plus complexe notamment lorsque les données ne sont pas collectées avec la même fréquence d'acquisition ou ne sont pas horodatées de la même manière. Par exemple, les paramètres d'un véhicule provenant du bus CAN ont des fréquences différentes variant de 1Hz à 100 Hz. Une alternative aux fichiers tabulaires peut être alors le format XML (Extensible Markup Language) décrit plus loin dans le document.

Enfin, les données sous forme de séries temporelles peuvent ne pas convenir à certains types d'études et il est parfois nécessaire d'effectuer une transformation des données en séries longitudinales, par exemple lorsque nous calculons des profils de vitesse le long d'un itinéraire.

D'après l'exemple ci-dessous, nous pouvons extraire des données individuelles en utilisant la commande AI du langage LCR.
Le fichier brut obtenu est :

```
0 18:13:00:64 V=0077 I=0025 L=0073 T=0346 D=0070 K=0002 P=     N=0002
5 18:13:04:23 V=     I=1536 L=     T=0229 D=     K=0001 P=     N=0002
0 18:13:05:65 V=0089 I=0048 L=0038 T=0156 D=0102 K=0001 P=     N=0002
0 18:13:08:64 V=0078 I=0026 L=0058 T=0271 D=0064 K=0001 P=     N=0002
0 18:13:13:76 V=0111 I=0049 L=0040 T=0132 D=0104 K=0001 P=     N=0002
0 18:13:21:55 V=0092 I=0070 L=0174 T=0681 D=0214 K=0010 P=     N=0005
0 18:13:26:61 V=0086 I=0048 L=0042 T=0178 D=0122 K=0001 P=     N=0002
0 18:13:29:35 V=0079 I=0025 L=0034 T=0158 D=0058 K=0001 P=     N=0002
0 18:13:44:81 V=0090 I=0152 L=0043 T=0173 D=0320 K=0001 P=     N=0002
```

commande AI

Ce fichier ne peut être directement utilisé dans un logiciel de statistique. Une étape de mise en forme des données s'impose. Il peut être transformé de la manière suivante :

```
id;station_id;timestamp;lane;speed;length;tiv;div;occ_time;axles
1;1;2010-10-11 18:13:00.64;0;77;73;25;70;346;2
2;1;2010-10-11 18:13:04.23;5;;;;;229;2
3;1;2010-10-11 18:13:05.65;0;89;38;48;102;156;2
4;1;2010-10-11 18:13:08.64;0;78;58;26;64;271;2
5;1;2010-10-11 18:13:13.76;0;111;40;49;104;132;2
6;1;2010-10-11 18:13:21.55;0;92;174;70;214;681;5
7;1;2010-10-11 18:13:26.61;0;86;42;48;122;178;2
8;1;2010-10-11 18:13:29.35;0;79;34;25;58;158;2
9;1;2010-10-11 18:13:44.81;0;90;43;152;320;173;2
```

données mises en forme

Dans l'exemple ci-dessus, nous avons enrichi les données d'origine en rajoutant un identifiant pour chaque véhicule, l'identifiant de la station de mesure et la date de la mesure. Nous remarquons la présence de données manquantes dans le fichier brut. La station SIREDO a détecté un véhicule sur la boucle numéro 5 mais n'a pas pu déterminer ni sa vitesse et ni sa longueur.

4.2 - Analyse de la qualité des données

Avant d'analyser et d'exploiter les données, il est nécessaire de détecter les données manquantes. Il ne faut avoir une confiance aveugle dans les jeux de données brutes même s'ils proviennent d'équipements certifiés et/ou homologués.

L'analyse de la qualité passe par deux principales étapes :

- l'évaluation et la quantification des données manquantes ;
- le contrôle des valeurs et unités de mesure.

4.2.1 - Détection des doublons

Les fichiers brutes contenant les données peuvent contenir des données en doublons. Pour détecter des doublons, on peut par exemple imposer des contraintes d'unicité sur certaines variables ou champs, par exemple, deux véhicules ne peuvent pas passer en même temps sur un capteur d'une

même voie. Il convient de comprendre d'où viennent les doublons. La répétition d'un même donnée peut résulter de fréquences d'acquisition différentes entre appareils de mesure. Un GPS peut être configuré pour délivrer une position toutes les secondes et il est alors cohérent que les données GPS prennent des valeurs consécutives identiques si la fréquence d'acquisition d'autres paramètres tel que l'accélération et la vitesse ont une fréquence plus élevée. Une donnée peut apparaître plusieurs fois dans un jeu de données parce qu'elle appartient à deux ensembles. Par exemple, un véhicule sera présent dans un fichier contenant les données individuelles d'une voie de circulation et sera aussi dans le fichier contenant les données d'en sens de circulation regroupant plusieurs voies.
La présence de valeurs identiques consécutives peut être le résultat d'un dysfonctionnement d'un capteur ou d'un appareil de mesure. On peut le vérifier en traçant par exemple la distribution des valeurs prises par la variable ou en analysant l'évolution de la variable au cours du temps.

4.2.2 - Évaluation et quantification des données manquantes

4.2.2.a - Origine des données manquantes

Les données manquantes peuvent avoir diverses origines dont :

- des erreurs de manipulation ;
- la défaillance de capteurs ;
- le masquage de capteurs ;
- des bugs d'application ;
- des erreurs d'inattention des individus ;
- la réinitialisation d'un instrument de mesure ;
- la non réception de signaux,...

Les données manquantes peuvent être flagrantes (absence de fichiers) ou passer inaperçues (les données d'un capteur sont noyées dans un flux d'information).

4.2.2.b - Détection des données manquantes

Il est nécessaire de contrôler la fréquence d'acquisition des données recueillies afin de détecter le manque d'enregistrements. Par exemple, si un véhicule traceur circule sous un tunnel, le GPS ne délivrera plus de données de positionnement et de vitesse.

Il arrive que, en l'absence de données de mise à jour, certains appareils de mesure n'indiquent pas que la donnée est manquante utilisent des valeurs stockées en mémoire « périmées ». Par exemple, un système embarqué ne sachant pas qu'un de ces capteurs est défaillant ou ne peut délivrer correctement un signal fournit une valeur constante correspondant à la dernière valeur transmise par le capteur. Pour détecter ces anomalies, nous pouvons tracer les distributions des variables, repérer des points singuliers et regarder de plus près les enregistrements consécutifs pour lesquels la variable n'évolue pas.

La non-présence de certaines données peut avoir des répercussions sur d'autres variables ou valeurs. On vérifiera par exemple que lorsque le module de positionnement ne peut plus se localiser, l'application d'aide à la conduite n'affiche pas d'informations erronées au conducteur.

Enfin, une attention particulière sera portée à la conversion automatique des données. Il peut en effet arriver qu'une donnée non disponible soit convertie par la valeur zéro, ce qui peut engendrer des erreurs de calcul.

4.2.2.c - Décodage / codage des valeurs manquantes

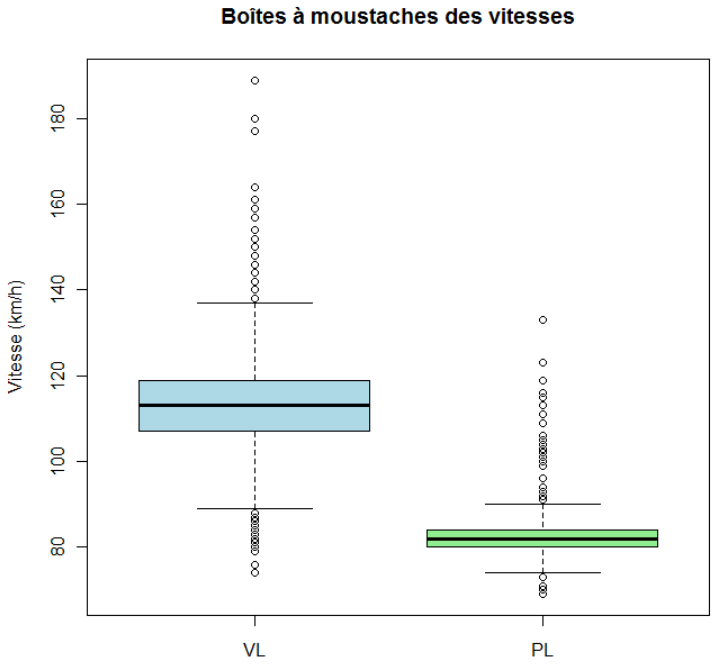
Les appareils de mesure n'utilisent pas les mêmes codes pour les données manquantes.
Des logiciels vont coder la valeur manquante par une valeur aberrante tel que -1, 255, 65535 qui ne correspond pas à une valeur incluse dans la plage de fonctionnement du capteur. Dans ces cas là, il faudra remplacer ces valeurs par la valeur interne d'indisponibilité du logiciel utilisée pour le traitement et l'exploitation des résultats. Ce recodage évite des erreurs de calculs puisque le logiciel de statistique, de base de données ignorera les données non disponibles dans le calcul d'indicateurs.

Certains logiciels codent les valeurs manquantes par des chaînes de caractères tel que « NA » (not available) ou « NULL ». Ces chaînes de caractères peuvent poser des problèmes lors de l'importation des données. On veillera à remplacer ces chaînes de caractères par le code d'indisponibilité du logiciel de traitement des données.

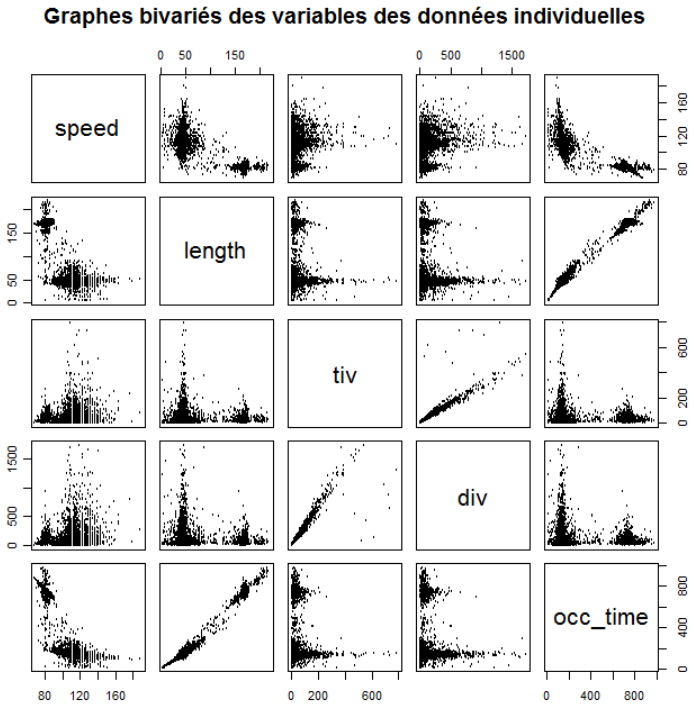
4.2.3 - Contrôle des valeurs des variables

Pour contrôler les valeurs et les unités de mesures, nous pouvons calculer pour chaque variable des éléments statistiques tels que le minimum, le maximum, la moyenne, l'écart-type et les quantiles les plus significatifs : 1% (1er centile), 10% (1er décile), 25% (Q_1, 1er quartile), 50 % (Q_2, médiane), 75 % (Q_3, 3ème quartile), 90 % (dernier décile), 99 % (dernier centile).
Nous pouvons visualiser les distributions des variables en utilisant des boîtes à moustaches qui représentent de manière synthétique les principales caractéristiques d'un échantillon. Elles mettent en évidence les valeurs atypiques appelées aussi « outliers » qui correspondent aux valeurs inférieures à Q_1-1.5*(Q_3-Q_1)et supérieures à Q_3+1.5*(Q_3-Q_1).

