

Titre: Diffusion de statistiques d'achalandage de transport collectif
provenant d'un système de paiement par cartes à puces à l'aide
des technologies XML et SVG
Title:

Auteur: François Vassivière
Author:

Date: 2008

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Vassivière, F. (2008). Diffusion de statistiques d'achalandage de transport collectif
provenant d'un système de paiement par cartes à puces à l'aide des technologies
XML et SVG [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/8275/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8275/>
PolyPublie URL:

**Directeurs de
recherche:** Martin Trépanier
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

DIFFUSION DE STATISTIQUES D'ACHALANDAGE
DE TRANSPORT COLLECTIF PROVENANT D'UN
SYSTÈME DE PAIEMENT PAR CARTES À PUCES
À L'AIDE DES TECHNOLOGIES XML ET SVG

FRANÇOIS VASSIVIÈRE

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

AVRIL 2008



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-41582-5
Our file Notre référence
ISBN: 978-0-494-41582-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DIFFUSION DE STATISTIQUES D'ACHALANDAGE
DE TRANSPORT COLLECTIF PROVENANT D'UN
SYSTÈME DE PAIEMENT PAR CARTES À PUCES
À L'AIDE DES TECHNOLOGIES XML ET SVG

présenté par: VASSIVIÈRE François

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. RIVETTE Mickaël, Docteur, président

M. TRÉPANIÉ Martin, ing., Ph.D., membre et directeur de recherche

M. CHAPLEAU Robert, ing., Ph.D., membre

DÉDICACE

À Catoue, mon amour qui m'a soutenu et encouragé grandement durant toutes les étapes de ce projet. J'aimerais dédier également ce mémoire à ma famille, ma belle-famille et mes amis qui m'ont aussi tous soutenu dans les moments plus difficiles. Merci à tous, du fond du cœur.

REMERCIEMENTS

Je tiens à exprimer mes profonds remerciements à mon directeur de recherche, le professeur Martin Trépanier du département de mathématiques et de génie industriel, pour m'avoir motivé, soutenu et subventionné durant ce projet de recherche au sein de ma maîtrise à l'École Polytechnique de Montréal. Sa grande disponibilité et toutes les connaissances qu'il m'a transmises dans ses domaines d'expertises des systèmes d'informations et du transport collectif ont permis l'accomplissement de cette aventure, ce qui n'aurait pu se faire sans ses conseils méthodologiques et techniques.

Mes remerciements vont également à la Société de Transport de l'Outaouais pour leur aide financière et pour les données de cartes à puces qu'ils nous ont offertes.

Enfin, j'aimerais remercier l'ensemble du département de génie industriel et le groupe Polygistique de l'École Polytechnique de Montréal pour toute l'aide et le support qu'ils m'ont apporté durant les années précédentes ; un merci en particulier à Marie-Hélène Leroux, étudiante au doctorat, pour son soutien et sa compagnie.

RÉSUMÉ

La carte à puce (« smart cards ») agit comme médium efficace en tant que titre de transport, ce système de perception automatisé comporte plusieurs avantages tant pour l'utilisateur que pour l'exploitant dans la gestion quotidienne opérationnelle et financière. Ainsi, les systèmes de perception par cartes à puces en transport collectif servent principalement au contrôle des revenus par une validation du titre de transport des usagers qui empruntent le service. Jumelés à d'autres STI (système de transport intelligents) et des systèmes embarqués, les systèmes de perception automatisés par cartes à puces fournissent une grosse quantité d'informations aussi bien sur le service de transport offert que sur le comportement de ses usagers. Ces données peuvent toutefois être entachées d'erreurs.

La Société de transport de l'Outaouais (STO) s'est lancée en 1997 dans la mise en oeuvre d'un Système informatisé de validation des titres (SIVT) par cartes à puces sur son réseau d'autobus afin d'en contrôler les transactions et les revenus. Le lancement fût effectif en 2001. Des recherches furent déjà réalisées en automne 2003 sur les potentialités d'analyse de ces données par les départements de génie industriel et de génie civil de l'École Polytechnique de Montréal. En 2005, les travaux de recherche se sont poursuivis dans le domaine, en développant un algorithme de dérivation des destinations des usagers, ceux-ci n'ayant pas à valider leurs descentes sur le réseau. Le présent projet s'inscrit donc à la suite de ces travaux.

Il a donc pour principal objectif de modéliser les caractéristiques opérationnelles du réseau de transport de la STO afin de diffuser celles-ci à travers un site intranet à la disposition des planificateurs. La mise en oeuvre de ce projet comprend l'implantation de l'algorithme de dérivation des destinations afin de modéliser des statistiques nécessitant les informations sur les descentes telles la charge à bord des autobus. Ceci implique donc de modéliser de manière formelle toutes les données nécessaires à la dérivation des statistiques désirées tout en respectant les hypothèses et les contraintes qui leur sont

associées. Il s'agit ensuite de choisir une technologie de diffusion flexible et efficace, facilitant les contraintes de la modélisation ainsi que le transfert des données sources à partir du système SIVT.

L'intérêt de cette recherche étant de fournir une aide à la planification en permettant l'analyse directe des statistiques dérivées par les données fournies par ce nouveau Système de transport intelligent (STI). Une validation de ces données, ainsi que celles dérivées par l'algorithme de dérivation des destinations, sont aussi des points abordés dans ce mémoire. Une revue de littérature a été réalisée sur la planification des transports en commun, les diverses approches utilisées, dont la modélisation orientée-objet en transport permettant un traitement fiable et intelligent des données de transport. Les systèmes de transport intelligents, et les travaux sur le traitement des données de cartes à puces, particulièrement dans les domaines du traitement et de l'imputation des erreurs, ainsi que l'analyse du comportement des usagers furent tous étudiés.

Partant d'une validation du modèle-objet du système de transport de la STO, toutes les données nécessaires à la dérivation des statistiques opérationnelles désirées ont été modélisées pour être utilisées au sein de fichiers XML (eXtended Markup Language) structurés, servant de base au contenu du site Web. Une procédure d'extraction quotidienne des données du serveur SIVT en des fichiers XML représentant chacun les activités opérationnelles d'une ligne de transport a été créée. Des patrons XSL (eXtendend StyleSheet Language) de transformation du contenu de ces fichiers XML ont été programmés pour afficher des pages Web au contenu HTML (Hypertext Markup Language) ou SVG (Scalable Vector Graphisc) dépendemment du niveau de précision des statistiques désirées. Il en résulte un site intranet interactif diffusant chaque jour le portrait opérationnel des lignes de transport actives du réseau. Ceci permet aux planificateurs qui utilisent ce site de faire des ajustements au niveau microscopique du service offert en plus de leurs permettre une détection facile de certains types d'erreurs relatives aux données, tout en fournissant des pistes aux sources potentielles de ces erreurs. De plus, des calculs ont été effectués dans le but d'établir des facteurs

d'ajustement de la marge d'erreurs contenues dans les données, vu que certains passagers n'utilisent pas la carte à puces.

ABSTRACT

Smart cards act as a medium effective as a ticket, this automated collection system has several advantages for both the user and the operator in the daily management of operations and finances. Thus, this system of perception in public transit is used primarily to control income of validation of the transport users using the service. Coupled with other ITS (Intelligent Transportation System) and embedded systems, automated collection systems by smart cards provide a large amount of information on both the transport service offered and on the behavior of its users. The data collected can be erroneous.

The Company of transport of Outaouais (STO) launched out in 1997 in the implementation of a Computerized System of Validation of Transport documents (SIVT in French) by smart card on its network of bus having for goal to control the transactions and its incomes. The launch was effective in 2001. Research was being made in autumn 2003 on the potential analysis of this data by the departments of industrial engineering and civil engineering from the École Polytechnique de Montreal. In 2005, research has continued in the field, developing an algorithm to estimate the destinations of users, as their validation is not requested from the descent. This project is therefore as a continuation from result of these previous works.

The main objective is to model the operational characteristics of the transportation system of the STO and to diffuse these statistics through an intranet site available to the planners. The implementation of this project includes the programming of the algorithm that estimate destinations to model requiring statistical information such as the load on buses. This means that the model need to include all the data necessary for the derivation of statistics desired while respecting the assumptions and constraints associated with them. The next step is to choose a technology flexible and efficient enough for data diffusion, facilitating the constraints of the model as well as the transfer of data sources from SIVT system.

The significance of this research was to provide assistance in planning by allowing direct analysis of the statistics derived from data provided by this new intelligent transportation system (ITS). A validation of the data, as well as those derived by the algorithm determining the alighting stops of the users, are also issues addressed in this paper. A literature review was conducted on the planning of public transport, the various approaches, including object-oriented modeling in transportation which allows a reliable and intelligent way to structure data related to transport. Intelligent Transportation Systems, and research made on the treatment of the data from smart cards, particularly in the areas of treatment and charging errors, as well as analysis of the behavior of all users were studied.

Based on a validation of the object model of the transport system of the STO, all the data necessary to derive the desired operational statistics were to be modeled to fit XML (Extensible Markup Language) structure in order to serve as a basis for the content Web site. An extraction procedure to extract daily data from the server SIVT in XML files representing each operational activities of a specific road was established. Some XSL (eXtendend StyleSheet Language) templates were used to transform the content of these XML files to be displayed on the various Web pages in a final format such as HTML (Hypertext Markup Language) and SVG (Scalable Vector Graphisc) depending on the level of precision desired statistics. The result is an interactive intranet site broadcasting each day's operational display of every bus roads active on the network. This enables planners who use this data to evaluate any adjustments that could be made at the microscopic level of service in addition to allow them easily detect certain types of errors in the data, while also providing some guidance on potential sources of these errors. In addition, calculations were performed with the aim of establishing factors for adjusting the margin of errors in the data, given that some passengers do not use the smart card.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	ix
TABLE DES MATIÈRES	xi
LISTE DES TABLEAUX.....	xv
LISTE DES FIGURES.....	xvi
LISTE DES ACRONYMES	xx
LISTE DES ANNEXES.....	xxi
INTRODUCTION	1
CHAPITRE 1: REVUE DE LITTÉRATURE ET CONTEXTE DU PROJET	7
1.1 L'utilisation des données en planification du transport en commun.....	7
1.1.1 Approches classiques	8
1.1.2 L'approche totalement désagrégée.....	13
1.1.3 L'approche orientée–objet	15
1.2 Les systèmes de paiement par cartes à puces au sein des STI	18
1.2.1 Les différents STI utilisés dans le transport en commun	18
1.2.2 Les systèmes de perception automatisés par cartes à puces.....	23
1.2.3 Avantages et Enjeux.....	26
1.2.4 Contexte de la Société de transport de l'Outaouais	31
1.3 Le traitement des données de cartes à puces.....	34

1.3.1	Traitement des erreurs : imputation et enrichissement des données	34
1.3.2	Analyses des comportements des usagers et utilisation en planification.	38
1.3.3	Diffusion des statistiques à l'interne pour la planification.....	45
1.3.4	Avantages et inconvénients, vie privée.....	46
CHAPITRE 2: REVUE TECHNIQUE DES TECHNOLOGIES		49
2.1	Le XML.....	49
2.1.1	Historique.....	50
2.1.2	Exemple d'utilisation	52
2.2	Les DTD et les schémas XSD.....	55
2.2.1	Les DTD.....	56
2.2.2	Les XSD	58
2.3	Transformer les données XML à l'aide du XSL.....	59
2.3.1	Limites du XSL.....	62
2.4	Autres technologies reliées XML.....	63
2.5	Le SVG	64
2.5.1	Définition et historique (Fundamentals of SVG programming)	64
2.5.2	Exemple et utilisation.....	65
2.5.3	SVG et interactivité grâce au ECMA script.....	68
2.6	Technologies d'extraction et de publication	69
2.6.1	De la base de donnée au XML	69
2.6.2	Site Web interactif avec ASP.....	71
CHAPITRE 3: MÉTHODOLOGIE.....		72
3.1	Considérations pour la méthodologie générale du projet.....	72

3.1.1	Données et modélisation objet du système de transport de la STO	74
3.1.2	Présentation et implantation de l'algorithme de dérivation des destinations.....	80
3.2	Présentation de l'approche méthodologique pour le développement d'un outil de diffusion de statistiques	91
3.3	Étapes de modélisation primaire	94
3.3.1	Modélisation des statistiques opérationnelles et leurs hypothèses.....	94
3.3.2	Identification des objets et des tables reliées aux statistiques à calculer .	98
3.4	Détermination de la structure du document XML	101
3.4.1	Structure générale des fichiers XML	102
CHAPITRE 4: MISE EN ŒUVRE DE L'OUTIL DE VISUALISATION		109
4.1	Architecture globale de l'outil de visualisation	109
4.2	Procédure d'extraction des données en fichiers XML	111
4.3	Fonctionnalités du site Web interactif	114
4.3.1	Configuration du serveur IIS et organisation des répertoires.....	114
4.3.2	Utilisation de la page ASP	116
4.3.3	Page des statistiques globales de la journée.....	119
4.3.4	Page des statistiques globales d'une ligne	120
4.3.5	Page SVG interactive sur la charge à bord.....	122
4.4	Analyses statistiques sommaires	130
4.4.1	Détection des erreurs à l'aide de l'outil de visualisation	130
4.4.2	Analyses statistiques sur la ligne 37.....	133
4.4.3	Comparaison des données de comptes à bord avec l'outil créé	136

CONCLUSION.....	140
RÉFÉRENCES.....	144
ANNEXES.....	151

LISTE DES TABLEAUX

Tableau 1-1 Description des 4 classes d'objets de transport (inspiré de Trépanier et Chapleau, 2001)	17
Tableau 1-2 Applications des STI en transport collectif (inspiré de Benko & Mikkelsen, 2005)	19
Tableau 1-4 Avantages des nouveaux systèmes de perception en transport en commun	27
Tableau 2-1: Différences entre le XML et le HTML (inspiré de Morrison, 2002; Bos, B, 1999)	50
Tableau 2-2 Autres technologies reliées au XML.....	63
Tableau 3-1 Champs ajoutés dans la table sivr_transactionbus	86
Tableau 3-2 Modélisation des statistiques opérationnelles.....	95
Tableau 3-3 Liste et description des attributs d'un fichier XML type	105
Tableau 3-4 Description des attributs contenus dans l'élément META.....	106
Tableau A-1 Particularités techniques de base du XML.....	153
Tableau A-2 Types d'attributs dans les DTD (inspiré de Evjen et al., 2007).....	154
Tableau A-3 Éléments XSLT utilisés les plus fréquemment (inspiré de Morrison, 2002)	159

LISTE DES FIGURES

Figure I-1 Travaux de recherche reliant les données de cartes à puces à la planification pour la STO.....	4
Figure 1-1 La procédure séquentielle classique (inspiré de Chapleau, 2000 et Trépanier, 1999).....	10
Figure 1-2 Différences entre l'approche agrégée et l'approche désagrégée (Tranchant 2005).....	12
Figure 1-3 Déplacement selon l'approche totalement désagrégée (Trépanier et al., 2004)	14
Figure 1-4 Concepts de l'orienté - objet (Trépanier et Chapleau, 2001).....	16
Figure 1-5 Types de relations qui unissent les objets (Trépanier, 1999).....	18
Figure 1-6 Illustration du fonctionnement d'un AVL avec GPS.....	22
Figure 1-7 Fonctionnement de la carte à puce sans contact (inspiré de Trépanier, 2004)	25
Figure 1-8 Présentation du territoire de la STO (Tranchant, 2005)	32
Figure 2-1 Indentification d'un patron et application d'un template (inspiré de Lejeune, 2004).....	61
Figure 2-2 Anatomie d'un dessin type en SVG.....	66
Figure 2-3 Dessin SVG d'un rectangle par dessus un rond.....	67
Figure 3-1 Méthodologie générale du projet.....	73

Figure 3-2 Diagramme des interfaces entre les systèmes du SIVT (inspiré de TRÉPANIER 2004).....	74
Figure 3-3 Modèle objet du réseau de transport de la STO	78
Figure 3-4 Différence sur la modélisation objet entre 2007 et 2004.....	78
Figure 3-5 Détermination des destinations «Normales» et «Dernier» à partir des fuites de lignes.....	82
Figure 3-6 Recherche de déplacement pour trouver une destination d'un déplacement «Simple».....	84
Figure 3-7 Statistiques de base sur les distances relatives aux 2 premiers types de descentes.....	90
Figure 3-8 Répartitions des types de descentes estimées en janvier 2007.....	90
Figure 3-9 Méthodologie de correction et d'ajout de contenu à l'outil.....	93
Figure 3-10 Identification des objets et de leurs tables reliées dans le système SIVT ..	100
Figure 3-11 Structure hiérarchique générale des éléments pour une ligne X en direction Y	103
Figure 3-12 Exemple de fichier XML pour la ligne 42 (portions transactions).....	104
Figure 3-13 Exemple de fichier XML pour la ligne 42 (portions info arrêts)	104
Figure 4-1 Architecture globale de l'outil de visualisation des statistiques	110
Figure 4-2 Fonctionnement de la procédure d'extraction des données au format XML	112
Figure 4-3 Paramétrage du logiciel genXML_charges	114
Figure 4-4 Présentation des fichiers du site Web.....	115
Figure 4-5 Page de départ du site Web «genstats_sto.asp»	116
Figure 4-6 Choix du jour dans le mois sélectionné.....	117

Figure 4-7 Choix de l'affichage des statistiques globales de la journée.....	117
Figure 4-8 Sélection d'une ligne – direction pour l'affichage de ces statistiques	118
Figure 4-9 Choix d'un voyage pour l'affichage du profil de charge	118
Figure 4-10 Affichage des statistiques quotidiennes pour toutes les lignes.....	119
Figure 4-11 Moyenne et somme des statistiques globales	120
Figure 4-12 Statistiques globales de la ligne 37 en direction 0 le 2007-01-09	121
Figure 4-13 Profil de charge interactif du voyage 1584574 sur la ligne 37 en direction 0	122
Figure 4-14 Affichage du diagramme de la répartition globale des titres pour tout le voyage.....	123
Figure 4-15 Affichage des statistiques globales sur les correspondances.....	124
Figure 4-16 Choix de l'affichage des montants, descendants ou bien de la charge.....	125
Figure 4-17 Présentation du pourcentage de la distance totale par palier de charge	126
Figure 4-18 - Affichage des statistiques de titres d'un seul arrêt.....	127
Figure 4-19 Affichage des statistiques de validations et correspondances d'un arrêt ...	128
Figure 4-20 Facteur d'ajustement de charge en bout de ligne	129
Figure 4-21 Statistiques anormales observées sur un voyage d'une ligne	131
Figure 4-22 Détection d'erreur: montées en bout de ligne	131
Figure 4-23 Un autre voyage ayant des statistiques potentiellement anormales	132
Figure 4-24 Autres anomalies observées sur un voyage.....	132
Figure 4-25 Diagramme espace-temps de la ligne 37, le 9 janvier 2007.....	134
Figure 4-26 Statistiques quotidiennes pour la ligne 37, janvier 2007.....	135

Figure 4-27 Comparaison des valeurs de montants réelles et ajustées pour un voyage spécifique.....	137
Figure 4-28 Comparaison des valeurs de charge à bord réelles et ajustées pour un voyage spécifique.....	138
Figure A-1 Évolution des langages de programmation Web (inspiré de Képéklian, 2000).....	151
Figure A-2 Aperçu d'un document XML dans le logiciel Internet Explorer.....	152
Figure A-3 Choix de l'affichage du contenu des enfants d'un nœud parent.....	152
Figure A-4 Expression Xpath au sein d'un élément XSLT.....	158
Figure A-5 Modélisation de la couleur du rond suite à un événement «onclick» sur le rectangle.....	163
Figure C-1 Schéma du paquet d'extraction DTS.....	174
Figure C-2 Affichage du graphique sans utiliser le facteur d'ajustement PCENT.....	214
Figure C-3 Affichage du graphique après l'application du facteur d'ajustement PCENT.....	215

LISTE DES ACRONYMES

ASP =	Active Server Page
AVL =	Advanced Vehicule Location Systems
CSS =	Cascading Style Sheets
CTA =	Chicago Transit Authority
DTD =	Documents Type Definition
DTS =	Data Transformation Services
EEPROM =	Electrically Erasable Programmable Read-Only Memory
Enquête O/D =	Enquête Origine Destination
GPS =	Global Positionning System
HTML =	HyperText Markup Language
Km =	Kilomètre
Matrice O/D =	Matrice Origine Destination
MOOT =	Modélisation Orientée – Objet en Transport
Passagers-km =	Passagers * Kilomètres parcourus
SIG =	Systèmes d'Information Géographique
SIVT =	Système informatisé de validation des titres de transport
SQL =	Structured Query Language
STI =	Systèmes de Transport Intelligents
STO =	Société de Transport de l'Outaouais
SVG =	Scalable Vector Graphics
VB =	Visual Basic
XML =	eXtended Markup Language
XSD =	eXtended Schema Defenition
XSL =	eXtended Stylesheet Language

Définition utile:

« Dériver une information » = Déterminer une information à partir d'autres informations

LISTE DES ANNEXES

ANNEXE A:	AJOUTS À LA REVUE TECHNIQUE DES TECHNOLOGIES	151
A.1	Diagramme temporel de l'évolution des langages.....	151
A.2	Visualisation d'un document XML dans un navigateur Web.....	151
A.3	Particularités techniques du XML de base.....	152
A.4	Déclaration des attributs dans les DTD.....	154
A.5	Déclaration des éléments dans les DTD	155
A.6	Exemple complet d'un DTD	156
A.7	Exemple complet d'un schéma XSD	157
A.8	Précisions sur Xpath et les éléments XSLT	158
A.9	Exemple de réaffectation d'une valeur à une variable en XSL.....	160
A.10	Exemple d'interactivité d'un dessin SVG à l'aide d'ECMS script.....	161
ANNEXE B:	AJOUTS À LA MÉTHODOLOGIE DE PROJET	164
B.1	Étapes préliminaires menant à l'algorithme de dérivation des destinations ..	164
B.2	Programmation de l'algorithme de dérivation des destinations.....	166
B.3	DTD et schémas XSD des documents XML utilisés pour le site Web.....	170
ANNEXE C:	COMPOSANTES TECHNIQUES DE L'OUTIL DÉVELOPPÉ	174
C.1	Schéma du paquet d'extraction DTS.....	174
C.2	Exemples de code de l'application d'extraction genXML_charges	175
C.3	Code de la page ASP.....	181
C.4	Code du template XSL «sto_table_global.xsl».....	192

C.5	Exemples de code de la page du graphique interactif SVG	196
C.6	Exemple de l'application de PCENT sur le graphique interactif SVG	214

INTRODUCTION

Les inquiétudes reliées aux changements climatiques et l'augmentation croissante de la population dans les grands centres urbains stimulent les développements technologiques dans le domaine des transports en commun. Ce mouvement favorable est accompagné d'une augmentation graduelle de la clientèle, ce qui crée de nouveaux défis de gestion de l'offre de service pour satisfaire les usagers.

Depuis quelques années, avec l'avènement de l'informatique, les compagnies de transports ont davantage intégré les technologies de l'information aux différentes opérations reliées à la gestion de leur offre de service. Cette offre de service comprend plusieurs étapes commençant par la gestion des horaires jusqu'à la planification. Au Québec, ceci implique historiquement des modèles complexes basés sur des données d'enquêtes origines - destinations. En parallèle, les systèmes d'information aux usagers ont fait leur entrée grâce à l'accessibilité des systèmes d'information géographiques (système d'information géographique (SIG) permettant l'analyse et la visualisation de données spatiales des systèmes de positionnement géographiques (GPS)). Ainsi, les outils aux usagers permettent à ces derniers de vérifier les horaires de service et d'établir leur itinéraire en fonction de leur destination. Ces outils informatiques sont regroupés sous le terme maintenant consacré de « systèmes de transports intelligents » (STI).

Encore plus récemment, certaines sociétés de transport en commun se sont intéressées à la modernisation de leur système de perception des titres. Ainsi, les cartes à puces sont apparues dans le domaine des transports afin de remplacer le paiement par espèces et les billets traditionnels. Ces cartes sont dotées de mémoire et sont en mesure d'interagir avec des bornes informatisées prévues à cet effet. De plus, les cartes à puces permettent aux usagers d'effectuer toutes les transactions sur leur réseau de transport en commun jusqu'à l'épuisement de leur solde ou jusqu'à la fin de leur abonnement, ceux-ci pouvant être renfloués ou renouvelés à diverses bornes conçues pour l'accessibilité des usagers. Ce nouveau type de perception des frais facilite grandement les opérations de gestion

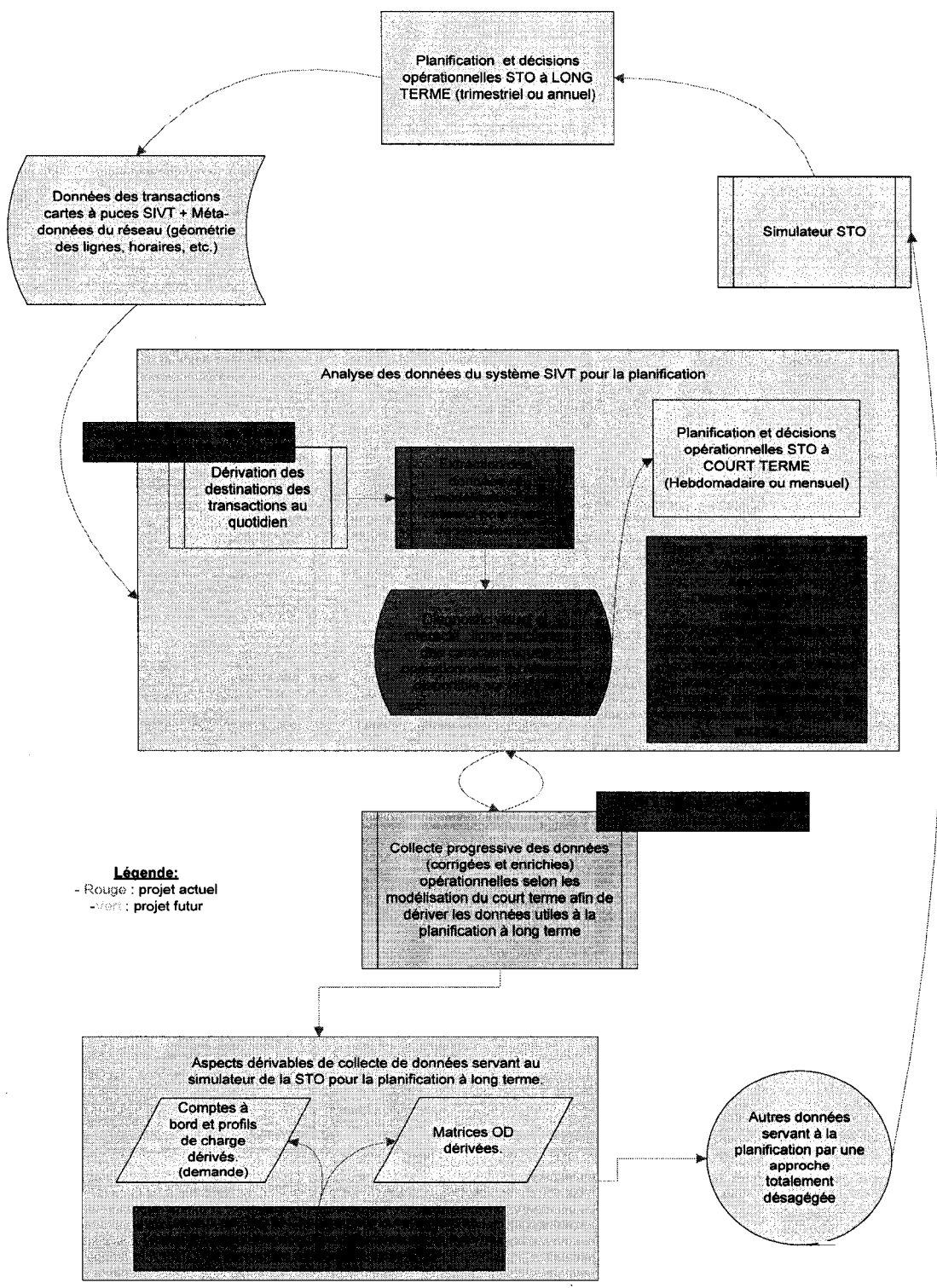
financière des sociétés de transport qui l'utilisent. Par ailleurs, les cartes à puces possèdent des fonctions allant bien au-delà du contrôle de perception. En effet, quelques chercheurs se sont penchés sur le potentiel d'analyse des données recueillies à l'aide de ces cartes.

C'est notamment dans le contexte de la Société de transport de l'Outaouais (STO) qu'ont été effectués deux projets de maîtrise en recherche relativement au potentiel des données de leur système informatisé pour la validation de transactions par carte à puce (SIVT), en place depuis 2001. Le premier projet fut réalisé par Romain Poilpré de 2003 à 2004. Il portait sur le potentiel des données recueillies par le système SIVT de la STO. Les analyses effectuées par M. Poilpré ont permis de modéliser plusieurs statistiques sur les montées, les horaires ainsi que sur les types d'erreurs recensées dans le système (POILPRÉ, 2004). Ensuite, de 2004 à 2005, Nicolas Tranchant a réalisé un projet s'attaquant aux lacunes du système SIVT en termes de génération de données, c'est-à-dire l'absence des destinations reliées à chacune de ces transactions sur le réseau (TRANCHANT, 2005). L'ignorance de la destination des usagers ne permet pas l'élaboration de statistiques opérationnelles plus complexes, lesquelles sont utilisées en planification à court et à long terme. Les travaux de Tranchant ont permis d'élaborer un algorithme permettant de dériver les destinations de la majorité des déplacements effectués durant le mois de juillet 2003. De plus, ces travaux ont permis de raffiner la détection et la classification des erreurs reliées à l'utilisation du système SIVT.

Ces deux travaux de recherche ont donc mené à l'obtention de plusieurs types de statistiques opérationnelles qui peuvent être très utiles à la planification du service. Toutefois, ces données ne peuvent être utilisées concrètement afin d'aider à l'élaboration du service, car il n'existe pas d'intégration de ces résultats au système SIVT actuel. En effet, l'état actuel des choses ne permet pas le calcul des destinations au quotidien ni l'extraction, la modélisation et la présentation visuelle des statistiques opérationnelles des lignes du réseau. Il s'agit donc de la problématique du projet de recherche actuel, soit de proposer une modélisation, des caractéristiques opérationnelles du réseau de la

STO et de faire le montage d'un outil visuel, interactif et partagé qui permet la consultation rapide et simple des résultats statistiques par les personnes concernées, soit en premier lieu les planificateurs.

La Figure I-1 permet de mieux situer l'apport des différents travaux sur l'utilisation des données de cartes à puces de la société de transport de l'Outaouais. Les apports du projet en cours y sont aussi identifiés ainsi que les travaux subséquents qui permettront éventuellement d'utiliser ces données pour de la planification à long terme, soit le plus haut niveau de planification. La prochaine section énumérera les objectifs reliés au projet actuel.



Légende:
- Rouge : projet actuel
- vert : projet futur

Figure I-1 Travaux de recherche reliant les données de cartes à puces à la planification pour la STO

Le principal objectif de ce mémoire est de développer un outil de diffusion de statistiques opérationnelles dérivées des cartes à puces au sein du réseau intranet de la STO. La diffusion de ces statistiques nécessite le prétraitement des données extraites de la base de données opérationnelles, leur analyse et leur mise en forme dans un cadre conceptuel qu'il faut développer. Un sous objectif consiste à vérifier, à l'aide de l'outil de diffusion élaboré, la validité des données de transaction du système SIVT et de l'algorithme de destination développé dans le projet précédant en comparant les résultats avec de véritables comptes à bord fournis par la STO, en plus de fournir une analyse de certaines statistiques opérationnelles pour le mois de janvier 2007.

Pour réaliser les objectifs associés à ce mémoire, il vous est présenté dans un premier temps une revue de la littérature portant sur les aspects opérationnels les plus importants reliés au développement des systèmes de perception par cartes à puces en transport en commun. Nous y retrouvons une introduction au transport en commun et à sa planification et un survol rapide des différentes approches reliées à l'analyse du transport en commun (agrégées, désagrégées, totalement désagrégées et orientée objet). Par la suite, une section de la revue de littérature est consacrée aux différents STI que l'on retrouve en transport en commun avec une attention particulière sur les systèmes de perception automatisé (SPA par carte à puce ou non) et les enjeux qu'ils représentent. Un aperçu de ce qui existe actuellement comme système de diffusion des statistiques opérationnelles à partir des données des SPA sera aussi présenté. Finalement, un survol des différents travaux portant sur l'analyse des données recueillies par les systèmes SPA sera présenté.

Dans le second chapitre, il est question des technologies associées au projet. Cette revue technique couvre la modélisation des données (*Extensible Markup Language* ou XML), leur présentation (*Scalable Vector Graphics* ou SVG, *Extensible Stylesheet Language* ou XSL) en plus de fournir des détails sur les technologies d'acquisition et de diffusion de ceux-ci.

Le chapitre 3 présente la méthodologie générale du projet de recherche. Ces étapes couvrant la modélisation des données du système SIVT au format XML et leur diffusion sont présentées à travers plusieurs exemples axés sur des bases temporelles quotidiennes et sur la base des transactions par lignes du côté opérationnel. De plus, un résumé des travaux de Tranchant sur l'algorithme de dérivation des destinations sera présenté puisque les résultats de celui-ci sont utilisés dans la réalisation technique de ce projet.

Le chapitre 4 présente la mise en œuvre des différentes expérimentations touchant les statistiques opérationnelles: statistiques générales sur les montées, profil de charge à bord illustré sur un graphique interactif et le nombre de passagers - kilomètres. Le site web créé pour l'occasion et ses fonctionnalités y est discuté, encore une fois à l'aide de nombreux exemples. Une comparaison des données de comptes à bord avec les résultats du site web créé permet de valider les données du système SIVT et de juger de la justesse de l'algorithme de dérivation des destinations. Finalement, Des analyses dérivées des données du site Web sont aussi présentées.

CHAPITRE 1: REVUE DE LITTÉRATURE ET CONTEXTE DU PROJET

Dans ce chapitre de revue de littérature, il sera question d'une multitude de sujets reliés au transport en commun en général, mais plus particulièrement aux cartes à puces et leurs technologies affiliées. De plus, une vue d'ensemble des travaux effectués sur le traitement des données recueillies par ces systèmes de perception automatisés sera aussi commentée.

1.1 L'utilisation des données en planification du transport en commun

Il est bien important d'expliquer les fondements de la méthodologie qui est utilisée dans ce projet de diffusion des statistiques opérationnelles du réseau de transport de la STO. Nous verrons plus loin qu'il s'agit de la modélisation orientée – objet en transport (MOOT) qui est utilisée à la base de la méthodologie du projet, il faut donc expliquer pourquoi nous avons choisi d'utiliser cette approche.

Tout d'abord, il faut déterminer quel type d'utilisation des données en planification du transport en commun cadre le mieux avec les données qui sont recueillies par le système de perception par carte à puces de la STO. Le système en question récolte des quantités colossales de données transactionnelles provenant de l'utilisation du réseau par ses usagers. Ces données sont de nature spatiale et temporelle en plus d'être reliées à un individu en particulier, sans connaître toutefois son identité. Il faut donc une approche particulière afin d'analyser ces données. L'approche totalement désagrégée est idéale dans ce cas, puisqu'elle traite les déplacements de façon individuelle en plus de tenir compte des aspects spatio-temporels. C'est d'ailleurs l'approche qui est utilisée lors de l'analyse des enquêtes origine – destination du grand Montréal à des fins de planification (Trépanier et al., 2004).

Nous avons donc une approche d'utilisation de nos données, qui est totalement désagrégée. Comment peut-on concrètement l'appliquer maintenant? C'est là qu'intervient la modélisation orientée–objet qui est, si l'on veut une méthode d'application du traitement totalement désagrégée puisqu'elle s'inscrit dans la continuité de celui-ci. En effet, la modélisation orientée–objet d'un réseau de transport apporte un cadre structurel aux données et aux procédures de traitement. De plus, il s'agit de la méthode parfaite à utiliser dans un environnement fortement informatisé comme celui de la STO, particulièrement parce qu'elle permet de modéliser les données sans imposer de restrictions par rapport à un langage informatique particulier (Trépanier et Chapleau, 2001). Le chapitre 3, sur la méthodologie et la modélisation des données, va démontrer ce point de vue. En utilisant la modélisation des objets de transport du réseau de la STO on pourra valider le passage des données organisées dans leur base de donnée SIVT vers un fichier XML structuré pour le traitement d'informations désiré dans notre site Web.

C'est pourquoi il est donc important d'expliquer au lecteur les fondements de ces deux approches en commençant par les approches classiques qui ont mené à l'approche totalement désagrégée et l'approche orientée–objet.

1.1.1 Approches classiques

Dans cette section il sera question des différents axes de modélisation en transport qui ont mené à l'approche totalement désagrégée et orientée–objet. Il s'agira premièrement de l'approche agrégée, qui, malgré ses lacunes, est encore utilisée aujourd'hui en planification ainsi que de l'approche désagrégée se rapprochant plus de la réalité en permettant l'analyse des déplacements individuels.

1.1.1.1 L'approche agrégée

Les méthodes utilisées dans l'approche agrégée découlent de l'ère pré-informatique et les premiers logiciels qui ont été créés pour faciliter la tâche des planificateurs étaient donc dérivés directement des procédures papiers existantes (Trépanier, 1999).

La procédure séquentielle classique est la meilleure représentation de l'approche agrégée. Elle permet d'estimer la demande en transport pour un territoire donné et de simuler les effets engendrés par les déplacements sur le réseau. La figure suivante présente les cinq grandes phases de la procédure séquentielle classique.

Procédure séquentielle classique

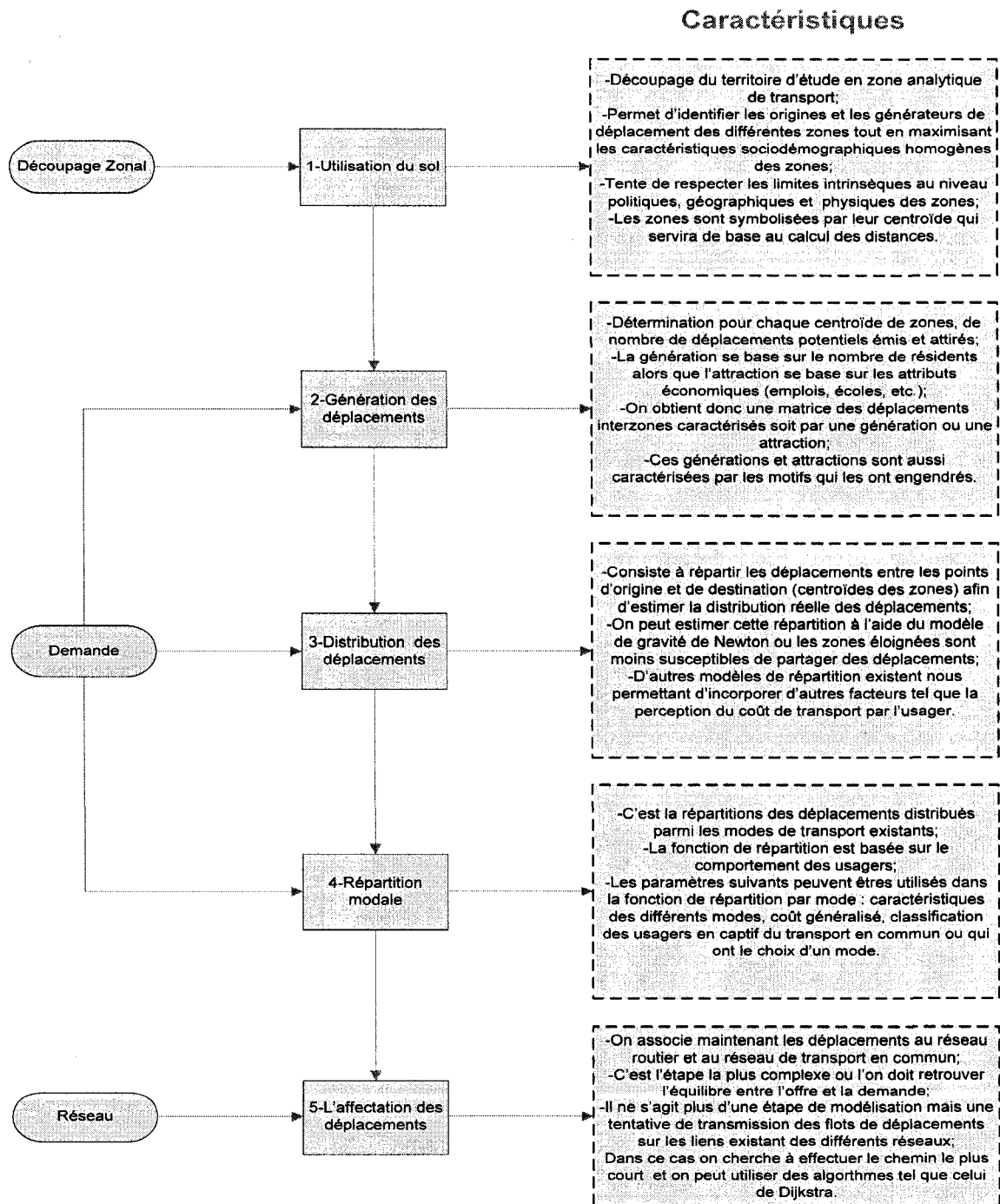


Figure 1-1 La procédure séquentielle classique (inspiré de Chapleau, 2000 et Trépanier, 1999)

Notons qu'à la suite de cette procédure, on peut aussi regarder les impacts au niveau de l'évaluation économique des résultats du processus en s'attardant, entre autres, aux bénéfices que peuvent en retirer les usagers (gains de temps et coûts) ainsi qu'aux coûts perçus par les opérateurs et les investisseurs.

L'approche agrégée (reliée à l'approche séquentielle classique) fait toutefois l'objet de critiques. Trépanier en recense l'essentiel dans sa thèse en 1999 ainsi que Tranchant dans son mémoire de 2005 :

- l'essence de la modélisation agrégée est remise en cause dans la littérature. On doute de l'adaptabilité de la méthode face aux changements du milieu, ne pouvant ainsi apporter des alternatives aux problèmes de planifications;
- on juge, entre autres, la structure de modélisation trop rigide, citant, par exemple, l'utilisation de taux fixe (donc ne prenant pas compte des caractéristiques zones et régions étudiées) dans l'étape de génération des déplacements;
- on reproche à l'étape de répartition modale de ne faire que répartir les déplacements entre les modes sans offrir d'interaction avec les étapes précédentes;
- le choix du système de zone, donc le découpage, a une très forte influence sur les résultats obtenus au cours des étapes subséquentes. Il n'existe pas de vraies méthodes composées de règles claires qui guident le coupage. Tout repose donc sur l'expérience;
- l'approche par agrégation (fonctions) est aussi sujette à des critiques. Les données agrégées peuvent masquer les vraies relations et donc ne pas refléter les changements socioculturels (et économiques) à travers le temps;
- d'autres portent sur le choix des variables de comportement et l'imprécision des résultats.

Toutes ces critiques ont grandement stimulé les travaux sur le développement d'une autre méthodologie abandonnant la base d'observation au niveau des groupes d'utilisateurs et des divisions zonales pour se pencher sur les déplacements individuels et leurs attributs.

1.1.1.2 L'approche désagrégée

L'approche désagrégée se rapproche plus de la réalité et permet de corriger les erreurs et critiques liées à l'approche agrégée en caractérisant les déplacements et leurs attributs de façon individuelle pour chacun des individus. La figure suivante montre bien la différence entre les deux approches, soit la perte d'information liée à l'agrégation des données avant l'utilisation de celles-ci jusqu'à la fin de l'application des procédures.

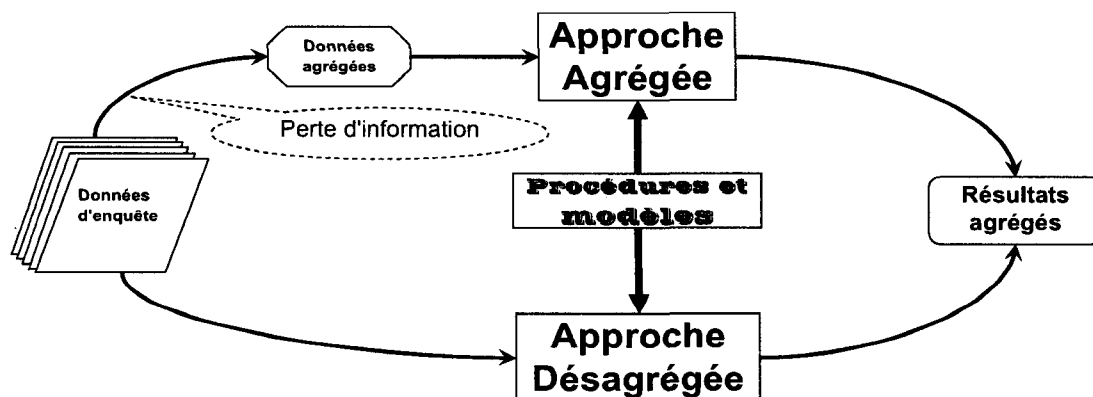


Figure 1-2 Différences entre l'approche agrégée et l'approche désagrégée (Tranchant 2005)

Partant des enquêtes origines-destinations de type ménage (recensement d'informations socio-démographiques, i.e., sexe, âge, statut), la méthode générale effectue l'adéquation de plusieurs alternatives de transport. On offre à l'utilisateur l'alternative entre, par exemple, la nécessité de se déplacer ou non, de prendre l'auto ou non, etc. Chacune de ces alternatives sont évaluées en factorisant le vecteur des valeurs quantitatives des attributs de l'alternative par le vecteur des valeurs socio-démographiques de l'utilisateur étudié. Une fois la procédure terminée, on obtient plusieurs matrices origine-destination qui peuvent être cumulées pour servir au processus d'affectation sur le réseau de transport (Trépanier, 1999).

Les plus grands modèles basés sur l'approche décrite précédemment, soit le modèle de répartition nominale et celui de répartition hiérarchique, sont également sujets à des critiques. Premièrement, ils supposent la connaissance de toutes les données qui révèlent les préférences des utilisateurs quant aux choix de leurs alternatives de transport. Deuxièmement, ces deux modèles utilisent des coefficients (relatifs aux attributs des alternatives) fixes ou basés sur des moyennes qui ne pourront tenir compte des variations d'appréciation des individus (préférences) face à leurs alternatives de déplacements (Tranchant, 2005). Finalement, on juge ces méthodes encore trop mathématiques, ne s'attardant pas encore à la compréhension des phénomènes extérieurs et de leurs potentielles corrélations avec le problème.

1.1.2 L'approche totalement désagrégée

L'approche totalement désagrégée décortique davantage les quantités colossales d'informations disponibles grâce aux enquêtes origines-destinations. En effet, cette approche scinde en 3 groupes de variables les éléments du déplacement individuel (Chapleau, 1986):

1. les variables sur le ménage : le nombre de personnes, le lieu de domicile, la possession d'une ou plusieurs automobile, etc.;
2. les variables sur la personne : on parle ici des renseignements sur l'âge, le sexe, le statut, le revenu, la possession du permis de conduire, etc.;
3. les variables sur le déplacement : on recense ici, l'origine, la destination, l'heure de départ, les modes utilisés, les lignes et les routes utilisées, les points de jonctions, etc.

De plus, cette approche vient révolutionner les approches classiques en faisant fi des découpages territoriaux, soutenant plutôt l'utilisation des systèmes d'information géographiques. En effet, ces systèmes permettent la localisation fine et parfois unique des extrémités des déplacements observés, car ils peuvent, entre autres, fournir les coordonnées X et Y des générateurs de déplacements. Par exemple, les grandes enquêtes

origines–destinations de la région de Montréal de 1987 et 1993 ont utilisé les codes postaux afin de géo-référencer les diverses origines et destinations des déplacements, alors que les enquêtes subséquentes ont caractérisé les localisations de façon encore plus précise en utilisant les coordonnées des adresses civiques (Tranchant, 2005).

On peut donc maintenant schématiser de façon très précise les déplacements individuels des différents usagers avec l'approche totalement désagrégée. La figure suivante démontre bien cette définition du déplacement selon cette approche.

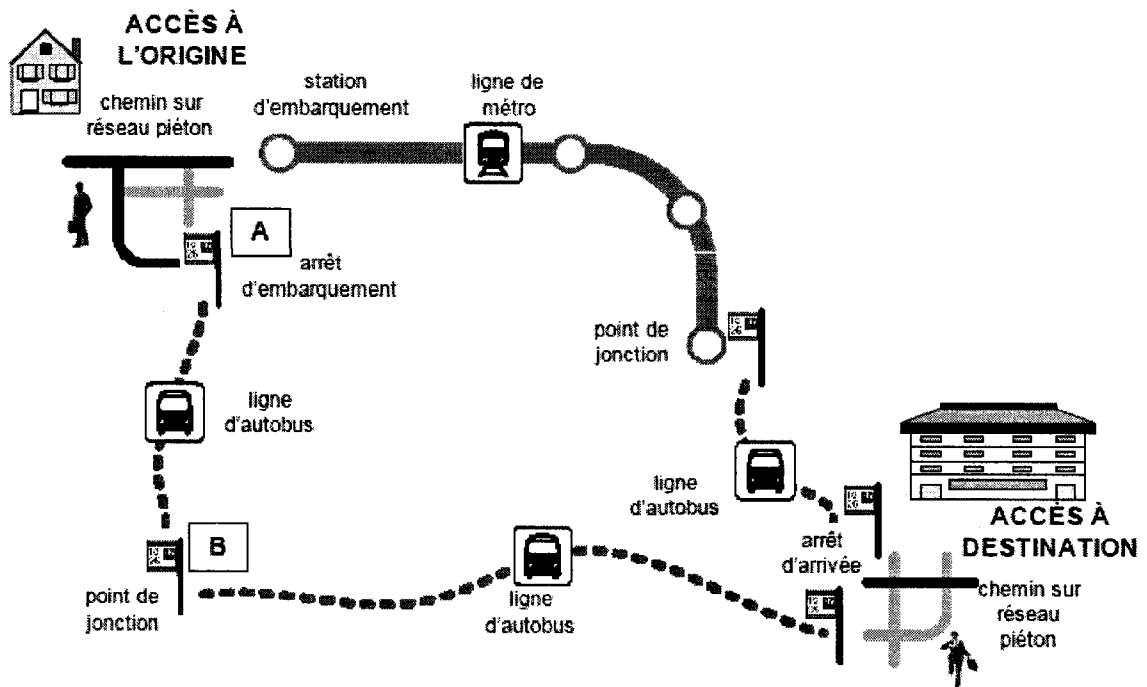


Figure 1-3 Déplacement selon l'approche totalement désagrégée (Trépanier et al., 2004)

Partant de l'accès à l'origine, l'utilisateur accède à son arrêt ou à sa station d'embarquement par le réseau piétonnier. Il emprunte alors une ou plusieurs lignes du système de transport en commun, passant ainsi par des points de jonction (correspondance), au besoin, afin d'atteindre sa destination. Le système de perception par cartes à puces permet d'enregistrer les transactions (et tous leurs attributs, tel que l'heure, le numéro d'arrêt et ses coordonnées X, Y, la direction, etc.) de montée (A) et de correspondance (B) effectuées par l'utilisateur.

Pour résumer, on peut citer la récapitulation des éléments désagrégés à partir d'une matrice d'enquête origines–destinations rapportée dans la thèse de Trépanier en 1999 :

- désagrégation des caractéristiques socio – démographiques des déplacements;
- désagrégation de la chronologie;
- désagrégation des itinéraires;
- désagrégation des flots;
- désagrégation des modes.

De plus, il mentionne que cette approche permettra de dériver de nouvelles informations sur l'individu et ses déplacements, comme ce sera le cas avec les travaux effectués sur la dérivation des destinations permettant maintenant de savoir l'arrêt de descente de l'utilisateur.

Finalement, l'approche totalement désagrégée a d'ailleurs été à la source de plusieurs travaux de recherche du groupe MADITUC (Modèle d'Analyse Désagrégée des Itinéraires de Transport Urbain Collectif) de l'école Polytechnique de Montréal, qui ont développé toute une suite de logiciels servant à effectuer des analyses en planification telle que : prestation du service, extension des réseaux de métro et de train, financement, prévisions de la demande, etc. (Trépanier et al., 2004).

1.1.3 L'approche orientée–objet

L'approche orientée–objet en transport vise la transposition des concepts de la programmation orientée–objet en informatique à l'analyse de ce que l'on peut retirer des enquêtes origines–destinations. C'est aussi l'extension de l'approche totalement désagrégée, mais adapté aux technologies informatiques qui gèrent des interactions complexes au sein de systèmes transactionnels bien structurés comme un système de perception par cartes à puces, par exemple. On peut aussi dire que l'approche orientée–objet est un moyen d'appliquer concrètement l'approche totalement désagrégée.

Les concepts

En orienté-objet, tous les objets sont associés à une classe et deviennent des instances de celle-ci. Chaque classe contient également des propriétés (attributs) ainsi que des méthodes qui leur permettent d'effectuer des actions. Chaque instance d'objets d'une classe aura donc ses propriétés et ses méthodes. De plus, il est possible de créer des sous-classes à partir du principe de l'héritage afin de généraliser les méthodes à travers les classes de même famille. En informatique (programmation C++ par exemple), ces concepts sont grandement utilisés, car ils permettent d'approcher le plus fidèlement possible la réalité lors de la résolution de problèmes. La figure suivante permet de mieux illustrer ces concepts (Trépanier et Chapleau, 2001).

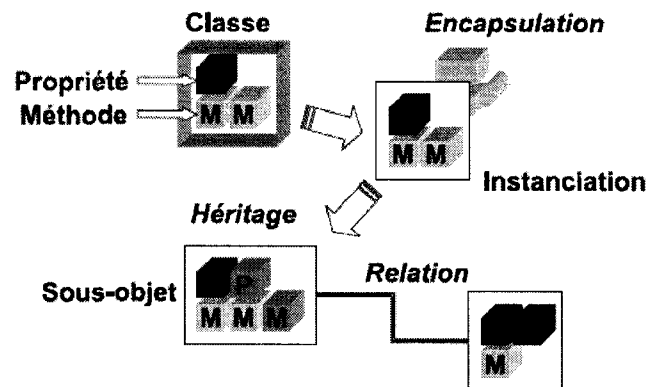






Figure 1-4 Concepts de l'orienté - objet (Trépanier et Chapleau, 2001)

Les métaclasses et les relations

La transposition de ces concepts dans le domaine de la planification des transports en général ainsi que du traitement des données d'enquêtes origine-destination a mené à l'identification de quatre grandes classes d'objets (Trépanier et Chapleau, 2001) au sein de la modélisation orientée-objet en transport (MOOT). Le tableau suivant explique chacune de ces métaclasses :

Tableau 1-1 Description des 4 classes d'objets de transport (inspiré de Trépanier et Chapleau, 2001)

<i>Classe</i>	<i>Description</i>	<i>Exemples</i>
	-Ils sont les acteurs du transport. Ces objets « décident » et contribuent à leur mouvement. Ils représentent tantôt un groupe de personnes, un objet mobile ou une masse déplacée.	<ul style="list-style-type: none"> • Autobus; • Train; • Métro; • Personne; • Ménage.
	-Ils sont les descripteurs du mouvement.	<ul style="list-style-type: none"> • Itinéraire; • Ligne TC; • Voyage; • Rue.
	-Ils possèdent une localisation fixe dans le temps et l'espace. Leur rôle est de décrire le territoire et servir de balise à l'espace utilisable pour fins de transport. Ils supportent donc les activités de transport.	<ul style="list-style-type: none"> • Arrêt; • Station de métro; • Municipalité; • Secteur; • Générateur.
	-Ils sont des groupes d'objets interreliés. Ils peuvent être opérationnels, informationnels ou multifonctionnels.	<ul style="list-style-type: none"> • Réseau TC; • Enquête OD; • Système urbain; • Région.

Lors de la modélisation d'un réseau de transport, il y aura plusieurs instances de ces 4 grandes classes qui seront créées. Plusieurs types de relations pourront lier ces différents objets. La figure suivante démontre tous les types de relations qui peuvent être utilisés pour lier 2 objets différents.

