

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Elaboration d'un outil générique pour la visualisation de séries temporelles sur PDA

Genicot, S.

Award date:
2005

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES
NOTRE-DAME DE LA PAIX, NAMUR
Institut d'Informatique
Année académique 2004-2005

Interfaces Homme - Machine

-

**Elaboration d'un outil générique pour la
visualisation de séries temporelles sur PDA**

Stéphane Genicot

Mémoire présenté en vue de l'obtention du grade de Maître en Informatique

Résumé

Le domaine des séries temporelles est très vaste. Il englobe une gamme de catégories toutes aussi différentes les unes des autres mais dont l'objectif final reste le même, à savoir : l'analyse et la prédiction de tendances à partir de ces séries. Actuellement, un grand nombre des applications permettant de réaliser ces analyses et prédictions sont développées pour les ordinateurs classiques et il existe déjà un certain nombre de visualisations pour PDA et plus particulièrement dans ce domaine. Cependant, les PDAs souffrent de limitations, notamment au niveau de la représentation d'un grand volume d'informations ainsi qu'au niveau de l'interaction avec les utilisateurs. Une amélioration des interfaces permettrait d'augmenter l'utilisabilité de ces applications et surtout leur intérêt.

Faisant suite à un travail déjà réalisé dans ce domaine, l'objectif de ce mémoire est de présenter une possibilité d'implémentation d'un outil générique pour la visualisation de séries temporelles sur PDA ainsi que des moyens d'interaction avec les utilisateurs.

Mots-clés : Séries temporelles, visualisations, interactions, couleurs, PDA, mobiles, « .NET ».

Abstract

The application field of time series is very large. It includes a series of practices all different from the others but whose final objectives are still the analysis and trends prediction starting from those series. Currently, a great number of the applications, making those analysis and predictions possible, are being developed for classic computers but some visualizations especially conceived for PDA already exist and more particularly in the mobile field. However, PDAs do suffer from limitations, especially when displaying a large amount of information and dealing with user's interaction. An improvement of the interface would enable an increment of the applications' usability and principally their interest.

Based on a work already achieved in this field, this master's dissertation suggests a generic tool's implementation on a PDA, especially for time series visualization and user's interactions.

Keywords: Time series, visualizations, interactions, colours, PDA, mobiles, ".NET".

Remerciements :

Ce mémoire a demandé un investissement personnel important et n'aurait pas pu être réalisé sans l'aide de nombreuses personnes.

Je tiens à remercier le Professeur Monique Noirhomme pour avoir accepté d'être le promoteur de ce mémoire et pour l'aide apportée lors de la rédaction de ce mémoire par ses lectures critiques et constructives.

Je voudrais également remercier mon superviseur à Udine, le Professeur Luca Chittaro, pour l'accueil réservé ainsi que pour toute l'aide apportée lors de ces trois mois de stage en Italie et plus particulièrement pour avoir orienté mon travail pendant cette période.

Je voudrais aussi remercier tous les membres du laboratoire HCI à Udine (Demis, Augusto, Stefano, Lucio, Luca) pour l'accueil qu'ils m'ont réservé, les conseils judicieux qu'ils m'ont prodigués, leur disponibilité durant la réalisation de mon stage et l'ambiance chaleureuse dans laquelle ils m'ont permis de travailler.

Merci aussi à Frédéric Randolet pour m'avoir permis d'accéder au code source de son application, ce qui m'a fortement aidé dans la compréhension de son système et à y apporter des modifications.

Merci à Etienne Cuvelier pour m'avoir aidé à analyser les résultats de mes questionnaires d'évaluation.

Et pour terminer, je tiens à remercier tous les membres de ma famille, et plus particulièrement ma mère, pour m'avoir aidé, encouragé (supporté !) durant toute la rédaction de ce mémoire ainsi que lors des relectures.

Table des matières

Table des matières	i
Liste des tableaux	v
Liste des figures	vi
0 Introduction	1
1 Définitions	3
1.1 Acronymes	3
1.2 Lexique	4
2 Etat de l'art	7
2.1 Introduction.....	7
2.2 L'information	8
2.3 Visualisation de l'information.....	8
2.3.1 Qu'est-ce que la visualisation ?.....	8
2.3.2 Qu'est-ce que la visualisation de l'information ?.....	9
2.3.3 Principes essentiels de la visualisation	10
2.3.4 Typologies des représentations.....	11
2.4 La notion de couleur.....	13
2.4.1 Classification des couleurs.....	13
2.4.2 Les sept contrastes de la couleur	15
2.4.3 Les représentations de la couleur en informatique	15
2.5 Conception des interfaces.....	17
2.5.1 Principes ergonomiques.....	18
2.5.2 Règles propres aux PDA	20
2.6 Applications existantes.....	21
2.6.1 Applications existantes sur le Web	22
2.6.2 Applications existantes sur PDA	24
2.7 Conclusion	26
3 Qu'est-ce qu'une série temporelle ?	27
3.1 Définitions	27
3.1.1 Quelques définitions issues d'Internet.....	27
3.1.2 Définitions mathématiques	27
3.1.3 Caractéristiques.....	29
3.2 Quels sont les buts d'une série temporelle ?.....	29
3.2.1 Prévoir.....	30
3.2.2 Relier les variables	30
3.2.3 Déterminer la causalité.....	31
3.2.4 Distinguer entre court et long terme	31
3.2.5 Etudier des anticipations des agents.....	31
3.2.6 Repérer les tendances et cycles	31
3.2.7 Corriger des variations saisonnières.....	31
3.2.8 Détecter les chocs structurels.....	32
3.2.9 Contrôler les processus	32
3.3 Que pouvons-nous retirer des séries temporelles ?.....	32
3.3.1 Analyse des tendances	32

3.3.2	Analyses de la saisonnalité	33
3.4	Exemples	33
4	Analyse des besoins	36
4.1	Introduction	36
4.1.1	Qu'est-ce qu'une analyse des besoins ?	37
4.1.2	Pourquoi une analyse des besoins ?.....	37
4.2	Les utilisateurs	38
4.2.1	Types d'utilisateur.....	38
4.2.2	Caractéristiques de l'utilisateur	38
4.3	Environnement de travail	41
4.3.1	Localisation du produit.....	41
4.3.2	Position.....	42
4.3.3	Matériel	42
4.3.4	Logiciels et système d'exploitation.....	42
4.4	Tâches.....	42
4.4.1	Création d'un profil	42
4.4.2	Modification d'un profil	43
4.4.3	Suppression d'un profil	44
4.4.4	Exploration des fichiers	45
4.4.5	Lancement de la partie visualisation (et réaliser une analyse).....	46
4.4.6	Configuration de l'application.....	47
4.4.7	Interaction avec une visualisation.....	47
4.4.8	Affichage du statut de l'utilisateur	48
4.4.9	Consultation de l'aide	49
5	Applications sur PDA	50
5.1	Introduction.....	50
5.1.1	Utilité du PDA	50
5.1.2	Système d'exploitation.....	51
5.1.3	Caractéristiques techniques.....	51
5.2	Le PDA utilisé pour l'application	52
5.3	Les limites rencontrées.....	52
5.3.1	Au niveau du processeur	52
5.3.2	Au niveau de l'écran	53
5.3.3	Au niveau de la mémoire.....	53
5.3.4	Au niveau de l'interaction	54
5.4	Les solutions apportées.....	55
5.4.1	Au niveau de l'écran.	55
5.4.2	Au niveau de la mémoire.....	55
5.4.3	Au niveau de l'interaction.	55
5.5	Outils de développements sur PDA.....	56
5.5.1	La plateforme de développement « Microsoft .NET »	56
5.5.2	L'architecture de la plateforme « Microsoft .NET ».....	57
5.5.3	Le Compact Framework(1.1) de Microsoft.....	59
5.5.4	Les classes de la librairie objet.....	60
5.5.5	La plateforme de développement J2ME.....	60
5.5.6	L'architecture de la plateforme J2ME	61
6	Outil générique de visualisation sur PDA	63

6.1	Visualisations principales	63
6.1.1	Brick Wall Chart	63
6.1.2	Stacked Bar Chart	65
6.1.3	Bar Chart	66
6.2	Visualisations secondaires	66
6.2.1	Mountain	67
6.2.2	Multi-Lines.....	68
6.2.3	DashBoard	70
6.3	Interactions avec les utilisateurs	71
6.4	Relations entre les visualisations.....	73
6.4.1	Cas 1 : Passage complet.....	73
6.4.2	Cas 2 : Passage partiel	74
6.4.3	Cas 3 : Passage avec ajout de paramètres	74
6.4.4	Cas 4 : Passage avec suppression de paramètres	75
7	Implémentation	76
7.1	Choix de l'outil de développement	76
7.2	Choix du langage de programmation.....	76
7.2.1	Il existe de nombreux langages, lequel choisir ?.....	77
7.2.2	Common Language Specification	77
7.2.3	Le langage C#	77
7.3	Détails techniques de l'implémentation	78
7.3.1	Chargement et sauvegarde des données.....	78
7.3.2	Classes principales de l'application	81
7.3.3	Codage des couleurs.....	82
7.4	Limites et améliorations possibles.....	83
7.4.1	Au niveau du paramétrage de l'application	83
7.4.2	Au niveau du son	83
8	Evaluation de l'application	86
8.1	Objectifs	86
8.2	Echantillon d'utilisateurs.....	86
8.3	Présentation du questionnaire	87
8.3.1	Informations personnelles	87
8.3.2	Expériences précédentes dans les services mobiles	87
8.3.3	Scénarii	88
8.3.4	Evaluation générale de l'interface.....	90
8.3.5	Evaluation des visualisations	90
8.3.6	Visualisation préférée	91
8.3.7	Commentaires généraux	91
8.4	Résultats et analyses	91
8.4.1	Test de Mc Nemar	91
8.4.2	Tests ANOVA : Analyse de la variance à deux facteurs.....	94
8.4.3	Test de la médiane	98
8.5	Remarques et améliorations	100
8.5.1	Remarques négatives.....	100
8.5.2	Remarques positives.....	100
8.6	Conclusion	101
9	Conclusion	102

10 Bibliographie.....	104
Annexes A : Manuel d'utilisation.....	106
A.1 Liste des fichiers	106
A.2 Démarrage de l'application	108
A.3 Manipulation des profils.....	110
A.4 Lancer la partie visualisation.....	111
A.5 Configurer l'application	114
A.6 Créer une visualisation.....	115
A.7 Interaction avec une visualisation	118
A.8 Autres fonctionnalités	119
Annexes B : Tables.....	123
B.1 Table de Chi carré (χ^2)	123
B.2 Table de Fisher	124
B.3 Tableaux de calculs des tests de l'ANOVA à deux facteurs.....	125
B.4 Tableaux de calculs des tests de la médiane.....	127

Liste des tableaux

Tableau 1-1 : Acronymes utilisés dans la suite du mémoire.....	3
Tableau 3-1 : Exemples de séries temporelles (Criminalité).....	35
Tableau 3-2 : Exemples de séries temporelles (Démographie).....	35
Tableau 3-3 : Exemples de séries temporelles (Finance).....	35
Tableau 3-4 : Exemples de séries temporelles (Météorologie).....	35
Tableau 3-5 : Exemples de séries temporelles (Production).....	35
Tableau 5-1: Caractéristiques du PDA utilisé pour l'application.....	52
Tableau 5-2 : Langages de programmation supportés par ".NET".....	56
Tableau 5-3 : Classes du Framework ".NET".....	60
Tableau 6-1 : Caractéristiques des visualisations.....	73
Tableau 8-1 : Informations personnelles.....	87
Tableau 8-2 : Expériences précédentes dans les graphismes et les services mobiles.....	88
Tableau 8-3 : Résultats des scénarii.....	90
Tableau 8-4 : Evaluation générale de l'interface.....	90
Tableau 8-5 : Evaluation des visualisations.....	91
Tableau 8-6 : Visualisation préférée.....	91
Tableau 8-7 : Valeurs de Fisher.....	97

Liste des figures

Figure 2-1 : Transmission de l'information selon Shannon.....	8
Figure 2-2 : Représentation et transformation des données.....	10
Figure 2-3 : Axe des représentations.....	12
Figure 2-4 : Cercle chromatique [Itten].....	13
Figure 2-5 : Synthèse additive des couleurs.....	14
Figure 2-6 : Synthèse soustractive des couleurs.....	15
Figure 2-7 : Le codage RGB.....	16
Figure 2-8 : Le modèle HSL.....	17
Figure 2-9 : Temps de rémanence lié au type de mémoire.....	19
Figure 2-10 : Exemples de modèles de pages Web exploitables par un PDA.....	21
Figure 2-11 : Visualisation sous forme d'un tableau.....	22
Figure 2-12 : Visualisation sous forme de graphiques.....	23
Figure 2-13 : "Map of the Market" (Application de SmartMoney).....	23
Figure 2-14 : Pocket PC Charts.....	24
Figure 2-15 : Pocket Stock Monitor.....	25
Figure 2-16 : Variété de produits HillCast.....	25
Figure 2-17 : MidCast PRO™ - Pocket PC.....	26
Figure 3-1 : Nombre de morts par accident et vente de vin.....	33
Figure 3-2 : Rendement du titre Alcatel.....	34
Figure 3-3 : Séries temporelles aux propriétés différentes.....	34
Figure 4-1 : Quelques modèles de PDA.....	36
Figure 5-1 : Pocket PC utilisé pour l'application.....	52
Figure 5-2 : Clavier de Pocket PC.....	54
Figure 5-3 : Plateforme "Microsoft .NET".....	57
Figure 5-4 : Compact Framework (1.1).....	59
Figure 5-5 : Plateforme J2ME.....	61
Figure 5-6 : Package javax.microedition.io.....	62
Figure 6-1 : Brick Wall Chart à 1 et 2 paramètre(s).....	64
Figure 6-2 : Brick Wall Chart à 4 paramètres.....	64
Figure 6-3 : Stacked Bar Chart (Janvier – Juin / Juillet – Décembre).....	65
Figure 6-4 : Bar Chart.....	66
Figure 6-5 : Mountain.....	67
Figure 6-6 : Mountain (avec zoom sur un mois).....	68
Figure 6-7 : Multi-Lines.....	69
Figure 6-8 : Multi-Lines (avec zoom sur un mois).....	69
Figure 6-9 : DashBoard (Valeurs générales et extrémales).....	70
Figure 6-10 : DashBoard (Sélection du mois à visualiser).....	70
Figure 6-11 : Informations supplémentaires.....	72
Figure 6-12 : Sauvegarde de la configuration.....	72
Figure 6-13 : Passage sans aucune modification (dans tous les cas).....	74
Figure 6-14 : Passage sans modification de paramètres (partiel).....	74
Figure 6-15 : Passage avec ajout de paramètres.....	75
Figure 6-16 : Passage avec suppression de paramètres.....	75
Figure 7-1 : Aiguillage de l'instruction de création d'une visualisation selon son type.....	81
Figure A-1 : Fichiers sources de TSviz_2K5.....	106
Figure A-2 : Fichiers à transférer sur le Pocket PC.....	107
Figure A-3 : Fichiers présents sur le Pocket PC après transfert.....	108

Figure A-4 : Ouverture de l'application.....	108
Figure A-5 : Ouverture du menu principal.	109
Figure A-6 : Création d'un profil.	110
Figure A-7 : Menus présents dans la partie visualisation (1).	113
Figure A-8 : Menus présents dans la partie visualisation (2).	113
Figure A-9 : Message de confirmation pour quitter le programme.	113
Figure A-10 : Configuration de l'application.....	114
Figure A-11 : Messages d'aide.	115
Figure A-12 : Enregistrement d'une configuration.	115
Figure A-13 : Chargement d'une visualisation.	116
Figure A-14 : Brick Wall Chart - Bar Chart.	117
Figure A-15 : Stacked Bar Chart de Janvier à Juin et de Juillet à Décembre.....	117
Figure A-16 : Mountain et zoom sur le mois de Janvier.	117
Figure A-17 : Multi-Lines et zoom sur le mois de Janvier.	118
Figure A-18 : Interaction "Switch".....	119
Figure A-19 : Explorateur de fichiers.	120
Figure A-20 : Statut de l'utilisateur.	121
Figure A-21 : Topics d'aide.....	122
Figure B-1 : Table de Chi carré.	123
Figure B-2 : Table de Fisher.	124

0 Introduction

De nos jours, d'énormes masses d'informations sont produites chaque année. Grâce à l'utilisation d'Internet, il est désormais possible pour une grande majorité de personnes d'avoir accès à ces informations et cela, en temps réel. L'accès à ces informations, de manière permanente et en temps réel, permet une prise de décision beaucoup plus rapide, efficace et facile pour les utilisateurs.

Parmi toute cette masse d'informations diffusées sur Internet, nous retrouvons les séries temporelles. Celles-ci sont utilisées dans des domaines aussi variés et distincts les uns que les autres : nous les retrouvons dans des domaines tels que la finance, la météorologie, la courantométrie, la médecine, l'espace et bien d'autres... Cependant, elles ont toutes la même caractéristique : représenter des informations à intervalles réguliers sur une période plus ou moins longue. Afin d'observer et d'analyser ces séries temporelles, diverses techniques de visualisations ont été élaborées. Cependant, un grand nombre de ces techniques ont été développées afin d'être utilisées sur ordinateurs (que ceux-ci soient dans des bureaux ou au domicile des utilisateurs).

Depuis quelques années, nous avons pu assister à l'émergence d'appareils d'un genre nouveau grâce aux systèmes embarqués. Ces appareils, dont le PDA fait partie, visent à offrir aux utilisateurs un large éventail de possibilités quant aux applications qu'ils développent. Ceci en permettant une mobilité que n'offrent pas les ordinateurs classiques. En effet, si, auparavant, les séries temporelles ne pouvaient être visualisées que sur ordinateurs, ce n'est plus le cas aujourd'hui et, l'utilisation du PDA pour visualiser ces séries temporelles ne peut être négligée. Un grand nombre d'applications réservées aux ordinateurs sont désormais accessibles pour ces systèmes embarqués.

Ces appareils embarqués disposent d'avantages sur les ordinateurs standards tels leur mobilité, mais ils présentent également plus de limitations technologiques que ce soit au niveau de la mémoire disponible ou de l'affichage des données consécutif à la taille de l'écran.

Ce mémoire a pour but de présenter la conception, le développement et l'évaluation d'un outil générique pour la visualisation de séries temporelles sur PDA. Les séries temporelles envisagées sont diverses et non focalisées sur un seul type de données.

Le premier chapitre présente quelques définitions et certains termes techniques qui seront utilisés ultérieurement.

Le deuxième chapitre expose l'état de l'art des techniques de visualisation existantes pour des séries temporelles ainsi que des applications existantes développées pour les sites Web et les PDA.

Le troisième chapitre présente ce que sont les séries temporelles, comment elles ont été définies ainsi que les buts/objectifs qu'elles permettent de poursuivre.

L'analyse des besoins est développée au chapitre quatre. Pour tout nouvel outil, il faut se demander et déterminer l'utilité de développer une nouvelle application pour la visualisation de séries temporelles.

Le cinquième chapitre présente le PDA utilisé pour l'application ainsi que les limitations techniques rencontrées avec ce PDA.

Dans le sixième chapitre, nous présentons les concepts ergonomiques mis en œuvre pour la création d'un tel outil ainsi que les différentes visualisations pouvant être mises en œuvre sur PDA.

C'est au chapitre sept qu'est développée la partie traitant de l'implémentation proprement dite. Les outils de développement et les langages y seront présentés et choisis. Les différentes limites et améliorations possibles y sont également décrites.

Une évaluation de l'application par des utilisateurs est présentée au chapitre huit.

Le chapitre neuf conclut en reprenant les points importants des différents chapitres et l'orientation à donner à de futurs développements. Il suggère également quelques améliorations pour l'application.

Et enfin, ce mémoire se clôturera par les annexes. Celles-ci contiennent un mode d'emploi de l'application qui a été développée.

1 Définitions

1.1 Acronymes

Acronyme	Signification
[ADO]	ActiveX Data Object
[ASP]	Active Server Pages
[BC]	Bar Chart
[BIOS]	Basic Input/Output System
[BWC]	Brick Wall Chart
[CDC]	Connected Device Configuration
[CLDC]	Connected Limited Device Configuration
[CLI]	Common Language Infrastructure
[CLR]	Common Language Runtime
[DB]	DashBoard
[DNA]	Distributed interNet Architecture
[ECMA]	European Computer Manufacturers Association
[GPRS]	General Packet Radio
[IHM]	Interaction Homme Machine
[IIS]	Internet Information Service
[IV]	Information Visualization
[JAE]	Java Application Environment
[JSP]	Java Server Pages
[JVM]	Java Virtual Machine
[KVM]	Kilobyte Virtual Machine
[ML]	Multi-Lines
[ODBC]	Open DataBase Connectivity
[PDA]	Pocket Digital Assistant
[SBC]	Stacked Bar Chart
[SDRAM]	Synchronous Dynamic Random Access Memory
[SQL]	Structured Query Language
[SOAP]	Simple Access Object Protocol
[TFT]	Thin Film Transistor
[UMTS]	Universal Mobile Telecommunication System
[XML]	eXtensible Markup Language

Tableau 1-1 : Acronymes utilisés dans la suite du mémoire.

1.2 Lexique

Quelques définitions de termes qui reviendront dans la suite du mémoire :

- A -

- ADO (Active Data Object)

ADO est un framework dont l'objet est la prise en charge des accès aux bases de données. Contrairement à ses concurrents, ADO se caractérise par deux orientations particulières : une optimisation pour l'utilisation des accès en mode déconnecté aux bases de données et l'usage d'XML pour l'échange de données.

- Application Server - Serveur d'application

Serveur regroupant un ensemble de services permettant aux différentes parties d'une application répartie, d'entrer en interaction. Dans le cadre de la plate forme « .NET », ces services sont intégrés dans les versions Serveur et Advanced Server de Windows 2000 et Windows XP. Entre autres services, nous pouvons citer le serveur Web, les conteneurs de composants, le service de nommage et d'annuaire, le service de transaction et le service de message (messaging).