Dans l'exemple ci-dessus, nous avons tracé les boîtes à moustaches des données vitesses en fonction du type de véhicule.



Le graphique ci-dessus met en évidence des vitesses atypiques supérieures à 160km/h entre 16h et 19h sur l'autoroute A10 limitée à 110 km/h. Ce sont des données extrêmes qui ne sont pas aberrantes. Concernant les poids-lourds, des vitesses supérieures à 110 km/h nous laissent suggérer qu'il faudrait mieux classer ces véhicules dans la catégorie des véhicules légers. Les véhicules ont été classés selon la longueur. Il aurait été judicieux de tenter une classification hiérarchique des véhicules selon différents paramètres tels que la vitesse et la longueur comme le laissent suggérer les graphes bivariés ci-dessous.



Sur le graphe ci-dessus, nous avons représenté les graphes bivariés des variables des données individuelles. Il met clairement en évidence les corrélations entre certaines variables et on détecte des modes correspondant aux VL et PL.

4.2.3.a - Détection des données aberrantes

Lorsque nous détectons des valeurs aberrantes, plusieurs choix s'offrent à nous :

- supprimer les enregistrements ;
- conserver les données mais en ne les prenant pas en compte lors de l'analyse et l'exploitation des données ;
- conserver les données sans rien faire ;
- corriger les données aberrantes ;

suppression d'enregistrements

Il s'agit de la solution la plus radicale lorsque les observations sont difficilement exploitables. Dans ce cas, il faut vérifier que la taille de l'échantillon est encore pertinente et que le nombre d'enregistrements supprimés n'est pas trop élevé.

Correction d'enregistrements

Nous pouvons essayer de corriger les données erronées en utilisant d'autres sources de données. Par exemple, lorsqu'un véhicule circule dans des « canyons » urbains, nous pouvons corriger les erreurs de positionnement en utilisant des algorithmes de map-matching et d'autres sources de données

telles qu'une carte vectorielle du réseau et les données cinématiques du véhicule.

Nous pouvons aussi remplacer les valeurs aberrantes par une valeur imputée statistiquement, par exemple la moyenne ou la médiane.

4.2.3.b - Détection des valeurs extrêmes

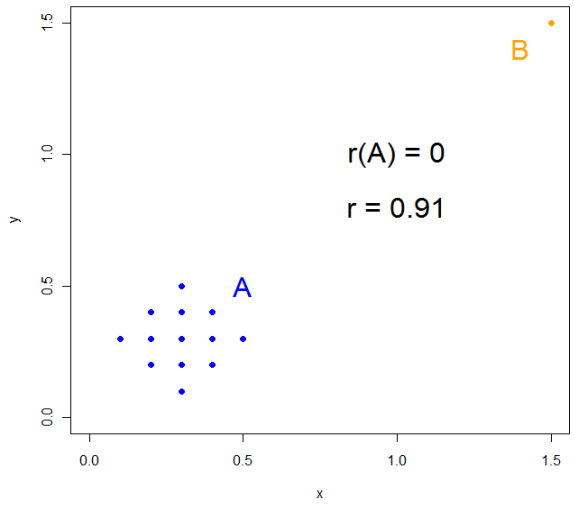
Les valeurs extrêmes ne sont pas des erreurs de mesure. Ces données ne sont pas à retirer de la base de données mais il est cependant utile de les identifier. En effet, ces données peuvent fausser certains tests et modèles statistiques non robustes aux valeurs extrêmes.
Dans le domaine de l'accidentologie, les valeurs extrêmes, les comportements extrêmes sont très recherchés afin de détecter des situations de quasi-accident, des états et des trajectoires limites voire défaillantes.

D'après [Tufféry,2007] plusieurs choix s'offrent à nous afin d'atténuer l'effet des valeurs extrêmes :

- écarter les données extrêmes ;
- découper la variable en classes ;
- « winsoriser » la variable.

Nous pouvons écarter certaines données, certains jeux de données ou individus lorsque les valeurs extrêmes peuvent fausser les résultats de tests ou modèles sensibles aux données extrêmes.

Par exemple, d'après le graphique ci-dessous, les deux séries x et y de l'ensemble A ne sont pas corrélées (le coefficient de corrélation est nul). Si on rajoute le point B, point extrême, le coefficient de corrélation vaut alors r_{A+B}=0.91. Les données semblent corrélées à cause du point B alors qu 'elles ne le sont probablement pas.



En découpant la variable en classes, les données extrêmes sont neutralisés en les plaçant dans la première et la dernière classe.

La « winsorisation » de la variable consiste à remplacer les valeurs de la variable supérieures au 99ème centile par ce centile et les valeurs inférieurs au 1er centile par celui-ci. Dans l'exemple précédent, le 99ème centile des vitesses des véhicules vaut 148km/h. La winsorisation des valeurs extrêmes consiste alors à remplacer les vitesses supérieurs à 148 km/h par cette même valeur.

En ingénierie du trafic, la vitesse de référence est le 85ème centile des vitesses noté V85. Cet indicateur de vitesse qui caractérise la vitesse des usagers sur une infrastructure permet de s'affranchir des vitesses extrêmes.

4.2.3.c - Unités de mesure

Il convient d'uniformiser les unités de mesure afin d'éviter des erreurs de calcul. Il est en effet pas rare qu'un capteur tel que le GPS fournisse une vitesse exprimée en $m.s^{-1}$ alors qu'un autre capteur fournira une vitesse exprimée en km/h.
Une attention particulière sera aussi donnée sur les horodates des données puisque les capteurs et appareils ne travaillent pas forcément avec les mêmes référentiels :

- temps UTC ;
- l'heure locale ;
- temps GPS ;
- temps UNIX (nombre de secondes écoulées depuis le 1er janvier 1970).

Enfin lors de la conversion, l'importation et l'exportation des données, il faut prendre soin à ne pas dégrader la précision des données. Par exemple, les coordonnées d'un point GPS doivent comporter au minimum 7 décimales pour une précision de l'ordre du mètre.

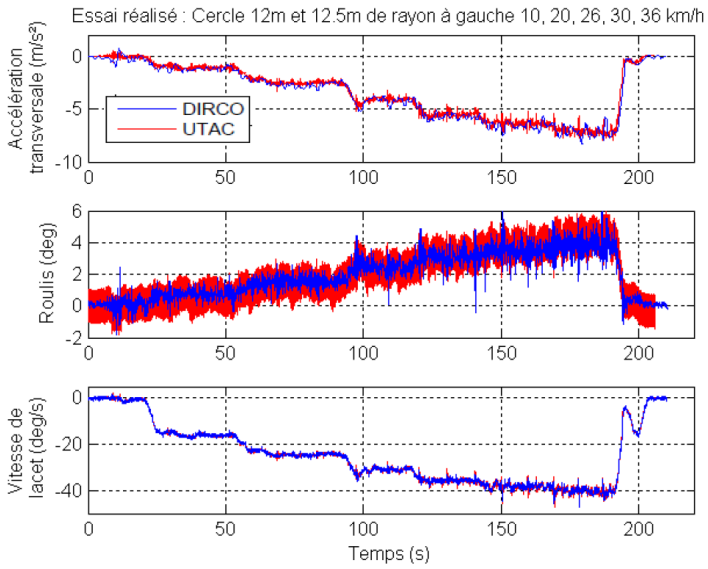
Les données CAN

Les constructeurs automobiles sont généralement réticents à donner des informations sur les données circulant sur le bus CAN.
On peut par retro-ingénierie deviner certains paramètres tel que la vitesse, l'angle au volant, le régime moteur, le rapport de boîte,... mais il est préférable de disposer de la cartographie constructeur du bus CAN.

4.3 - Traitement des données

4.3.1 - Lissage et interpolation des données

Les données peuvent présenter une variabilité trop élevée qu'il est alors nécessaire de lisser. Cette forte variabilité peut être en partie due à une mauvaise précision de l'appareil de mesure.



Le graphique ci-dessus est extrait du rapport d'évaluation du DIRCO (DIspositif de Recueil de donnée Conducteur), système embarqué permettant de recueillir les paramètres cinématiques d'un véhicule.
Nous notons que les 3 paramètres mesurés (accélération transversale, le roulis et la vitesse de lacet) sont très bruitées. Il semble nécessaire de lisser les données afin d'éliminer le bruit.

Pour interpoler, estimer ou lisser, nous pouvons utiliser les méthodes suivantes :

- interpolation linéaire ;
- interpolation polynomiale ;
- interpolation par polynômes locaux ;
- estimateur de Nadaraya-Watson ;
- moyenne mobile ;
- interpolation par splines,...

Le choix de la méthode d'interpolation dépend de la nature des données. Les estimateurs dépendent généralement d'un paramètre de lissage h que l'on peut déterminer par minimisation de critères tel que le critère de validation croisée.

4.3.2 - Calcul de nouvelles variables à partir des données brutes

Nous pouvons être amenés à calculer de nouvelles variables à partir des données existantes. Il convient de vérifier que la précision des données en entrée permet de calculer de nouveaux paramètres. Par exemple, la précision des données d'accélération et de vitesse calculées à partir de données vidéo est conditionnée par la résolution des images vidéo et la fréquence d'acquisition des images.
Le calcul de l'abscisse curviligne d'un véhicule le long d'un trajet ou itinéraire peut être fait en utilisant différentes sources de données : les positions et vitesses GPS, les variables odomètre et vitesse provenant des paramètres CAN, une cartographie numérique,...