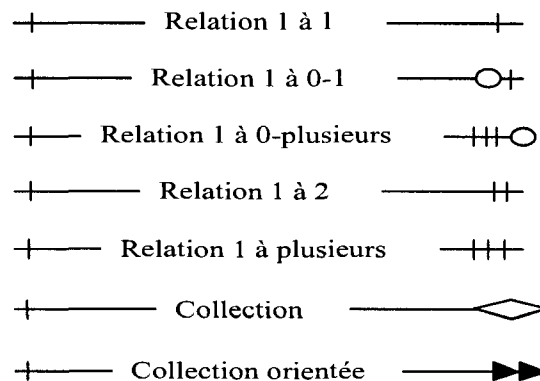


Figure 1-5 Types de relations qui unissent les objets (Trépanier, 1999)

La prochaine section traite des différents systèmes de transport intelligents (STI) utilisés en transport en commun et plus particulièrement des systèmes de perception par cartes à puces.

1.2 Les systèmes de paiement par cartes à puces au sein des STI

Les systèmes de transport intelligents (STI) sont des systèmes qui utilisent l'électronique, les communications et le traitement d'information afin d'améliorer l'efficacité et la sécurité des transports terrestres (Benko et Mikkelsen, 2005). Nous verrons dans cette section les différents STI qui sont utilisés dans le domaine des transports en commun et plus particulièrement, au sein de ceux-ci, les systèmes de paiement par cartes à puces. Nous terminerons la section par une présentation du contexte de la société de transport de l'Outaouais.

1.2.1 Les différents STI utilisés dans le transport en commun

Comme dans tous les domaines, l'importance de la technologie ne peut être négligée quant à l'avenir du transport public. Les sociétés de transport seront toujours alignées sur l'amélioration du service offert tout en diminuant les coûts associés. Les avancées technologiques pourront aider les sociétés de transport à être les plus efficaces possibles avec leurs ressources disponibles. L'augmentation de la couverture des réseaux ainsi que celle du nombre de véhicules en opération sont aussi des facteurs incitatifs à

l'automatisation de certaines procédures qui permettront un meilleur monitoring des réseaux de transport public.

Il y a donc une panoplie d'applications des STI en transport en commun que les sociétés de transport public peuvent décider d'adopter afin de les aider à mieux gérer leur réseau. Évidemment, il y a toute la question des coûts, des bénéfices et des risques qui doit être évaluée avant d'entreprendre l'implantation de tels systèmes, qui selon Gillen et al. (2001) peut être une procédure complexe pouvant prendre quelques années.

Le tableau suivant présente la description des principales applications des STI dans le domaine du transport en commun. Le cas des AVL (Advanced Vehicle Location system) sera discuté plus précisément par après.

Tableau 1-2 Applications des STI en transport collectif (inspiré de Benko & Mikkelsen, 2005)

<i>Technologie</i>	<i>Description</i>
1- Les systèmes de perception automatisés	Les systèmes de perception automatisés utilisent soit les cartes à puces, des cartes magnétiques ou bien une combinaison d'un type de carte et d'argent comptant (billets). Ces systèmes facilitent grandement la perception des titres de transport et sont couverts amplement dans le reste de ce chapitre.
2- Les systèmes d'information aux voyageurs	<p>Les systèmes d'informations aux voyageurs peuvent fournir des renseignements statiques ou en temps réel aux usagers du réseau. Les informations en temps réel sont les plus convoitées et peuvent afficher les informations suivantes :</p> <ul style="list-style-type: none"> • conditions de route, de congestion et d'accidents, changeant minute par minute; • conditions du réseau (train et métro), informations sur les pannes; • changement de trajet pour les raisons ci-dessus; • information sur l'horaire et l'heure de passage des autobus; • identification du prochain arrêt de métro ou d'autobus; • estimation du temps d'arrivée à destination dépendamment de l'heure de la journée; • Information sur les correspondances à prendre en chemin et leurs départs. <p>De plus, on peut retrouver des systèmes d'aide aux chauffeurs pour des routes congestionnées ou pour le guider dans ses déplacements. Des systèmes d'informations sur les attractions touristiques peuvent aussi être installés.</p>

Tableau 1-2 Applications des STI en transport collectif (inspiré de Benko & Mikkelsen, 2005) (suite et fin)

3- Les systèmes de sécurité	<p>Plusieurs crimes peuvent se produire sur un réseau de transport public et les usagers considèrent que leur sécurité est la responsabilité de la société en question. Plusieurs technologies peuvent être utilisées afin de réduire l'incitation aux crimes ou bien pour intervenir plus rapidement possible:</p> <ul style="list-style-type: none"> • caméras et télévisions en circuit fermé; • téléphones d'urgence; • alarmes; • systèmes de perception automatisés; • les Advanced Vehicle Locations Systems (AVL); • signes d'urgences sur les véhicules. <p>En plus de cela, il y a bien sûr les patrouilles et le renforcement des conséquences en cas de délit qui peuvent aider à prévenir les crimes dans le transport en commun.</p>
4- Les systèmes de signalisation prioritaire	<p>Les systèmes de signalisation prioritaire permettent de donner un traitement de faveur aux autobus qui transportent plus de passagers que les autres véhicules. Ces priorités peuvent être passives, (voies réservées à certaines heures) ou actives avec des détections d'approche aux intersections, afin de faciliter le passage prioritaire des autobus. Il y a aussi des feux de signalisation prioritaire pour les autobus à certaines intersections-clés qui permettent aux autobus de partir les premiers.</p>
5- Les systèmes d'arrimage de précision	<p>Ces systèmes permettent aux véhicules de s'arrêter toujours au même endroit à chaque arrêt afin de minimiser le temps de chargement des passagers de même que pour faire monter les personnes à mobilité réduite à bord des autobus. De plus, on minimise ainsi les dommages potentiels fait aux véhicules lors de l'approche des arrêts.</p>
6- Les systèmes informatisés de gestion de flotte	<p>Aussi appelés «Fleet management systems» ou «Computer-aided Dispatch (CAD)», ces systèmes intègrent les données opérationnelles de transport (horaires, opérations, planifications) afin de faciliter le travail des planificateurs et des superviseurs dans leurs décisions. Couplé à un AVL, ils permettent le suivi en temps réel des véhicules pour intervenir le plus rapidement possible avec le chauffeur qui contacte la centrale en cas de problème.</p>

Les AVL (*Advanced Vehicle Locations Systems*) méritent une attention particulière, car ils sont à la base de plusieurs des applications STI présentées dans le tableau précédent. En effet, cette technologie est considérée comme faisant partie des systèmes

informatisés de gestion de flotte, lesquels se concentrent sur l'amélioration des opérations des véhicules, les efforts sur la planification et la sécurité. De plus, elle servira aux systèmes de perception automatisée qui utilisent la localisation d'arrêts de montées. Le tout sera traité plus amplement dans la section suivante. En sachant en temps réel la location exacte (dépendamment des technologies utilisées) de leurs véhicules sur leur réseau, la firme est en mesure d'avoir de l'information claire et objective sur chacun de ces autobus (Gillen et al., 2001).

Il existe quatre technologies de navigation principales qui peuvent être utilisées à la base des AVL, elles sont présentées dans la liste ci-dessous (Benko et Mikkelsen, 2005):

1. le «Dead reckoning» : Cette technologie de navigation est parmi les plus vieilles. Elle détermine la location d'un véhicule en mesurant la distance parcourue à partir d'un point connu et la direction du véhicule. Les capteurs de «Dead reckoning» peuvent mesurer la distance et la direction à partir d'un point fixe (pour un arrangement de base, un odomètre et un compas peuvent être utilisés pour calculer la position à partir d'un arrêt particulier d'une ligne). Habituellement, ces systèmes servent de recours en cas de panne du système GPS par exemple. Les limites de cette technologie sont au niveau de la précision qui va diminuer avec un terrain non nivelé ainsi qu'avec l'augmentation de la distance parcourue.
2. le «signpost/odometer» : Cette technologie de navigation a été l'une des plus populaire avant l'avènement des systèmes GPS. Elle utilise des balises de retransmission le long du trajet du véhicule. Celui-ci communique, à l'aide d'une onde radio courte portée, la valeur de son odomètre à chaque balise rencontrée afin de déterminer la position du véhicule. Le défaut principal est le coût d'installation et d'entretien très élevé à mesure que le réseau s'agrandit. De plus, dès que le chauffeur doit dévier du parcours on perdra la trace du véhicule.
3. la «radio navigation/location» : les systèmes de radio – localisation utilisent des signaux à basse fréquence pour communiquer avec l'autobus et ainsi le localiser.

L'interférence provenant des lignes de hautes tensions ou de stations de transformateurs peut nuire à la réception du signal.

4. les systèmes de localisation GPS : Cette technologie de localisation est maintenant la plus populaire et la plus fiable. Les systèmes GPS installés à bord des autobus reçoivent des signaux en provenance de satellites en orbite autour de la terre afin de localiser ceux-ci. La figure suivante illustre bien le fonctionnement d'un AVL utilisant le GPS.

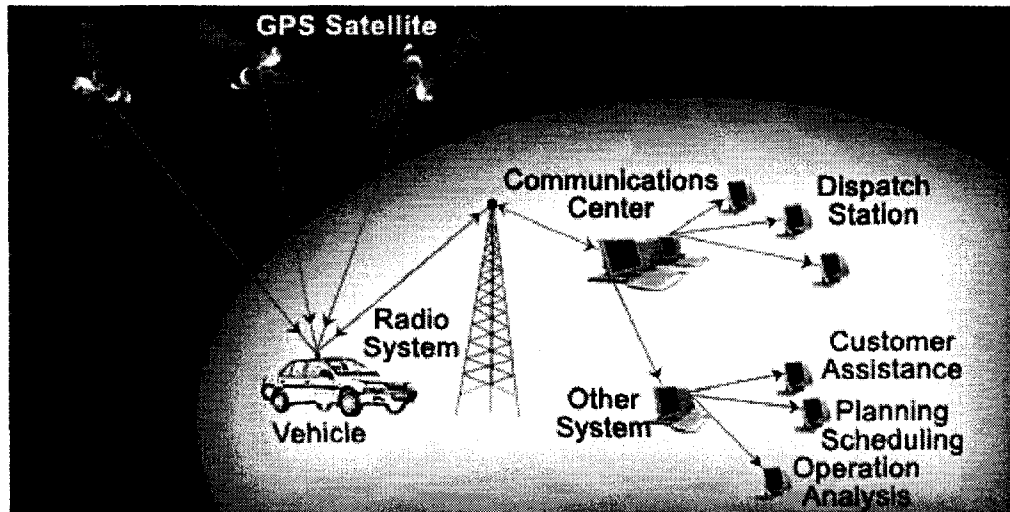


Figure 1-6 Illustration du fonctionnement d'un AVL avec GPS¹

Une fois les coordonnées cartésiennes recensées par les satellites, celles-ci sont envoyées à un centre de communication par ondes radio, puis sont retransmises aux différents postes de travail requis par Intranet ou Internet.

Cette technologie apporte bien des avantages aux compagnies de transport qui les utilisent; Gillen et al. (2001) ont élaboré la liste suivante :

¹ Tiré du site Web nommé "Automatic Vehicule Location Advantage", Roseindia Disponible : <http://www.roseindia.net/technology/vehicle-tracking/automatic-vehicule-location-advantage.shtml>

- augmente l'efficacité de la répartition des ressources opérationnelles;
- offre un service plus fiable;
- donne une réponse plus rapide aux interruptions de service;
- augmente la sécurité du chauffeur et de ses passagers (alarmes silencieuses avec la localisation précise des véhicules);
- réponds plus rapidement aux pannes mécaniques et permet ainsi de réduire les coûts de maintenance;
- offre une meilleure qualité de données automatiquement collectées pour l'agence à moindre coût.

Les deux auteurs ont aussi réalisé une étude sur la rentabilité, en terme de productivité, que les AVL sont susceptibles d'apporter aux sociétés de transport public. Ils ont conclu que l'utilisation des AVL augmentait la productivité, et ce, autant en mesurant le rendement en terme de paramètres-passagers ou bien de paramètres de service. Les auteurs mentionnent de plus que les sources de ces gains de productivité proviennent d'une meilleure utilisation du capital, d'une diminution du nombre d'autobus requis et d'une utilisation plus efficace du carburant (Gillen et al., 2001).

1.2.2 Les systèmes de perception automatisés par cartes à puces

Les systèmes de perception automatisés par cartes à puces fonctionnent à l'aide d'un système central de base de données, de lecteurs installés à bord des véhicules et de cartes à puces fournies aux usagers. Commençons d'abord par donner un peu d'information sur les cartes à puces en tant que telles.

La carte à puce («smart card») est une technologie qui remonte à la fin des années 1960. L'idée d'une carte possédant une puce informatique a été brevetée en 1968 par les chercheurs allemands Dethloff et Grötrupp (Shelfer et Procaccino, 2002). Les Japonais, en 1970, ont eux aussi fait breveter une carte. La première puce destinée à une utilisation «commerciale» est le fruit des travaux de Motorola et Bull en 1977.

Parmi les cartes que l'on retrouve aujourd'hui, il y a les «memory cards» et les «processors – enabled cards». Les cartes munie de micro processeurs sont les vraies «smart cards», puisqu'elles permettent le traitement, l'encryptions et l'écriture de données sur leur mémoire EEPROM (electrically erasable programmable read-only memory), tandis que les cartes de mémoire ne possèdent que du hardware leur permettant d'être utilisées en lecture seule.

Pour ces deux types de cartes, il y a deux types d'interface possible avec lesquels elles pourront communiquer avec d'autres dispositifs. On choisit donc parmi ces 2 types de technologie de communication dépendamment de l'application pour laquelle la carte est destinée (Blythe, 2004) :

1. Les cartes à puces avec contact : Aussi appelées en anglais «contacts cards», celles-ci fonctionnent avec un jeu de 6 ou 8 connecteurs de métal que l'on retrouve à la surface de la carte. Lorsque celle-ci est insérée dans un appareil de lecture, des contacts montés sur ressorts établissent une connexion avec chacun des connecteurs de la carte. Les meilleurs exemples sont ceux de la carte téléphonique et de la carte bancaire. Ces cartes sont sujettes à la norme ISO 7816.
2. Les cartes à puces sans contact : Aussi appelée en anglais «contactless cards», celles-ci établissent un signal avec un lecteur à l'aide d'ondes radio (principe des RFID, « radio frequency identification ») ou infrarouges. Par contre la carte doit être placée tout près du lecteur pour qu'il y ait une communication entre les deux. Ces cartes sont sujettes à la norme ISO 14443.

Shelfer et Procaccino (2002) ont identifié 3 applications principales de la carte à puces. Il y a la simple authentification (identification d'une personne), l'autorisation (vérification de la validité de cette personne dans une base de données), et finalement la transaction (prélèvement d'un paiement ou enregistrement d'une transaction sur la carte, comme par exemple en transport en commun).

On peut terminer la section en expliquant le fonctionnement de la perception de carte à puce en prenant l'exemple du réseau de transport de la STO, qui utilise des cartes avec micro processeur de type «contactless» (figure 1.7). Tout d'abord, chaque usager possède une carte (avec photo) identifiée à son nom. Les puces des cartes possèdent des données sur le type de titre de transport de l'utilisateur ainsi que sa date de validité. Lorsqu'un usager monte à bord d'un autobus, il passe sa carte devant un lecteur. Celui-ci transmet un signal d'excitation à la carte, qui lui retourne alors ses données. Le lecteur indique ensuite à l'aide d'un signal sonore et lumineux si la carte de l'utilisateur est valide pour la ligne et la direction qu'il a choisi d'emprunter (Trépanier, 2004).

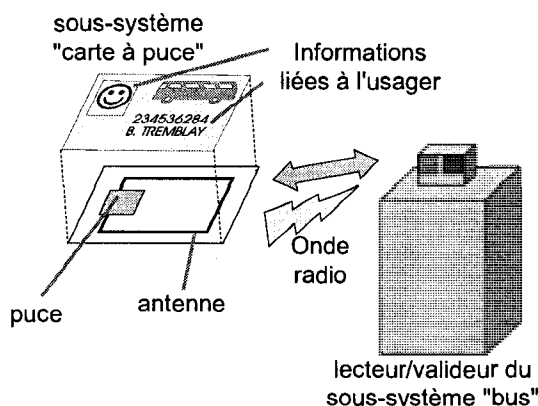


Figure 1-7 Fonctionnement de la carte à puce sans contact (inspiré de Trépanier, 2004)

Pour effectuer leur travail de validation, les lecteurs à bord des autobus sont reliés à un micro ordinateur sur lequel les informations des tracés et des voyages sont chargées. Ces données proviennent du système opérationnel de la STO (HASTUS) et passent par le serveur central SIVT. Les explications sur les éléments non visibles du système de perception, et par conséquent, le fonctionnement du système central et ses relations avec les autres systèmes sont disponibles à la section 3.1.1.1 de ce mémoire.

Notons en terminant que le système de perception de la STO enregistre aussi l'arrêt de montée de l'utilisateur à l'aide de son AVL fonctionnant par GPS. Ceci est un autre avantage de l'application de ce STI permettant de comparer les coordonnées GPS de

l'autobus avec la base de données des arrêts géocodés dans le micro-ordinateur du véhicule. Un AVL n'utilisant pas le GPS comme technologie de localisation ne serait pas assez précis pour localiser les arrêts de montée. Les données que la combinaison de ces STI nous permet de collecter sont donc encore plus précises et plus riches en potentiel d'exploitation.

1.2.3 Avantages et Enjeux

Cette section traite des systèmes de perception automatisés en transport en commun, principalement pour les systèmes supportant les cartes à puces ou un autre type de technologie, lorsque mentionné dans la littérature. Plutôt que de mettre l'emphase sur les aspects plus techniques d'un tel système, il s'agit ici de répertorier les différents enjeux auxquels les compagnies de transports font face s'ils choisissent de faire l'adoption de cette technologie. En premier lieu, les faits rapportés ci-dessous proviennent d'une étude réalisée par Dinning et Collura (1995) auprès des sociétés de transport en commun de six grandes villes Américaines : Boston, Chicago, New York, San Francisco, Seattle et Washington.

Dans ces six villes, les usagers utilisent des cartes dans lesquelles une valeur monétaire est enregistrée, mais les technologies diffèrent d'une ville à l'autre. Par exemple, la carte à bande magnétique est employée à Boston alors qu'à Washington on utilise la carte «sans contact». À New York, la carte peut être utilisée à d'autres fins que de voyager sur le réseau de transport, ce qui implique l'adoption d'un système ouvert pouvant interagir avec de tierces parties. (Dinning et Collura, 1995)

Avant de présenter les enjeux reliés aux nouveaux systèmes de perception, les auteurs ont élaboré une liste de facteurs incitatifs à l'adoption de ceux-ci. Le tableau suivant les résume et rapporte d'autres travaux sur le sujet.

Tableau 1-3 Avantages des nouveaux systèmes de perception en transport en commun

<i>Avantage</i>	<i>Description</i>
1-Améliorer la satisfaction des passagers	<p>Plusieurs auteurs s'entendent sur le fait que la carte à puce est plus facile d'utilisation pour le paiement comparativement au paiement par espèce et par billets. (Card Technology today, 2003; Blythe, 2000). Dinning et Collura (1995) rapportent aussi l'avantage de ne pas avoir à racheter souvent de nouveaux titres de transports. De plus, Bonneau (2002) énonce le fait que les usagers de la Washington Metropolitan Area Transit Authority (WMATA) pourront utiliser un portail Internet, leur permettant de renflouer le solde de leur carte après s'être identifiés. Le tout étant relié à une carte de crédit ou au compte régulier de débit.</p> <p>Il n'y a pas que la flexibilité du paiement qui peut être un avantage pour les utilisateurs, il y a aussi la rapidité de la transaction. L'avis sur ce sujet est cependant partagé. Alors qu'on énonce un gain variable en fonction du nombre de passagers dans l'utilisation des cartes à puces en autobus (Card Technology today, 2003), une étude de Chira-Chavala et Coifman (1996) démontre que le gain de temps, bien que présent, n'est pas pour autant significatif. De plus, ces deux derniers auteurs ont comparé les temps de traitement entre une carte de type «contactless» et une «contact» card. Ils ont trouvé que les temps de traitement étaient très similaires alors que Shefler et Procaccino (2002) déclarent un gain potentiel de 20 à 30% avec la carte fonctionnant par radiofréquence. On ne connaît toutefois pas les détails de leurs tests.</p>
2-Améliorer la qualité et la sécurité	<p>Un autre point sur lequel tous les auteurs s'entendent, c'est au niveau de la fraude. Tous les auteurs mentionnés dans la rubrique 1 de ce tableau énoncent que les cartes à puces vont permettre de diminuer la fraude qui fait perdre beaucoup d'argent aux sociétés de transport qui utilisent les systèmes de perception par espèces et billets.</p> <ul style="list-style-type: none"> • Une étude intéressante sur l'appréciation des cartes à puce par les chauffeurs d'autobus, apporte un autre point sur la sécurité. En effet, Chira-Chavala et Coifman (1996) ont recueillis, auprès des

Tableau 1-4 Avantages des nouveaux systèmes de perception en transport en commun (suite)

	<p>chauffeurs les commentaires suivants relativement aux cartes à puce :</p> <ul style="list-style-type: none"> • Diminution de la charge de travail à chaque arrêt; • Perception plus simple et plus rapide; • Se sentent valorisés dans leur travail puisque les usagers aiment la technologie; <p>Moins de validation en roulant (attente d'argent de l'utilisateur), donc les chauffeurs gardent les yeux sur la route.</p>
<p>3-Créer de nouvelles sources de revenus</p>	<p>Quelques auteurs parlent du potentiel d'enregistrement de données des cartes à puces pour apprendre à connaître les habitudes de déplacement des usagers et pouvoir ainsi adopter une tarification mieux adaptée à leurs besoins en plus de pouvoir instaurer des programmes de loyauté (récompenses). On peut mentionner d'ailleurs sur ce sujet, Dinning et Collura (1995) et M. Blythe (2004).</p> <p>De plus, on commence à parler de cartes multi-fonctions (Shefler et Procaccino, 2002) et il y aura fort à parier qu'il y aura un fort appétit pour ce genre de technologie dans le futur, comme le mentionne le Card Technology today (2003) : «Le secteur où les sociétés de transport de masse vont voir un impact sera les programmes de cartes qui pourront être ouverts à de multiples participants et à de nouvelles relations commerciales créatives entre les applications en transport (émetteurs de cartes) et ceux qui offrent d'autres applications utilisant des cartes à puces (d'autres émetteurs de cartes et d'applications.».</p>
<p>4-Réduire le traitement associé aux systèmes de perception classique</p>	<p>Le traitement des billets, et de l'argent comptant n'est pas seulement un fardeau à bord des autobus. Il y a aussi un traitement à effectuer, à posteriori, de tout cet argent perçu à chaque jour par tous les autobus du réseau chez les sociétés de transport. L'utilisation de systèmes de perception automatisés, va donc aussi réduire la charge de travail à ce niveau (Dinning et Collura, 1995).</p>

Tableau 1-5 Avantages des nouveaux systèmes de perception en transport en commun (suite et fin)

5-Remplacer les vieux équipements désuets	Les équipements actuels, tirant à la fin de leur cycle de vie, peuvent aussi être un incitatif pour passer à un nouveau système de perception automatisé. C'est d'ailleurs ce qui s'est passé aux sociétés de transport de New York et Chicago (Dinning et Collura, 1995).
6-Implémenter des réglementations de tarifs intégrées	Un autre des avantages de ces systèmes de perception peut se démarquer dans les grandes villes où il peut y avoir plusieurs opérateurs de transport. En effet, grâce à cette technologie il sera possible d'intégrer des réglementations tarifaires communes afin de faciliter la coordination entre les compagnies de transport (Blythe, 2004; Bonneau, 2002). Ceci serait d'ailleurs très pratique pour le transport privé, entre autres, qui peut utiliser une tarification à la distance parcourue. Avec les cartes à puces, par exemple, un usager pourra se déplacer sur plus d'un réseau avec sa carte, alors que celle-ci tiendra compte de chacune des structures tarifaires des différents opérateurs de réseaux (Dinning et Collura, 1995).
7-Faciliter la prise de décision et la collecte de données opérationnelles.	Ici, il s'agit de l'avantage marquant pour les chercheurs et les opérateurs de réseau, soit la collecte massive de données longitudinales qui pourront être utilisées pour effectuer diverses analyses ou bien à des fins de planification. Le tout sera couvert dans la section 1.3.

Évidemment, l'adoption d'un nouveau système de perception automatisé comporte, malgré tous ces avantages, des enjeux auxquels il faut faire face. L'étude de Dinning et Collura (1995) auprès de six grandes villes Américaines en rapporte quelques uns :

1. au niveau de la planification stratégique et de la réglementation tarifaire : Pour plusieurs régions, la coordination au niveau des réglementations tarifaires semble un sujet délicat. En effet, les différents opérateurs veulent tout de même garder une certaine autonomie au niveau de leur structure de tarification. L'objectif des pourparlers se ramène donc souvent à la coordination de services intermodaux afin de maximiser la productivité. Au niveau de la planification stratégique, il

faut essayer de planifier tous les changements au niveau opérationnel, financier et informationnel que le nouveau système apportera. Par exemple, au point de vue financier, les auteurs rapportent que l'ajout des nouveaux équipements a entraîné des coûts supplémentaires substantiels à la société de transport de New York qui ont dû mettre à jour toute leur infrastructure d'alimentation électrique afin de supporter les nouveaux équipements. Finalement, encore au niveau stratégique, il faut penser à l'implantation du système au niveau opérationnel. Une implémentation graduelle par objets distincts du réseau est recommandée. Il ne faut pas oublier aussi qu'un nouveau système de perception automatisé par carte à puces possède une certaine période de rodage en terme de stabilisation des données recueillies pour des fins d'analyses (Bagchi et White, 2005);

2. au niveau de la technologie : Ici les auteurs mentionnent le choix entre les technologies «contact» et «contactless» pour les cartes à puces en plus du fait de prévoir une architecture ouverte pour ces cartes afin d'envisager l'ajout de nouvelles applications à celles-ci afin de suivre l'évolution technologique;
3. au niveau du paiement et de sa gestion : Si on envisage d'autres applications en parallèle à l'utilisation de la carte de transport et notamment au niveau d'une application monétaire (i.e. portefeuille électronique) il va falloir «ouvrir» les frontières de l'architecture financière de la société de transport à des institutions privées comme des banques qui auront certainement leurs exigences face à la gestion de l'argent. Il faudra donc s'assurer de bien discuter des arrangements à prendre entre les secteurs publics et privés;
4. au niveau de l'acceptation des usagers : L'adoption d'un nouveau système de perception aura très certainement des impacts sur les usagers et les employés de la compagnie de transport qui implante un tel système. Il faut s'attendre une résistance face au changement et donc planifier des actions en conséquence afin de faciliter la courbe d'apprentissage. De plus il faudra surtout s'assurer d'une

bonne formation aux employés en leur fournissant des procédures à suivre, notamment au niveau des cartes rejetées.

5. au niveau de la confidentialité : Évidemment il y a toute la question de la confidentialité des usagers qui utiliseront une carte à laquelle un numéro d'identification unique sera associée. Il faudra donc que les sociétés de transport clarifient leur politique de confidentialité face à l'utilisation des données recueillies par ces systèmes afin de rassurer les utilisateurs. Plus de détails concernant la confidentialité seront énoncés dans la section 1.3.3.

Malgré ces enjeux, les avantages pèsent d'avantage dans la balance pour les société de transport public et il y a présentement un engouement, si l'on peut s'exprimer ainsi, envers les systèmes de perception par carte à puces, puisqu'on recense l'implantation de pas moins de 46 projets de la sorte au sein de 24 pays différents (Card Technology today, 2003).

1.2.4 Contexte de la Société de transport de l'Outaouais

La société de transport de l'Outaouais (*STO*) est une société de transport public que l'on peut qualifier de taille moyenne opérant environ 200 autobus au service de quelques 240 000 habitants. Le réseau de la STO couvre une partie de la capitale nationale, Ottawa, à partir du sud, jusqu'au secteur de Hull plus au nord. D'ouest en est, le réseau assure la couverture des secteurs d'Aylmer jusqu'à Buckingham en passant par le secteur de Gatineau. On peut d'ailleurs voir l'étendue du réseau sur la figure suivante :

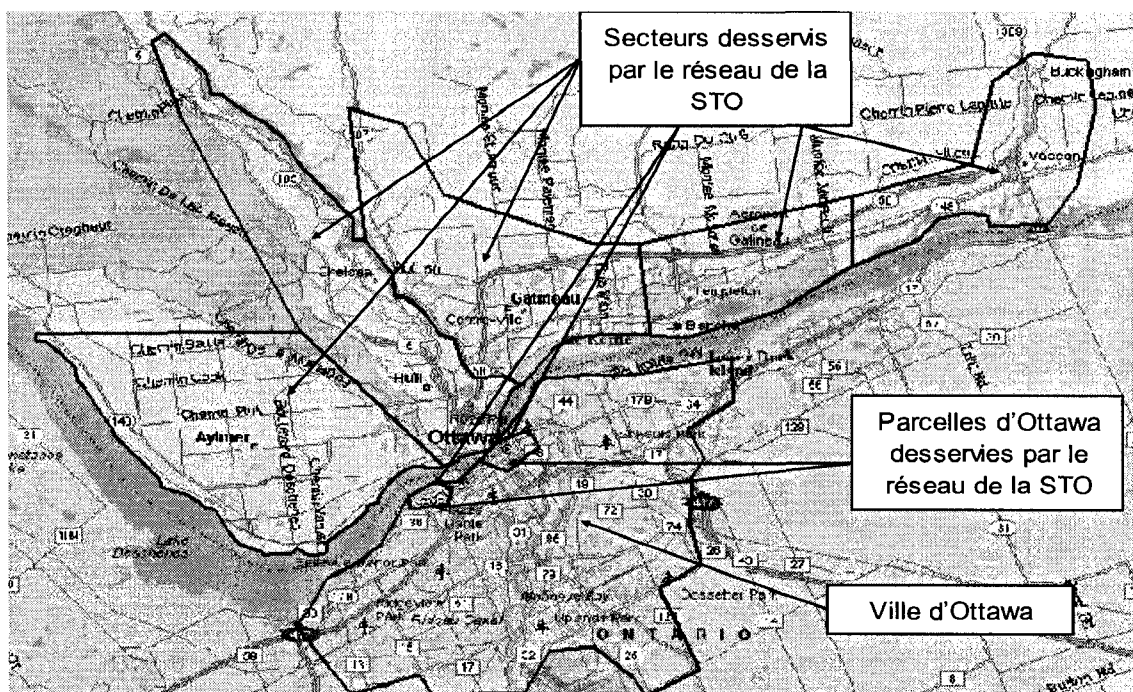


Figure 1-8 Présentation du territoire de la STO (Tranchant, 2005)

On peut poursuivre cette section en présentant un résumé de l'historique de la société de transport.² Tout d'abord, avant 1971, le transport en commun était assuré par des compagnies privées. Un an après l'émission des lettres patentes et la création de la Commission de transport de la Communauté régionale de l'Outaouais (CTCRO) en 1971, la société fit l'acquisition des parcs d'autobus des 3 compagnies privées qui assuraient le transport. De plus, ils adoptèrent le symbole et des couleurs qui caractérisent encore aujourd'hui la Société. Un an plus tard, ils signèrent avec Ottawa une entente sur le transport interprovincial assurant l'interchangeabilité des correspondances entre la STO et OC Transpo (société de transport de Ottawa). Pour subvenir à leurs besoins grandissants et à l'augmentation du personnel, la compagnie s'est établie en 1977 dans de nouveaux locaux. En 1981, la société effectua quelques

² Toutes les données historiques proviennent du site Web suivant :
http://www.sto.ca/a_propos/historique_f.html

changements par rapport à ses offres de service. En effet, La Commission confia, par contrat, à Trans-Spec Outaouais la tâche d'effectuer le transport des personnes handicapées et elle abandonna le service de transport des écoliers.

En 1983, suite à la création du service informatique, la compagnie commença à informatiser son réseau et confia un contrat de 1,5 millions de dollars à la firme Fischback and More pour l'informatisation de celui-ci. Deux ans plus tard, ils commencèrent le développement et l'implantation du système Sagepas (Système automatisé pour la gestion du parc autobus). De plus ils octroyèrent un contrat afin d'agrandir la capacité du garage administratif à 200 autobus. En 1986, ils lancèrent un système d'information aux usagers, le système Infobus, leur permettant de connaître à l'avance l'heure de passage des 2 prochains autobus.

Pour faciliter la perception dans ces autobus, la STO fit abolir, en 1987, l'utilisation de la monnaie de papier et des pièces d'un cent comme mode de paiement. Dix ans plus tard, on commença à mettre sur pied le projet de système de perception automatisé par carte à puces en choisissant un fournisseur. Durant les années antérieures, le réseau de la STO a continué à s'étendre et a reçu un prix, en juin 1998, «Réseau de transport en commun de l'année» lors du congrès annuel de l'Association Canadienne de Transport Urbain (ACTU) de Terre-Neuve. Le travail sur le projet de perception automatisé fit naître le système informatisé pour la validation des titres de transport (SIVT) qui fut officiellement lancé en 2001.

D'autres innovations ont suivi au cours des dernières années et parmi celles-ci on peut mentionner entre autres : le développement de son site Internet (2001 et 2002), l'expérimentation auprès de la clientèle d'un autobus hybride-électrique, le lancement d'un programme de prélèvement bancaire automatique pour la clientèle régulière du transport en commun (2002) et l'élaboration d'une étude de faisabilité détaillée du projet Rapibus (réseau de bus spécial reliant les différentes agglomérations). On peut donc dire que la STO est une société de transport avant-gardiste, à tout le moins au Québec, puisqu'elle est la première à emboîter le pas sur les cartes à puces.

Finalement, il faut aussi mentionner tous les projets concernant le traitement des données de cartes à puces qui ont été réalisés par les chercheurs de l'École Polytechnique en partenariat avec la STO. Ces recherches ont commencé en 2003 avec Romain Poilpré (voir la figure I-1 et la section suivante) et se poursuivent encore cette année avec le projet actuel et d'autres projets futurs, vu l'intérêt porté par les deux parties.

1.3 Le traitement des données de cartes à puces

Dans la prochaine section il sera question de plusieurs travaux de recherche qui ont impliqué l'utilisation des données de cartes à puces dans le domaine des transports en commun. Tout d'abord on couvrira le traitement des erreurs associées aux données recueillies. Ensuite on traitera de l'utilisation des données dans l'analyse des comportements des usagers ainsi que de l'utilisation en planification. Finalement, on terminera la section par les aspects de confidentialité associés aux données de cartes à puces.

1.3.1 Traitement des erreurs : imputation et enrichissement des données

Outre les avantages nommés au cours de ce chapitre, les données fournies par les systèmes de perception automatisés par carte à puces apportent aussi leurs lots de problèmes. En effet, les données recueillies par ces systèmes sont souvent entachées d'erreurs. La littérature sur le sujet des systèmes de perception automatisé a déjà recensé le phénomène (par exemple voir Utsunomiya et al., (2006) en section 1.3.2), mais peu de chercheurs ont réellement abordé le sujet en profondeur. Toutefois, les chercheurs Chapleau et Chu se sont attaqués au problème en proposant non seulement des techniques robustes d'identification et d'imputation des erreurs, mais aussi des méthodes d'enrichissement des données de cartes à puces.

Les premiers travaux d'envergure concernant la détection des erreurs recensés dans cette revue de littérature sont sûrement ceux de Trépanier et al. (2007). Ils ont dû identifier de façon claire les erreurs retrouvées dans les transactions du système SIVT de la STO afin

que celles-ci aient le moins d'impact possible sur les résultats retournés par leurs algorithmes de dérivation des destinations. Une partie de ces études peut être considérée comme étant une base des recherches de Chapleau et Chu. Toutefois, les travaux de Trépanier et al. sont décrits plus en détail dans le chapitre 3, étant donné que l'implantation de l'algorithme de dérivation des destinations a fait partie de ce projet.

Les analyses de Chapleau et Chu (2007a) consistent donc à utiliser certaines règles pour vérifier les contraintes spatio-temporelles des objets d'un réseau de transport public afin d'identifier toute donnée erronée, suspecte ou inapplicable et d'en imputer une valeur plausible. Les concepts utilisés pour corriger ces erreurs sont la régularité des opérations en transport public ainsi que la régularité de l'historique des déplacements des usagers.

Voici tout d'abord les critères sur lesquels se basent les auteurs pour identifier les transactions «fausses» :

- une transaction est jugée comme étant inapplicable si elle ne s'est pas effectuée dans le service planifié pour la journée concernée;
- une transaction est considérée erronée quand un de ces attributs est inconsistant avec les concepts du réseau de transport public;
- une transaction est jugée suspecte quand le véhicule dans lequel elle a été effectuée ne respecte pas les logiques spatio-temporelles du réseau de transport public.

Donc parmi ces trois types d'erreurs, on peut retrouver les exemples suivants recensés sur les données de cartes à puces provenant de la STO:

- montée sur une mauvaise ligne (comparé au service planifié en cours);
- heure de montée 0000;
- pas de réinitialisation de voyage;
- temps d'arrêt trop long à un arrêt;

- bris d'un équipement GPS, montées toujours au même arrêt;
- arrêt précédent répété;
- montée en bout de ligne;
- etc.

Une fois ces erreurs identifiées, on peut utiliser les 2 concepts de régularité mentionnés précédemment afin de ré – imputer les bonnes transactions. Par exemple, du côté du concept de la régularité opérationnelle, on peut repérer le bon voyage à affecter à une transaction en regardant si au moins un des voyages de la course du jour est bon. De plus, on peut rétablir le service complet en utilisant les données de plusieurs journées. Donc, une fois le service replanifié, on peut enrichir les données pour mieux réaffecter les mauvaises transactions aux bons voyages. Un des moyens d'enrichir les données est aussi de se servir des transactions erronées dont la cause est une erreur humaine, comme les montées en bout de ligne par exemple. Ceci nous permet d'obtenir de l'information sur les durées planifiées de voyage ainsi que les durées maximales.

Ensuite, à l'aide de techniques de data mining, on est capable de reproduire des transactions, comme par exemple, une montée à un arrêt et une heure précise à partir de l'historique d'une carte. Par après, si on veut ré – imputer cette transaction, on doit aussi la revalider par rapport à l'horaire.

Les résultats obtenus par l'étude de Chapleau et Chu (2007a) démontrent que sur une journée de semaine typique, le concept de régularité des opérations, à l'aide de la reconstruction d'horaires, est capable d'imputer une valeur pour 95,1% des transactions enregistrées et que le concept de régularité des déplacements des usagers, qui va chercher les attributs des déplacements en analysant les habitudes du va et viens des usagers, est capable de recréer les informations de 52,4% (succès de 76% pour le bon arrêt d'embarquement seulement) des transactions. Ces travaux permettront donc aux sociétés de transport de mieux cibler les secteurs de faiblesse (causant des erreurs) de

leur système de perception en plus de se munir d'une méthode d'imputation des erreurs qui améliorera la valeur des données de cartes à puces en planification.

D'ailleurs, une fois cette méthode de correction d'erreurs mise en place, les chercheurs ont effectués des travaux sur la reconstruction de matrices origines – destinations. Toutefois, les matrices qu'ils ont dérivées le sont par section de déplacement seulement, à cause des problèmes liés à la détermination des transferts (voir plus bas) qui fait en sorte que le modèle actuel assume que chaque transaction est associée à un déplacement et un motif, ce qui ne reflète pas nécessairement la réalité (Chapleau et Chu 2007b).

Les mêmes auteurs ont aussi réalisé une étude sur l'enrichissement des données de cartes à puces. Basé sur leurs travaux antérieurs, dont ceux sur la connaissance de la destination des transactions, les chercheurs ont utilisé des méthodes de reconstruction d'itinéraires et de reconstruction d'horaire afin d'identifier les transferts (les vraies chaînes de déplacements) selon des contraintes spatio-temporelles. En effet, les auteurs dénoncent les limites potentielles associées à l'utilisation d'une valeur fixe (exemple 2 heures entre 2 montées consécutives) dans la détermination des correspondances sur le réseau (Chapleau et Chu, 2008) : «Une valeur fixe qui ne prend pas en considération le temps passé en véhicule et la direction empruntée classifera invariablement toutes les montées comprises dans le seuil fixé comme étant des correspondances et donc faisant partie d'une chaîne de déplacements.». En effet, les déplacements avec de courtes durées d'activité (ou chaîne de déplacements) peuvent être masqués et aussi des déplacements de retour (exemple : un aller-retour au vidéo pour retourner un DVD) peuvent être comptés comme étant des transferts. Donc, afin de mieux identifier les bons transferts, Chapleau et Chu (2008) proposent d'utiliser les 2 méthodes suivantes :

- la reconstruction d'itinéraires : elle permet d'enregistrer l'origine et la destination des déplacements en plus du temps de départ, du détail des lignes choisies et des activités de transfert tout en conservant les attributs du détenteur de la carte de façon désagrégée. Ceci est supérieur aux matrices O – D selon eux;

- la reconstruction d'horaire : en se servant de la connaissance des temps planifiés de départ et d'arrivée de voyage, de la séquence des arrêts et de la distance qui les sépare, on peut estimer l'heure de descente de l'utilisateur lorsqu'on connaît sa destination.

Ces deux méthodes permettront donc d'empêcher la surestimation des correspondances par les systèmes de perception automatisée et d'identifier ainsi les bonnes chaînes (vraies) de déplacements. Les résultats de leur algorithme ont d'ailleurs démontré que le système surestimait les correspondances par une marge d'environ 40%. Ces travaux sont donc prometteurs dans l'optique de rendre les données de cartes à puces plus fiables pour leur utilisation en planification à plusieurs niveaux. Par contre, concernant cette étude en particulier, il serait intéressant, vu que cette analyse se sert des destinations dérivées, de voir si un changement de statut d'une transaction (de correspondance à non correspondance) aura un impact sur les futures destinations à dériver, car une bonne partie repose sur ce critère ainsi que sur l'historique général des déplacements des usagers.

Il serait intéressant aussi, concernant tous ces travaux, d'obtenir le temps de traitement requis pour imputer et enrichir les données d'une journée de transaction afin de déterminer s'il est possible d'intégrer ces traitements à ceux déjà en place pour ce projet (voir chapitre 3 et 4). Ceux-ci requièrent déjà un temps non négligeable de traitement, il se peut donc que la combinaison de ces traitements de données, bien que bénéfiques, ne soit pas encore envisageable pour une utilisation au jour le jour.

1.3.2 Analyses des comportements des usagers et utilisation en planification

L'aspect longitudinal des données fournies par les systèmes de cartes à puces a permis l'émergence de nouveaux types de recherches dans le domaine du transport en commun. Une fois ces systèmes mis en place depuis un certain temps il est en effet possible d'analyser le comportement des usagers d'un réseau à travers le temps. On verra dans cette section que ces analyses peuvent apporter de l'information utile en planification à différents niveaux.

Bagchi et White (2004) se sont d'abord intéressés au rôle que devaient prendre les données collectées par les cartes à puces. Ils ont reconnu le caractère longitudinal de ces données, qui, une fois reliées à un individu ou à une carte et ses déplacements, permettent de reconstruire les itinéraires et ainsi d'analyser les comportements, tâche difficile à effectuer à l'aide des sources de données habituelles. D'ailleurs, ceux-ci se sont questionnés à savoir si cette nouvelle source de données pourrait éventuellement remplacer les méthodes d'acquisition de données actuelles, soit les enquêtes origine - destinations et les sondages auprès de la clientèle dans les autobus. Ils ont conclu sur ce sujet que les données de cartes à puces seraient un excellent complément aux autres sources d'acquisition de données en plus de réduire la fréquence d'acquisition de celles-ci. Elles ne pourraient toutefois pas les remplacer pour les raisons suivantes :

- impossibilité d'enregistrer la destination des usagers;
- aucune information disponible sur les générateurs des déplacements de ces usagers;
- impossibilité d'enregistrer la satisfaction des clients.

Les auteurs se sont d'ailleurs servis de la combinaison des données de cartes à puces et de données de comptes à bord afin de calculer les remboursements auxquels les sociétés de transport ont droit pour le transport des aînés et des étudiants. Toutefois, ceux-ci n'émettent pas de préférence quant à quelle source de données devrait être utilisée à la base des calculs versus celle qui servirait de vérification ou de calibration des résultats.

Finalement ces deux chercheurs anglais ont réalisé une étude de terrain afin de déterminer le pourcentage de correspondance par groupe d'utilisateur sur le réseau de «First in Bradbord» à West Yorkshire en Angleterre. Le manque d'informations au niveau des destinations et la fenêtre de temps imposée comme paramètre définissant une correspondance (30 minutes dans leur cas) sont des facteurs qui peuvent mettre la validité de leurs résultats en doute.

La dérivation des transferts à partir de données provenant de systèmes de perception automatisée a aussi inspiré les travaux de Hofmann et O'Mahony (2005). Ceux-ci ont utilisé les données provenant des cartes magnétiques. Afin de dériver les transferts, les auteurs ont utilisé un algorithme itératif pour migrer la classification des déplacements simples et de ceux considérés comme transfert dans une autre base de données. Le critère d'identification des transferts est le recensement de deux montées à l'intérieur d'une période de temps de 90 minutes. Les résultats obtenus comptabilisent 1,6 millions de transfert sur 8 millions de déplacements observés sur une période de 4 mois. De plus les chercheurs ont identifié le centre-ville comme étant la zone la plus riche en transfert parmi le secteur d'utilisation. Nous savons maintenant (section 1.3.1) que l'absence de données spatiales précises (arrêt d'embarquement et direction) et la connaissance de la destination (connaissance du générateur de déplacement) des usagers peuvent remettre en doute la validité des résultats obtenus par ces auteurs. Toutefois, Hofmann et O'Mahony (2005) n'énoncent pas ces limites dans leur travail.

Dans un autre article de Bagchi et White (2005), les auteurs réitèrent les avantages et les inconvénients à tirer des données de cartes à puces et poursuivent leurs travaux sur l'aspect de la connaissance des comportements des usagers les plus fréquents du réseau de transport étudié. Cette fois-ci ils ont calculé le taux de roulement des usagers du transport en commun sur une période qui selon eux est propice à l'observation de changements de comportements, soit sur une année. Les compagnies de transport pourront se servir de ces résultats afin de minimiser les pertes dues au taux de roulement et d'en augmenter les gains. Les auteurs émettent toutefois des réserves quant à la validité des données recueillies (sur les taux de roulement en particulier) par un système fraîchement installé. Ils proposent donc un temps de rodage de deux ans suite à l'implantation avant de se lancer dans ce type d'analyse en plus de toujours valider les résultats à l'aide des sources de données traditionnelles.

D'autres études ont été réalisées du côté de la Chicago Transit Authority (CTA) concernant le potentiel d'utilisation des données de cartes à puces en planification du

transport (Utsunomiya et al., 2006). Les auteurs énoncent que les sociétés de transport évaluent leur service en se basant sur des sources de données coûteuses (enquêtes OD, comptes à bord, etc.) et peu fiables. Ils soutiennent donc que les données fournies par les systèmes de perception par cartes à puces constituent une énorme source d'information (mentionnent le caractère longitudinal des données) de laquelle on peut dériver plusieurs aspects utiles aux planificateurs en transport. Ayant accès aux adresses de facturation des usagers de la CTA, les auteurs ont réalisé une étude sur la distance de marche séparant cette adresse à l'arrêt utilisé en début de journée (première montée après 3 heures AM). Ils ont trouvé que les distances de marche étaient plus élevées pour se rendre au service de train (0,2 à 0,3 mile) que pour accéder à l'autobus (0 à 0,1 mile). Selon eux, ces résultats sont justifiés pour les deux raisons suivantes : premièrement, le réseau de train de Chicago représente seulement 12% de la grandeur de celui des autobus et deuxièmement, ils avancent que les gens sont plus susceptibles de marcher sur une grande distance pour se rendre à un mode de transport plus rapide et plus fiable. En plus des travaux sur les distances d'accès, les chercheurs ont analysé la fréquence et la consistance d'utilisation du réseau ainsi que la détermination des correspondances (2 heures d'intervalle dans leurs cas) au sein du réseau. Finalement les auteurs ont énoncé les limites de l'utilisation de telles données, mentionnant entre autres que (Utsunomiya et al., 2006) :

- les destinations des usagers sont inconnues;
- les données contiennent des erreurs : des transactions manquantes et des mauvais numéros de lignes ou de voyage;
- le fait de ne pas utiliser les technologies AVL (*Advanced Vehicle Location*) en conjonction avec les données opérationnelles du système de perception automatisé n'aide justement pas à prévenir les erreurs.

Selon eux, il y a donc beaucoup de travail à effectuer dans le domaine de l'identification et l'imputation des erreurs et qu'il faudra d'ailleurs faire preuve de précaution dans l'utilisation de ces données menant à des décisions en planifications.

Les chercheurs Morency, Trépanier et Agard (2006) ont eux aussi effectués des travaux sur l'analyse du comportement des usagers à partir de données de cartes à puces. Basé sur un jeu de données de la STO s'échelonnant sur dix mois ils ont analysé la variabilité spatio-temporelle de la mobilité quotidienne des usagers. Pour ce faire, ils ont utilisé des techniques de data - mining (création d'agrégats à l'aide de la technique des k-moyennes et hiérarchie ascendante) leur permettant de caractériser les groupes d'usagers et de séparer l'horaire des journées en période d'utilisations les plus probables, soit la pointe AM, le jour, la pointe PM et le soir. L'objectif de cette méthode est de révéler des groupes naturels d'usagers – semaines. Les principaux résultats de leurs travaux sont les suivants :

- analyse de l'utilisation quotidienne du réseau par les différents groupes d'usagers : Tout d'abord il y a l'identification d'un groupe travail qui effectue ces déplacements en majorité dans la pointe AM et la pointe PM avec seulement 5% de déplacement dans la journée. Ensuite, il y a le groupe d'activité de jour avec des déplacements en pointe AM et de jour en majorité. Finalement le 40% du reste des usagers qui ont des déplacements qui ne sont pas associés à un modèle en particulier ou bien qui n'ont pas utilisé le réseau suffisamment durant la période observée.
- classification des titres de transport aux différents groupes d'usagers : On apprend que 60% des titres adultes appartiennent au groupe de travail et que l'on identifie la majorité (mais moins forte) des étudiants au groupe de jour, alors que les aînés appartiennent au groupe dont le patron d'utilisation du réseau n'est pas bien identifié.
- Appartenance aux groupes à travers le temps : les auteurs ont trouvé que l'appartenance aux différents groupes est restée stable à travers le temps, mais qu'il peut y avoir des variations, comme par exemple, pour la semaine de relâche.

Outre ces analyses, les auteurs ont étudié le taux d'utilisation du réseau de transport en commun ainsi que le taux d'acquisition de nouveaux arrêts par les usagers.

Tout comme Bachi et White (2004), les chercheurs reconnaissent les perspectives d'analyse offertes par les cartes à puces, tout en élargissant l'horizon de celles-ci : «elles permettent aussi d'envisager le croisement de différentes variables contextuelles (météo, événement, congés scolaires) afin de comprendre leurs effets sur la modification ponctuelle des comportements des usagers. Ceci permettrait aussi l'analyse de la saisonnalité des comportements versus l'étude d'un jour moyen en enquête origine – destination.» (Morency et al., 2006). De plus, ils en viennent à la même conclusion concernant l'utilité des autres sources de données : «Bien évidemment, elles (les données de cartes à puces) ne peuvent se substituer à ces enquêtes régionales puisqu'elles ne ciblent que les usagers d'un réseau particulier.» (Morency et al., 2006). Ces travaux apportent donc de nouveaux types d'informations que les analystes de réseaux de transport en commun auront avantage à intégrer à leur planification.

L'analyse de la variabilité de la demande de transport est un autre sujet qui a été abordé par les auteurs précédents. En effet, la demande de transport varie avec le temps, les saisons, les heures et la météo. Il est dans l'intérêt des sociétés de transport d'avoir une bonne vue d'ensemble de la variation de la demande de transport afin de réduire les coûts et d'ajuster conséquemment le service. Les chercheurs ont donc, encore une fois à partir des données de cartes à puces provenant de la STO, étudié les comportements suivants (Morency et al., 2007a):

- une étude sur les arrêts les plus utilisés par types de titres;
- une étude sur le taux d'acquisition de nouvel arrêt par semaine (pour une période de 41 semaines consécutives);
- une étude de la fréquence d'utilisation de ces arrêts et de la répartition des heures de montées utilisées.

Leurs études ont donc démontré que les données de cartes à puces ont tout le potentiel nécessaire à l'élaboration du profil continu de l'utilisation du réseau de transport public par différents types de cartes. Finalement les auteurs nous rappellent que ces indices de régularité observés ici, ont été dérivés seulement à partir des données brutes de montées. Il s'agit donc d'un bon potentiel d'analyse utilisable en planification et qui ne nécessite pas d'énormes calculs comparativement à l'utilisation de statistiques qui requièrent les informations sur les descentes des usagers.

Dans le cadre de la 11^{ème} conférence mondiale sur la recherche en transport (WCTR, *World Conference on Transportation Research*), ces mêmes auteurs ont présenté un papier sur la modélisation des indicateurs de performance du côté de l'offre de transport et du service aux passagers. Cette fois-ci, le calcul de ces statistiques ont nécessité l'utilisation de l'algorithme de dérivation des destinations créé par Trépanier et al. (2007). Voici une liste des indicateurs de performance qui ont été calculés par les chercheurs, toujours à partir du jeu de données provenant du système de perception de la STO. (Morency et al., 2007b) :

- côté de l'offre : comparaison du service planifié, soit, les véhicules – kilomètres et les véhicules – heures, avec le service qui a réellement été offert aux utilisateurs pour une journée spécifique;
- côté de l'utilisation du service : calcul des passagers – kilomètres et des passagers – heures ainsi que de la charge à bord.

Ces calculs permettent de voir comment ces deux côtés du réseau de transport évoluent à travers le temps afin d'évaluer la performance de celui-ci. Entre autres, les auteurs ont vérifié l'adhérence à l'horaire planifié des autobus. Ils ont trouvé, pour l'observation d'une ligne spécifique durant le mois de février 2005, une adhérence parfaite à l'horaire dans 17,5% des observations. Le reste des répartitions se présentant ainsi : 18,9% des observations avec une avance sur l'horaire et 63,6% des observations avec un retard. Finalement, les auteurs réitèrent les précautions à l'égard de l'utilisation des données de cartes à puces (Morency et al., 2007b) :

- il faut que le système d'information qui supporte les données de transactions soit suffisamment performant comme cette approche requiert beaucoup de traitement de données;
- il faut que les données analysées soient pré-validées, car on sait que les données provenant de cartes à puces ne sont pas sans erreurs;
- il faut aussi se rappeler que ce ne sont pas tous les usagers du réseau qui possèdent la carte à puce, et qu'il faudra donc valider et balancer les résultats obtenus à l'aide de comptes à bord.

L'étude présentée dans ce dernier article se rapproche beaucoup du travail du projet actuel en terme de modélisation des statistiques opérationnelles à partir de données de cartes à puces. Toutefois, il manque à leur travail, l'aspect de «démocratisation» des données permettant une utilisation quotidienne des statistiques par les planificateurs.

1.3.3 Diffusion des statistiques à l'interne pour la planification

La disponibilité et la distribution des données de transport ont toujours été un problème pour les différents acteurs du milieu, soit les opérateurs, les planificateurs, les managers ainsi que le public en général. Chapleau a démontré en 2002 que, dans le cas des grandes enquêtes de ménages, il est possible d'offrir au public de l'information sur la mobilité en général. Des applications interactives basées sur des feuilles de calcul Excel sont utilisées à Montréal afin de diffuser des statistiques agrégées sur la mobilité urbaine (Chapleau et al., 1997). Dans un article de Tang et Waters (2005), l'on discute de la participation du public dans la planification du transport. Ils montrent les avantages d'utiliser des techniques de diffusions «légères» comme les sites Web, le eXtended Markup Language (XML) et le Scalable Vector Graphic (SVG) afin de communiquer les données de transport au public. Ils proposent ainsi une façon de s'attaquer aux problèmes de fiabilité des données en permettant la validation par le public.

Pour ce qui est de la diffusion de données opérationnelles à l'interne, il n'y a pas beaucoup d'articles sur le sujet, mais Menzies (2004) propose quelque chose dans ce

domaine. À la société de transport de Winnipeg, on utilise un site intranet afin de diffuser à l'interne les données collectées par les systèmes de comptage à bord des autobus. Le traitement des données est effectué durant la nuit. L'accès à ces données fraîches tous les jours permet aux opérateurs de 1) trouver les équipements défectueux parmi les véhicules 2) identifier les lignes d'autobus congestionnées et 3) procurer un feedback et des suggestions immédiates aux planificateurs de service. Cette idée est bien sûr à la base de ce projet.