- ASP

Active Server Pages : Technologie de Microsoft permettant de réaliser des sites web dynamiques, c'est-à-dire liés à des bases de données.

- B -

- Bluetooth

Norme de liaison radio, donc sans fils, permettant de connecter des appareils entre eux (téléphone, PDA, périphériques).

- C -

- C# (C Sharp)

C# (prononcer C Sharp) est un langage inventé par Microsoft qui permet de réaliser des sites web dynamiques. C'est le langage vedette supporté par la plateforme « .NET ». C# est très proche du langage Java. Il supporte l'héritage simple, l'implémentation multiple d'interface, les propriétés, la surcharge d'opérateur, les événements, ...

- CLR (Common Language Runtime)

Environnement permettant l'exécution de code compilé sous forme de MSIL. Juste avant l'exécution, le CLR compile le code MSIL en code machine natif grâce à un compilateur Just In Time (JIT). Pendant l'exécution la mémoire est automatiquement gérée par le CLR, grâce à un Garbage Collector.

- COM (Component Object Model)

COM est un framework qui permet la programmation d'objets distribués sur la plate forme Windows. COM est le fondement de OLE (documents composites) et du format de composants ActiveX. Il définit un format d'interfaces indépendant des langages utilisés et permet la communication entre applications Windows au niveau des objets. Le périmètre de COM est restreint aux communications intra-processus ou inter-processus au sein d'une même machine.

- CTS (Common Type System) - Système de typage standard

Le CTS est un standard de définitions de types de données. Il définit le format interne des types élémentaires (entiers, monétaires, nombres à virgule flottante, chaînes, tableaux ...) et des types complexes (structures, objets ...) utilisés par la plate forme « .NET ». Ces types de données ont été conçus de manière à faciliter l'implémentation d'une large gamme de langages de programmation.

- D -

- dot Net

Technologie de Microsoft permettant de réaliser des sites web dynamiques, c'est-à-dire liés à des bases de données.

- E -

- Emulateur

Logiciel permettant de simuler un terminal sur un ordinateur. Par exemple, un émulateur WAP permet d'afficher des sites WAP sur un ordinateur.

- F -

- Framework

L'utilisation d'un framework, ou application semi finie, permet la réalisation rapide d'un site par une réutilisation de fonctionnalités existantes et leur adaptation pour un besoin spécifique.

- G -

- GDI (Graphic Device Interface)

Le GDI est la bibliothèque qui prend en charge les fonctionnalités d'affichage de la plate forme Windows. Parmi les fonctions supportées, nous avons l'affichage de texte, la gestion des couleurs, le dessin de formes géométriques, la prise en charge de l'impression et du traitement des images. Avec « .NET » le GDI possède aussi désormais une bibliothèque qui prend en charge toutes les fonctionnalités de dessin 2D à l'instar de Java 2D chez Sun.

- I -

- IIS (Internet Information Server)

IIS est le serveur Web de Microsoft. Il joue un triple rôle au sein de la plate forme « .NET » : l'émission de contenu web statique, la génération de contenu web dynamique via les ASP (Active Server Pages) et il sert également de point d'entrée pour les Web Services. A ce titre, IIS est la plateforme de déploiement pour les applications « .NET » utilisant le web.

- ISP (Internet Service Provider)

Le fournisseur d'accès est le lien entre l'utilisateur et le réseau mondial Internet. C'est sa porte d'entrée sur Internet. Le fournisseur d'accès est aussi appelé provider.

- M -

- Managed Code - Code pris en charge, code "managé"

Se dit d'un code C++ qu'il est possible de compiler en code intermédiaire MSIL et qui est exécutable sous le contrôle de l'environnement d'exécution CLR. Ceci implique notamment que la mémoire est gérée par le Garbage Collector et non pas par le développeur. Pour écrire du code C++ managé, il est nécessaire d'utiliser des mots clés spécifiques qui sont des extensions propriétaires introduites par Microsoft.

- MSIL (Microsoft Intermediate Language)

Langage intermédiaire constitué d'instructions élémentaires de type 'assembleur évolué'. Tous les langages supportés par la plateforme « .NET » peuvent être compilés en code intermédiaire MSIL. Ce code intermédiaire peut ensuite être exécuté par le CLR, l'environnement d'exécution de la plateforme « .NET ». MSIL permet l'indépendance au langage de la plateforme « .NET ».

- P -

- PDA

Personal Digital Assistant : Ordinateur de poche qui permet de gérer ses contacts, ses rendez-vous, etc. Les principales technologies de PDA sont Palm et PocketPC.

- PocketPC

Technologie d'assistants personnels développée par Microsoft.

- R -

- Reflection - Réflexions

Mécanisme du CLR qui permet à du code de « se connaître lui-même ». La réflexion permet de récupérer des informations sur le code, et notamment, la liste de types déclarés : classes, structures, énumérations, ... La réflexion permet également d'obtenir la description de chacun des types : variables membres, méthodes, ...

- U -

- Unmanaged Code - Code non pris en charge, code non "managé"

Se dit d'un code C++ qui n'est pas compilé en MSIL mais en code machine. Il s'agit généralement de code écrit avant l'existence de « .NET ». Dans un code non managé, la gestion de la mémoire est à la charge du développeur. Un code non managé peut éventuellement utiliser des services de l'infrastructure « .NET ».

- X -

- XML (eXtended Markup Language)

Langage extensible permettant de décrire toute information.

2 Etat de l'art

2.1 Introduction

De nos jours, il est difficile de trouver une application ou un système qui ne soit informatisé. En effet, de plus en plus de données sont produites chaque année. Pour les traiter, il est nécessaire de faire appel à l'informatique. Qui plus est, l'informatique est devenue un outil courant dans notre vie de tous les jours. En partant du principe que ces données sont destinées à être analysées afin de permettre une prise de décisions, il faut parvenir à les rendre compréhensibles par les personnes qui vont les analyser et à les présenter sous une forme claire.

Cependant, les problèmes posés par la création de telles quantités de données sont les suivants :

- Comment pouvons-nous utiliser l'informatique pour représenter ces quantités de données de manière(s) compréhensible(s) aux utilisateurs ?
- Comment pouvons-nous arriver à les représenter sous une forme claire de manière à visualiser distinctement et immédiatement les informations les plus importantes ?

Actuellement, les techniques de visualisations classiques ne permettent pas de représenter autant d'informations de manière suffisante et de ce fait, d'atteindre un niveau de compréhension suffisant pour les utilisateurs.

Dans le but de remédier à ce problème, des techniques de visualisation de l'information ont commencé à voir le jour. Ces techniques ont pour objectif de permettre aux utilisateurs d'analyser ces gigantesques quantités d'informations et surtout de les comprendre. Pour cela, des logiciels d'explorations de données ont été élaborés. Ils se basent, tout comme ces techniques de visualisation, sur la capacité qu'a le système visuel humain à analyser et traiter les informations.

La branche de l'informatique qui comprend la visualisation de l'information est l'Interaction Homme-Machine (IHM). Le but de cette branche est de trouver le meilleur moyen d'interagir avec les utilisateurs afin de subvenir à leurs besoins. En effet, non seulement les données produites sont de plus en plus nombreuses à analyser mais également de plus en plus complexes. Il faut donc trouver des méthodes pour permettre de modéliser ces données de façon à les rendre utilisables par l'utilisateur.

Dans la visualisation des informations, nous nous attarderons donc à donner un sens à ces données brutes inutilisables sous leurs formes originales. Pour cela, des outils doivent être mis

à disposition des utilisateurs de telle sorte qu'ils puissent traiter les données dont ils ont besoin pour réaliser une analyse et prendre une décision.

2.2 L'information

Selon Nicolas Schöffer¹, l'information est

- l'émission, la réception, la création, la retransmission de signaux groupés oraux ou écrits, sonores, visuels ou audiovisuels,
- en vue de la diffusion et de la communication d'idées, de faits, de connaissances, d'analyses, de concepts, de thèses, de plans, d'objets, de projets, d'effets de toutes sortes, dans tous les domaines,
- par un individu, par des groupes d'individus ou par un ou plusieurs organismes agissant ou rétroagissant ainsi sur leur environnement immédiat, proche ou lointain, et dont le but est de déclencher éventuellement des processus dialectiques plus ou moins amples alimentant l'échange, base naturelle et indispensable de l'animation de la vie sociale.

Quelques définitions de la notion « d'information »

- Élément de connaissance susceptible d'être représenté à l'aide de conventions pour être conservé, traité ou communiqué.
- Signification attribuée à des données, dans un contexte précis et en fonction du cadre de référence utilisé.
- Connaissance consignée, inscrite, dans le but de sa transmission.

Comme nous le constatons dans les différentes définitions ci-dessus et sur le schéma suivant (schéma de Shannon), pour que l'information soit transmise entre la source/l'émetteur et le récepteur, il doit exister un canal. Ce canal est un code commun aux deux termes de la communication et si ce code n'est pas commun, nous ne nous comprenons pas.

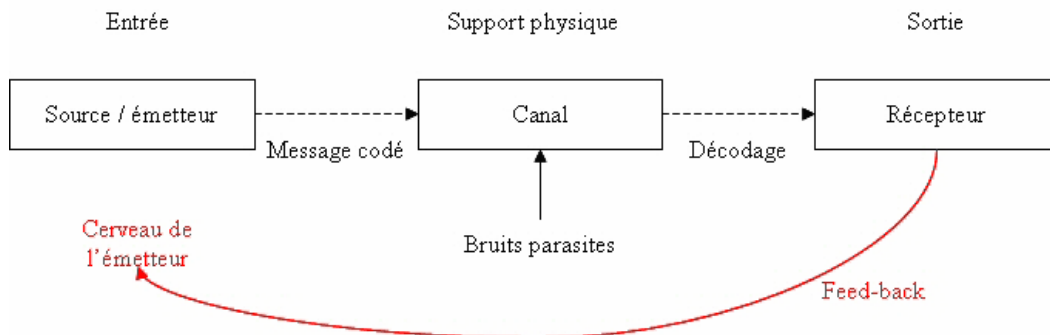


Figure 2-1 : Transmission de l'information selon Shannon

2.3 Visualisation de l'information

2.3.1 Qu'est-ce que la visualisation ?

La visualisation consiste à traduire, à transcrire ou encore à coder l'information retenue dans une forme visuelle pertinente: il s'agit de construire un schéma graphique, une image, etc.

¹ Extrait d'un article de Jean Mandelbaum du 1^{er} août 1982 paru dans Le Monde.

adéquat par rapport à l'information qu'il faut communiquer ainsi qu'aux relations logiques sous-jacentes. De façon plus formelle, nous pouvons parler d'un processus cognitif qui permet de former une image mentale d'un élément, et de le comprendre. Par visualisation, il faut comprendre l'utilisation de représentations visuelles interactives (i.e. supportées par les ordinateurs) de données pour amplifier la cognition de celles-ci.

Il faut donc considérer chaque type de représentation visuelle (tableaux, schémas, organigrammes, graphiques, cartes, plans, dessins figuratifs, photographies, etc.) comme une forme de communication spécifique, comme un langage, possédant ses règles et son code. Il s'agit de formes d'expression relativement conventionnelles (le code de la route, les symboles qui désignent les disciplines sportives aux jeux olympiques en sont de bons exemples) qu'il faut respecter si nous voulons garantir l'efficacité de la communication. Il faut en effet que le rédacteur et les apprenants possèdent le même langage, les mêmes règles, les mêmes codes. Les facteurs socioculturels sont donc d'une importance capitale.

2.3.2 Qu'est-ce que la visualisation de l'information ?

La visualisation de l'information, bien qu'elle soit liée à la visualisation scientifique (la modélisation informatique de données brutes), est à proprement parler un sous-champ d'Interaction Homme-Machine [DEMAINE96]. En effet, il s'agit d'un ensemble de techniques permettant de représenter des données structurées.

Il est possible d'extraire plusieurs points importants de la définition donnée au point précédent :

- La visualisation d'informations est une transformation de celles-ci (les informations possédant une correspondance dans le monde physique) afin de les transposer dans un espace de représentation visuelle (c'est-à-dire généralement en deux ou trois dimensions).
- L'interprétation visuelle de l'information met en jeu des aspects cognitifs (cf. Section 2.3.3). ils doivent être pris en compte lors de l'étude d'aspects informatiques de la visualisation. Par induction de la définition de l'information, les principes cognitifs, le mécanisme de perception et de reconnaissance d'informations visuelles doivent être compris lors de la conception de techniques de visualisation.
- La visualisation de l'information ne doit pas être dissociée d'une éventuelle possibilité d'interaction avec l'information via les représentations produites. D'ailleurs, le domaine informatique dans lequel la visualisation d'informations s'est développée est l'IHM (Interaction Homme-Machine). Il est donc tout à fait naturel de retrouver le terme « interaction » dans la définition proposée plus haut.

Les visualisations permettent la réalisation d'une série de tâches correspondant à des objectifs :

- La recherche : cela correspond à une recherche simple suite à une question simple et qui peut trouver sa réponse sur simple présentation des données.
- La navigation : cette particularité (explorer les données en voyageant à travers) permet une interactivité avec l'utilisateur.
- L'analyse : certains outils proposent une analyse des données.
- La surveillance : des écarts inhabituels de comportement peuvent être décelés, grâce à des visualisations, par des systèmes de mise à jour de données.

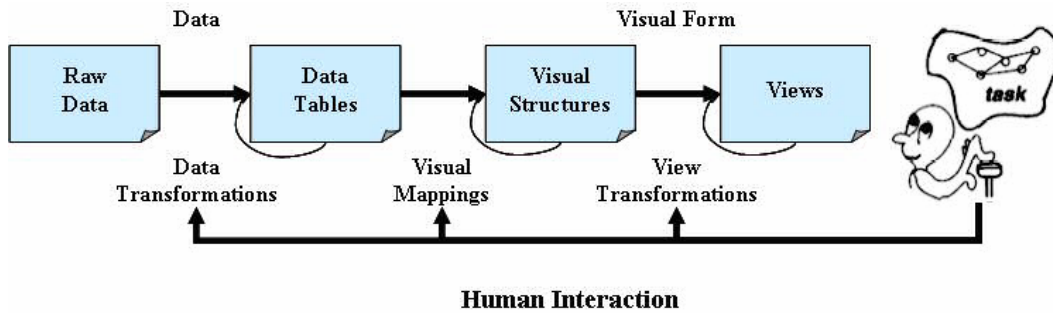


Figure 2-2 : Représentation et transformation des données.

En tirant parti de la vitesse de traitement et des capacités graphiques des ordinateurs, la visualisation de l'information permet aux utilisateurs d'interpréter une masse d'informations.

Le fait que nous disposions d'informations sur les données, distingue la visualisation de l'information des méthodes employées en statistique où nous cherchons justement à découvrir des relations entre les données.

Le but de la visualisation de l'information est de représenter de façon cohérente et claire un nombre important de données afin qu'une personne puisse prendre conscience des informations structurelles présentes dans ces données [POULINGEAS04]. Pour cela, il faut tenir compte du fait que l'utilisateur peut être amené à manipuler la représentation que nous lui offrons (ce qui implique une visualisation et des interfaces adaptées).

Deux problèmes apparaissent alors :

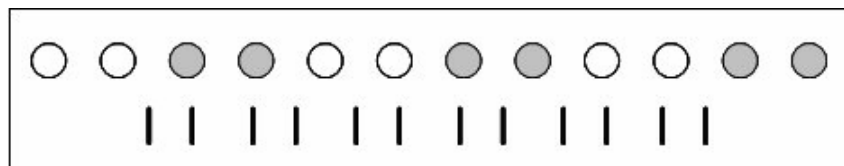
- Le contexte : nous devons toujours avoir une vue globale (même imprécise) de la totalité des données.
- Le focus : nous devons pouvoir sélectionner une donnée particulière.

A titre d'exemple : Dans l'explorateur de Windows, nous avons le focus (puisque nous pouvons arriver à sélectionner n'importe quel fichier d'une arborescence d'un disque). Par contre, nous perdons souvent le contexte (à force de développer la hiérarchie d'un disque, nous ne voyons plus la totalité des répertoires situés à la racine de celui-ci).

2.3.3 Principes essentiels de la visualisation

- Le principe d'association

L'œil associe spontanément les formes qui se ressemblent (association par similarité ou égalité) ou qui sont plus proches (association par proximité). Exemple d'illusions d'optiques : dans le premier cas, l'œil regroupe deux par deux les points des cercles noirs et des cercles blancs, dans le second, il rassemble les lignes les plus proches.



Pour cette raison, dans un texte imprimé, l'usage des espaces blancs et des interlignes revêt une grande importance. La structure des paragraphes, et donc la cohérence logique du texte, peuvent disparaître complètement si les interlignes sont inadéquats.

- Le principe de monosémie

C'est faire correspondre à chaque variable cognitive (élément d'informations, données ou relation) une et une seule variable visuelle (couleur, forme, etc.). Si dans une carte de géographie, le bleu représente l'eau (les cours d'eau, la mer, l'océan, etc.), cette couleur ne doit plus être utilisée pour représenter un autre élément d'information.

- Le principe de l'effet proportionnel

Les données doivent être traduites sous une forme visuelle dont les variations sont interprétables en fonction des valeurs propres des données. Les « camemberts » ou les histogrammes sont une bonne illustration de ce principe. De même, la variation d'intensité d'une même couleur peut traduire un phénomène progressif (plus le bleu est intense, plus la profondeur de l'eau est grande).

- Le principe du moindre coût

Il faut toujours opter pour la représentation la plus économique et la plus simple. La compréhension et la communication obéissent spontanément à la loi du moindre effort.

- Le principe de simplification

En conséquence, les dessins figuratifs et les illustrations sont plus lisibles s'ils sont simplifiés. Un dessin est souvent plus explicite qu'une photographie qui, réaliste, ne permet pas de sélectionner l'information pertinente. De même une représentation est plus lisible en noir et blanc qu'en couleur sauf si celle-ci est l'unique façon de représenter l'information pertinente.

- Le principe de familiarisation

Il est toujours préférable, dans la mesure du possible, d'utiliser des pictogrammes (modèles graphiques normalisés), des symboles et des conventions dont l'usage s'est déjà répandu. Les composantes culturelles ou socioculturelles sont dans cette perspective très importantes.

2.3.4 Typologies des représentations

Il existe plusieurs classifications des formes de représentation. Certaines ressemblent plus que d'autres à l'objet, à la réalité qu'elles représentent. Aussi les classifications se structurent-elles autour de ce que nous avons appelé une "échelle d'iconicité". Celle-ci rend compte du degré de ressemblance de la représentation, de son degré d'analogie avec ce qui est représenté. Une photo du chien est par exemple, plus réaliste, plus ressemblante qu'un dessin. Un dessin réaliste en couleur est plus proche de l'objet réel qu'un dessin au trait ou qu'une silhouette.

Cet axe progressif permet de passer des représentations les plus concrètes, les plus ressemblantes, aux plus abstraites, c'est-à-dire aux moins ressemblantes. L'intérêt de cette classification paraît évident si elle reprend comme critères les catégories symboliques plus générales. Ces catégories sont aussi celles des images mentales :

- d'une part, l'abstrait, l'arbitraire et le fonctionnement des langages naturels (ex : la langue française) ou artificiels (ex : la logique formelle, le langage mathématique).
- d'autre part, le concret, l'analogique et les représentations figurées.

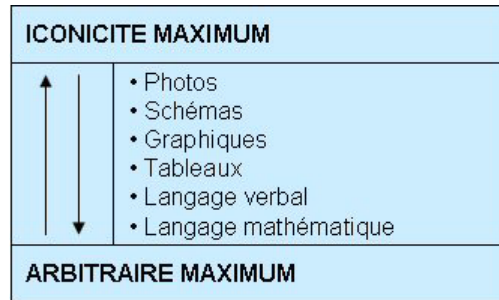


Figure 2-3 : Axe des représentations.

Autrement dit, cette classification permet de structurer les formes de représentation en fonction du degré d'abstraction des connaissances et des représentations cognitives qu'elles sont susceptibles de véhiculer. Nous distinguons donc :

- La photographie

De toutes les représentations – présentées sur un support papier –, la photo est la plus ressemblante. Elle est le produit d'une saisie du réel, fondée sur un processus complexe (électromécanique, optique et chimique) dans lequel l'intervention de l'auteur, contrairement aux autres formes graphiques, est relativement limitée. Pourtant une trop grande ressemblance ne facilite pas toujours la reconnaissance de l'objet représenté. Trop de détails par exemple, peuvent nuire à l'identification d'une forme générale.

- Les schémas

Nous regroupons dans cette catégorie les dessins, les croquis et les schémas. Les uns et les autres sont le résultat d'un processus d'analyse et de sélection de l'information, bref d'un processus de schématisation. Contrairement à la photographie, ils sont une figuration simplifiée, fonctionnelle et modélisante du réel.

Le degré d'iconicité de ces représentations peut cependant varier fortement et c'est en cela qu'elles se différencient. Il existe des dessins réalistes, stylisés et descriptifs, mais aussi des schémas strictement conventionnels tels que les diagrammes et les organigrammes.

- Les premiers représentent le plus souvent un réel observable: animaux, paysages, organes, objets, etc.
- Les seconds constituent le référent visuel d'un phénomène complexe, souvent abstrait (ex : les phases distinctes d'un processus global et leurs relations).
- Les graphiques

- Les tableaux

Ils présentent des données chiffrées ou verbales dans une forme visuelle qui en rend la lecture aisée tout en faisant apparaître des relations entre certaines d'entre elles.

Cette classification possède d'autres avantages pour le rédacteur. Elle apprend d'abord que plus les représentations sont abstraites, plus elles sont conventionnelles. Ensuite, elle demande l'apprentissage d'un « mode d'emploi », d'un code rigoureux pour les lire et les comprendre. Les légendes des cartes de géographie, les symboles des cartes météorologiques synoptiques constituent de bons exemples.