4.3.3 - Annotation d'événements

Certains événements ou conditions extérieures à l'expérimentation influent sur les résultats attendus. Il convient de les incorporer dans la base de données puisque ce sont des variables explicatives qu'il faut prendre en compte lors de l'exploitation et le calcul d'indicateurs.
Parmi ces événements, nous trouvons :

- les conditions météorologiques (pluie, vent, neige, verglas, brouillard, soleil rasant,...) ;
- les réductions de vitesses (pics d'ozone, travaux,...) ;
- les incidents ;
- les accidents ;
- les travaux ;
- les grandes manifestations (match de football, grèves,...) ;
- les congés scolaires ;

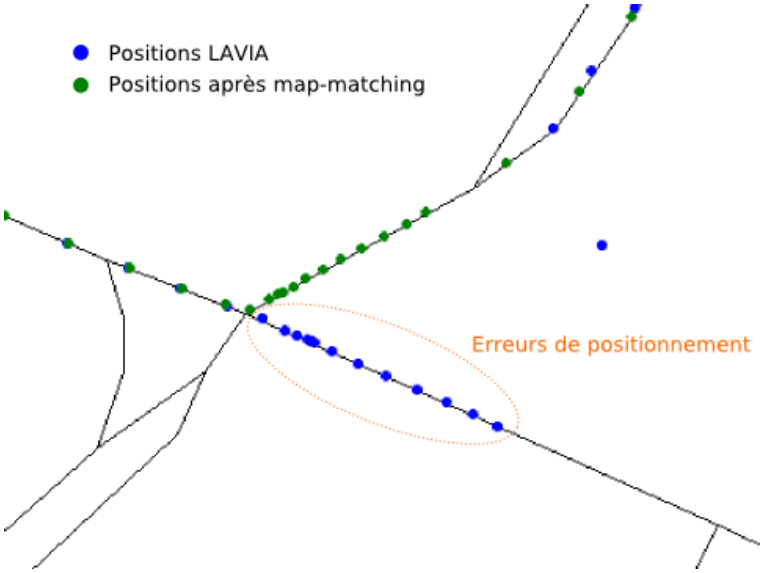
Il faudra écarter certaines situations lorsque par exemple, nous souhaitons analyser la congestion récurrente des heures de pointe du matin et du soir.

4.3.4 - Traitement des données GPS

Les positions du véhicule peuvent être brutes, c'est à dire délivrées directement par le système GPS ou être localisées par un algorithme de mapmatching temps réel sur un réseau. Dans les deux cas, il est souhaitable d'appliquer un algorithme de mapmatching temps différé sur les positions GPS.
Pour calculer la position d'un véhicule, un algorithme de mapmatching prend en entrée plusieurs sources de données : les positions GPS, une cartographie vectorielle et éventuellement des données de vitesse, de direction et l'odomètre du véhicule.

Lors du traitement des données GPS, on peut aussi calculer l'abscisse curviligne du véhicule le long d'un itinéraire afin de tracer des profils de vitesse, d'accélération,... Cette abscisse curviligne est très utile lorsque nous souhaitons transformer des données temporelles en données longitudinales.

Dans l'exemple ci-dessous, le système LAVIA qui dispose de son propre logiciel de mapmatching temps réel commet des erreurs de positionnement (points en bleu). Un algorithme temps différé de mapmatching qui travaille de manière globale sur l'ensemble du trajet corrige ces erreurs (points en vert).



Actuellement, les GPS « low cost » ont une précision qui varie entre 3 et 5 mètres. Cette précision est généralement suffisante pour des applications d'aide à la conduite telles que les indications de la vitesse réglementaire et de zones de vigilance. Par contre, elle est insuffisante lorsque l'on veut localiser le véhicule au niveau de la voie de circulation.

Pour localiser un véhicule au niveau de la voie de circulation, il faut utiliser des GPS bi-fréquences ou différentiels.

4.4 - Calcul d'indicateurs globaux

Lors d'expérimentations, nous disposons généralement de données très fines. Il est nécessaire de calculer des indicateurs globaux afin de caractériser une trajectoire, un trajet, une situation de trafic, des situations de conduite,...
Dans le cadre de l'élaboration de base de données, ces indicateurs globaux permettent de classifier nos données pour aider l'utilisateur dans le choix de jeux de données.

4.5 - Vérification de la synchronisation des données

Il est nécessaire de vérifier avant l'expérimentation et après l'expérimentation que les appareils de mesure sont bien synchronisés. Les équipements terrain reliés à un CIGT sont généralement synchronisés à un serveur de temps et les équipements embarqués sont synchronisés par l'intermédiaire d'un GPS.

Il est important de vérifier a posteriori, que les jeux de données sont bien synchronisés. Des dérives temporelles peuvent être préjudiciables lors du traitement des données notamment lorsque nous cherchons à fusionner des jeux de données.

Dans le projet MoCoPo, où un recueil vidéo est effectué à l'aide d'un drone, la synchronisation de la vidéo est validée en comparant les trajectoires des véhicules avec les données GPS des véhicules sondes.

5 - Les données

La base de données de l'opération SERRES pourra héberger les données suivantes :

- des données de trafic (données individuelles et agrégées) ;
- des données du projet MoCoPo ;
- des données de trajectoires (opération de recherche MTT) ;
- des données de véhicules traceurs.

5.1 - Données de trafic

Durant l'opération MTT, le CETE IdF avait établi un catalogue de l'ensemble des données individuelles collectées par les différents groupes utilisant ce type de données. Dans ce catalogue sont recensés la nature des données collectées, les appareils de mesure utilisés, les utilisations qui sont faites de ces données ainsi que les traitements et les exploitations réalisés sur ces données.

5.1.1 - Données individuelles

Les données individuelles sont des données relatives à des véhicules circulant sur une section courante recueillies par différents capteurs implantés dans la chaussée.

Les principaux attributs d'une donnée individuelle recueillie sur une section courante sont :

- l'heure de passage du véhicule ;
- la voie de circulation ;
- la vitesse du véhicule ;
- la longueur et le type du véhicule ;
- les temps et distances intervéhiculaires ;
- les attributs relatifs au nombre d'essieux et poids du véhicule.

5.1.1.a - Attributs de base d'une donnée individuelle

Les données individuelles peuvent être stockées dans une base de données de la manière suivante :

Attribut	Type de données	Commentaires
Identifiant	Entier (auto-incrément)	Clé primaire Permet d'identifier de manière unique l'enregistrement
Identifiant de la station de comptage	Entier	Clé étrangère

Horodate	Date/Heure	'yyyy-mm-dd hh:mm:ss.nnn' ou 'yyyy-mm-dd hh:mm:ss.nnn+hh'
Voie de circulation	Entier	Clé étrangère
Vitesse instantanée	Entier	En km/h
Longueur	Entier	En décimètres

5.1.1.b - Attributs déductibles d'autres champs et données

Les attributs qui peuvent être calculés à partir des données de base décrites précédemment peuvent ne pas être stockés dans la base. Il est toutefois possible de développer des fonctions qui calculent ces attributs à la demande de l'utilisateur ou d'un client se connectant à la base de données.

Voici une liste non exhaustive d'attributs calculés par l'application :

- temps de présence du véhicule ;
- distances inter-véhiculaires (TIV et DIV) ;
- classe de silhouette du véhicule ;
- contexte de circulation (débit local, données agrégées 6 minutes) ;
- conditions de trafic (état du trafic, classe de congestion, indicateurs de congestion,...).

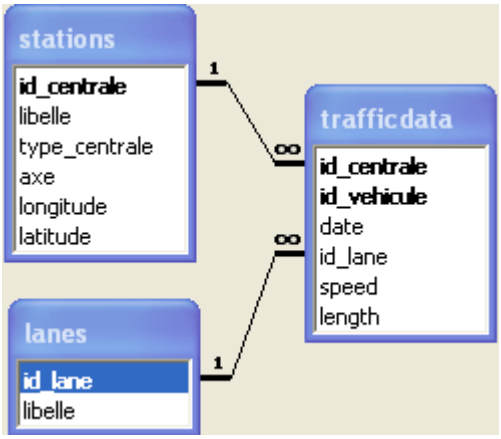
5.1.1.c - Stockage des données individuelles

Pour stocker des données individuelles dans un SGBDR, la base de données doit comprendre au minimum 2 tables :

- la table des stations de comptage
- la table des données individuelles

La table des stations de comptage contiendra les caractéristiques de la station de comptage ainsi que celles de la section courante où sont implantées les boucles électromagnétiques.

Concernant les données individuelles, il se peut qu'à la place d'une seule table de données, il y en ait plusieurs de tailles moindres. Cela dépendra du volume de données stockées dans la base, de la puissance du serveur hébergeant le SGBDR et des performances attendues.



Relation entre les tables

5.1.2 - Données agrégées de trafic

Les données agrégées de trafic sont des variables globales permettant de quantifier le trafic sur une section ou un point du réseau.
Les 3 variables les plus couramment utilisées et exploitées sont :

- le débit Q ;
- la moyenne harmonique des vitesses ;
- le taux d'occupation TO.

D'autres variables sont utilisées pour caractériser le trafic. Par exemple le vitesse V85 correspondant au 85ème centile des vitesses.

Les données agrégées sont calculées à partir des données individuelles. L'agrégation des données s'effectue par unité de temps (1 minute, 6 minutes, 1 heure, 1 journée) et en regroupant les voies de circulation par canaux ou sens de circulation.

Elles ont été très largement étudiées durant ces dernières décennies. A partir de ces données, on est capable de calculer des indicateurs de congestion, des temps de parcours, d'estimer l'étendue et la dureté d'une congestion,...

5.2 - Trajectoires des véhicules

5.2.1 - Les données MOCoPo

Le projet MOCoPo (Mesure et mOdélisation de la COngestion et de la Pollution) est un projet PREDIT dont l'objectif est de recueillir des données de trajectoires et de pollution afin d'améliorer la connaissance dans le domaine de la congestion et d'affiner les modèles de trafic.

Les trajectoires des véhicules sont recueillies à l'aide d'un drone équipé d'une caméra HD. Le volume de données vidéo est très important puisqu'il est estimé à plusieurs TéraOctets. Nous ne

nous intéressons pas à ce type de données mais à l'exploitation qui est faite de ces données. Les trajectoires des véhicules sont extraites des données vidéo par des algorithmes de traitement d'images.

Nous aurons pour chaque site :

- des tables de trajectoires. Dans cette table nous aurions des indicateurs globaux concernant les trajectoires : statistiques sur les vitesses (moyenne, écart-type, quantiles,...), statistiques sur les accélérations, nombre de changements de voies, vitesses auxquelles s'effectuent les changements de voies, durée des changements de voies, distances de changement de files,... L'idée serait de mettre à disposition des tableaux de données sur lesquels les chercheurs puissent pratiquer de la statistique exploratoire (ACP, classification,...) dans la perspective de modéliser.

- des tables contenant des données des trajectoires des véhicules : horodate, vitesse, voie de circulation, TIV, DIV, créneaux avant et arrière, créneau d'insertion arrière, créneau d'insertion avant, vitesses des véhicules de la file cible et courante,...Toutes les variables utilisées pour la modélisation des lois de poursuite et de changement de voie.

- une table des véhicules : longueur, type de véhicule,...
- des tables relatives aux sites d'expérimentation.