1.3.4 Avantages et inconvénients, vie privée

Comme on l'a vu tout au cours de ce chapitre, les systèmes de perception par carte à puces offrent bien des avantages aux sociétés de transport, tant au point de vue financier, qu'à celui de la planification. Toutefois, les questions de confidentialité associées à ces systèmes pourraient compromettre la vitesse de leur implémentation. La Commission nationale de l'informatique et des libertés (2003) a démontré qu'il était possible de reconstituer les déplacements privés d'une personne en particulier à l'aide des données de ces systèmes de perception automatisés. Évidemment cette reconstitution sera plus ou moins difficile à effectuer en fonction du niveau de confidentialité associé au système de perception utilisé. Clarke a identifié la ville de Melbourne comme étant l'endroit où les déplacements individuels anonymes n'étaient plus possibles, car la municipalité avait totalement éliminé les systèmes de perception par argent ou billets, ne laissant la place qu'aux cartes identifiées. Toutefois, ce dernier stipule que les données saisies par les cartes à puces en transport ne sont guère différentes des données que l'on peut recueillir à l'aide des cartes de crédit et les bases de données de la police (Clarke, 2001).

Les chercheurs Attoh-Okine et Shen (1995) rapportent un autre avantage inhérent aux données de perception enregistrées par carte à puces. Ils énoncent que les systèmes de paiement par carte, et les potentielles applications personnalisées de celles-ci, attirent inévitablement les gens vers les transports collectifs. Du même coup ceci aurait pour effet de fournir aux sociétés de transport toutes les informations démographiques nécessaires afin de répondre aux demandes de rapports du gouvernement et de cibler

certaines groupes de gens pour des fins de marketing au niveau tarifaire (Attoh-Okine & Shen, 1995). Il s'agit donc d'un autre exemple d'utilisation de données privées, même si, dans ce cas, on n'a pas nécessairement à relier les données à des utilisateurs en particulier.

Outre l'utilisation des données, il y a bien sûr la carte elle-même qui représente un problème puisqu'elle peut contenir plusieurs renseignements personnels, surtout dans le cas d'une carte multifonctionnelle. Tous les auteurs mentionnés dans cette section s'entendent d'ailleurs sur ce point. Clark (2001) soutient qu'il n'y a pas que la perte matérielle (ou vol) qui constitue une menace. Il évoque, entre autres, quelques failles dans l'utilisation des cartes de type «contactless» qui pourraient permettre une retransmission du contenu de celles-ci via les radiofréquences. De leur côté, Attoh-Okine et Shen (1995) mentionnent que la sécurité, dans les applications de cartes à puces, est atteinte durant les étapes de design initial chez les manufacturiers de puces. La carte serait donc ouverte à l'intrusion tout au long de son cycle d'application, soit de sa production jusqu'à chacune de ces transactions.

Bolchini et Schreiber (2002) se sont penchés sur ce genre de problème en revoyant l'architecture globale entourant les cartes à puces et non pas seulement les aspects techniques matériels. Ceux-ci imputent une très grande responsabilité aux designers de bases de données qui devront effectuer leur modélisation en tenant compte du mode de gestion des données sur une carte à puce ainsi que de leurs limites. Ceci implique les considérations suivantes :

- identifier les différentes entités à être enregistrées et définir leurs cardinalités et leurs relations;
- pour chaque entité, décider si on alloue un espace «On Card» (sur la carte) ou bien si on le confie à un autre système d'information;
- pour chacun des «On Card» on décide des types d'accès, d'encryptions et de règles de confidentialité.

Concernant les types d'accès, les auteurs ont d'ailleurs défini trois types d'intervenants par rapport à la carte :

1. le propriétaire de données «Data Owner» qui a l'accès complet (en lecture et en écriture) à toute l'information de celle-ci. Il s'agit dans la majorité des cas de l'institution émettrice de la carte;
2. l'invité «Guest» qui est soit une institution ou un usager qui accède aux données, mais en lecture seulement et sur les secteurs de la carte auquel il a accès;
3. le propriétaire de la carte «Card Owner» qui possède la carte, mais pas toutes les données des institutions. Il joue donc le rôle d'un invité.

Bolchini et Schreiber (2002) ont donc démontré qu'une bonne structure des entités enregistrées et une bonne définition des intervenants dans le système couplé à une décentralisation des enregistrements permet d'être plus efficace en matière de confidentialité.

CHAPITRE 2: REVUE TECHNIQUE DES TECHNOLOGIES

Ce chapitre présente les différentes technologies qui sont utilisées dans la réalisation de l'outil de visualisation des données du réseau de la STO. Il est d'abord question du métalangage de présentation XML et de ses dérivés. Le SVG fait aussi l'objet d'une attention particulière puisqu'il est au cœur de ce projet. Après l'exposition des technologies utilisées pour extraire les données de la base de la STO en format XML, le chapitre termine par une explication des langages utilisés pour l'extraction de données XML et la génération des pages Web dynamiques.

2.1 Le XML

Le contenu de cette section permet de présenter et de valider le choix du XML comme technologie de modélisation de données dans le but de diffusion sur le Web. Sa simplicité, sa flexibilité et sa compatibilité avec plusieurs autres technologies informatiques en font un outil incontournable dans la réalisation de ce projet.

L'Extensible Markup Language est un langage balisé, simple et flexible, dérivé du SGML (Simple Generalized Markup Language ISO 8879) (Bray et al., 2006). Le XML est un langage générique puisque nous pouvons en dériver d'autres langages spécifiques comme le XSL (*Extensible StyleSheet Language*) ou bien le SVG (*Scalable Vector Graphics*). Ajoutons les points suivants :

- le XML est un langage balisé (utilisation des chevrons < >) tout comme l'est le HTML (*Hypertext Markup Language*);
- le XML a été créé pour décrire les données;
- les balises XML ne sont pas prédéfinies, on doit les créer nous-mêmes et ainsi créer notre propre langage XML;
- le XML utilise le *Documents Type Definition* (DTD) ou un schéma XML pour définir et décrire les données qui y sont contenues;

- le XML a été recommandé par le consortium du W3C (*World Wide Web Consortium*).

Il peut y avoir ambiguïté sur ce qui distingue le XML du HTML. Le tableau suivant énonce les principales différences entre les deux technologies.

Tableau 2-1: Différences entre le XML et le HTML (inspiré de Morrison, 2002; Bos, B, 1999)

	<i>XML</i>	<i>HTML</i>
Langage balisé	Oui	Oui
Langage descriptif (de contenu)	Oui	Non
Langage de style	Non	Oui
Possède des schémas ou des DTD	Oui	Non
Possède une structure rigide	Oui	Non
Possède des balises prédéterminées	Non	Oui
But	Transporter l'information et la décrire. L'XML ne fait rien en soi. Les balises sont utilisées pour décrire les données entre celles-ci.	Un page HTML classique possède des balises de styles pour mettre en forme les données qui y sont contenues dans le but de les afficher dans un navigateur.
Complémentarité et flexibilité	Le XML est un langage multi - plateformes, sans restrictions logicielles ou "hardware" utilisé pour transmettre de l'information. Le XML a besoin d'autres outils complémentaires pour l'affichage stylisé.	Le HTML peut avoir des dépendances de plateformes ou de logiciels dépendamment des versions. Dans le futur on voit le HTML comme étant complémentaire au XML pour permettre l'affichage stylisé de ces données.

Pour terminer sur la présentation magistrale du XML, la section suivante fait état de son historique, qui remonte au SGML comme mentionné précédemment.

2.1.1 Historique

Pour retracer l'historique du XML, il faut remonter de quelques décennies dans le passé... C'est dans les années soixante, avec l'avènement de l'informatisation des différents domaines de l'édition sous formes de systèmes de «traitement de texte» qu'est né le SGML (*Standard Generalized Markup Language*). Soutenu par de grandes firmes

telles que IBM, ce langage à balises descriptives avait pour but d'être l'échine des systèmes destinés à faire du traitement de texte, de la mise en page de texte, de la recherche d'information et de l'échange de documents (Képéklian, 2000).

Un peu plus tard, dans les années 1990, est apparu le HTML, lui aussi dérivé du SGML. Ce langage recommandé par le W3C, a été conçu pour la publication de l'information sur Internet et ses avatars. Il constitue une armature rigide (mais moins stricte que le XML) et structurée servant à décrire et présenter différents contenus dans un navigateur Web. Durant toute cette décennie le HTML a évolué jusqu'à la version 4.0, acquérant en chemin des améliorations telles les hyperliens, les graphiques, les tableaux, les applets (petites applications Web interactives ou non) JAVA et les procédures événementielles.

Suite à cela, puisque les pages HTML étaient très statiques, les programmeurs ont fait naître le DHTML (Dynamic HTML) afin de pallier ce problème. Concrètement cette technologie permet au contenu de la page Web de se modifier suite aux actions de l'utilisateur ou de tout autre événement. Le DHTML utilise donc les propriétés du langage par objet, soit les différents objets (titres, cadres, images, boutons, etc.) possédant des propriétés (couleurs, police de caractère, contenu, etc.) et des procédures, dans ce cas-ci, événementielles (clique, « mouseover », etc.).

Par contre, toutes ces technologies n'ont par contre pas rendu les documents plus « intelligents » ou servi à séparer les données de la mise en forme comme on peut le faire avec le XML. C'est ainsi que ce dernier est né, en 1998, afin de donner un sens à l'information et de pouvoir la décrire, la transformer et la rendre extensible, le tout, en étant dissocié des caractéristiques du contenant.

Pour la suite des événements, on peut considérer que le XML sera là pour rester. En effet, son apparition est venue mettre un terme à l'évolution du HTML, puisque le W3C a décidé que la version 4.0 de celui-ci serait la dernière version qui serait approuvée. Le HTML sera plutôt renforcé en adoptant les principes de base du XML devenant ainsi un descendant de celui-ci, soit le XHTML. Possédant ainsi les règles appartenant à la

syntaxe propre du XML, le XHTML est, dès lors, un langage strict qui ne tolère aucune erreur de syntaxe ni de structure. Ce changement a permis au XHTML d'être visualisable par tous les outils XML tout en restant compatible avec les navigateurs actuels en plus de produire des documents dont on peut définir la structure et le contenu.

Un diagramme temporel de l'évolution des langages mentionnés dans cette section est disponible en Annexe A.1.

2.1.2 Exemple d'utilisation

L'exemple qui est utilisé dans cette section s'inscrit dans le cadre comparatif du HTML et du XML. En plus de bien démontrer la différence entre les 2 langages, il permet de mettre en évidence la lacune du HTML au niveau de l'extraction des données significatives pouvant être utilisées par d'autres logiciels ou outils informatiques au-delà du simple navigateur destiné à un usager humain.

L'exemple suivant est tout d'abord une portion de code HTML tirée d'une page Internet provenant d'une boutique de livres en ligne. Il s'agit donc du code qui informe le navigateur Web comment représenter le contenu de la page du panier d'achat d'un client potentiel :³

```
<td bgcolor="#FFFFFF" width = "51%">
<a href=" ../81332713233407">
<em>Debt of Honor</em></a>
<br>
Tom Clancy;
Paperback</b>
<font size=2 face="Verdana, Helvetica, Courier" color=#000000>
<NOBR> Price: <font color=#990>$6.99</font></b></NOBR><br>

</td>
<td bgcolor="#FFFFFF" width = "51%">
<a href=" ../81332713233407">
<em>The Hunt for Red October</em></a>
<br>
```

³ Ces exemples sont une adaptation et une traduction tiré du livre suivant : *Java for COBOL programmers, Second Edition* by John C. Byrne and Jim Cross, copyright 2002, Charles River Media, Inc. All rights reserved

```
Tom Clancy;
Hardcover</b>
<font size=2 face="Verdana, Helvetica, Courier" color=#000000>
<NOBR> Price: <font color=#990>$18.99</font></b></NOBR><br>
```

Comme on peut le voir, ce code est difficilement lisible à l'oeil nu, mais il est destiné au navigateur qui le décodera facilement. Si toutefois l'on voulait extraire de l'information de ce document HTML pour l'utiliser dans un autre logiciel, on s'attaquerait alors à une tâche plus ardue. Il faudrait construire un genre d'algorithme particulier à ce document permettant à force d'essais et erreurs, par exemple, d'extraire une information comme le prix du livre :

- Rechercher la chaîne de caractères <NOBR> Price :
- Sauter par-dessus la balise de déclaration de style ();
- Enregistrer les caractères situés après le \$ et jusqu'à la prochaine balise de font;
- Convertir les caractères en chiffres.

On ne pourrait toutefois pas garantir que cet algorithme fonctionnerait à tout coup et spécialement si des changements étaient apportés au site Web. Encore plus, il y aurait fort à parier que rien ne fonctionnerait sur un document HTML d'un autre revendeur de livre.

Le XML, par contre, sera toujours prêt à être lu et analysé par une autre application, car il est beaucoup plus prévisible et descriptif. Voici ce à quoi pourrait ressembler le code XML de cette commande :

```
<Commande NoCommande="81332713233407">
<LigneCommande>
<Titre>Debt of Honor</Titre>
<Auteur>Tom Clancy</Auteur>
<TypeCouverture>Paperback</TypeCouverture>
<Prix>$6.99</Prix>
</LigneCommande>
<LigneCommande>
<Titre>The Hunt for Red October</Titre>
<Auteur>Tom Clancy</Auteur>
<TypeCouverture>Hardcover</TypeCouverture>
<Prix>$18.99</Prix>
</LigneCommande></Commande>
```

On remarque immédiatement que cette syntaxe est beaucoup plus simple et surtout compréhensible par un usager à même le code du document. En effet, il est possible de « lire » un document XML. Ceci est principalement dû au fait que les éléments qui sont entre les balises < > ont été définis par la personne qui a créé le document. Par exemple, l'élément <Titre> marque le début de l'élément en tant que tel et </Titre> en marque la fin. Entre les deux se situe le contenu de cet élément soit « Debt of Honor » ou bien « The Hunt for Red October » et ainsi de suite autant de fois que l'on rencontrerait l'élément <Titre>.

On remarque aussi que le XML supporte la hiérarchisation des éléments, soit le fait de pouvoir contenir des éléments au sein des autres comme c'est le cas pour <LigneCommande> qui contient les éléments Titre, Auteur, TypeCouverture et Prix. On peut, si on le désire, faire autant de niveaux d'hiérarchisation qu'on le souhaite. Par exemple, on pourrait ajouter un élément <Auteurs> qui encapsulerait <Auteur> pour permettre l'ajout de livres qui possèdent plus d'un auteur. On obtiendrait ceci :

```
<LigneCommande>
<Titre>Debt of Honor</Titre>
<Auteurs>
<Auteur>Tom Clancy</Auteur>
<Auteur>John Smith</Auteur>
</Auteurs>
<TypeCouverture>Paperback</TypeCouverture>
<Prix>$6.99</Prix>
</LigneCommande>
```

Une autre particularité concerne l'élément <Commande NoCommande = "81332713233407">.

Il possède ce qu'on appelle un attribut, dans ce cas-ci, « NoCommande » qui vient ajouter de l'information supplémentaire sur l'élément. Ainsi, il serait possible de cibler une commande en particulier grâce à l'attribut numéro de commande.

Un élément peut posséder plus d'un attribut afin de le décrire plus en détail. Parfois, il est même plus avantageux, pour être plus précis, de n'utiliser que des attributs pour décrire un élément plutôt que d'avoir du texte d'inscrit entre ses balises respectives (<element> </element>). Ceci permet, entre autres, de réduire le nombre d'éléments et d'atteindre plus rapidement et plus efficacement l'information désirée. En effet, il existe

des fonctions en XSL (voir section 2.3) pour atteindre directement la valeur d'un attribut ou bien pour effectuer une opération spéciale en fonction de ce dernier.

Finalement, il faut aussi mentionner qu'un document XML doit, en théorie, commencer par une déclaration d'entête. Cette dernière contiendra la version du XML utilisée ainsi que le langage d'encodage utilisé pour les caractères. En voici un exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Une déclaration d'en-tête peut également contenir les chemins de références vers des fichiers de types DTD, XSD, XSL, ou domaine de nom. Nous verrons ces types de documents dans les prochaines sections, à commencer par les DTD « Document Type Definition ». Nous explorerons également les schémas XSD qui servent à décrire les documents XML pour ainsi valider leur contenu. Pour terminer cette section, on peut mentionner deux éléments relatifs au XML qui sont présentés en annexe : premièrement on retrouve en annexe A.2 un exemple de visualisation d'un document XML dans un navigateur Web et en annexe A.3 on retrouve quelques informations sur les particularités techniques du XML de base.

2.2 Les DTD et les schémas XSD

Comme on l'a vu dans les sections précédentes, le XML est déjà un langage balisé très rigide qui rend ainsi les systèmes et les applications qui l'utilisent plus stables et plus robustes. Toutefois, l'homme étant ce qu'il est, il sera toujours susceptible de faire des erreurs dans l'accomplissement de ses tâches. Pour contrer ces erreurs possibles dans un document XML, ce dernier s'est doté d'un outil appelé schéma qui permet de détecter les erreurs potentielles de structure et de forme (Evjen et al., 2007, pp. 135-210).

Il existe deux sortes de schémas pour définir les données d'un document XML. Le premier est le DTD pour « Document Type Definition » et le second se nomme XML Schemas (XSD). Tous deux sont créés par l'auteur du document XML afin de modéliser, si l'on veut, ce qui peut y être entré comme contenu, comme, par exemple : les valeurs possibles d'un attribut, les éléments qui doivent être nuls ou ceux qui sont

requis, le nombre d'occurrences maximales ou minimales d'un élément, les types de variables possibles pour un attribut, etc. Nous verrons, dans cette section, les deux types de schémas ainsi que leur force et leurs faiblesses.

Il est à noter toutefois, avant de commencer, que, bien que ces documents constituent une bonne pratique pour créer des documents XML sans erreurs, ils ne sont toutefois pas obligatoires à leur déploiement au sein de toute solution informatique. Par contre, dans un cadre de grande envergure, ces schémas peuvent s'avérer très importants puisqu'ils permettent de qualifier un document XML comme étant valide, si celui-ci est conforme à son schéma. Cela apporte une sorte de valeur ajoutée au document. Les différents utilisateurs d'une application peuvent donc se protéger, d'une certaine façon, de l'émetteur d'un document XML vers leur système s'il se conforme aux schémas mis en place et vice-versa. Voici, en premier lieu, le survol de l'utilisation des DTD et de leurs créations.

2.2.1 Les DTD

L'utilisation et la création des DTD (*Document Type Definition*) remontent à l'ancêtre du XML, soit le SGML. Il ne s'agit donc pas d'un nouveau-né et c'est ce qui en fait un léger défaut puisque le DTD possède son propre langage, si l'on peut s'exprimer ainsi. Avant de rentrer dans les détails du code des DTD, voyons comment les documents XML sont reliés à leurs schémas DTD. C'est à l'aide du « document type declaration » qui doit être placé dans l'en-tête du document XML (soit avant une balise d'élément), comme mentionné dans la section précédente. En voici un exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE BIBLIO SYSTEM "BIBLIO.dtd" >
```

Ici, le terme BIBLIO représente l'élément racine que l'on retrouverait dans le document XML, et BIBLIO.dtd, suivant le mot SYSTEM, indique la référence au fichier contenant le code du schéma DTD dans le même dossier que BIBLIO.xml. Voici maintenant quelques généralités sur les attributs et les éléments qui établissent la structure logique

des documents XML et qui sont la raison d'être principale des DTD. Ceci permettra de mieux comprendre un exemple complet de DTD.

2.2.1.1 Les attributs

Les attributs sont faits pour contenir des informations courtes et précises relatives aux éléments et peuvent être soumis à plusieurs règles contraignantes, ce qui en fait une sorte d'avantage pour ceux-ci :

- ils peuvent être contraints à une liste de valeurs prédéterminées;
- ils peuvent avoir des valeurs par défaut;
- ils peuvent être sujets à des types de données (chiffres, chaîne de caractères, date, etc.);
- ils contiennent de l'information concise.

En revanche, les attributs ne peuvent contenir de très longues chaînes de caractères ou contenir de l'information hiérarchisée. De plus, on ne peut ignorer les espaces blancs (sans caractères) au sein des attributs. L'annexe A.4 présente les informations nécessaires à la déclaration des attributs dans un DTD. Voyons maintenant comment sont déclarés et régis les éléments au sein d'un DTD.

2.2.1.2 Les éléments

Les éléments, pour leur part, sont déclarés ainsi au sein d'un DTD :

```
<!ELEMENT NomElement Type >
```

Ainsi, le nom de l'élément va déterminer le nom des balises utilisées pour décrire ceux-ci dans le document XML. Ce nom d'élément doit bien sûr être unique pour le DTD en question. Pour ce qui est du Type, le XML supporte 4 différents types d'éléments qui sont déterminés à partir de leur contenu :

- `Empty` – L'élément ne contient aucun contenu. (Il peut cependant contenir des attributs et c'est la plus part du temps le cas);

- `Element-only` – L'élément contient seulement des éléments enfants;
- `Mixed` – l'élément contient à la fois des éléments enfants et du contenu (caractères);
- `Any` – l'élément peut contenir tout ce qui est permis par le DTD. (Un élément qui a carte blanche sur sa structure).

On retrouve en annexe A.5 des précisions supplémentaires sur les déclarations d'éléments de type «Element-only» et «Mixed». De plus, en annexe A.6 on retrouve un exemple complet d'un DTD qui permettra de bien saisir l'ensemble de ce qui a été énoncé dans cette section.

Maintenant que l'on connaît les DTD et leurs utilités, il faut aussi regarder les XSD qui accomplissent le même travail, mais qui utilisent les mêmes notations et les mêmes règles que le XML.

2.2.2 Les XSD

Tout d'abord, comme mentionné précédemment, vu que les schémas XSD utilisent la même structure et la même syntaxe que le XML, il ne reste que les spécificités du langage à apprendre pour pouvoir l'utiliser. Cependant, énumérer toutes les spécificités et les particularités des XSD serait beaucoup trop long et dépasserait de beaucoup le cadre de ce mémoire. Un bref résumé de son utilisation sera donc présenté, en plus de montrer un exemple XSD pour le même fichier XML `entrainementjournal` présenté auparavant (se référer à l'annexe A.6).

- les XSD sont beaucoup plus flexibles que les DTD ils permettent la déclaration et la validation de plusieurs types de données comme les entiers les dates, les booléens, les chaînes de caractères, des durées, etc;
- en plus de permettre plusieurs types de données, les schémas XSD permettent de déclarer des types « sur mesure » à l'aide de la balise `-xsd:simpleType`. À l'aide de ces types personnalisés, on peut restreindre davantage la définition des types à

l'aide de l'élément `xsd:restriction` entre autres. On pourrait par exemple se créer un type personnalisé pour une valeur d'attribut qui ne peut qu'être qu'un entier de 1 à 10;

- les énumérations de valeurs sont en fait traitées comme des types personnalisés dans les XSD. (Voir l'exemple en annexe A.7).

Ces trois premiers points de la liste se portaient bien à la déclaration des attributs. Par contre, pour décrire les éléments et leurs relations entre eux, il faut faire appel aux éléments `xsd:complexType` qui en conjonction avec `xsd:complexContent` peuvent mettre en valeur ces relations. Ainsi les éléments qui ont des enfants, verront ceux-ci déclarés en tant que type complexe sur mesure, puisqu'il n'existe pas de types complexes déjà faits comme il en existe pour les attributs (`xsd:integer` etc.). Ainsi on peut déclarer des éléments enfants qui peuvent être des choix, ou bien des séquences prédéfinies d'éléments. On retrouve en annexe A.7 l'exemple en XSD du DTD en annexe A.6 :

Bien que plus long, le XSD est certainement le plus flexible et le plus sécuritaire des deux types de schémas, surtout qu'il permet de définir sans ambiguïté les différents types de données que l'on doit s'attendre à trouver dans le fichier XML. Ceci est d'autant plus important pour l'échange de données informatisées EDI qui demandent un minimum d'information et de structure pour bien préparer les différents champs des bases de données en fonction du type de données qu'elles vont recevoir. Voyons maintenant le XSL et comment il permet de mettre en forme et de transformer les documents XML.

2.3 Transformer les données XML à l'aide du XSL

Dans cette section nous verrons un survol des caractéristiques du XSL qui est une technologie de feuille de style, mais qui permet d'en faire beaucoup plus que celles que le lecteur connaît peut-être déjà, comme les CSS (*Cascading Style Sheets*). En effet, comme le XSL a seulement été construit afin de servir de feuille de style au XML, celui-ci permet donc d'aller plus loin en offrant la possibilité de modifier la structure des

documents XML et de transformer leur contenu. La programmation du XSL se fait à l'aide de code XML puisqu'il en est un langage dérivé. Actuellement, le composant du XSL qui est supporté par les navigateurs Web est le XSLT qui nous permet de transformer un document XML en un contenu visible et agréable pour l'utilisateur (via le code de la feuille de style). Il permet donc de transformer du XML en HTML afin d'être présenté dans un navigateur Web à l'aide d'une feuille de style CSS si désiré. Tout ceci peut sembler complexe, mais un exemple viendra éclaircir le sujet plus loin dans la section (Lejeune, 2004, pp. 77-95; Morrison, 2002).

Afin de comprendre comment le XSL fonctionne, il faut partir du document XML. Avant qu'une feuille XSL soit appliquée à un document XML, un processeur XML «lit» le document XML et détermine sa structure hiérarchique afin de créer ni plus ni moins une arborescence de nœuds contenant l'information du fichier.

Une fois cet «arbre» mis en place, le processeur XSL peut appliquer ses règles de transformations sur l'arbre ou certains des nœuds dépendamment du contenu du fichier XSL. Le processeur part donc du haut de l'arbre, soit du nœud racine, et essaie de faire correspondre ces patrons aux différents nœuds de l'arbre dans le but de leur appliquer les styles XSL. Ces patrons sont enregistrés dans ce que l'on appelle des «templates». Ainsi, tout en parcourant l'arbre du document, lorsqu'une correspondance est trouvée, la portion de l'arbre associée qui correspond au patron courant est transformée par le *template* approprié. Les règles de transformation de ce *template* sont donc appliquées et un arbre de résultat «result tree» est généré. Appliquons ceci à la figure suivante :

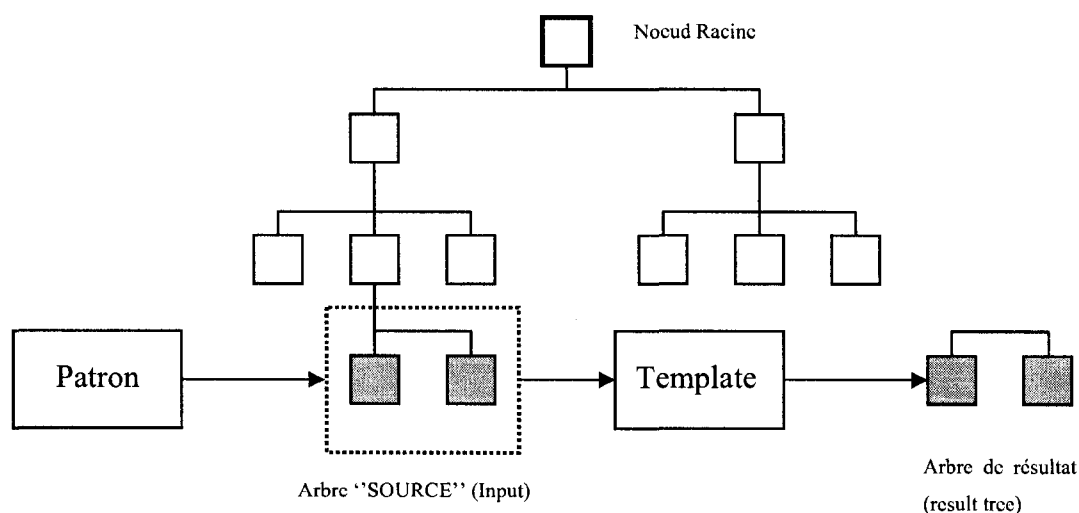


Figure 2-1 Identification d'un patron et application d'un template (inspiré de Lejeune, 2004)

Donc, en résumé, les différents patrons du document XSL parcourent l'arbre de données XML afin de trouver le ou les nœuds (portion d'arbre) correspondants. Ensuite à chacun de ces patrons sont associées des *templates* qui appliquent les règles de transformation aux nœuds sélectionnés afin de donner un nouvel arbre de résultats.

L'arbre de résultats peut correspondre à plusieurs choses, car comme on l'a indiqué en début de section, le XSL est plus qu'un langage de style, c'est un langage de transformation. Voici une liste d'exemples de type de contenu que l'on peut générer à l'aide du XSL :

- il peut s'agir de code HTML qui peut être visualisé par un navigateur Web;
- il peut s'agir d'un tout nouveau document XML;
- il pourrait aussi s'agir d'un document SVG interactif, pouvant lui aussi être visualisé dans un navigateur Web.

Voyons maintenant plus en détail comment le XSL effectue ces transformations et avec quels outils il les réalise.

Afin de sélectionner les bons nœuds et d'appliquer les bonnes transformations, le XSL utilise une combinaison d'objets XSLT avec un langage d'adressage XPath. Le XSLT

est directement dérivé du XML et utilise donc un langage balisé. XPath par contre n'est pas un langage dérivé du XML, il est utilisé dans un contexte où le langage XML n'est pas applicable, soit en tant qu'attribut des objets XSLT.

Donc les objets XSLT sont un ensemble de balises de test ou de sélection qui possèdent comme attribut des expressions XPath afin de cibler les nœuds désirés. Ainsi, le parseur (engin intégré aux navigateurs Web qui traite les règles XML et XSL) XSL vérifie les différentes conditions de ces objets et appliquera les *templates* pour chaque nœud qui passent le test, si l'on peut s'exprimer de cette façon.

On peut retrouver en annexe A.8 des précisions sur le fonctionnement de XPath ainsi que des informations supplémentaires sur les éléments XSLT qui sont le plus fréquemment utilisés. De plus, des exemples complets, utilisés dans le cadre de mon projet, seront montrés en annexe C.4.

2.3.1 Limites du XSL

Bien que le XSL soit assez flexible et qu'il permette d'effectuer la majorité des tâches d'un utilisateur moyen, ce langage présente toutefois certaines limites notamment au niveau de la dérivation d'information complexe. Par exemple, lorsqu'on a des calculs à effectuer en boucle et ainsi à augmenter la valeur d'une variable à chaque itération, le XSL ne nous permet pas d'effectuer cette opération toute banale en programmation :

```
x = x + 1;
```

On présente en annexe A.9 une portion de code nous permettant de contourner cette lacune du XSL.

Aussi, quand ce que l'on veut calculer devient trop compliqué, l'on peut faire appel au *javascript* ou au *vbscript* afin de programmer ce qui nous est nécessaire dans un véritable langage de programmation. Il s'agit alors d'appeler une fonction *javascript* ou autre dans un objet `xsl:value-of` et ainsi apparaîtra dans l'arbre de résultats, la valeur retournée par la fonction. Pour terminer la section sur le XML, un tableau présentant quelques autres technologies reliées au XML sera présenté dans la prochaine section.

2.4 Autres technologies reliées XML

Le XML devenant de plus en plus populaire autant sur le Web que dans les entreprises utilisant des EDI, plusieurs technologies relatives au XML se sont développées. Elles sont quelques fois dérivées du XML même, donc utilisent la même structure ou bien sont totalement indépendantes. Voici donc un tableau récapitulatif de quelques unes de ces technologies.

Tableau 2-2 Autres technologies reliées au XML

<i>Nom</i>	<i>Description</i>
XQuery	<p>Permet d'effectuer des requêtes sur un document XML afin de retourner des valeurs précises d'un ou plusieurs éléments ou attributs. Il s'agit un peu du SQL pour le XML. XQuery utilise une syntaxe ressemblant aux langages de programmation connus avec un mélange de XPath :</p> <pre data-bbox="500 1059 1034 1108">For \$v in //vehicules [@annee="2005"] Return \$v</pre> <p>Donc cette requête retournerait tous les éléments véhicules ayant un attribut <i>annee</i> valant 2005.</p>
XPointer	<p>XPointer ressemble beaucoup à XPath. Il permet de s'adresser à certains regroupements de nœuds au sein de l'arbre de données XML en plus de permettre l'adressage dans XLink pour rechercher ou référencer des éléments dans un autre document XML par exemple.</p>
XLink	<p>La seule raison d'être du XPointer est d'apporter un moyen de référencer des portions d'autres documents XML (externes) afin de créer des liens XLink puissants. Xlink utilise des éléments de liens, <i>linking elements</i>, qui permettent de décrire les caractéristiques des liens. Voici un exemple simple qui crée un lien vers un élément employé :</p> <pre data-bbox="500 1655 1290 1761"><employees xmlns:xlink=http://www.w3.org/1999/xlink xlink:href="employes.xml"> Current Employes </employees></pre> <p>Pour chaque élément XLink, la portion XPointer se situe dans la valeur de l'attribut <i>href</i>.</p>

Tableau 2-3 Autres technologies reliées au XML (suite et fin)

XHTML	Version de HTML suivant les mêmes règles de structures que le XML.
Autres	Il existe différentes adaptations de ces différentes technologies développées spécialement pour les interfaces mobiles ou les interfaces adaptées aux personnes non-voyantes, etc.

Dans la prochaine section, il sera présenté une autre technologie dérivée du XML mais qui mérite une section entière puisqu'il occupe une place importante dans le cadre de ce projet.

2.5 Le SVG

Le *Scalable Vector Graphics* (a.k.a SVG) est une autre des technologies de diffusion de données utilisée dans le cadre de ce projet. Sa capacité à créer des graphiques interactifs et sa compatibilité avec la plupart des navigateurs Web en font un outil de choix pour atteindre certains des objectifs de ce mémoire. Les prochaines sections relatent ce qu'est le SVG et présentent un bref aperçu historique en plus de fournir quelques exemples explicatifs.

2.5.1 Définition et historique (Fundamentals of SVG programming)

Le SVG est un langage de représentation d'objets géométriques pouvant être manipulés et représentés au moyen de fonctions SVG. Cette technologie nous permet ni plus ni moins de contrôler l'affichage de chacun des pixels de l'écran afin d'y faire afficher du texte ou des images. Mais attention, cela veut dire que dans une image, chaque forme géométrique de base doit y être codée et décrite, ce qui veut dire qu'il n'y a rien de gratuit si l'on peut s'exprimer ainsi. De plus, parmi les vastes spécifications du SVG, celui-ci permet l'utilisation du langage de script javascript/ECMAScript afin d'introduire des effets d'animation et de permettre la création et la manipulation de patrons visuels complexes.

Le SVG est un autre des langages dérivés du XML, qui utilise donc sa structure et ses règles. Il est lui aussi régi par le W3C. La création du SVG est due au travail collaboratif de plusieurs grandes compagnies du monde informatique tel que : Adobe, Hewlett-Packard, IBM, Microsoft, Netscape et SUN Microsystems. Le SVG est actuellement à la version 1.1 recommandée depuis le 15 novembre 2002 par le W3C et il y a aussi une version 1.2 en version brouillon depuis ce temps (Campesato, 2004).

Voici une petite liste des avantages du SVG :

- il est compact et le code source portable;
- il fonctionne sous plusieurs systèmes d'exploitation et ne nombreux navigateurs Web;
- on ne perd pas de résolution lorsque l'on zoom sur un dessin SVG;
- on peut créer des images complexes grâce à des transformations, des rotations et des ajustements d'échelles, ce qui serait très fastidieux dans un autre langage;
- il est facilement généré sur le Web ou par des applications Java;
- on peut transformer à l'aide du XSLT, le contenu de documents XML en SVG.

Voyons maintenant, dans la prochaine section quelques objets du SVG à l'aide d'un exemple.

2.5.2 Exemple et utilisation

Avant de commencer, voici un petit schéma de «l'anatomie» d'un dessin SVG type que l'on retrouve à la page suivante.

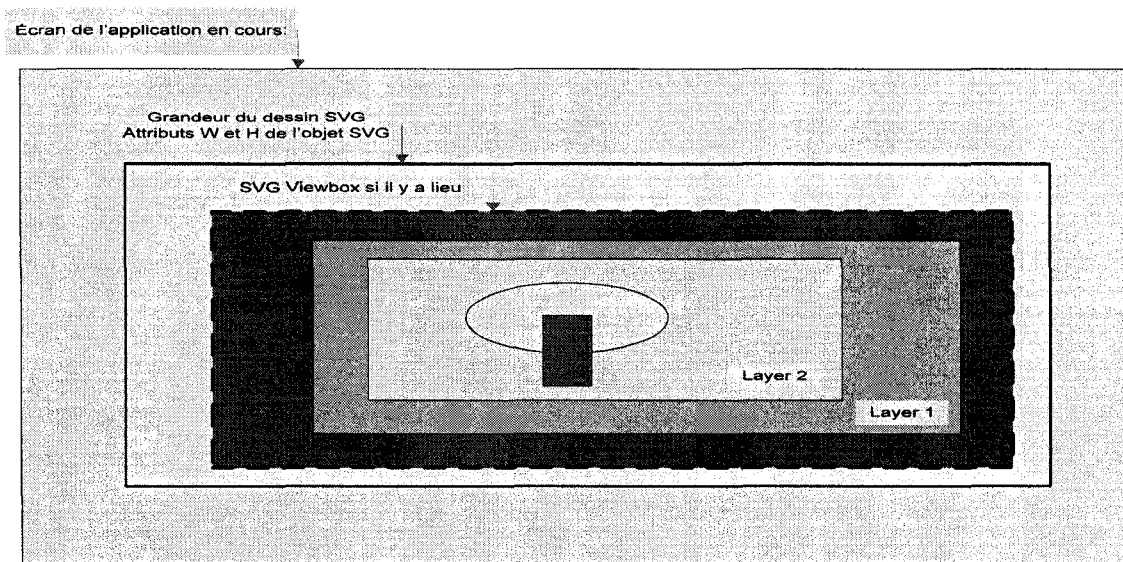


Figure 2-2 Anatomie d'un dessin type en SVG

Dans l'écran de l'application en cours où l'on verra s'afficher le SVG, le créateur du fichier doit décider de la grandeur que prendra le dessin en spécifiant les attributs *Width* et *Height* de l'élément SVG qui est l'élément racine de tout dessin SVG.

```
<svg width="800" height="400" >
```

La plupart du temps, le rectangle qui contient le dessin n'est pas délimité par une ligne frontière et donc l'utilisateur en bout de ligne ne voit pas ce rectangle virtuel mais il est bien là. Ensuite, le créateur a aussi la possibilité de créer ce que l'on appelle un *viewbox*, qui permet de décaler le rectangle de dessin original en déclarant deux nouvelles coordonnées pour le coin supérieur gauche et le coin inférieur droit du nouveau rectangle. Les valeurs que l'on rentrera ensuite, par exemple, le centre d'un cercle en x et en y, seront calculés à partir des nouvelles coordonnées. Le *viewbox* est aussi un attribut de l'élément SVG et on entre ses valeurs ainsi :

```
<svg width="800" height="400" viewBox="0 0 400 800" >
```

Les coordonnées du *viewbox* entrées ci-dessus, n'entraînerait aucun déphasage du dessin par rapport au rectangle original.

Finalement, avant de dessiner des formes comme tel, il faut spécifier sur quels *layers* ou couches on va dessiner. Ceci va bien sûr avoir une incidence sur ce que l'on va voir au premier plan ou en arrière-plan. Un layer se définit tout simplement à l'aide d'un élément `<g>` la plupart du temps, sans attributs. Voici maintenant un petit exemple du dessin d'un rectangle par-dessus un rond :

-le code :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg viewBox="0 0 1024 768" width="100%" height="100%"
xmlns:svg="http://www.w3.org/2000/svg" color-interpolation-filters="sRGB">
  <g>
    <circle cx="100" cy="100" r="60" fill="blue" id="rond"
stroke="red" onclick="modifRond(evt, 'rond');" />
    <g>
      <rect x="110" y="100" height="30.5" width="102.5"
fill="red" stroke="black"/>
    </g>
  </g>
</svg>
```

Comme le deuxième layer qui contient le *rect* est imbriqué, si on le veut comme élément enfant du premier, ce layer est donc en avant-plan et le rectangle se dessine par-dessus le rond comme on peut le voir sur l'image ci-contre :

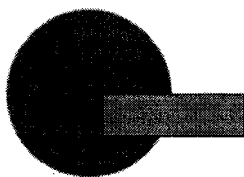


Figure 2-3 Dessin SVG d'un rectangle par dessus un rond

Ce petit exemple est bien sûr peu représentatif de tout ce que l'on peut faire avec le SVG, mais il permet d'en comprendre un peu mieux le fonctionnement. Voyons maintenant comment on peut ajouter un peu d'interactivité aux dessins SVG.

2.5.3 SVG et interactivité grâce au ECMA script

Une des grandes forces du SVG est de pouvoir ajouter de l'interactivité au dessin à l'aide du ECMA script, un langage de «scriptage» ressemblant au javascript. Ce code permet d'accéder à tous les éléments qui composent le dessin SVG et de modifier à notre guise leurs attributs. On pourrait par exemple exécuter les opérations suivantes :

- modifier la couleur ou la grandeur d'une forme;
- afficher ou éteindre certaines formes ou même certains layers;
- faire afficher ou mettre en évidence certains textes au sein du dessin;
- animer certaines formes (déplacements ou changements de grandeurs);
- etc.

Toutes ces opérations ne se font pas d'elles-mêmes. Par contre, elles se produisent à la suite de l'accomplissement d'événements précis. Il s'agit donc de procédures événementielles. Voici la plupart des événements utilisés pour déclencher ces procédures :

- sur clic ou double-clic;
- sur passage du curseur de la souris au-dessus de la forme visée;
- sur retrait du curseur de la souris;
- sur chargement de la page;
- etc.

On retrouve en annexe A.10 un exemple de code complet d'interactivité avec le ECMS script sur le même dessin SVG présenté à la section précédente.

La prochaine section porte sur les différentes technologies d'extraction de données nous permettant de générer des fichier XML à partir de bases de données.

2.6 Technologies d'extraction et de publication

On a vu dans ce chapitre, le fonctionnement et le potentiel, en terme de technologie de diffusion d'information, du XML. Toutes fois les données que l'on souhaite diffuser vont être enregistrées, dans la très grande majorité des cas, au sein d'une base de données. Vu la popularité grandissante du XML, plusieurs outils ont été développés au cours des dernières années afin de faciliter l'extraction des données en documents XML hiérarchisés au gré des utilisateurs. Deux techniques d'extraction seront présentées ici.

De plus, lorsque nous avons les documents désirés en notre possession, il faut les publier sur le web et créer une page de navigation qui permettra d'afficher les bons résultats en fonction des requêtes des utilisateurs. Nous verrons que ceci peut être réalisé à l'aide des pages ASP (*Active Server Page*). Package, programmes VB, intégrés

2.6.1 De la base de donnée au XML

Comme mentionné en introduction de cette section, il y a de très forte chance que les données que l'on souhaite introduire au sein de nos documents XML proviennent de bases de données. Afin de faciliter la tâche d'extraction des données au format XML, les développeurs se sont inspirés du langage d'interrogation des bases de données, le SQL (*Structured Query Language*). Ils ont donc créé le RDBXML (*Relational DataBase eXtended Markup Language*) qui combine le pouvoir de hiérarchisation (éléments parents-enfants et attributs) du XML avec les instructions classiques du SQL. On peut donc créer des éléments qui ne sont pas nécessairement en lien avec des champs de notre base de donnée et leurs affecter des attributs qui, au contraire, sont directement reliés à des valeurs contenues dans la source de donnée. L'exemple suivant, réalisé à l'aide du logiciel «Stylus Studio Entreprise Edition 2007» peut permettre au lecteur de mieux saisir le concept. Cet outil d'extraction a d'ailleurs été utilisé afin de créer des versions préliminaires des documents XML nécessaires au projet. On a identifié en gras les mots clés relatif au SQL.

```

XMLEMENT(name "LI",
  XMLATTRIBUTES (y.num_li AS "NOLI", y.DIR2 AS "DIR"),
  XMLEMENT(name "VOYS",
    (
      SELECT
        XMLEMENT(name "VOY",
          XMLATTRIBUTES (v.idVoyage AS "IDV", v.HeureDebut AS "HD",
v.HeureFin AS "HF", v.idCourse AS "IDC"))
          FROM STO.dbo.sivt_Voyage_Reel v, STO.dbo.sivt_Course_Reel c
          WHERE v.numLigne = y.num_li AND v.Direction = y.DIR2 AND
v.idCourse = c.idCourse AND c.DateService = '1/6/2005'
          ORDER BY v.HeureDebut
        ),
      XMLEMENT(name "ARRL",
        (
          SELECT
            XMLEMENT(name "ARR",
              XMLATTRIBUTES (t.NUM_ARRET AS "NOA",t.NUM_ORDRE AS "ODR",
u.COORD_X AS "X",u.COORD_Y AS "Y"))
              FROM STO.dbo.SSLI_ARRET t, STO.dbo.ARRET u
              WHERE t.NUM_LI = y.num_li AND t.NUM_SSLI = y.num_ssli AND
t.PeriodeDebut <= '1/6/2005' AND T.PeriodeFin >= '1/6/2005' AND u.NUM_ARRET
= t.NUM_ARRET AND u.PeriodeDebut <= '1/6/2005' AND u.PeriodeFin >=
'1/6/2005'
              ORDER BY t.NUM_ORDRE )
          )
        )
      )
    FROM STO.dbo.testpoly_dir_ok y
    WHERE y.Periodedebut <= '1/6/2005' AND y.PeriodeFin >= '1/6/2005'
    ORDER BY y.num_li, y.DIR2;

```

On peut voir ici qu'on a à la source l'élément LI contenant un ensemble de VOY (voyage) au sein de l'élément parent VOYS (voyages) ainsi qu'un ensemble d'ARR (arrêt) au sein de l'élément parent ARRL (arrêt - ligne). Chacun de ces éléments contiennent des attributs dont les valeurs sont directement reliées à des champs de la base de donnée comme on le voit par des instructions SELECT.

Des outils d'extraction similaires sont aussi intégrés directement aux serveurs de base de données tels que SQL Server de Microsoft (depuis la version 2000) et Oracle.

Toutefois il se peut que l'on ait à transformer de façon considérable (utiliser l'information afin d'en dériver d'autre par exemple) les données contenues dans la base de données avant de les inclure dans le fichier XML final. Les techniques d'extraction telle que celle mentionnée précédemment deviennent alors insuffisantes et on doit alors se pencher vers la création de véritables programmes d'extraction personnalisés à la

tâche à accomplir. On peut réaliser de tels programmes à l'aide VB.Net (*Visual Basic.Net*) par exemple, qui a été utilisé dans le cadre de ce projet. La partie d'extraction à partir de la base de donnée se fait alors à l'aide d'un «package DTS» (*Data Transformation Services*) qui permet une extraction rapide et paramétrée des données désirée afin de minimiser la charge de travail sur le serveur de base de donnée. La mise en forme du document XML se fait donc à posteriori à l'aide du programme VB.Net une fois les données nécessaires extraites. Des détails, concernant le programme d'extraction réalisé, seront mentionnés au chapitre 4 ainsi qu'en annexe C.2.

2.6.2 Site Web interactif avec ASP

L'ASP est un langage de programmation, développé par Microsoft, qui permet de créer des pages Web dynamiques. Plus précisément, il permet d'exécuter des scripts du côté serveur (la plupart du temps VB Script ou Java Script) qui prennent en considération les instruction conditionnelles afin de réaliser des tâches simples ou compliquées, le tout, directement à partir du navigateur Web (Herellier & Mérigod, 1999)

L'ASP permet donc de prendre en considération les différentes requêtes de l'utilisateur qui interagit avec la page (à l'aide d'un formulaire) et de renvoyer le contenu désiré (paramétré en fonction de la requête) dans un format compréhensible par le navigateur Web, soit par exemple du HTML ou du SVG.

C'est l'ASP qui est utilisé dans ce projet afin de gérer les requêtes des utilisateurs sur le site Web créé. Le code de la page ASP de départ qui est utilisé sera présenté en annexe C.3.

CHAPITRE 3: MÉTHODOLOGIE

3.1 Considérations pour la méthodologie générale du projet

La méthodologie de cette recherche s'appuie à la base sur l'approche orientée - objet dans les systèmes de transport. Il est essentiel de bien identifier et comprendre les objets et les types de liens qui les unissent afin de construire une structure de données solide destinée à l'outil générateur de statistiques. De plus, ce type d'approche par objet nous assure un haut niveau de compatibilité de l'outil développé avec d'autres utilisations potentielles à la STO.

Nous nous attarderons d'abord sur le modèle objet de la STO afin de comprendre sur quoi les chercheurs précédents ont basé leur méthodologie afin de dériver les destinations de chacun des déplacements des usagers sur le réseau. L'estimation des destinations est une composante clé dans ce projet puisque la plupart des statistiques que nous dérivons prennent en considération la connaissance des points de descente associés aux transactions effectuées sur le réseau. De plus, comme nous le verrons dans ce chapitre, l'analyse du modèle objet par rapport aux destinations nous aura permis d'affiner notre méthodologie générale de projet.

Suite à cette analyse, le détail des étapes de modélisation préliminaire de la méthodologie générale, qui est tout de suite présentée ici en figure 3-1, sera développé en section 3.2.

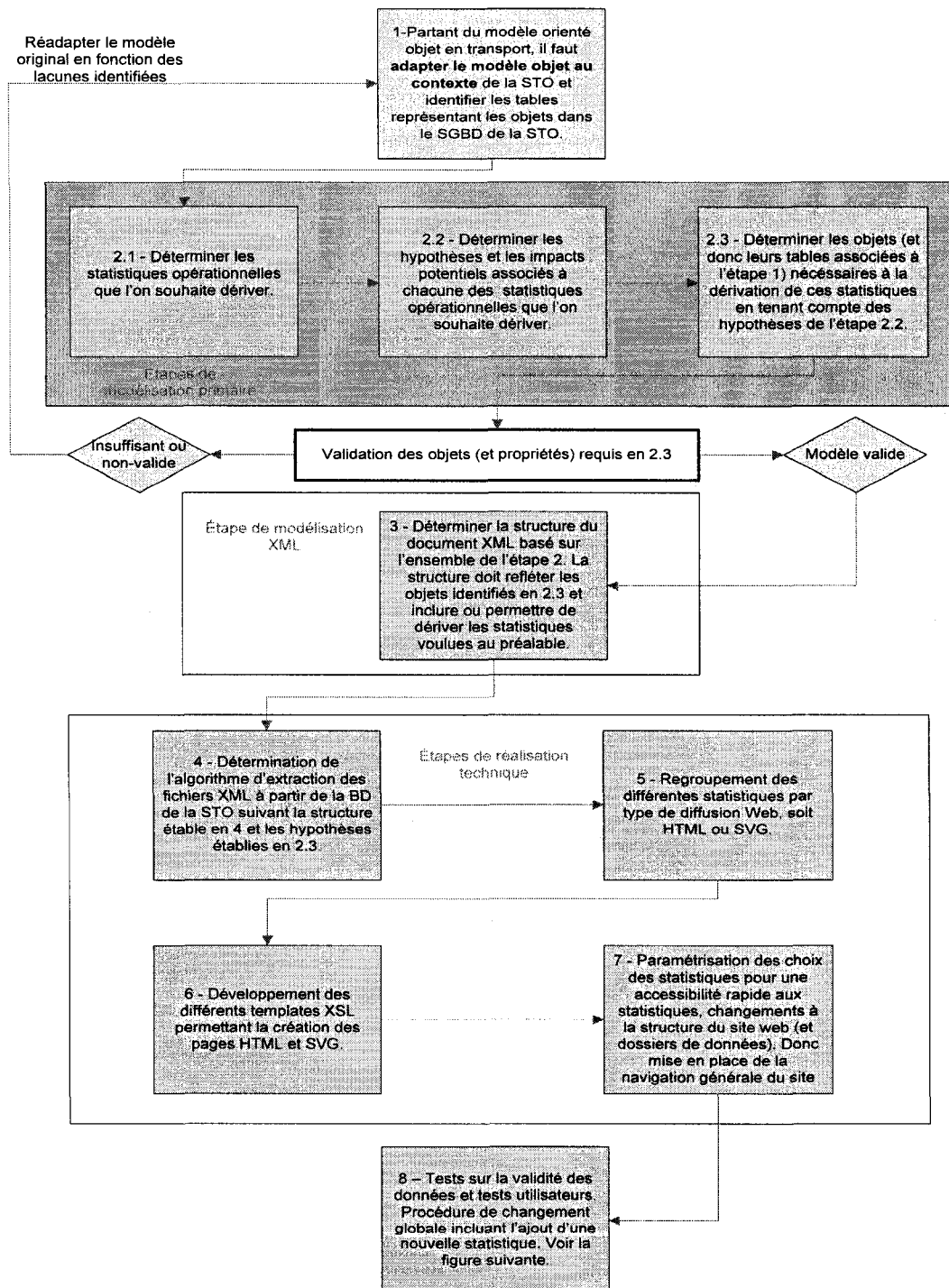


Figure 3-1 Méthodologie générale du projet

3.1.1 Données et modélisation objet du système de transport de la STO

Avant de présenter le modèle objet du système de transport de la STO, voici une brève présentation des données de cartes à puces et du système SIVT (base de données) de la société de transport.

3.1.1.1 Brève présentation des données de cartes à puces

Au cœur du système de perception par carte à puces se trouve le Système Informatisé de Validation des Titres (SIVT) dont l'objectif est de faciliter la distribution, la perception et le contrôle des titres de transport. L'objectif secondaire de ce système est d'utiliser les données à des fins de contrôle, de statistiques et de planification.

La figure suivante présente les 4 sous-systèmes qui composent le système SIVT ainsi que les différents flux d'informations qui circulent entre ceux-ci. Les différents groupes seront explicités par la suite (Tranchant, 2005).

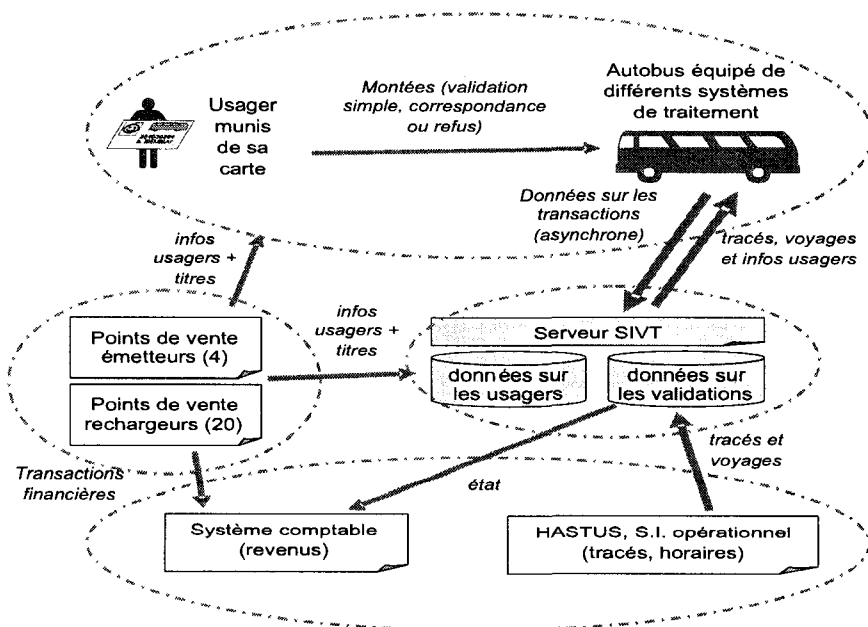


Figure 3-2 Diagramme des interfaces entre les systèmes du SIVT (inspiré de TRÉPANIÉ 2004)

- Le sous-système « embarqué ». Il comporte les sous-systèmes « carte à puces » et « Bus » qui interagissent ensemble afin de valider les montées, localiser les embarquements, les compter, permettre une communication continue avec les chauffeurs et transmettre les données au sous-système central.
- Le sous-système « point de vente ». Celui-ci a été séparé en deux autres sous-systèmes : points de vente «émetteurs» (initialisation électronique et personnalisation physique de cartes) et points de vente «rechargeurs» (émission et rechargement de cartes, mensuel, etc.). Un rôle additionnel consiste à envoyer des données au système central et aux sous-systèmes externes. Les points de vente sont en relation avec le système comptable de la STO pour assurer l'intégrité des transactions financières.
- Les deux sous-systèmes externes « HASTUS » et « système comptable ». Le premier est un logiciel de planification donnant des informations au serveur SIVT sur la gestion de l'exploitation du réseau et des chauffeurs ; le deuxième nous renseigne sur la partie administrative de la STO. Leur rôle est donc d'interagir avec le système central (serveur SIVT).
- Le sous-système « central » (serveur SIVT) Son rôle principal est d'interagir avec tous les autres sous-systèmes, que ce soit pour configurer des appareils à distance ou pour rapatrier des données, mais aussi de traiter l'information reçue et la stocker.