En résumé, cette échelle d'iconicité exprimerait aussi le degré croissant de difficulté de lecture, de compréhension et d'exploitation. Plus la représentation est abstraite, plus elle serait difficile à interpréter et à utiliser par les apprenants. Ces derniers aspects constituent pour le rédacteur un important critère de sélection des illustrations.

2.4 La notion de couleur

L'utilisation des couleurs est courante dans les visualisations de l'information. Il existe trois couleurs primaires à partir desquelles il est possible de composer les autres couleurs. Cependant, il peut être utile d'élargir cette base. Le cercle chromatique [ITTEN86] présente ainsi une palette de 12 couleurs dont les six principales sont : le vert, le bleu, le violet, le rouge, l'orange et le jaune.

Le cercle chromatique est un élément fondamental de l'enseignement esthétique des couleurs, car il représente le classement des couleurs. Dans le cercle de J. Itten (ci-contre), le bleu s'oppose à l'orange car leur mélange donne du gris. Par contre, dans le cercle d'Ostwald, le bleu est placé face au jaune, ce qui donne un mélange pigmentaire vert.



Figure 2-4 : Cercle chromatique [Itten].

En utilisant les couleurs présentes dans le cercle cognitif et en les combinant, il est possible de composer n'importe quelle couleur.

2.4.1 Classification des couleurs

Toutes les couleurs ne possèdent pas les mêmes caractéristiques et sont utilisées dans des cas différents pour faire passer des messages, attirer l'attention d'une personne ou encore faire percevoir des sensations diverses.

Comment sont classifiées les couleurs ?

- Couleur primaire : Une couleur primaire ne peut être créée par le mélange d'autres couleurs. En mélangeant les couleurs primaires entre elles, nous obtenons toutes les autres couleurs. Les trois couleurs primaires sont le rouge, le jaune et le bleu.
- Couleur secondaire : Nous obtenons une couleur secondaire en mélangeant deux couleurs primaires entre elles. Le rouge (magenta) et le jaune donnent l'orangé, le jaune et le bleu (cyan) donnent le vert, et le rouge et le bleu (cyan) donnent le violet.
- Couleur intermédiaire : En mélangeant une couleur primaire à une couleur secondaire, nous obtenons une couleur intermédiaire (i.e. le jaune-orangé).
- Couleurs complémentaires : Les couleurs qui se trouvent opposées dans le cercle chromatique sont appelées couleurs complémentaires. (i.e. la couleur complémentaire du rouge est le vert).
- Couleurs chaudes : Couleurs allant du jaune au rouge-violet sur le cercle chromatique, c'est-à-dire : jaune, jaune-orangé, orangé, rouge-orangé, rouge, rouge-violet.
- Couleurs froides : Couleurs allant du bleu-violet au jaune-vert sur le cercle chromatique, c'est-à-dire : bleu-violet, bleu, bleu-vert, vert, jaune-vert.
- Couleurs pâles ou claires : Teintes contenant plus ou moins de blanc.

- Couleurs foncées ou sombres : Teintes contenant plus ou moins de noir.
- Couleurs saturées ou lumineuses : Teintes pures ne contenant en principe ni blanc, ni noir, ni gris, ni couleur complémentaire. Cependant, cette définition peut être nuancée, ce qui étend un peu la palette des teintes saturées.
- Couleurs insaturées ou atténuées de gris : Teintes contenant plus ou moins de gris ou de leur couleur complémentaire. Les théoriciens de la couleur emploient aussi l'expression « couleurs ternes » pour désigner ces couleurs. Cette expression ne véhicule aucun sens péjoratif.

2.4.1.1 Synthèse additive et soustractive

Il existe deux types de synthèse de couleur [CCMa]:

- La synthèse additive est le fruit de l'ajout de composantes de la lumière. Les composantes de la lumière sont directement ajoutées à l'émission, c'est le cas pour les moniteurs ou les télévisions en couleur. Lorsque nous ajoutons les trois composantes Rouge, Vert, Bleu (RVB), nous obtenons du blanc. L'absence de composante donne du noir. Les couleurs secondaires sont le cyan, le magenta et le jaune car :
 - Le vert combiné au bleu donne du cyan
 - Le bleu combiné au rouge donne du magenta
 - Le vert combiné au rouge donne du jaune

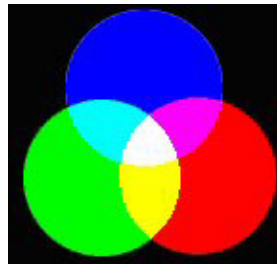


Figure 2-5 : Synthèse additive des couleurs.

- La synthèse soustractive permet de restituer une couleur par soustraction, à partir d'une source de lumière blanche, avec des filtres correspondant aux couleurs complémentaires : jaune, magenta, et cyan. L'ajout de ces trois couleurs donne du noir et leur absence produit du blanc. Les composantes de la lumière sont ajoutées après réflexion sur un objet, ou plus exactement sont absorbées par la matière. Ce procédé est utilisé en photographie et pour l'impression des couleurs. Les couleurs secondaires sont le bleu, le rouge et le vert car :
 - Le magenta (couleur primaire) combiné avec le cyan (couleur primaire) donne du bleu
 - Le magenta (couleur primaire) combiné avec le jaune (couleur primaire) donne du rouge
 - Le cyan (couleur primaire) combiné avec le jaune (couleur primaire) donne du vert

Deux couleurs sont dites « complémentaires » si leur association donne du blanc en synthèse additive, ou du noir en synthèse soustractive.

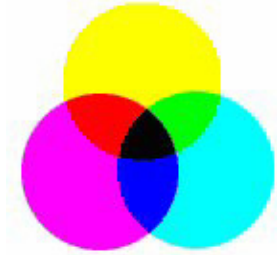


Figure 2-6 : Synthèse soustractive des couleurs.

2.4.2 Les sept contrastes de la couleur

Les sept contrastes de la couleur sont à la base de presque tous les effets de couleur [ITTEN86]. Nous parlons de contrastes lorsque nous pouvons constater, entre deux effets de couleurs qu'il faut comparer, des différences ou des intervalles sensibles. Lorsque ces différences atteignent un maximum, nous parlons de contrastes d'opposition ou de contrastes polaires (chaud-froid, blanc-noir, petit-grand). De la même façon, nous définirons le contraste comme étant « l'opposition marquée entre deux régions d'une image, plus précisément entre une région sombre et une région claire de cette image » [KADDOUR99].

- **Contraste de la couleur en soi**

Le plus simple de tous. Il n'exige pas de grands efforts de la vision. Il s'agit de l'opposition entre des couleurs saturées nettement différentes.

- **Contraste clair-obscur**

Il s'agit de l'opposition entre une couleur pâle et une couleur foncée, une couleur claire et une couleur sombre.

- **Contraste chaud-froid**

C'est l'opposition entre une couleur chaude et une couleur froide. Ce contraste a pour effet de faire paraître les teintes chaudes plus chaudes lorsqu'elles sont placées près de teintes froides et vice-versa.

- **Contraste des complémentaires**

Ce terme désigne deux couleurs pigmentaires dont le mélange donne un gris-noir de ton neutre. Il s'agit de l'opposition entre les couleurs diamétralement opposées sur le cercle chromatique.

- **Contraste simultané**

C'est l'opposition entre deux couleurs qui ne sont pas exactement complémentaires.

- **Contraste de qualité**

C'est l'opposition entre une couleur saturée et une couleur insaturée, ou une couleur lumineuse et une couleur atténuée (à laquelle nous avons ajouté du gris ou de sa couleur complémentaire).

- **Contraste de quantité**

C'est l'opposition entre « peu et beaucoup », « petit et grand ». La surface consacrée à chaque couleur influence leur impact dans une composition. Pour créer une composition équilibrée, il ne suffit pas de consacrer des surfaces égales à chaque couleur, il faut également tenir compte de la clarté et de la saturation de chacune des teintes.

2.4.3 Les représentations de la couleur en informatique

Afin de pouvoir manipuler correctement des couleurs et échanger des informations colorimétriques, il est nécessaire de disposer de moyens permettant de les catégoriser et de les

choisir. Ainsi, il n'est pas rare d'avoir à choisir la couleur d'un produit avant même que celui-ci ne soit fabriqué.

Dans ce cas, une palette de couleurs est présentée, et la couleur convenant le mieux au besoin est choisie. La plupart du temps, le produit (véhicule, bâtiment, etc.) possède une couleur qui correspond à celle choisie.

En informatique, de la même façon, il est essentiel de disposer d'un moyen pour choisir une couleur parmi toutes celles utilisables. Or la gamme de couleurs possibles est très vaste.

Nous appellerons donc «espace de couleurs» la représentation mathématique d'un ensemble de couleurs. Il en existe plusieurs, parmi lesquels les plus connus sont [CCMb] :

- Le codage RGB (Rouge, Vert, Bleu, en anglais RGB, Red, Green, Blue).
- Le codage TSL (Teinte, Saturation, Luminance, en anglais HSL, Hue, Saturation, Luminance).

Le spectre de couleurs qu'un périphérique d'affichage permet d'afficher est appelé gamut ou espace colorimétrique. Les couleurs n'appartenant pas au gamut sont appelées couleurs hors-gamme.

1. Le codage RGB

Le codage RGB, mis au point en 1931 par la Commission Internationale de l'Eclairage (CIE) consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs :

- rouge (de longueur d'onde égale à 700,0 nm),
- vert (de longueur d'onde égale à 546,1 nm),
- bleu (de longueur d'onde égale à 435,8 nm).

Cet espace de couleurs correspond à la façon dont les couleurs sont généralement codées informatiquement, ou plus exactement à la manière dont les tubes cathodiques des écrans d'ordinateurs représentent les couleurs.

Etant donné que le codage RGB repose sur trois composantes proposant la même gamme de valeur, il est représenté généralement graphiquement par un cube dont chacun des axes correspond à une couleur primaire :

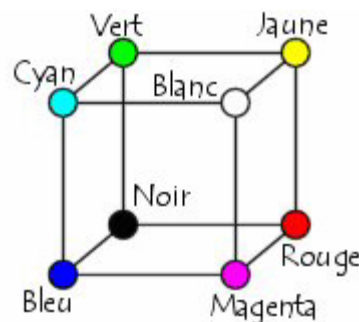


Figure 2-7 : Le codage RGB.

Ainsi, le modèle RGB propose de coder sur un octet chaque composante de couleur, ce qui correspond à 256 intensités de rouge (28), 256 intensités de vert et 256 intensités de bleu, soit 16777216 possibilités théoriques de couleurs différentes, c'est-à-dire plus que ne peut en discerner l'oeil humain (environ 2 millions). Toutefois, cette valeur n'est que théorique car elle dépend fortement du matériel d'affichage utilisé.

2. Le codage TSL

Le modèle HSL (Hue, Saturation, Luminance, ou en français TSL), s'appuyant sur les travaux du peintre Albert H. Munsell (qui créa l'Atlas de Munsell), est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'oeil humain. En effet, le modèle RGB aussi adapté soit-il pour la représentation informatique de la couleur ou bien l'affichage sur les périphériques de sortie, ne permet pas de sélectionner facilement une couleur.

Dans ce sens, le réglage de la couleur en RGB dans les outils informatiques se fait généralement à l'aide de trois glisseurs ou de trois cases avec les valeurs relatives de chacune des composantes primaires. Or, l'éclaircissement d'une couleur demande d'augmenter proportionnellement les valeurs respectives de chacune des composantes.

Ainsi le modèle HSL a été mis au point afin de pallier à cette lacune du modèle RGB.

Le modèle HSL consiste à décomposer la couleur selon des critères physiologiques :

- la teinte (en anglais Hue), correspondant à la perception de la couleur (T-shirt mauve ou orange),
- la saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère vif ou terne (T-shirt neuf ou délavé),
- la luminance, indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre (T-shirt au soleil ou à l'ombre).

Voici une représentation graphique du modèle HSL, dans lequel la teinte est représentée par un cercle chromatique et la luminance et la saturation par deux axes :

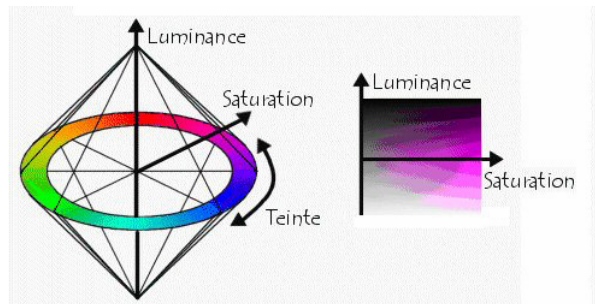


Figure 2-8 : Le modèle HSL.

Le modèle HSL a été mis au point dans le but de permettre un choix interactif rapide d'une couleur, pour autant il n'est pas adapté à une description quantitative d'une couleur.

2.5 Conception des interfaces

De nombreuses règles visant à développer de bonnes interfaces ont été mises en évidence depuis des années. Cependant, ces règles ne sont pas toujours faciles à énoncer et peuvent parfois apparaître contradictoires.

Certaines règles émises sont propres à des applications ou à des utilisateurs particuliers. Elles sont précises et ne nécessitent pas énormément d'interprétation de la part du concepteur. Mais, de leur spécificité, il est parfois difficile d'en tirer des enseignements généraux.

Les principes généraux d'ergonomie sont plus faciles à mémoriser que toutes les règles. Ils ont l'avantage d'être moins nombreux. Ils doivent toutefois être adaptés pour la conception de chaque interface et plus précisément en fonction de la tâche à accomplir et, du contexte dans lequel ils se trouvent.

Il faut donc bien distinguer les principes (généraux) et les règles (particulières). Afin d'être certain de bien concevoir une interface, l'idéal serait de pouvoir se baser sur les principes généraux suivants :

- Faire une interface adaptée à l'utilisateur et donc connaître préalablement l'utilisateur.
- Réduire la charge cognitive de sorte que l'utilisateur doive mémoriser un minimum.
- Aider l'utilisateur à avoir un modèle mental clair de l'application.
- Ne pas lui demander de faire plusieurs choses simultanément ou permettre des automatismes.

Hélas, ces principes sont beaucoup trop généraux pour être directement utiles aux concepteurs. Ils doivent donc être détaillés et explicités pour permettre aux concepteurs de réaliser une interface pleinement satisfaisante pour les utilisateurs.

Dans la partie suivante, ces principes seront détaillés afin de déterminer lesquels sont adéquats dans la situation qui nous occupe, à savoir l'interface de notre « outil de visualisation » sur PDA.

Bien que les principes et les règles aident le développeur dans sa tâche de conception d'une interface, la conception d'un logiciel adapté aux utilisateurs n'est pas en mesure de résoudre l'ensemble des problèmes humains posés par l'introduction de l'informatique. Inversement, il ne peut y avoir un processus d'informatisation réussi sans optimisation de la manière dont l'homme est amené à communiquer avec l'ordinateur.

2.5.1 Principes ergonomiques

Comme annoncé dans la partie précédente, les principes généraux exposés sont trop vagues pour permettre au concepteur de réaliser une interface adéquate. De nombreuses personnes ont rédigé des ouvrages détaillant les principes ergonomiques d'une bonne interface et parmi ceux-ci, nous pouvons citer :

- Logiciels interactifs et ergonomie [Marie-France Barthelet, 1988]
- Corpus minimal de recommandations ergonomiques [Jean Vanderdonckt, 1993]
- Human-Computer Interaction [Jenny Preece, 1994]
- Description des objets interactifs [J. Vanderdonckt, I. Provot, B. Sacré, 1995]

Ce que nous pouvons déjà dire, c'est qu'une interface informatique doit satisfaire à deux grands critères :

- Utilité
 - Les fonctionnalités proposées par le système doivent être utiles : l'interface doit servir à faire quelque chose, et ce quelque chose doit être pertinent au regard des objectifs de l'utilisateur cible. L'application doit servir un besoin.
 - Un système peut respecter tous les critères d'utilisabilité mais être inutile. C'est l'adéquation entre l'activité et l'outil qui permettra de dire si cet outil est utile.

- Les méthodes d'analyse de l'activité nous permettent de cerner quelles fonctionnalités doit fournir l'application, autrement dit quelles fonctionnalités sont utiles.
- Un système de bonne qualité ergonomique devra être à la fois utile et utilisable.
- Utilisabilité : La norme ISO 9241 définit l'utilisabilité de la façon suivante :
 "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."
 (Source: ISO 9241 part 11: forthcoming standard giving guidance on usability on requirements for office works with visual display terminals).

De façon générale, l'utilisabilité d'un système comprend :

- Sa facilité d'utilisation,
- Sa facilité d'apprentissage,
- Son efficacité d'utilisation,
- Son utilisation sans erreur,
- La satisfaction de ses utilisateurs

De même, nous pouvons énoncer quelques principes très généraux de conception d'interfaces :

- Cohérence : la cohérence est à adopter dans toute l'application.
- Concision : des manipulations trop longues (plusieurs écrans, plusieurs commandes) peuvent fatiguer l'utilisateur, surtout lorsque le travail est répétitif.
- Feed-back informatif : il faut informer l'utilisateur sur l'état du système.
- Gestion des erreurs : il faut détecter rapidement les erreurs de manipulation et d'interprétation des commentaires et permettre de les corriger.
- Retour en arrière : la réversibilité des actions est un paramètre important.
- Flexibilité : il faut s'adapter à l'évolution cognitive de l'utilisateur mais également s'adapter à l'évolution cognitive d'un ensemble d'utilisateurs.
- Contrôle explicite : ce principe signifie que même si c'est le logiciel qui a le contrôle, l'interface doit apparaître comme étant sous le contrôle de l'utilisateur.
- Séquencement des opérations : il s'agit de l'adéquation entre l'ordre des opérations fixé par la machine et celui nécessaire à l'opérateur pour effectuer sa tâche dans n'importe quel environnement.
- Le temps de réponse : les informations qui entrent ou qui sortent du cerveau humain transitent par une mémoire à court terme pendant 2 à 6 secondes au maximum.
 - Par conséquent, si, en cours de travail, nous sommes interrompus par un événement extérieur, nous risquons d'effacer l'information de la mémoire à court terme. Il est pénible de devoir conserver trop longtemps l'information dans la mémoire à court terme. C'est le cas si le temps de réponse est supérieur à 4 secondes.

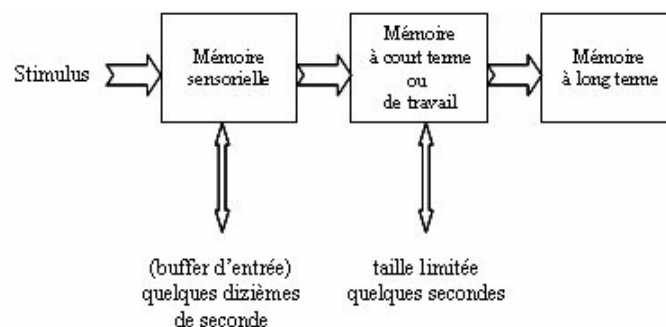


Figure 2-9 : Temps de rémanence lié au type de mémoire.

- Par contre, un temps de réponse plus long pour une opération qui ne demande pas de mémorisation (l'impression, fin de travail), ne pose pas de problème à condition toutefois que l'utilisateur soit averti par un message affiché sur l'écran.
- Si le temps de réponse est trop long, il faut l'afficher. L'utilisateur pourra éventuellement faire autre chose en attendant. Sinon, cela peut provoquer de l'anxiété car il ne sait pas si ce temps d'attente fait suite à des contraintes techniques ou à une erreur de sa part. Toujours pour la même raison, il faut éviter les temps de réponse variables.

2.5.2 Règles propres aux PDA

Après avoir pris connaissance des différents principes ergonomiques pour les interfaces en général, nous pouvons donner quelques recommandations plus appropriées aux appareils mobiles tels que les PDA.

En règle générale, la conception d'une présentation de qualité pour les appareils mobiles nécessite de planifier en fonction de l'économie plutôt que de la créativité. En effet, la taille de l'écran du PDA devient une contrainte majeure dans la conception des interfaces. L'élément clé consiste à décider quelles sont les informations qui sont essentielles, puis à les présenter d'une façon claire, concise et facilitant la navigation.

Pour obtenir des résultats efficaces, le concepteur doit respecter un certain nombre de règles dans la conception de fenêtres pour appareils mobiles et appareils de poche :

- Maximiser l'utilisation de l'écran tout en évitant d'encombrer celui-ci;
- Minimiser la longueur des fenêtres;
- Éviter les graphiques qui réduisent la lisibilité.

Il est toujours plus facile de concevoir une application pour des appareils mobiles et des PDA lorsque l'utilisateur peut citer en référence une application existante de son organisation. En gardant cela à l'esprit, voici quelques considérations visant la conception efficace d'une interface :

- Pour éviter à l'utilisateur de faire défiler le texte, il faut essayer d'adapter au maximum la longueur des « fenêtres » à la largeur de l'écran.
- Il faut également respecter une hiérarchie à plat pour la navigation. Les informations requises ne doivent jamais être structurées selon plus de trois niveaux de profondeur, sinon l'utilisateur risque de s'égarer.
- Lorsque la version PDA d'une application est sur un contenu semblable, la structure de navigation doit refléter aussi étroitement que possible la structure de l'application existante. Les utilisateurs qui connaissent bien une application existante ne doivent pas se trouver perdus lorsqu'ils explorent celle-ci au moyen d'un appareil mobile, bien que les aides à la navigation soient propres à ce type d'appareil.
- De même, la conception d'un contenu présenté sur un appareil mobile doit adopter des images semblables. Sa présentation doit ressembler à celle de l'application existante.
- Les fonctionnalités de recherche, plutôt que d'être globales, doivent être contextuelles dans la mesure du possible. Dans le cas contraire, l'utilisateur doit pouvoir effectuer un choix dans une liste prédéfinie. Une option, lui permettant de restreindre le champ de la recherche, doit lui être offerte.