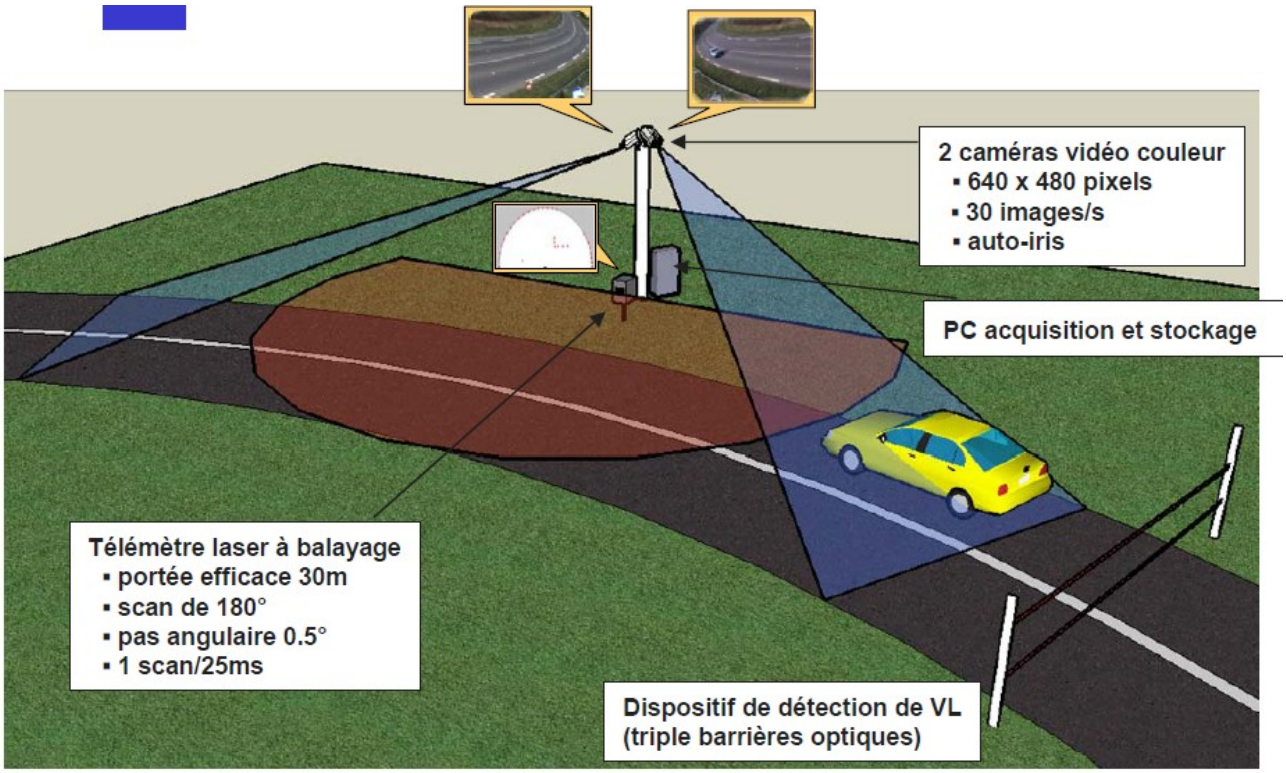
5.3 - Observatoires de trajectoires des véhicules

Durant l'opération de recherche MTT (métrologie des trajectoires et du trafic) des OdT (observatoires de trajectoires) ont été développés permettant de recueillir de nombreux paramètres liés à la trajectoire et au comportement de l'usager. Initialement, il avait été prévu d'élaborer une base de données de trajectoires permettant aux personnes d'utiliser les données et les résultats des travaux de recherche MTT.

Les données de trajectoires sont recueillies via deux types d'équipement terrain :

- des systèmes embarqués à l'intérieur des véhicules ;
- des équipements bord de route.

5.3.1 - Description d'un observatoire de trajectoires



Observatoire des trajectoires[1]

Comme le montre le schéma ci-dessus, les observatoires de trajectoires développés dans le projet SARI/RADARR comprennent 3 principaux éléments :

- un dispositif de détection de VL
- 2 caméras vidéo qui vont enregistrer la trajectoire du véhicule en amont et en aval du virage
- un télémètre laser qui va mesurer très précisément la trajectoire à l'intérieur du virage

5.3.2 - Systèmes embarqués

Les systèmes embarqués recueillent différents types d'information. Ils peuvent collecter les données suivantes :

- la position et la vitesse du véhicule via un GPS ;
- les données cinématiques du véhicule via les données transitant sur le bus CAN (vitesse,accélération,etc.) ;
- des données vidéo ;
- des capteurs tels que les radars permettant de qualifier l'environnement extérieur du véhicule ;
- les données issues d'une centrale inertielle.

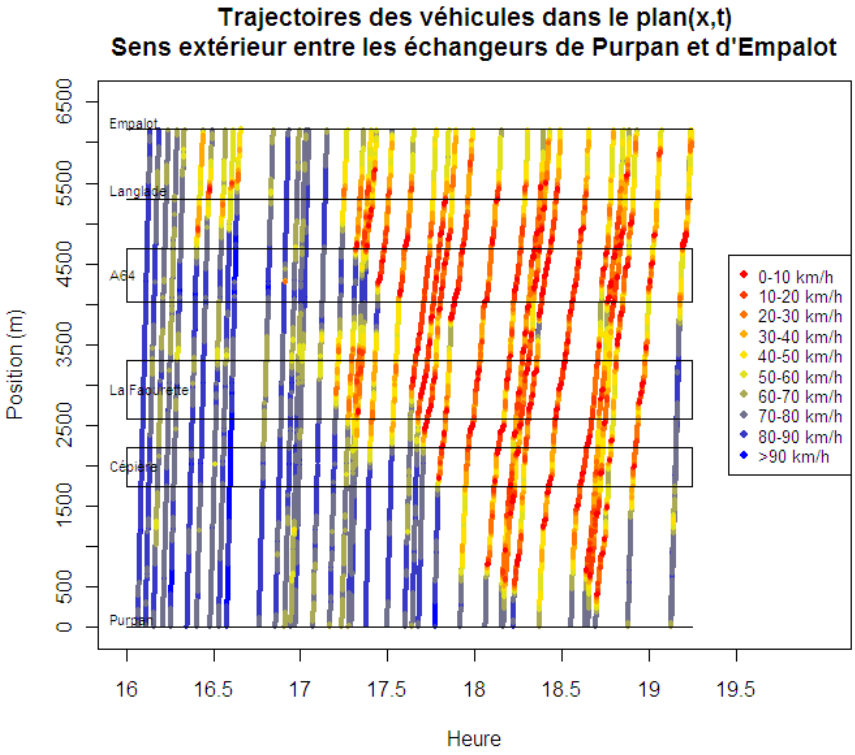
1 Extrait de la présentation « Offre globale de mesures de trajectoires » (Fabien Menant, IFSTTAR, Éric Violette, CETE NC)

5.4 - *Véhicules traceurs*

Les données provenant de véhicules traceurs ou FCD (Floating Car Data) permettent de mesurer des paramètres relatifs à un véhicule tel que sa position ou sa vitesse. Il existe différents niveaux d'instrumentation d'un véhicule traceur.

Instrumentation légère

Une instrumentation légère consiste à équiper un véhicule d'un simple GPS qui va récupérer des données horodatées de position et de vitesse. Les smartphones équipés de puce GPS peuvent également être utilisés. L'utilisation d'un GPS permet par exemple, d'obtenir la trajectoire du véhicule dans le plan (x,t), d'obtenir des profils de vitesse sur des itinéraires et de calculer des temps de parcours. Le graphique ci-dessous a été obtenu lors d'une campagne de mesure de temps de parcours en utilisant 5 véhicules traceurs munis de simple GPS data-logger.



Instrumentation classique

Généralement, les véhicules traceurs sont équipés d'un système embarqué qui récupère des données GPS ainsi que les données cinématiques du véhicule à partir du bus CAN. Cette instrumentation peut être utilisée dans le cadre des FOT « Field Operation Test » où l'on collecte des données sur le comportement de l'usager en situation de « conduite naturelle ».

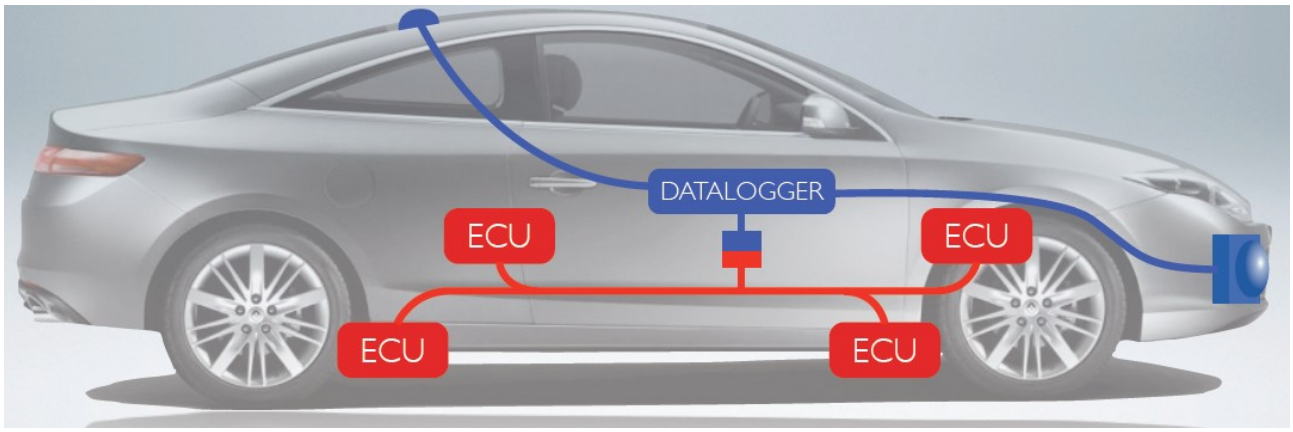
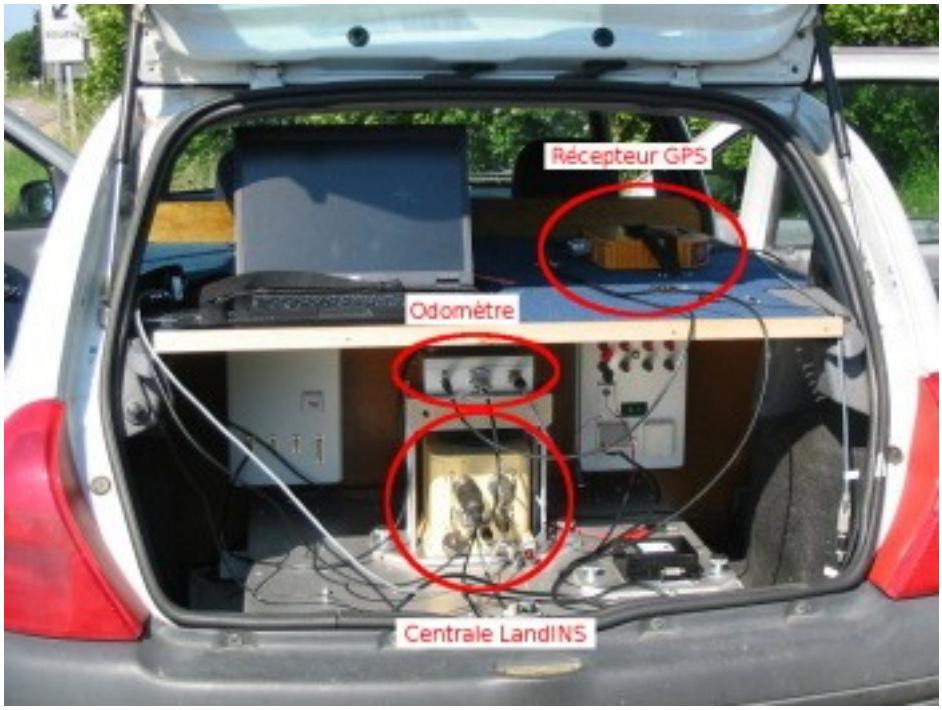


Illustration 1: Instrumentation CEESAR projet Eurofot

Ci-dessus, un exemple d'une instrumentation réalisée par le CEESAR dans le cadre du projet EuroFot.