Dans le sous-système bus, on distingue trois composantes principales :

- le système de validation par cartes à puces;
- le système de comptage des passagers;
- le système de localisation.

Le système de validation par carte à puces interagit directement avec la carte à puces personnalisée de l'utilisateur par onde radio. Ce système assure la validation ou le refus des

titres de transports contenus sur la carte à puces des usagers montant à bord. Le système de validation possède de plus une interface d'interaction avec le chauffeur lui permettant de valider, aux terminus, chacun des voyages que le système central lui demande d'effectuer. Une unité centrale maîtresse coordonne les données de positionnement (système de localisation) GPS avec les données de cartes à puces. Dans le cas idéal, toutes les données transactionnelles de la journée sont transmises par infrarouge au système central une fois la journée terminée.

Le système de comptage compte le nombre d'usager montant et descendant de l'autobus à l'aide d'un ordinateur et de capteurs infrarouges. Toutefois il n'est pas relié au système GPS et il ne recommence pas à compter à chaque voyage.

Finalement, le **système de localisation** traite les données GPS afin de coordonner la valeur courante de la position de l'autobus avec la valeur du plus proche arrêt dans le voyage validé (le chauffeur doit indiquer sur quelle ligne et dans quelle direction il va circuler) au préalable par le chauffeur. Les données GPS brutes ne sont pas stockées dans le système central.

Toutes ces composantes permettent de fournir, au jour le jour, une quantité impressionnante de données riches en informations spatio-temporelles, opérationnelles et individuelles. Toutefois, il est évident que toutes les données nominales relatives aux usagers ou aux employés sont écartées de tout traitement, puisqu'elles sont sujettes à des clauses de confidentialité.

À noter également que les informations recueillies par le système peuvent comporter des erreurs. Ceci serait principalement dû à des erreurs humaines de la part des chauffeurs qui doivent interagir avec le système de validation pour saisir le bon voyage (ligne, direction, heure de départ) à effectuer. Si le mauvais voyage (mauvaise combinaison de ligne - direction surtout) est sélectionnée, on ne pourra pas faire de correspondances entre le positionnement de l'autobus et les coordonnées des arrêts prévus (Chapleau & Chu, 2007a).

Maintenant que nous avons vu comment les données sont prélevées, voyons maintenant le modèle objet du système de transport de la STO.

3.1.1.2 Présentation du modèle objet de la STO

À partir du système SIVT recueillant les données des cartes à puces, il est possible d'identifier plusieurs objets qui peuvent correspondre ou non à différents éléments de la base de données. La méthode utilisée ici est l'approche orientée objet en transport qui, comme on l'a vu au chapitre 1, découle directement de l'Approche Totalement Désagrégée. Elle considère donc une analyse individuelle des objets du système de transport en plus de préconiser leur état et leur comportement à travers le temps et l'espace (Trépanier, 1999). Il est à noter que la modélisation objet peut se faire en se basant sur un modèle relationnel de données (MRD) d'une base de données d'un réseau de transport, mais qu'elle peut aussi être faite indépendamment des données disponibles. En effet, un modèle objet robuste et assez général pourrait très bien servir de base à la création d'une base de données d'un nouveau réseau de transport. De plus, l'ajout de nouveaux objets, pour tenir compte des destinations par exemple, entraînera forcément des changements, du côté du MRD et de ses tables associées.

La figure 3-3, montre le modèle objet de la STO utilisé dans le cadre de ce projet. De plus, on y retrouve le nombre d'instances des différentes classes d'objets en fonction des données de janvier 2007. Le modèle objet est séparé en 4 grands groupes d'objets :

- A. Les objets réseau sont les éléments « créés » par la STO qui définissent le réseau sur lequel les autobus roulent (autobus, lignes, lignes arrêts);
- B. Les objets opérationnels STO sont reliés aux opérations des autobus de la STO (chauffeurs, autobus, garage, voiture);
- C. Les objets administratifs STO sont reliés aux éléments à tout l'aspect perception des cartes à puces (titres, points de transactions, cartes);
- D. Les objets Usagers de la STO, partant de la demande des usagers. (usagers, chaîne de déplacements, déplacements).

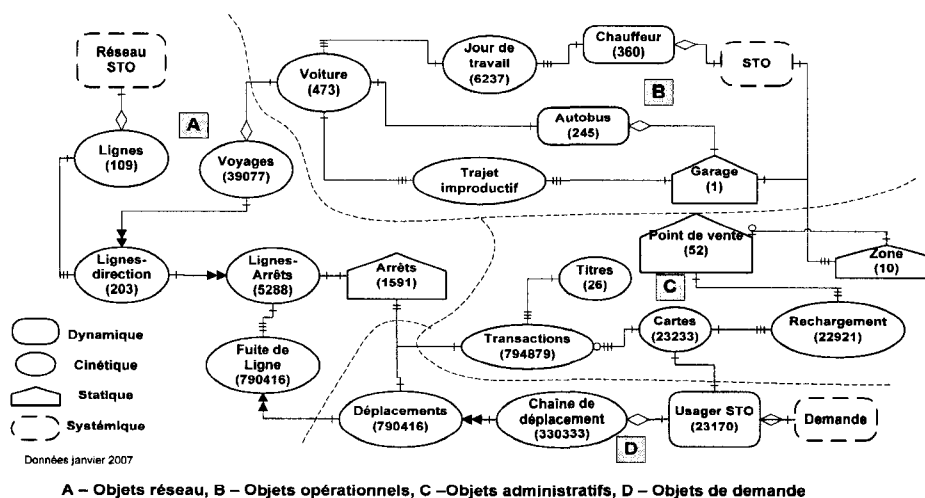


Figure 3-3 Modèle objet du réseau de transport de la STO

Le diagramme utilisé par Tranchant dans son mémoire différait un peu de celui-ci, car il ne tenait pas encore compte des objets servant à dériver les destinations des déplacements. La figure suivante reprend une partie du diagramme précédent pour y indiquer les différences avec l'ancien diagramme.

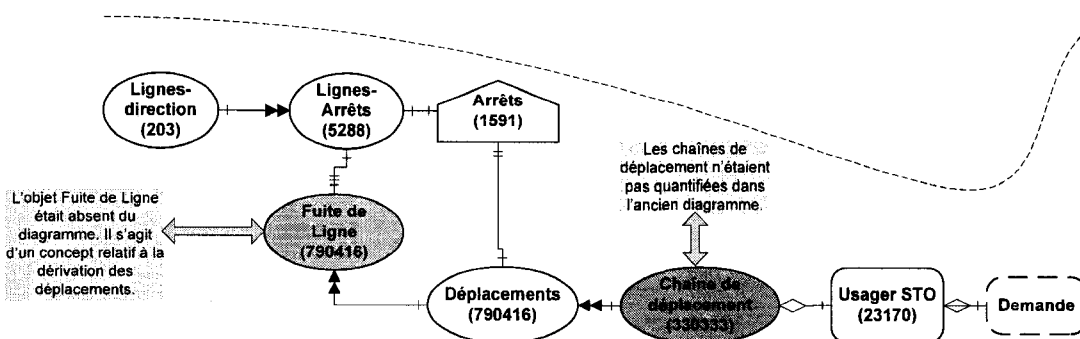


Figure 3-4 Différence sur la modélisation objet entre 2007 et 2004

Avant d'expliquer les différences notées dans cette figure, il est important de définir certains des objets de ce diagramme. Le tout sera repris un peu plus en détail lors de l'explication de l'algorithme de dérivation des destinations (Trépanier et al., 2007) sera couvert.

- **Chaîne de déplacement** : Un usager u effectue dans une journée p (unité de base pour le recensement des déplacements d'un usager; on pourrait aussi utiliser une

unité de temps plus grande, tel que la semaine ou le mois.) un ensemble V_{up} de D^j_{up} déplacements avec ($j \in \{1, \dots, X_p\}$) ou X_p est un nombre entier réel de déplacements effectués dans la journée. L'origine du premier déplacement est l'origine de la chaîne et la destination du dernier déplacement est la destination de la chaîne. Ceci implique aussi que dès qu'un usager effectue une chaîne de déplacement dans une journée, il y a obligatoirement au minimum un déplacement. Le concept des chaînes de déplacement est très utile pour dériver les destinations finales en étudiant la régularité des chaînes.

- Fuite de ligne : Sur une ligne L dans un sens S, définit par un ensemble Q de N arrêts A_i sujet à un ordre O_i ($i \in \{1, \dots, N\}$) un usager u (au jour p , comme pour l'exemple des chaînes) monte à l'arrêt A_k d'ordre O_k pour effectuer un déplacement D^j_{up} . La fuite de ligne de ce déplacement D^j_{up} sera donc l'ensemble $Q' \in Q$ de $N - k$ arrêts A_l sujet à un ordre O_l ($l \in \{k + 1, \dots, N\}$). En d'autres mots, l'ensemble des arrêts lignes restant, d'ordre supérieur à l'arrêt A_k d'embarquement. Le concept des fuites de lignes est utilisé entre autres pour trouver l'arrêt de descente lorsqu'une transaction de correspondance est enregistrée à une heure ultérieure à l'heure de transaction à l'arrêt A_k .

Comme on le voit sur la figure 3-4, il y a deux principaux changements : premièrement, à chacun des déplacements est maintenant associé une « Fuite de ligne » correspondant à l'ensemble des arrêts ayant un ordre supérieur à l'arrêt d'embarquement pour le déplacement (sur une ligne – direction donnée qui contient des lignes – arrêts) étudié. Cet ensemble d'arrêts contiendra forcément l'arrêt de descente de l'utilisateur. Le concept des fuites de lignes s'est donc développé en bâtissant l'algorithme et occupe maintenant bien sa place dans la modélisation tenant compte de la destination. Deuxièmement, on observe aussi que les instances de l'objet des chaînes de déplacement n'étaient pas quantifiées auparavant, car, bien que le concept existait à la base de la modélisation, il

n'était pas dénombré, sûrement parce qu'il n'était pas encore très utilisé. Il est toutefois aussi un concept clé dans la modélisation tenant compte de la destination.

Il nous semblait très important de montrer tout de suite les changements entre les deux versions du modèle objet de la STO. Il y a deux principales raisons pour expliquer cette démarche :

1. Comme le projet repose sur la connaissance des destinations pour dériver les statistiques, il est essentiel de présenter le fondement et le fonctionnement de l'algorithme de destination.
2. L'ajout de ce nouveau « concept des destinations » fait réaliser qu'il a un impact (de changement) sur la modélisation de base du réseau de la STO et qu'il a forcément (évidemment) un impact sur la structure et le fonctionnement de la base de donnée de la STO. L'ajout d'un nouveau concept doit remettre en question son impact quant à la modélisation de base du réseau de transport étudié. Le fait d'avoir comparé ces deux diagrammes permet d'élargir les bornes de la méthodologie (diagramme présenté en section 3.2) et de la rendre plus générale, de façon à ce que le système développé puisse s'adapter aux changements.

La prochaine section décrit l'algorithme de dérivation des destinations.

3.1.2 Présentation et implantation de l'algorithme de dérivation des destinations

Dans cette section, il sera question de l'algorithme de dérivation des destinations développé par Trépanier et al. (2007). En premier lieu est présenté un survol des étapes préliminaires menant à l'algorithme, puis l'algorithme lui-même. Ensuite, il y a une explication de l'implantation de l'algorithme tel qu'il est installé dans la base de données SIVT avec ses paramètres.

3.1.2.1 Étapes préliminaires menant à l'algorithme

Un certain nombre d'étapes ont dû être réalisées avant de passer à la modélisation de l'algorithme comme tel. Un résumé des étapes importantes est présenté en annexe B.1. Voici, les grandes lignes de ce qui sera présenté dans cet annexe (Tranchant, 2005).

1. prétraitement des erreurs;
2. traiter de façon désagrégée les données de déplacement;
3. analyse comportementale de la clientèle « carte à puces »;
4. prétraitement des erreurs Approche de dérivation de la demande.

3.1.2.2 Algorithme de dérivation des destinations

Une fois les concepts nécessaires à la réalisation de l'algorithme bien démontrés, les chercheurs ont pu se pencher sur le vif du sujet. Avant de montrer le modèle, il faut préciser que les auteurs ont définis 3 types de destinations pouvant être dérivées :

1. Les destinations dérivées d'un déplacement qualifié de « Normal » c'est-à-dire qu'il n'est **pas le dernier** de la journée et que d'autre(s) transaction(s) de montée(s) sont enregistrée(s) ultérieurement dans la journée. Donc un déplacement parmi la chaîne de déplacements de l'utilisateur.
2. Les destinations dérivées d'un déplacement qualifié de « Dernier » c'est-à-dire le dernier déplacement d'une chaîne de déplacements effectués dans la journée. À ne pas confondre avec un déplacement unique.
3. Le dernier cas est qualifié de « Simple » c'est-à-dire qu'ils s'adressent aux déplacements simples qui requièrent une analyse des habitudes de déplacements (sur les jours de déplacements précédents) de l'utilisateur afin de déterminer la descente.

Voici les indices du modèle (Trépanier et al., 2007):

i = Usager du réseau (identifié à une carte unique)

Donc $J_{ik} = \{V_{irk}\}$, $r = 1, \dots, N_k$. Le nombre de déplacement faits par un usager dans la journée est nécessairement égal ou inférieur au nombre de fuites de lignes, car les voyages peuvent être composés d'un ou plusieurs déplacements et peuvent inclure des correspondances.

La distance devient maintenant un élément clé dans l'estimation de la descente, car il est très probable que l'usager va marcher entre ces différents déplacements et qu'il cherchera donc à minimiser la distance qui le sépare d'un autre déplacement sur le réseau. On établit alors la distance $d = (a, b)$, comme étant la distance euclidienne entre les locations a et b . On cherche alors à déterminer l'arrêt de descente d_{irk} de la fuite de ligne courant où la distance entre le prochain arrêt d'embarquement $S_{(r+1)ik}^B$ est minimum.

On obtient ainsi l'équation :

$$d_{irk} = z \rightarrow \min_z d(S_{(r+1)ik}^B, z) \quad z \in \{V_{rik}\}$$

$$r < N_k \quad d(S_{(r+1)ik}^B, z) < M \quad (\text{fonction objectif pour descente type 1})$$

On voit ici qu'on a inséré une constante M qui sert à déterminer une distance maximale de marche que l'usager pourrait faire afin de se rendre au prochain déplacement. Maintenant, pour le second type de descente à estimer qui concerne le dernier déplacement de la journée, soit ($r = N_k$) on prend soit l'embarquement d'origine de la journée si on est sur la même ligne :

$$d'_{irk} = z \rightarrow \min_z d(S_{rik}^B, z) \quad z \in \{V_{rik}\}$$

$$r = N_k \quad d(S_{rik}^B, z) < M \quad (\text{fonction objectif pour descente type 2 même ligne})$$

Ou bien on regarde le premier embarquement du lendemain si on n'est pas sur la même ligne (soit $d(S_{1ik}^B, z) > M$):

$$d''_{irk} = z \rightarrow \min_z d(S_{ri(k+1)}^B, z) \quad z \in \{V_{rik}\}$$

$$r = N_k \quad d(S_{ri(k+1)}^B, z) < M \quad (\text{fonction objectif pour descente type 2 autre ligne})$$

Finalement pour l'estimation des descentes de type 3, donc qui n'ont pas pu être déterminées par l'une des 3 fonctions objectives précédentes, il faut faire appel au concept de régularité en analysant des déplacements antérieurs de l'utilisateur qui se seraient produits dans la plage horaire la plus similaire à celle étudiée. Pour ce faire, on utilisera donc aussi l'ensemble des destinations trouvées dans le mois pour cet usager. On nommera cet ensemble de destinations D_i avec S_i l'ensemble des arrêts reliés à ces destinations. Avec s étant le point de départ de d , on obtient :

$$\forall d_{irk}, d'_{irk}, d''_{irk} \in D_i \Rightarrow \exists S_{irk}^B \in S_i \quad S_{irk}^B = \text{départ}(d_{irk})$$

Si d'''_{irk} représente le cas d'une destination tel que mentionné au précédent paragraphe, on cherche alors un voyage similaire antérieur (sur la même ligne évidemment) qui minimise la différence entre les temps de départ des deux déplacements similaires :

$$d'''_{irk} = z \quad z \in D_i \quad \rightarrow \min_y h(t(y), t(S_{irk}^B))$$

$$R(y) = R(S_{irk}^B) \quad y = \text{départ}(z)$$

Ici, $h(a,b)$ est la différence de temps entre les embarquements a et b et $t(x)$ est le temps d'arrivée du bus. Les $R(x)$ correspondent à l'arrêt x de la ligne R . La figure suivante explique la méthode mentionnée pour ce type de destinations comme au jour $n+1$ sur la figure, qui représente un déplacement simple.

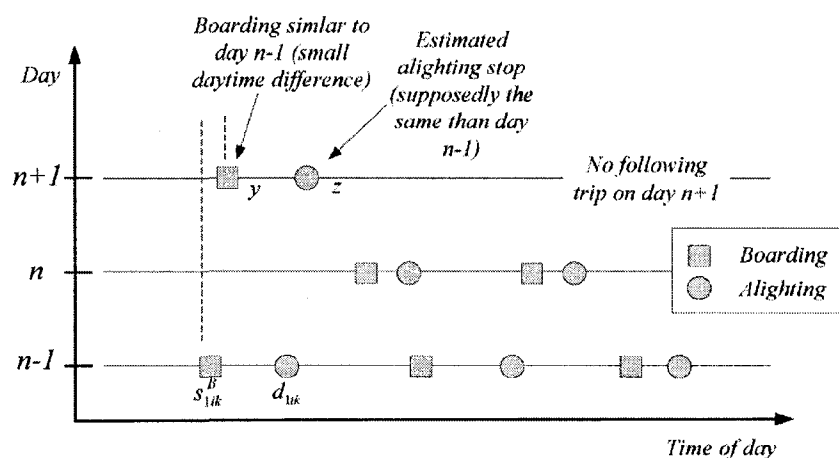


Figure 3-6 Recherche de déplacement pour trouver une destination d'un déplacement «Simple»

La figure montre que l'on s'est servi d'un déplacement similaire fait au jour $n-1$ afin de trouver la destination en $n + 1$.

Maintenant que nous avons expliqué en détail l'algorithme de dérivation des destinations, voyons à présent comment il a été implanté dans la base de données de la STO et où sont enregistrés les résultats.

3.1.2.3 Implantation de l'algorithme dans le système SIVT de la STO

Comme l'algorithme élaboré dans la section précédente nécessite le traitement d'une très grande quantité de données (surtout dans le cas de l'estimation des destinations de type 3 où l'on doit explorer les déplacements antérieurs des usagers), il faut que l'outil utilisé soit efficace. Il faut donc tenir compte des considérations suivantes afin de choisir l'outil de développement souhaité pour l'algorithme :

- toutes les données nécessaires se retrouvent dans les tables du système SIVT;
- le serveur SIVT ne doit pas être trop sollicité afin de ne pas nuire aux opérations quotidiennes;
- le résultat des destinations doit être aussi renvoyé dans le système SIVT, car il facilitera grandement la suite du projet et qu'il faut limiter au maximum les différentes sources de données;
- le système SIVT fonctionne sous Microsoft SQL Server 2000;
- nous ne pouvons pas développer d'application Access sur le serveur pour des raisons de sécurité.

Face à ces considérations, nous avons donc éliminé le choix d'utiliser un programme externe au serveur SQL, qui bien que potentiellement plus rapide à traiter l'algorithme comme tel, perdrait beaucoup en performance (et en temps d'utilisation du serveur) simplement à s'approvisionner en données, sans oublier la réimportation des résultats sur le serveur.

La solution était donc de travailler à même le serveur SQL afin d'avoir un accès rapide aux données et à la diffusion des résultats. L'algorithme a été codé en Transact SQL (PL/SQL), soit un langage qui permet le développement de fonctions et de procédures en plus de bénéficier de la facilité d'accès aux données par le biais des requêtes SQL. La procédure travaille donc chaque nuit afin de diminuer les requêtes au serveur.

Avant de présenter les paramètres utilisés pour la procédure, voici comment on propose de modifier la table de transaction du système SIVT (sivt_transactionbus) afin d'y inclure le résultat des différentes transactions.

Principalement, il y a : l'identificateur de la transaction (numéro), le numéro de carte (usager), le numéro du titre associé à la carte, la date de transaction, le type de transaction (validation, correspondance, refus), l'heure de montée, le numéro de ligne, le sens, l'heure de départ de la ligne, le numéro d'arrêt et plusieurs autres informations.

L'algorithme va ajouter quatre champs supplémentaires à cette table afin d'y apporter les informations sur la potentielle destination associée à chacune des transactions. Il a donc fallu utiliser une instruction `alter table` afin d'ajouter les quatre champs requis :

```
alter table dbo.sivt_transactionbus ADD
  {DER_typeErreur} [tinyint] NULL ,
  {DER_Num_Arret_Dest} [int] NULL ,
  {DER_Distance_Dest} [int] NULL ,
  {DER_Typed} [tinyint] NULL ;
```

Voici la description des champs ajoutés dans la table `sivt_transactionbus`, on y retrouve aussi le type de champ ainsi que sa description :

Tableau 3-1 Champs ajoutés dans la table `sivt_transactionbus`

<i>Champ</i>	<i>Type de champ</i>	<i>Description</i>
DER_typeErreur	Entier	Détermine le type d'erreur potentiel trouvé sur la transaction analysée. (1)
DER_Num_Arret_Dest	Entier	Contient le numéro d'arrêt de descente trouvé par l'algorithme.

Tableau 3-2 Champs ajoutés dans la table sivr_transactionbus (suite et fin)

DER_Distance_Dest	Entier	Contient la distance en mètres entre l'arrêt de descente estimé et l'arrêt de la transaction suivante, si il y a lieu.
DER_Typed	Entier	Détermine le type de descente estimé. (2)

(1) Les types d'erreurs détectés (à partir de ce qui a été trouvé dans l'étape préliminaire discuté en section 3.3.1) par la procédure sont les suivants :

- Type 1 – Lorsque la transaction s'est effectuée sur la ligne 0, 500 ou 900;
- Type 2 – Lorsque la transaction s'effectue sur une ligne – sens qui n'existe pas dans la table SSLI_ARRET, soit car elle n'existe pas ou bien qu'elle existe, mais dans une période de validité excluant la date de transaction;
- Type 3 – Lorsque la transaction s'effectue sur un arrêt qui n'existe pas dans la table ARRET, soit car il n'existe pas ou bien qu'il existe, mais dans une période de validité excluant la date de transaction;
- Type 4 – Lorsque la transaction s'effectue sur une ligne – sens – arrêt qui n'existe pas dans la table SSLI_ARRET, soit car cette combinaison n'existe pas ou bien qu'elle existe, mais dans une période de validité excluant la date de transaction;
- Type 5 – Lorsque la transaction s'est effectuée en bout de ligne;
- Type 6 – Lorsque l'heure de la transaction est '0000';
- Type 0 – Lorsqu'il n'y a aucune erreur.

(2) Les types de descente estimés sont sensiblement les mêmes que ceux énoncés en section 3.2.2 :

- Type 1 – Estimation d'une descente pour une distance faible (respect du critère de distance `critereDIST`) entre l'arrêt de la ligne précédente et de la montée suivante. Donc pour le cas qui n'est pas le dernier déplacement de la journée.

- Type 2 – Estimation d’une descente pour une distance faible (respect du critère de distance `critereDIST`) entre l’arrêt de la ligne active et l’arrêt de la première montée de la journée. Cas du dernier déplacement de la journée.
- Type 3 – Estimation d’une descente pour le cas d’une distance trop grande dans le cas 1 ou 2. On essaie donc de trouver un embarquement similaire à l’aide d’une destination estimée une autre journée. Les estimations de type 3 respectent la valeur du paramètre `critereHRE` qui détermine l’écart maximal acceptable entre l’heure de la transaction similaire utilisée et l’heure de la transaction actuelle utilisée.
- Type 4 – Estimation d’une descente pour le cas d’une distance trop grande dans le cas 1 ou 2. On essaie donc de trouver un embarquement similaire à l’aide à l’aide d’une destination estimée une autre journée. Les estimations de type 4 **ne respectent pas** la valeur du paramètre `critereHRE`.
- Type 99 – Les cas 99 sont les cas à la base qui ont échoué aux critères de distance des cas de type 1 et 2. On a ensuite tenté de leur appliquer un cas 3 ou 4, si rien n’a encore été trouvé, le type de destination reste toujours à 99. On n’a donc pas pu trouver de destination pour cette transaction.
- Type 0 – Le type 0 se produit lorsque la procédure de détection des erreurs a trouvé une erreur majeure concernant la transaction étudiée. On ne tente donc pas de trouver une destination pour une transaction erronée.

Maintenant que nous avons une idée de la signification de la valeur des champs qui seront dérivés et ajoutés dans les tables par l’algorithme, voici le détail des étapes de la procédure codée en PL/SQL :

NOTE : des exemples de code (étapes 1 à 3) détaillé de la procédure se retrouve en annexe B.2.

1. Création de vues, d’indexes et de fonctions de conversions d’heures nécessaires à la procédure;

2. Création des procédures d'ajout d'information qui serviront à mettre à jour la table `sivt_transactionBus` avec les valeurs trouvées par l'algorithme;
3. Lancement de la procédure de détection des erreurs `PROCEDURE Poly_determiner_type_erreur_I;`
4. Lancement de la procédure de détermination des destinations. Cette procédure utilise 2 paramètres qui pourraient être changés par les opérateurs du système de transport, car ils ont un effet sur la précision des résultats donnés par l'algorithme :
 - Le critère de distance, `critereDIST`, qui détermine la distance maximale à laquelle entre l'arrêt de descente estimé et celui d'une autre montée. Dans la procédure, il est **fixé à 2000 mètres**.
 - Le critère d'heure de transaction, `critereHRE`, qui détermine le delta temps en heure entre une heure de montée pour laquelle on veut déterminer une destination et une heure de transaction d'une montée similaire. Dans la procédure, ce delta temps est fixé à 3 heures. Les déplacements similaires doivent donc avoir été fait à travers une **fenêtre de temps de 3 heures** du déplacement actif.

Cette procédure est destinée à être exécutée dans la nuit afin de déterminer les destinations de la journée précédente. Pour terminer cette section, voici deux graphiques : un présentant des statistiques de bases sur les distances trouvées pour les types de destinations 1 et 2, et un présentant la répartition des types de destinations trouvées par l'algorithme pour le mois de janvier 2007 et ses 794879 transactions.

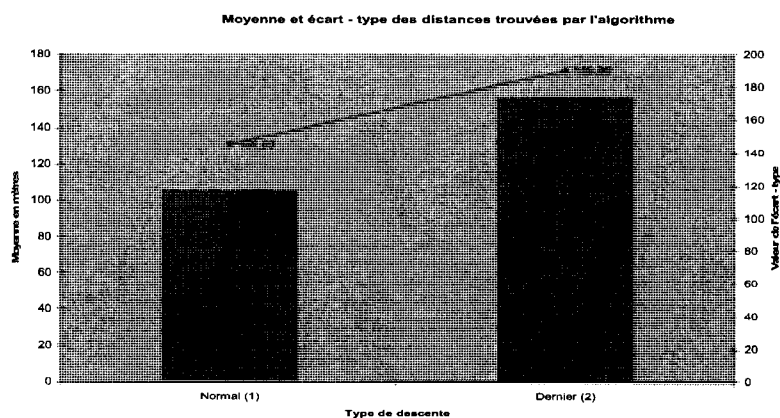


Figure 3-7 Statistiques de base sur les distances relatives aux 2 premiers types de descentes

Sur ce graphique, on remarque que la moyenne des distances et l'écart – type des distances trouvées entre le point de descente estimé et le premier point d'embarquement de la journée sont supérieures à ce qui est trouvé dans le cas d'un déplacement qui n'est pas le dernier de la journée. On peut penser que l'utilisateur cherchera toujours à minimiser les distances de marche dans le cas de correspondances, mais que s'il ne revient pas chez lui par la même ligne qu'il a emprunté le matin même, les distances de marche seront forcément plus grande, tout au moins en moyenne. Voyons maintenant la répartition des types de destinations trouvées.

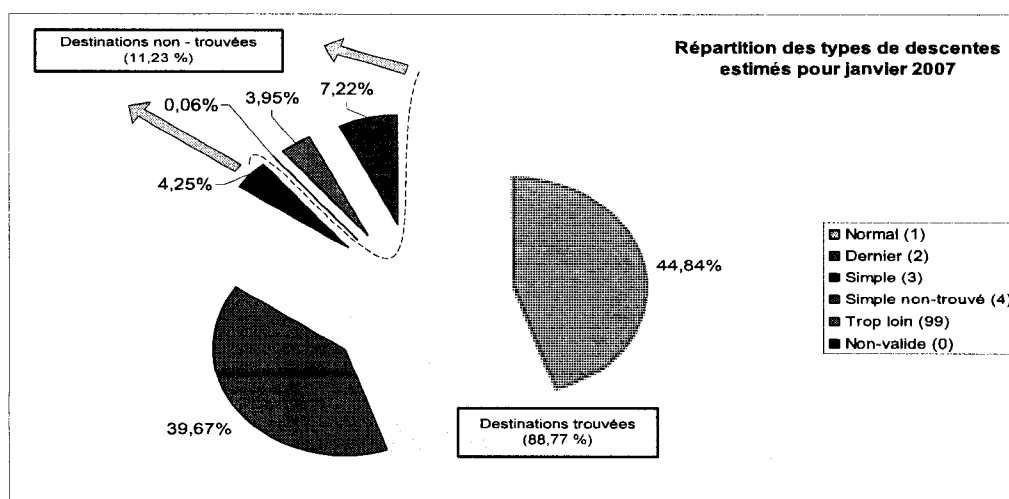


Figure 3-8 Répartitions des types de descentes estimées en janvier 2007

On remarque que l'algorithme, avec les critères mentionnés précédemment, a réussi à trouver 88,77 % des destinations (l'estimation des descentes les plus logiques pour le comportement normal d'un usager) pour l'ensemble des 794879 transactions effectuées en janvier 2007. La plupart des destinations non - trouvées (7,22 % des 11,23 % non - trouvées) proviennent de transactions jugées non valides par la première procédure de détection des erreurs. Bien qu'avant de tirer une conclusion véritable il faudrait poursuivre l'analyse sur plusieurs mois, on peut tout de même s'apercevoir que l'algorithme est capable d'estimer la très grande majorité des destinations sur le réseau de transport.

Maintenant que l'on sait d'où les destinations proviennent et comment elles sont ajoutées au système SIVT, poursuivons avec la méthodologie générale du développement de l'outil.

3.2 Présentation de l'approche méthodologique pour le développement d'un outil de diffusion de statistiques

La méthodologie présentée dans cette section retrace l'ensemble des étapes réalisées dans ce projet. Elle est en fait séparée en deux approches distinguées. La première est pour la méthodologie dite générale, qui est si l'on veut applicable au premier passage du développement de l'outil de diffusion de statistiques opérationnelles. Il s'agit de la seule méthodologie qui aurait été appliquée si le résultat final avait été parfait au premier coup. La seule itération possible dans l'approche générale provient de la validation du modèle objet suite à l'étape 2.3 (voir figure 3-1). En effet, comme mentionné précédemment dans ce chapitre, il faut vérifier si notre modèle objet supporte tous les concepts (relatifs aux statistiques, ou bien carrément un ajout d'un concept de base) nécessaire à la poursuite d'un développement juste. Nous avons vu que c'était le cas pour l'ajout du concept des destinations des usagers qui a entraîné un changement sur le modèle objet de la STO.

La seconde méthodologie présentée est une méthodologie post-développement. Il s'agit en effet d'une méthodologie de correction d'erreurs et d'ajout de contenu au site Web développé. La correction d'erreur peut porter sur des lacunes concernant l'utilisabilité⁴ du site ou bien sur des erreurs relatives aux données qui sont présentées aux utilisateurs. Pour ce qui est de l'ajout de contenu au site, il peut s'agir soit de l'ajout d'une nouvelle statistique opérationnelle (déclenchement de la procédure 8.1 (1), sur la figure 3-9) ou bien d'une nouvelle fonctionnalité (déclenchement de la procédure 8.1 (2), sur la figure 3-9). Cette seconde approche méthodologique est très importante puisqu'elle permettra de poursuivre le développement de l'outil ou bien de l'adapter à une nouvelle entreprise de transport suite à une analyse comparative de leurs modèles objets respectifs bien sûr. Voici donc la figure de cette seconde approche méthodologique :

⁴ L'**utilisabilité** ou **usabilité**¹ est définie par la norme ISO 9241 comme « le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié ». - <http://fr.wikipedia.org/wiki/Utilisabilit%C3%A9>

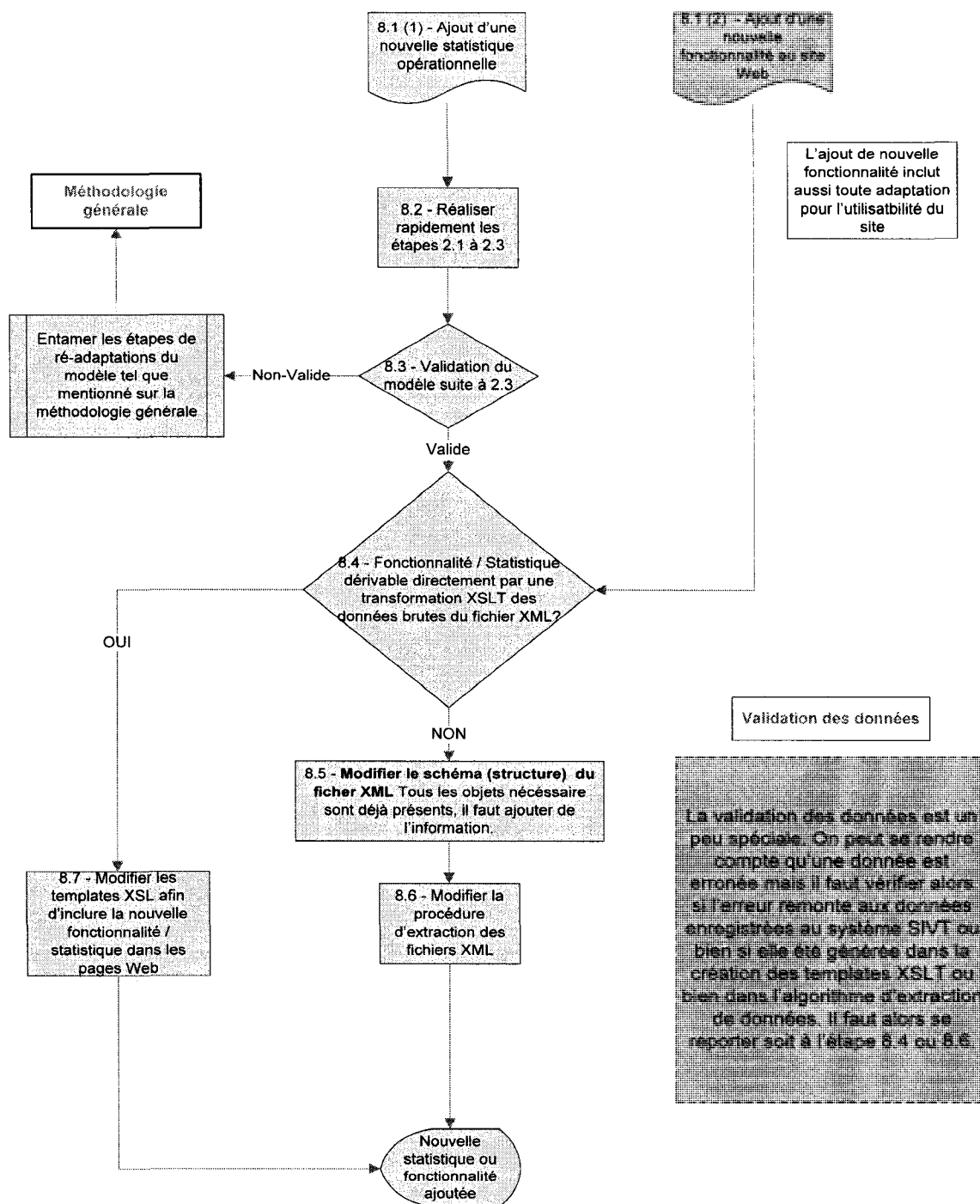


Figure 3-9 Méthodologie de correction et d'ajout de contenu à l'outil

Les prochaines sections présenteront le détail des différentes étapes de la méthodologie général, aussi, des exemples seront montrés, «en cours de route», pour illustrer quelques

cas qui ont nécessité l'utilisation de la méthode de correction et d'ajout de contenu à l'outil.

3.3 Étapes de modélisation primaire

Comme on peut le remarquer avec le titre de la section, nous allons commencer directement avec l'ensemble des procédures 2.1 à 2.3 (figure 3-1). En effet, la modélisation objet du réseau de transport de la STO a été couverte précédemment dans ce chapitre et on considère aussi qu'elle est adaptée au concept de la dérivation des destinations des usagers.

3.3.1 Modélisation des statistiques opérationnelles et leurs hypothèses

Le tableau suivant couvrira les étapes 2.1 et 2.2, soit la définition des statistiques opérationnelles que l'on souhaite diffuser sur le site Web, ainsi que les différentes hypothèses qui peuvent s'y attacher. Avant de présenter le tableau, voici un rappel de quelques notations et définitions qui pourront être utiles à la définition de statistiques opérationnelles :

Pour **une même journée**, sur **une ligne** d'autobus **X** dans **une direction Y**, on utilisera les notations suivantes :

i = Usager du réseau (identifié à une carte unique)

j = Ordre d'arrêt sur une ligne

l = Heure de départ du voyage

k = Type de titre de transport

t = Type de transaction

$R = \{S^j\}$ où chaque S^j représente un des **arrêts** d'une **ligne R** (dans une seule direction)

$J = \{V^l\}$ l'ensemble des voyages effectués. V étant un voyage en cours (du terminus de départ d'une ligne au terminus d'arrivée, dans un sens)

Comme nous allons vouloir fonctionner par agrégation pour cumuler les différentes statistiques, et par souci de ne pas répéter trop d'informations dans nos fichiers XML,

nous ne pourrions pas conserver les différentes heures associées à chacune des transactions, donc pour chaque arrêt, il ne sera conservé que l'heure de la première montée dans le cas où il y en a plusieurs. Il n'y a donc pas d'indice relatif à l'heure de transaction. Finalement il faut mentionner que les voyages sélectionnés ont déjà été validés, car ils proviennent d'une table appelée `sivt_Voyage_Reel` qui ne contient que des voyages réels effectués, car ils ont passé le test de validation créé par la STO. On ne fait donc que vérifier si la transaction a été effectuée dans le cadre d'un voyage réel.

Tableau 3-3 Modélisation des statistiques opérationnelles

<i>Statistique opérationnelle</i>	<i>Définition et formulation</i>	<i>Hypothèses et considérations</i>
<p><i>Pour chaque voyage</i> $V^l \in J, \forall l$</p> <p><u>Montants par arrêts :</u></p> <ul style="list-style-type: none"> • Global; • Par titre; • Par type de transaction. 	<p><i>Pour tous les arrêts de la ligne</i> $S^j \in R, \forall j \rightarrow$</p> <p>Un montant $M = 1$.</p> <ul style="list-style-type: none"> • Global: $\sum_i M_{ikt}$; • Par titre: $\sum M_{ikt} \forall k$; • Par type de transaction : $\sum M_{ikt} \forall t$; 	<p>La transaction se soit effectuée sur un arrêt appartenant à la ligne et la direction en cours et qu'elle soit valide pour la période incluant la date de transaction.</p> <p>-Pour les titres k, nous en avons comptabilisé 4 sortes :</p> <ol style="list-style-type: none"> 1. Adultes; 2. Étudiants; 3. Aînés; 4. Autres (l'ensemble des autres titres). <p>-Pour les types de transactions, nous en avons comptabilisé 2 :</p> <ol style="list-style-type: none"> 1. Validation (montée); 2. Correspondance.
<p><i>Pour chaque voyage</i> $V^l \in J, \forall l$</p> <p><u>Montants globaux par voyage :</u></p> <ul style="list-style-type: none"> • Par titre; • Par type de transaction. 	<p>Un montant $M = 1$.</p> <ul style="list-style-type: none"> • Global: $\sum_i M_{ikt}$; • Par titre: $\sum M_{ikt} \forall k$; • Par type de transaction : $\sum M_{ikt} \forall t$; 	<p>Idem. Il s'agit d'une statistique qui se rattache au voyage global et non à la transaction.</p>

Tableau 3-4 Modélisation des statistiques opérationnelles (suite)

<i>Statistique opérationnelle</i>	<i>Définition et formulation</i>	<i>Hypothèses et considérations</i>
<p>Pour chaque voyage</p> $V^l \in J, \forall l$ <p><u>Descendants par arrêts :</u></p> <ul style="list-style-type: none"> • Global; 	<p>Pour tous les arrêts de la ligne</p> $S^j \in R, \forall j \rightarrow$ <p>Un descendant $D = 1$.</p> <ul style="list-style-type: none"> • Global: $\sum_i D_i$; 	<p>La descente doit être dérivée d'une transaction se soit effectuée sur un arrêt appartenant à la ligne et la direction en cours et qu'elle soit valide pour la période incluant la date de transaction.</p> <p>Évidemment, on ne tient plus compte des titres et des types de transactions.</p> <p>La difficulté ici est de recréer, tout en gardant l'ordre des arrêts de la ligne, une transaction de descente simple dans le cas d'un arrêt où il n'y a pas de montée. En effet, le système SIVT n'est pas fait à la base pour enregistrer une descente simple. Chaque transaction de montée a tout de suite sa transaction de descente associée si il y a lieu. Il faudra créer ces «transactions» dans l'extracteur de données XML.</p>
<p>Pour chaque voyage</p> $V^l \in J, \forall l$ <p><u>La charge à bord à chaque arrêt:</u></p> C_j	<p>Pour tous les arrêts de la ligne</p> $S^j \in R$ <p>Un descendant $D = 1$.</p> <p>Un Montant $M = 1$.</p> <p>À $j = 1$: $C_j = \sum_i M_{ikt}$;</p> <p>$\forall j > 1$,</p> $C_j = C_{j-1} + \left(\sum_i M_{ikt} - \sum_i D_i \right);$	<p>Il faut répéter les mêmes hypothèses et considérations que pour les descentes. Encore une fois, il est très important, afin d'avoir des données exactes, de recréer des transactions de descentes uniques (arrêt sans montants) suivant le bon ordre des arrêts de la ligne.</p>

Tableau 3-5 Modélisation des statistiques opérationnelles (suite et fin)

<i>Statistique opérationnelle</i>	<i>Définition et formulation</i>	<i>Hypothèses et considérations</i>
<p>Pour chaque voyage $V^l \in J, \forall l$</p> <p><u>L'ajustement de charge en bout de ligne:</u> A</p>	<p>L'ajustement pour que la charge finale soit de 0 au dernier arrêt de la ligne.</p> <p>$S^j \in R$</p> <p>$\Delta j = \max(j) :$</p> <p>$A = \text{absolu}(0 - C_j)$</p>	<p>Comme l'algorithme ne peut dériver la globalité des destinations effectuées sur le réseau, on peut calculer l'ajustement de la charge à bord en comptant l'ajustement requis jusqu'à une charge nulle.</p>
<p>Pour chaque voyage $V^l \in J, \forall l$</p> <p><u>Le nombre de passager - kilomètres:</u> PKM</p>	<p>La somme sur tous les arrêts de la charge ponctuelle à l'arrêt j multiplié par la distance entre l'arrêt $(j-1)$ et j.</p> <p>Soit n l'ordre final de la ligne. À partir de l'arrêt d'ordre $j > 1$:</p> <p>$S^j \in R$</p> <p>$\forall j > 1, \dots, n$</p> <p>$d(a,b)$ est en kilomètres.</p> <p>À $j = 1$: $PKM_{S^{(j)}} = 0$;</p> <p>Donc pour le voyage global :</p> <p>$PKM_{V^l} =$</p> <p>$PKM_{S^{(j-1)}} + C_j \cdot d(S^{(j-1)}, S^j)$</p>	<p>Cette statistique requiert la connaissance des coordonnées (X, Y) GPS des arrêts de la ligne afin de calculer la distance séparant chacun des arrêts. Il faudra donc stocker ces coordonnées. Ici la distance $d(a,b)$ en kilomètres entre chaque arrêt sera calculé en utilisant la formule de distance euclidienne entre deux points.</p>
<p>Pour chaque voyage $V^l \in J, \forall l$</p> <p><u>Le facteur d'ajustement global des passagers :</u> $PPERI$ vers l'extérieur $PCENT$ vers le centre-ville</p>	<p>Il s'agit de facteurs d'ajustements pour tenir compte de l'ensemble des passagers, donc inclure ceux qui n'utilisent pas de cartes à puces.</p>	<p>Pour calculer ces facteurs il faut utiliser les données de compte à bord. Plus de détail sur le sujet seront donnés dans la section 3.4.1 et dans le chapitre 4.</p>

Le tableau 3-2, qui présente l'ensemble des formulations et des hypothèses relatives aux statistiques opérationnelles de base n'inclus pas nécessairement toutes les variantes possibles de ces statistiques que l'on veut souligner. On peut penser par exemple à la charge maximale d'un voyage, ou bien à l'ensemble des montants du type étudiant pour toutes les lignes pour une journée précise. Il faut se rappeler que ces statistiques ont été formulées pour un cadre temporel et géométrique stricte : pour **une même journée**, sur **une ligne** d'autobus **X** dans **une direction Y**, mais on peut aussi jouer avec ces bornes afin d'établir les statistiques voulues, tant que les considérations de base ne sont pas changées. L'important, donc, est d'avoir bien formulé la base et d'avoir identifié les hypothèses qui devront être prises en compte afin de mettre en place le calcul de ces statistiques à partir des données qui seront extraites quotidiennement du serveur SIVT. Voyons maintenant les objets et les tables qui seront reliées aux calculs de ces statistiques.

3.3.2 Identification des objets et des tables reliées aux statistiques à calculer

Afin de couvrir la dernière étape de la modélisation primaire, soit la détermination des objets et des tables associées nécessaires aux calculs des statistiques identifiées précédemment, il faut bien sûr débiter avec le modèle objet. Comme la très grande majorité des statistiques reposent sur l'objet «transactions» du modèle objet, il faut donc considérer l'ensemble des objets (et leurs relations) qui nous permettent d'avoir une transaction sur le réseau, mais toujours en respectant notre contrainte de base qui est : pour **une même journée**, sur **une ligne** d'autobus **X** dans **une direction Y**. En même temps il nous sera possible d'identifier les tables associées aux différents objets. Pour ce qui est des hypothèses relatives aux statistiques, il faut s'assurer, entre autre, que les tables contiennent les informations nous permettant de valider la géométrie des objets réseaux (ligne – direction – arrêts) utilisés à la date de transaction choisie. D'autres considérations plus compliquées à mettre en place, comme la réinsertion de transactions de descentes uniques, seront prises en charge par le programme d'extraction de données. La figure 3-10 sur la page suivante permet d'illustrer le tout.

Pour ce qui est des statistiques opérationnelles reliées aux facteurs d'ajustements des passagers globaux, PPERI et PCENT, leur calculs requièrent l'accès aux données d'une autre base de données appelé STO – COMPTES qui contient les données de différents compte à bords. Ces tables ne sont pas présentent sur la figure 3-10. De plus les données de cette base sont très sommaires et elles ne contiennent aucune date (ni numéro de voyage) nous permettant de faire le lien avec un voyage précis afin de calculer le facteur voulu. Nous avons donc émis beaucoup de réserve quant à la validité de ces facteurs que nous avons estimés. Néanmoins nous avons jugé pertinents de montrer qu'ils existent, car il ne faut pas oublier que certains usagers n'utilisent pas la carte à puce (donc pas d'enregistrements de transactions dans le système SIVT) et qu'il faut répertorier cette proportion manquante à l'aide de ces facteurs. Les détails concernant l'estimation de ce calcul seront présentés au chapitre 4.

Pour une même journée, sur une ligne d'autobus X dans une direction Y

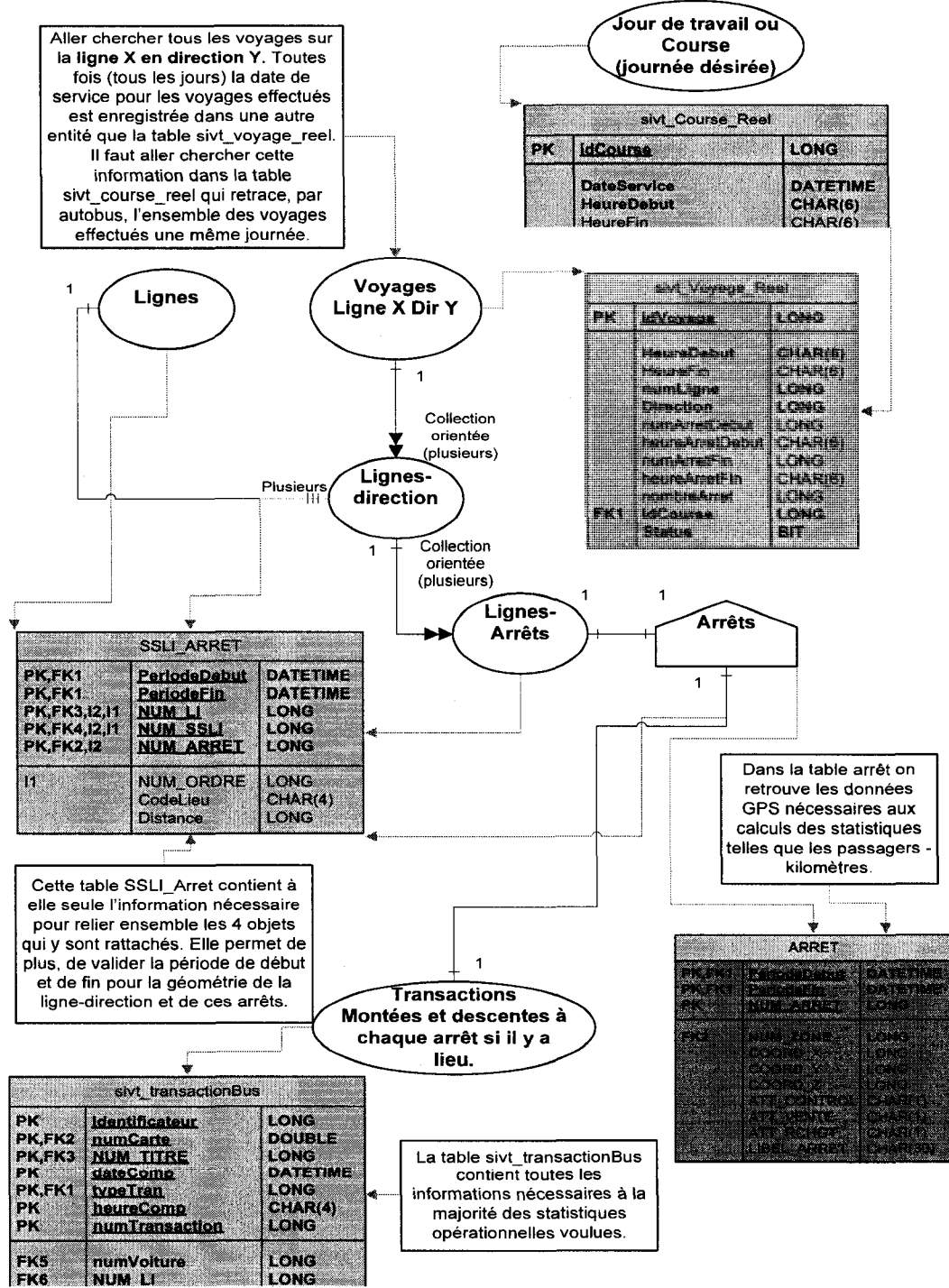


Figure 3-10 Identification des objets et de leurs tables reliées dans le système SIVT

Sur la figure 3-10, l'on remarque que les objets relatifs à la dérivation des destinations sont absents. C'est normal puisque, comme on l'a mentionné précédemment, on considère maintenant que les destinations font partie de la réalité du système et que l'information est directement accessible à partir de la table `sivt_transactionBus`. On n'a donc pas besoin de montrer les objets qui servent à obtenir ces destinations. Aussi, sur cette figure, on peut voir que nous avons indiqué les types de liens entre chaque objets (1 à 1, 1 à plusieurs, 1 à collection orientée, etc.). Ceci nous sera utile afin de déterminer la structure hiérarchique parents – enfants du fichier XML qui sera extrait chaque jour pour chaque ligne. Justement, avant de passer à la prochaine étape qui est de déterminer la structure du document XML, on remarque sur la figure 3-1 qu'il faut valider l'ensemble des étapes de modélisation primaire. Il n'y a pas vraiment de procédure détaillée pour cette procédure, il faut simplement valider si le modèle objet mis en place, et les données disponibles, supportent l'ensemble des statistiques opérationnelles désirées. Nous concluons que le modèle actuellement en place (figure 3-3) est valide et nous permet de calculer toutes les statistiques formulées dans le tableau 3-2.

3.4 Détermination de la structure du document XML

Nous avons vu au chapitre deux, les rudiments du fonctionnement du XML et des technologies qui y sont rattachées. Le XML a été choisi comme technologie pour ce projet pour les raisons suivantes :

- il permet le développement d'un outil destiné à être diffusé sur le Web;
- il ne requiert pas de pré requis logiciels ou de système d'exploitation;
- il permet de structurer les données de façon claire, logique et lisible par l'humain;
- il sert de source de donnée unique (et indépendante du serveur de données SIVT) à plusieurs *templates* XSLT qui serviront à offrir plusieurs statistiques aux utilisateurs;

- il permet la génération de graphiques SVG interactifs.

Bien que ce ne soit pas le serveur de données SIVT qui travaille lorsque l'utilisateur demande l'affichage d'une page de statistique basé sur les documents XML, il faut toutefois prendre en considération que nous avons un impact, lors de la création de la structure, sur la taille (en bytes) des documents XML générés. Nous pouvons, en effet, en plus d'avoir une structure adéquate, jouer sur la taille des mots qui décriront les différents éléments et attributs du fichier afin de minimiser la taille de ceux-ci. Le fait d'avoir des fichiers ayant un petit nombre de kilooctets (ko) aidera à minimiser l'espace disque requis sur le serveur Web, en plus de minimiser le temps de chargement des pages du côté de l'utilisateur. Voyons maintenant la structure du fichier XML.

3.4.1 Structure générale des fichiers XML

La première étape consiste à se faire une idée générale de la structure hiérarchique des différents éléments du fichier. Ceci s'effectue facilement à partir la figure 3-11 qui montre les différents objets et les types de relations qui les unissent. Toutefois, il ne faut pas oublier les bornes que nous nous étions fixés, soit: pour **une même journée**, sur **une ligne** d'autobus **X** dans **une direction Y**. Nous avons décidé, tout d'abord, que la valeur de la date serait enregistrée en dehors de du fichier XML. En effet, nous utiliserons plutôt le nom du dossier, contenant tous les fichiers XML extraits, pour y stocker la date. De plus, nous avons décidé de créer qu'un seul fichier par ligne, afin de créer évidemment moins de fichiers et moins de confusion. On retrouvera donc dans un même fichier, le contenu des opérations effectuées dans les 2 directions :

1. **Une ligne** d'autobus **X** dans **une direction 0**;
2. **Une ligne** d'autobus **X** dans **une direction 1**.