- Dans le cas d'application induisant une séquence d'écran, l'insertion d'un bouton « Précédent » sur chaque écran, ayant la même fonction que dans une application ordinaire, est très importante. C'est une des fonctions les plus utilisées sur un appareil mobile.
- Pour éviter que l'utilisateur ne s'égare, il faut lui fournir un moyen lui permettant de revenir à la fenêtre d'accueil ou à la fenêtre initiale après avoir visité une autre fenêtre.
- Autant que possible, utiliser des numéros comme « input ».
- Utiliser des abréviations courantes (i.e. les codes postaux au lieu des villes).
- Si les lettres sont utilisées, il faut avoir recours à des mécanismes de raccourcis (i.e. la première lettre à la place du mot entier comme clé).
- Proposer des choix (i.e. nombres, list-box, radio-buttons, link-lists) ou une valeur par défaut quand c'est possible, sinon même des longues listes se révèlent plus utilisables de par la fonction de défilement.

Ci-dessous, deux exemples de pages Web conçues et présentées clairement :



Figure 2-10 : Exemples de modèles de pages Web exploitables par un PDA.

L'émulateur constitue une aide importante à la conception d'applications pour PDA. Il permet à l'utilisateur d'obtenir une vue, conforme à l'impression, des « fenêtres » qu'il est en train de créer, sans devoir au préalable les synchroniser sur un appareil mobile.

2.6 Applications existantes

Actuellement, il existe diverses applications offrant des services qui permettent d'analyser des séries temporelles. Certains de ces services sont accessibles via des sites Web. Ils adoptent même parfois des interfaces dont le format correspond à celui d'un écran de PDA tandis que d'autres sont directement disponibles sur PDA sous la forme d'applications à télécharger et à installer sur les mobiles.

Toutefois, il convient de noter que toutes les applications disponibles ne sont pas développées pour étudier n'importe quelle sorte de séries temporelles. En effet, la plupart des applications

disponibles sont prévues pour des séries temporelles telles que les marchés financiers (achats, ventes, évolutions de stocks,...) ou encore la météorologie.

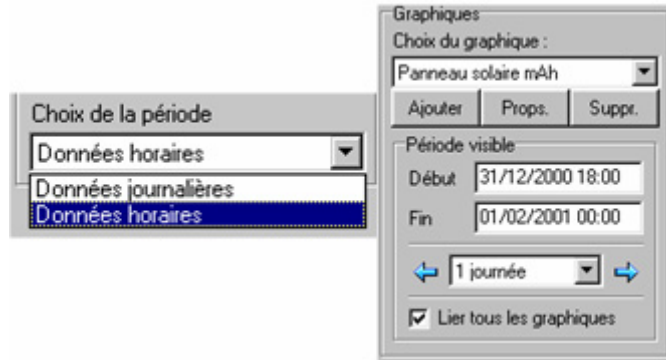
2.6.1 Applications existantes sur le Web

2.6.1.1 CimSta²



« CimSta » est un logiciel d'acquisition et de visualisation de données pour les stations Météo par Modem, GSM, RS-232 ou Boucle de courant. Il autorise l'utilisation de scripts puissants pour la récupération des données, la mise à jour de l'heure, le paramétrage, le mode Minitel, les valeurs instantanées, etc. Une visualisation des données par tableaux ou par graphiques ainsi que diverses exportations sont disponibles.

Comme annoncé précédemment, ce logiciel permet de paramétrer les intervalles qui seront utilisés pour récolter les diverses valeurs lors de chaque jour ainsi que la durée de collecte des informations.



Il est possible de visualiser les informations sous la forme d'un tableau ou sous la forme de graphiques :

	TN	TX	TM	ITN	ITX
01/08/94	16.4	24.8	20.1	05h36	12h00
02/08/94	16.3	30.1	23.6	04h36	15h42
03/08/94	20.4	34.7	27.1	01h54	15h36
04/08/94	20.8	32.6	26.2	05h30	15h00
05/08/94	21.3	28.2	23.9	03h42	13h18
06/08/94	19.8	31.6	25.4	02h30	14h48
07/08/94	21.4	31.0	24.7	08h42	15h12
08/08/94	21.0	29.6	24.9	05h30	15h06
09/08/94	21.3	28.2	23.3	02h30	10h54
10/08/94	16.5	23.8	20.0	06h24	14h42
11/08/94	16.5	22.7	18.9	01h06	09h36
12/08/94	14.1	25.4	19.6	04h48	12h30
13/08/94	16.5	24.2	18.8	07h24	13h00
14/08/94	14.0	26.9	21.0	03h00	14h42
15/08/94	16.0	28.2	22.4	04h42	15h30
16/08/94	17.2	29.2	22.9	03h18	13h18
17/08/94	17.1	24.7	20.3	05h36	12h00

Informations sur la station:
 Nom : St Julien Leoville
 Identifiant : F_STJ2
 Dep : 33 Com : 423 Num : 1
 Type : B407 AR1 255C v

Informations sur l'acquisition:
 Nombre de blocs lus : 31
 Du 01/08/1994 au 31/08/1994

Tableau:
 Choix de la période : Données journalières

Affichage sélectif de capteurs:
 BA Batterie
 HH Durée d'humectation
 RG Rayonnement global
 RR Pluviomètre
 T Température de l'air
 TS Température supplémentaire
 U Humidité
 V Anémomètre

Figure 2-11 : Visualisation sous forme d'un tableau.

² <http://www.cimel.fr>

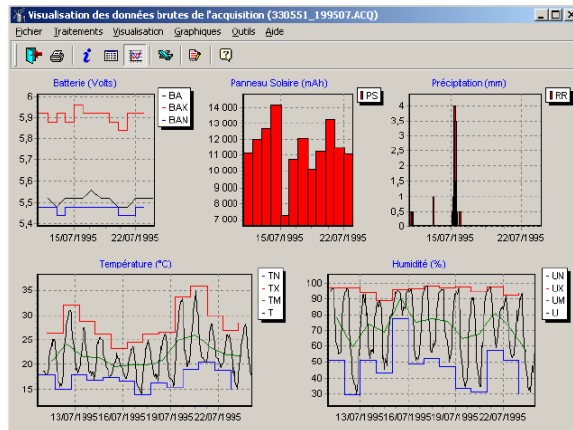


Figure 2-12 : Visualisation sous forme de graphiques.

2.6.1.2 SmartMoney³

L'application « SmartMoney » a été lancée en 1996. Les lecteurs du magazine leur adressaient des requêtes afin d'obtenir des conseils pour investir. Les éditeurs, n'ayant jamais trouvé un site Web permettant de satisfaire ces demandes, ont alors décidé de la créer eux-mêmes.

- « SmartMoney » est un manager financier personnel très complet et facile à utiliser.
- Avec « SmartMoney », il est possible d'équilibrer le budget, de maintenir à jour les cartes de crédit, de contrôler les factures et les dépôts permanents...
- « SmartMoney » est conçu pour que les débutants puissent démarrer au commencement et les investisseurs plus expérimentés puissent sélectionner et choisir.

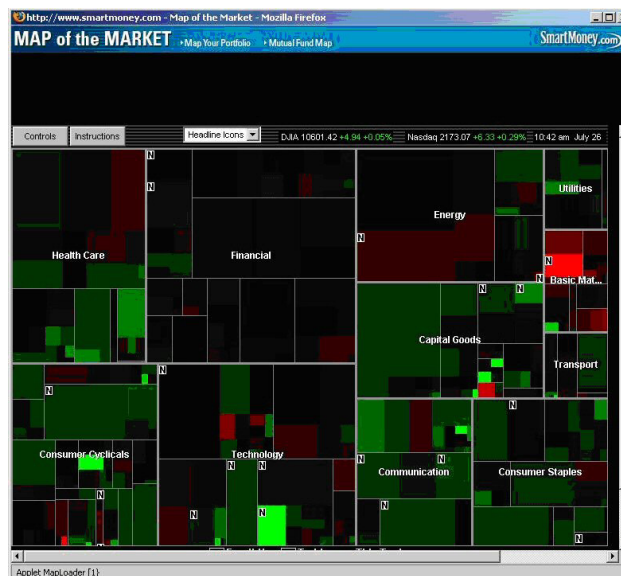


Figure 2-13 : "Map of the Market" (Application de SmartMoney).

La « Map of the Market » constitue une des visualisations graphiques développées par SmartMoney. Chaque variable, présente dans la visualisation, est basée sur la forme et la couleur. Cela permet de voir l'évolution des actions (dans ce cas-ci, les actions sont répertoriées par domaine/secteur d'activité)

³ <http://www.smartmoney.com>

2.6.2 Applications existantes sur PDA

2.6.2.1 PocketPCCharts⁴

« Pocket Chart » est un logiciel qui permet d'accéder à des données boursières et cela, en temps réel. Toutefois, un des avantages de ce logiciel est qu'il est aussi utilisable sans qu'il soit nécessaire d'être connecté à Internet. Après avoir téléchargé les données qui les intéressaient, les utilisateurs peuvent consulter et gérer les informations sans avoir besoin d'une connexion.

Comme son nom l'indique, l'application « Pocket Chart » a été conçue pour être uniquement utilisée sur des PDA's de type Pocket PC.

A l'heure actuelle, ce logiciel fait partie des meilleurs jamais conçus dans le domaine des visualisations/graphismes et de l'interaction. En effet, les visualisations représentent avec l'interactivité et l'ergonomie un des points forts de l'application. Il est tout à fait possible de passer (via un zoom) d'un intervalle de temps très court (quelques heures) à un intervalle de temps très long (jusqu'à 3 ans). De plus, chaque graphique peut être configuré entièrement de façon indépendante vis-à-vis des autres, ce qui peut offrir différentes perspectives aux utilisateurs afin d'aider ces derniers à réaliser leurs analyses.

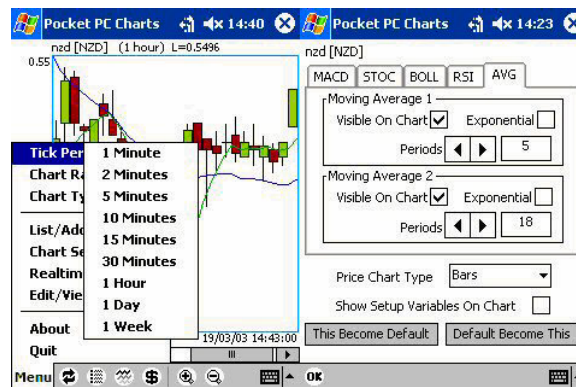


Figure 2-14 : Pocket PC Charts.

2.6.2.2 Pocket Stock Monitor⁵

« Personal Stock Monitor » (PSM) est un compagnon de bureau pour les investisseurs en ligne. En communiquant avec divers serveurs de cotation d'Internet, « Personal Stock Monitor » obtient le cours des actions d'un ou plusieurs portefeuilles d'investissements. Il suffit d'entrer les éléments boursiers pour que « Personal Stock Monitor » appelle, selon une fréquence précisée, un serveur parmi la douzaine possible, et obtienne ainsi les cours. Il les organise en vues pratiques et affiche ces éléments dans une barre défilante sur le bureau.

⁴ <http://www.pocketpccharts.com>

⁵ <http://www.pocketx.net/products/psm.html>

Les investissements peuvent être organisés en un système hiérarchisé de dossiers, d'une façon analogue à celui de l'Explorer, avec une fonction glisser-déplacer qui facilite la gestion. Lorsque l'utilisateur est à son bureau, il peut minimiser la fenêtre principale de « Personal Stock Monitor » et le laisser travailler pour lui dans l'arrière plan. Il l'informe automatiquement des événements auxquels il s'intéresse selon des règles que l'utilisateur a, préalablement, établies.



Figure 2-15 : Pocket Stock Monitor.

La technologie utilisée pour développer cette application est une technologie Java (PersonalJava). Cela permet au service d'être supporté par différents appareils équipés de la Java Virtual Machine (JVM). Il est connecté en permanence grâce à un système de streaming basé à Hong Kong, spécialement étudié pour les PDA.

Quelques fonctionnalités de Personal Stock Monitor :

- Mini-Teletex Screen : les données sont mises à jour en temps réel.
- Portfolio : Gestion d'un portfolio avec évolution, gain et perte en temps réel.

2.6.2.3 Midcast⁶

Produit de HillCast Technologies, « Midcast Pro » représente la dernière version de la société en ce qui concerne les applications (développées en Java) de traitement de données financières en temps réel pour unités mobiles. Elle permet aux courtiers, investisseurs et institutions d'accéder aux actions des marchés financiers d'une manière totalement libre car elle est conçue pour des réseaux sans fils (téléphones portables et PDA).



Figure 2-16 : Variété de produits HillCast.

« Midcast Pro » permet d'afficher aussi bien des graphiques représentant l'évolution des cours de diverses actions que des tableaux récapitulatifs de toutes les informations disponibles pour les utilisateurs.

⁶ <http://www.hillcast.com>



Figure 2-17 : MidCast PRO™ - Pocket PC.

2.7 Conclusion

Bien que certains aspects de la visualisation de l'information aient été intégrés à plus d'une application Internet et de gestion, ses incidences sur l'élaboration d'interfaces-utilisateurs sont inconnues jusqu'à ce jour. La plupart des obstacles techniques ont été franchis. La difficulté que doit maintenant surmonter la visualisation de l'information est d'être acceptée par les utilisateurs ; ces derniers étant habitués aux représentations linéaires traditionnelles de l'information. À moins que nous ne puissions créer un graphisme puissant pour structurer les métadonnées, les progrès de la recherche en visualisation passeront grandement inaperçus.

3 Qu'est-ce qu'une série temporelle ?

3.1 Définitions

L'analyse de longues séries temporelles est un domaine important de la fouille de données. En effet, il s'agit d'un cas exemplaire de données qu'il est plus facile d'accumuler en grande quantité que d'analyser. L'extraction d'informations à partir de ces données passe souvent par une étape de visualisation, celle-ci peut être rendue difficile de part le nombre de points. Toutefois, les définitions de séries temporelles sont nombreuses et peuvent se présenter soit sous une forme textuelle soit sous une forme plus mathématique.

3.1.1 Quelques définitions issues d'Internet

- Observation d'une variable faite à travers le temps.
- Une série de mesures prises à des moments consécutifs dans le temps.
- Une séquence d'observations ordonnées dans le temps et faites à intervalles réguliers.
- Un ensemble de données consistant en des valeurs et une échelle de temps. En général, l'échelle de temps est équitablement espacée (par ex : chaque jour, chaque mois,...)
- La valeur d'une variable générée successivement dans le temps. Graphiquement, une série temporelle est représentée avec le temps en abscisse et la valeur de la fonction en ordonnée.
- La distribution d'une donnée qui représente les changements consécutifs de valeur sur une période donnée. En général, les espacements entre les prises de valeur sont équitablement espacés (par ex : chaque jour, chaque mois,...)

Cependant, même si les sources divergent en ce qui concerne les définitions, il est possible d'y retrouver des termes constants : variable, changement, temps, espacement, équitablement.

3.1.2 Définitions mathématiques

- Une série temporelle est une suite d'observations indicées par le temps, la date à laquelle l'observation est faite est une information importante sur le phénomène observé.
- La suite d'observations $(y_t, t \in T)$ d'une variable y à différentes dates t est appelée série temporelle. Habituellement, T est dénombrable, de sorte que $t = 1, \dots, T$.

Remarque : en mathématiques, la définition de série temporelle ci-dessus correspond à la définition d'une suite, $\{u_n\}_{n \in \mathbb{N}}$, tandis que nous nommerons série la suite définie à partir de

la somme des termes de la suite : $S_n = \sum_{i=0}^n u_i$

Une série temporelle peut être [DUFOUR98]

- continue – Dans certains domaines (i.e. en physique), la variable X_t peut être observée de façon continue, i.e. l'indice t peut prendre toutes les valeurs dans un intervalle de nombres réels. Dans un tel cas, nous parlons de série continue. De telles situations sont rares dans les données économiques.
- discrète – Une série est discrète lorsque l'ensemble des valeurs possibles de t est un ensemble discret, i.e. T peut être considéré comme un sous-ensemble des nombres entiers. Il y a deux types importants de série discrète, suivant que les observations correspondent à des
 - niveaux : séries enregistrées instantanément (i.e., prix, stocks),
 - flux : séries cumulées sur un intervalle de temps (i.e. revenu, consommation). Lorsque nous analysons une série de flux, il est important de tenir compte de l'intervalle de temps considéré.

Une autre particularité des séries temporelles est la diversité des formats sous lesquels il est possible de trouver les données. Les séries temporelles se distinguent par la richesse de l'information temporelle qu'il est possible d'associer aux valeurs qu'elles collectent. Nous pouvons trouver :

1. Aucune information temporelle, la série est seulement une suite de valeurs ordonnées chronologiquement : nous parlerons dans ce cas de séries chronologiques.
2. Chaque valeur est associée à une date : il est possible de calculer et de comparer des durées entre les instants associés à chacune des valeurs : nous parlerons de séries datées.
3. Chacune des dates de la série est séparée de la date précédente du même intervalle de temps (par exemple la période d'échantillonnage) : nous parlerons de séries régulières.

Contrairement à l'économétrie traditionnelle, le but de l'analyse des séries temporelles n'est pas de relier des variables entre elles, mais de s'intéresser à la dynamique d'une variable [CHEVILLON04]. Cette dernière est en effet essentielle pour deux raisons :

- Les avancées de l'économétrie ont montré que nous ne pouvons relier que des variables qui présentent des propriétés similaires, en particulier une même stabilité ou instabilité ;
- Les propriétés mathématiques des modèles permettant d'estimer le lien entre deux variables dépendent de leur dynamique.

Une série temporelle est donc toute suite d'observations correspondant à la même variable : il peut s'agir de données macroéconomiques (le PIB d'un pays, l'inflation, les exportations...), microéconomiques (les ventes d'une entreprise donnée, son nombre d'employés, le revenu d'un individu, le nombre d'enfants d'une femme...), financières (le CAC40, le prix d'une option d'achat ou de vente, le cours d'une action), météorologiques (la pluviosité, le nombre de jours de soleil par an), politiques (le nombre de votants, de voix reçues par un candidat...), démographiques (la taille moyenne des habitants, leur âge...). En pratique, tout ce qui est chiffrable et variable en fonction du temps.

3.1.3 Caractéristiques

La dimension temporelle est, ici, importante car il s'agit de l'analyse d'une chronique historique : des variations d'une même variable au cours du temps, afin de pouvoir en comprendre la dynamique. Les données de panel s'intéressent pour leur part à la variabilité de caractéristiques entre individus, agents, entreprises.

La périodicité de la série, en revanche, n'importe pas : il peut s'agir de mesures quotidiennes, mensuelles, trimestrielles, annuelles... voire même sans périodicité. Nous représentons en général les séries temporelles sur des graphiques de valeurs (ordonnées) en fonction du temps (abscisses).

Une telle observation constitue un outil essentiel qui permet au modélisateur ayant un peu d'expérience de tout de suite se rendre compte des propriétés dynamiques principales, afin de savoir quel test statistique pratiquer.

La statistique se préoccupe de porter des jugements sur une population à partir de l'observation d'un échantillon de cette population. La plupart du temps l'ordre dans lequel sont échantillonnées les observations n'a pas d'importance. L'exemple le plus simple que nous puissions prendre est celui des sondages d'opinions.

L'analyse des séries temporelles est très différente de l'analyse statistique habituelle car l'ordre des observations revêt ici une importance primordiale. Une série temporelle est définie comme une suite d'observations indexées par le temps. Nous pouvons prendre comme exemple en économie une série de prix, de taux d'intérêt, etc. Mais nous pouvons trouver bien d'autres exemples dans les autres disciplines.

Nous avons le sentiment en observant un graphique de ces séries que la valeur prise au temps t dépend fortement de la valeur prise au temps $t-1$. Le processus qui les engendre est dynamique. Nous voulons en construisant un modèle, acquérir de l'information sur ce processus théorique que nous appelons Processus de Génération des Données ou PGD.

Le problème est alors de trouver le modèle pratique qui se rapprochera le plus possible du processus théorique et ensuite de l'estimer. Une fois cette étape franchie, nous pourrons faire de la prévision ou du contrôle avec ce modèle.

3.2 Quels sont les buts d'une série temporelle ?

Le but poursuivi est la formulation d'un modèle statistique. Il est une représentation congruente du processus stochastique (inconnu) qui a généré la série observée. Tout comme en probabilités/statistiques, il faut bien comprendre la différence entre

- le processus sous-jacent qui génère des données (data generating process),
- sa réalisation telle que nous l'observons sur l'échantillon historique à notre disposition,
- les futures réalisations et le modèle que nous construisons afin de tenter de le représenter.

Par représentation congruente, il s'agit d'un modèle qui soit conforme aux données sous tous les angles mesurables et testables.

Parmi les multiples applications de l'analyse des séries temporelles, il est possible d'en distinguer neuf principales [CHEVILLON04] :

3.2.1 Prévoir

La fonction première pour laquelle il est intéressant d'observer l'historique d'une variable vise à en découvrir certaines régularités afin de pouvoir extrapoler et d'établir une prévision. Il s'agit ici de comprendre la dynamique qui relie une observation à celles qui l'ont précédées et de supposer, sous réserve que nous puissions justifier une telle hypothèse, que les mêmes causes produisent les mêmes effets. Avec une analyse fine, il est même possible d'établir des prévisions "robustes" vis-à-vis de ruptures brusques et de changements qui ne peuvent être anticipés.

3.2.2 Relier les variables

Il est important de savoir a priori si certaines relations sont "économétriquement" possibles et d'éviter les équations qui ne présentent aucun sens. Prenons pour exemple : soit la demande en valeur à la date t , notée D_t , et l'inflation notée i_t . Pouvons-nous faire l'hypothèse que l'inflation influence positivement la demande ? Ce qui reviendrait à dire qu'en période de forte inflation, les citoyens souhaitent consommer davantage qu'une période où elle est faible. Ce que nous pouvons noter :

$$d_t = \alpha + \beta i_t + \varepsilon_t$$

Où ε_t représente un écart entre la demande et ce que peut prévoir l'inflation. Si notre modèle représente bien la manière dont est générée la demande, ε_t doit être de moyenne nulle.