Instrumentation lourde

Une instrumentation plus lourde consiste à équiper le véhicule de caméras ou webcam afin de filmer le comportement de l'usager et de capteurs tels des radars afin de connaître l'environnement dans lequel évolue le véhicule.



Véhicule MRT de l'IFSTTAR

6 - Format d'échange de données

Nous nous intéressons dans cette partie aux formats d'échange de données couramment utilisés non propriétaires.
Nous nous focalisons plus particulièrement sur 3 types de format de données :

- le format CSV ou texte permettant l'échange de données sous forme de tableau ;
- le format SQL utilisé par des logiciels manipulant des bases de données relationnelles (logiciels SIG, systèmes de gestion de bases de données (SGBD)) ;
- le format XML, format d'échange couramment utilisé pour l'échange de données sur internet et entre systèmes d'information.

6.1 - Le format CSV

6.1.1 - Description

Un fichier CSV (Comma-Separated Values) est un fichier texte et est donc lisible par n'importe quel éditeur de texte.
Il contient des données sous forme de tableau.
Chaque ligne contient des données séparées par un caractère de séparation, généralement une virgule, un point-virgule ou une tabulation.
Un fichier CSV peut contenir une ligne d'en-tête contenant les noms des colonnes.

6.1.2 - Avantages

Le format CSV est un format standard.
La plupart des logiciels dont les tableurs, les systèmes de gestion de bases de données et les SIG importent et exportent des données dans ce format.

6.1.3 - Inconvénients

Un fichier CSV ne contient que des données. Nous n'avons pas d'informations sur la nature de ces données. Il est nécessaire de préciser ou de valider la nature des différents champs du fichier CSV.
Un fichier CSV ne peut contenir que des données sous forme de tableau.

6.2 - Le format SQL

6.2.1 - Description

Le format SQL« Structured Query Language » n'est pas à proprement parler un format de fichier mais un langage de base de données relationnelle.
Le langage SQL permet de manipuler et d'interroger une base de données relationnelle.
Ce langage comporte entre autres:

- un langage de définition de données (LDD) ;
- un langage de manipulation de données (LMD).

Le LDD permet de définir et de décrire la structure des données de la table tandis que le LMD

permet de manipuler (ajout, modification, extraction) des données.

Le langage SQL permet aussi la gestion des droits d'accès aux données mais cela ne fait pas l'objet du document présent.

Un fichier d'échange de données SQL est en réalité un script SQL contenant :

- des instructions de création de tables décrivant la structure des données à insérer ;
- des instructions d'insertion de données.

Ci-dessus, un exemple de script SQL qui crée la table trafficdata et y ajoute quelques enregistrements.

```
CREATE TABLE trafficdata
(
  id integer NOT NULL PRIMARY KEY,
  date timestamp with time zone NOT NULL,
  lane integer,
  speed integer,
  length integer
);

INSERT INTO trafficdata (id,date,lane,speed,length)
VALUES (1,'2009-05-01 12:00:00.00+02',0,100,50),
           (2,'2009-05-01 12:00:02.93+02',1,90,44),
           (3,'2009-05-01 12:00:03.18+02',2,80,55),
           (4,'2009-05-01 12:00:04.11+02',0,77,88),
           (5,'2009-05-01 12:00:05.23+02',1,98,48),
           (6,'2009-05-01 12:00:07.38+02',0,53,151);
```

Nous pouvons ajouter d'autres objets ou instructions au script précédent non compatibles avec les normes SQL et propres à certains SGBDR. Par exemple, des fonctions qui vérifient la validité des données à insérer (triggers), des instructions de génération de clé primaire (séquences), des fonctions de calcul (procédures stockées),...

6.2.2 - Avantages

Le format SQL permet l'export et l'import de données entre différents systèmes de gestion de base de données relationnelle.
Un fichier SQL se suffit à lui-même : l'importation de données dans ce format ne nécessite pas d'informations complémentaires.
La langage SQL fait l'objet de normes ISO. Des fichiers SQL conformes aux normes ISO seront donc lisibles par la plupart des SGBDR les respectant.

6.2.3 - Inconvénients

Le langage SQL ne peut être interprété que par des logiciels de bases de données relationnelles.

6.3 - Le format XML

Actuellement, le format XML (Extensible Markup Language) est utilisé comme format d'échange entre systèmes d'information. C'est un format où les données sont structurées de manière arborescente par des balises.
La puissance du langage XML vient de son extensibilité permettant de décrire n'importe quel jeu de données. C'est une bonne alternative aux fichiers CSV lorsqu'un fichier contient des données à structures variables qu'il est difficile de mettre sous la forme de tableaux.
On évitera cependant de mettre des données au format XML lorsqu'elles peuvent être présentées sous forme de fichiers CSV puisque les fichier XML de par leur structure sont volumineux.

6.3.1 - Exemple de fichier XML

Pour nos données de trafic, nous pouvons ajouter en plus des données individuelles, les caractéristiques de la station de mesure de laquelle proviennent les données, des conditions météorologiques, des informations permettant de caractériser la spécificité de nos données,...

L'exemple ci-dessous décrit dans le formalisme XML une station de comptage.

```
<stationComptage identifiant="RADN422BLAESHEIM_NORD(NS)">
  <libelle>BLAESHEIM_NORD (NS)</libelle>
- <localisation>
-   <localisationASP>
      <idAxe>N422</idAxe>
      <sens>Strasbourg/Colmar (NS)</sens>
      <prAmont>8</prAmont>
      <offsetAmont>23</offsetAmont>
    </localisationASP>
    <localisationGPS>
      <longitude>1.5</longitude>
      <latitude>47.5 </latitude>
    </localisation GPS>
  </localisation>
  </stationComptage>
```

Une donnée individuelle peut être décrite de la manière suivante dans un fichier XML :

```
<data id= « 454654 »>
  <date>2007-24-07 01:00:00+02</date>
  <lane>0</lane>
  <speed>120</speed >
  <length>150</length>
</data>
```

6.3.2 - Fichier XSD

Un fichier XML peut être fourni avec un « XML Schema » (contenu dans un fichier XSD (XML Schema Description)) définissant la structure du document et permettant sa validation.

Dans l'exemple ci-dessous nous décrivons le contenu de l'objet « data » et l'ensemble des attributs d'une donnée individuelle.

```
<xsd:complexType name="data">
  <xsd:sequence>
   <xsd:element name="date"   type="xsd:dateTime"/>
   <xsd:element name="lane" type="xsd:positiveInteger"/>
   <xsd:element name="speed"   type="xsd:positiveInteger"/>
   <xsd:element name="length"  type="xsd:PositivInteger"/>
  </xsd:sequence>
 </xsd:complexType>
```

Dans la description de la structure du document XML, nous pouvons aussi définir de nouveaux types de données en ajoutant des contraintes aux types de données usuels. L'exemple ci-dessous impose à la valeur de l'attribut vitesse des données individuelles d'être comprise entre 10 et 200 km/h.

```
<xsd:simpleType name="speedInteger">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="10"/>
    <xsd:maxInclusive value="200"/>
  </xsd:restriction>
</xsd:simpleType>
```

6.3.3 - Avantages

Le format XML permet de décrire très précisément les données contenues dans un document.
Les données d'un document XML sont structurées et hiérarchisées. Ces fichiers peuvent être exploités avec des outils de type DOM (Document Objet Model) qui permet de manipuler, d'exploiter et de transformer les données contenues dans des fichiers XML.

6.3.4 - Inconvénients

Un document XML ne peut être importé directement dans un logiciel. Il est nécessaire d'utiliser un analyseur syntaxique (parser XML) ou des outils de type DOM permettant d'une part de vérifier la cohérence et la validité du document et d'autre part d'extraire les informations à stocker dans une base de données.

La verbosité du langage XML est un inconvénient. Les fichiers XML sont beaucoup plus volumineux que les fichiers CSV. Chaque valeur est entourée par des balises contenant le nom de la variable contrairement à des fichiers CSV où les valeurs sont séparées par des délimiteurs. On peut toujours compresser les fichiers XML mais leurs tailles seront toujours très nettement supérieurs à ceux de fichiers CSV compressés.

7 - Base de données SERRES

Comme il existe de nombreuses confusions autour du terme « base de données » nous en rappelons la définition en informatique (source wikipedia) :
« En informatique, une **base de données** (Abr. : « BD » ou « BdD » ou encore DB en anglais) est un lot d'informations stockées dans un dispositif informatique. Les technologies existantes permettent d'organiser et de structurer la base de données de manière à pouvoir facilement manipuler le contenu et stocker efficacement de très grandes quantités d'informations.»

Parfois nous confondons base de données avec système de gestion de bases de données (SGBD) qui est un logiciel qui manipule une base de données. Pour constituer une base de données, il n'est pas nécessaire d'utiliser un SGBD, les données peuvent être stockées dans des fichiers texte ou dans des tableurs.

Base de données intranet / internet SERRES

La base de données internet développée dans le cadre de l'opération de recherche SERRES est un site internet ou une plate-forme permettant à un utilisateur authentifié de récupérer des données relatives à SERRES. Concernant, la mise à disposition des données, comme la plate-forme ne doit proposer que des données de qualité, l'étape consistant à valider et/ou transformer les données brutes ou pré-traitées est à la charge soit du fournisseur de données s'il dispose les outils de traitement où aux administrateurs de la plate-forme. On peut aussi autoriser les fournisseurs de données à publier directement leurs données sur le site mais elles ne sont rendues public qu'après validation par l'administrateur ou la personne compétente dans le domaine.

7.1 - Types et structure de bases de base données

Nous nous intéressons ici à la manière de stocker les données du côté serveur. Il existe plusieurs méthodes pour stocker les données. Nous en donnons par la suite les principales :

- des fichiers plats ;
- données stockées dans un système de gestion de base de données relationnelles (SGBDR) ;
- une partie des données stockées dans un SGBR et des fichiers plats.

7.1.1 - Fichiers plats

Les données sont stockées sous forme de fichiers. On peut regrouper différents fichiers et les compresser afin de faciliter le téléchargement des données. Chaque fichier ou jeu de données doit être documenté. L'utilisateur doit savoir si le contenu des fichiers l'intéresse avant de les télécharger. La documentation des jeux de données peut être stocké de manière dynamique dans une base de données. Ainsi, l'utilisateur peut rechercher les données qui l'intéressent selon plusieurs critères.