Pour ce faire, il faut donc créer 2 éléments lignes et attribuer à chacun un attribut de direction. Ensuite, la structure des éléments enfants de ces 2 éléments lignes sera identique et seul le contenu changera. La figure suivante montre la hiérarchie des éléments qui se trouveront dans chacun des éléments ligne :

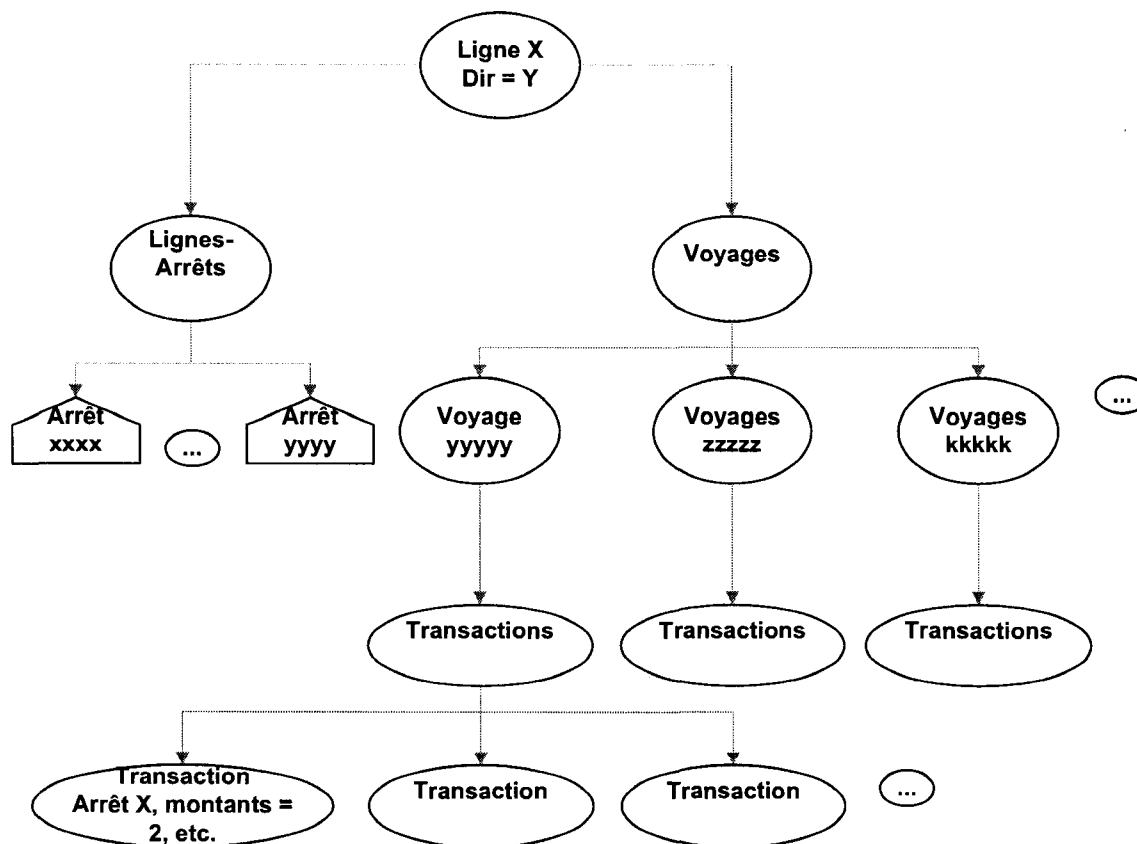


Figure 3-11 Structure hiérarchique générale des éléments pour une ligne X en direction Y

On remarque sur la figure 3-11 qu'on a choisi de créer un élément Lignes – Arrêts, contenant comme enfants les informations de tous les arrêts de la ligne – direction. On aurait pu ne pas mettre cet élément, et, à la place, pour chaque voyage, créer un élément transaction pour chaque arrêt, même si il n'y avait pas eu de montant ou de descendant à celui-ci. Cette dernière option aurait cependant entraîné une énorme répétition d'informations et aurait grossit inutilement la taille de chaque fichier.

Une fois la structure des éléments mis en place, il suffit de rajouter des attributs à tous ses éléments afin de bien distinguer chacun d'eux. Il faut simplement s'assurer que l'ensemble des attributs nous permettent de dériver toutes les informations nécessaires aux calculs des statistiques sans tomber dans le superflu. À cette étape, il n'est pas non plus essentiel d'intégrer des résultats de calculs aux valeurs des attributs, puisque les *templates* XSLT qui serviront à l'affichage de final des statistiques pourront effectuer

eux aussi des calculs dans leurs transformations. Voici donc ce à quoi ressemble un fichier XML typique extrait de la base de données. L'exemple est pour la ligne 42 et est séparé en 2 figures :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<root>
- <LI NOLI="42" DIR="0">
+ <VOYS>
+ <ARRL>
</LI>
- <LI NOLI="42" DIR="1">
+ <VOYS>
- <VOY IDV="87286" HD="0443" HF="0459" IDC="1013838">
- <TRS>
<TRSA NOA="1098" HT="453" NBM="1" VAL="1" COR="0" T1="1" T2="0" T3="0" />
</TRSA>
<META AC="1" GV="1" GC="0" GED="0" GAD="1" GAN="0" AE="1" PPERI="1" PCENT="1" />
</VOY>
- <VOY IDV="86879" HD="0555" HF="0652" IDC="1013763">
- <TRS>
<TRSA NOA="5010" HT="603" NBM="1" VAL="1" COR="0" T1="0" T2="0" T3="0" />
<TRSA NOA="5014" HT="604" NBM="1" VAL="1" COR="0" T1="1" T2="0" T3="0" />
<TRSA NOA="2610" HT="612" NBM="1" VAL="0" COR="1" T1="0" T2="1" T3="0" />
<TRSA NOA="2600" HT="614" NBM="1" VAL="0" COR="1" T1="1" T2="0" T3="0" />
<TRSA NOA="2065" HT="619" NBM="1" VAL="0" COR="1" T1="1" T2="0" T3="0" />
<TRSA NOA="2058" HT="619" NBD="1" />
<TRSA NOA="2028" HT="620" NBM="1" VAL="1" COR="0" T1="1" T2="0" T3="0" />
<TRSA NOA="1311" HT="603" NBD="1" />
<TRSA NOA="1254" HT="604" NBD="1" />
<TRSA NOA="1247" HT="639" NBM="1" VAL="1" COR="0" T1="1" T2="0" T3="0" />
<TRSA NOA="1074" HT="614" NBD="3" />
<TRSA NOA="1216" HT="612" NBD="1" />
</TRSA>
<META AC="0" GV="4" GC="3" GED="1" GAD="5" GAN="0" AE="5" PPERI="1" PCENT="1" />
</VOY>
<VOY IDV="181355" HD="0626" HF="0730" IDC="1030880">
<TRS />
<META AC="0" GV="0" GC="0" GED="0" GAD="0" GAN="0" AE="0" PPERI="1" PCENT="1" />
</VOY>
- <VOY IDV="86564" HD="0651" HF="0807" IDC="1013703">
```

Figure 3-12 Exemple de fichier XML pour la ligne 42 (portions transactions)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<root>
- <LI NOLI="42" DIR="0">
+ <VOYS>
+ <ARRL>
</LI>
- <LI NOLI="42" DIR="1">
+ <VOYS>
- <ARRL>
<ARR NOA="5064" ODR="0" X="18446198" Y="5031014" NOM="YORK ST/KING EDWARD AV" />
<ARR NOA="5008" ODR="1" X="18446245" Y="5030962" NOM="KING EDWARD AV/GEORGE ST" />
<ARR NOA="5010" ODR="2" X="18446312" Y="5030849" NOM="KING EDWARD AV/RIDEAU ST" />
<ARR NOA="5014" ODR="3" X="18446136" Y="5030714" NOM="RIDEAU ST/WALLER ST" />
<ARR NOA="5022" ODR="4" X="18445738" Y="5030472" NOM="RIDEAU ST/SUSSEX DR" />
<ARR NOA="5030" ODR="5" X="18445450" Y="5030275" NOM="WELLINGTON ST/ELGIN ST" />
<ARR NOA="5034" ODR="6" X="18445266" Y="5030161" NOM="WELLINGTON ST/METCALFE ST" />
<ARR NOA="5042" ODR="7" X="18445007" Y="5030010" NOM="WELLINGTON ST/BANK ST" />
<ARR NOA="5050" ODR="8" X="18444706" Y="5029829" NOM="WELLINGTON ST/N LYON ST" />
<ARR NOA="6040" ODR="9" X="18444207" Y="5030067" NOM="N PORTAGE BRIDGE/PORTAGE N" />
<ARR NOA="2612" ODR="10" X="18444189" Y="5030693" NOM="DE L* HOTEL DE VILL/DU PORTAGE" />
<ARR NOA="2610" ODR="11" X="18444113" Y="5030652" NOM="DU PORTAGE/DE L* HOT" />
<ARR NOA="2608" ODR="12" X="18443915" Y="5030476" NOM="DU PORTAGE/LAVEL" />
<ARR NOA="2600" ODR="13" X="18443614" Y="5030418" NOM="DU PORTAGE/BRETELLE 3" />
<ARR NOA="2554" ODR="14" X="18443385" Y="5030358" NOM="ALEXANDRE-TACHE" />
<ARR NOA="2550" ODR="15" X="18443230" Y="5030338" NOM="ALEXANDRE-TACHE/HANSON" />
<ARR NOA="2548" ODR="16" X="18443018" Y="5030311" NOM="ALEXANDRE-TACHE/MILLAR" />
<ARR NOA="2544" ODR="17" X="18442749" Y="5030275" NOM="ALEXANDRE-TACHE" />
<ARR NOA="2065" ODR="18" X="18442578" Y="5030250" NOM="ALEXANDRE-TACHE/ST-JOSEPH" />
<ARR NOA="2062" ODR="19" X="18442337" Y="5030215" NOM="ALEXANDRE-TACHE/VIGER" />
<ARR NOA="2058" ODR="20" X="18442174" Y="5030192" NOM="ALEXANDRE-TACHE/DE LA COOPER" />
<ARR NOA="2046" ODR="21" X="18441872" Y="5030134" NOM="ALEXANDRE-TACHE/BRETELLE 108" />
<ARR NOA="2028" ODR="22" X="18441630" Y="5030065" NOM="ALEXANDRE-TACHE/BRETELLE 107" />
<ARR NOA="2022" ODR="23" X="18441433" Y="5030013" NOM="ALEXANDRE-TACHE/BOUDRIA" />
<ARR NOA="2023" ODR="24" X="18441223" Y="5029956" NOM="ALEXANDRE-TACHE/ST-THERESE" />
<ARR NOA="2004" ODR="25" X="18441023" Y="5029901" NOM="ALEXANDRE-TACHE/ST-DOMINIQUE" />
<ARR NOA="2000" ODR="26" X="18440619" Y="5029715" NOM="ALEXANDRE-TACHE/ST-RAYMOND" />
<ARR NOA="6011" ODR="27" X="18440405" Y="5029547" NOM="CHEMIN D*AVLMER/ALEXANDRE-TACHE" />
<ARR NOA="1382" ODR="28" X="18440254" Y="5029445" NOM="CHEMIN D*AVLMER/BRETELLE 1" />
<ARR NOA="1381" ODR="29" X="18440125" Y="5029355" NOM="CHEMIN D*AVLMER" />
```

Figure 3-13 Exemple de fichier XML pour la ligne 42 (portions info arrêts)

On remarque premièrement sur la figure 3-12, la présence des deux éléments lignes pour les 2 directions (les deux éléments enfants principaux de la direction 0 ont été cachés). On voit ensuite les détails des transactions de 3 voyages (IDV 87286, 86879 et 181355) ayant respectivement 1, 12 et 0 transactions (montées et descentes incluses).

Pour ce qui est de la figure 3-13, on remarque que les éléments enfants des voyages ont été cachés afin de montrer les éléments arrêts et leurs attributs sous l'élément parent ARRL (arrêts – ligne). Le tableau suivant montre la liste des attributs et leur description pour chacun des éléments afin de permettre au lecteur de mieux saisir les informations du fichier. De plus, sont fourni en annexe B.3 les schémas XSD et DTD qui permettent de valider les documents XML ainsi créés pour le site Web.

Tableau 3-6 Liste et description des attributs d'un fichier XML type

<u>Éléments</u>	<u>Attributs et description</u>
<u>Ligne</u> : LI	<ul style="list-style-type: none"> • NOLI : Numéro de ligne • DIR : direction (0 ou 1)
<u>Voyage</u> : VOY	<ul style="list-style-type: none"> • IDV : # Identification du voyage • HD : Heure de départ du voyage • HF : Heure de fin • IDC : # Identification de la course associée
<u>Transaction</u> : TRSA	<ul style="list-style-type: none"> • NOA : # de l'arrêt (identification) • HT : Heure de la première transaction à l'arrêt • NBM : Nombre de montants total • NBD : Nombre de descendants total • VAL : Nombre de validations total • COR : Nombre de correspondances total • T1 : Nombre d'adultes total • T2 : Nombre d'étudiants total • T3 : Nombre d'aînés total

Tableau 3-7 Liste et description des attributs d'un fichier XML type (suite et fin)

<u>Arrêt</u> : ARR	<ul style="list-style-type: none"> • NOA : # de l'arrêt (identification) • ODR : # d'ordre de l'arrêt • X : Coordonnée GPS en X de l'arrêt • Y : Coordonnée GPS en Y de l'arrêt • NOM : Libellé de l'arrêt
---------------------------	---

Le lecteur a peut-être remarqué que nous n'avons pas donné d'information sur l'élément «**META**» associé à chaque voyage. Il s'agit en fait d'un élément particulier contenant plusieurs attributs reliés à de l'information qui a dû être rajoutée suite à l'application de la procédure 8.1 (1) ou 8.1 (2) présenté à la figure 3-9. En effet, au tout début du projet, cet élément n'existait pas et il s'est peu à peu rempli d'attributs pour pallier certaines difficultés techniques du site Web ou bien pour permettre l'ajout de nouvelles statistiques, comme PPERI et PCENT, qui n'étaient pas prévues au départ. Évidemment, certaines de ces informations sont reliées à des aspects techniques du site Web qui seront pour la plus part développés dans le prochain chapitre, mais on peut tout de même les présenter ici. Le tableau suivant décrit donc les attributs contenus dans l'élément «**META**» et la raison de leur ajout.

Tableau 3-8 Description des attributs contenus dans l'élément META

<u>Attributs</u>	<u>Description et raison de l'ajout</u>
AC	<ul style="list-style-type: none"> • <u>Description</u> : Valeur de l'ajustement de charge en bout de ligne. • <u>Raison</u> : Plus facile de le calculer à priori dans le document XML que de le dériver par la suite en programmation XSLT. Permet aussi de voir rapidement pour l'ensemble des voyages si la majorité des descentes on été trouvées ou bien de détecter des erreurs possibles sur l'agrégation des montants et descendants (ajustements négatifs, etc.)
GV	<ul style="list-style-type: none"> • <u>Description</u> : Valeur de la totalité des transactions de type «validation» pour le voyage. • <u>Raison</u> : Plus facile de le calculer à priori dans le document XML que de le dériver par la suite en programmation XSLT.

Tableau 3-9 Description des attributs contenus dans l'élément META (suite et fin)

GC	<ul style="list-style-type: none"> • <u>Description</u> : Valeur le la totalité des transactions de type «correspondance» pour le voyage. • <u>Raison</u> : Idem a GV.
GED	<ul style="list-style-type: none"> • <u>Description</u> : Valeur le la totalité des transactions de titre «étudiants» pour le voyage. • <u>Raison</u> : Idem a GV.
GAD	<ul style="list-style-type: none"> • <u>Description</u> : Valeur le la totalité des transactions de titre «adultes» pour le voyage. • <u>Raison</u> : Idem a GV.
GAN	<ul style="list-style-type: none"> • <u>Description</u> : Valeur le la totalité des transactions de titre «aînés» pour le voyage. • <u>Raison</u> : Idem a GV.
AE	<ul style="list-style-type: none"> • <u>Description</u> : Valeur de la charge maximale atteinte à bord de l'autobus pour le voyage effectué. • <u>Raison</u> : Afin de créer une échelle relative (Échelle qui s'ajuste en fonction du maximum à afficher tout en gardant le même espace disponible pour l'affichage. Donc la valeur de chaque élément à afficher est multipliée par un facteur d'ajustement qui est calculé à chaque voyage en fonction de la valeur de AE.) pour le graphique de charge à bord en SVG il fallait connaître pour chaque voyage, la valeur de la charge maximum.
PPERI	<ul style="list-style-type: none"> • <u>Description</u> : Valeur de l'ajustement global du nombre de passager en direction PÉRIPHÉRIQUE pour tenir compte des utilisateurs qui n'ont pas la carte à puce. • <u>Raison</u> : Facteur qui pourra être utile aux opérateurs, présentement il est estimé par voyage.
PCENT	<ul style="list-style-type: none"> • <u>Description</u> : Valeur de l'ajustement global du nombre de passager en direction CENTRE pour tenir compte des utilisateurs qui n'ont pas la carte à puce. • <u>Raison</u> : Facteur qui pourra être utile aux opérateurs, présentement il est estimé par voyage.

Pour ce qui est de PPERI et PCENT, la valeur des 2 facteurs a été calculée dans les 2 directions puisqu'au moment du calcul, on ne connaissait pas encore la signification de ceux-ci dans la base de données de comptage. En effet, nous rappelons, qu'avec la base de donnée sur les comptes fournie par la STO, il a été impossible d'établir précisément

ces ratios car nous n'avons pas obtenu la date précise de ces opérations de comptage, seulement la ligne, direction et heure de passage. Les ratios sont donc calculés sur une moyenne de voyages et non pas sur un voyage précis. Nous estimons qu'à l'heure actuelle, ce ratio est quasi inutilisable. Nos instruments prévoient cependant la présence d'un tel ratio par voyage.

Cette étape de la méthodologie met fin à ce chapitre, puisque les étapes suivantes sont un peu plus techniques et qu'elles ne requièrent plus de modélisation à partir du modèle objet de la STO qui a été présenté en début de chapitre. Il sera donc question du programme d'extraction des données et de la structure et du fonctionnement des différentes pages du site Web.

CHAPITRE 4: MISE EN ŒUVRE DE L'OUTIL DE VISUALISATION

Dans ce chapitre, nous allons présenter la structure globale de l'outil de visualisation développé. Ensuite, nous allons voir le fonctionnement de l'outil, soit de l'algorithme d'extraction des données au format XML jusqu'à la page Web interactive en format SVG qui présente les statistiques spécifiques par voyages effectués. L'utilisation entière du site, destiné aux planificateurs, sera présentée. Une section portant sur des analyses statistiques sommaires réalisées à l'aide de l'outil sera présentée en fin de chapitre.

4.1 Architecture globale de l'outil de visualisation

Avant de présenter le fonctionnement du site Web qui diffuse les statistiques qui ont été modélisées au chapitre précédent, il est important de présenter l'architecture générale qui mène à cet outil de visualisation. Ceci permet de voir les différentes entités impliquées dans le processus global, de voir les relations qui les unissent ainsi que les technologies sur lesquelles elles sont basées. La figure suivante démontre donc bien cette architecture qui part des données contenues sur le serveur SIVT jusqu'à l'affichage d'une page Web de statistiques sur le poste de travail d'un utilisateur.

On peut donner un peu plus de détails sur le «processeur nocturne» que l'on voit sur la figure. Il y a en effet deux étapes qui devront être réalisées chaque nuit afin de traiter les données de la dernière journée qui ont été téléchargées dans le serveur SIVT. Il faut tout d'abord dériver les destinations de la journée en question. Ceci est fait à l'aide de l'algorithme de dérivation des destinations qui a été expliqué au chapitre trois. Cette procédure est effectuée directement sur le serveur SIVT (qui utilise SQL Server 2000) à l'aide du script PL SQL (Transact SQL). Le calcul de ces destinations prend environ une quinzaine de minutes. Ensuite, une fois les destinations dérivées, on peut lancer la procédure d'extraction, programmée en VB.Net qui va créer les fichiers XML nécessaires au fonctionnement du site Web. Ces fichiers sont enregistrés sur le serveur Web IIS (*Internet Information Services*), qui contient aussi les autres fichiers XSL et

ASP, aussi nécessaires à l'outil de visualisation des statistiques. La procédure d'extraction prend en moyenne de 30 à 40 minutes pour se compléter (voir l'exemple de code en annexe).

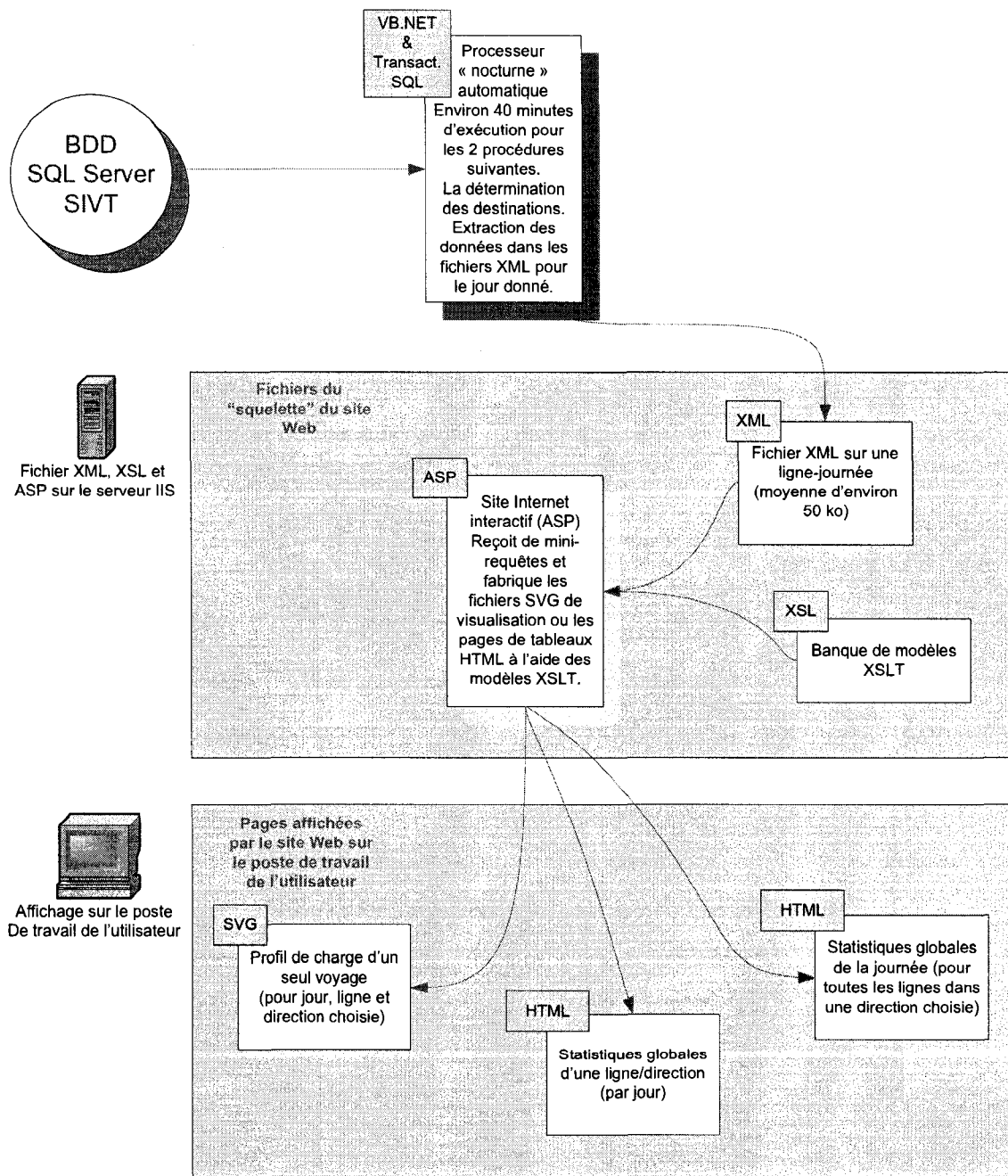


Figure 4-1 Architecture globale de l'outil de visualisation des statistiques

Évidemment, une fois les données d'une journée traitées, extraites et enregistrées sur le serveur Web, elles seront accessibles en tout temps à un utilisateur qui désire en obtenir des statistiques ou effectuer des comparaisons avec d'autres journées. Voyons maintenant le fonctionnement de la procédure d'extraction des données en différents fichiers XML.

4.2 Procédure d'extraction des données en fichiers XML

Le programme d'extraction des données au format XML a été programmé en VB.Net, car le langage offre une plus grande puissance ainsi qu'une plus grande flexibilité pour le traitement des données que le PL/SQL. En effet, afin de respecter les hypothèses énoncées dans le tableau 3.2 et de calculer les paramètres de l'élément META indiqués dans le tableau 3.4, il faut effectuer une série de traitements compliqués sur les données.

Le choix du VB.Net, est aussi dû à une question de sécurité. Au début du projet, il était question d'extraire les données du jour dans une base de données Access et de les traiter à l'aide du VBA (*Visual Basic for Applications*), mais la STO nous a indiqué son aversion envers le fait d'avoir un fichier Access sur leur serveur. En effet, ce type de fichiers pourrait être facilement modifié par une personne mal intentionnée et représenterait ainsi une menace à la sécurité de leur système. Un programme créé en VB.Net, peut être compilé en un exécutable, rendant ainsi sa modification très difficile.

Comme mentionné dans la section précédente, ce programme doit être exécuté toutes les nuits suite à la procédure de dérivation des destinations, car les fichiers XML extraits utilisent les destinations calculées. Une fois l'extraction terminée, chaque fichier XML est enregistré dans le répertoire «data» du serveur IIS qui régit le site Web.

La figure suivante explique le fonctionnement sommaire de la procédure d'extraction, soit à partir de l'extraction des données du jour sélectionné vers le fichier «contenant» temporaire qui seront traités par le programme VB.Net, jusqu'à l'enregistrement des fichiers XML créés. Des exemples de code de la procédure détaillé seront présentés en annexe C.2 et le schéma du paquet d'extraction DTS sera présenté en annexe C.1.

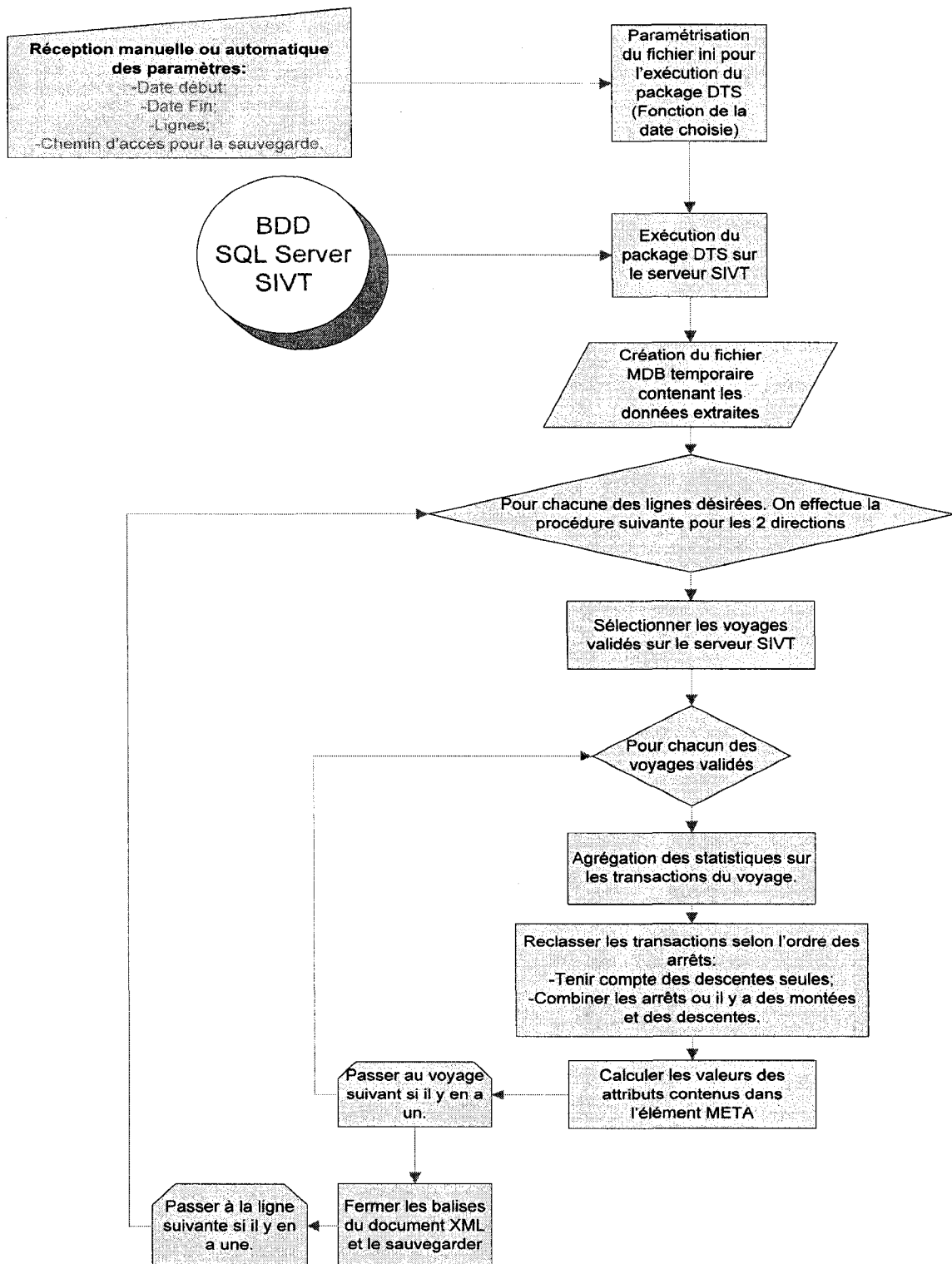


Figure 4-2 Fonctionnement de la procédure d'extraction des données au format XML

Comme on le voit sur la figure précédente, il est possible d'entrer des paramètres d'extraction de façon manuelle ou d'y entrer les paramètres qui lanceront la procédure d'extraction avec les paramètres programmés dans l'application. Ceux-ci sont des paramètres par défaut destinés à faire l'extraction des données de toutes les lignes du réseau pour la date-système. Le lancement automatique de l'application peut être programmé à l'aide des «Scheduled Tasks» du panneau de configuration de Windows.

L'extraction à l'aide des paramètres entrés manuellement peut être utile pour recueillir les données pour plusieurs journées antérieures ou pour une ligne spécifique. On procède alors de la façon décrite dans les prochains paragraphes.

On appelle l'application «genXML_charges.exe» à l'aide du «Run» du menu «Start» de Windows. L'application doit recevoir une valeur pour les 4 paramètres suivants :

- paramètre 1 : le répertoire désiré pour l'extraction des fichiers;
- paramètre 2 : la date de départ pour laquelle on veut extraire des fichiers XML;
- paramètre 3 : la date de fin;
- paramètre 4 : le numéro de ligne désiré, en particulier.

Par exemple :

```
genXML_charges XML 2007-01-01 2007-03-31 0
```

Ceci va demander au logiciel d'extraire tous les fichiers XML pour toutes les transactions qui ont eu lieu du 1^{er} janvier 2007 au 31 mars 2007. Le «0» à la fin signifie que l'on souhaite avoir les résultats pour toutes les lignes du réseau.

À noter que si l'on veut utiliser un seul des paramètres mais que l'on souhaite obtenir les valeurs par défaut pour les autres, il faudra tout de même entrer une valeur pour les quatre. Les valeurs suivantes permettront donc de dire au programme que l'on souhaite obtenir les valeurs par défaut :

- Param 1 : Allo
- Param 2 : Allo
- Param 3 : Allo
- Param 4 : Allo --- ou bien --- 0 --- ou bien --- "" (chaîne de caractère vide)


```

C:\WINDOWS\system32\cmd.exe - genXML_charges Allo Allo Allo 33
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
U:\>cd C:\Documents and Settings\frvass\Bureau\App
U:\>e:
C:\Documents and Settings\frvass\Bureau\App>genXML_charges Allo Allo Allo 33

```

Figure 4-3 Paramétrage du logiciel genXML_charges

Cet appel prend les 3 premiers paramètres par défaut, mais demande l'extraction pour la ligne 33 seulement. La prochaine section porte sur le fonctionnement du site Web interactif diffusant les statistiques opérationnelles.

4.3 Fonctionnalités du site Web interactif

Cette section couvrira tous les aspects fonctionnels du site Web développé en illustrant ceux-ci à l'aide de nombreuses captures d'écran. Premièrement, nous verrons brièvement la configuration du serveur IIS. Ensuite nous aborderons l'utilisation de la page ASP, puis nous terminerons en passant en revue toutes les fonctionnalités des différentes pages de résultats du site Web.

4.3.1 Configuration du serveur IIS et organisation des répertoires

La configuration du serveur IIS ne nécessite pas beaucoup d'instructions particulières. Il suffit de créer un sous-site Web et d'y activer l'utilisation des scripts ASP pour que les pages du site Intranet fonctionnent. Le tout peut se faire à partir du panneau de configuration de Windows XP (à noter qu'il faut la version professionnelle). Il est par contre important d'expliquer l'organisation des répertoires du sous-site qui contiennent les fichiers nécessaires au bon fonctionnement de celui-ci. La capture d'écran suivante prise à partir du logiciel MS FrontPage nous montre les sous-répertoires ainsi que les fichiers qui se trouvent sous le répertoire principal «C:\Inetpub\wwwroot» (ou tout autre

4.3.2 Utilisation de la page ASP

La page ASP, nommée «genstats_sto.asp», est la pierre angulaire du site Web. Elle permet à l'utilisateur de choisir les statistiques qu'il désire visualiser. La page fonctionne à l'aide d'un formulaire contenant des menus déroulants et des boutons. Afin de guider l'utilisateur dans la recherche des résultats qu'il désire, les menus déroulants et les boutons apparaissent au fur et à mesure que celui-ci effectue davantage de choix en partant du général vers le particulier. Les figures suivantes expliquent concrètement cette approche, montrant la progression entre le choix des statistiques de la journée globale, vers le choix de l'affiche des statistiques d'un voyage spécifique sur une ligne et une direction donnée. Voici donc le point de départ de la page :



SIVT-Web / Choix du rapport statistiques

Mois et
Année: [choisir]

Date:

Ligne:

Voyage:

Figure 4-5 Page de départ du site Web «genstats_sto.asp»

Cette page ASP utilise le fichier «code.asp» qui contient un certain nombre de fonctions (de date notamment) utilisée par «genstats_sto.asp». Ceci nous permet de regrouper les dossiers (nommés sous le format AAAA-MM-JJ, comme mentionné précédemment) de données contenant les fichiers XML par mois et année afin de faciliter la recherche d'une journée spécifique, comme on le voit sur la figure suivante :

SIVT-Web / Choix du rapport statistiques

Mois et Année: janvier 2007

Date: (choisir)

Ligne:

voyage:

(choisir)

Lundi, le 1er

Mardi, le 2

Mercredi, le 3

Jeudi, le 4

Vendredi, le 5

Samedi, le 6

Dimanche, le 7

Lundi, le 8

Mardi, le 9

Mercredi, le 10

Jeudi, le 11

Vendredi, le 12

Samedi, le 13

Dimanche, le 14

Lundi, le 15

Mardi, le 16

Mercredi, le 17

Jeudi, le 18

Vendredi, le 19

Samedi, le 20

Dimanche, le 21

Lundi, le 22

Mardi, le 23

Mercredi, le 24

Jeudi, le 25

Vendredi, le 26

Samedi, le 27

Dimanche, le 28

Lundi, le 29

Figure 4-6 Choix du jour dans le mois sélectionné

Une fois le jour sélectionné, un premier choix d’affichage de statistiques global est offert à l’utilisateur, soit les statistiques globales de toutes les lignes pour la journée sélectionnée. L’utilisateur doit également choisir une direction (0 ou 1) pour obtenir les résultats selon le sens global désiré (l’affichage des résultats générés par cette sélection sera détaillé dans la prochaine section, soit pour le choix de la date du mardi 11 janvier 2005) :

SIVT-Web / Choix du rapport statistiques

Mois et Année: janvier 2005

Date: Mardi, le 11

Ligne: (choisir)

voyage:

Direction Globale: Direction 0

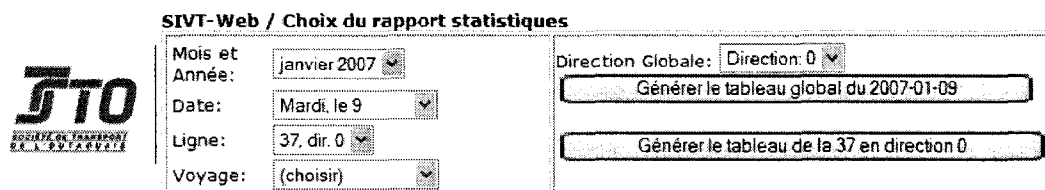
Générer le tableau global du 2005-01-11

Détails en section 4.3.3

Figure 4-7 Choix de l’affichage des statistiques globales de la journée

S’il le désire, l’utilisateur peut approfondir son choix en sélectionnant une ligne et une direction données suite au choix du jour comme indiqué à la figure 4-7. Un nouveau bouton permettant l’affichage des statistiques d’une seule ligne est alors présenté pour que l’utilisateur ait accès à des résultats plus détaillés. Par exemple, on pourrait choisir d’afficher les résultats de la ligne 37 en direction 0 (vers le centre, mais à une autre date,

ici le 9 janvier 2007). L'explication des statistiques générées par cette page sera présentée à la section 4.3.4.



SIVT-Web / Choix du rapport statistiques

Mois et Année: janvier 2007

Date: Mardi, le 9

Ligne: 37, dir. 0

Voyage: (choisir)

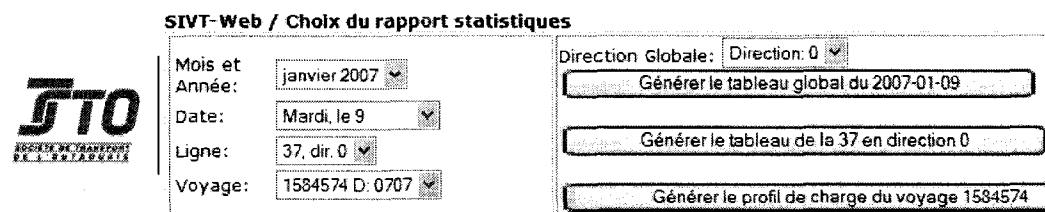
Direction Globale: Direction: 0

Générer le tableau global du 2007-01-09

Générer le tableau de la 37 en direction 0

Figure 4-8 Sélection d'une ligne – direction pour l'affichage de ces statistiques

Finalement l'utilisateur peut continuer sa recherche au dernier niveau, soit par la sélection d'un voyage à une heure de départ particulière effectué sur la ligne et la direction sélectionnées, toujours en fonction de la date choisie. Ceci va mener l'utilisateur vers la page interactive SVG montrant le profil de charge du voyage sélectionné :



SIVT-Web / Choix du rapport statistiques

Mois et Année: janvier 2007

Date: Mardi, le 9

Ligne: 37, dir. 0

Voyage: 1584574 D: 0707

Direction Globale: Direction: 0

Générer le tableau global du 2007-01-09

Générer le tableau de la 37 en direction 0

Générer le profil de charge du voyage 1584574

Figure 4-9 Choix d'un voyage pour l'affichage du profil de charge

L'exemple de la page générée par ce choix du voyage 1584574 partant à 7h07 sera présenté à la section 4.3.5. La page ASP sert donc d'interface au choix de l'utilisateur, une fois sa sélection effectuée, elle transmet les différents paramètres choisis (date, ligne, direction et voyage le cas échéant) au «template» XSL de la page associée. Le navigateur Web effectue alors la transformation du code XSLT sur le fichier XML en question et affiche le résultat à l'écran. Le code de la page ASP est présenté en Annexe C.3. Passons maintenant à la description du contenu des 3 principales pages de résultats.

4.3.3 Page des statistiques globales de la journée

Cette page permet à l'utilisateur d'observer les statistiques de toutes les lignes du réseau à une date et une direction donnée. On avait choisi précédemment la date du 11 janvier 2005, voici donc les résultats que nous obtenons sur cette page :



Statistiques quotidiennes pour toutes les lignes en direction: 0 - pour: Mardi, le 11 janvier 2005

LIGNE	# Montées	# Descentes	MAX des Charges	MAX # Validations	# Correspondances	# Adultes	# Érudits	# Aînés
Ligne 1	18	18	17	17	1	10	0	0
Ligne 3	5	5	5	1	4	0	0	5
Ligne 5	79	70	46	75	4	54	8	0
Ligne 11	76	68	34	76	0	60	4	2
Ligne 17	121	108	28	119	2	92	9	0
Ligne 21	22	15	6	19	2	12	0	5
Ligne 26	417	392	54	393	24	295	25	3
Ligne 27	471	421	50	435	36	269	76	1
Ligne 28	145	79	33	137	8	128	3	0
Ligne 30	49	49	13	46	3	35	5	2
Ligne 31	866	757	49	711	155	375	220	44
Ligne 33	596	548	35	483	113	286	137	29
Ligne 35	595	521	40	494	101	325	122	14
Ligne 37	1514	1289	66	1283	221	936	236	99
Ligne 39	698	774	58	771	127	494	194	20
Ligne 42	256	229	35	234	22	142	66	7
Ligne 43	291	266	72	247	44	150	87	11
Ligne 44	375	352	50	371	4	285	36	3
Ligne 45	556	446	80	517	39	416	50	4
Ligne 46	439	413	52	428	11	327	43	1
Ligne 47	14	10	11	8	6	3	7	0
Ligne 48	26	25	26	26	0	22	2	0
Ligne 49	146	88	30	95	51	65	45	7
Ligne 50	259	279	54	289	10	213	25	2
Ligne 51	164	150	37	154	10	124	3	2
Ligne 54	35	10	21	24	11	29	2	0
Ligne 55	194	184	53	188	6	149	15	4

Figure 4-10 Affichage des statistiques quotidiennes pour toutes les lignes

Ces résultats correspondent à l'agrégation des statistiques suivantes pour tous les voyages effectués à la date et la direction choisie pour toutes lignes ayant eu des transactions lors de la journée choisie :

- le nombre de montées total;
- le nombre de descentes total;
- le maximum de la charge maximum recensée parmi tous les voyages;
- le nombre de validations total;
- le nombre de correspondances total;

- le nombre d'adultes total;
- le nombre d'étudiants total;
- le nombre d'ânés total.

À la fin de cette page générée par le «template» XSL «sto_table_global.xml» (le code de cette page se retrouve en annexe C.4), on retrouve la somme ainsi que la moyenne de toutes ces statistiques :

Ligne 760	86	28	52	86	0	0	34	0
Ligne 837	28	18	21	24	4	3	22	1
Ligne 839	184	158	70	184	0	0	73	0
MOYENNE:	218	180	36	194	24	129	36	5
SOMME:	17193	14204		15297	1896	10179	2831	427

Figure 4-11 Moyenne et somme des statistiques globales

Cette page permet donc d'avoir un portrait rapide de ce qui s'est passé sur le réseau à une date donnée. Voyons maintenant la page des statistiques globales d'une ligne en particulier.

4.3.4 Page des statistiques globales d'une ligne

Comme mentionné sur la figure 4-8, l'utilisateur peut choisir d'afficher les statistiques d'une ligne en particulier, toujours après avoir choisi une date et une direction. L'exemple choisi précédemment est le mardi 9 janvier 2007 sur la ligne 37 en direction 0 (soit la direction centre-ville). Voici les résultats affichés par cette page qui utilise le «template» XSL «sto_table_ligne.xml» (le code de cette page n'est pas présenté en annexe, car il ressemble beaucoup à celui du template «sto_table_global.xml») :


Statistiques quotidiennes ligne: 37 - direction: 0 - pour: Mardi, le 11 janvier 2005

VOYAGE ID	Heure Départ	Facteur d'ajustement PPERI	Facteur d'ajustement PCENT	# Montées	# Descentes	# Charge MAX	# Validations	# Correspondances	# Adultes	# Enfants	# Aînés	# Passagers-KM
66728	0526	Facteur: 0,87	Facteur: 1,34	34	33	31	34	0	31	0	0	124,49
71520	0541	Facteur: 1,81	Facteur: 1,46	22	22	20	22	0	17	0	1	71,37
65809	0559	Facteur: 1,81	Facteur: 1,46	20	19	19	20	0	16	1	0	0
65820	0610	Facteur: 2,55	Facteur: 1,99	27	24	22	27	0	23	0	1	98,82
65898	0620	Facteur: 1	Facteur: 1,3	35	31	34	35	0	30	2	1	146,85
66212	0636	Facteur: 2,37	Facteur: 1,73	29	27	22	29	0	26	1	0	94,77
67152	0647	Facteur: 1,39	Facteur: 1,59	31	29	27	31	0	24	0	0	103,05
67129	0658	Facteur: 1,39	Facteur: 1,59	32	40	44	51	1	34	4	0	213,2
67126	0711	Facteur: 1,38	Facteur: 1,62	44	44	35	41	3	35	3	0	129,42
66118	0722	Facteur: 1,3	Facteur: 1,5	62	57	55	58	4	37	7	2	295,57
66206	0734	Facteur: 1,23	Facteur: 1,64	60	55	47	59	1	43	6	0	235,55
66226	0749	Facteur: 1	Facteur: 2,02	59	50	43	48	11	31	13	2	237,07
66670	0757	Facteur: 1	Facteur: 2,02	36	33	27	30	6	19	14	2	97,36
139039	0809	Facteur: 0,55	Facteur: 1,05	21	20	16	16	5	10	2	2	37,26
66102	0815	Facteur: 1	Facteur: 0,65	25	24	22	24	1	14	5	0	89,71
72092	0817	Facteur: 1	Facteur: 1,04	65	57	46	58	7	41	9	3	235,26
66963	0829	Facteur: 1	Facteur: 1,04	38	36	30	35	3	22	4	7	127,45
67154	0905	Facteur: 1,39	Facteur: 2,28	23	23	19	23	2	16	0	4	64,89
72012	0920	Facteur: 1,09	Facteur: 2,45	24	22	20	22	2	16	2	5	84,35
62936	0922	Facteur: 1,09	Facteur: 2,45	19	18	14	18	1	13	1	0	34,38
66375	0941	Facteur: 0,96	Facteur: 1,33	20	27	20	25	4	14	3	7	84,71
67162	1022	Facteur: 1,33	Facteur: 1,37	28	25	17	25	3	11	1	14	66,45
66712	1024	Facteur: 1,33	Facteur: 1,37	28	25	22	26	3	15	7	4	76,3
67156	1053	Facteur: 1,4	Facteur: 1,67	16	14	13	13	3	10	0	3	36,02
76468	1103	Facteur: 1,34	Facteur: 1,14	26	26	17	20	6	15	5	1	66,93
72012	1156	Facteur: 0,99	Facteur: 1,98	20	16	14	14	6	9	5	2	46,64
71775	1206	Facteur: 1,03	Facteur: 1,51	19	17	17	15	4	10	1	5	80,02

Figure 4-12 Statistiques globales de la ligne 37 en direction 0 le 2007-01-09

On remarque ici qu'on obtient de l'information sur tous les voyages effectués sur la ligne 37 en direction 0. De plus, comme on le voit sur la figure précédente (figure 4-12) quelques colonnes se sont rajoutées à celles présentées pour la page des statistiques globales afin de définir les détails plus précis relatifs aux résultats d'une seule ligne. Ces nouvelles colonnes sont les suivantes :

- l'identificateur du voyage;
- l'heure de départ du voyage;
- le facteur d'ajustement PPERI;
- le facteur d'ajustement PCENT;
- le nombre de passagers-KM.

Rappelons-nous ce qui a été dit précédemment dans ce mémoire, concernant PPERI et PCENT (section 3.4.1). La valeur des 2 facteurs a été calculée dans les 2 directions puisqu'au moment du calcul, on ne connaissait pas encore la signification de ceux-ci

dans la base de donnée de comptage. On devrait ici retrouver seulement la valeur du facteur PCENT, pour «Centre». Finalement, comme pour la page des statiques globales, la moyenne et la somme des résultats sont aussi calculées. Voyons maintenant les fonctionnalités de la page SVG contenant le graphique de la charge à bord d'un voyage spécifique.

4.3.5 Page SVG interactive sur la charge à bord

Comme mentionné dans la section 4.3.2 nous allons analyser le graphique de charge à bord du voyage 1584574 partant à 7h07 sur la ligne 37 en direction centre. Voici l'affichage de la page une fois chargée :

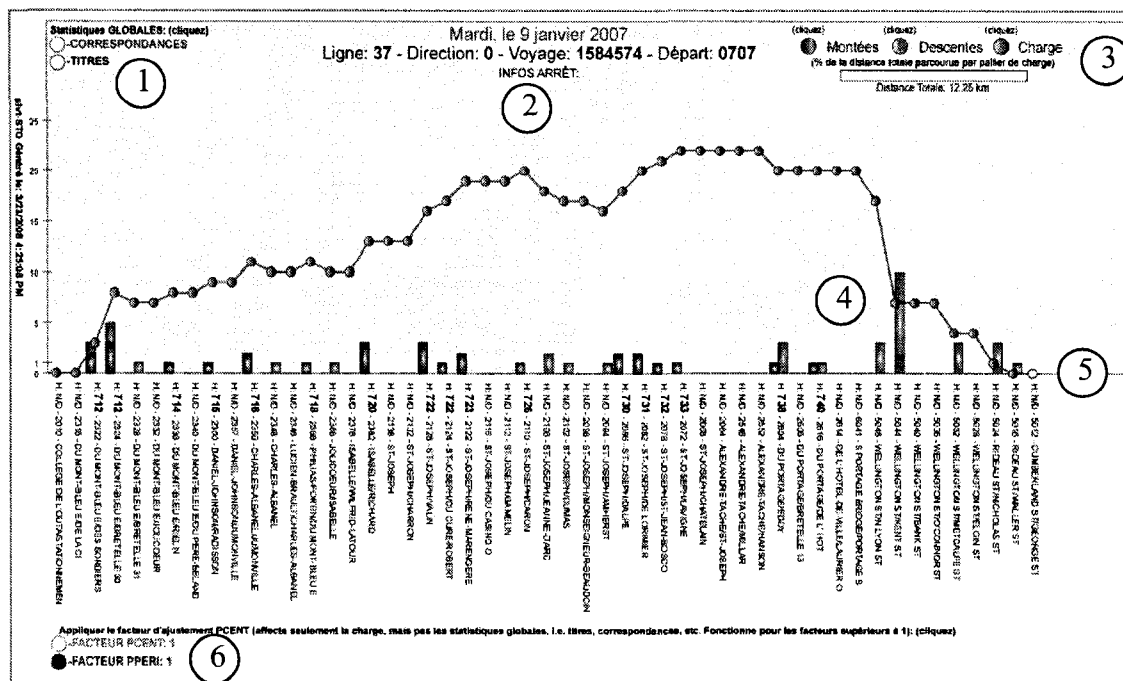


Figure 4-13 Profil de charge interactif du voyage 1584574 sur la ligne 37 en direction 0

Avant d'expliquer les différents points en détail, il est important d'expliquer ce qu'on voit à priori. Pour chaque arrêt de la ligne en abscisse (on a aussi le nom de l'arrêt et l'heure de la première transaction si il y a eu une montée), on voit la charge à bord (les points gris), les barres de montées (de 4 couleurs différentes selon le cas), ainsi que les barres de descentes. L'échelle du graphique se trouve à gauche.

Au point 1 se trouvent deux cercles dans lesquels on peut cliquer afin de faire apparaître un diagramme circulaire pour les statistiques désirées. Par exemple si l'on choisit les TITRES, le cercle sur lequel on a cliqué deviendra orange pour nous indiquer qu'on affiche le diagramme circulaire des TITRES pour le voyage global :

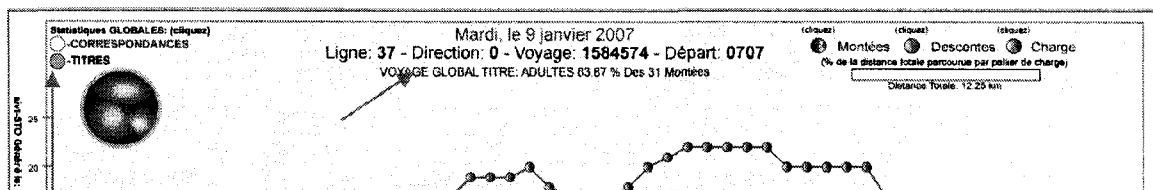


Figure 4-14 Affichage du diagramme de la répartition globale des titres pour tout le voyage

On remarque qu'il y a différentes couleurs pour chacun des secteurs afin d'identifier les différents titres. Voici les couleurs possibles selon le cas :

- vert foncé pour les adultes;
- vert pour les étudiants;
- vert pâle pour les aînés;
- brun pour l'ensemble des autres titres.

Maintenant, si on passe le curseur de la souris au dessus des différents secteurs, il apparaîtra en 2, des informations sur le dit secteur, soit son pourcentage de répartition. Si on retire complètement le curseur du graphique, on indique seulement le nombre total de montées pour le voyage. À la suite de l'exercice, on a pu remarquer que pour les 31 montées, il y a 89% d'adultes, 16% d'autres titres, aucun aîné et aucun étudiant.

Ensuite, si l'utilisateur veut plutôt avoir les statistiques globales du voyage sur les correspondances, il n'a qu'à cliquer sur le cercle bleu. Le cercle des titres redeviendra (voir flèche rouge à droite) blanc et un nouveau graphique apparaîtra à la place de l'ancien. Des informations spécifiques (flèche rouge centrale) à chaque secteur seront affichées également lors d'un déplacement du curseur de la souris au dessus de ceux-ci :

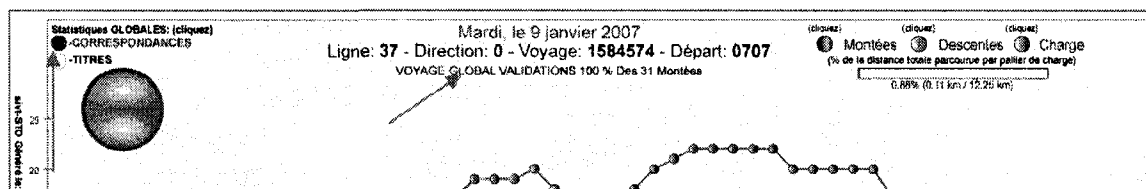


Figure 4-15 Affichage des statistiques globales sur les correspondances

Dans ce cas-ci, il n'y a que deux couleurs possibles pour les différentes sections :

- gris pour les validations;
- violet pour les correspondances.

Finalement, pour clore l'explication de la section 1 et 2 (voir la figure 4-13), si l'utilisateur ne veut pas voir aucun des deux diagrammes globaux, il n'a qu'à appuyer sur les cercles jusqu'à ce que la couleur de fond de ceux-ci soit blanche. À noter que d'autres informations relatives à d'autres sections pourront s'afficher en 2.

Regardons maintenant les sections 3 et 4. En section 3, on retrouve premièrement 3 cercles dans lesquels on peut cliquer afin de faire apparaître ou non les montants, les descendants ou la charge sur le graphique. Un clic pour disparaître (cercle blanc), un autre clic pour réapparaître (cercle de couleur):

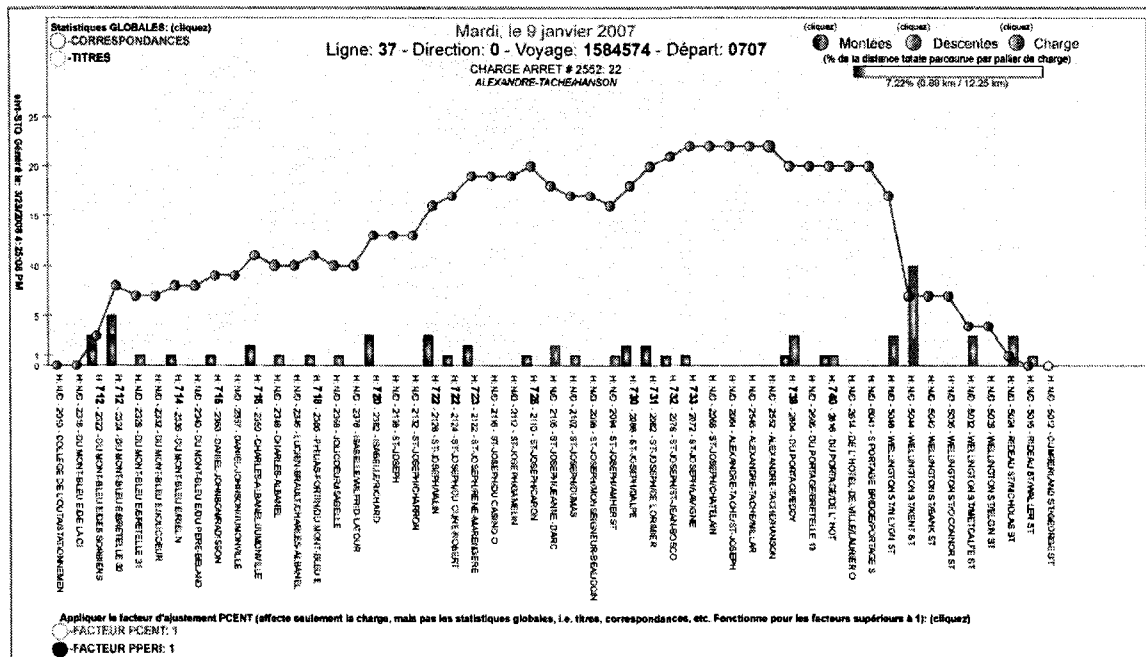


Figure 4-17 Présentation du pourcentage de la distance totale par palier de charge

À noter que si plus loin dans le parcours on trouve une même valeur de charge, celle-ci ne sera pas comptabilisée avec le palier en cours, c'est pourquoi on parle de palier de charge indépendant. L'utilisateur a pu peut-être aussi remarquer que les cercles des points de charge devenaient plus gros et de couleur bleue lorsque l'on passe notre curseur par-dessus. De plus, de l'information sur l'arrêt (charge, numéro d'arrêt et libellé) s'affichent en section 2 de la page.