Dans ce cas, si nous notons $E[D_t]$ l'espérance mathématique, i.e. la moyenne, celle-ci doit pouvoir satisfaire :

$$\begin{aligned} E[D_t] &= E[\alpha + \beta i_t + \varepsilon_t] \\ \Leftrightarrow E[D_t] &= \alpha + \beta E[i_t] + E[\varepsilon_t] \\ \Leftrightarrow E[D_t] &= \alpha + \beta E[i_t] \end{aligned}$$

Car $E[\varepsilon_t] = 0$. Or nous avons vu sur la graphique que l'inflation avait tendance à ne pas trop s'éloigner de 2%, sa moyenne doit donc être constante et se situer entre 1 et 3%, mettons 2% pour simplifier. Dans ce cas $E[D_t] = \alpha + \beta \times 2$ est constante, ce qui est contradictoire avec notre observation précédente qui montrait que la demande en valeur était monotone et donc que sa moyenne variait au cours du temps. Une relation comme $d_t = \alpha + \beta i_t + \varepsilon_t$ n'a donc aucun sens ; en revanche, il est statistiquement possible qu'il faille s'intéresser au lien entre l'inflation et le taux de croissance de la demande. L'analyse des séries temporelles permet de savoir quelles équations sont a priori grotesques.

Expliquer le niveau ou parfois la variance du niveau, par des modèles à peu de paramètres

- Modèle sans variable explicative

$$Y_t = f(Y_{t-1}, Y_{t-2}, \dots) + u_t$$

- Modèle avec variable explicative

$$Y_t = f(X_t) + u_t$$

où u_t est une erreur

- statique : X_t ne contient pas de valeurs passées de $\{Y_t\}$ et les u_t sont non corrélés entre eux
- dynamique : les u_t sont auto-corrélés ou X_t contient des valeurs retardées de Y_t

3.2.3 Déterminer la causalité

Une approche dynamique permet aussi de s'intéresser aux relations de causalité. Pour qu'un mouvement en provoque un autre, il est nécessaire qu'il le précède. Une simple concomitance de deux événements révèle davantage une source commune. L'utilisation de retards d'une variable, i.e. de ses valeurs aux périodes précédentes, dans les équations autorise la mesure des effets de causalité et permet également de connaître la durée de transmission entre une source et son effet.

3.2.4 Distinguer entre court et long terme

Certaines lois de comportement ne sont jamais vérifiées en pratique car elles ne s'appliquent que sur les équilibres de long terme. A plus courte échéance, des variations contrarient perpétuellement leur mise en oeuvre. Cependant, des ajustements transitoires s'opèrent continuellement afin de s'approcher de ces équilibres. Nous reformulerons alors le modèle sous la forme d'un mécanisme dit de correction d'équilibre (ou d'erreur), selon lequel un écart (une erreur) positif par rapport à l'équilibre de long terme entraîne une variation de court terme négative, afin de réduire cet écart.

3.2.5 Etudier des anticipations des agents

Comment prendre en compte les anticipations des agents ? Dans une décision entre épargne et consommation, ce ne sont pas seulement les revenus actuels et passés qui comptent, mais aussi l'idée que nous nous faisons de l'avenir. Il faut donc, dans certaines équations, faire intervenir des valeurs avancées des variables, via leurs anticipations en utilisant la manière dont celles-ci ont été formées dans le passé.

3.2.6 Repérer les tendances et cycles

Des méthodes dynamiques repèrent des tendances mouvantes des données. Par différence, l'écart entre le niveau de la variable (localement monotone) et la position de sa tendance est en moyenne nul : il repère la position dans le cycle. Selon le modèle de tendance utilisé, il est possible d'analyser les interactions entre diverses variables afin d'atteindre un équilibre entre méthodes économétriques et purement statistiques.

3.2.7 Corriger des variations saisonnières

Comme constaté sur les exemples introductifs, la série de demande présente des variations régulières trimestrielles que nous avons nommées « variations saisonnières ». Celles-ci peuvent être stables au cours du temps et ainsi l'écart entre les premiers et deuxième trimestre sera le même en 1991 et en 2004. En retirant cet effet habituel et en lissant la série, il est alors possible de comparer le niveau entre ces années. La correction des variations saisonnières (cvs) devient plus complexe quand les comportements évoluent davantage ; l'écart entre deux trimestres consécutifs peut se modifier et la série cvs apportera alors une information supplémentaire.

3.2.8 Détecter les chocs structurels

Un choc structurel est défini comme une modification permanente ou temporaire de la façon dont est générée une variable. Ils sont fréquents, souvent difficiles à anticiper et à mesurer. Il est cependant essentiel de savoir qu'une telle rupture a eu lieu car sa présence change les interactions et équilibres, souvent radicalement. L'ignorer engendre alors des effets contraires aux buts poursuivis.

3.2.9 Contrôler les processus

Lorsqu'une autorité fixe librement le niveau d'une variable ayant une forte influence sur le reste de l'économie, comme par exemple le taux d'intérêt directeur sur lequel la banque centrale a autorité, il doit à la fois quantifier l'ampleur de son impact et mesurer la durée de transmission de son effet dans l'économie. En retour, cette autorité peut prendre en compte son propre comportement afin d'anticiper les évolutions d'une variable cible, comme l'inflation.

3.3 Que pouvons-nous retirer des séries temporelles ?

Pour atteindre les buts cités plus haut, il est nécessaire que des patterns soient observés à partir des séries temporelles et décrits de manière plus ou moins formelle. La plupart des patterns de séries temporelles peuvent être décrits avec 2 composants de base : la tendance et la saisonnalité.

Il est classique de décomposer une série temporelle $\{Y_t, t = 1, \dots, n\}$ en tendance m_t (trend), effet saisonnier s_t , erreur U_t . Les modèles développés peuvent être soit additifs soit multiplicatifs :

- modèle additif : $y_t = m_t + s_t + U_t$ où $E(U_t) = 0$
- modèle multiplicatif : $y_t = m_t s_t U_t$ où $E(U_t) = 1$

Par exemple, les séries montrant une saisonnalité qui a de plus en plus d'ampleur (cas des ventes de champagne), sont souvent mieux ajustées par un modèle multiplicatif que par un modèle additif.

3.3.1 Analyse des tendances

Nous parlerons de tendance linéaire lorsque la série peut se décomposer en

$$x_n = an + b + e_n \text{ où } n = 1, 2, \dots$$

Plus généralement, nous parlons de tendance polynômiale lorsque la série peut se décomposer en

$$x_n = a_1 n^p + a_2 n^{p-1} + \dots + a_{p+1} + e_n \text{ où } n = 1, 2, \dots$$

expression dans laquelle e_n est un résidu où ne figure plus la tendance et qui, de ce fait, a une allure relativement homogène dans le temps. De même nous pourrions définir des tendances logarithmiques, exponentielles, etc.

La tendance peut être multiplicative dans certaines séries : $x_n = t_n e_n$ où $n = 1, 2, \dots$

Et où t_n prend l'une des formes (linéaire, polynômiale, etc.) évoquées plus haut. C'est alors le logarithme des données (si elles sont positives !) qui présente une tendance additive.

3.3.2 Analyses de la saisonnalité

Nous parlerons de composante périodique, ou de périodicité, lorsque la série se décompose en

$$x_n = s_n + e_n \text{ où } n = 1, 2, \dots$$

où s_n est périodique (c'est à dire $s_{n+T} = s_n$ pour tout n , où T est la période, supposée entière) et où e_n est un résidu non-périodique et sans tendance.

Par exemple $s_n = \cos \frac{n\pi}{6}$ a une période $T = 12$.

Lorsque la période T est de 6 mois ou un an, comme c'est le cas dans beaucoup de phénomènes socio-économiques, nous avons plutôt l'habitude de parler de saisonnalité ou de composante saisonnière.

Comme précédemment, il arrive que nous ayons recours à un modèle multiplicatif.

3.4 Exemples

Les cas dans lesquels nous pouvons utiliser les séries temporelles pour produire des analyses sont très nombreux. En effet, les séries temporelles trouvent leur utilisation dans des domaines très variés (la finance, la météorologie, l'industrie, la production ou encore la criminalité).

Exemples classiques

Voici toute une liste de cas où les séries temporelles ont été utilisées [ARAGON04]:

- Le nombre de morts par accident est une série où la saisonnalité contribue à expliquer le niveau. Le niveau moyen reste stable et il y a des fluctuations saisonnières. Un service de santé publique peut vouloir prédire le nombre de morts pour chacun des 6 prochains mois de manière à voir où et quand faire une campagne de prévention, mais il peut aussi vouloir une vision synthétique de la situation, un aperçu de la tendance sur l'année.
- Les ventes de champagne et les ventes de vin australien montrent une saisonnalité qui contribue à expliquer le niveau. La moyenne et la variabilité de ces séries augmentent avec le temps. Ce sont des séries hétéroscédastiques (c'est-à-dire à variance non constante) dont la variance à une date est fonction de la moyenne à cette date. La chambre de commerce d'une région viticole peut être intéressée par l'examen de la tendance des ventes débarrassée de ce qui peut se passer à court terme alors qu'un syndicat de transporteurs a besoin de savoir combien de bouteilles devront partir de la région le mois prochain.

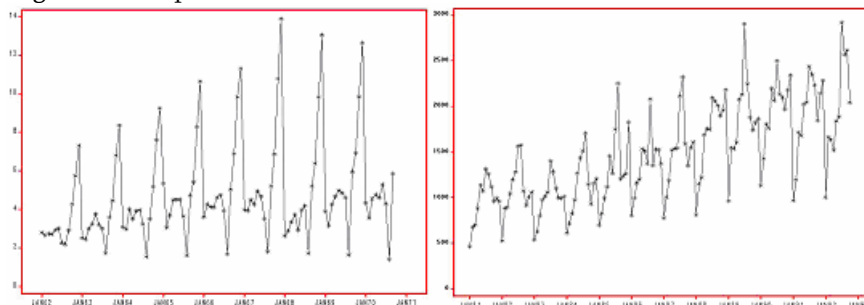


Figure 3-1 : Nombre de morts par accident et vente de vin.

- Le rendement du titre Alcatel est une série hétéroscédastique dont la variabilité change au cours du temps de façon non significative. Le rendement d'une action est

très souvent de moyenne nulle et les rendements à 2 dates consécutives souvent non corrélés. Il n'y a donc pas de problème de prévision du rendement mais la prévision de sa variabilité est utile.

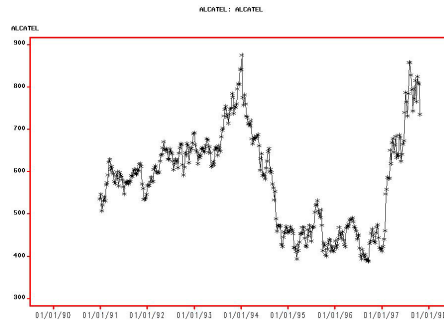


Figure 3-2 : Rendement du titre Alcatel.

Nous voyons sur ces exemples que nous ne pouvons pas, pour une série temporelle, parler d'observations indépendantes. Par conséquent, tous les calculs et raisonnements de statistique mathématique basés sur l'indépendance entre les observations doivent être revus. La dépendance temporelle permet d'améliorer la prévision de la valeur d'une série à une date future connaissant le présent et le passé.

Sur la figure suivante, quatre graphiques montrent des séries ayant des propriétés différentes :

- Le panneau (a) présente deux séries qui oscillent autour d'une valeur comprise entre 0 et 0,01 : elles sont stables autour de leur moyenne. Nous parlons dans ce cas de séries stationnaires.
- En (b) , l'inflation décroît fortement jusqu'en 1999 pour remonter ensuite : elle n'oscille pas autour d'une moyenne bien qu'elle ne soit jamais très loin de 2%; sous réserve de tests statistiques fins, elle semble moins stable que les séries en (a) , elle est donc peut-être non stationnaire.
- La série (c), quant à elle croît sur l'ensemble de l'échantillon observé ; nous parlons dans ce cas de tendance et sa moyenne n'est pas constante (sa moyenne entre 1990 et 1995 est radicalement différente de celle mesurée entre 2000 et 2004).
- Enfin, le panneau (d) reproduit la même série mais hors effet de prix (l'inflation étant toujours positive les prix croissent naturellement sans que les volumes augmentent nécessairement) : la tendance à la hausse provenait uniquement de l'augmentation des prix et la demande en volume décroît au début des années 1990.

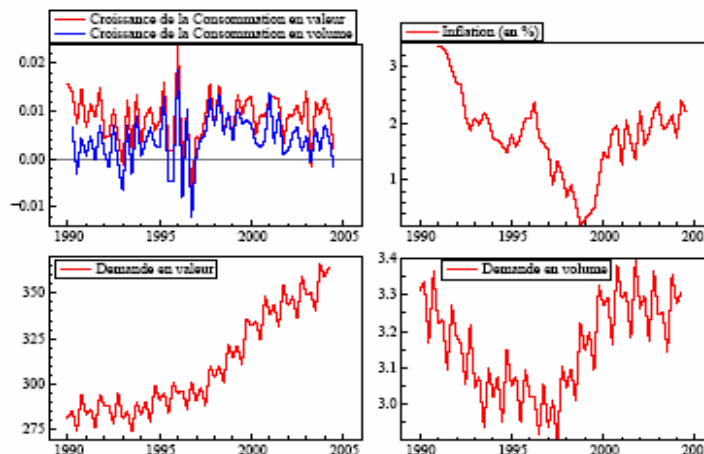


Figure 3-3 : Séries temporelles aux propriétés différentes.

Autres exemples

Ces exemples sont tirés de bibliothèques se trouvant sur Internet et sont classés par domaine. Ils ne représentent qu'un échantillon de ceux qui existent.

CRIME
Monthly Boston armed robberies Jan.1966-Oct.1975 Deutsch and Alt (1977). Hyde Park purse snatchings in Chicago 28 day periods; Jan'69 - Sep '73 Hay. Monthly Minneapolis public drunkenness intakes Jan.'66-Jul'78. Natural logarithms: monthly Boston armed robberies.

Tableau 3-1 : Exemples de séries temporelles (Criminalité).

DEMOGRAPHY
Percent of Men with full beards, 1866 - 1911. Daily total female births in California, 1959. Accidental deaths in USA: monthly. 1973 - 1978. Annual immigration into the United States: thousands. 1820 - 1962. U.S. population at 10 year intervals. 1790 - 1980. Standardized mortality per 1000 persons, England, 1866 to 1911.

Tableau 3-2 : Exemples de séries temporelles (Démographie).

FINANCE
Dow Jones utility index Aug 28-Dec 18 '72 (Wall Street Journal). IBM common stock closing prices: daily, 17th May 1961 - 2nd November 1962. Daily closing price of IBM stock, Jan. 01 1980 - Oct. 08 1992. Implicit price index, U.S., 1890 - 1974. The value of SP500, June 06, 1960 to June 06, 2000. Annual common stock price, U.S., 1871 to 1970.

Tableau 3-3 : Exemples de séries temporelles (Finance).

METEOROLOGY
Annual snowfall in Buffalo, 1910-1972. CO2 (ppm) mauna loa, 1965-1980. Volcanic dust veil index, northern hemisphere, 1500-1969. Annual rainfall at fortaleza, Brazil, 1849-1979. Surface air "temperature change" (degrees Celsius) for the globe, 1880 - 1985. Monthly degree days per heating in Chicago, Feb. 1931 - Aug. 1977.

Tableau 3-4 : Exemples de séries temporelles (Météorologie).

PRODUCTION
Monthly basic iron production in Australia: thousand tonnes. Jan 1956 - Aug 1995. Monthly Australian beer production: Jan 1991 - Aug 1995 Monthly production of clay bricks: million units. Jan 1956 - Aug 1995. US motor vehicle production and percent change in US GNP. 1950 - 1987. Monthly production of Portland cement: thousands of tonnes. Jan 1956 - Aug 1995. Monthly production of chocolate confectionery in Australia: tonnes. July 1957 - Aug 1995.

Tableau 3-5 : Exemples de séries temporelles (Production).

4 Analyse des besoins

4.1 Introduction

Depuis quelques années, nous avons pu assister à l'émergence d'appareils d'un genre nouveaux grâce aux systèmes embarqués. Le but de ces appareils, dont le PDA fait partie, est de pouvoir offrir aux utilisateurs un large éventail de possibilités quant aux applications qu'ils développent.

De nombreuses applications ayant pour sujet les séries temporelles ont déjà été développées sur ordinateurs. Elles permettent de visualiser des séries de tout genre dans le but d'en retirer des analyses et de prévoir leur évolution. Comme nous l'avons dit, ces séries temporelles peuvent être de tout type et représenter une palette de domaines très variés. Cela peut aller du marché boursier à la météorologie en passant par des statistiques de vente, d'agriculture et bien d'autres...

Dans le cas qui nous concerne, il est évidemment possible d'associer au PDA ces séries temporelles et la consultation de leurs informations. En effet, pouvoir contrôler l'évolution du cours de ses actions peut s'avérer particulièrement intéressant pour les personnes possédant un portefeuille d'actions. Cela, quel que soit l'endroit, la position (que ce soit en marchant dans la rue ou en étant assis chez soi) ou encore le moment de la journée. Ils pourront ainsi être informés des variations du marché en temps réel.

Pour un chef de projet, pouvoir connaître et manipuler à n'importe quel moment des données de productivité (telles que des quantités de production nécessaire à la réalisation d'un produit) peut s'avérer une information utile. Cela facilite la vie de celui qui s'en sert.

Cependant, une application développée ne rencontrera le succès escompté que s'il existe des utilisateurs prêts à s'en servir. Il est important de se poser la question de l'utilité de cette application et cela, avant même de débiter les phases de conception et d'implémentation. C'est dans ce but qu'est réalisée cette étude afin de constater s'il existe bien un réel besoin de la part des utilisateurs, de développer une nouvelle application.



Figure 4-1 : Quelques modèles de PDA.

4.1.1 Qu'est-ce qu'une analyse des besoins ?

Un produit interactif peut être développé pour de multiples raisons. Il peut, et c'est souvent le cas, répondre à un besoin détecté par une ou plusieurs personnes. Il peut être aussi le résultat d'une passion dans un domaine donné.

Dans ce cadre, une analyse de besoins est, comme l'expression l'indique, une manière de s'interroger de façon structurée sur les besoins en tenant compte de tout l'apport potentiel du produit dans ce domaine.

En fait, l'analyse des besoins constitue logiquement l'origine des réflexions sur la conception. Immédiatement, elle soulève de nombreuses questions auxquelles il va falloir apporter des réponses, même provisoires.

Analyser les besoins en conception, c'est tenter un recensement des besoins en interrogeant divers acteurs. C'est aussi une manière d'analyser ses pratiques. Nous pouvons ainsi soulever, dans le cadre de l'analyse, certaines problématiques qu'il faudra prendre le temps de clarifier.

Si nous ne voulons pas que les produits développés restent dans un tiroir, l'analyse des besoins est le préalable indispensable à toute démarche de développement.

L'analyse des besoins débouche sur les conclusions reprises dans le cahier des charges du produit ; c'est en partie cette analyse qui justifie la raison du produit.

Les pratiques et les procédés auxquels il est possible d'avoir recours pour mener une analyse des besoins sont extrêmement divers. Les pratiques sont différentes d'une institution à une autre. Elles sont différentes au sein même d'une institution.

Une analyse des besoins peut représenter :

- Un simple recensement de demandes auprès des utilisateurs. Leurs demandes peuvent être spontanées ou influencées par une politique ouverte ou fermée au produit qui sera développé.
- Une enquête circonstanciée réalisée avec le concours de l'ensemble des acteurs concernés.

4.1.2 Pourquoi une analyse des besoins ?

A la suite de nombreux échecs, certains domaines ont reconnu l'analyse des besoins comme une phase déterminante du processus de conception. C'est le cas de la conception de systèmes informatiques, et, plus largement, de la gestion de projet.

Plusieurs études ont montré que la phase d'analyse des besoins est la source de la majorité des erreurs de conception des applications informatiques.

Une étude a montré que si la phase d'analyse des besoins occupe en moyenne 2% du temps total de conception, elle est à l'origine de 56% des erreurs commises dans la conception des produits. Or cette analyse représente un coût insignifiant par rapport aux autres phases comme le développement, l'implémentation, les tests, ...

Pourtant, le coût relatif de correction d'erreurs dues à une mauvaise analyse des besoins est croissant à mesure de l'avancement du processus global.

Enfin, plus une erreur d'analyse des besoins est détectée tardivement, plus le coût induit sera élevé.

En informatique, les méthodes et outils pour analyser le besoin des utilisateurs constituent aujourd'hui un domaine en soi : l'ingénierie des besoins. Celui-ci possède ses méthodes et outils propres, sa recherche et ses publications spécifiques.

4.2 Les utilisateurs

4.2.1 Types d'utilisateur

L'avantage de cet outil, qui faut-il le rappeler est générique puisqu'il permet une visualisation de séries temporelles de tous horizons, est qu'il peut se retrouver dans les mains de toute une gamme d'utilisateurs. Parmi les utilisateurs qui consultent des séries temporelles sur PDA, nous pouvons tout aussi bien retrouver des utilisateurs lambda que des personnes utilisant cette application dans un but professionnel.

Ne serait-il pas intéressant pour un courtier, un contremaître ou encore un météorologue de connaître des informations relatives à son domaine professionnel sans pour autant se trouver en face de son ordinateur dans un bureau ?

A partir de cela, nous pourrions définir une infinité de catégories d'utilisateurs si nous nous basons sur leur domaine professionnel et ainsi définir l'environnement de l'application [NOIRHOMME04]. Cependant, il est plus intéressant de les classer en utilisateur « lambda » et utilisateur « expert ».