Avantages

En stockant les données sous forme de fichiers, les données sont déjà formatées. Nous évitons ainsi au niveau de la plate-forme d'échange le développement d'outils de récupération et de formatage des données stockées dans un SGBDR.
C'est une solution économique en termes de ressources matérielles : on consomme peu de ressources systèmes (charge CPU et mémoire) et donc le site peut être accueilli sur un serveur mutualisé (serveur partagé par plusieurs site internet).
D'autre-part, la base de données est nettement plus facile à administrer, à maintenir et à mettre en œuvre puisque nous gérons des fichiers de jeux de données et non des données.

Inconvénients

L'utilisateur doit télécharger dans sa globalité le ou les jeux de données. Il ne peut pas sélectionner un échantillon du jeu de données ou certaines variables. Il est obligé d'analyser les données contenues dans le fichier afin d'en extraire celles dont il a besoin, étape qui aurait pu être réalisée par la base de données. L'analyse, l'extraction des données pertinentes peuvent décourager les utilisateurs à utiliser la base de données. Enfin des fichiers trop volumineux peuvent dissuader plus d'un à les télécharger.

7.1.2 - SGBDR

Les données sont stockées dans un système de gestion de bases de données relationnelles. Il faut donc définir des modèles de base de données pour chaque types de données. La modélisation peut s'avérer complexe dès que nous disposons de plusieurs sources de données. L'utilisation d'un SGBDR doit être pertinente. Est-ce cela vaut la peine de mettre en place une architecture complexe d'interrogation et d'extraction des données ? Faut-il par exemple stocker toutes les données individuelles de trafic dans la base ou l'on peut se contenter de stocker des données agrégées qui synthétisent les données individuelles mises à disposition sur le site internet ?

Avantages

L'utilisateur peut sélectionner les données et les variables dont il a besoin. Du côté de l'application qui gère la base de données, nous pouvons mettre en œuvre des filtres et des critères de sélection très élaborés afin que l'utilisateur final puisse récupérer les données qui l'intéressent. Comme les données sont accessibles par l'application, on peut aussi proposer des outils de visualisation graphique des données : histogrammes, courbes, visualisation des trajectoires des véhicules....

Inconvénients

Le stockage des données dans un SGBDR demande le développement d'une application Web assez complexe. Il est notamment nécessaire de développer des modules d'import et d'export des données.
Le stockage des données demande de disposer d'importantes ressources (CPU et RAM) du côté de l'application Web et du SGBR.

7.1.3 - Solution mixte entre SGBDR et fichiers plats

Une solution mixte entre SGBR et fichiers consiste à ne stocker dans la base de données qu'une partie des données. Les données stockées dans la base de données sont les données sur lesquelles peuvent s'appliquer des requêtes par l'application gérant le base. Les données détaillées qui n'interagissent pas directement avec l'application sont stockées dans des fichiers. Prenons l'exemple de données individuelles de trafic. L'utilisateur souhaite télécharger des données individuelles correspondant à des états de trafic bien déterminés. L'application effectuera des requêtes sur des données agrégées (données 6 minutes) à partir desquels elle déterminera les jeux de données correspondant aux situations de trafic demandées par l'utilisateur. L'application indiquera à l'utilisateur les jeux de données disponibles correspondant à ses besoins et les mettra à sa disposition.
On peut aussi stocker dans la base de données uniquement les méta-données qui vont décrire et synthétiser les fichiers de données.

Avantages

Nous disposons d'une base de données optimisée et performante. Nous ne stockons dans le SGBDR que ce qui est nécessaire. La sélection des jeux de données peut se faire à partir d'indicateurs globaux (caractéristiques de trajectoire, situations de trafic, conditions d'expérimentation,..) et on laisse à l'utilisateur la tâche d'analyser les données détaillées contenues dans les jeux de données sélectionnés par l'utilisateur.

Inconvénients

On retrouve ici les mêmes inconvénients que ceux cités pour les fichiers plats, à savoir que c'est à l'utilisateur d'analyser les jeux de données afin d'extraire les données qui l'intéressent.

7.2 - Importation des données

Entre le fichier brut issu de la station de mesure et la mise à disposition de données individuelles depuis la base de données, le contrôle des données peut s'effectuer à différents stades de l'importation :

- lors de la conversion du fichier de données brutes en fichier d'échange de données ;
- lors de l'insertion des données dans la base de données ;
- après insertion des données dans une base de données.

7.3 - Exportation des données

Lors de l'exportation des données depuis la plate-forme d'échange, si nous avons opté pour le stockage des données dans un SGBDR, nous devons récupérer et mettre en forme les données. Durant cette étape, nous pouvons calculer de nouveaux indicateurs, filtrer les données en fonction du besoin de l'utilisateur. Comme nous l'avons dit précédemment, la base de données produira des

fichiers dans un format d'échanges qui facilitera l'import de données dans un logiciel de statistique, un tableur ou un SGBDR.
Pour des données individuelles de trafic, lors de l'extraction des données nous pouvons :

- vérifier la cohérence de certaines variables telles que la vitesse, la longueur du véhicule, le temps de présence,...
- calculer des attributs supplémentaires: classe de silhouette du véhicule,TIV,...
- calculer des données agrégées (données 6 minutes, données horaires,...) ;
- ajouter des informations sur le contexte de circulation (conditions météo, conditions de trafic, lieu de la mesure,...).

7.4 - Les triggers

Les triggers ou déclencheurs exécutent des fonctions lors de certaines actions sur la base de données : insertion, modification ou suppression de données.
Nous pouvons faire appel à des déclencheurs sur certaines tables de notre base de données afin de vérifier la cohérence des données à insérer et aussi de calculer de nouveaux attributs (classes du véhicule,TIV,...)

Exemple : création d'un trigger sous PostgreSQL de vérification des données lors d'insertion de données dans la table « trafficdata »

```
CREATE TRIGGER check_data
BEFORE INSERT ON trafficdata FOR EACH ROW
EXECUTE PROCEDURE check_insert_data();
```

Le trigger ci-dessous appelle la fonction « check_insert-data » à chaque tentative d'insertion de données dans la table « trafficdata ».

```
CREATE OR REPLACE FUNCTION check_data() RETURNS TRIGGER AS
$body$
 DECLARE
        v_date timestamp with time zone;
        iduser integer;
 BEGIN
        SELECT INTO v_date MAX(date) FROM trafficdata;
        IF v_date >  NEW.date THEN
          RAISE EXCEPTION 'Date error';
        END IF;
        IF NEW.length < 10  OR NEW.length > 400 THEN
           --   RAISE EXCEPTION 'Length error';
           NEW.length=NULL;
        END IF;
        IF NEW.speed <= 0  OR NEW.speed > 250 THEN
            -- RAISE EXCEPTION 'Speed error';
            NEW.speed=NULL;
        END IF;
        RETURN NEW;
 END;
$body$
```

```
  LANGUAGE 'plpgsql';
```

La fonction ci-dessus est lancée par le trigger « check_data » avant insertion de données dans la table « trafficdata ». Elle vérifie notamment la cohérence des deux champs vitesse et longueur et n'accepte pas les données si l'horodate est antérieure aux autres enregistrements déjà stockés (ce qui est contestable puisque les données ne sont pas forcément ordonnées par date d'arrivée).

7.5 - Les procédures stockées

Les procédures stockées sont des fonctions pré-compilées, stockées sur le serveur et directement exécutées par la base de données. Ces fonctions peuvent être programmées à l'aide de langage de haut-niveau tel que PL/PGSQL pour le SGBDR PostrgeSQL, ou en langage de bas niveau tel que le langage C.
Par le biais de ces procédures stockées, nous pouvons définir des fonctions de calcul de distances intervéhiculaires, de temps de présence ou encore de catégories de véhicules à partir des données de base stockées dans la base de données.

```
CREATE OR REPLACE FUNCTION getTIV(integer)
  RETURNS double precision AS
$BODY$
 DECLARE
  v_id ALIAS FOR $1;
  rec RECORD;
  t2 double precision;
  t1 double precision;
  result double precision;
 BEGIN
  SELECT INTO rec * from trafficdata where id = v_id;
  t2 := extract(epoch FROM rec.date);
  SELECT INTO t1 extract(epoch FROM MAX(date)) from trafficdata where id < v_id
and lane = rec.lane;
  result :=(t2-t1)-(0.36*rec.length)/rec.speed;
  IF result <= 0 THEN
   -- RAISE EXCEPTION 'TIV négatif';
    RETURN NULL;
  END IF;
  RETURN result;
 END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
```

La fonction ci-dessus getTIV(id) calcule l'interdistance en secondes du véhicule d'identificateur « id » avec le véhicule qui le précède.

La requête suivante permet d'extraire des données individuelles en y ajoutant les variables type du véhicule, TIV et DIV en faisant appel à 3 fonctions stockées dans la base de données.

```
    SELECT *,getvehiculetype(length),gettiv(id),getdiv(id), FROM trafficdata
```

Nous pouvons également programmer des procédures plus complexes, par exemple, le contexte de

circulation de chaque véhicule en attribuant pour chaque enregistrement des données 6 minutes.

8 - Architecture Web

8.1 - Technologies Internet

Les 3 principaux langages utilisés pour la programmation Web sont :

- ASP et plate forme .NET (technologie Microsoft) ;
- JAVA (JSP,...) ;
- PHP.

Nous n'étudions pas la technologie Microsoft puisqu'il s'agit d'une solution propriétaire même s'il existe une implémentation libre du framework .NET : Mono. Si nous cherchons à développer une apllication puissante et offrant de bonne performance, la technologie alternative à la plate forme .NET est le JAVA.

Ici nous n'essayons pas de déterminer quel est la meilleure technologie la plus adaptée à notre base de données. Cela dépend des compétences des développeurs de la plate-forme et quelle architecture nous souhaitons mettre en œuvre. Nous donnons cependant quelques éléments de comparaison entre ces deux technologies.

PHP

PHP est un langage de script non typé . Il est donc théoriquement plus lent que Java . PHP est très répandu puisque la technologie PHP est beaucoup plus facile à mettre en œuvre et elle est accessible par tous. De nombreux systèmes, logiciels, portails et sites internet sont réalisés en PHP. Il existe de nombreux framework PHP dont CakePHP, Symphony et ZEND qui permettent de mieux structurer le code PHP et d'accélérer le développement de l'application.

JAVA

Concernant la technologie JAVA, elle est bien adaptée au développement d'importantes applications Web. La technologie nécessite l'utilisation d'un conteneur web (le plus connu étant Tomcat) qui exécute des « servlets » (classes JAVA). Comme pour le PHP, il existe de nombreux framework permettant de mieux structurer le code JAVA, de produire de code de qualité et d'accélérer le développement d'applications.
La technologie JAVA demande beaucoup plus d'investissements que la langage PHP et n'est donc pas appropriée pour de petits projets de site internet.