Pour continuer sur cette portion du graphique (section 4, tout le graphique), l'utilisateur peut cliquer sur les différentes barres de montées (ici on a cliqué à l'endroit indiqué par la flèche rouge) afin d'avoir un diagramme circulaire montrant la répartition des titres de montants à l'arrêt en question. On remarquera que contrairement aux statistiques globales du voyage, le cercle du graphique n'est pas entouré d'orange et les informations inscrites en glissant notre curseur dessus, ne concernent aussi, que l'arrêt interrogé (cf. figure 4-14) :

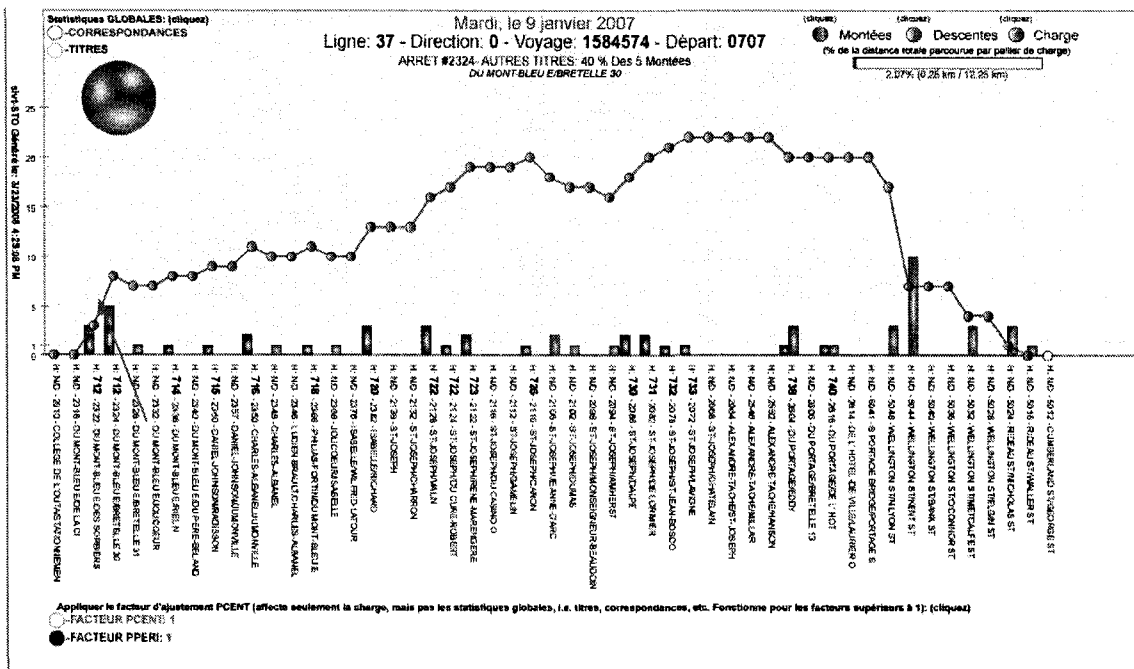


Figure 4-18 - Affichage des statistiques de titres d'un seul arrêt

On se doute maintenant que l'on puisse en faire autant pour les statistiques de correspondances et validations d'un arrêt en particulier. C'est en effet possible, il s'agit simplement de cliquer (ici on a cliqué sur l'arrêt 2110 tel qu'indiqué par la flèche rouge) sur le point de charge d'un arrêt. Dans ce cas-ci, pas de cercle bleu autour du graphique (cf. figure 4-15) pour indiquer qu'il s'agit des statistiques du voyage global :

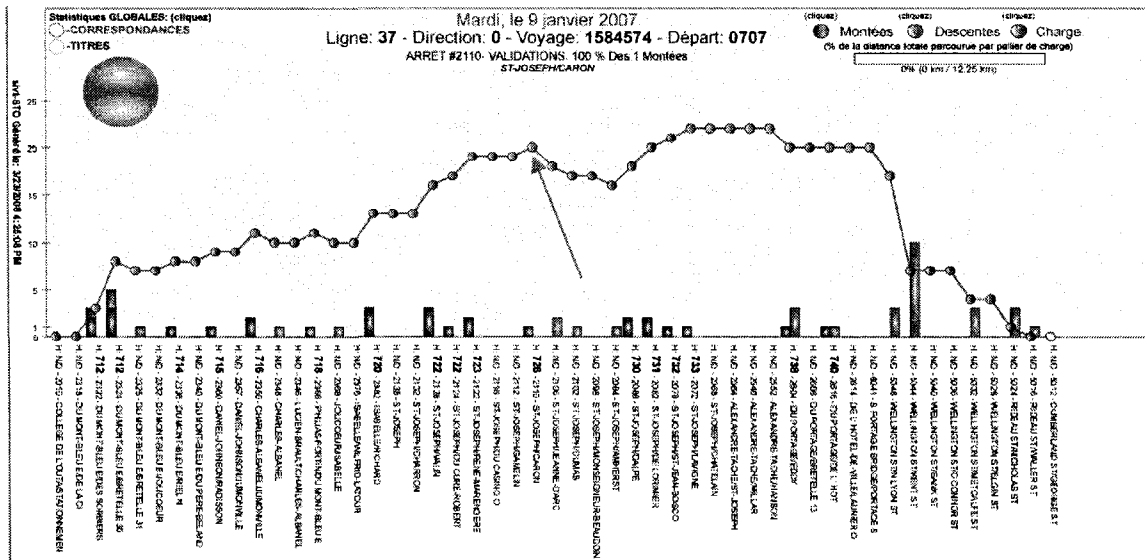


Figure 4-19 Affichage des statistiques de validations et correspondances d'un arrêt

S'il n'y a pas de montées à l'arrêt choisi, on indiquera à l'utilisateur en section 2 de la page, qu'il n'y a pas de montants à cet arrêt, et donc, aucun graphique ne sera affiché.

Ensuite, la section 5 de la page concerne la valeur de l'ajustement de la charge en fin de ligne afin de ramener la charge à 0. Il s'agit simplement aller placer le curseur de la souris sur le point de charge jaune du dernier arrêt de la ligne (voir flèche rouge). Il s'affichera alors en section 2 la valeur de l'ajustement de la charge, ici l'ajustement est de 0:

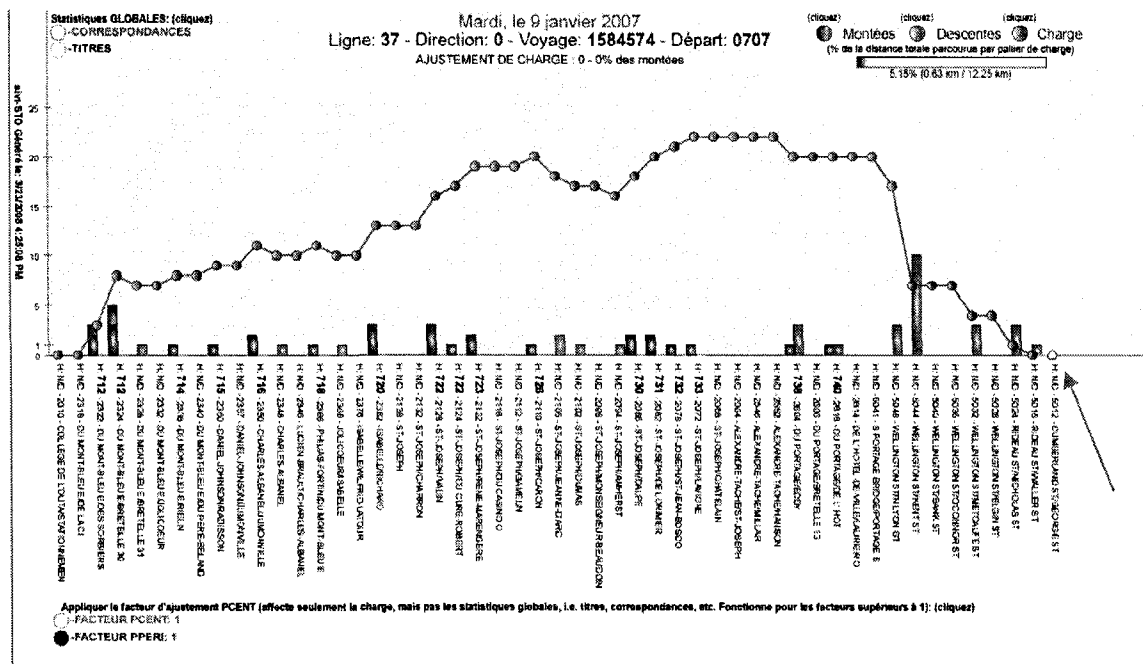


Figure 4-20 Facteur d'ajustement de charge en bout de ligne

Finalement, la section 6 de la page découlant du « template » XSL «sto_charge_svg.xml» (des exemples de code de cette page se retrouve en annexe C.5) permet d'activer les facteurs d'ajustement de la charge globale calculé à partir des comptes à bord. Ceux-ci fonctionnent un peu comme la section 1 de la page, on clique pour faire apparaître une nouvelle charge avec une nouvelle échelle en fonction de la valeur du facteur et vice versa. Si le facteur est inférieur ou égal à 1, il n'y aura pas de changement au graphique. À titre d'exemple, seul le bouton du facteur d'ajustement PCENT fonctionne. Une capture d'écran sur cette fonctionnalité est d'ailleurs fournie en annexe C.6.

Ce graphique interactif SVG permet aux utilisateurs d'avoir accès à une importante quantité de statistiques au niveau d'un voyage précis. Les informations sur la charge à bord et la répartition des titres des montants pourront être utiles à la planification et servir à différentes analyses comparatives au fur et à mesure que l'outil emmagasiner plusieurs journées de service.

La prochaine section porte sur diverses analyses qui ont été faites à l'aide de données recueillies par cet outil.

4.4 Analyses statistiques sommaires

Ce dernier chapitre porte sur certaines analyses que nous avons pu réaliser à l'aide de l'outil de visualisation des statistiques. Nous verrons en premier lieu comment l'outil peut nous aider à détecter des erreurs sur les données de cartes à puces. Ensuite nous verrons quelques résultats d'analyses statistiques réalisées sur les aspects opérationnels d'une ligne du réseau de la STO.

4.4.1 Détection des erreurs à l'aide de l'outil de visualisation

On a noté dans la revue que les systèmes de perception automatisés n'étaient pas sans erreur et qu'il était important de s'attarder à la détection et l'imputation de celles-ci afin que les analyses qui découlent des données recueillies par ces systèmes ne soient pas erronées.

En plus de fournir des informations opérationnelles au quotidien, l'outil de visualisation des statistiques développé, permet aussi de détecter quelques unes des erreurs recensées dans la section 3.1.2.3. Ces erreurs, que l'on peut détecter facilement, sont les suivantes et proviennent la plupart du temps d'une erreur humaine, par exemple :

- lorsque la transaction s'est effectuée en bout de ligne;
- lorsque le chauffeur n'a pas rentré le bon numéro de ligne, de direction, ou de voyage.

En effet, ce genre d'erreurs s'identifie assez bien avec la page de statistiques globales d'une ligne direction en regardant les aspects suivants :

1. la valeur du nombre de montant est identique à la charge maximale observée durant le voyage;
2. le nombre de passagers-kilomètres est anormalement bas ou trop élevé pour la charge observée.

Voici, à titre d'exemple, une capture d'écran sur les statistiques globales de la ligne 37 en direction 0 observées le mardi 16 janvier 2007. On remarque sur la figure, les statistiques du voyage 1599278 de 7h45, entourées en rouge :

Statistiques quotidiennes ligne: 37 - direction: 0 - pour: Mardi, le 16 janvier 2007

VOYAGE ID	Heure Départ	Facteur d'ajustement PPERI	Facteur d'ajustement PCENT	# Montées	# Descentes	# Charge MAX	# Validations	# Correspondances	# Adultes	# Étudiants	# Aînés	# Passagers-KM
1599397	0522	Facteur: 1	Facteur: 1	29	28	23	29	0	22	2	1	77.48
1604048	0534	Facteur: 1	Facteur: 1	25	24	20	24	1	18	1	3	70.24
1601906	0548	Facteur: 1	Facteur: 1	21	20	19	21	0	15	0	0	42.66
1601519	0614	Facteur: 1	Facteur: 1	35	35	29	32	3	26	0	1	120.33
1600902	0623	Facteur: 1	Facteur: 1	27	27	22	27	0	21	2	1	67.57
1598681	0641	Facteur: 1	Facteur: 1	36	36	27	36	0	24	2	0	103.96
1598326	0654	Facteur: 1	Facteur: 1	7	7	5	6	1	6	0	0	5.65
1598179	0703	Facteur: 1	Facteur: 1	52	52	52	51	1	31	8	1	214.09
1608551	0708	Facteur: 1	Facteur: 1	41	40	31	29	12	29	3	0	128.55
1600347	0720	Facteur: 1	Facteur: 1	46	45	28	43	3	24	4	0	140.97
1603102	0732	Facteur: 1	Facteur: 1	66	65	50	58	8	45	3	0	244.96
1601120	0739	Facteur: 1	Facteur: 1	53	51	55	50	3	25	14	0	177.98
1599278	0745	Facteur: 1	Facteur: 1	7	0	7	7	0	2	2	0	0

Figure 4-21 Statistiques anormales observées sur un voyage d'une ligne

Il y a en effet un problème sur les valeurs des indicateurs de ce voyage. La valeur du nombre de montées est égale à la charge maximale observée et le nombre de passagers-kilomètres est de 0. Le graphique de la charge à bord pourra sans doute éclairer ces anomalies. Voici ce que l'on observe pour le voyage en question :

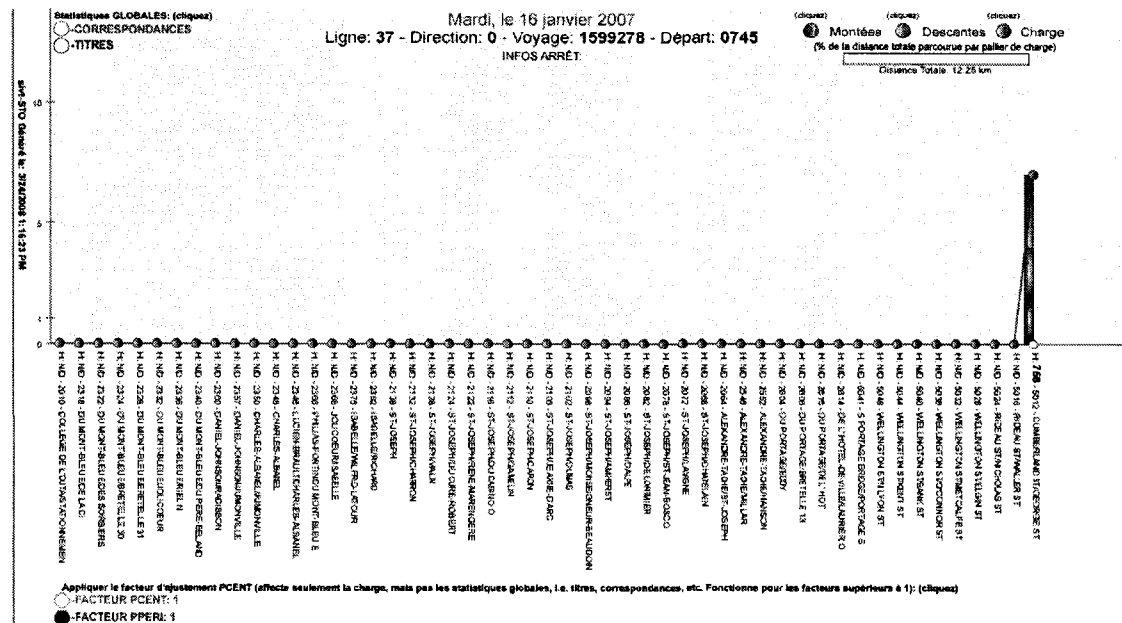


Figure 4-22 Détection d'erreur: montées en bout de ligne

On remarque ici qu'on est en présence d'une série de transactions de montées en bout de ligne. Ceci explique notre valeur de montant égale à la charge observée ainsi que la valeur nulle du nombre de passagers-kilomètres. Maintenant, s'il est difficile de savoir ce qui s'est réellement produit ici, on peut émettre l'hypothèse suivante : le chauffeur a oublié d'activer son voyage en cours et il s'en est rendu compte uniquement à la fin et l'a activé au terminus. Voici maintenant un autre type d'erreur observable facilement par l'outil, mais qui mène à d'autres hypothèses sur les causes de l'erreur. Encore une fois on commence par observer la page de statistiques globales (mêmes titres de colonnes que la figure 4-21):

1610717	1754	Facteur: 1	Facteur: 1	1	1	1	0	1	1	0	0	0,2
1610718	1759	Facteur: 1	Facteur: 1	14	14	7	9	3	6	4	1	14,04
1607606	1837	Facteur: 1	Facteur: 1	24	33	17	13	2	0	8	1	64,05
1604743	1917	Facteur: 1	Facteur: 1	24	23	24	23	3	19	10	1	39,56
1610918	2014	Facteur: 1	Facteur: 1	19	19	11	19	0	9	2	0	37,06
1608074	2121	Facteur: 1	Facteur: 1	21	20	12	19	2	12	4	1	33,56
1606567	2217	Facteur: 1	Facteur: 1	19	16	12	18	1	10	3	0	0
1605735	2317	Facteur: 1	Facteur: 1	7	7	5	7	0	6	1	0	6,7
MOYENNE:				25	24	17	21	4	13	5	1	31,81
SOMME:				1325	1283	879	1125	100	696	280	63	2743,73

Figure 4-23 Un autre voyage ayant des statistiques potentiellement anormales

On voit ici que le voyage 1604743 partant à 19h17 pourrait potentiellement être erroné puisque la valeur de la charge maximale est égale au nombre de montées. Toutefois, le nombre de passagers-kilomètres semble être d'une valeur normale. Il faut donc aller voir le graphique de la charge à bord :

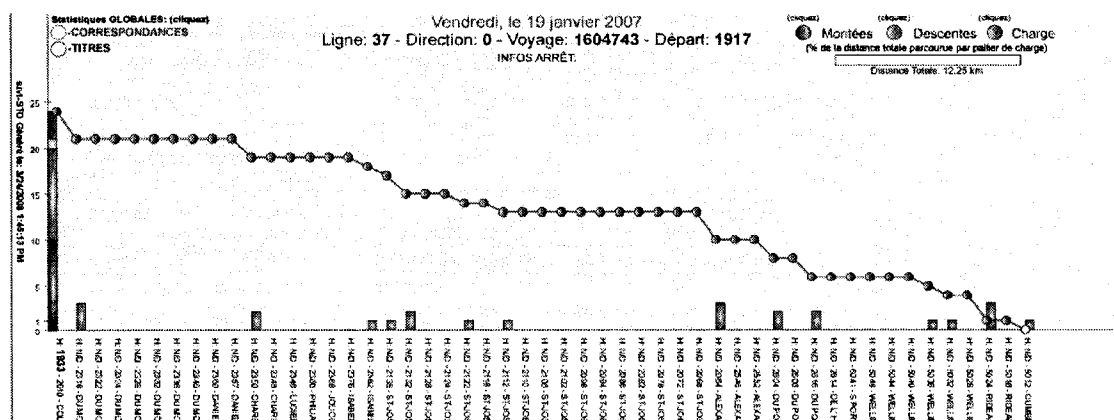


Figure 4-24 Autres anomalies observées sur un voyage

Cette fois-ci, on remarque que toutes les montées se sont effectuées au même arrêt, soit le premier de la ligne dans cette direction. On peut considérer qu'il y a eu effectivement 24 montants durant ce voyage, mais qu'il est très peu probable qu'ils soient tous montés au premier arrêt de la ligne. On peut donc émettre les hypothèses suivantes face à la cause du problème : il peut s'agir d'une défaillance du système GPS qui fait en sorte qu'il n'y a plus de reconnaissance de la position du véhicule et donc plus d'indentification de l'arrêt en cours, laissant ainsi les transactions s'accumuler au premier arrêt. Sinon, il peut aussi s'agir d'une erreur humaine du chauffeur, ayant oublié d'activer le départ du voyage en cours, ou bien que celui-ci ait sélectionné une mauvaise ligne entraînant l'incapacité de l'ordinateur de bord à reconnaître les arrêts en concordance avec les coordonnées GPS qu'il reçoit.

Un autre cas d'erreur que nous avons pu observer est celui des voyages vides avec aucune transaction, qui, encore une fois, peuvent être attribuables à une erreur humaine ou une défaillance d'équipement.

La détection d'erreur constitue donc un autre avantage de l'outil développé. Il permet le diagnostic rapide de certaines anomalies grâce à la page des statistiques globales nous permettant de cibler les voyages à analyser avec le graphique de charge à bord. Ce graphique nous permet ensuite de confirmer ou d'infirmer les anomalies détectées. En plus de nous fournir une très bonne piste sur la cause du problème, il permet ainsi de mieux en identifier la source et d'agir en conséquence pour éviter que cela se reproduise. Si cette méthode, après avoir effectué quelques tests (vérifier avec les chauffeurs si ils ont oubliés de valider leurs voyages dans le cas d'erreurs humaines, ou bien vérifier les équipements GPS si tout semble correct du côté des chauffeurs), s'avère efficace, il serait intéressant d'automatiser cette détection d'erreurs pour que celles-ci soient plus rapidement détectées par les utilisateurs du site.

4.4.2 Analyses statistiques sur la ligne 37

Une fois l'outil développé, nous avons pu l'utiliser pour produire plus rapidement quelques statistiques plus poussées sur une des lignes les plus occupées du réseau de la

STO, soit la ligne 37. Premièrement nous avons réalisé un diagramme espace-temps sur tous les voyages d'une même direction effectués un jour de semaine, soit le mardi 9 janvier 2007. Tous les voyages qui nous semblaient erronés (pour certains voyage, il y a aussi le type d'erreur suivant : aucune transactions sont recensées, ce qui est anormal) ont été retirés de la source de données du graphique. Pour réaliser le graphique nous avons dû effectuer une interpolation de l'heure des transactions pour des arrêts où il y a eu seulement des descentes afin de conserver une résolution satisfaisante:

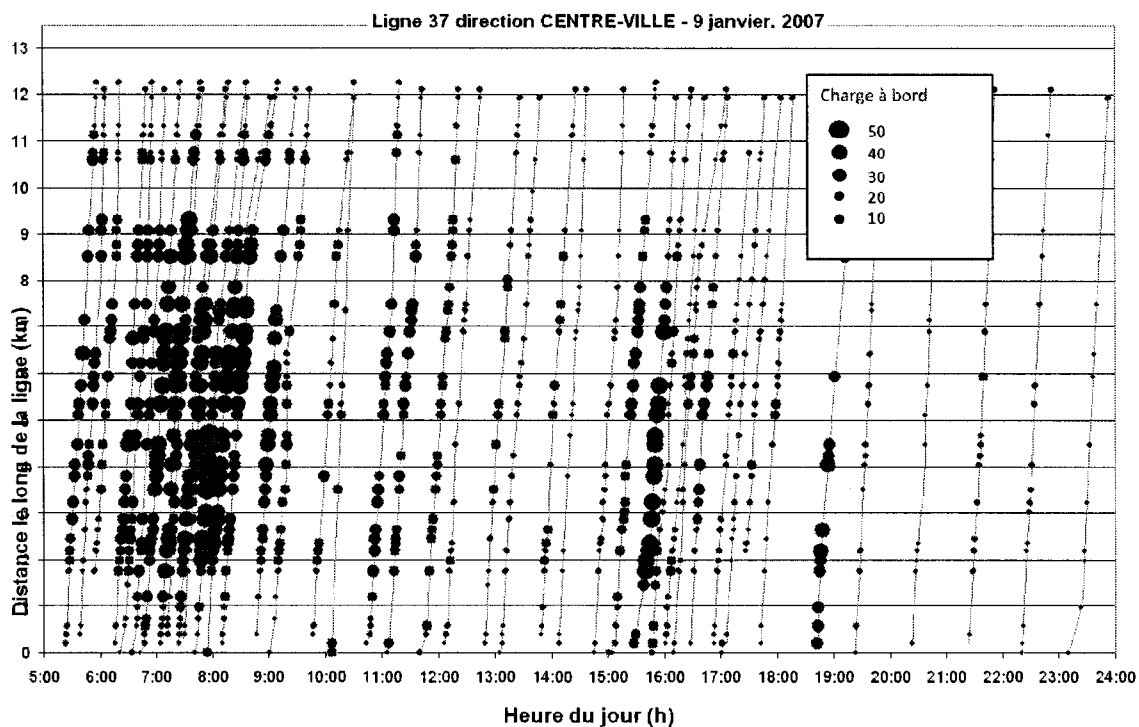


Figure 4-25 Diagramme espace-temps de la ligne 37, le 9 janvier 2007

Ce diagramme espace-temps nous donne de l'information sur la vitesse et la progression des voyages au cours de la journée. Le diagramme nous montre aussi la charge à bord à chaque arrêt où il y a eu une transaction de montée ou de descente. On voit notamment la haute densité de passagers le matin (direction centre-ville). On remarque aussi que les données GPS apportent des données précises (temps et distance) pour chaque arrêt où il y a eu une transaction. Ceci pourrait être utile afin de vérifier l'adhérence à l'horaire si nécessaire.

D'autres types d'études ont pu aussi être réalisés sur une plus longue période en analysant plusieurs fichiers XML d'une même ligne. Nous avons réalisé une étude des données statistiques quotidiennes de la ligne 37 encore une fois, mais sur une période d'un mois. L'évolution de la charge à bord maximale et du nombre de passagers-kilomètres a donc été recensée pour chaque jour du mois. La figure suivante présente les résultats :

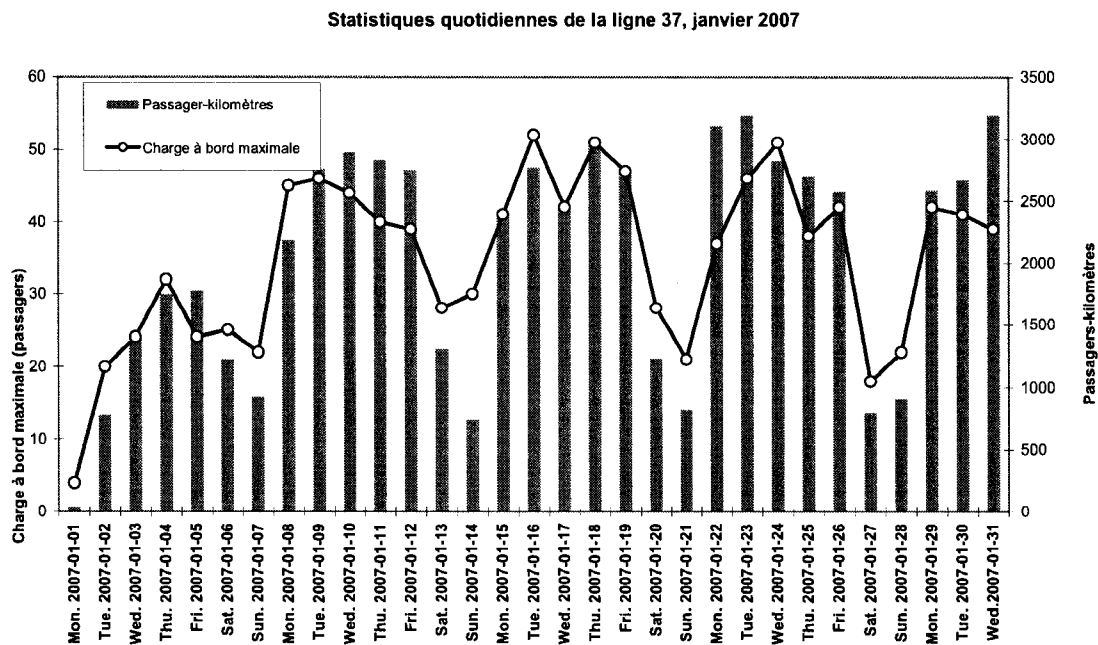


Figure 4-26 Statistiques quotidiennes pour la ligne 37, janvier 2007

On remarque évidemment à première vue, la grande différence entre les jours de semaine et les fins de semaine, ce qui est normal. Une variation moins importante entre les jours de semaine est aussi notable. De plus, on voit l'effet du début du mois de janvier (passagers-kilomètres plus faible qu'à l'habitude), où les gens sont encore en vacances (jusqu'au 4 ou 5 janvier). À partir de cette observation mensuelle et sachant que les horaires sont assez uniformes pour les jours de semaine (avec quelques variations pour les soirs de semaine), il y aurait peut-être ici une possibilité d'améliorer le service par rapport à la demande.

4.4.3 Comparaison des données de comptes à bord avec l'outil créé

Comme on l'a mentionné au début de ce mémoire, un des objectifs était de vérifier la validité des données de transaction du système SIVT et de l'algorithme de destination développé dans le projet précédant, en comparant les résultats avec de véritables comptes à bord fournis par la STO. De plus, comme il ne s'agit pas de la totalité des usagers de la STO qui utilisent la carte à puces, il serait intéressant d'avoir une idée du facteur d'ajustement à appliquer aux données de transactions afin d'avoir un portrait plus proche de la réalité. Nous avons toutefois calculé (malgré les limites énoncées, voir section 3.4.1) des facteurs d'ajustement à partir de ces données de comptages.

Nous pouvons quand même présenter quelques résultats sur ce sujet, en présentant d'abord comment nous avons calculé le facteur d'ajustement utilisé.

Pour chacun des voyages i d'une ligne, on peut calculer le facteur d'ajustement f_i :

$$f_i = \frac{C_i}{B_i}$$

Ici, C_i est la valeur du nombre de passagers comptés à bord du véhicule lors du voyage i (à une date précise dans la réalité) et B_i est le nombre de montées avec cartes à puces pour le même voyage. À noter que nous avons seulement pris des valeurs de comptage pour des jours de semaine et que nous avons transposé ces résultats pour des jours de semaines également.

Nous allons prendre ici, l'exemple du voyage de 7h12 sur la ligne 37 en direction centre-ville (année 2005) pour exposer nos résultats. Voici les étapes à réaliser pour calculer le facteur d'ajustement :

1. recensement des valeurs de comptages pour la ligne 37 en direction centre-ville effectués des jours de semaines en 2005 à des périodes identiques;
2. nous avons recensé une seule valeur correspondant à ces critères : 95 montants et une charge de 65. À noter que les valeurs (pour les montants) des années précédentes étaient de 71 en 2003 et 66 en 2004;

3. nous avons effectué l'agrégation de tous les montants de tous les voyages effectués durant la même période où les comptes ont été réalisés. Un critère du minimum entre 5 minutes plus tôt et 5 minutes plus tard que 7h12 a été utilisé afin de sélectionner les bons voyages;
4. une fois cette agrégation terminée, nous avons divisé le nombre total de montants par le nombre de voyages qui répondaient aux critères de période et d'heure;
5. nous avons effectué le ratio du comptage sur le résultat trouvé, ce qui donne un facteur d'ajustement de 1,62, soit 62%.

On peut maintenant regarder, sur une période de un mois comment les valeurs recensées par le système SIVT (valeurs réelles et valeurs ajustées avec le facteur) se comparent à la valeur de référence du comptage de 2005. Les jours de semaines du mois de janvier ont été utilisés :

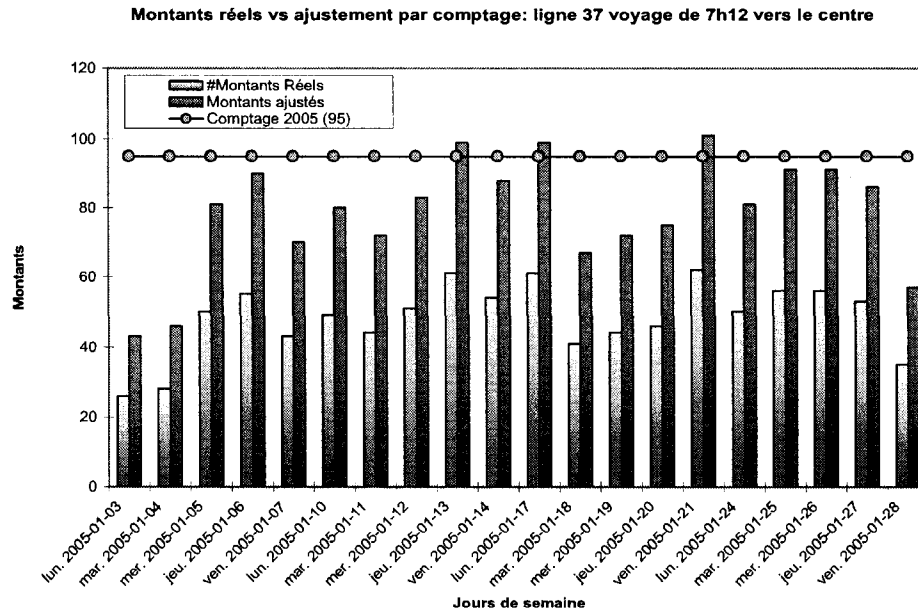


Figure 4-27 Comparaison des valeurs de montants réelles et ajustées pour un voyage spécifique

On remarque ici que les résultats reflètent d'avantage les comptages des années précédentes puisque seulement 3 voyages durant le mois ont recensé des montants

supérieurs à 95. La moyenne des écarts absolus (omis les valeurs du lundi 3 et mardi 4 janvier qui ne sont pas représentatives) par rapport au comptage, est de 14,16, ce qui semble normal (pas trop élevé).

On peut aussi faire l'exercice pour la valeur de la charge à bord qui a été observée. Toutefois le ratio n'a pas été calculé pour ce paramètre. Nous avons donc utilisé le 1,62 aussi, afin d'obtenir une idée. La charge de référence est de 65 ici :

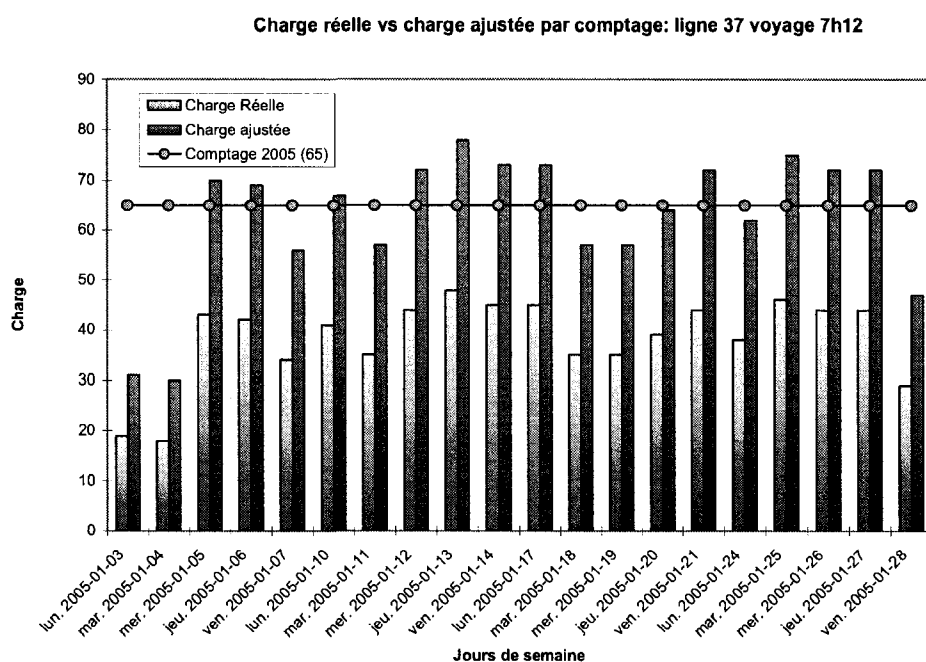


Figure 4-28 Comparaison des valeurs de charge à bord réelles et ajustées pour un voyage spécifique

On remarque ici que l'écart par rapport à la charge de comptage est inférieur à celui des montants. En effet, on note une moyenne absolue des écarts de 7,3. Évidemment, on sait que l'utilisation 1,62 n'est pas totalement justifiée ici, mais on peut aussi dire que la charge à bord est une caractéristique sujette à moins de variation que le nombre total de montants d'un voyage. Ce que l'on remarque aussi, c'est que l'algorithme de dérivation des destinations est efficace en répertoriant des charges similaires à celles observées en comptage. Évidemment, il faudrait beaucoup plus d'analyses pour confirmer cette observation.

De toute façon, on ne peut se fier totalement à tous les résultats rapportés dans cette section. Premièrement, il n'y a définitivement pas assez de comptages au cours d'une année pour d'avoir une idée précise du nombre d'utilisateurs pour le voyage en question. De plus, comme on ne sait pas la date précise de ces comptages, le fait d'agréger sur la période entière correspondante est peut-être une mauvaise idée, ne connaissant pas la variation du taux d'utilisation au sein d'une même période. Finalement, en général, nous avons trouvé des facteurs d'ajustements trop grands, la plupart du temps, en les comparant à l'estimation du nombre d'usagers de la STO qui ne possèdent pas de cartes à puces et qui est de 15% environ. Ce chiffre peut cependant être trompeur, puisque certaines lignes de la STO circulent à Ottawa et les utilisateurs ontariens ne possèdent pas non plus de cartes à puces, de plus, on ne sait pas si ce chiffre provient directement de l'utilisation du réseau ou bien des ventes de cartes à puces; il faudra donc tenir compte de ce fait ou bien établir différents pourcentages par secteur, par exemple. Il y aura donc beaucoup de travail à faire sur ce sujet afin d'obtenir des facteurs d'ajustements qui reflètent la réalité.

CONCLUSION

Ce mémoire a présenté une méthodologie de modélisation des caractéristiques opérationnelles d'un réseau de transport collectif à partir de données de cartes à puces. Cette modélisation, basée sur l'analyse des différents objets de transport grâce à l'approche orientée – objet et l'approche totalement désagrégée, a permis de développer une structure robuste de fichiers XML servant à alimenter la diffusion de statistiques opérationnelles à travers un site intranet.

La revue de littérature nous a permis de recenser les différentes méthodes utilisées en planification des transports en commun, de comprendre le fonctionnement et les enjeux des systèmes de perception automatisés par cartes à puces au sein des technologies des différents STI, ainsi que d'étudier les différents travaux de recherche portant sur le traitement des données de ces cartes.

L'étude des méthodes utilisées en planification nous a permis de voir les lacunes associées aux procédés qui n'accordaient pas assez d'importance à tous les aspects du déplacement individuel, comme c'est le cas avec l'approche séquentielle classique.

Nous avons vu par la suite que l'approche totalement désagrégée et sa continuité, l'approche orientée – objet représentaient parfaitement les aspects reliés à la collecte des données transactionnelles recueillies par les cartes à puces, puisqu'elles sont capables de traiter toutes les caractéristiques d'un déplacement individuel et qu'elles s'intègrent facilement à un environnement hautement informatisé.

Nous avons vu aussi les différentes technologies associées aux systèmes de transport intelligents qui, avec les AVL qui utilisent le positionnement par GPS permettent la mise en œuvre d'un système de perception automatisé récoltant des données de transactions aussi riches que celui utilisé par la STO. Réussir l'implantation d'un tel système est toutefois une tâche ardue nécessitant beaucoup de travail et d'investissements. Les nombreux avantages que l'on retire de l'implantation d'un tel système en valent toutefois les efforts. En effet, en plus d'être plus facile d'utilisation pour les chauffeurs

et les utilisateurs du réseau, les systèmes de perception automatisés facilitent grandement l'intégration des différentes réglementations tarifaires permettant à un utilisateur muni de sa carte de voyager sur plusieurs réseaux de transport. Outre les avantages reliés à la perception, les chercheurs ont rapidement capitalisé sur le potentiel de l'utilisation de ces données pour les aspects opérationnels du transport collectif. Ayant maintenant accès à une quantité volumineuse de données sur de grandes périodes continues, les chercheurs ont pu analyser le comportement des utilisateurs, sans compromettre leur confidentialité, pour mieux comprendre les impacts de ceux-ci sur la demande de transport. L'augmentation du nombre de travaux utilisant ces données a aussi fait ressortir les nombreuses erreurs qui peuvent s'incruster au sein des données. Des travaux sur la correction et l'imputation des erreurs ont donc été mis de l'avant.

Bien que chacun de ces travaux aie apporté des résultats, très peu ont pensé à démocratiser l'information pour les présenter aux planificateurs afin qu'ils puissent réellement l'utiliser au jour le jour. Nous avons donc, en quelque sorte, remédié au problème en réalisant un outil de diffusion des statistiques opérationnelles destiné aux planificateurs. Nous avons vu que la technologie XML était très bien adaptée à la modélisation des caractéristiques opérationnelles voulues. En effet, comme nous pouvons être maîtres de la structure et du contenu des fichiers XML, et que ceux-ci supportent le principe de hiérarchisation des éléments qu'ils contiennent, ceci fait en sorte que cette technologie s'adapte très bien à la modélisation orientée – objet d'un réseau de transport. Une fois les données bien modélisées, nous avons vu qu'il existait une multitude de technologies dérivées du XML pour mettre en forme le contenu de celui-ci. À l'aide du XSLT, nous avons développé différents patrons de transformation du contenu des fichiers XML, nous permettant d'afficher nos résultats soit en tableaux HTML ou bien en graphiques interactifs utilisant la technologie SVG.

La méthodologie de modélisation développée a mis l'emphase sur la validation du modèle-objet de transport afin que celui-ci soit capable de supporter tous les concepts et toutes les hypothèses relatives aux statistiques que l'on souhaite dériver. Ceci nous a

assuré une bonne transition entre les données contenues dans la base de donnée SIVT, les données dérivées par l'algorithme de dérivation des destinations, et les fichiers XML mis en forme à l'aide de la modélisation orientée-objet. Ensuite, concernant le développement de l'outil de visualisation des statistiques, nous avons mis en place une méthodologie détaillée sur les étapes à suivre lorsqu'on souhaite ajouter de nouvelles fonctionnalités ou bien de nouveaux concepts à l'outil. Outre ces contributions concernant la structure de développement de l'outil, nous avons vu que celui-ci permettrait à ses utilisateurs plusieurs avantages. Premièrement, ils auront le choix de visualiser de façon globale les statistiques opérationnelles du réseau ou bien de se concentrer plus précisément sur une ligne, jusqu'à l'observation de la charge à bord d'un voyage spécifique. Ceci permettra donc d'effectuer au besoin des micros ajustements au niveau du service offert. Deuxièmement, l'outil permettra la détection de certains types d'erreurs sur les données en plus de fournir des pistes sur la source potentielle de ces erreurs. Ceci aidera aussi à détecter du matériel défectueux lorsque les opérateurs ou les planificateurs informeront les mécaniciens des autobus potentiellement en faute. Finalement, une fois l'outil mis en place depuis quelque temps, il permettra à ses utilisateurs d'effectuer des analyses comparatives des données à travers le temps.

Le système développé a toutefois ses limites. Premièrement, on sait qu'il peut y avoir beaucoup d'erreurs au sein des données de cartes à puces. Bien que la STO ait fait des efforts dans ce domaine, il faudrait ajouter des procédures de détection et d'imputation d'erreurs supplémentaires, surtout pour le cas des correspondances, afin d'obtenir des données plus justes. Ce sera d'autant plus important lorsque les travaux de recherches seront rendus à utiliser le traitement des données de cartes à puces à un plus haut niveau de planification. Aussi, concernant les données, il ne faut pas oublier que le système ne couvre pas l'ensemble des usagers du réseau puisque certains d'entre eux n'utilisent pas la carte à puces. Il faudra donc travailler davantage sur l'obtention de bons facteurs d'ajustements nous permettant de calculer la marge d'erreur sur les données. Deuxièmement, concernant l'outil développé, il y a certaines limites face aux technologies informatiques utilisées. Nous avons trouvé que l'extraction des données

dans les fichiers XML était trop longue, il faudra envisager à l'avenir de développer une procédure d'extraction à l'aide des outils disponibles dans SQL server afin d'accélérer le traitement. Aussi, les performances du site Web diminuent au fur et à mesure que le nombre de répertoires de données augmente. C'est un sujet sur lequel il faudra se pencher. Finalement, nous avons trouvé que le traitement de l'affichage des données avec le XSL et le SVG était assez lourd à programmer et que l'avenir de ces technologies informatiques n'était pas assuré.

Concernant les perspectives de cet outil, nous croyons qu'il a le potentiel nécessaire pour être développé davantage, la méthodologie pour cette tâche étant déjà mise en place. Cependant, il faudra effectuer une revue des technologies de mise en forme du XML avant de se lancer dans un développement plus poussé. Selon nous, il serait intéressant d'ajouter un peu plus de statistiques opérationnelles comme les véhicules-heures, la vitesse commerciale ainsi que des informations plus précises sur les heures de passages aux arrêts. De plus, il serait intéressant d'automatiser davantage la détection des erreurs si des procédures n'ont pas déjà été mises en place pour régler ce problème.

Pour compléter l'analogie développée sur les travaux reliés aux cartes à puces, il faut présenter la quatrième étape des travaux qui permettrait de faire la boucle sur le processus de planification à long terme de la STO à l'aide des données des cartes. En d'autres mots, l'étape suivante serait d'établir des modèles de dérivation de matrices origines – destinations, basées sur la même modélisation des données que celles utilisées dans le cadre de ce projet. Évidemment ces modèles ne produiraient des résultats représentatifs que sur une période prolongée d'analyse. Ainsi, il serait possible d'agréger ces données aux autres données nécessaires par le simulateur de la STO afin d'effectuer une planification à long terme. Cette dernière étape représenterait donc une utilisation concrète des données de cartes à puces, soit jusqu'au plus haut niveau de planification.

RÉFÉRENCES

ATTOH-OKINE, N.O., SHEN, D.L., (1995). Security Issues of Emerging Smart Cards Fare Collection Application in Mass Transit, *Vehicle Navigation and Information Systems Conference, 1995. Proceedings. In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS. 'A Ride into the Future'*, pp 523-526.

BAGCHI, M., WHITE, P.R., (2004). What role for smart-card data from bus system?, *Municipal Engineer*, Volume 157, March 2004, pp. 39-46.

BAGCHI, M., WHITE, P.R., (2005). The potential of public transport smart card data, *Transport Policy*, Volume 12, Issue 5, September 2005, pp. 464-474.

BENKO, M., MIKKELSEN, P., (2005). Service and Technologies, *ITS Decision*, Septembre 2005. Disponible (page de départ et plus particulièrement):
http://www.calccit.org/itsdecision/serv_and_tech/Public_transit_tech/Transit/transit_report.htm.

BLYTHE, P., (2000). Transforming Access to and payment for transport services through the use of smart cards, *ITS Journal*, Volume 6, pp. 45-68.

BLYTHE, P., (2004). "Improving public transport ticketing through smart cards", *Municipal Engineer*, Volume 157, March 2004, pp. 47-54.

BOLCHINI C., F.A. SCHREIBER, (2002). Smart card embedded information systems: a methodology for privacy oriented architectural design, *Elsevier Science B.V., Data & Knowledge Engineering 41*, pp. 159-182.

BONNEAU, W. and editors. (2002). The role of smart cards in mass transit systems, *Card Technology Today*, June 2002, p.10.

BOS, B., XML in 10 points, *W3C Recommendation*, March 1999, revise en November 2001. Disponible: <http://www.w3.org/XML/1999/XML-in-10-points.html.en>.

BRAY, T., PAOLI, J., SPERBERG-McQUEEN, C.M., MALE, E., YERGEAU, F., Extensible Markup Language (XML) 1.0 (Fourth Edition), *WC3 Recommendation*, 2006. Disponible: <http://www.w3.org/TR/2006/REC-xml-20060816/#sec-intro>.

BYRNE, J. C., CROSS J., (2002). "Java for COBOL programmers" (2ième edition), *Charles River Media*, United States of America: Charles River Media, Inc.

CAMPESATO, O., (2004). "Fundamentals of SVG Programming" (1ère edition), *Charles River Media*, United States of America: Charles River Media, Inc.

CHAPLEAU, R. (1986). Transit Network Analysis and Evaluation with a Totally Disaggregate Approach, *World Conference on Transportation Research, Vancouver*.

CHAPLEAU, R., TRÉPANIÉ, M., LAVIGUEUR, P., ALLARD, B., (1997). Origin-Destination Survey Data Dissemination in a Metropolitan Context: A Multimedia Experience, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1551, Transportation Research Board of the National Academies, Washington, D.C., pp. 26-36.

CHAPLEAU, R., (2000). Planification des transports, *extraits de notes de cours*, École Polytechnique de Montréal.

CHAPLEAU, R., (2002). Multiple Uses of Large-Scale Origin-Destination Surveys, With a Totally Disaggregate Approach: The Montreal Experience, *Proceedings of the 30th Annual Conference of the Canadian Society of Civil Engineering*, Montreal, pp. 333-342.

CHIRA-CHAVALA, T., COIFMAN, B., (1996). Effects of Smart Cards on Transit Operators, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1521, Transportation Research Board of the National Academies, Washington, D.C., 1996, pp. 84-90.

CHU, K.K.A., & CHAPLEAU, R., (2007). Imputation techniques for missing fields and implausible values in public transit smart card data. *11th World Conference on Transportation Research, University Berkeley, USA (Track D4 – Data collection methods)*.

CHU, K.K.A., & CHAPLEAU, R., (2007). Modeling transit travel patterns from location-stamped smart card data using a disaggregate approach. *11th World Conference on Transportation Research, University Berkeley, USA (Track D1 – Transport demand modeling – passenger and freight)*.

CHU, K.K.A., & CHAPLEAU, R., (2008). Enriching Archived Smart Card Transaction Data for Transit Demand Modeling. *87th Annual Meeting of the Transportation Research Board, Washington, D.C., USA, janvier 13-17, 2008*.

CLARKE, R. (2001). Person location and person tracking: Technologies, risks and policy implications, *Information Technology & People*, Volume 14, Issue 2, pp. 206-231.

CNIL – Commission nationale de l'informatique et des libertés. *Recommandation relative à la collecte et au traitement d'informations nominatives par les sociétés de transports collectifs dans le cadre d'applications billettiques*, CNIL, Délibération N° 03-038, 2003.

DINNING, M., COLLURA, J., (1995). Institutional issues concerning the implementation of electronic payment systems in public transit, *Steps forward: proceedings of the Second World Congress on Intelligent Transport Systems, Tokyo, Japan, 1995*, pp 1462-1469.

EVJEN, B., FERGUSON, S., KAY, M., SHARKEY, K., THANGARATHINAM, T., VERNET, A., (2007). *Professional XML*, Wiley, Indianapolis, Indiana: Wiley Publishing, Inc.

GILLEN, D., CHANG, E., JOHNSON, D. (2001). Productivity Benefits and Cost Efficiencies from Intelligent Transportation System Applications to Public Transit. Evaluation of Advanced Vehicle Location, *Transportation Research Record 1747*, pp.89-96.

HERELLIER, J.-M., MÉRIGOD, P., (1999). "Pages Web Dynamiques avec ASP – PHP – SQL" (1ère edition), *CampusPress*, France, Paris : CampusPress.

HOFMANN, M., O'MAHONY, M., (2005). Transfer Journey Identification and Analyses from Electronic Fare Collection Data, *Proceedings of the 8th International, IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 13-16, 2005*, pp. 825-830.

KÉPÉKLIAN, G., L'historique des langages de l'Internet, de SGML à XHTML, *XMLfr : L'espace francophone*, 2000, XML Quel historique? Disponible : <http://xmlfr.org/documentations/articles/000321-0001>.

LEJEUNE, Y., (2004). "XML" (1ère edition), *Micro Application*, France, Paris: Micro Application.

MENZIES, B., (2004). Integrating Technologies for Transit Service Management, *Proceedings of the Canadian Urban Transit Association*, Volume 04F, Montreal, 43 p.

MORENCY, C., TRÉPANIÉ, M., AGARD, B., (2006). Contributions des données de cartes à puces à la compréhension des rythmes d'activités en transport, *Congrès annuel de l'Association des transports du Canada à Québec (Québec) 9 au 11 avril 2006*.

MORENCY, C., TRÉPANIÉ, M., AGARD, B., (2007). Measuring transit use variability with smart-card data, *Transport Policy*, Volume 14, Issue 3, 2007, pp. 193-203.

MORENCY, C., TRÉPANIÉ, M., AGARD, B., (2007). Profiling transit supply from smart card data. *11th World Conference on Transportation Research, University Berkeley, USA (Track E2 – Transportation system performance measurement)*.

MORRISON, M., (2002). "Sams Teach Yourself XML in 24 Hours" (2ième édition), *Sams*, United States of America: Sams Publishing.

SHELFER, K., PROCACCINO, D., (2002). Smart Card Evolution, *Communications Of The ACM July 2002*, Volume. 45, No. 7, pp. 83-88.

Smart Card move onward, Market Survey, *Card technology today*, October 2003.

TANG, K.X., WATERS, N.M., (2005). The internet, GIS and public participation in transportation planning, *Progress in Planning*, Volume 64, Issue 1, July 2005, pp. 7-62.

TRANCHANT, N., (2005). Modèle de dérivation des déplacements en transport collectif à partir de données de cartes à puces. École Polytechnique de Montréal, Québec, Canada.

TRÉPANIÉ, M., (1999). Modélisation totalement désagrégée et orientée - objet appliquée aux transports urbains, mémoire de thèse, École Polytechnique de Montréal.

TRÉPANIÉ, M., BARJ, S., DUFOUR, C., POILPRÉ, R., (2004). Examen des potentialités d'analyse des données d'un système de paiement par carte à puce en transport urbain. *L'Utilisation des Systèmes de Transport Intelligents (STI) à l'appui de la gestion de la circulation" du congrès annuel de l'Association des transports du Canada à Québec (Québec)*, p.4, 10-14.

TRÉPANIÉ, M., CHAPLEAU, R., (2001). Modélisation Analyse orientée-objet et totalement désagrégée des données d'enquêtes ménages origine-destination, *Canadian Journal of Civil Engineer*, No 28, pp. 48-58.

TRÉPANIÉ, M., CHAPLEAU, R., TRANCHANT, N. (2007) Individual Trip Destination Estimation in Transit Smart Card Automated Fare Collection System, *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, Taylor & Francis, Volume 11, Issue 1, pp. 1-15.

UTSUNOMIYA, M., ATTANUCCI, J. and WILSON, N. (2006) Potential Uses of Transit Smart Card Registration and Transaction Data to Improve Transit Planning. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1971*, TRB, National Research Council, Washington, D.C., pp. 119–126.

ANNEXE A: AJOUTS À LA REVUE TECHNIQUE DES TECHNOLOGIES

A.1 Diagramme temporel de l'évolution des langages

Voici le diagramme représentant l'évolution des langages du Web présentés dans la section 2.1.1.