4.2.2 Caractéristiques de l'utilisateur

4.2.2.1 Utilisateurs « lambda »

Attributs physiques :

- Age :

L'âge de l'utilisateur « lambda » n'est pas déterminant pour l'utilisation de l'outil générique car il ne visualisera les séries temporelles que pour son information personnelle. L'âge sera bien souvent fonction de la série temporelle qui sera analysée. Il est fort peu probable de voir un individu mineur s'intéresser aux fluctuations du marché boursier si celui-ci n'est pas apte à pouvoir détenir un portefeuille d'actions. Par contre, il n'existe pas de limite supérieure d'âge pour manipuler l'application.

- Incapacités physiques (handicaps) :

Exceptés les 2 handicaps qui peuvent rendre l'utilisation de l'application sur un PDA plus compliquée, à savoir un handicap visuel (pour l'écran) et un handicap manuel (pour le stylo), aucun autre handicap physique ne peut empêcher d'utiliser l'application en question.

Toutefois, même en cas de handicap visuel (pour autant qu'il ne corresponde pas à une cécité), des solutions logicielles et des méthodes de visualisations spéciales existent afin de faciliter l'utilisation de l'application. Dans le cas d'un handicap manuel (par exemple des tremblements), l'utilisation du stylo risque d'être rendue beaucoup plus difficile voire impossible.

- Sexe :

Le sexe n'est pas à proprement parler un critère dans l'utilisation de l'application, aussi bien les hommes que les femmes sont aptes à l'utiliser. Aucune théorie ne permet de dire que les hommes seront plus nombreux à utiliser l'application. Encore une fois, cela dépendra certainement du champ d'application de la personne qui analyse la série temporelle.

Attributs mentaux :

- Capacités particulières :

Cette application ne nécessite en aucune manière des capacités particulières de la part des utilisateurs. La seule capacité qui est requise pour l'utiliser est celle de pouvoir se représenter mentalement les représentations affichées ainsi que leur signification.

- Handicap mental :

Il est préférable que les utilisateurs ne souffrent d'aucun handicap mental pour leur permettre une utilisation adéquate de l'application.

- Motivation :

Nous sommes en droit d'attendre que la motivation des personnes qui utilisent l'application soit positive. En effet, elle leur permettra, où qu'ils se trouvent et à n'importe quel moment de la journée, de se renseigner, d'obtenir et de consulter diverses informations.

Qualifications et connaissances :

- Expérience avec l'ordinateur et le PDA :

De nos jours, en plus de posséder un ordinateur au bureau, de nombreuses familles en disposent à domicile. Ce qui fait que peu de personnes sont perdues lors de son utilisation. Les compétences des utilisateurs avec l'ordinateur peuvent donc aller du simple utilisateur débutant à l'utilisateur plus expérimenté.

Contrairement à l'ordinateur, le PDA est un outil que moins de personnes ont à disposition. Cependant, comme son fonctionnement ne diffère pas vraiment de celui d'un ordinateur classique, cette lacune est rapidement comblée.

Pour la combinaison des informations et de l'ordinateur, le meilleur moyen de trouver de l'information sur les séries temporelles (météorologie, marchés financiers, etc.) en temps réel et à n'importe quel moment de la journée est encore Internet. Bien que d'autres sources soient possibles (journaux financiers, télévision, radio,...), celles-ci ne sont pas mises à jour à la même fréquence qu'Internet. Il est donc plausible de penser que les utilisateurs sont familiarisés aux logiciels graphiques.

- Qualifications générales :

Pour les utilisateurs de type utilisateur « lambda », tous les types de qualifications sont valables. Il ne faut pas oublier que ce type d'utilisateur ne maîtrise pas forcément les domaines relatifs aux séries temporelles qu'il consulte mais qu'il possède juste quelques principes basiques.

Pour ce qui est de l'utilisateur « expert », nous sommes en droit d'attendre qu'il maîtrise le domaine dans lequel il effectue une analyse.

- Connaissances des langues :

Etant donné que les utilisateurs sont pour la majorité francophones et que le français est la langue maternelle, cela devrait être la langue la plus utilisée. Toutefois, nous ne pouvons nier l'importance de la langue anglaise au niveau de la technologie. Sa connaissance constitue donc un avantage indéniable pour la maîtrise de l'application elle-même.

Caractéristiques de travail :

- Fonction :

L'utilisateur « lambda » souhaite avoir à sa disposition un outil lui permettant de consulter, visualiser et analyser des séries temporelles mais seulement à titre d'information.

L'utilisateur « expert », lui, a besoin d'un outil qui lui permet de traiter des informations en relation avec son domaine professionnel.

Initialement, les fonctions seront donc identiques pour les 2 catégories d'utilisateurs. Cependant, certaines visualisations sont mises à disposition pour aider l'utilisateur expert à suivre plus en détail l'évolution des différentes séries.

- Flexibilité :

Lorsque nous nous plaçons au niveau de l'application, les séries temporelles peuvent être consultées à tout moment selon les besoins de l'utilisateur. En effet, indépendamment du fait que la mise à jour de ces données doit se faire dans un laps de temps défini selon les caractéristiques de chaque série, l'application permet à l'utilisateur de ne pas être contraint de respecter ces horaires pour obtenir une information.

4.2.2.2 Utilisateurs « experts »

Attributs physiques :

- Age :

Tout comme l'utilisateur « lambda », l'utilisateur « expert » n'a pas besoin d'appartenir à une catégorie d'âge bien précise pour vouloir consulter ou effectuer des opérations sur les séries temporelles.

- Incapacités physiques (handicaps) :

Les handicaps physiques, que nous soyons « lambda » ou « expert », poseront toujours les mêmes problèmes aux utilisateurs. Il est donc préférable de ne pas souffrir de handicaps visuels ou manuels.

- Sexe :

Le sexe n'est pas à proprement parler un critère dans l'utilisation de l'application, les hommes, tout comme les femmes, sont aptes à l'utiliser. Aucune théorie ne permet de dire que les hommes seront plus nombreux à utiliser l'application. Encore une fois, cela dépendra certainement du champ d'application de la personne qui analyse la série temporelle.

Attributs mentaux :

- Capacités particulières :

Les visualisations étant les mêmes pour tous les utilisateurs, aucune capacité supplémentaire n'est demandée par rapport aux utilisateurs « lambda ». Cependant, les utilisateurs « expert » qui ont l'habitude de consulter les informations des séries temporelles sont habitués aux différentes sortes de visualisations. Ils possèdent donc plus de connaissances que les utilisateurs « lambda ».

- Handicap mental :

Il est préférable que les utilisateurs ne souffrent d'aucun handicap mental pour leur permettre une utilisation adéquate de l'application.

- Motivation :

La motivation de ces utilisateurs « expert » ne peut qu'être positive car ils pourront se renseigner de manière beaucoup plus aisée.

Qualifications et connaissances :

- Expérience avec l'ordinateur et le PDA :

Les utilisateurs « expert » vont, comme leur catégorie l'indique, consulter plus fréquemment l'évolution des séries temporelles. Ils sont donc habitués à utiliser un ordinateur et à se rendre sur des sites relatant ce genre d'informations. Les compétences de ce type d'utilisateur avec l'ordinateur sont donc généralement plus affinées. En ce qui concerne le PDA, aller plus

souvent consulter les informations ne signifie pas pour autant qu'ils le font au moyen d'un PDA. Toutefois, les plus férus de technologie et de mobilité auront l'occasion d'en utiliser un. Ici aussi, les utilisateurs sont normalement familiarisés aux logiciels graphiques.

- Qualifications générales :

Aucune qualification particulière n'est requise pour pouvoir se servir de ce type d'applications graphiques, cependant, les utilisateurs « expert » ont une approche plus complète des domaines analysés et de leur fonctionnement sans pour autant en connaître tous les mécanismes et subterfuges.

- Connaissances des langues :

Le niveau de connaissance des langues ne doit pas nécessairement être distinct par rapport aux utilisateurs « lambda ». Le français, et l'anglais pour certaines actions, seront les langues utilisées.

Caractéristiques de travail :

- Fonction :

Basiquement, l'utilisateur « expert » doit pouvoir effectuer les mêmes opérations sur les séries temporelles en général, comme pouvoir consulter les informations générales et plus en détail sur certains paramètres. Dans cette optique, certaines visualisations sont mises à sa disposition pour l'aider à suivre l'évolution des différentes séries.

- Flexibilité :

Lorsque nous nous plaçons au niveau de l'application, les séries temporelles peuvent être consultées à tout moment selon les besoins de l'utilisateur. En effet, indépendamment du fait que la mise à jour de ces données doit se faire dans un laps de temps qui est défini selon les caractéristiques de chaque série, l'application permet à l'utilisateur de ne pas être contraint de respecter ces horaires pour obtenir une information.

4.3 Environnement de travail

Que les utilisateurs soient des utilisateurs « lambda » ou « expert », ils utiliseront le produit dans un environnement qui sera plus ou moins identique à chaque fois. Il n'est donc pas nécessaire de développer la partie environnement pour chacun d'entre eux et dans ce qui suit, la description de l'environnement sera valable pour chaque type d'utilisateur.

4.3.1 Localisation du produit

L'avantage du PDA (par rapport à l'ordinateur) est que celui-ci peut être utilisé et emmené partout avec soi. Toutefois, afin d'assurer une utilisation optimale du produit, notamment au niveau de la lecture des informations sur l'écran, il est préférable de se trouver dans un endroit éclairé.

Un PDA peut être utilisé de plusieurs façons :

- Soit le PDA ne dispose pas d'une technologie sans fils et dans ce cas, il peut être relié à un ordinateur (celui-ci est connecté à Internet) dans l'optique de se synchroniser avec cet ordinateur. C'est un moyen pratique et courant de rapatrier les informations nécessaires que contient l'ordinateur vers le PDA. De cette façon, l'utilisateur peut mettre à jour son agenda, ses mails, ses documents et bien d'autres applications encore...
- Soit le PDA dispose d'un accès à Internet grâce à la technologie sans fils (telle que la norme Bluetooth) et dans ce cas, l'utilisateur peut directement rapatrier les informations qu'il souhaite consulter à l'endroit même où il se trouve.

4.3.2 Position

Le PDA a justement été conçu pour pouvoir être utilisé dans plusieurs positions sans pour autant diminuer l'efficacité de celui-ci. Il peut tout aussi bien être utilisé dans la rue en marchant qu'en étant assis au bureau par exemple. Cependant, du point de vue de la position, quelques contraintes propres au PDA telles que la sensibilité de l'écran et le besoin de précision du stylo font qu'il est préférable de l'utiliser en étant assis.

4.3.3 Matériel

L'application sera exécutée (de façon stricte) sur un PDA de type Pocket PC.

Un stylo sera fourni pour accompagner le PDA et servira aux interactions entre l'utilisateur et l'application. Pour résoudre les problèmes d'interaction liés au stylo (voir Chapitre VI), un clavier infra-rouge peut être éventuellement utilisé.

4.3.4 Logiciels et système d'exploitation

Pour l'utilisateur (qu'il soit « lambda » ou « expert ») qui souhaite synchroniser son PDA avec son ordinateur afin d'exécuter l'application, il faudra utiliser un logiciel de synchronisation qui sera compatible avec le Pocket PC mis à disposition. Dans le cas présent, le logiciel ActiveSync fera parfaitement l'affaire.

4.4 Tâches

Il est possible d'identifier plusieurs tâches pouvant être réalisées grâce à l'application. Cependant, que nous soyons un utilisateur occasionnel, régulier ou un expert, ces tâches s'effectueront toutes de la même manière et nous pouvons donc considérer que, dans ce qui va suivre, les différents types d'utilisateur seront regroupés.

Les tâches qu'il est possible de réaliser sont les suivantes :

- Création, modification et suppression d'un profil
- Consulter l'explorateur de fichiers
- Lancer la partie visualisation
- Configurer l'application
- Interagir avec une visualisation
- Afficher le statut de l'utilisateur
- Consulter l'aide

4.4.1 Création d'un profil

1. But de la tâche :

Le but est de se créer un profil contenant les caractéristiques relatives à l'utilisateur, qui seront nécessaires pour le bon fonctionnement de l'application.

2. Alternative :

S'il ne passe pas, à l'origine, par cette fonctionnalité, l'utilisateur ne sera pas capable d'utiliser l'application car il ne pourra pas lancer la partie visualisation si aucun profil n'a été créé.

3. Sortie :

La réalisation de cette tâche donnera un fichier XML (non connu de l'utilisateur) qui contiendra les informations introduites par l'utilisateur à savoir : son nom d'utilisateur,

l'information du stylo, la langue d'aide, la visualisation par défaut, la série temporelle, la date de création et de modification du profil.

4. Fréquence de la tâche :

Cette tâche doit être réalisée au minimum à chaque fois qu'un nouvel utilisateur veut se servir de l'application. Cependant, elle peut également être réalisée à chaque fois qu'un utilisateur désire se créer un nouveau profil pour posséder différentes caractéristiques.

5. Durée de la tâche :

L'introduction et la sélection des informations pour remplir le profil ne doivent pas dépasser la minute.

6. Flexibilité de la tâche :

Si l'utilisateur ne possède pas de profil et qu'il doit passer par cette tâche, il doit nécessairement la réaliser en premier sans quoi il ne pourra pas continuer et visualiser des séries temporelles.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire les options qui lui sont proposées et en sélectionner. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

A priori, puisque cette tâche est la première à être réalisée, elle n'est nullement dépendante d'une autre tâche. Toutefois, si l'utilisateur ne connaît pas les séries temporelles mises à sa disposition, il peut consulter l'explorateur de fichiers pendant la réalisation de son profil.

9. Tâches jointes :

Aucune tâche n'a besoin d'être réalisée pour que l'utilisateur puisse se créer un profil.

4.4.2 Modification d'un profil

1. But de la tâche :

Le but est de modifier un profil contenant les caractéristiques relatives à l'utilisateur qui seront nécessaires pour le bon fonctionnement de l'application.

2. Alternative :

S'il ne passe pas par cette fonctionnalité, cela n'empêchera pas l'utilisateur de manipuler l'application car il pourra toujours lancer la partie visualisation avec le profil existant créé préalablement.

3. Sortie :

La réalisation de cette tâche donnera un nouveau fichier XML (non connu de l'utilisateur) qui contiendra les informations introduites et modifiées par l'utilisateur telles : son nom d'utilisateur, l'information du stylo, la langue d'aide, la visualisation par défaut, la série temporelle, la date de création et de modification du profil.

4. Fréquence de la tâche :

Elle peut être réalisée à chaque fois qu'un utilisateur désire modifier un de ses profils pour posséder de nouvelles caractéristiques.

5. Durée de la tâche :

L'introduction et la sélection des informations pour modifier le profil ne doivent pas dépasser la minute.

6. Flexibilité de la tâche :

L'utilisateur n'est pas obligé de modifier son profil pour pouvoir utiliser l'application. Dès qu'il désire changer un (ou plusieurs) paramètre(s) de son profil, il devra le faire avant de lancer la partie visualisation car cela reviendrait à créer un nouveau profil.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire les options qui lui sont proposées et en sélectionner. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

A priori, cette tâche n'est nullement dépendante d'une autre, excepté le fait que l'utilisateur doit disposer d'un profil existant, c'est-à-dire qu'il doit déjà être passé par l'étape de création d'un profil à un moment antérieur. Toutefois, si l'utilisateur ne connaît pas les séries temporelles mises à sa disposition, il peut consulter l'explorateur de fichiers pendant la réalisation de son profil.

9. Tâches jointes :

Aucune tâche n'a besoin d'être réalisée pour que l'utilisateur puisse se créer un profil.

4.4.3 Suppression d'un profil

1. But de la tâche :

Le but est de supprimer un profil contenant les caractéristiques relatives à l'utilisateur qui seront nécessaires pour le bon fonctionnement de l'application. Cette tâche peut se réaliser lorsqu'un utilisateur estime qu'il n'a plus besoin de garder son profil ou qu'il souhaite le supprimer pour en créer un nouveau.

2. Alternative :

S'il ne passe pas par cette fonctionnalité, cela n'empêchera pas l'utilisateur de manipuler l'application car il pourra toujours lancer la partie visualisation avec le profil existant qu'il a créé préalablement.

3. Sortie :

La réalisation de cette tâche effacera le fichier XML (non connu de l'utilisateur) qui contient les informations introduites par l'utilisateur à savoir : son nom d'utilisateur, l'information du stylo, la langue d'aide, la visualisation par défaut, la série temporelle, la date de création et de modification du profil.

4. Fréquence de la tâche :

Elle peut être réalisée à chaque fois qu'un utilisateur a créé un profil. Cependant, dans ce cas pour continuer à utiliser l'application, il devra s'en créer un nouveau.

5. Durée de la tâche :

La suppression d'un profil ne doit pas dépasser la minute.

6. Flexibilité de la tâche :

À partir du moment où l'utilisateur désire supprimer son profil, il devra s'en recréer un nouveau avant de lancer la partie visualisation.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire les options qui lui sont proposées et en sélectionner. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

A priori, cette tâche n'est nullement dépendante d'une autre excepté le fait que l'utilisateur doit disposer d'un profil existant, c'est-à-dire qu'il doit déjà être passé par l'étape de création d'un profil à un moment antérieur.

9. Tâches jointes :

Aucune tâche n'a besoin d'être réalisée pour que l'utilisateur puisse supprimer un profil.

4.4.4 Exploration des fichiers

1. But de la tâche :

Cette tâche permet à l'utilisateur de consulter les différents fichiers qui se trouvent dans l'application. Cet explorateur va lui faire voir les fichiers relatifs aux profils des utilisateurs, les séries temporelles et les visualisations développées dans l'application. Il pourra ainsi consulter les descriptions que ces fichiers proposeront.

2. Alternative :

C'est la seule fonctionnalité qui permet de naviguer à travers les fichiers de l'application. Il n'existe pas d'autres moyens pour les consulter.

3. Sortie :

L'utilisateur pourra (comme sur un ordinateur) voir l'arborescence des fichiers qui se trouvent dans l'application.

4. Fréquence de la tâche :

Cette tâche peut être réalisée à chaque fois que l'utilisateur travaille dans l'application. Elle peut même être réalisée plusieurs fois pendant qu'il utilise l'outil.

5. Durée de la tâche :

Cette tâche peut varier selon le nombre de fichiers qui sont présents dans l'application et le nombre de fichiers que l'utilisateur veut explorer.

6. Flexibilité de la tâche :

À partir du moment où l'utilisateur désire supprimer son profil, il devra s'en recréer un nouveau avant de lancer la partie visualisation.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire une arborescence de fichiers afin de trouver les renseignements qu'il cherchait. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

A priori, cette tâche n'est nullement dépendante d'une autre s'il est considéré qu'il s'agit d'une tâche de consultation d'informations.

9. Tâches jointes :

Cette tâche peut s'inscrire dans la réalisation d'un profil où l'utilisateur aurait besoin de consulter certaines informations afin de sélectionner un paramètre dans la création (ou modification) de son profil.

4.4.5 Lancement de la partie visualisation (et réaliser une analyse)

1. But de la tâche :

Une fois que l'utilisateur s'est créé un profil, il peut lancer la partie visualisation de l'application pour afficher les séries temporelles qui ont été introduites et ainsi en tirer des analyses.

2. Alternative :

S'il veut pouvoir analyser les séries temporelles et créer des visualisations, il doit nécessairement passer par cette étape sans quoi, il ne pourra rien faire.

3. Sortie :

Lorsque cette tâche est réalisée, l'utilisateur arrive dans la fenêtre principale de la partie visualisation.

4. Fréquence de la tâche :

Tant que l'utilisateur ne doit pas consulter l'explorateur de fichiers, supprimer ou sélectionner un autre profil pour une nouvelle analyse, il peut se contenter de n'effectuer cette tâche qu'une seule fois lors de son utilisation de l'application.

5. Durée de la tâche :

La durée de cette tâche peut varier fortement. Selon que l'utilisateur effectue une analyse approfondie de la série temporelle, cela peut aller de quelques minutes à un laps de temps beaucoup plus important. Il est possible de passer un long moment à configurer l'application, créer des visualisations, les analyser et chercher les informations sur ces visualisations.

6. Flexibilité de la tâche :

Cette tâche doit être réalisée avant de vouloir commencer à configurer l'application et créer des visualisations.

7. Compétences et connaissances :

L'utilisateur doit savoir comment configurer l'application (ou du moins suivre les instructions pour le faire) et créer des visualisations. Il doit également être capable de tirer des conclusions suite secondaires aux analyses qu'il effectuera.

8. Dépendance de la tâche :

Cette tâche est dépendante d'autres telles que la configuration de l'application et la création de visualisations dans la mesure où elle en est composée.

9. Tâches jointes :

Les tâches qui sont jointes à celle-ci sont la configuration de l'application, la création de visualisations, l'interaction avec une visualisation, la consultation du statut de l'utilisateur et la consultation de l'aide.

4.4.6 Configuration de l'application

1. But de la tâche :

Les paramètres sélectionnés sont propres à la série temporelle choisie par l'utilisateur. La réalisation de cette tâche va permettre de définir les caractéristiques dont l'application devra tenir compte pour créer les visualisations.

2. Alternative :

Il n'existe pas d'autre fonctionnalité permettant de configurer l'application pour analyser des séries temporelles.

3. Sortie :

L'utilisateur reçoit un écran d'avancement et de confirmation de la sauvegarde de la configuration.

4. Fréquence de la tâche :

Cette tâche doit se faire de façon fréquente lorsqu'un utilisateur veut analyser une série temporelle. En effet, puisqu'il est impossible de réaliser toutes les visualisations à partir d'une seule configuration, l'utilisateur sera obligé de repasser par cette tâche s'il veut une analyse globale de sa série.

5. Durée de la tâche :

Cette tâche ne prend pas énormément de temps car il n'y a que quelques paramètres à sélectionner, elle ne doit donc pas durer plus de une ou deux minutes.