8.2 - Types de Plate-forme envisagés

8.2.1 - Site statique

Un site statique est constitué de pages HTML et qui ne dépend pas d'un SGBDR. Ce type de site est très facile à mettre en œuvre mais il est difficile à maintenir puisque le contenu n'est pas gérer dynamiquement. De plus il est très difficile d'affecter des droits sur le fichiers hébergés sur le site. Nous déconseillons les sites statiques pour notre plate-forme d'échange de données puisque d'une part se pose des problèmes de confidentialité des données (Il est cependant possible de faire une gestion très fine des droits même en site statique en utilisant les module d'authentifications des serveurs web (apache par exemple) via des sources de données d'utilisateurs (LDAP ou SQL par exemple)), d'autre part une gestion non dynamique des données est inappropriée pour une plate-forme d'échange de données.

8.2.2 - Plate-forme de gestion de contenus

Comme leur nom l'indique, les systèmes de gestion de contenu (CMS en anglais) sont des logiciels permettant de gérer dynamiquement du contenu. Il existent plusieurs types de CMS. Pour notre plate-forme de partage de données, les outils CMS sont ceux nous permettant de mettre en place une plate-forme d'échange de données.
Les 2 systèmes de gestion de contenu les plus pertinents pour notre plate-forme de partage de données sont : Joomla ! et Drupal. Ils sont tous les deux gratuits et Open Source.
Ces deux systèmes peuvent être adaptés à nos besoins par l'intermédiaire de nombreuses extensions. Sur ces deux systèmes, ils existent notamment des extensions permettant de produire des sites de type « communautaire » avec une bonne gestion des documents.

Ici, nous ne rentrons pas dans les fonctionnalités détaillés de ces deux outils. Joomla ! est le système de gestion de contenu le plus répandu et il est beaucoup plus simple à mettre en œuvre que Drupal.
Drupal est beaucoup plus flexible et plus puissant que Joomla! mais en contre partie il requiert d'avantage d'investissements que Joomla!. Si l'on veut produire un site simple, on s'orientera vers une solution basée sur Joomla !. Si au contraire, on cherche à développer un plate-forme plus complexe, on choisira un site basé sur Drupal.

8.2.3 - Le site NGSIM

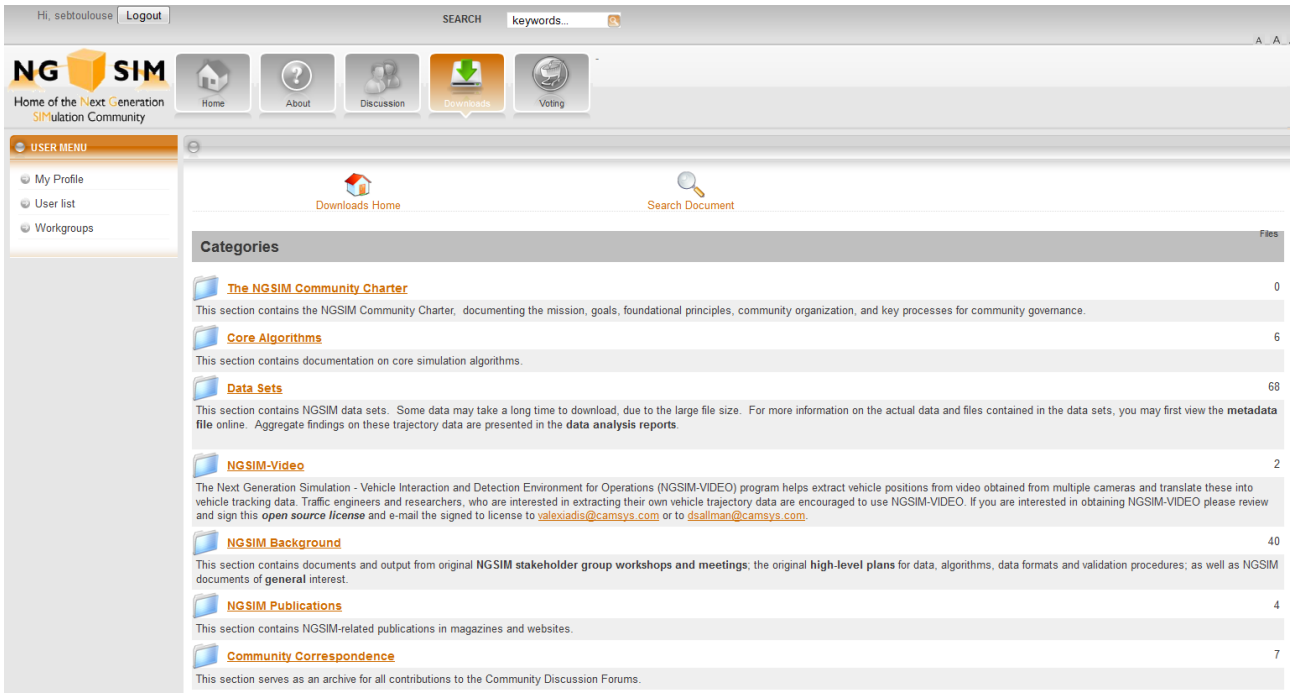
NGSIM (Next Generation of Simulation) est un programme de recherche américain lancée par l'administration fédérale des autoroutes du département des transports des États-Unis. Durant ce programme, des données de trajectoires ont été collectées sur différents types d'infrastructure afin de développer de nouveaux modèles de comportements des usagers et de simulation dynamique.
Les données NGSIM sont publics et une large communauté s'est développée autour de ces données. Elles sont accessibles via le site communautaire http://ngsim-community.org/. Outre les données vidéo, nous trouvons notamment les logiciels de traitement de ces données, les données traitées, des papiers relatifs à l'exploitation de ces données et de modèles développés à partir de ces données.
Très connus des chercheurs, la description du site NGSIM nous semble incontournable dans cette

note méthodologique sur les bases de données.

8.2.3.a - Architecture du site

Le site internet NGSIM a été développé en utilisant le système de gestion de contenu Joomla! et utilise les extensions suivantes « Community Builder » et « DocMan ».
L'extension « CommunityBuilder » permet de gérer les utilisateurs du site et de construire une communauté d'utilisateurs des données NGSIM.
L'extension « Docman » permet de gérer les fichiers et les téléchargements.

Pour accéder aux données NGSIM, il est nécessaire de s'identifier. Cette authentification permet d'accéder aux données NGSIM mais aussi au logiciel de traitement vidéos, aux articles et papiers relatifs aux données, aux algorithmes et modèles développés à partir des données NGSIM. On accède aussi à un forum permettant d'échanger autour des modèles de simulation, des algorithmes et des données NGSIM.



Site NGSIM

8.2.4 - Plate-forme de type LAMP

LAMP signifie Linux-Apache-MySQL-PHP. IL s'agit des plate-formes utilisé pour l'utilisation de sites développés en PHP. Il existe des variantes de ces plate-formes ou l'on peut par exemple remplacer le SGBDR MySQL par PostgreSQL.

8.2.4.a - Serveur Apache

L'architecture Web d'une application repose généralement sur le logiciel de serveur HTTP Apache C'est un logiciel libre et le serveur HTTP le plus populaire et le plus utilisé.

8.2.4.b - Scripts PHP

Le langage PHP est un langage libre et est le plus utilisé pour la conception de sites internet dynamiques.
Les applications en PHP sont relativement simples à mettre en place et le langage PHP s'interface très facilement avec des systèmes de gestion de base de données dont PostgreSQL et MySQL.

8.2.4.c - PostgreSQL ou MySQL

Les deux SGBDR les plus utilisé avec le langage PHP sont PostgreSQL et MySQL.
PostgreSQL est plus performant que MySQL dans le traitement et l'exploitation de grosses bases de données.
MySQL est davantage utilisé pour la gestion de bases de données de petite taille.

Certains frameworks PHP permettent de développer des applications indépendantes de la base de données en utilisant des couches génériques d'accès aux données.

8.2.5 - Plate-forme JEE

JEE (anciennement J2EE) est l'acronyme de Java Entreprise Edition. Le terme JEE est cependant couramment encore employé pour désigner la plate-forme on peut trouver également le terme plus générique JEE (sans le 2).

Plus qu'une simple plate-forme, JEE est une spécification, une norme. Elle défini un ensemble de bonnes pratiques et de spécification d'architecture logique multi-niveaux (voir plus loin architecture trois tiers).

La spécification défini ainsi entre autres la plate-forme elle-même sur laquelle seront hébergées et exécutées les applications ainsi qu'un serveur de référence (GlassFish). Ce serveur de référence n'est cependant pas nécessaire à l'exécution de l'application WEB peut être également remplacé par une autre implémentation libre ou propriétaire (JBOSS ou Geronimo par exemple).

8.2.5.a - Conteneur de Servlet et de pages JSP

La partie essentielle et non réductible est l'usage d'un conteneur de Servlet et de pages JSP comme TOMCAT ou JETTY par exemple. Ces conteneurs permettent de mettre en place une couche d'architecture très importante (que nous détaillerons plus loin) la couche MVC. C'est la couche en charge d'assurer la présentation, c'est à dire ce que voit l'utilisateur, des données qui sont stockées et gérées par les couches métiers de l'application WEB.

Les pages JSP et des Servlet permettent rapidement de développer déjà une couche de présentation pour l'application WEB. Cependant dès que le projet commence à prendre de l'ampleur, il est

rapidement nécessaire d'organiser les ressources, il est alors possible d'utiliser des frameworks dédiés à la couche de présentation comme par exemple STRUTS ou JSF. Ces frameworks permettent de respecter la logique du MVC mais obligent à une certaine organisation du projet.

Le conteneur peut être utilisé seul ou conjointement avec un serveur WEB comme Apache (voir partie sur les plate-forme de type LAMP, selon la charge du serveur.

8.2.5.b - Couches métiers et accès aux données

Dans ces couches nous trouvons tous ce qui essentiel à la bonne gestion des données. Nous détaillerons son contenu dans la partie « Architecture trois tiers ».

L'important à retenir pour le moment, c'est que sur la plate-forme JAVA depuis la version 6, on trouve nativement un frameworks d'abstraction des données : JPA. Plusieurs implémentations existent dont la plus connu est Hibernate.

La couche d'accès aux données utilise les services d'un SGBD à l'identique d'une plate-forme LAMP. L'interface logique couramment employée est JDBC.

Dans l'ensemble des spécifications JEE on trouve un élément assez fréquemment, les EJB. Les Entreprise Java Bean sont des composants transactionnels distribuées. Il faut imaginer les EJB comme des composants applicatifs disposant d'interfaces de communications avec l'extérieur et ainsi d'autre EJB. Chaque EJB est responsable de son propre domaine métier. Cette logique permet d'industrialiser la conception et la réalisation des applications WEB mais accrois la complexité de l'ensemble. Les EJB ont souvent été décriés pour cette raison, ce qui a permis émergences de nouvelles façon d'organiser les projets grâce notamment aux conteneurs légers comme SPRING.

La dernière implémentation des EJB (la version 3) prends acte du passé et vient combler les lacunes des version précédentes. Cependant le conteneur léger reste encore une très bonne technique permettant de gérer facilement les évolutions d'un projet d'architecture web.