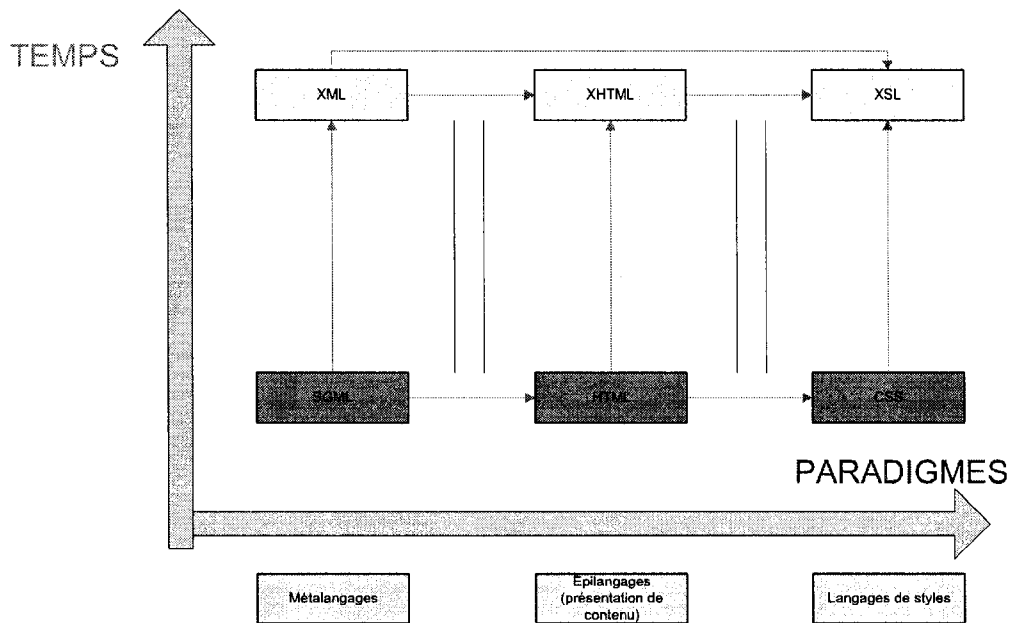


Figure A-1 Évolution des langages de programmation Web (inspiré de Képéklian, 2000)

A.2 Visualisation d'un document XML dans un navigateur Web

Voici maintenant un exemple de la représentation du code XML dans un navigateur Web, dans ce cas-ci, Internet Explorer. Bien qu'il n'y ait aucune mise en forme et que ce ne soit pas très joli au premier aperçu, il est néanmoins possible de lire facilement l'information.

```

- <Commande NoCommande="81332713233407">
- <LigneCommande>
  <Titre>Debt of Honor</Titre>
  - <Auteurs>
    <Auteur>Tom Clancy</Auteur>
    <Auteur>John Smith</Auteur>
  </Auteurs>
  <TypeCouverture>Paperback</TypeCouverture>
  <Prix>$6.99</Prix>
</LigneCommande>
- <LigneCommande>
  <Titre>The Hunt of the Red October</Titre>
  - <Auteurs>
    <Auteur>Tom Clancy</Auteur>
    <Auteur>Jake Redsmith</Auteur>
  </Auteurs>
  <TypeCouverture>Hard Cover</TypeCouverture>
  <Prix>$26.99</Prix>
</LigneCommande>
</Commande>

```

Figure A-2 Aperçu d'un document XML dans le logiciel Internet Explorer

Comme on le voit ci-dessus, pour chaque élément parent qui possède des enfants, donc pour les éléments qui sont sujets au concept de hiérarchisation, Internet Explorer met un signe « - » afin de pouvoir manipuler l'affichage des enfants à la guise de l'utilisateur. Par exemple, cacher les auteurs de la première ligne de commande et cacher la seconde ligne de commande entièrement :

```

- <Commande NoCommande="81332713233407">
- <LigneCommande>
  <Titre>Debt of Honor</Titre>
  + <Auteurs>
  <TypeCouverture>Paperback</TypeCouverture>
  <Prix>$6.99</Prix>
</LigneCommande>
+ <LigneCommande>
</LigneCommande>
</Commande>

```

Figure A-3 Choix de l'affichage du contenu des enfants d'un noeud parent

A.3 Particularités techniques du XML de base

Voici un tableau présentant quelques particularités techniques supplémentaires relativement au XML. Il s'agit en fait de règles particulières que l'on doit prendre en considération lorsque l'on crée un document XML.

Tableau A-1 Particularités techniques de base du XML

<i>Particularités</i>	<i>Exemple</i>
-Les noms des éléments sont sensibles aux majuscules et aux minuscules.	<Amis> <amis> seront considérés comme deux éléments différents.
-Tout élément doit avoir une balise de fermeture à moins qu'il ne s'agisse d'un élément vide.	<Amis> avec </Amis> ou bien <Amis />. (élément vide).
-Des ensembles d'éléments doivent obligatoirement être compris entre une autre paire d'éléments. Le code en exemple montre bien l'erreur sur la fermeture de l'élément <Amis> avant <Ami>. La hiérarchisation ne fonctionne plus.	<pre> <Amis> <Ami nom="Max" age="23"> </Ami> <Ami nom="Max" age="23"> </Amis> </Ami> </pre>
-La valeur des attributs doit apparaître entre guillemets. Une erreur sera retournée autrement.	<Ami nom="Max" age="23">
-Chaque document doit avoir un élément racine. Celui-ci doit être fermé obligatoirement de la manière classique s'il n'y a aucune information dans le document.	<Racine> </Racine>
-Certains caractères doivent être codifiés différemment pour ne pas entrer en conflit avec l'encodage XML.	<p>Les symboles suivants :</p> <ul style="list-style-type: none"> - < se traduit par &lt; - > se traduit par &gt; - " se traduit par &quot; - ' se traduit par &apos; - & se traduit par &amp; <p>À noter que le ; fait parti du code, donc, par exemple : <Ami nom="D'Auteuil"></p> <p>Doit s'écrire : <Ami nom="D&apos;Auteuil"></p>

A.4 Déclaration des attributs dans les DTD

Voici comment sont déclarés les attributs dans un DTD et comment leurs valeurs y sont régies:

```
<!ATTLIST NomElement NomAttribut TypeAttribut Default >
```

La section par défaut (*Default*) correspond soit à une valeur par défaut ou bien à l'un des trois mots réservés suivants :

- #REQUIRED – l'attribut est requis;
- #IMPLIED – l'attribut est optionnel;
- #FIXED – l'attribut à une valeur fixe;
- default – la valeur par défaut de l'attribut.

Pour ce qui est du type d'attribut, il y a une dizaine de choix possibles, ils sont énumérés dans le tableau suivant :

Tableau A-2 Types d'attributs dans les DTD (inspiré de Evjen et al., 2007)

<i>Type d'attribut</i>	<i>Explication</i>
CDATA	Correspond à du texte qui ne sera pas analysé par l'application, donc non traité et qui n'est pas soumis aux règles du XML.
Type énuméré	Une série de valeurs de chaînes de caractères, par exemple : <pre><!ATTLIST Auto couleur (rouge bleue verte) "bleue"></pre> Correspond aux valeurs possibles de couleurs, et, par défaut, c'est la couleur bleue.
NOTATION	Une notation déclarée à un autre endroit du DTD.
ENTITY	Une entité externe binaire (par exemple une photo JPEG).

Tableau A-3 Types d'attributs dans les DTD (inspiré de Evjen et al., 2007) (suite et fin)

ENTITIES	De multiples entités binaires externes, séparés par des espaces blancs.
ID	Identificateur unique.
IDREF	Référence à un autre identificateur déclaré ailleurs dans le DTD.
IDREFS	Plusieurs références à d'autres identificateurs.
NTOKEN	Un nom constitué de valeurs alphanumériques tels des virgules, des points et des _ (underscore).
NTOKENS	De multiples NTOKEN.

A.5 Déclaration des éléments dans les DTD

Les éléments du type «Element-only» qui contiennent des enfants utilisent des symboles afin de décrire les relations entre les éléments enfants et leur parent. En voici la liste et leur signification :

- Parenthèses (()) – Inclut une séquence ou un choix d'éléments enfants.
- Virgule (,) – Sépare les items dans une séquence, établissant ainsi le bon ordre d'apparition des éléments.
- Barre horizontale (|) – Sépare les items représentant un choix.
- Point d'interrogation (?) – Indique que cet élément enfant doit apparaître exactement une fois ou pas du tout.
- Addition (+) – Indique que cet élément enfant doit apparaître au moins une fois.
- Astérix (*) – Indique que cet élément enfant peut apparaître pour un nombre d'occurrence aléatoire.

- Aucun symbole indique que l'élément enfant doit apparaître une seule fois.

Un exemple est le meilleur moyen d'illustrer tous ces symboles :

```
<!ELEMENT CV (intro, (education | experience)+, passetemps?, references* )>
```

Donc, si on analyse cet exemple, l'élément `CV` ne contient que des éléments enfants qui sont régis selon les règles suivantes : l'élément `intro`, ne peut apparaître qu'une seule fois, ensuite les éléments `education` ou `experience` doivent apparaître au moins une fois, comme le signe `+` l'indique en dehors de la parenthèse. L'élément `education` doit apparaître exactement une fois, mais `experience` peut apparaître plusieurs fois. Il s'agit donc d'établir une liste des emplois ou expérience suivant chaque bloc d'éducation. Ensuite l'élément `passetemps` peut apparaître une fois ou pas du tout et finalement les `references` peut apparaître peu importe le nombre d'occurrences nécessaires.

Pour les éléments de type «Mixed», qui peuvent contenir à la fois du texte et des éléments enfants, il est seulement intéressant de noter qu'ils peuvent être déclarés ainsi :

```
<!ELEMENT NomElement (#PCDATA | ListeElements) >
```

La section `#PCDATA` indique que l'élément peut contenir du contenu tapé («character data») d'où `PCDATA` pour `Parsed Character Data`. «Parsed» pour indiquer que le logiciel qui traite le contenu XML doit aussi analyser ce texte et donc qu'il faut respecter les exceptions lorsqu'on tape des valeurs.

A.6 Exemple complet d'un DTD

Pour conclure sur les DTD, il est présenté ici un exemple complet d'un DTD et du document XML qu'il décrit. Il s'agit d'un journal de bord pour des entraînements sportifs.

```
<!ELEMENT entrainementjournal (session)+>
<!ELEMENT session (duree, distance, location, commentaires)>
<!ATTLIST session
date CDATA #IMPLIED
type (course | notation | velo) "course"
battements CDATA #IMPLIED >
<!ELEMENT duree (#PCDATA) >
<!ATTLIST duree
```

```

unites (secondes | minutes | heures) « minutes » >
<!ELEMENT distance (#PCDATA) >
<!ATTLIST distance
Unites (miles | kilometres | tours) « miles » >
<!ELEMENT location (#PCDATA) >
<!ELEMENT commentaires (#PCDATA) >

```

Voici maintenant le fichier XML associé à ce DTD :

```

<?xml version="1.0"?>
<!DOCTYPE entrainementjournal SYSTEM ``etjr.dtd`` >
< entrainementjournal >
<session date=``11/19/05`` type=``course`` battements=``158```>
  <duree unites=``minutes``>50</duree>
  <distance unites=``miles``>5.5</distance>
  <location>Warner Park</location>
  <commentaires> Entrainement matinal peu efficace</commentaires>
</session>
<session date=``11/21/05`` type=``velo`` battements=``153```>
  <duree unites=``heures``>1.5</duree>
  <distance unites=``miles``>25.5</distance>
  <location>Canal Lachine</location>
  <commentaires> Entrainement matinal agreable</commentaires>
</session>
<session date=``11/27/05`` type=``course`` battements=``163```>
  <duree unites=``heures``>2.5</duree>
  <distance unites=``miles``>15.5</distance>
  <location>Canal Lachine</location>
  <commentaires> Entrainement du midi agreable</commentaires>
</session>
</entrainementjournal >

```

A.7 Exemple complet d'un schéma XSD

Voici un exemple d'un schéma XSD qui décrit le même fichier XML utilisé pour l'annexe A.6.

```

<?xml version="1.0"?>
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema>
<xsd:element name="entrainementjournal">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="sessionType" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="sessionType">
<xsd:sequence>
  <xsd:element name="duration" type="xsd:duration" />
<xsd:element name="distance" type="distanceType" />
  <xsd:element name="location" type="xsd:string" />

```

```

    <xsd:element name="commentaires" type="xsd:string" />
</xsd:sequence>
<xsd:attribute name="date" type="xsd:date" use="required" />
<xsd:attribute name="type" type="typeType" use="required" />
<xsd:attribute name="battements" type="xsd:positiveInteger" />
</xsd:complexType>
<xsd:complexType name="distanceType">
<xsd:simpleContent>
    <xsd:extension base="xsd:decimal">
        <xsd:attribute name="unites" type="unitesType" use="required"/>
    </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="typeType">
<xsd:restriction base="xsd:string">
    <xsd:enumeration value="course" />
    <xsd:enumeration value="natation" />
    <xsd:enumeration value="velo" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="unitesType">
<xsd:restriction base="xsd:string">
    <xsd:enumeration value="miles" />
    <xsd:enumeration value="kilometres" />
    <xsd:enumeration value="tours" />
</xsd:restriction> </xsd:simpleType>

```

Si on compare les deux schémas, on se rend compte que le XSD semble plus complexe puisqu'il faut toujours se définir des types personnalisés à chaque fois que l'on souhaite avoir des choix de valeurs restreints. Par contre le XSD permet de mieux saisir la structure même du véritable document XML puisqu'il utilise aussi les mêmes règles.

A.8 Précisions sur Xpath et les éléments XSLT

Pour débiter, voici un exemple d'utilisation de XPath au sein d'un objet XSLT :

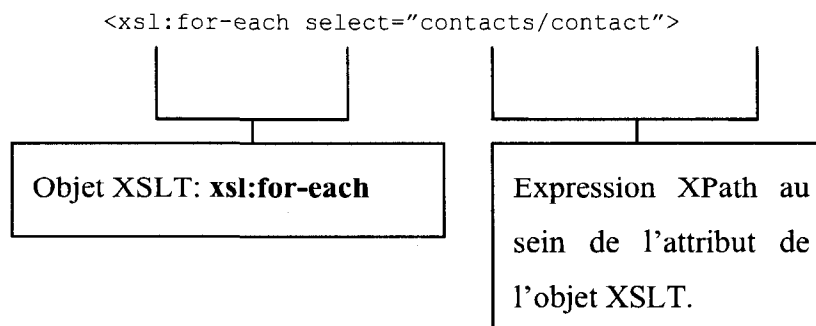


Figure A-4 Expression Xpath au sein d'un élément XSLT

Maintenant que l'on connaît mieux le fonctionnement du XSL, voyons un peu plus en détail l'objet XSLT *template*. Ce qu'il y a entre les balises `<xsl:template match="/"> ... </xsl:template>` va directement influencer ce que l'on va trouver dans l'arbre de résultats. Il est probable que l'on ait à utiliser un seul template pour parvenir à nos fins, mais la plupart du temps il nous faudra créer plusieurs templates pour transformer différentes portions de l'arbre du document XML original.

Dans l'exemple indiqué ci haut, on a mis l'expression Xpath `"/` dans la valeur de l'attribut `match` de l'objet `template`. Il s'agit du patron le plus vaste que l'on puisse appliquer à un document puisque celui-ci sélectionne l'entité de l'arbre, soit à partir de l'élément racine. La très grande majorité des documents XSL vont commencer par ce template que l'on applique à l'entité du document pour ensuite appliquer d'autres templates à des groupes de nœuds plus restreints de l'arbre. Voici maintenant une liste des différents objets (éléments) XSLT qui sont utilisés les plus fréquemment :

Tableau A-4 Éléments XSLT utilisés les plus fréquemment (inspiré de Morrison, 2002)

<i>Objet XSLT</i>	<i>Description</i>
<code><xsl:value-of</code>	Un élément crucial qui permet l'insertion d'une valeur du document XML source dans l'arbre de résultats. On peut y insérer une foule de formules mathématiques ou autre si l'on désire dériver de l'information à partir du document original.
<code><xsl:if</code>	Performe un test de condition sur une sélection. On peut utiliser différents opérateurs mathématiques pour les tests numériques.
<code><xsl:for-each</code>	Parcourt en boucle les éléments d'un document.
<code><xsl:apply-templates</code>	Applique un <i>template</i> dans une feuille de style
<code><xsl:choose> <xsl:when test="..."> </xsl:when> <xsl:otherwise> </xsl:otherwise> </xsl:choose></code>	Permet d'effectuer le code situé entre les deux balises <i>when</i> lorsque le test renvoie la valeur vraie. C'est un peu comme un <i>if</i> et un <i>else</i> , mais dans lequel on devrait imbriquer un autre bloc <i>choose</i> pour simuler un <i>elseif</i> .

Tableau A-5 Éléments XSLT utilisés les plus fréquemment (inspiré de Morrison, 2002) (suite et fin)

<code><xsl:sort></code>	Permet de faire un tri dans les nœuds dans des fins d'affichage.
<code><xsl:variable> <xsl:value-of select="" /> </xsl:variable></code>	Permet d'affecter une valeur à une variable afin de faciliter la dérivation d'information. Si l'on veut déclarer une variable de façon conditionnelle, on a qu'à insérer un bloc <i>choose</i> entre sa balise d'ouverture et de fermeture.
<code><xsl:param></code>	Un peu similaire aux variables, les paramètres permettent de stocker les valeurs que l'on désire y insérer en plus de pouvoir recevoir des paramètres provenant d'un script d'une page Web dynamique. C'est-à-dire que l'on peut avec cette balise passer en paramètres certaines valeurs que l'on pourra utiliser dans une expression Xpath, par exemple. Nous verrons cette utilisation plus tard dans ce mémoire. De plus, les paramètres peuvent être utilisés lors de l'appel de <i>template</i> récursif, que nous verrons dans la section limites du XSL.

A.9 Exemple de réaffectation d'une valeur à une variable en XSL

Pour nous permettre d'effectuer la banale opération $x = x + 1$; en XSL, il faut utiliser un nouvel objet XSLT qui est le `xsl:call-template`. Voici un exemple d'une telle situation où l'on veut agréger la valeur d'une statistique (total des passagers kilomètres) pour chacun des nœuds semblables rencontrés dans le document XML. On commence d'abord par appeler le template une première fois et on fournit en paramètre la variable à incrémenter ainsi qu'une variable compteur :

```
<xsl:call-template name="TOTAUX">
<!-- Passe en paramètres la valeur de la charge max et un compteur -->
<xsl:with-param name="i" select="0" />
<xsl:with-param name="PASSAGERS_KM_GLOBAL_LOCAL" select="0" />
</xsl:call-template>
```

Ensuite, dans le *template* comme tel, on vérifie qu'on n'est pas à la fin de notre boucle (ici, au dernier voyage), on prend la valeur en cours et on rappelle le *template* en incrémentant notre paramètre de la valeur en cours.

```
<xsl:template name="TOTAUX">
  <!-- Parametres -->
  <xsl:param name="i">0</xsl:param>
  <xsl:param name="PASSAGERS_KM_GLOBAL_LOCAL">0</xsl:param>
  <xsl:choose>
    <xsl:when test="$i &lt;= $NBOYS">
<xsl:variable name="PASSAGERS_KM_PONCT"
select="Frank:returnPKM(root/LI[@DIR=$direction]/ARRL/ARR,
root/LI[@DIR=$direction]/VOYS/VOY[position()=$i]/TRS/TRSA)" />
      <xsl:call-template name="TOTAUX">
        <xsl:with-param name="i" select="$i + 1" />
        <xsl:with-param name="PASSAGERS_KM_GLOBAL_LOCAL"
select="$PASSAGERS_KM_GLOBAL_LOCAL + $PASSAGERS_KM_PONCT"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <td align="center"> <xsl:value-of select="format-
number(($PASSAGERS_KM_GLOBAL_LOCAL div count($VOYS/VOY)), '0.##')" />
      </td>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Lorsqu'on voit que la boucle est terminée, donc que le test du *when* renvoie une valeur *false*, on inscrit la valeur finale dans l'arbre de résultats (section *otherwise*).

A.10 Exemple d'interactivité d'un dessin SVG à l'aide d'ECMS script

Voici maintenant un exemple d'interactivité avec le ECMS script sur le même dessin SVG présenté à la section 2.5.2. Le script que nous allons rajouter permettra de faire changer la couleur du cercle à l'aide d'un clic sur le rectangle. Voici d'abord le code afin de l'expliquer par la suite :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg viewBox="0 0 1024 768" width="100%" height="100%"
xmlns:svg="http://www.w3.org/2000/svg" color-interpolation-filters="sRGB">
  <script type="text/ecmascript">
    <![CDATA[

      function modifRond(event, ids)
      {
        var svgDocument =null;
        var target      =null;
        var gcNode       =null;
```



```

        target = event.getTarget();
        svgDocument = target.getOwnerDocument();
        gcNode = svgDocument.getElementById(ids);
        var verif = "";
        verif = gcNode.getStyle().getPropertyValue("fill");

        if(verif == "orange")
        {

gcNode.getStyle().setProperty('fill','blue');

        }
        else
        {

gcNode.getStyle().setProperty('fill','orange');
        }
    }
    ]}]>
</script>
<g>
    <circle cx="100" cy="100" r="60" fill="blue" id="rond"
stroke="red" />
    <g>
        <rect x="110" y="100" height="30.5" width="102.5"
fill="red" stroke="black" onclick="modifRond(evt, 'rond');" />
    </g>
</g>
</svg>

```

On remarque en premier lieu que l'on a rajouté l'appel de fonction «onclick» dans l'objet rectangle du SVG. Ceci permet d'appeler la fonction «modifRond» avec les paramètres *evt* et "rond". Le premier paramètre contient, si on veut la réussite de l'événement «onclick» et le deuxième contient la valeur de l'identificateur de l'élément *circle* dont on veut changer la couleur. Il est très pratique, lorsque l'on veut utiliser du script de nommer les éléments SVG avec un identificateur unique. Ceci nous permettra d'atteindre plus facilement les différents éléments à l'aide de la fonction *getElementById()*. L'identificateur se place dans l'attribut *id* des éléments.

La fonction *modifRond* commence donc par saisir l'événement pour pouvoir créer une instance (instancier) la variable du document global ainsi: `svgDocument = target.getOwnerDocument();` Ensuite, on peut aller affecter à notre variable de nœud *gcNode* l'élément qu'on désire modifier avec le *getElementById(ids)*. On se rappelle que le SVG découle du XML et donc que sa structure est elle aussi arborescente. Une fois qu'on a l'élément désiré, on peut accéder à ces différentes propriétés, comme le

''fill'' et en changer la valeur. Dans ce cas-ci, on vérifie la couleur à chaque clic de souris pour ainsi effectuer un changement à chaque clic aussi. Voici le résultat :

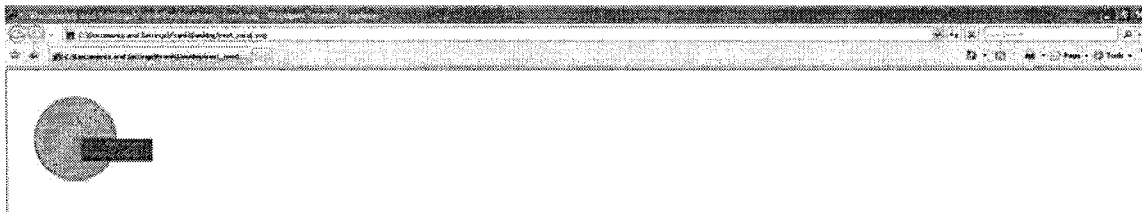


Figure A-5 Modification de la couleur du rond suite à un événement «onclick» sur le rectangle

ANNEXE B: AJOUTS À LA MÉTHODOLOGIE DE PROJET

B.1 Étapes préliminaires menant à l'algorithme de dérivation des destinations

Un certain nombre d'étapes ont dû être réalisées avant de passer à la modélisation de l'algorithme de dérivation des destinations tel que mentionné en section 3.1.2.1 :

5. Prétraitement des erreurs : Comme certaines parties du système SIVT doivent interagir avec des humains, il arrive que ceux-ci laisse passer quelques erreurs. Il peut s'agir du chauffeur qui entre le mauvais voyage dans le système de validation à bord de l'autobus ou bien un opérateur qui peut avoir fait des erreurs sur des horaires de voyage par exemples. Ces erreurs humaines entraînent donc des erreurs au niveau des données et celles-ci doivent être identifiées afin d'isoler ultimement les transactions non valides des transactions valides. Voici la liste des principales erreurs identifiées :
 - a. Présence le lignes 0 et 900 ne contenant pas de « vrais » arrêts;
 - b. Définition du réseau incomplète. Mauvaise correspondance du nombre de lignes et d'arrêts - lignes.
 - c. Certaines montées sont effectuées en bout de ligne. Si un voyage n'est pas saisi, l'ordinateur de bord attribue les montées au dernier arrêt de la ligne précédente ou à l'arrêt qui lui semble plus probable.
 - d. L'heure de départ du voyage est 0000.

Pour l'utilisation des données du présent projet, la STO avait établi quelques procédures de prétraitement des données afin de limiter le nombre d'erreur. L'algorithme mis en place pour trouver les destinations effectue toujours un traitement des erreurs, mais utilise les données prétraitées par la STO.

6. Traiter de façon désagrégée les données de déplacement : Il s'agit d'une série d'étapes de conceptualisation des différents éléments du réseau : représentation spatialisée et définition des éléments (objets) du réseau (arrêts, arrêts - ligne, ligne) et de leurs différentes propriétés (numéro de l'arrêt, coordonnées XY, numéro de ligne, sens, numéro de séquence, etc.). On y parle aussi de couverture du territoire (densité spatiale des lignes) et on compare quelques statistiques (longueur des lignes vs intervalle des arrêts) entre les différents types de lignes offertes aux usagers. Ceci a permis de mieux connaître le réseau disponible et de mieux comprendre la demande globale.
7. Analyse comportementale de la clientèle « carte à puces » : Ici, l'auteur a effectué une analyse de l'objet maître de la modélisation, soit l'utilisateur STO, qui donne un sens au réseau de transport. Un classement a été réalisé afin de rassembler de façon plus succincte les 36 titres de transports identifiés dans la base de données. Il en résulte 4 grandes classes, soit : les adultes, les étudiants, les aînés et les employés STO. Une analyse de la densité de montés sur les différents secteurs du réseau a aussi été menée. Il en résulte que les secteurs de Hull-Chelsea et Ottawa (les 2 sont fournisseurs d'emplois) sont les plus achalandés du réseau de transport.
8. Approche de dérivation de la demande : Les 3 étapes précédentes ont permis de mieux comprendre et cerner les concepts clés de la dérivation de la demande étudiés dans cette étape, à savoir le déplacement, la chaîne de déplacement et la notion de régularité. L'auteur effectue le recensement des propriétés de chacun de ces concepts (exemple : information spatio-temporelle de la montée, une date, une heure et un arrêt sont des propriétés d'un déplacement.). La notion de régularité des usagers est ensuite examinée à l'aide de l'étude des proportions d'arrêts et lignes utilisés par les usagers ainsi que les fréquences absolues et relatives. En ajoutant l'aspect temporel par type de jour et heures de montées, il a été possible de cerner les habitudes de déplacements pour les 3 principales

classes d'usagers (donc excluant les employés de la STO). On peut ainsi mieux regrouper les déplacements (leurs origines) dit réguliers, soit sujets à un intervalle de temps maximum d'une heure environ et une distance maximale de 2 kilomètres.

B.2 Programmation de l'algorithme de dérivation des destinations

Voici quelques exemples d'éléments à implanter à même le serveur MS SQL (version 2000), pour ensuite implanter l'algorithme de dérivation des destinations. Ceci permet au lecteur de voir des exemples de code PL/SQL (transact SQL).

-Modification aux tables

Modification de la structure de la table SIVT_TRANSACTIONBUS

```
alter table dbo.sivt_transactionbus ADD
    [DER_typeErreur] [tinyint] NULL ,
    [DER_Num_Arret_Dest] [int] NULL ,
    [DER_Distance_Dest] [int] NULL ,
    [DER_Typed] [tinyint] NULL ;
```

Création de la table POLY_INFOTABLES. Cette table sert à établir des statistiques sur le contenu des tables.

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Poly_INFOTABLES]') and OBJECTPROPERTY(id, N'IsUserTable')
= 1)
drop table [dbo].[Poly_INFOTABLES]
GO

CREATE TABLE [dbo].[Poly_INFOTABLES] (
    [tp_table] [varchar] (250) COLLATE French_CI_AS NULL ,
    [tp_nbenr] [numeric](18, 0) NULL
) ON [PRIMARY]
GO
```

-Vues

Les vues nécessaires doivent être créées pour assurer le fonctionnement des procédures :

```

CREATE VIEW dbo.Poly_voyage_reel_date
AS
SELECT      dbo.sivt_Voyage_Reel.*, dbo.sivt_Course_Reel.DateService AS
Dateservice
FROM        dbo.sivt_Voyage_Reel INNER JOIN
           dbo.sivt_Course_Reel ON dbo.sivt_Voyage_Reel.idCourse =
dbo.sivt_Course_Reel.idCourse

CREATE VIEW dbo.Poly_v_transaction_client
AS
SELECT      dbo.sivt_cartes.NumClient AS NumClient, dbo.sivt_transactionBus.*
FROM        dbo.sivt_cartes INNER JOIN
           dbo.sivt_transactionBus ON dbo.sivt_cartes.NumCarte =
dbo.sivt_transactionBus.numCarte

CREATE VIEW dbo.Poly_dir_ok
AS
SELECT      PeriodeDebut, PeriodeFin, num_li, num_ssli, REPLACE(DIR1, 'Nord',
1) AS DIR2
FROM        dbo.testpoly_dir_sud

CREATE VIEW dbo.Poly_dir_sud
AS
SELECT      PeriodeDebut, PeriodeFin, num_li, num_ssli, REPLACE(direction,
'sud', 0) AS DIR1
FROM        dbo.ss_ligne
WHERE       (NOT (num_li = 900))

CREATE VIEW dbo.Poly_statMonteesBrutes
AS
SELECT      TOP 100 PERCENT dateComp, typeTran, COUNT(*) AS nbmontees
FROM        dbo.sivt_transactionBus
GROUP BY   dateComp, typeTran
ORDER BY   dateComp, typeTran

CREATE VIEW dbo.Poly_v_Boutligne
AS
SELECT      dbo.SSLI_ARRET.NUM_LI, dbo.SSLI_ARRET.NUM_SSLI,
dbo.SSLI_ARRET.NUM_ARRET, dbo.SSLI_ARRET.PeriodeDebut,
           dbo.SSLI_ARRET.PeriodeFin
FROM        dbo.SSLI_ARRET INNER JOIN
           (SELECT      PeriodeDebut, PeriodeFin, NUM_LI,
NUM_SSLI, MAX(NUM_ORDRE) AS NUMMAX
           FROM          SSLI_ARRET
           GROUP BY     NUM_LI, NUM_SSLI, PeriodeDebut,
PeriodeFin) v_LIGNBMAX ON dbo.SSLI_ARRET.NUM_LI = v_LIGNBMAX.NUM_LI AND
           dbo.SSLI_ARRET.NUM_SSLI = v_LIGNBMAX.NUM_SSLI AND
           dbo.SSLI_ARRET.NUM_ORDRE = v_LIGNBMAX.NUMMAX AND
           dbo.SSLI_ARRET.PeriodeDebut = v_LIGNBMAX.PeriodeDebut
AND         dbo.SSLI_ARRET.PeriodeFin = v_LIGNBMAX.PeriodeFin

```

-Index

La creation des index suivants améliore le fonctionnement des procedures.

```
drop index sivt_transactionbus.idx_arrets_presents
CREATE INDEX idx_arrets_presents ON sivt_transactionbus (DER_Num_Arret_Dest)

drop index sivt_transactionbus.idx_typed
CREATE INDEX idx_typed ON sivt_transactionbus (DER_typed)

drop index sivt_transactionbus.idx_numcarte
CREATE INDEX idx_numcarte ON sivt_transactionbus (numcarte)
```

-Procédures

Les procedures suivantes doivent être ajoutées.

-Procédures d'ajout d'information

```
CREATE PROCEDURE Poly_ajouter_destination_I (@ide int, @der_num_arret_dest
int, @der_distance_dest int, @der_typed tinyint) AS
-- Martin Trépanier, août 2007
-- Projet STO - Polytechnique
-- Cette procédure ajoute la destination dans sivt_transactionbus

BEGIN
    SET NOCOUNT ON
    UPDATE sivt_transactionbus SET
der_num_arret_dest=@der_num_arret_dest, der_distance_dest=@der_distance_dest,
der_typed=@der_typed WHERE identificateur=@ide
END
GO

CREATE PROCEDURE Poly_ajouter_code_type_erreur_I (@ide int, @code tinyint) AS
-- Martin Trépanier, août 2007
-- Projet STO - Polytechnique
-- Cette procédure ajoute le type d'erreur dans la table sivt_transactionbus

BEGIN
    UPDATE sivt_transactionbus SET der_typed=255,
der_typeerreur=@code WHERE identificateur=@ide
END
GO
```

-Procédure pour détecter les erreurs

Il est nécessaire d'exécuter cette procédure à chaque nuit, avant d'exécuter la procédure de détermination des destinations. Il faut mettre la date du jour comme premier

paramètre, puis la valeur "0" (zéro) au paramètre suivant pour traiter toutes les cartes confondues.

```

CREATE PROCEDURE Poly_determiner_type_erreur_I (@datedeb datetime,
@datefin datetime, @nc bigint) AS
-- Martin Trépanier, août 2007
-- Projet STO - Polytechnique
-- Cette procédure permet de caractériser les types d'erreurs rencontrés par
une mise à jour de sivt_transactionbus

-- Déclaration des variables
declare @ide int
declare @num_li int
declare @num_sens int
declare @num_arret int
declare @heure_ligne char(4)
declare @compteur int
declare @datecomp datetime

SET @compteur=0
SET NOCOUNT ON

-- Ouverture du curseur maître
if @nc=0
    BEGIN
        DECLARE montees CURSOR READ_ONLY FOR select identificateur, num_li,
num_sens, num_arret, heureligne, datecomp from sivt_transactionbus where
datecomp >= @datedeb and datecomp <= @datefin
    END
ELSE
    BEGIN
        DECLARE montees CURSOR READ_ONLY FOR select identificateur, num_li,
num_sens, num_arret, heureligne, datecomp from sivt_transactionbus where
datecomp >= @datedeb and datecomp <= @datefin and numcarte=@nc
    END

    OPEN montees
    FETCH NEXT FROM montees into @ide, @num_li, @num_sens, @num_arret,
@heure_ligne, @datecomp
    WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @compteur=@compteur + 1
        -- Erreur type 1
        IF @num_li= 0 or @num_li= 500 or @num_li=900
            EXEC Poly_ajouter_code_type_erreur_I @ide, 1
        ELSE
            -- Erreur type 2
            if NOT exists(select num_li from SSLI_ARRET where num_li=@num_li
and num_ssli=@num_sens and (PeriodeDebut <= @datecomp) AND (PeriodeFin >=
@datecomp))
                EXEC Poly_ajouter_code_type_erreur_I @ide, 2
            else
                -- Erreur type 3
                IF NOT EXISTS(select NUM_ARRET FROM ARRET WHERE
NUM_ARRET=@num_arret and (PeriodeDebut <= @datecomp) AND (PeriodeFin >=
@datecomp))

```



```

EXEC Poly_ajouter_code_type_erreur_I @ide, 3
ELSE
  -- Erreur type 4
  IF NOT EXISTS(select num_li from SSLI_ARRET where
num_li=@num_li and num_ssli=@num_sens and num_arret=@num_arret and
(PeriodeDebut <= @datecomp) AND (PeriodeFin >= @datecomp))
    EXEC Poly_ajouter_code_type_erreur_I @ide, 4
  ELSE
    IF EXISTS(select num_li from
Poly_v_boutligne where num_li=@num_li and num_ssli=@num_sens and
num_arret=@num_arret and (PeriodeDebut <= @datecomp) AND (PeriodeFin >=
@datecomp))
      EXEC Poly_ajouter_code_type_erreur_I
@ide, 5
    ELSE
      IF @heure_ligne = '0000'
        EXEC
Poly_ajouter_code_type_erreur_I @ide, 6
      ELSE
        EXEC
Poly_ajouter_code_type_erreur_I @ide, 0

    FETCH NEXT FROM montees into @ide, @num_li, @num_sens, @num_arret,
@heure_ligne, @datecomp
    END

PRINT 'Nombre d''enregistrements traités: ' + CONVERT(char(10),@compteur)

CLOSE montees
DEALLOCATE montees
GO

```

B.3 DTD et schémas XSD des documents XML utilisés pour le site Web

Comme on l'a vu au chapitre 2, il est possible de créer des DTD ou des schémas XSD afin de valider le contenu des documents XML que nous créons. Voici donc le DTD et le schéma XSD qui sont en mesure de valider les documents XML créés pour le site Web.

-Le DTD

```

<?xml version="1.0" encoding="UTF-16LE"?>
<!ELEMENT TRSA EMPTY>
<!ATTLIST TRSA

    NOA CDATA #REQUIRED
    HT CDATA #REQUIRED
    NBM CDATA ""
    VAL CDATA ""
    COR CDATA ""
    T1 CDATA ""

    T2 CDATA ""
    T3 CDATA ""

```

```

    NBD CDATA "">
<!ELEMENT TRS (TRSA)*>
<!ELEMENT META EMPTY>
<!ATTLIST META
    AC CDATA #REQUIRED
    GV CDATA #REQUIRED
    GC CDATA #REQUIRED
    GED CDATA #REQUIRED
    GAD CDATA #REQUIRED
    GAN CDATA #REQUIRED
    AE CDATA #REQUIRED
    PPERI CDATA #REQUIRED
    PCENT CDATA #REQUIRED>
<!ELEMENT VOY (TRS,META)>
<!ATTLIST VOY
    IDV CDATA #REQUIRED
    HD CDATA #REQUIRED
    HF CDATA #REQUIRED
    IDC CDATA #REQUIRED>
<!ELEMENT VOYS (VOY)+>
<!ELEMENT ARR EMPTY>
<!ATTLIST ARR
    NOA CDATA #REQUIRED
    ODR CDATA #REQUIRED
    X CDATA #REQUIRED
    Y CDATA #REQUIRED
    NOM CDATA #REQUIRED>
<!ELEMENT ARRL (ARR)+>
<!ELEMENT LI (VOYS,ARRL)>
<!ATTLIST LI
    NOLI CDATA #REQUIRED
    DIR CDATA #REQUIRED>
<!ELEMENT root (LI)+>

```

-Le schema XSD

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="LI"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="LI">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="VOYS"/>

        <xs:element ref="ARRL"/>
      </xs:sequence>
      <xs:attribute name="DIR" use="required" type="xs:integer"/>
      <xs:attribute name="NOLI" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

```

```

<xs:element name="VOYS">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="VOY"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="VOY">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="TRS"/>
      <xs:element ref="META"/>
    </xs:sequence>
    <xs:attribute name="HD" use="required" type="xs:integer"/>
    <xs:attribute name="HF" use="required" type="xs:integer"/>
    <xs:attribute name="IDC" use="required" type="xs:integer"/>
    <xs:attribute name="IDV" use="required" type="xs:integer"/>
  </xs:complexType>
</xs:element>
<xs:element name="TRS">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="TRSA"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TRSA">
  <xs:complexType>
    <xs:attribute name="COR" type="xs:integer"/>
    <xs:attribute name="HT" use="required" type="xs:integer"/>
    <xs:attribute name="NBD" type="xs:integer"/>
    <xs:attribute name="NBM" type="xs:integer"/>
    <xs:attribute name="NOA" use="required" type="xs:integer"/>
    <xs:attribute name="T1" type="xs:integer"/>
    <xs:attribute name="T2" type="xs:integer"/>
    <xs:attribute name="T3" type="xs:integer"/>
    <xs:attribute name="VAL" type="xs:integer"/>
  </xs:complexType>
</xs:element>
<xs:element name="META">
  <xs:complexType>
    <xs:attribute name="AC" use="required" type="xs:integer"/>
    <xs:attribute name="AE" use="required" type="xs:integer"/>
    <xs:attribute name="GAD" use="required" type="xs:integer"/>
    <xs:attribute name="GAN" use="required" type="xs:integer"/>
    <xs:attribute name="GC" use="required" type="xs:integer"/>
    <xs:attribute name="GED" use="required" type="xs:integer"/>
    <xs:attribute name="GV" use="required" type="xs:integer"/>
    <xs:attribute name="PCENT" use="required"/>
    <xs:attribute name="PPERI" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="ARRL">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="ARR"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
</xs:complexType>
</xs:element><xs:element name="ARR">
  <xs:complexType>
    <xs:attribute name="NOA" use="required" type="xs:integer"/>
    <xs:attribute name="NOM" use="required"/>
    <xs:attribute name="ODR" use="required" type="xs:integer"/>
    <xs:attribute name="X" use="required" type="xs:integer"/>
    <xs:attribute name="Y" use="required" type="xs:integer"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

ANNEXE C: COMPOSANTES TECHNIQUES DE L'OUTIL DÉVELOPPÉ

C.1 Schéma du paquet d'extraction DTS

Voici le schéma du paquet d'extraction des données programmé sur le serveur SIVT. On ne peut pas voir tous les détails techniques de ce «Data Transformation Service package», mais on voit au moins les tables de la base de donnée SIVT qui sont affectées.

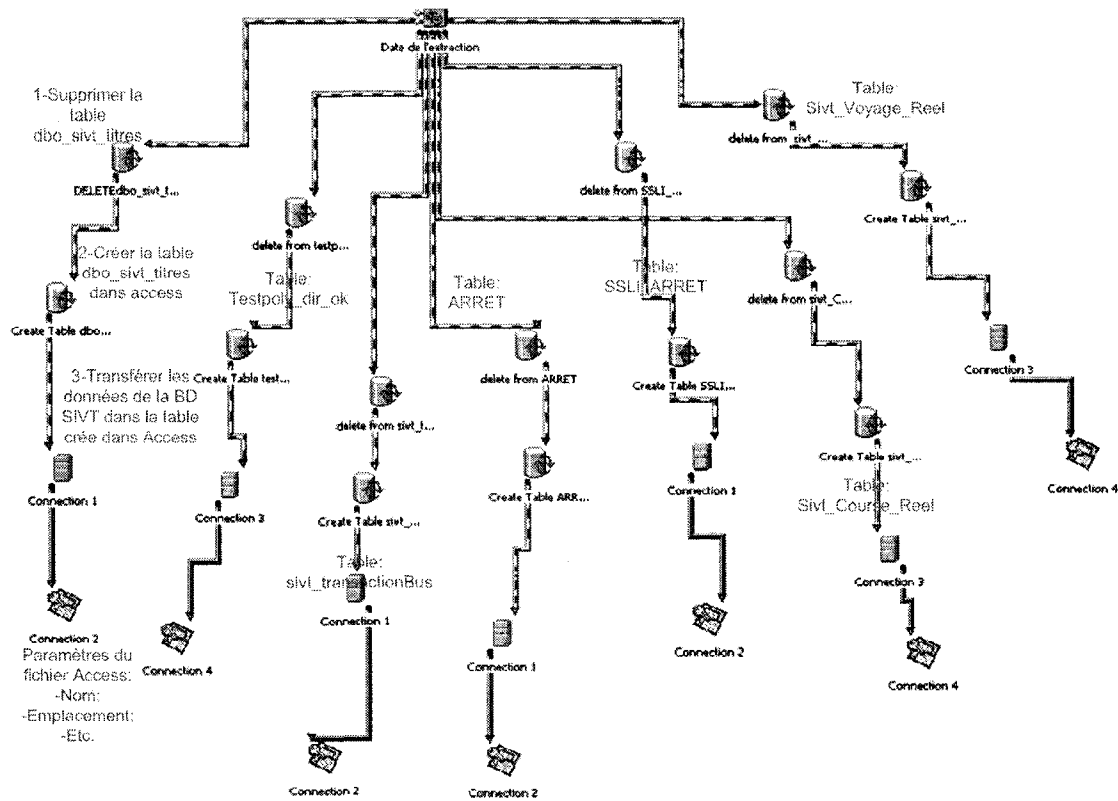


Figure C-1 Schéma du paquet d'extraction DTS

C.2 Exemples de code de l'application d'extraction genXML_charges

Voici des exemples de code de l'application programmée en Visual Basic. Net. On verra en premier lieu la procédure main présentée en entier afin de voir l'intégration des paramètres d'entrée de l'application ainsi que l'appel du package DTS :

```

**PROCÉDURE MAIN**

Imports System.Data.SqlClient 'Pour les connections directes à SQL Server
Module main
    Private WithEvents mobjPkgEvents As DTS.Package

    Sub Main()

'*****
**
'FRANÇOIS VASSIVIÈRE SEPTEMBRE 2007
'
'PROCÉDURE MAIN Appelle la procédure de génération des fichiers XML
'GenererXML_VF2
'autant de fois que nécessaire selon les 4 paramètres facultatifs suivants:
'
'ARGUMENT 1 : le chemin d'extraction pour les futurs fichiers XML
'              par défaut : "C:\Temp\XML_" & date1 ou date1 est la date
actuelle
'ARGUMENT 2:  la première date de départ pour l'algorithme si l'on veut
'extraire
'              des fichiers pour plus d'une journée
'ARGUMENT 3:  la date de fin le cas échéant
'ARGUMENT 4:  la ligne désirée en particulier, la valeur par défaut étant
'toutes *
'              les lignes du réseau.
'
'Note: la procédure prends en moyenne une heure à exécuter par journée.
'      Voir le document explicatif pour plus de détails sur la procédure et
ses
'      prérequis.

'*****
**

        'Variables générales

        Dim la_date As String 'la date en cours de traitement
        Dim la_ligne As Integer 'la ligne en cours de traitement
        Dim f As Integer 'fichier d'écriture pour le fichier INI du package
DTS
        Dim fich_extract As String 'Chemin d'accès pour l'enregistrement des
fichiers XML
        Dim date1 As String 'Première date, optionnel
        Dim date2 As String 'Deuxième date, optionnel
        Dim date_limite As String

```

```

Dim args(3) As String 'Arguments entrés par l'utilisateur, le cas
échéant
Dim args_default(3) As String 'Arguments par défaut
Dim le_dir As String 'Chemin d'accès du dossier par jours
d'extraction.
Dim i As Integer
Dim strSQL As String
Dim start, finish As Double 'Variables utilisées pour créer un délai
de 5 secondes avant la
'fermeture des connections.

'Arguments par défaut
date1 = My.Computer.Clock.GmtTime.Date
date2 = My.Computer.Clock.GmtTime.Date

fich_extract = "C:\Temp\XML_" & date1

args_default(0) = fich_extract
args_default(1) = "2007-01-01"
args_default(2) = "2007-01-31"
args_default(3) = 37

'Vérification des arguments entrés par l'utilisateur

If My.Application.CommandLineArgs.Count > 0 Then
    For i = 0 To My.Application.CommandLineArgs.Count - 1
        args(i) = My.Application.CommandLineArgs.Item(i)
    Next
End If

For i = 0 To 3
    If args(i) = "" Or args(i) = "Allo" Then
        args_default(i) = args_default(i)
    Else
        args_default(i) = args(i)
    End If
Next

'Boucler sur la date le cas échéant

If (My.Computer.FileSystem.DirectoryExists(args_default(0)) = False)
Then
    My.Computer.FileSystem.CreateDirectory(args_default(0))
End If

Do While CDate(args_default(1)) <= CDate(args_default(2))

    'Créer une copie d'une DB vide pour recevoir les données du
Package à chaque itération

    FileCopy("C:\Temp\DB\bobby.mdb", "C:\Temp\donnees_du_jour.mdb")

    My.Computer.FileSystem.CreateDirectory(args_default(0) & "\" &
args_default(1))

```

```

    la_date = args_default(1)
    date_limite = args_default(2)
    la_ligne = CInt(args_default(3))
    le_dir = args_default(0) & "\" & args_default(1) & "\"

    'Fichier
    f = FreeFile()
    Microsoft.VisualBasic.FileOpen(f, "C:\Temp\param_extraction.ini",
OpenMode.Output)

    'Ecriture dans le fichier INI du package pour lui fournir les
paramètres en cours

    Microsoft.VisualBasic.Print(f, "[DateExtr]" & vbCrLf)
    Microsoft.VisualBasic.Print(f, "ladate=" & la_date)
    Microsoft.VisualBasic.FileClose(f)

    'Appeler la procédure d'exécution du Package DTS
    Call RUN_PACK()

    'Ouvrir la connection DNSless vers le fichier ACCESS pour aller
chercher la liste des lignes du réseau
    Dim cnn2 As New ADODB.Connection

    cnn2.Open("Driver={Microsoft Access Driver
(*.mdb)};Dbq=C:\Temp\donnees_du_jour.mdb;Uid=Admin;Pwd=;")

    If args_default(3) > 0 Then 'Si il n'y a qu'une seule ligne à
générer
        GenererXML_VF2(la_ligne, la_date, le_dir)
    Else 'Sinon
        Dim rec_li As New ADODB.Recordset
        strSQL = "SELECT DISTINCT NUM_LI FROM SSLI_ARRET where NUM_LI
<900 order by NUM_LI"
        rec_li.Open(strSQL, cnn2)
        rec_li.MoveFirst()

        Do Until rec_li.EOF
            la_ligne = rec_li.Fields("NUM_LI").Value
            GenererXML_VF2(la_ligne, la_date, le_dir)
            rec_li.MoveNext()
        Loop
        'Fermer la connection et detruire le fichier
        rec_li.Close()
        rec_li = Nothing
    End If

    'Fermer la connection au fichier ACCESS
    cnn2.Close()
    cnn2 = Nothing

    start = Microsoft.VisualBasic.DateAndTime.Timer
    ' Set end time for 5-second duration. S'assurer que la connection
s'est bien terminée.
    finish = start + 5.0
    Do While Microsoft.VisualBasic.DateAndTime.Timer < finish
        Application.DoEvents()
        ' Do other processing while waiting for 5 seconds to elapse.
    Loop

```



```

'Destruction du fichier Access
My.Computer.FileSystem.DeleteFile("C:\Temp\donnees_du_jour.mdb")

'Itération sur la date jusqu'à la date limite le cas échéant.
args_default(1) = CStr((DateAdd(DateInterval.Day, 1,
CDate(args_default(1))))))

Loop

MsgBox("Génération du fichier XML Terminée!")