6. Flexibilité de la tâche :

Il est impératif de commencer par réaliser cette tâche une fois que la partie visualisation a été lancée sans quoi l'application ne pourra pas déterminer les caractéristiques nécessaires aux visualisations.

7. Compétences et connaissances :

L'utilisateur doit pouvoir savoir quelles séries temporelles et quels paramètres il souhaite sélectionner pour créer les visualisations.

8. Dépendance de la tâche :

Aucune dépendance n'est liée à cette tâche.

9. Tâches jointes :

Les tâches qui sont jointes à celle-ci sont la création de visualisations, l'interaction avec une visualisation, la consultation du statut de l'utilisateur et la consultation de l'aide.

4.4.7 Interaction avec une visualisation

1. But de la tâche :

Les paramètres sélectionnés sont propres à la série temporelle qui a été choisie par l'utilisateur. La réalisation de cette tâche va permettre de définir les caractéristiques dont l'application devra tenir compte pour créer les visualisations.

2. Alternative :

Il n'existe pas d'autre fonctionnalité permettant de manipuler les visualisations représentant les séries temporelles.

3. Sortie :

L'utilisateur voit sur l'écran d'avancement et de confirmation de la sauvegarde de la configuration.

4. Fréquence de la tâche :

Cette tâche se fera de façon assez fréquente pour analyser une série temporelle car il n'est pas possible de réaliser toutes les visualisations à partir d'une seule configuration.

5. Durée de la tâche :

Cette tâche ne prend pas énormément de temps car il n'y a que quelques paramètres à sélectionner, elle ne doit donc pas durer plus de une ou deux minutes.

6. Flexibilité de la tâche :

Il est impératif de commencer par réaliser cette tâche une fois que la partie visualisation a été lancée sans quoi l'application ne pourra pas déterminer les caractéristiques nécessaires aux visualisations.

7. Compétences et connaissances :

L'utilisateur doit pouvoir savoir quelles séries temporelles et quels paramètres il souhaite sélectionner pour créer les visualisations.

8. Dépendance de la tâche :

Aucune dépendance n'est liée à cette tâche.

9. Tâches jointes :

Les tâches qui sont jointes à celle-ci sont la création de visualisations, l'interaction avec une visualisation, la consultation du statut de l'utilisateur et la consultation de l'aide.

4.4.8 Affichage du statut de l'utilisateur

1. But de la tâche :

Cette tâche va permettre à l'utilisateur de connaître l'état de son statut pendant l'utilisation de l'application. Elle lui indiquera quels sont les paramètres sélectionnés parmi les suivants : son nom d'utilisateur, la visualisation par défaut, la série temporelle et les paramètres choisis.

2. Alternative :

S'il ne souhaite pas utiliser cette fonctionnalité, l'utilisateur peut toujours retourner dans la fenêtre de configuration pour directement voir ce qu'il a sélectionné dans cette fonctionnalité.

3. Sortie :

Un petit panneau contenant les diverses informations sera affiché à l'utilisateur en suraffichage des autres parties de l'écran.

4. Fréquence de la tâche :

L'utilisateur peut effectuer cette tâche autant de fois qu'il le désire.

5. Durée de la tâche :

La tâche est vraiment minimaliste et ne doit pas prendre longtemps, l'affichage est instantané et sa consultation ne doit prendre qu'une dizaine de secondes.

6. Flexibilité de la tâche :

La seule contrainte à cette tâche, c'est qu'il faut être passé par plusieurs étapes au préalable pour la réaliser dont la création d'un profil et la configuration de l'application.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire le panneau affiché afin de trouver les renseignements qu'il cherchait. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

Aucune dépendance n'est liée à cette tâche.

9. Tâches jointes :

Les tâches qui sont jointes à celle-ci sont la création de profil et la configuration de l'application.

4.4.9 Consultation de l'aide

1. But de la tâche :

Pour l'utilisateur, la réalisation de cette tâche va lui permettre de mieux connaître les différentes fonctionnalités pendant l'utilisation de l'application. Les textes d'aide (disponibles en français ou en anglais) sont destinés à expliquer de façon brève quelques points clés de l'application.

2. Alternative :

Elle ne sera pas utilisée par quelqu'un qui connaît l'application ou qui n'a pas besoin de l'aide. Cependant, il n'est possible, à aucun autre endroit de l'application, de consulter une aide.

3. Sortie :

Une fenêtre contenant les différents textes explicatifs sera affichée dans la langue que l'utilisateur a sélectionnée dans son profil. La liste des sujets possibles sera présentée sous la forme d'une liste au sein de laquelle l'utilisateur n'aura qu'à sélectionner un sujet.

4. Fréquence de la tâche :

L'utilisateur peut effectuer cette tâche autant de fois qu'il le désire.

5. Durée de la tâche :

La durée de cette tâche peut varier fortement. Selon que l'utilisateur effectue une recherche approfondie ou non, cela peut aller de quelques secondes à quelques minutes. Toutefois, le nombre de sujets explicatifs étant limité, la durée ne sera en moyenne pas très longue.

6. Flexibilité de la tâche :

Aucune contrainte n'est liée à cette tâche.

7. Compétences et connaissances :

L'utilisateur doit juste pouvoir lire le panneau affiché afin de trouver les renseignements qu'il cherchait. Il n'est pas nécessaire de posséder de connaissances particulières à ce niveau.

8. Dépendance de la tâche :

Aucune dépendance n'est liée à cette tâche.

9. Tâches jointes :

Aucune autre tâche n'est liée à celle-ci.

5 Applications sur PDA

5.1 Introduction

Les PDA (Personal Digital Assistant, littéralement assistants numériques personnels, aussi appelés organisateurs) sont des ordinateurs de poche composés d'un processeur, de mémoire vive, d'un écran tactile et de fonctionnalités réseaux dans un boîtier compact d'extrêmement petite taille [CCMc].

5.1.1 Utilité du PDA

Le PDA est un ordinateur de poche dont l'usage est prévu originalement dans un but d'organisation. Il fournit généralement les applications suivantes :

- Un agenda, pour l'organisation de l'emploi du temps, avec des mécanismes de rappel visuels ou auditifs.
- Un gestionnaire de tâches faisant office d'aide-mémoire pour les tâches à effectuer.
- Un carnet d'adresses (gestionnaires de contacts).
- Un logiciel de messagerie, rendant possible la consultation de ses mails ainsi que la rédaction de nouveaux messages.
- Des outils de synchronisation avec le PC (notamment les logiciels Microsoft, tel Outlook), simplement en les connectant sur un socle et en appuyant sur un bouton.

Les assistants personnels proposent des outils de bureautique allégés tels qu'un traitement de texte, un tableur, une calculatrice, des visualiseurs pour un grand nombre de formats de fichiers (fichiers PDF, images, etc.).

Pour un utilisateur classique, un PDA est à considérer comme un appareil de lecture. Il permet d'apporter quelques modifications (nom, numéro de téléphone ou rendez-vous). Il présente des fonctions intéressantes : gestion des comptes, utilisation d'une calculatrice (ou d'un convertisseur de monnaies, de mesures...).

Les PDA sont également de plus en plus utilisés pour des usages de géolocalisation, de cartographie et de navigation routière lorsqu'ils sont couplés à un dispositif de géolocalisation (GPS, Global Positioning System).

5.1.2 Système d'exploitation

Les PDA possèdent des systèmes d'exploitation dont la définition est adaptée à la résolution d'affichage de l'écran et dont les fonctionnalités correspondent aux caractéristiques de ce type d'appareil.

Il existe plusieurs systèmes d'exploitation pour PDA, correspondant la plupart du temps à des types de PDA différents et portés par des constructeurs différents, au même titre qu'il existe des ordinateurs Mac et PC.

Plus précisément, nous pouvons distinguer quatre familles d'assistants personnels numériques :

- Les appareils de type PalmOS (Palm Pilot ou compatibles) promu par la société Paml,
- Windows Mobile ou Pocket PC (anciennement Windows CE), promu par la société Microsoft.,
- Les Psion,
- Les autres.

Ces quatre familles possèdent à peu près les mêmes caractéristiques et les mêmes fonctionnalités avec une prise en main différente mais surtout des applications incompatibles entre elles. Il est à noter qu'il existe des systèmes d'exploitation Linux développés spécifiquement pour les deux premiers types de machines.

5.1.3 Caractéristiques techniques

Lorsqu'un utilisateur souhaite utiliser un PDA, il est important de veiller aux caractéristiques suivantes :

- **L'écran et le dispositif de saisie**

Les écrans doivent être du type sensible et ils s'utilisent avec un stylet. Dans la plupart des cas, ce stylet constitue le seul moyen d'effectuer la saisie de texte.

- **L'autonomie**

L'autonomie du PDA est fonction des caractéristiques de la batterie.

L'autonomie est l'un des points essentiels qui distingue les ordinateurs de poche les uns des autres. Selon les modèles, elle varie de 7 heures 30 à 50 heures. Ces écarts très importants s'expliquent principalement par la consommation d'énergie des différents types d'écrans (couleur ou monochrome, rétroéclairé ou non) et par la capacité des sources d'alimentation (batteries ion-lithium, NiMH, ou piles).

Les différents types de batterie dont sont pourvus les PDA :

- **Ni-Cad** (Nickel / Cadmium).
- **Ni-Mh** (Nickel / Métal Hybride) plus performant que les batteries Nickel-Cadmium.
- **Li-Ion** (Lithium / Ion).
- **Li-Polymer** (Lithium / Polymère) plus légères que les batteries Li-Ion.

- **L'encombrement et le poids**

Le PDA est destiné à être emporté partout et doit donc tenir dans la main ou la poche. Ses dimensions et son poids doivent donc être choisis de façon à être les plus petits possible, en gardant à l'esprit le besoin d'ergonomie et de surface d'affichage

5.2 Le PDA utilisé pour l'application

Pour réaliser ce travail, c'est un PDA de type Pocket PC avec les caractéristiques suivantes qui a été utilisé :

Système d'exploitation	Microsoft Pocket PC 2002
Processeur	400 MHz Intel PXA250 Application Processor
Mémoire	SDRAM : 64 MB Flash ROM : 32 MB
Ecran	Ecran couleur TFT translectif Couleur 65K (i.e. 65,536 de couleurs) Ecran tactile
Résolution	240 x 320 pixels Pas de .24mm
Méthode de capture	Reconnaissance écrite (écrite manuscrite) Soft de clavier Reconnaissance de caractères Enregistrement de voix
Audio	Haut-parleurInput
Connexion	Sans fil jusqu'à 115Kbps
Dimensions	13,5 x 8,5 x 1,5 cm
Batterie	1400 mAh Lithium Polymer
Poids	184 gr, batterie incluse

Tableau 5-1: Caractéristiques du PDA utilisé pour l'application.



Figure 5-1 : Pocket PC utilisé pour l'application.

5.3 Les limites rencontrées

5.3.1 Au niveau du processeur

Les processeurs des PDA (en l'occurrence du Pocket PC puisque c'est de celui-ci dont il a été question pour ce travail) n'ont pas la puissance des ordinateurs actuels et cela est tout à fait compréhensible. Il est difficile d'imaginer la réalisation d'applications qui fonctionneront aussi bien sur PDA que sur PC.

La puissance des processeurs de l'iPAQ Pocket PC Series va de 206 à 400 MHz. Cette puissance de 400 MHz est cependant satisfaisante pour les applications communes qui doivent fonctionner sur un PDA telles que les logiciels de bloc-notes, les gestionnaires d'agenda, les logiciels de synchronisation ou encore les logiciels de courrier électronique.

Cependant, certains programmes nécessitant une puissance plus accrue ne pourront fonctionner correctement sur un PDA. Il faut notamment penser aux programmes multimédias.

Nous veillerons à ce que l'application développée puisse fonctionner de façon correcte sur le PDA et reste dans les limites du processeur. Dès lors, un des enjeux du bon fonctionnement de l'application est le traitement initial des données.

5.3.2 Au niveau de l'écran

Comme il a été dit précédemment, la résolution de l'écran du Pocket PC est de 240 x 320 pixels (cf. Section 5.2). En regard à ce que peut offrir un écran d'ordinateur, c'est tout simplement minuscule et cela constitue très certainement le principal défaut des PDA. Cela représente surtout le premier frein au développement d'applications destinées à être utilisées sur ces appareils.

En dépit de la taille réduite de ces écrans (quelques centimètres de large sur quelques centimètres de haut seulement), un autre facteur négatif est l'impossibilité d'avoir plusieurs fenêtres ouvertes en même temps (les applications multi-fenêtrées sont donc à éliminer) ou applications actives au même moment.

De même, les menus déroulants trop longs sont à proscrire.

Cela dit, l'écran d'un PDA est un écran TFT. Cela aura pour conséquence de fournir à l'utilisateur qui le regarde un confort visuel appréciable. En effet, les écrans TFT peuvent être regardés sous différents angles sans pour autant que l'utilisateur ne perde en précision ou en lisibilité. Le problème de ces écrans TFT est que pour offrir ces services, ils consomment beaucoup plus de batteries qu'un écran classique.

Enfin, il est possible d'afficher 65,536 couleurs (cf. Section 5.2). Ces couleurs représentent une palette variée qui ne constitue nullement un frein au développement d'applications sur le PDA.

5.3.3 Au niveau de la mémoire

Avant tout, il est important de savoir que la mémoire d'un PDA ne fonctionne pas de la même façon que celle d'un ordinateur quant à la persistance des données. La mémoire présente dans le PDA est de la mémoire vive, cela signifie que si l'utilisateur laisse la batterie du PDA se décharger complètement, il peut perdre toutes les données qui s'y trouvent.

Tout comme la puissance du processeur, la mémoire d'un PDA n'est pas comparable avec celle d'un ordinateur de bureau mais la mémoire que possède l'appareil utilisé pour réaliser ce travail est suffisante pour toutes les applications de base qui ont été citées précédemment (cf. Section 5.1.1) et également pour certaines applications qu'un utilisateur pourrait développer lui-même. En effet, la SDRam, qui est présente, offre un bon rapport performance/prix.

Cependant, le problème rencontré (autre que la permanence des données) est le fait que l'application ne peut manipuler des fichiers trop volumineux à la fois. Il a donc fallu trouver

une solution pour manipuler la même quantité de données sans pour autant dépasser la capacité possible de la mémoire.

5.3.4 Au niveau de l'interaction

Une des caractéristiques principales des PDA est leur portabilité. Mais ce qui ne se remarque peut-être pas de prime abord, c'est que les modes d'interaction qu'il y a moyen d'avoir avec un PDA sont beaucoup moins développés que sur un ordinateur de bureau classique.

En effet, il n'est nullement question de souris, de clavier, etc. Tout cela est remplacé par quelques boutons de raccourcis se trouvant sur le PDA (des boutons donnant accès aux applications les plus couramment utilisées telles que les agendas, les blocs-notes) et un stylo.

Le stylo est l'outil le plus utilisé sur le PDA. Il a pour mission de faire oublier le clavier classique à l'utilisateur. Les PDA ont intégré un dispositif de reconnaissance manuscrite mais également de reconnaissance de caractères.

En ce qui concerne la reconnaissance de caractères, nous pouvons scinder cette technique en 2 modèles :

- Le modèle caractère par caractère. Un seul caractère est introduit à la fois, le caractère est reconnu et seulement ensuite, le caractère suivant peut alors être introduit.
- Le modèle à la volée. Une suite de caractères est introduite et dès que l'utilisateur a fini de les introduire, ces caractères sont reconnus. Une nouvelle suite de caractères peut alors être introduite.



Figure 5-2 : Clavier de Pocket PC.

La méthode à la volée permet de compenser la lenteur qui provient de l'utilisation du stylo en lieu et place du clavier. En effet, cela constitue un problème lors du développement de l'application : cela peut prendre énormément de temps pour l'utilisateur de configurer l'application au démarrage. Il faudra donc trouver un moyen pour ne pas perdre de temps uniquement dans l'insertion des données.

Une autre limite rencontrée au niveau de l'interaction est due à l'utilisation du stylo, non pour écrire ou introduire des données mais pour sélectionner des options. Comparé à un ordinateur classique avec sa souris, il est impossible sur un PDA de faire apparaître le « clic droit » et de ce fait accéder à des sous-menus contextuels selon l'application.

Cette restriction peut s'avérer importante, surtout lorsque des visualisations sont affichées, et que l'utilisateur désire sélectionner un point du graphique pour accéder à un sous-menu comme il le ferait en cliquant sur le bouton droit de sa souris.

5.4 Les solutions apportées

5.4.1 Au niveau de l'écran.

Pour pallier au problème de la taille réduite de l'écran, la solution utilisée est de pourvoir les graphiques d'une barre de défilement. Cette solution a eu pour avantage de ne pas modifier conceptuellement les visualisations dont la taille était supérieure à celle offerte par l'écran du PDA. En effet, cette augmentation de l'espace « disponible » permet de ne pas devoir compacter les visualisations et ainsi respecter la contrainte d'un seul écran pour chacune d'entre-elles.

Mieux, elle permet à l'utilisateur d'introduire des séries temporelles de tailles diverses sans que cela ne change la visibilité et la lisibilité des visualisations.

5.4.2 Au niveau de la mémoire.

Initialement, les données relatives aux séries temporelles sont stockées dans des fichiers XML. Cependant, les données sont stockées sous leur forme « brute », c'est-à-dire que toutes les données relatives à un jour sont rassemblées mais ne sont pas triées selon le paramètre qu'elles représentent.

Or, si chaque fois que l'utilisateur veut créer une visualisation, il faut que l'application commence à parcourir les fichiers, effectue un parsing et trie les données, cela prendrait trop de temps pour réaliser cette opération. Le tri et la répartition des données se font une seule fois lors de chaque configuration et sélection des paramètres à analyser. Chaque série est décomposée en un ensemble de différentes listes stockées en mémoire dans un souci de diminuer le temps nécessaire à la création des visualisations. Lors de cette répartition, des listes de minima et maxima sont également créées afin de ne pas devoir parcourir l'ensemble des listes créées précédemment.

5.4.3 Au niveau de l'interaction.

La solution choisie pour résoudre le(s) problème(s) d'interaction au niveau du stylo a été de développer dans l'application des menus contextuels. La seule interaction que le stylo pouvait avoir avec la visualisation sur l'écran du PDA était de déterminer une position, il a fallu trouver un autre mécanisme.

Selon les paramètres choisis par l'utilisateur et en fonction des besoins nécessaires, certains menus apparaissent comme accessibles et d'autres pas. La sélection de certaines propriétés (avant le contact entre l'écran et le stylo) permet à l'application de déterminer quelle fonctionnalité elle doit exécuter lorsque l'utilisateur lui désigne une position sur la visualisation.

5.5 Outils de développements sur PDA

Au moment de décider quel allait être l'outil de développement utilisé pour mener à bien la réalisation de cette application, deux choix se sont imposés de façon majoritaire : la plateforme « Microsoft .NET » et la plateforme J2ME. Ces 2 plateformes seront détaillées (architecture, composants, langages supportés, ...) afin de déterminer l'outil le plus approprié.

5.5.1 La plateforme de développement « Microsoft .NET »

5.5.1.1 Présentation

Dans la littérature actuelle, de plus en plus d'applications sont développées sous le label « .NET » ou « dotnet ». En réalité, il s'agit de la même chose : en anglais "point" se dit "dot". Dans ces pages nous utiliserons principalement le terme « .NET ».

« .NET » est un standard proposé par la société Microsoft, pour le développement d'applications d'entreprises multi-niveaux, basées sur divers composants. « Microsoft .NET » constitue la réponse de Microsoft à la plateforme J2EE de Sun. Le but principal de « .NET » est de permettre de développer simplement des applications Web inter opérables, reposant sur une architecture totalement nouvelle. Les aspects les plus intéressants de « .NET » se situent au niveau de la plateforme de développement, des langages et des protocoles qu'elle met en avant [THOMAS05].

La plateforme « .NET » a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plateformes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation.

L'environnement de développement unifié qui permet d'écrire aussi bien des programmes Windows (Winform) que des sites Web dynamiques (ASP.NET) s'appelle « Visual Studio .NET ». Désormais « .NET » propose un environnement 100% objet. Toutefois, cette nouvelle plateforme « .NET » représente une technologie entièrement nouvelle, qui apporte son lot de nouveautés au prix d'une compatibilité avec l'existant pas toujours garantie.

Avec « .NET », Microsoft propose le support de plus de 27 langages de programmation, partageant de façon cohérente une hiérarchie de classes fournissant des services de base. Les applications dans « .NET » ne s'exécutent plus en code machine natif : elles abandonnent le code Intel x86 au profit d'un langage intermédiaire, le MSIL, s'exécutant dans une sorte de machine virtuelle, la Common Language Runtime (CLR).

Les différents langages de programmation supportés par « .NET »				
Ada	APL	C# / C++	Visual Basic	Cobol
Eiffel	Fortran	Haskell	ML	J#
Jscript	Mercury	Oberon	Objective Caml	Oz
Pascal	Perl	Python	Scheme	SmallTalk

Tableau 5-2 : Langages de programmation supportés par ".NET".

Avec l'arrivée de « .NET », pratiquement toutes les composantes de l'architecture Microsoft DNA doivent faire face à des changements très profonds dont les principaux sont les suivants :

- le serveur Web IIS abandonne le modèle multi-threadé (performant mais fragile) au profit d'un modèle multi-processus.
- la technologie ASP, qui cède la place à ASP.NET (initialement baptisée ASP+), ou les scripts interprétés cèdent la place à des codes compilés à leur première invocation, à la façon des pages JSP
- COM+ 2.0, qui est un modèle de composants distribués totalement original ne conserve aucun des héritages de la lignée COM/DCOM/COM+. A ce titre, COM+ 2.0 ne fait plus appel à la Registry de Windows pour l'enregistrement des composants locaux ou distants.
- Un nouveau langage nommé C# (à prononcer « see sharp ») voit le jour : il s'agit d'un langage objet moderne, sorte de synthèse entre C++ et Java. Le créateur de C# est Anders Hejlsberg, qui fût l'architecte de plusieurs langages et outils chez Borland, dont le célèbre Delphi
- Le nouveau modèle de programmation, basé sur SOAP (Simple Object Access Protocol) et les Web Services change fondamentalement la façon de concevoir ses applications, et ouvre la voie vers un nouveau métier : la fourniture de services Web en ligne. Ces « Web Services » sont basés sur le protocole XML.