8.3 - Types d'architecture d'application Web

Les architectures présentées ci après permettent toutes de respecter un concept essentiel en conception et développement logiciel : la séparation des préoccupations.

Le principe est de séparer en entités distinctes, certaines parties de l'application WEB selon leur rôle non pas fonctionnel mais plutôt architectural. La programmation orienté objet permet déjà de découper sous formes d'entités qui interagissent les unes sur les autres cependant, il subsiste un risque important que l'on peut désigner sous le nom de code spaghetti : lorsque l'on tire sur un spaghetti, c'est l'ensemble du plat qui vient …

Pour éviter ceci il est nécessaire de séparer en couches les applications. A l'intérieur de chaque couche, on retrouve les objets en interactions. Cependant chaque couche n'a pas connaissance de la complexité interne de ses homologues et ne connaît uniquement que les services que la couche

concerné met à disposition.

Ces différentes couches s'appuient elles-mêmes sur les bonnes pratiques désignées sous le terme de Patrons de Conceptions (Design Pattern) qui ne sont en fait que la synthèse de bonnes pratiques de la communauté des développeurs en réponse à des problèmes récurrents de conception. Ces patrons ont été regroupé dans un ouvrage connu sous le nom de GoF « Gang of Four » : *Design Patterns – Elements of Reusable Object-Oriented Software en 1995.*

8.3.1 - Architecture trois tiers

Une architecture trois tiers est composée de 3 couches : une couche de présentation, une couche métier et une couche d'accès aux données. Nous séparons ainsi la présentation des données et des résultats (interface entre l'utilisateur et l'application) de la partie applicative mais aussi la base de données et son accès des processus métiers de l'application.

Nous détaillerons ci après la couche de présentation réalisé grâce au Patron de Conception MVC.

La couche métier d'une application est responsable de la gestion des données métiers et par conséquent des règles dites de « gestion » de ces même données. D'une manière général, la couche est séparée en deux sous parties constituée d'une part par ses fameuses règles de gestion (les business activities) et les entités métiers proprement dites. Les règles de gestion définissent comment les entités doivent être créées ou détruites et dans quel ordre, selon les cas d'utilisation définis lors de la phase de conception préliminaire du projet.

La couche d'accès aux données est elle responsable de l'adaptation au paradigme objet, des données issues d'une base de données (par exemple relationnelle). Son rôle est essentiellement de rendre indépendant la partie logicielle du type de SGBD utilisé. Ainsi même si on décide de modifier le type de support matériel des données (fichiers plats ; SGBD ; fichiers XML ; etc) la couche métier ne sera pas modifiée.

8.3.2 - Patron MVC

Le patron de conception MVC (Modèle-Vue-Contrôleur) est un méta patron très répandue et est un standard dans le développement d'applications Internet. Nous encourageons à utiliser cet solution pour le développement d'applications.
Le patron MVC permet de séparer en sous partie la couche de présentation :

- le modèle ;
- la vue ;
- le contrôleur.

Le modèle
C'est simplement l'accès à la couche métier de l'application.

La vue
C'est la partie visuelle de la couche de présentation. Il s'agit d'un ensemble d'entités (les différentes vues de l'application). Cela peut être différentes pages HTML ou des widget dans une application

lourde. Chaque vues contient un certains nombre de propriétés comme par exemple les différents composant d'interaction avec l'utilisateur ainsi que différentes actions possibles.

Le contrôleur
Les actions possibles des vues sont redirigées vers le contrôleur. C'est l'aiguilleur de la couche de présentation. C'est lui qui en charge de gérer les actions et les successions des différentes vues en fonction de la dynamique de l'application et des cas d'utilisation.

Le patron MVC utilisé pour les application WEB est le MVC 2. Dans ce patron spécifique, on prend en compte que contrairement à une application classique (client lourd) il n'y pas de mise à jour direct de la vue selon les modifications de la couche métier.

8.4 - Principales fonctionnalités

La plate-forme internet doit pouvoir gérer les utilisateurs et fixer des droits sur les données hébergées par la plate-forme. Elle doit proposer un module de recherche de données. Enfin, si les données sont stockées dans un SGBDR, la plate-forme doit disposer d'une module d'import et d'export des données.
La fonction import de données ne sera accessible qu'aux fournisseurs de données qui les fourniront pré-traitées dans un format d'échange compréhensible par l'application.
La fonction export de données permettra à des exploitants, à des chercheurs d'accéder à des données fiables et labellisées.

8.4.1 - Gestion des utilisateurs

La gestion des utilisateurs est une obligation même si le site ne contient que des données publiques. Cela permet d'avoir des renseignements sur les utilisateurs et l'utilisation du site. Une grande partie des fournisseurs de données exige que l'accès à leurs données soit sécurisé et que l'utilisateur du site soit identifié.
L'usager du site internet est authentifié par un identifiant (email ou nom) et un mot de passe. Lors de l'inscription de l'utilisateur, des informations tels que son identité, son adresse professionnelle, sa fonction et des renseignement sur l'organisation dans laquelle il travaille seront demandées et stockées dans une table permettant la gestion des utilisateurs. Ces informations sont utiles et nécessaires aux fournisseurs de données et à l'administrateur pour l'attribution des droits d'accès aux données.
Il est très fortement conseillé de crypter les mots de passe des utilisateurs et si possible sans possibilité de retrouver le mot de passe.
Il est enfin souhaitable que le site internet puisse restreindre l'accès à certaines données à certains types d'utilisateurs.

8.4.2 - Recherche de données

Le site internet doit être hiérarchisé en catégories et rubriques permettant aux utilisateurs de trouver facilement ce qu'il recherche. Le site peut intégrer un moteur de recherche permettant aux utilisateurs de rechercher des données ou informations par méta-données ou mots-clés.

8.4.3 - Import de données

La fonction import de données permettra de stocker de volumineux jeux de données individuelles. La base de données ne stockera que les données fondamentales. Les attributs qui peuvent se séduire des autres variables pourront cependant être fournis aux demandeurs de données.
Comme nous manipulons de gros volumes de données, les fichiers d'échange de données seront tout d'abord copiés sur le serveur d'applications avant d'être importés dans la base de données.
A tout moment, l'utilisateur sera tenu au courant de l'avancement de l'importation (transfert du fichier sur le serveur et importation des données sous PostgreSQL) et des éventuelles anomalies rencontrées.

8.4.4 - Export de données

L'extraction d'informations de la base de données se fera via des requêtes SQL. Les calculs d'attributs supplémentaires non stockés dans la base se feront soit directement par le SGBDR via des procédures, soit par le serveur applicatif via des scripts. Le fichier de résultat sera tout d'abord créé sur le serveur d'applications avant d'être exporté à l'utilisateur via un navigateur internet.

9 - Hébergement de la base de données

Le choix de l'hébergement est fortement lié au choix de la technologie adopté ainsi que la structure de la base de données.

9.1 - Hébergement en externe

9.1.1 - Serveurs mutualisés de type LAMP

Les serveurs mutualisés sont partagés par différents utilisateurs et sites internet. Ils s'adressent à des sites à faible et moyen trafic. Généralement, les sites dynamiques hébergés utilise la technologie PHP couplée avec une base de données MySQL ou PostgreSQL.

Avantages

La maintenance du serveur est effectuée par l'hébergeur, ce qui permet à l'administrateur du site de se concentrer sur le contenu des applications et du site internet
Le coût de l'hébergement sur des serveurs mutualisés est faible (< 10 € / mois pour les premières offres) puisque les ressources sont partagées entre utilisateurs. Le prix dépend des ressources souhaitées : taille de la base de données, taille du site internet, bande passante,...

Inconvénients

La configuration du serveur est souvent limitée et certains outils ne peuvent pas être installés (c'est le cas de Java par exemple). Selon la charge et le nombre de consultations simultanées les pages web peuvent être longues à être mises à jour.

9.1.2 - Serveurs mutualisés avec accès total

On dispose d'un serveur complet virtuel qu'il est possible de gérer en totalité à distance via une connexion en ligne de commande sécurisée.
Dans ce cas de figure, il est possible d'installer n'importe quel outils ou services sur le serveurs à distance. Les coûts restent faible (identiques aux serveurs mutualisés de type LAMP). La bande passante peut être garantie selon différents niveaux qu'il est possible de prendre en option. L'hébergeur assure un minimum de maintenance limité à la plate-forme de virtualisation, mais pas sur le serveur virtuel à proprement parlé.

9.1.3 - Serveurs dédiés

Les serveurs dédiés présentent l'inconvénient d'avoir un coût mensuel assez élevé (> 100 € / mois). L'accès est total au serveur. La bande passante est dédiée et garantie. Différentes offres d'assistance peuvent être négociées.

9.2 - Hébergement centre serveur du ministère

Vous trouverez les différentes solutions et plate-formes proposés par le ministère du développement durable sur le site: http://intra.informatique.sg.i2/

On trouve deux offres :

- LAMP au centre serveur de Paris ;
- JEE au CETE du Sud Ouest.

Les application LAMP sont plutôt destinées aux applications de niveau service alors que les applications nationales seront plutôt hébergées à Bordeaux.

9.3 - Hébergement en interne

La ZELT dispose de deux serveurs ainsi que d'un système de stockage de masse permettant de mettre en place rapidement des serveurs virtuels.

Actuellement il est possible sans surcoût de mettre en place un nouveau serveur dédié à la plate-forme d'échanges de données. Cependant il faut prendre en compte une restriction, l'accès au serveur virtuel n'est possible qu'en réseau intranet (via I² et MOREA). Les établissements qui ne sont pas sous MOREA n'auront donc pas accès à ce serveur sauf à demander la mise en place d'une passerelle.

Le débit n'est pas garanti car l'accès est utilisé pour les besoins bureautique du département.

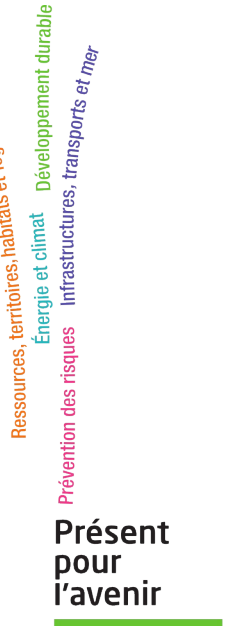
Cependant cette plate-forme est idéale lors des tests car nous avons la totale maîtrise de sa configuration et n'entraîne pas de coût de mise en place supplémentaires.

10 - Bibliographie

Buisson, C et Lesort J.B., Comprendre le trafic routie, CERTU, 2010

Saporta,G., Probabilités, analyse des données et statistique, TECHNIP,2006

Tufféry, S. Data Mining et statistique décisionnelle, TECHNIP, 2007



Centre d'Études Techniques de l'Équipement du Sud-Ouest

rue Pierre Ramond - BP 10
33166 Saint-Médard-en-Jalles Cedex
Tél : 05 56 70 66 33
Fax : 05 56 70 67 33
Courriel : cete-sud-ouest@developpement-durable.gouv.fr