End Sub

```

****Ensuite on peut présenter quelques éléments de la procédure principale**

GenererXML_VF2**

```

Sub GenererXML_VF2(ByVal ligne As Integer, ByVal la_date As String, ByVal
le_dir As String)

```

```

'*****
**
'FRANÇOIS VASSIVIÈRE SEPTEMBRE 2007
'
'PROCÉDURE GENERERXML_VF2 génère un fichier XML contenant toutes les données
'                               nécessaires à la création de la super page SVG
selon '                               les3 arguments suivants
'ARGUMENT 1 : la ligne du réseau
'ARGUMENT 2:  la date désirée (le jour ou ont eu lieu les transactions)
'ARGUMENT 3:  le chemin d'accès pour l'enregistrement des dossiers
'Note: Il y a un appel de procédure pour remplacer les apostrophes et pour
calculer
'       le facteur CAP d'ajustement (voir la procédure)
'*****
**

```

```

'Connection DNSless au fichier Access
Dim cnn As New ADODB.Connection

```

...

****Exemple d'ouverture d'un recordset et début d'écriture dans le fichier XML****

```

'Dimensionnement en fonction du nombre le plus grand

```

```

'INFO_LIGNE (geometrie et libele arret)

```

```

strSQL_arr = " SELECT DISTINCT t.NUM_ARRET AS NOA, t.NUM_ORDRE AS
ODR, u.COORD_X AS X, u.COORD_Y AS Y, u.LIBEL_ARRET AS NOM " & _
" FROM SSLI_ARRET t, ARRET u, testpoly_dir_ok y " & _

```

```

" WHERE (t.NUM_LI = " & ligne & ") AND (t.PeriodeDebut
<= #" & la_date & "#) " & _
" AND (t.NUM_SSLI = y.num_ssli) AND (y.DIR2 = "" &
direction & "') AND (t.PeriodeFin >= #" & la_date & "#) " & _
" AND (u.PeriodeDebut <= #" & la_date & "#) AND
(u.PeriodeFin >= #" & la_date & "#) " & _
" AND (t.NUM_ARRET = u.NUM_ARRET) AND (t.NUM_LI =
y.num_li) " & _
" ORDER BY t.NUM_ORDRE "

rec_arr.Open(strSQL_arr, cnn)

If direction <> 1 Then
'Fichier XML, écriture de l'entête

f = FreeFile()
Microsoft.VisualBasic.FileOpen(f, le_dir & "Ligne_" & ligne &
".xml", OpenMode.Output)

'Ecriture

Microsoft.VisualBasic.Print(f, "<?xml version=""1.0""
encoding=""ISO-8859-1""?>")
Microsoft.VisualBasic.Print(f, "<root>")

instance = True

txtpoly = ""
If Not rec_li.EOF Then
rec_li.MoveFirst()
End If
End If

j = 0
i = 0

'No de ligne
txtpoly = txtpoly & "<LI NOLI="" & rec_li.Fields("NOLI").Value & ""
DIR="" & direction & "">"
If Not rec_voy.EOF Then
rec_voy.MoveFirst()
txtpoly = txtpoly & "<VOYS>"

Do While Not (rec_voy.EOF)
'No de voyage + heure départ
txtpoly = txtpoly & "<VOY IDV="" &
rec_voy.Fields("IDV").Value & "" HD="" & rec_voy.Fields("HD").Value & ""
HF="" & rec_voy.Fields("HF").Value & "" IDC="" &
rec_voy.Fields("IDC").Value & "">"
txtpoly = txtpoly & "<TRS>"
heureD = rec_voy.Fields("HD").Value
k = 0
z = 0
charge = 0
charge_max = 0
'Voyages

rec_trs(i).Open(strSQL_trs(i), cnn)

```

```

rec_trs_dest(i).Open(strSQL_trs_dest(i), cnn)

If Not rec_trs(i).EOF Then
    rec_trs(i).MoveFirst()
    Do Until rec_trs(i).EOF
        'Tant que ce n'est pas la fin et que la transaction
se passe à l'arrêt en cours
        Do While rec_trs(i).Fields("NOA").Value = arrets(i,
k) And Not rec_trs(i).EOF

            no_arret = arrets(i, k)
            no_ordre = ordres_mont(i, k)
            If m = 0 Then
                heure = rec_trs(i).Fields("HT").Value
            End If

            'Comptabiliser les stats

            If rec_trs(i).Fields("TTR").Value = 1 Then
                nb_val = nb_val + 1
            End If

            If rec_trs(i).Fields("TTR").Value = 3 Then
                nb_coor = nb_coor + 1
            End If

            If rec_trs(i).Fields("TITRE").Value Like "ET*"
Then
                nb_etudiants = nb_etudiants + 1
            End If

            If rec_trs(i).Fields("TITRE").Value Like
"ADULTE*" Then
                nb_adultes = nb_adultes + 1
            End If

            If rec_trs(i).Fields("TITRE").Value Like "AINE*"
Then
                nb_aines = nb_aines + 1
            End If

            nb_montees = nb_montees + 1

            rec_trs(i).MoveNext()
            m = m + 1

            If rec_trs(i).EOF = True Then
                Exit Do
            End If

        Loop

        m = 0
        stats(i, k, 0) = no_ordre
        stats(i, k, 1) = nb_montees
        stats(i, k, 2) = nb_val
        stats(i, k, 3) = nb_coor
        stats(i, k, 4) = nb_adultes

```

```

stats(i, k, 5) = nb_etudiants
stats(i, k, 6) = nb_aines

global_valid(i) = global_valid(i) + nb_val
global_coor(i) = global_coor(i) + nb_coor
global_et(i) = global_et(i) + nb_etudiants
global_ad(i) = global_ad(i) + nb_adultes
global_an(i) = global_an(i) + nb_aines

h = 0
fin_h = 0
'Enregistrement des infos de descentes
If Not rec_trs_dest(i).EOF Then
    rec_trs_dest(i).MoveFirst()
    Do While Not (rec_trs_dest(i).EOF)
        ordres_dest(h, 1) =
rec_trs_dest(i).Fields("ODR").Value
        ordres_dest(h, 2) =
rec_trs_dest(i).Fields("NBD").Value
        ordres_dest(h, 3) =
rec_trs_dest(i).Fields("NOA").Value
        ordres_dest(h, 4) =
rec_trs_dest(i).Fields("HT").Value
        h = h + 1
        rec_trs_dest(i).MoveNext()
        fin_h = h
    Loop
    rec_trs_dest(i).MoveFirst()
End If

```

C.3 Code de la page ASP

Voici l'intégrité du code de la page ASP du site Web développé (la page code.asp, n'est pas présentée ici):

```

<%@ Language=VBScript %>
<html>
<%
set fs = CreateObject("Scripting.FileSystemObject")
set objxml = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")

%>
<head>
<title>SIVT-Web / Choix du rapport statistiques</title>
<base target="_self">
<!--#include file="code.asp"-->
<link rel="stylesheet" type="text/css" href="sto.css">
</head>

<body bgcolor="#FFFFCC">

<form method=get action="genstats_sto.asp" id=form1 name=form1>
<table border="0" width="100%" id="table1" height="150">
<tr>

```

```

        <td width="128">
        </td>
        <td>
        <b>SIVT-Web / Choix du rapport statistiques</b><br>
        <table border="1" width="100%" id="table2">
        <tr>
                <td>
                        <table border="0" width="100%" id="table3">
                                <tr>
                                        <td width="70"><span lang="fr-
ca">Mois et Année:</span></td>
                                        <td>
                                                <% 'Va chercher les dates pour
lesquelles il y a des données
fs.GetFolder(Server.MapPath("data"))
onchange="form1.submit()">
value="">(choisir)</option>
                                                <select size="1" name="mois_an"
                                                <option
                                                <%
                                                i = 0
                                                for each f in folder.subfolders
                                                        moisF =
                                                                if i = 0 then
                                                                        mois_before =
                                                                                <%
                                                                                <option
value="<%=getMoisAn(f.name)%>"<%=iif(request("mois_an")=getMoisAn(f.name),"
selected","")%><%=mois_before%></option>
                                                                                <%
                                                                                else
                                                                                        if mois_before <>
                                                                                                <%
                                                                                                <option
value="<%=getMoisAn(f.name)%>"<%=iif(request("mois_an")=getMoisAn(f.name),"
selected","")%><%=moisF%></option>
                                                                                                <%
                                                                                                mois_before =
                                                                                                        end if
                                                                                                end if
                                                                                                i = i + 1
                                                                                        next
                                                                                                <%
                                                                                                </select>
                                                                                                </td>
                                </tr>
                        </table>
                </td>
        </tr>

```

```

ca">Date:</span></td>
lesquelles il y a des données
fs.GetFolder(Server.MapPath("data"))
then %>
onchange="form1.submit()">
value="">(choisir)</option>
trim(request("mois_an")) then
nbfolders + 1
trim(request("mois_an")) then
CInt (mid(f.name,9,2))
To 0 Step -1
then
folder.subfolders
getMoisAn(f.name) = trim(request("mois_an")) then
<tr>
<td width="60"><span lang="fr-
<td>
<% 'Va chercher les dates pour
set folder =
%>
<%if request("mois_an") <> ""
<select size="1" name="jour"
<option
<%
DIM Arraydate()
for each f in folder.subfolders
if getMoisAn(f.name) =
nbfolders =
end if
next
ReDim Arraydate (nbfolders-1)
i = 0
for each f in folder.subfolders
if getMoisAn(f.name) =
Arraydate (i) =
i = i + 1
end if
next
for k = UBound(Arraydate) - 1
for j= 0 to k
if Arraydate(j)>Arraydate(j+1)
temp=Arraydate(j+1)
Arraydate(j+1)=Arraydate(j)
Arraydate(j)=temp
end if
next
next
for i = 0 to nbfolders-1
for each f in
if

```

```

annee=left(f.name,4)
mois=mid(f.name,6,2)
nojour=mid(f.name,9,2)
nojourInt=CInt(nojour)

Arraydate(i) = nojourInt then
= mois & "/" & nojour & "/" & annee
= CDate(dateC)
= cddate(dateP)
datePres = cddatepres(dateP)

<option value="<%=f.name%>"<%=iif(request("jour")=f.name,"
selected","")%><%=datePres%></option>

end if
end if
next
next
%>
</select>
<%end if%>
</td>
</tr>
<tr>
  |
```

```

EXnojour & "/" & EXannee
EXdateC = EXmois & "/" &
EXdateP = CDate(EXdateC)
EXdateF = cddate(EXdateP)

set f =
fs.GetFolder(Server.MapPath("data") & "\" & trim(request("jour")))
nbfiles = f.Files.Count
DIM ArrayLigne()
DIM ArrayLigneBon()
DIM ArrayDir()
ReDim ArrayLigne (nbfiles-1)
ReDim ArrayLigneBon(nbfiles-1)
ReDim ArrayDir (nbfiles-1)

i = 0

for each fil in f.files
If UCase(Right(fil.Name, 3)) = "XML" Then
objxml.Load fil.Path
z = 0
For t = 0 To
objxml.selectNodes("//root//LI").Length - 1
laligne =
objxml.selectNodes("//root//LI").Item(t).Attributes(0).Value
ladir =
objxml.selectNodes("//root//LI").Item(t).Attributes(1).Value
ArrayLigne (i) = CInt(laligne)
next
ArrayLigne (i) = CInt(laligne)
if i = 0 then
lapremiereligne =
fil.Name
end if
end if
i = i + 1
next

for k = UBound(ArrayLigne) - 1
for j= 0 to k
if
ArrayLigne(j)>ArrayLigne(j+1) then
temp=ArrayLigne (j+1)
ArrayLigne (j+1)=ArrayLigne (j)
ArrayLigne (j)=temp
end if
next
next

i = 0

for i = 0 to nbfiles-1
for each fil in f.files

```



```

If
UCase(Right(fil.Name, 3)) = "XML" Then
instr(1,fil.name, "_")
instr(1,fil.name, ".")
Right(Left(fil.name, posPoint-1), (posPoint - posUnder)-1)
= ArrayLigne(i) then
    laligne = CStr(ArrayLigne(i))
    objxml.Load fil.Path
objxml.selectNodes("//root//LI").Length > 1 then
    ladir = "0"
    %>
    <option
value="<%=fil.name%>_<%=laligne%>_<%=ladir%>"<%=iif(request("ligne")=fil.name
&"_"&laligne&"_"&ladir," selected","")%>><%=laligne%>, dir.
<%=ladir%></option>
    <%
    ladir = "1"
    %>
    <option
value="<%=fil.name%>_<%=laligne%>_<%=ladir%>"<%=iif(request("ligne")=fil.name
&"_"&laligne&"_"&ladir," selected","")%>><%=laligne%>, dir.
<%=ladir%></option>
    <%
    ladir = objxml.selectNodes("//root//LI").Item(0).Attributes(1).Value
    %>
    <option
value="<%=fil.name%>_<%=laligne%>_<%=ladir%>"<%=iif(request("ligne")=fil.name
&"_"&laligne&"_"&ladir," selected","")%>><%=laligne%>, dir.
<%=ladir%></option>
    <%
    end
if
    end if
end if

```

```

next
next
next
%>
</select>
<input type=hidden
name=fichxmlpremier
value="data/<%=trim(request("jour"))%>/<%=lapremiereligne%>"
<%end if%>
</td>
</tr>
<tr>
<td width="60"><span lang="fr-
ca">Voyage:</span></td>
<td>
<%if request("ligne") <> ""
<select size="1" name="voyage"
<option
<%
'Va chercher la liste des
voyages dans la ligne-direction donnée
nomfichier=trim(left(request("ligne"),instr(1,request("ligne"),".")+3)
)
set f =
fs.GetFolder(Server.MapPath("data") & "\" & trim(request("jour")))
set fil = fs.getfile(f.path &
"\ & nomfichier)
objxml.Load fil.Path
lignedir=trim(mid(request("ligne"),instr(1,request("ligne"),".")+5))
laligne=trim(left(lignedir,instr(1,lignedir,"_")-1))
ladir=trim(mid(lignedir,instr(1,lignedir,"_")+1))
set
lesnoeuds=objxml.selectNodes("//root//LI[@NOLI='" & laligne & "'][@DIR='" &
ladir & "']//VOYS")
for t = 0 to
lesnoeuds.item(0).childnodes.length -1
%>
<option
value="<%=lesnoeuds.item(0).childnodes(t).attributes(0).value%>"
<%=iif(request("voyage")=lesnoeuds.item(0).childnodes(t).attributes(0).value,
" selected","")%><%=lesnoeuds.item(0).childnodes(t).attributes(0).value%> D:
<%=lesnoeuds.item(0).childnodes(t).attributes(1).value%></option>
<%
next %>
</select>

```

```

value="data/<%=trim(request("jour"))%>/<%=nomfichier%>">
<input type=hidden name=fichxml

value="<%=laligne%>">
<input type=hidden name=laligne

value="<%=ladir%>">
<input type=hidden name=ladir

<%=end if%>
</td>
</tr>
</table>
</td>
<td>

<%= if request("jour") <> "" then %>

<span lang="fr-ca">Direction Globale:</span>
<% 'Choix de la direction Globale
%>
<select size="1" name="dir_global"

onchange="form1.submit()">
<option value="">(choisir)</option>
<%
dir1 = "0"
dir2 = "1"
vardir1 = "Direction: 0"
vardir2 = "Direction: 1"
%>
<%=iif(request("dir_global")=dir1," selected","")%><%=vardir1%></option>
<%
%>
<%=iif(request("dir_global")=dir2," selected","")%><%=vardir2%></option>
<%
%>
</select><br>
<INPUT type="submit" value="Générer le
tableau global du <%=trim(request("jour"))%>" id=submit3 name=submit3 >
<p>

<%= if request("ligne") <> "" then %>
<INPUT type="submit" value="Générer le
tableau de la <%=laligne%> en direction <%=ladir%>" id=submit2 name=submit2 >
<INPUT type="submit" value="Générer le
tableau Espace Temps de la <%=laligne%> en direction <%=ladir%>" id=submit2
name=submit4 >

<p>

<%= if request("voyage")

<> "" then %>
<INPUT type="submit"
value="Générer le profil de charge du voyage <%=trim(request("voyage"))%>"
id=submit1 name=submit1>

<%=end if%>

```

```

                                <%end if%>

                                <%end if%>

                                </td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
        </form>
        <%'Bouton submit enfoncé
if request("submit1") <> "" then

'Fonctionne
Dim objXML, objXSL, xslTemplate, xslProc, strSVG
set objXML = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXML.async = false
objXML.load Server.MapPath(trim(request("fichxml")))

the_date = now()
set objXSL = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXSL.async = false
objXSL.load Server.MapPath("sto_charge_svg.xsl")

set xslTemplate = Server.CreateObject("MSXML2.XSLTemplate")
xslTemplate.stylesheet = objXSL
set xslProc = xslTemplate.createProcessor()
if request("ligne") <> "" then
xslProc.addParameter "lignel", trim(request("laligne"))
xslProc.addParameter "direction1", trim(request("ladir"))
xslProc.addParameter "voyage1", trim(request("voyage"))
xslProc.addParameter "ladategen", Cstr(the_date)
xslProc.addParameter "ladateres", Cstr(EXdateF)

end if

xslProc.input = objXML
xslProc.transform()

strSVG = xslProc.output

Response.Clear
Response.ContentType = "image/svg+xml"
Response.Write("<?xml version='1.0' encoding='iso-8859-1'?>" &
mid(strSVG,40))
Response.End

end if

if request("submit2") <> "" then

'Fonctionne
Dim objXML2, objXSL2, xslTemplate2, xslProc2, strXSL
set objXML2 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXML2.async = false

objXML2.load Server.MapPath(trim(request("fichxml")))

```

```

the_date = now()
set_objXSL2 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXSL2.async = false
objXSL2.load Server.MapPath("sto_table_ligne.xsl")

Response.Write("<?xml version='1.0' encoding='iso-8859-1'?>" &
mid(strXSL,105))

set xslTemplate2 = Server.CreateObject("MSXML2.XSLTemplate")
xslTemplate2.stylesheet = objXSL2
set xslProc2 = xslTemplate2.createProcessor()
if request("ligne") <> "" then
xslProc2.addParameter "ligne1", trim(request("laligne"))
xslProc2.addParameter "direction1", trim(request("ladir"))
xslProc2.addParameter "ladategen", Cstr(the_date)
xslProc2.addParameter "ladateres", Cstr(EXdateF)

end if

xslProc2.input = objXML2
xslProc2.transform()

strXSL = xslProc2.output

Response.Clear
Response.ContentType = "text/html"
Response.Write("<html xmlns:msxsl='urn:schemas-microsoft-com:xslt'
xmlns:Frank='urn:galileo-com:msxsl-jscript-utilities'><head><META http-
equiv='Content-Type' content='text/html; charset=iso-8859-1'>" &
mid(strXSL,184))
Response.End

end if

if request("submit3") <> "" then

'Fonctionne
Dim objXML3, objXSL3, xslTemplate3, xslProc3, strXSL2
set objXML3 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXML3.async = false

objXML3.load Server.MapPath(trim(request("fichxmlpremier")))

the_date2 = now()
set_objXSL3 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXSL3.async = false
objXSL3.load Server.MapPath("sto_table_global.xsl")

Response.Write("<?xml version='1.0' encoding='iso-8859-1'?>" &
mid(strXSL2,105))

set xslTemplate3 = Server.CreateObject("MSXML2.XSLTemplate")
xslTemplate3.stylesheet = objXSL3
set xslProc3 = xslTemplate3.createProcessor()

```

```

if request("jour") <> "" then

xslProc3.addParameter "direction1", Cstr(trim(request("dir_global")))
xslProc3.addParameter "ladategen", Cstr(the_date2)
xslProc3.addParameter "ladateres", Cstr(EXdateF)
xslProc3.addParameter "pathcourant", Cstr(f.path)

end if

xslProc3.input = objXML3
xslProc3.transform()

strXSL2 = xslProc3.output

Response.Clear
Response.ContentType = "text/html"
Response.Write("<html xmlns:msxsl='urn:schemas-microsoft-com:xslt'
xmlns:Frank='urn:galileo-com:msxsl-jscript-utilities'><head><META http-
equiv='Content-Type' content='text/html; charset=iso-8859-1'>" &
mid(strXSL2,184))
Response.End

end if

if request("submit4") <> "" then

'Fonctionne
Dim objXML4, objXSL4, xslTemplate4, xslProc4, strXSL4
set objXML4 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXML4.async = false

objXML4.load Server.MapPath(trim(request("fichxml")))

the_date = now()
set objXSL4 = Server.CreateObject("MSXML2.FreeThreadedDOMDocument")
objXSL4.async = false
objXSL4.load Server.MapPath("sto_table_ligne_espace_temps.xsl")

Response.Write("<?xml version='1.0' encoding='iso-8859-1'?>" &
mid(strXSL,105))

set xslTemplate4 = Server.CreateObject("MSXML2.XSLTemplate")
xslTemplate4.stylesheet = objXSL4
set xslProc4 = xslTemplate4.createProcessor()
if request("ligne") <> "" then
xslProc4.addParameter "ligne1", trim(request("laligne"))
xslProc4.addParameter "direction1", trim(request("ladir"))
xslProc4.addParameter "ladategen", Cstr(the_date)
xslProc4.addParameter "ladateres", Cstr(EXdateF)

end if

xslProc4.input = objXML4
xslProc4.transform()

strXSL4 = xslProc4.output

```

```

    Response.Clear
    Response.ContentType = "text/html"
    Response.Write("<html xmlns:msxsl='urn:schemas-microsoft-com:xslt'
xmlns:Frank='urn:galileo-com:msxsl-jscript-utilities'><head><META http-
equiv='Content-Type' content='text/html; charset=iso-8859-1'>" &
mid(strXSL4,184))
    Response.End

end if

%>

</body>
</html>

```

C.4 Code du template XSL «sto_table_global.xsl»

Ceci va permettre au lecteur de voir quelques exemples de code de transformation XSLT (note: les fonctions de script ont été retirées). En premier lieu on voit le code concernant le passage des paramètres provenant du choix de l'utilisateur dans la page ASP. Ensuite on voit la déclaration des variables, puis on commence à voir le mélange du code HTML de la mise en forme de la page ainsi que les transformations XSLT :

```

<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-
microsoft-com:xslt"
    xmlns:Frank="urn:galileo-com:msxsl-jscript-utilities" >

<!-- Parametres passés par ASP / -->

<xsl:param name="direction1" >allo</xsl:param>
<xsl:param name="ladategen" >allo</xsl:param>
<xsl:param name="ladateres" >allo</xsl:param>
<xsl:param name="pathcourant" >allo</xsl:param>

<xsl:variable name="NBLIGNES" select="Frank:returnNBLignes($pathcourant)" />

<xsl:variable name="direction">
    <xsl:value-of select="$direction1" />
</xsl:variable>

<xsl:variable name="dategen">
    <xsl:value-of select="$ladategen" />
</xsl:variable>

<xsl:variable name="dateres">
    <xsl:value-of select="$ladateres" />
</xsl:variable>

```

```

<!-- Variable contenant le node-set des voyages / -->

<msxsl:script language="JScript" implements-prefix="Frank">
...
</msxsl:script>

<xsl:template match="/"> <!-- Root Match, préparer la mise en forme des
tableaux / -->

    <html>

        <head>
            <title>
                Statistiques quotidiennes pour toutes les lignes en
direction: <xsl:value-of select="$direction"/>
            </title>
        </head>

        <body background="rgb(161,161,161)">
            </img>
            <h1 style="background-color:#446600; color:#FFFFFF; font-
size:18pt; text-align:center; letter-spacing: 4pt">
                Statistiques quotidiennes pour toutes les lignes en
direction: <xsl:value-of select="$direction"/> - pour: <xsl:value-of
select="$dateres"/>
            </h1>

            <table align="center" border="2px">
                <tr>
                    <th>LIGNE</th>
                    <th># Montées</th>
                    <th># Descentes</th>
                    <th># MAX des Charges MAX</th>
                    <th># Validations</th>
                    <th># Correspondances</th>
                    <th># Adultes</th>
                    <th># Étudiants</th>
                    <th># Aînés</th>
                </tr>

                <xsl:call-template name="TOTAUX">
                    <!-- Passe en paramètres la valeur de la
charge max et un compteur -->

                    <xsl:with-param name="i" select="0" />
                    <xsl:with-param name="TVAL" select="0"
/>
                    <xsl:with-param name="TCOOR"
select="0" />
                    <xsl:with-param name="TAD" select="0"
/>
                    <xsl:with-param name="TAE" select="0"
/>
                    <xsl:with-param name="TET" select="0"
/>
                    <xsl:with-param name="TAN" select="0"
/>

```



```

/>
select="0" />
select="0" />
<xsl:with-param name="TAC" select="0"
<xsl:with-param name="TMONT"
<xsl:with-param name="TDESC"
</xsl:call-template>
</table>
</body>
</html>
</xsl:template>
<xsl:template name="TOTAUX">
  <!-- Parametres -->
  <xsl:param name="i">0</xsl:param>
  <xsl:param name="TVAL">0</xsl:param>
  <xsl:param name="TCOOR">0</xsl:param>
  <xsl:param name="TAD">0</xsl:param>
  <xsl:param name="TAE">0</xsl:param>
  <xsl:param name="TET">0</xsl:param>
  <xsl:param name="TAN">0</xsl:param>
  <xsl:param name="TAC">0</xsl:param>
  <xsl:param name="TMONT">0</xsl:param>
  <xsl:param name="TDESC">0</xsl:param>
  <xsl:choose>
  <xsl:when test="$i &lt;= $NBLIGNES - 1">
    <xsl:variable name="VAL">
      <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'GV', $i)" />
      </xsl:variable>
      <xsl:variable name="COOR">
        <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'GC', $i)" />
        </xsl:variable>
        <xsl:variable name="AD">
          <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'GAD', $i)" />
          </xsl:variable>
          <xsl:variable name="AE">
            <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'AE', $i)" />
            </xsl:variable>
            <xsl:variable name="ET">
              <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'GED', $i)" />
              </xsl:variable>
              <xsl:variable name="AN">
                <xsl:value-of
select="Frank:retourvaleurUnique($pathcourant, $direction1, 'GAN', $i)" />
                </xsl:variable>
                <xsl:variable name="AC">

```

```

        <xsl:value-of
select="Frank:returnValueUnique($pathcourant, $direction1, 'AC', $i)" />
      </xsl:variable>
      <xsl:variable name="MONT">
        <xsl:value-of select="$VAL + $COOR" />
      </xsl:variable>
      <xsl:variable name="DESC">
        <xsl:value-of select="$MONT - $AC" />
      </xsl:variable>
      <xsl:variable name="LIGNE">
        <xsl:value-of select="Frank:returnLignes($pathcourant,
Si)" />
      </xsl:variable>

      <tr>
        <td align="center" style="font-weight:bold; background:
#CCCCCC"><xsl:value-of select="$LIGNE"/></td>
        <td align="center"><xsl:value-of select="$MONT"/></td>
        <td align="center"><xsl:value-of select="$DESC"/></td>
        <td align="center"><xsl:value-of select="$AE"/></td>
        <td align="center"><xsl:value-of select="$VAL"/></td>
        <td align="center"><xsl:value-of select="$COOR"/></td>
        <td align="center"><xsl:value-of select="$AD"/></td>
        <td align="center"><xsl:value-of select="$ET"/></td>
        <td align="center"><xsl:value-of select="$AN"/></td>

      </tr>

      <xsl:call-template name="TOTAUX">
        <xsl:with-param name="i" select="$i + 1" />
        <xsl:with-param name="TVAL" select="$TVAL + $VAL" />
        <xsl:with-param name="TCOOR" select="$TCOOR + $COOR" />
        <xsl:with-param name="TAD" select="$TAD + $AD" />
        <xsl:with-param name="TAE" select="$TAE + $AE" />
        <xsl:with-param name="TET" select="$TET + $ET" />
        <xsl:with-param name="TAN" select="$TAN + $AN" />
        <xsl:with-param name="TAC" select="$TAC + $AC" />
        <xsl:with-param name="TMONT" select="$TMONT + $MONT" />
        <xsl:with-param name="TDESC" select="$TDESC + $DESC" />

      </xsl:call-template>

    </xsl:when>
    <xsl:otherwise>
      <tr style="font-weight:bold">
        <td colspan="1" align="right">MOYENNE:</td>
        <td align="center"><xsl:value-of select="round($TMONT div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TDESC div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TAE div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TVAL div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TCOOR div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TAD div
$NBLIGNES)"/></td>

```

```

        <td align="center"><xsl:value-of select="round($TET div
$NBLIGNES)"/></td>
        <td align="center"><xsl:value-of select="round($TAN div
$NBLIGNES)"/></td>
    </tr>
    <tr style="font-weight:bold">
        <td colspan="1" align="right">SOMME:</td>
        <td align="center"><xsl:value-of select="$TMONT"/></td>
        <td align="center"><xsl:value-of select="$TDESC"/></td>
        <td align="center"><xsl:value-of select="NA"/></td>
        <td align="center"><xsl:value-of select="$TVAL"/></td>
        <td align="center"><xsl:value-of select="$TCOOR"/></td>
        <td align="center"><xsl:value-of select="$TAD"/></td>
        <td align="center"><xsl:value-of select="$TET"/></td>
        <td align="center"><xsl:value-of select="$TAN"/></td>
    </tr>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

C.5 Exemples de code de la page du graphique interactif SVG

Encore une fois, on ne présentera pas tout le code la page, car ce serait beaucoup trop long. Seulement quelques exemples seront présentés. Le premier bloc est le code du début jusqu'au bloc de fonctions ECMAScript qui prennent en charge les procédures événementielles. C'est aussi dans cette portion que l'on reçoit les paramètres passés par la page ASP. On voit ensuite la déclaration des variables nécessaires :

```

<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<!-- Parametres passés par ASP / -->

<xsl:param name="ligne1" />
<!-- La ligne sert juste de titre au graphique le parametre ligne est utilise
apriori pour ouvrir le bon fichier XML / -->
<xsl:param name="direction1" />
<xsl:param name="voyage1" />
<xsl:param name="ladategen" />

<xsl:variable name="voyage">
    <xsl:value-of select="$voyage1" />
</xsl:variable>
<xsl:variable name="ligne">
    <xsl:value-of select="$ligne1" />

```

```

</xsl:variable>
<xsl:variable name="direction">
  <xsl:value-of select="$direction1" />
</xsl:variable>

<xsl:variable name="dategen">
  <xsl:value-of select="$ladategen" />
</xsl:variable>

<xsl:param name="direction_count">
  <xsl:value-of select="$direction1" />
</xsl:param>

<!-- Variables globales utilisées par plusieurs templates / -->

<!-- Variable contenant le node-set des transactions a declarer avant le
match / -->

<xsl:variable name="ARRS"
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/TRS/TRSA" />

<!-- Heure de depart du voyage/ -->

<xsl:variable name="DEP"
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/@HD" />

<!-- Variable contenant le node-set arrêts de la ligne -GÉOMÉTRIE / -->
<xsl:variable name="ARRODR" select="root/LI[@DIR=$direction]/ARRL/ARR" />

<!-- Valeur du nombre d'arrêts a tracer -->
<xsl:variable name="STOP" select="count(root/LI[@DIR=$direction]/ARRL/ARR)"
/>

<!-- Dernier Ordre dans la direction -->
<xsl:variable name="LAST_ODR"
select="root/LI[@DIR=$direction]/ARRL/child::ARR[position()=last()]/@ODR" />

<!-- Variables pour les parametres de positionnement du graphique -->
<xsl:variable name="LARGE">
  <xsl:value-of select="((1024-40) div $STOP) div 2" />
</xsl:variable>

<xsl:variable name="XINC">
  <xsl:value-of select="(((1024-40) div $STOP)+2)" />
</xsl:variable>

<xsl:variable name="HAUT">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@AE" />
</xsl:variable>

<!-- Ajustement de charge et stats -->

<xsl:variable name="ACHARGE">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@AC" />

```

```

</xsl:variable>

<xsl:variable name="TOTVAL">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@GV" />
</xsl:variable>

<xsl:variable name="TOTCOOR">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@GC" />
</xsl:variable>

<xsl:variable name="TOTADULTE">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@GAD" />
</xsl:variable>

<xsl:variable name="TOTETUDIANT">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@GED" />
</xsl:variable>

<xsl:variable name="TOTVIEUX">
  <xsl:value-of
select="root/LI[@DIR=$direction]/VOYS/VOY[@IDV=$voyage]/META/@GAN" />
</xsl:variable>

<xsl:variable name="TOTAUTRES">
  <xsl:value-of select="(($TOTVAL + $TOTCOOR) - ($TOTADULTE +
$TOTETUDIANT + $TOTVIEUX))" />
</xsl:variable>

<!-- Variables pour l'echelle -->

<xsl:variable name="HAUT_AJUSTE">
  <xsl:choose>
    <xsl:when test="$HAUT=0" >
      <xsl:value-of select="0" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:choose>
        <xsl:when test="($HAUT)-15 >= 15" >
          <xsl:value-of select="15" />
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="($HAUT)-15" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<!-- Variables la grosseur des points et des barres -->

<xsl:variable name="HAUT_REL">

```

```

        <xsl:choose>
        <xsl:when test="$SHAUT=0" >
            <xsl:value-of select="20" />
        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
            <xsl:when test="($SHAUT)-15 &gt;= 6" >
                <xsl:value-of select="8" />
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="10 + (($SHAUT)-15)" />
            </xsl:otherwise>
            </xsl:choose>
        </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>

<xsl:template match="/"> <!-- Root Match, préparer les layers SVG / -->

<svg viewBox="0 0 1024 768" width="100%" height="100%"
xmlns:svg="http://www.w3.org/2000/svg" >
    <script type="text/ecmascript">
        <![CDATA[

```

On présente ici seulement quelques unes des fonctions scriptées, comme ici pour créer les graphiques en pointe de tartes utilisés pour la répartition des titres et des correspondances :

```

function initializeChart(vertexCount, totAD, totET, totAN, totAUTRES)
{
    initializeDataPoints(vertexCount, totAD, totET,
totAN, totAUTRES);
    for (var v=0; v<vertexCount; v++)
    {
        angles[v] = 360. * dataPoints[v]/dataSum;
    }
}

function initializeDataPoints(vertexCount, totAD, totET,
totAN, totAUTRES)
{
    var j = 0.;
    var totj = 0.;
    for (var v=0; v<4; v++)
    {
        if(sectionOn[v] == 1)
        {
            if(v == 0)
            {
                dataPoints[j] = totAD;
                qualitatif[j] = 1;
                totj++;
                j++;
            }
        }
    }
}

```

```

    }
    if(v == 1)
    {
        dataPoints[j] = totET;
        qualitatif[j] = 2;
        totj++;
        j++;
    }
    if(v == 2)
    {
        dataPoints[j] = totAN;
        qualitatif[j] = 3;
        totj++;
        j++;
    }
    if(v == 3)
    {
        dataPoints[j] = totAUTRES;
        qualitatif[j] = 4;
        totj++;
        j++;
    }
}

}

dataSum = 0;

for (var z=0; z<totj; z++)
{
    dataSum += dataPoints[z];
}

}

function drawChart(totAD, totET, totAN, totAUTRES)
{
    var totVOY = totAD + totET + totAN + totAUTRES;
    var actionOUT = "";
    var circleNode = null;

    if (ARRETNO == null)
    {
        actionOUT
        ="document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL "
        +totVOY+ " MONTÉES';";
        circleNode =
        svgDocument.createElement("circle");
        circleNode.setAttribute("cx", basePointX);
        circleNode.setAttribute("cy", basePointY);
        circleNode.setAttribute("r", (radius+4));
        circleNode.setAttribute("style",
        "fill:orange;");

        gcNode.appendChild(circleNode);
    }
    else
    {

```

```

                                actionOUT
="document.getElementById('evtText1').firstChild.nodeValue ='ARRET #'
+ARRETNO+ "-" +NOMARRET+ ": " +totVOY+ " MONTÉES';";
                                }
                                for (var v=0; v<vertexCount; v++)
                                {

                                if (angles[v] >180) //create 2 paths for
angle value/2
                                {
                                angleSum2 = angleSum1 + (angles[v]/2);
                                offsetX1 =
radius*Math.cos (angleSum1*Math.PI/180);
                                offsetY1 =
radius*Math.sin (angleSum1*Math.PI/180);
                                offsetX2 =
radius*Math.cos (angleSum2*Math.PI/180);
                                offsetY2 =
radius*Math.sin (angleSum2*Math.PI/180);

                                currentX = basePointX + offsetX2;
                                currentY = basePointY - offsetY2;

                                //the vertical offset must be
substracted,
                                //so we need to flip the sign of
offsetY1
                                offsetY1 *= -1;

                                pointPath =
                                pointPath += "
"M"+basePointX+", "+basePointY;
                                pointPath += "
l"+offsetX1+", "+offsetY1;
                                pointPath += " A"+radius+", "+radius+"
0 0 0 ";
                                pointPath += currentX+", "+currentY;
                                pointPath += "
L"+basePointX+", "+basePointY+"z";

                                if (ARRETNO == null)
                                {

                                if (((Rond((angles[v] *
dataSum/360),2))) == totAD && sectionOn[0] == 1 && qualitatif[v] == 1 )
                                {
                                action =
                                "document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL
TITRE: ADULTES " +Rond((totAD/totVOY)*100),2)+ " % DES " +totVOY+ "
MONTÉES';";
                                fillColor =
                                "fill:url(#"+circleColors[0]+")";
                                }
                                if (((Rond((angles[v] *
dataSum/360),2))) == totET && sectionOn[1] == 1 && qualitatif[v] == 2)
                                {

```



```

action =
"document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL
TITRE: ÉTUDIANTS " +Rond(((totET/totVOY)*100),2)+ " % DES " +totVOY+ "
MONTÉES';";
fillColor =
"fill:url("#+circleColors[1]+")";
}
if (((Rond((angles[v] *
dataSum/360),2))) == totAN && sectionOn[2] == 1 && qualitatif[v] == 3)
{
action =
"document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL
TITRE: AÎNÉS " +Rond(((totAN/totVOY)*100),2)+ " % DES " +totVOY+ "
MONTÉES';";
fillColor =
"fill:url("#+circleColors[2]+")";
}
if (((Rond((angles[v] *
dataSum/360),2))) == totAUTRES && sectionOn[3] == 1 && qualitatif[v] == 4)
{
action =
"document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL
AUTRES TITRES " +Rond(((totAUTRES/totVOY)*100),2)+ " % DES " +totVOY+ "
MONTÉES';";
fillColor =
"fill:url("#+circleColors[3]+")";
}
}
else
{
if (((Rond((angles[v] *
dataSum/360),2))) == totAD && sectionOn[0] == 1 && qualitatif[v] == 1)
{
action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #"
+ARRETNO+ "-" +NOMARRET+ "-" TITRE: ADULTES " +Rond(((totAD/totVOY)*100),2)+ "
% DES " +totVOY+ " MONTÉES';";
fillColor =
"fill:url("#+circleColors[0]+")";
}
}
if (((Rond((angles[v] *
dataSum/360),2))) == totET && sectionOn[1] == 1 && qualitatif[v] == 2)
{
action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #"
+ARRETNO+ "-" +NOMARRET+ "-" TITRE: ÉTUDIANTS " +Rond(((totET/totVOY)*100),2)+
" % DES " +totVOY+ " MONTÉES';";
fillColor =
"fill:url("#+circleColors[1]+")";
}
}
if (((Rond((angles[v] *
dataSum/360),2))) == totAN && sectionOn[2] == 1 && qualitatif[v] == 3)
{
action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #"
+ARRETNO+ "-" +NOMARRET+ "-" TITRE: AÎNÉS " +Rond(((totAN/totVOY)*100),2)+ " %
DES " +totVOY+ " MONTÉES';";
fillColor =
"fill:url("#+circleColors[2]+")";
}
}

```

```

    }
    if (((Rond((angles[v] *
dataSum/360),2))) == totAUTRES && sectionOn[3] == 1 && qualitatif[v] == 4)
    {
        action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #'
+ARRETNO+ "-" +NOMARRET+ ": AUTRES TITRES "
+Rond(((totAUTRES/totVOY)*100),2)+ " % DES " +totVOY+ " MONTÉES';";
        fillColor =
"fill:url("#+circleColors[3]+")";
    }
}

svgDocument.createElement("path");
fillColor);
action);
actionOUT);

pieNode =
pieNode.setAttribute("d", pointPath);
pieNode.setAttribute("style",
pieNode.setAttribute("onmousemove",
pieNode.setAttribute("onmouseout",

gcNode.appendChild(pieNode);
angleSum1 += (angles[v]/2);
angleSum2 = angleSum1 + (angles[v]/2);

offsetX1 =
radius*Math.cos (angleSum1*Math.PI/180);
offsetY1 =
radius*Math.sin (angleSum1*Math.PI/180);

offsetX2 =
radius*Math.cos (angleSum2*Math.PI/180);
offsetY2 =
radius*Math.sin (angleSum2*Math.PI/180);

currentX = basePointX + offsetX2;
currentY = basePointY - offsetY2;

//the vertiacal offset must be
//so we need to flip the sign of
offsetY1 *= -1;

pointPath =
pointPath += "
pointPath += " A"+radius+", "+radius+"
pointPath += currentX+", "+currentY;
pointPath += "

L"+basePointX+", "+basePointY+"z";

```

```

svgDocument.createElement("path");
fillColor);
action);
actionOUT);

    pieNode =
    pieNode.setAttribute("d", pointPath);
    pieNode.setAttribute("style",
    pieNode.setAttribute("onmousemove",
    pieNode.setAttribute("onmouseout",

    gcNode.appendChild(pieNode);
    angleSum1 += (angles[v]/2);

    }
    else
    {
        angleSum2 = angleSum1 + angles[v];

        offsetX1 =
        radius*Math.cos(angleSum1*Math.PI/180);
        offsetY1 =
        radius*Math.sin(angleSum1*Math.PI/180);

        offsetX2 =
        radius*Math.cos(angleSum2*Math.PI/180);
        offsetY2 =
        radius*Math.sin(angleSum2*Math.PI/180);

        currentX = basePointX + offsetX2;
        currentY = basePointY - offsetY2;

        //the vertical offset must be
        //so we need to flip the sign of

        offsetY1 *= -1;

        pointPath =
        pointPath += "
        pointPath += " A"+radius+", "+radius+"
        pointPath += currentX+", "+currentY;
        pointPath += "

        if (ARRETNO == null)
        {

            if (((Rond((angles[v] *
            dataSum/360),2))) == totAD && sectionOn[0] == 1 && qualitatif[v] == 1 )
            {
                action =
                "document.getElementById('evtText1').firstChild.nodeValue ='VOYAGE GLOBAL

```



```

        if ((Rond((angles[v] *
dataSum/360),2))) == totAN && sectionOn[2] == 1 && qualitatif[v] == 3)
        {
            action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #"
+ARRETNO+ "-" +NOMARRET+ "- TITRE: AÎNÉS " +Rond(((totAN/totVOY)*100),2)+ " %
DES " +totVOY+ " MONTÉES';";
            fillColor =
"fill:url("#+circleColors[2]+")";
        }
        if ((Rond((angles[v] *
dataSum/360),2))) == totAUTRES && sectionOn[3] == 1 && qualitatif[v] == 4)
        {
            action =
"document.getElementById('evtText1').firstChild.nodeValue ='ARRET #"
+ARRETNO+ "-" +NOMARRET+ ": AUTRES TITRES "
+Rond(((totAUTRES/totVOY)*100),2)+ " % DES " +totVOY+ " MONTÉES';";
            fillColor =
"fill:url("#+circleColors[3]+")";
        }
    }

    pieNode =
svgDocument.createElement("path");
    pieNode.setAttribute("d", pointPath);
    pieNode.setAttribute("style",
fillColor);
    pieNode.setAttribute("onmousemove",
action);
    pieNode.setAttribute("onmouseout",
actionOUT);
    gcNode.appendChild(pieNode);
    angleSum1 += angles[v];
}
}
}

```

On continue maintenant avec le code des *layers* principaux de la page et le code de l'échelle interactive:

```

g id="MainLayer">
  <g id="BackGround" transform="translate(20,190)">
    <!--Graph Background -->
    <rect x="20" y="{-170}" width="1150" height="{360}"
style="fill:rgb(250,250,251);" stroke="black" />
    <g id="TitreL" transform="translate(20,190)">
    ...
    <!--Échelle parametree si necessaire -->
    <xsl:choose>

```

```

        <xsl:when test="$HAUT_AJUSTE &lt;= 5" >
            <xsl:choose>
                <xsl:when test="$HAUT_AJUSTE &gt; 0" >
                    <text x="-15" y="{-1 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        1
                    </text>
                    <line x1="-5" y1="{-1 * $HAUT_REL}"
x2="5" y2="{-1 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                    <text x="-15" y="{-5 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        5
                    </text>
                    <line x1="-5" y1="{-5 * $HAUT_REL}"
x2="5" y2="{-5 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                    <text x="-20" y="{-10 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        10
                    </text>
                    <line x1="-5" y1="{-10 * $HAUT_REL}"
x2="5" y2="{-10 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                    <text x="-20" y="{-15 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        15
                    </text>
                    <line x1="-5" y1="{-15 * $HAUT_REL}"
x2="5" y2="{-15 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                    <text x="-20" y="{-20 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        20
                    </text>
                    <line x1="-5" y1="{-20 * $HAUT_REL}"
x2="5" y2="{-20 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                </xsl:when>
                <xsl:otherwise>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:when>
            <xsl:otherwise>
                <xsl:choose>
                    <xsl:when test="$HAUT_AJUSTE &lt;= 10" >
                        <xsl:choose>
                            <xsl:when test="$HAUT_AJUSTE &gt; 0" >
                                <text x="-15" y="{-1 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    1
                                </text>
                                <line x1="-5" y1="{-1 * $HAUT_REL}"
x2="5" y2="{-1 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                <text x="-15" y="{-5 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    5
                                </text>
                            </xsl:when>
                        </xsl:choose>
                    </xsl:when>
                </xsl:choose>
            </xsl:otherwise>
        </xsl:when>

```

```

                                <line x1="-5" y1="{-5 * $HAUT_REL}"
x2="5" y2="{-5 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                <text x="-20" y="{-10 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    10
                                </text>
                                <line x1="-5" y1="{-10 * $HAUT_REL}"
x2="5" y2="{-10 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                <text x="-20" y="{-15 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    15
                                </text>
                                <line x1="-5" y1="{-15 * $HAUT_REL}"
x2="5" y2="{-15 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                                <text x="-20" y="{-20 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    20
                                </text>
                                <line x1="-5" y1="{-20 * $HAUT_REL}"
x2="5" y2="{-20 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                                <text x="-20" y="{-25 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    25
                                </text>
                                <line x1="-5" y1="{-25 * $HAUT_REL}"
x2="5" y2="{-25 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                </xsl:when>
                                <xsl:otherwise>
                                </xsl:otherwise>
                                </xsl:choose>

                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:choose>
                                <xsl:when test="$HAUT_AJUSTE > 0" >
                                <text x="-15" y="{-1 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    1
                                </text>
                                <line x1="-5" y1="{-1 * $HAUT_REL}"
x2="5" y2="{-1 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                <text x="-15" y="{-5 * $HAUT_REL + 5}"
style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    5
                                </text>
                                </text>
                                <line x1="-5" y1="{-5 * $HAUT_REL}"
x2="5" y2="{-5 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                                <text x="-20" y="{-10 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                                    10
                                </text>
                                </text>
                                <line x1="-5" y1="{-10 * $HAUT_REL}"
x2="5" y2="{-10 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

```

```

                    <text x="-20" y="{-15 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        15
                    </text>
                    <line x1="-5" y1="{-15 * $HAUT_REL}"
x2="5" y2="{-15 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                    <text x="-20" y="{-20 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        20
                    </text>
                    <line x1="-5" y1="{-20 * $HAUT_REL}"
x2="5" y2="{-20 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                    <text x="-20" y="{-25 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        25
                    </text>
                    <line x1="-5" y1="{-25 * $HAUT_REL}"
x2="5" y2="{-25 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />
                    <text x="-20" y="{-30 * $HAUT_REL +
5}" style="font-family:arial;font-size:7pt;text-anchor:left;" fill="blue">
                        30
                    </text>
                    <line x1="-5" y1="{-30 * $HAUT_REL}"
x2="5" y2="{-30 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-width:1" />

                    </xsl:when>
                    <xsl:otherwise>
                    </xsl:otherwise>
                    </xsl:choose>

                    </xsl:otherwise>
                    </xsl:choose>

                </xsl:otherwise>
                </xsl:choose>

                <xsl:choose>
                <xsl:when test="$HAUT_AJUSTE = 0" >
                    <!--Échelle standard -->
                    <text x="-15" y="{-1 *
$HAUT_REL + 5}" style="font-family:arial;font-size:7pt;text-anchor:left;"
fill="blue">
                        1
                    </text>
                    <line x1="-5" y1="{-1 *
$HAUT_REL}" x2="5" y2="{-1 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-
width:1" />

                    <text x="-15" y="{-5 *
$HAUT_REL + 5}" style="font-family:arial;font-size:7pt;text-anchor:left;"
fill="blue">
                        5
                    </text>
                    <line x1="-5" y1="{-5 *
$HAUT_REL}" x2="5" y2="{-5 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-
width:1" />

```



```

<text x="-20" y="{-10 *
$HAUT_REL + 5}" style="font-family:arial;font-size:7pt;text-anchor:left;"
fill="blue">
10
</text>
<line x1="-5" y1="{-10 *
$HAUT_REL}" x2="5" y2="{-10 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-
width:1" />

<text x="-20" y="{-15 *
$HAUT_REL + 5}" style="font-family:arial;font-size:7pt;text-anchor:left;"
fill="blue">
15
</text>
<line x1="-5" y1="{-15 *
$HAUT_REL}" x2="5" y2="{-15 * $HAUT_REL}" style="stroke:rgb(99,99,99);stroke-
width:1" />
</xsl:when>
<xsl:otherwise>
</xsl:otherwise>
</xsl:choose>

<!--Titre du graphique et Legende -->

<text x="537" y="{90 - (15 * 10)}" style="font-
family:arial;font-size:14pt;text-anchor:middle;baseline-shift:268,5;"
fill="blue">
Charge Ligne: <tspan font-weight="bold"
font-size="14pt" fill="purple" ><xsl:value-of select="$ligne" /></tspan> -
Direction: <tspan font-weight="bold" font-size="14pt" fill="purple"
><xsl:value-of select="$direction" /></tspan> - VOYAGE: <tspan font-
weight="bold" font-size="14pt" fill="purple" ><xsl:value-of select="$voyage"
/></tspan>
- Départ: <tspan font-weight="bold" font-
size="14pt" fill="purple" ><xsl:value-of select="$DEP" /></tspan>
</text>
<text x="66" y="{70 - (15 * 10)}" style="font-
family:Arial;font-weight:bold;font-size:8pt;text-anchor:middle;baseline-
shift:268,5;" fill="black">
Statistiques<tspan fill="purple" >
GLOBALES</tspan>:
</text>
<circle cx="135.5" cy="{-192 - (15 * 10)}"
id="rond_titres" r="4" fill="white" stroke="orange"
onclick="document.getElementById('PIECHARTARRET').getStyle().setProperty
('visibility', 'hidden'); cleanLayer(evt, 'PIECHARTGLOBAL'); init(evt,
'PIECHARTGLOBAL', {$TOTADULTE}, {$TOTETUDIANT}, {$TOTVIEUX}, {$TOTAUTRES});
document.getElementById('PIECHARTGLOBAL').getStyle().setProperty
('visibility', 'visible');modifRond(evt, 'rond_titres');"/>
<text x="165.5" y="{80 - (15 * 10)}" style="font-
family:Arial;font-weight:bold;font-size:9pt;text-anchor:middle;baseline-
shift:268,5;" fill="orange">
-TITRES
</text>
<circle cx="135.5" cy="{-202 - (15 * 10)}"
id="rond_coor" r="4" fill="white" stroke="blue"

```

```

onclick="document.getElementById('PIECHARTARRET').getStyle().setProperty
('visibility', 'hidden'); cleanLayer(evt, 'PIECHARTGLOBAL'); initCoor(evt,
'PIECHARTGLOBAL', {$TOTCOOR}, {$TOTVAL});
document.getElementById('PIECHARTGLOBAL').getStyle().setProperty
('visibility', 'visible'); modifRond(evt, 'rond_coor');"/>
      <text x="208.2" y="{70 - (15 * 10)}" style="font-
family:Arial;font-weight:bold;font-size:9pt;text-anchor:middle;baseline-
shift:268,5;" fill="blue">
          -CORRESPONDANCES
      </text>
      <rect x="950" y="{-190 - (15 * 10)}" height="20"
width="25" fill="url(#LEGENDM)" stroke="black"/>
      <text x="1034" y="{92.5 - (15 * 10)}" style="font-
family:Arial;font-size:11pt;text-anchor:middle;baseline-shift:268,5;"
fill="red">
          Montées
      </text>
      <rect x="950" y="{-160 - (15 * 10)}" height="20"
width="25" fill="url(#LEGENDED)" stroke="black" onclick="cacherDescentes(evt,
'BARS');"/>
      <text x="1034" y="{122.5 - (15 * 10)}" style="font-
family:arial;font-size:11pt;text-anchor:middle;baseline-shift:268,5;"
fill="red">
          Descentes
      </text>
      <circle cx="962.5" cy="{-121 - (15 * 10)}"
id="rond_charge" r="8" fill="url(#CHARGE)" stroke="black"
onclick="cacherCharge(evt); modifRond(evt, 'rond_charge');" />
      <text x="1034" y="{153.5 - (15 * 10)}" style="font-
family:arial;font-size:11pt;text-anchor:middle;baseline-shift:268,5;"
fill="red">
          Charge
      </text>

```

```

      <text x="962.5" y="{136.5 - (15 * 10)}"
style="font-family:arial;font-size:7pt;text-anchor:middle;baseline-
shift:268,5;" fill="black">
          (click On/Off)
      </text>
      <text x="-15" y="-170" style="font-
family:arial;font-size:8pt;text-anchor:middle;font-weight:bold;"
fill="orange" transform="rotate(90) translate(-170,205)">
          sivt-STO Généré le: <tspan font-weight="bold" font-
size="8pt" fill="purple" ><xsl:value-of select="$dategen" /></tspan>
      </text>
      <!--Test -->
      </g>
      <g id="INFO" transform="translate(20,190)">
          <text x="537" y="{122.5 - (15 * 10)}" id="evtText1"
style="font-family:arial;font-size:11pt;text-anchor:middle;baseline-
shift:268,5;" fill="red">INFOS ARRÊT:</text>

```

```

</g>
<g id="PIECHARTGLOBAL" transform="translate(20,190)">
</g>
<g id="PIECHARTARRET" transform="translate(20,190)">
</g>

```

Enfin on peut présenter le code de création des ronds de points de charge qu'on voit ici avec le mot clé *circle* :

```

<!-- Points de charge a bord -->
  <xsl:choose>
    <xsl:when test="not(string($CHARGE))" >
      <circle cx="{(($ORDC * $XINC)+$LARGE)}" cy="0"
r="4.8" style="fill:rgb(200,200,200);" stroke="black" />
    </xsl:when>
    <xsl:otherwise>
      <!-- Ne pas tracer de lignes quand c'est le premier
arret -->

      <xsl:choose>
        <xsl:when test="$ORDC=0" >
          <circle cx="{(($ORDC * $XINC)+$LARGE)}"
cy="-{($CHARGE * $HAUT_REL)}" r="4.8" style="fill:url(#Points);"
stroke="black"
onmouseover="document.getElementById('{ $ORDC }').getStyle().setProperty('font-
size', '10pt');
document.getElementById('{ $ORDC }').getStyle().setProperty('fill', 'red');
document.getElementById('evtText1').firstChild.nodeValue = 'CHARGE ARRET #
{ $ODR_NOA_LABEL}: { $CHARGE}';"

          onclick="document.getElementById('PIECHARTGLOBAL').getStyle().setPrope
rty ('visibility', 'hidden'); initCoor(evt, 'PIECHARTARRET', { $ARR_COOR_OK},
{ $ARR_VAL_OK}, { $ODR_NOA_LABEL}, '{ $ODR_NOA_NOM}'); eteintRond(evt);"

          onmouseout="document.getElementById('{ $ORDC }').getStyle().setProperty(
'font-size', '7pt');
document.getElementById('{ $ORDC }').getStyle().setProperty('fill',
'black');"/>
        </xsl:when>
        <xsl:otherwise >
          <circle cx="{(($ORDC * $XINC)+$LARGE)}"
cy="-{($CHARGE * $HAUT_REL)}" r="4.8" style="fill:url(#Points);"
stroke="black"
onmouseover="document.getElementById('{ $ORDC }').getStyle().setProperty('font-
size', '10pt');
document.getElementById('{ $ORDC }').getStyle().setProperty('fill', 'red');
document.getElementById('evtText1').firstChild.nodeValue = 'CHARGE ARRET #
{ $ODR_NOA_LABEL}: { $CHARGE}';"

          onclick="document.getElementById('PIECHARTGLOBAL').getStyle().setPrope
rty ('visibility', 'hidden'); initCoor(evt, 'PIECHARTARRET', { $ARR_COOR_OK},
{ $ARR_VAL_OK}, { $ODR_NOA_LABEL}, '{ $ODR_NOA_NOM}'); eteintRond(evt);"

          onmouseout="document.getElementById('{ $ORDC }').getStyle().setProperty(
'font-size', '7pt');

```

```

document.getElementById('{ $ORDC }').getStyle().setProperty('fill',
'black');"/>
                                </xsl:otherwise>
                                </xsl:choose>

                                <xsl:choose>
                                <xsl:when test="$ORDC=$LAST_ODR" >
                                    <circle cx="{(($ORDC * $XINC)+$LARGE)}"
cy="0" r="4.8" fill="yellow" stroke="black"

                                onmouseover="document.getElementById('evtText1').firstChild.nodeValue
='AJUSTEMENT DE CHARGE : {$ACHARGE} - ' + Rond(((( $ACHARGE div ($TOTVAL +
$TOTCOOR)) * 100},2) + '% des montées';
document.getElementById('LAST_LIGNE').getStyle().setProperty ('visibility',
'visible'); document.getElementById('LAST_CALCUL').getStyle().setProperty
('visibility', 'hidden');"
                                onmouseout="document.getElementById('evtText1').firstChild.nodeValue
='INFOS ARRÊT: '; document.getElementById('LAST_LIGNE').getStyle().setProperty
('visibility', 'hidden');
document.getElementById('LAST_CALCUL').getStyle().setProperty ('visibility',
'visible');"/>
                                </xsl:when>
                                <xsl:otherwise>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
</xsl:otherwise>
</xsl:choose>

```

C.6 Exemple de l'application de PCENT sur le graphique interactif SVG

Voici ce que l'on voit si l'on active cette fonction lorsque l'on a un facteur PCENT supérieur à 1 :

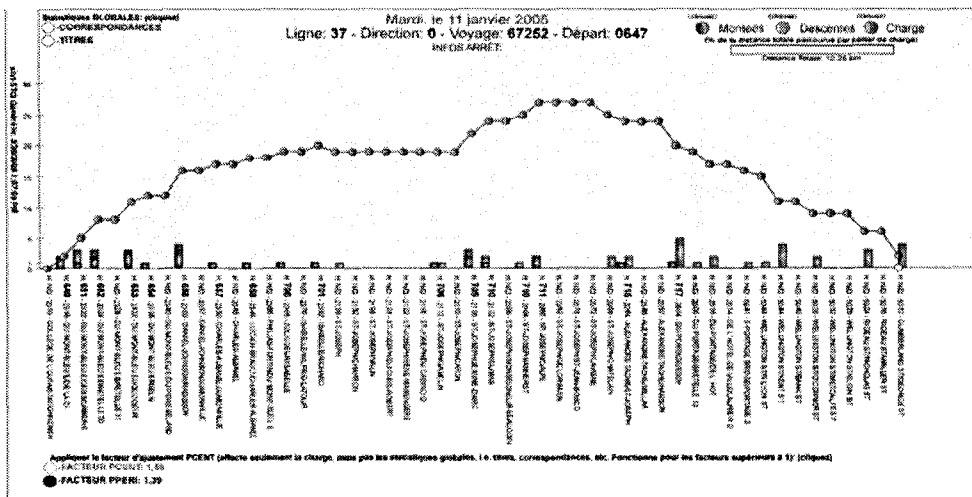


Figure C-2 Affichage du graphique sans utiliser le facteur d'ajustement PCENT.

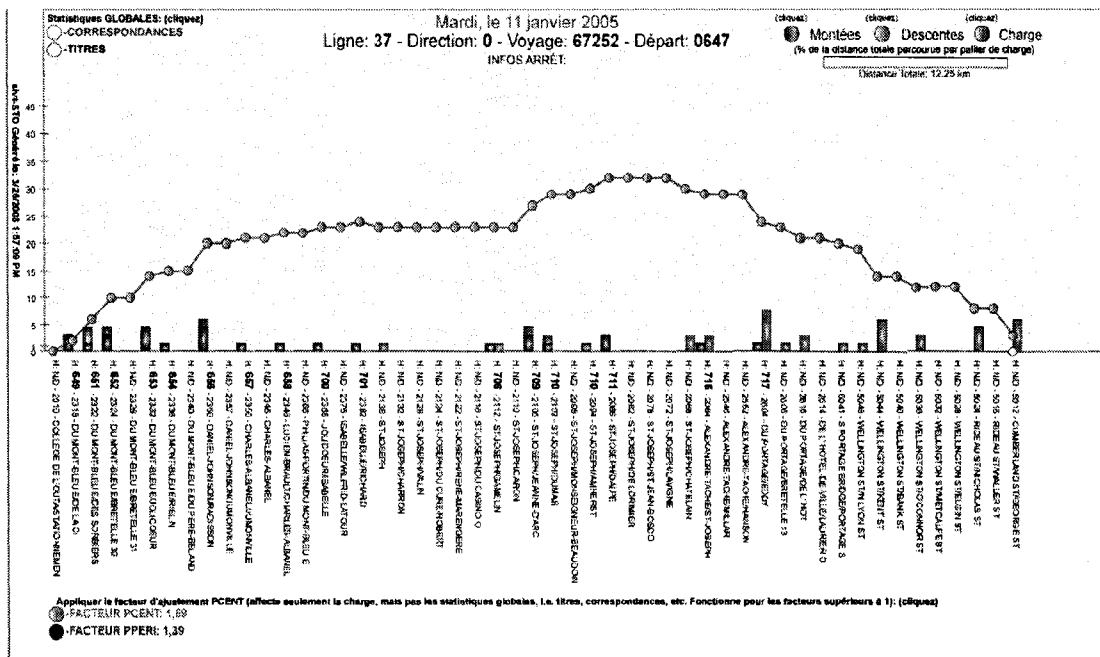


Figure C-3 Affichage du graphique après l'application du facteur d'ajustement PCENT.

Ici, on voit bien qu'après l'ajustement du facteur PCENT de 1,59, que l'échelle interactive a été modifiée pour tenir compte du facteur et que les points de charge à bord se sont mis en rouge pour indiquer que les valeurs originales sont modifiées en fonction du facteur PCENT.