La plateforme « .NET » fera un usage massif de standards issus d'organismes indépendants tels que le W3C⁷, l'IETF⁸ ou l'ECMA⁹. Aujourd'hui, la vision proposée par « .NET » est celle d'un Internet constitué d'une infinité d'applications Web inter opérables formant un réseau planétaire d'échange de services.

5.5.2 L'architecture de la plateforme « Microsoft .NET »

L'architecture « .NET » peut être définie de la façon suivante [LAUER01] :

- Il s'agit d'un ensemble de Services communs, utilisables depuis plusieurs langages objet.s
- Ces Services s'exécutent sous la forme d'un code intermédiaire indépendant de l'architecture sous-jacente.
- Enfin, ces Services s'exécutent dans un runtime (CLR) assurant les fonctions de gestion des ressources et de surveillance du bon fonctionnement des applications.

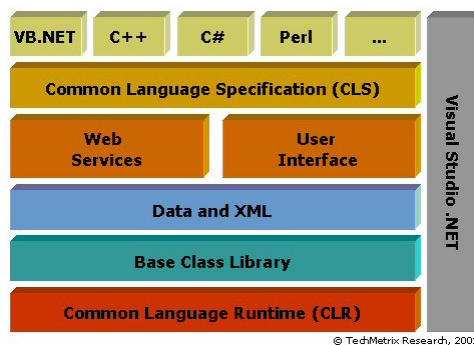


Figure 5-3 : Plateforme "Microsoft .NET".

⁷ World Wide Web. Consortium qui crée des standards pour le Web.

⁸ Internet Engineering Task Force. Groupe informel et auto-organisé dont les membres contribuent à l'ingénierie et à l'évolution des technologies de l'Internet.

⁹ European Computer Manufacturers Association. Association de constructeurs.

A la lecture de cette rapide description, nous ne pouvons nous empêcher de faire un parallèle avec Java. Le but premier de « .NET » est de fournir aux développeurs les moyens de créer des applications inter opérables depuis tout type de terminal : PC, assistant personnel, téléphone mobile.

5.5.2.1 Common Language Runtime (CLR)

La CLR permet de compiler le code source de l'application en un langage intermédiaire, baptisé MSIL (Microsoft Intermediate Language). Elle agit telle la machine virtuelle Java, en étant un environnement d'exécution prenant en charge des tâches de gestion des ressources (allocation mémoire et garbage collector). Elle fournit l'abstraction nécessaire entre l'application et le système d'exploitation sous-jacent. Lors de la première exécution de l'application, le code MSIL est à son tour compilé à la volée en code spécifique au système grâce à un compilateur JIT (Just In Time).

Dans le souci de fournir une plateforme stable afin d'atteindre le niveau de fiabilité exigé par les applications transactionnelles de e-Business, la CLR remplit également des tâches annexes telles que la surveillance de l'exécution des programmes. Dans la terminologie « .NET », il est question de « code managé » pour les programmes s'exécutant sous la surveillance de la CLR, et de code non managé (ou « unmanaged » en version originale) pour les applications ou composants s'exécutant en code machine natif, en dehors de la CLR.

La surveillance effectuée par la CLR concerne les erreurs de programmation classique. Celles-ci causent la majorité des défaillances logicielles depuis de longues années : accès à des éléments d'un tableau en dehors des limites, accès à des zones de mémoire non allouées, écrasement mémoire dus à des dépassements de tailles allouées.

5.5.2.2 « .NET » est multi-langages :

Avec la plateforme « .NET », Microsoft fournit plusieurs langages et les compilateurs associés, tels que C++, JScript, VB.NET (alias VB 7) et C#, un nouveau langage né avec « .NET ».

Des éditeurs tiers, partenaires de Microsoft, travaillent actuellement sur l'écriture des compilateurs pour d'autres langages, parmi lesquels nous trouvons des langages aussi divers et insolites que Cobol, Eiffel, CAML, Lisp, Python ou Smalltalk. Rational, l'éditeur du célèbre outil UML Rose, serait en train de finaliser un compilateur Java pour « .NET ».

5.5.2.3 Microsoft Intermediate Language

Tous ces langages sont compilés sous la forme d'un code binaire intermédiaire, indépendant du matériel et du système d'exploitation. Ce langage est MSIL : Microsoft Intermediate Language. Le MSIL est ensuite exécuté dans un runtime, la CLR (Common Language Runtime), remplissant sensiblement le même rôle que la JVM dans la plateforme Java. Le MSIL est alors traduit en code machine par un compilateur "Juste à temps" (JiT compiler).

5.5.2.4 Portabilité des applications

Les applications compilées sous la forme de code intermédiaire se présentent sous la forme de binaires exécutables portables (PE, Portable Executable).

Ainsi, Microsoft pourra proposer des implémentations complètes ou partielles de la plateforme « .NET » sur un vaste ensemble d'architectures matérielles et logicielles : PC Intel sous Windows 9x, Windows NT4, Windows 2000 ou des futures versions 64 bits de Windows,

Assistants personnels à base de microcontrôleurs sous PocketPC (ex Windows CE), et pourquoi pas d'autres systèmes d'exploitation ?

5.5.3 Le Compact Framework(1.1) de Microsoft

5.5.3.1 Qu'est-ce que le framework ?

Nous parlons généralement de «Framework» (traduisez «socle») pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

C'est un ensemble de classes qui facilitent la vie du développeur (liste chaînée, fonctions mathématiques,...) et qui permettent d'accéder au système (fichiers, fenêtres, graphiques, ...).

Le framework « .NET » comprend notamment :

- L'environnement d'exécution avec :
 - un moteur d'exécution, le CLR (Common Language Runtime).
 - un environnement d'exécution d'applications et de services web, appelé « ASP .NET » ;
 - un environnement d'exécution d'applications lourdes, appelé WinForms.
- Des services, sous forme d'un ensemble hiérarchisé de classes appelé Framework Class Library (FCL). La FCL est ainsi une librairie orientée objet, fournissant des fonctionnalités pour les principaux besoins actuels des développeurs. Le SDK (Software Development Kit) fournit une implémentation de ces classes.

Derrière les trois acronymes que sont CIL, CLI et CLR se cachent les trois couches de la programmation .Net : la Common Language Infrastructure, le Common Intermediate Language et le Common Library Runtime.

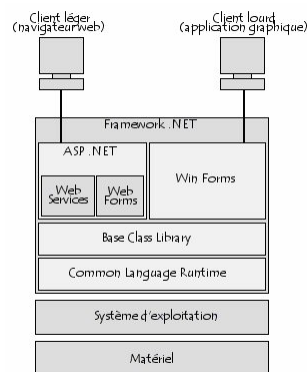


Figure 5-4 : Compact Framework (1.1).

Comme dit précédemment, un des grands avantages de « .Net » repose sur le nombre de langages de programmation que reconnaît la plateforme. Les plus connus sont ceux en provenance directe de chez Microsoft, comme C#, VB.NET, J# ou encore C++.NET.

Tout ceci ne se fait pas de manière automatique : pour programmer « .Net » avec ces langages, il faut utiliser des outils adaptés à « .Net », et non ceux utilisés normalement.

5.5.3.2 Gestion de la mémoire dans « .NET »?

La CLR possède un Garbage Collector (ou ramasse miettes) qui gère la mémoire pour les développeurs. Il détecte automatiquement qu'un objet n'est plus utilisé et libère la mémoire correspondante. Les programmes sont plus simples à écrire, les fuites mémoire par non libération n'existent plus.

5.5.4 Les classes de la librairie objet

La librairie objet du Framework Microsoft .NET (Framework Classes Library) est un ensemble de classes unifiées et hiérarchisées offrant un grand nombre de fonctionnalités standard. La FCL permet ainsi d'utiliser un grand nombre d'interfaces prédéfinies sans avoir à gérer de multiples librairies.

Les classes du framework .NET sont généralement présentées dans un tableau comme suit :

System.Web		System.Windows.Forms	
Services	UI	Form	Button
Caching	Security	MessageBox	ListControl
Configuration	SessionState	System.Drawing	
		Drawing2D	Printing
		Imaging	Text
System.Data		System.XML	
OLEDB	SQL	XLST	Serialization
Design	SQLTypes	XPath	
System			
Collections	IO	Security	Runtime
Configurations	Net	ServiceProcess	
Diagnostics	Reflection	Text	
Globalization	resources	Threading	

Tableau 5-3 : Classes du Framework ".NET".

5.5.5 La plateforme de développement J2ME

Sun propose une version allégée de J2SE (Java 2 Standard Edition), adaptée aux appareils de faible puissance appelée J2ME (Java 2 Micro Edition). Quant à J2ME, il cible les ordinateurs de poche et les machines "embarquées".

Chaque édition de cette plateforme fournit un environnement complet pour exécuter les applications basées sur Java, incluant une machine virtuelle Java (Java virtual machine = JVM) et les classes exécutables.

Les diverses spécifications qu'incluent J2ME sont toutes définies au travers du Java Community Process (JCP). À l'heure actuelle, il existe près de quarante requêtes distinctes de spécifications (Java Specification Requests = JSRs) concernant J2ME.

Dans J2ME, l'environnement d'exécution Java est adapté aux machines "à contraintes" - machines dont les capacités sont limitées en comparaison d'un ordinateur de bureau ou d'un serveur. Pour les plus basiques des machines, ces contraintes sont assez évidentes : mémoire extrêmement limitée, petite taille d'écran, méthodes d'entrées alternatives, et processeurs lents. Les machines les plus sophistiquées ne présentent que peu, voire aucune de ces contraintes, mais elles peuvent également tirer bénéfice d'un environnement optimisé et des nouvelles interfaces de programmation définies par J2ME.

Apprendre J2ME revient essentiellement à utiliser de nouvelles interfaces de programmation d'applications (APIs = application programming interfaces) et apprendre comment travailler en environnement restreint.

Il existe la possibilité d'utiliser les mêmes outils que ceux habituellement employés par les utilisateurs pour développer. En codant prudemment il est possible de développer des bibliothèques de classes qui sont portables sur n'importe quel appareil ou ordinateur possédant une machine virtuelle Java (JVM).

5.5.6 L'architecture de la plateforme J2ME

L'architecture de J2ME se découpe en plusieurs couches [SUN03] :

- Les profils : Ils permettent à une certaine catégorie de terminaux d'utiliser des caractéristiques communes telles que la gestion de l'affichage, des événements d'entrées/sorties (pointage, clavier, ...) ou des mécanismes de persistance (Base de données légère intégrée). Ces profils sont soumis à spécifications suivant le principe du JCP (Java Community Process)
- Les configurations : Elles définissent une plateforme minimale en terme de services concernant un ou plusieurs profils donnés.
- Les machines virtuelles : En fonction de la cible, la machine virtuelle pourra être allégée afin de consommer plus ou moins de ressources (KVM, CVM,...)
- Le système d'exploitation : L'environnement doit s'adapter au système d'exploitation existant (Windows CE, Palm Os, SavaJe, ...)

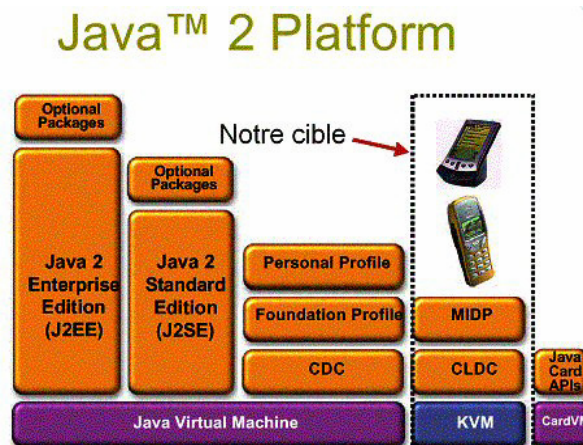


Figure 5-5 : Plateforme J2ME.

Cette architecture en couche a pour but de factoriser pour des familles de produits données un ensemble d'API permettant à une application de s'exécuter sur plusieurs terminaux sans modification de code.

Dans cette optique, la plateforme propose deux configurations :

- **CDC (Connected Device Configuration)** : Spécifie un environnement pour des terminaux connectés de forte capacité tels que les « set top boxes », les téléphones à écran, la télévision numérique, ... Les caractéristiques de l'environnement matériel proposé par la configuration CDC sont
 - Minimum de 512Ko de ROM et 256Ko de RAM, processeur 32 bits
 - Une connexion réseau obligatoire (sans fil ou pas)
 - Support des spécifications complètes de la machine virtuelle Java (CVM)
 - Cette configuration s'inscrit donc dans le cadre d'une architecture Java presque complète.
- **CLDC (Connected Limited Device Configuration)** : Cible les périphériques à ressources limitées ou faibles tels que les téléphones mobiles, les assistants personnels, ou les périphériques légers sans fil (wireless). Ces périphériques étant limités en terme de ressources, l'environnement classique ne permet pas de respecter les contraintes d'occupation mémoire liées à ces appareils. J2ME définit donc un ensemble d'API spécifique à CLDC et est destiné à utiliser les particularités de chaque terminal d'une même famille (profile). La liste suivante résume l'ensemble de ces caractéristiques :
 - Minimum de 160Ko à 512Ko de RAM, processeur 16 ou 32 bits, vitesse 16Mhz ou plus
 - Alimentation limitée, prise en charge d'une batterie
 - Connexion réseau non permanente, sans fil.
 - Interface graphique limitée ou inexistante

Définie par un sous-ensemble de classes Java s'exécutant dans la KVM (KiloByte Virtual Machine), cette configuration s'inscrit donc dans une logique d'économie de ressources avec une KVM de 40 à 80 Ko s'exécutant 30 à 80% moins rapide qu'une JVM normale. Il n'y a aucun compilateur Just-In-Time ni même de prise en charge des nombres flottants. Quant au Multi-threading et au Garbage Collector, ils demeurent supportés.

Toutefois, CLDC n'intègre pas la gestion des interfaces graphiques, la persistance ou les particularités de chaque terminal. Ces aspects ne sont pas de sa responsabilité.

MIDP (Mobile Information Device Profile)

MIDP est à la base de l'implémentation des classes liées à un profil donné. C'est dans ce profil qu'il est possible de trouver les méthodes permettant de gérer l'affichage, la saisie utilisateur et la gestion de la persistance (base de données).

Le package `javax.microedition.io` fait partie des briques non incluses dans J2SE tel qu'illustré dans le schéma suivant.

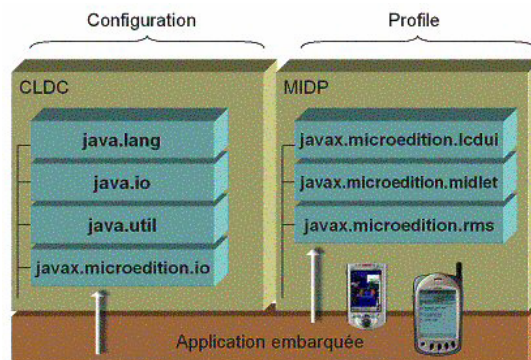


Figure 5-6 : Package `javax.microedition.io`.

6 Outil générique de visualisation sur PDA

6.1 Visualisations principales

Les visualisations principales développées lors de cette application sont au nombre de trois. Dans ce qui suit, nous détaillerons les caractéristiques, avantages et inconvénients de chacune de ces visualisations.

La caractéristique commune des visualisations principales est constituée par le fait d'être basée sur la couleur (contrairement aux visualisations secondaires qui sont plutôt basées sur la forme).

Ces trois visualisations principales (Brick Wall Chart, Stacked Bar Chart et Bar Chart) ont été extraites du travail de Frédéric Randolet qui sert de base à ce mémoire. Il est toutefois important de signaler que si l'aspect de ces visualisations a été conservé, leur conception a été revue afin de les rendre génériques.

6.1.1 Brick Wall Chart

La visualisation « Brick Wall Chart » est la première des visualisations principales. Elle est fortement basée sur la couleur et plus précisément sur une échelle de couleurs pour la représentation des informations.

Cette visualisation permet de représenter un nombre important de données et cela, simultanément sur un seul écran tout en restant relativement lisible pour l'utilisateur.

Lors de l'affichage de la visualisation « Brick Wall Chart », l'écran est organisé de la façon suivante : la fenêtre est scindée en 2 parties.

- La 1e partie (i.e. celle du haut) correspond à l'affichage proprement dit de la variation de la série temporelle.
- La 2eme partie (i.e. celle du bas) correspond à l'échelle de couleur utilisée pour représenter les variations de valeurs.

Cette visualisation est capable de représenter un, deux ou quatre paramètres de la série temporelle sans en changer le concept.

Comme il est possible de le constater sur les images, l'écran est divisé en 12 colonnes qui correspondent chacune à un mois de l'année :

- Chaque colonne est ensuite subdivisée (en hauteur) en rectangle de façon à représenter les jours de chaque mois.
- Chaque jour peut être subdivisé (en largeur) en deux ou quatre rectangles plus petits de façon à représenter un, deux ou quatre paramètres.

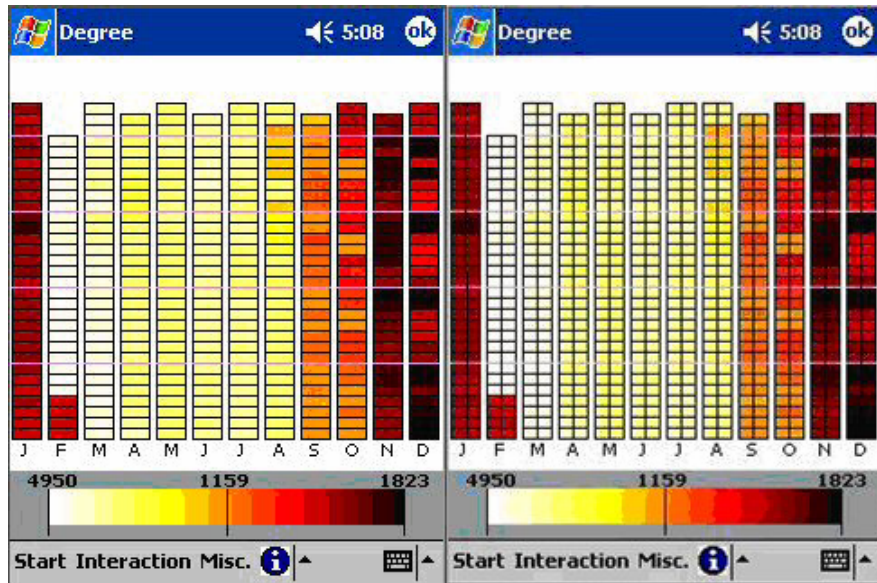


Figure 6-1 : Brick Wall Chart¹⁰ à 1 et 2 paramètre(s).

Chronologiquement, les mois sont ordonnés de la gauche vers la droite tandis que les jours sont ordonnés du bas vers le haut de l'écran.

Une fonction de « focus » est disponible pour cette visualisation. Il est possible d'obtenir une vision d'un mois en particulier et nous obtenons ainsi le Bar Chart. Cette vision détaillée s'obtient simplement en appuyant avec le stylet sur un des mois proposés à l'écran (i.e. une des colonnes).

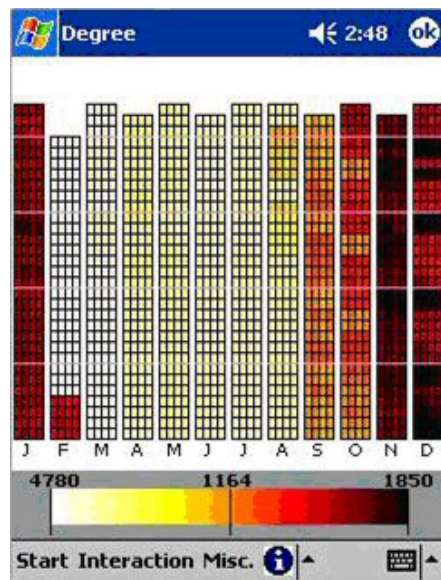


Figure 6-2 : Brick Wall Chart¹¹ à 4 paramètres.

¹⁰ Basée sur le travail de Frédéric Randolet (Mémoire de fin d'études, 2004).

¹¹ Basée sur le travail de Frédéric Randolet (Mémoire de fin d'études, 2004).

6.1.2 Stacked Bar Chart

La seconde visualisation principale est la « Stacked Bar Chart ». Cette représentation est basée à la fois sur la couleur et sur une analogie au calendrier. Cependant, tout en étant basée sur la couleur, cette visualisation n'utilise pas d'échelle de couleurs pour représenter les informations, contrairement à la précédente.

Cette visualisation s'affiche sur deux écrans (elle a donc été pourvue d'une barre de défilement verticale). En effet, le 1er écran constitue les informations relatives aux mois de l'année allant de Janvier à Juin tandis que le second écran renseigne l'utilisateur sur les mois de l'année allant de Juillet à Décembre.

L'idée de « Stacked Bar Chart » était bien de représenter chaque mois comme si l'utilisateur consultait un calendrier.

Cette visualisation n'est conçue que pour permettre à l'utilisateur de représenter deux paramètres de la série temporelle car elle a été partiellement imaginée dans un but de comparaison de deux paramètres l'un par rapport à l'autre.

Lors de l'affichage de la visualisation « Stacked Bar Chart », la fenêtre n'est pas scindée car il n'y a pas d'échelle des couleurs (comme précédemment expliqué). Deux couleurs seulement sont utilisées dans ce cas : la couleur rouge pour le premier paramètre et la couleur bleue pour le second paramètre.

Chaque barre verticale représente un jour de la semaine et les semaines sont séparées par une petite barre sur l'axe des abscisses.

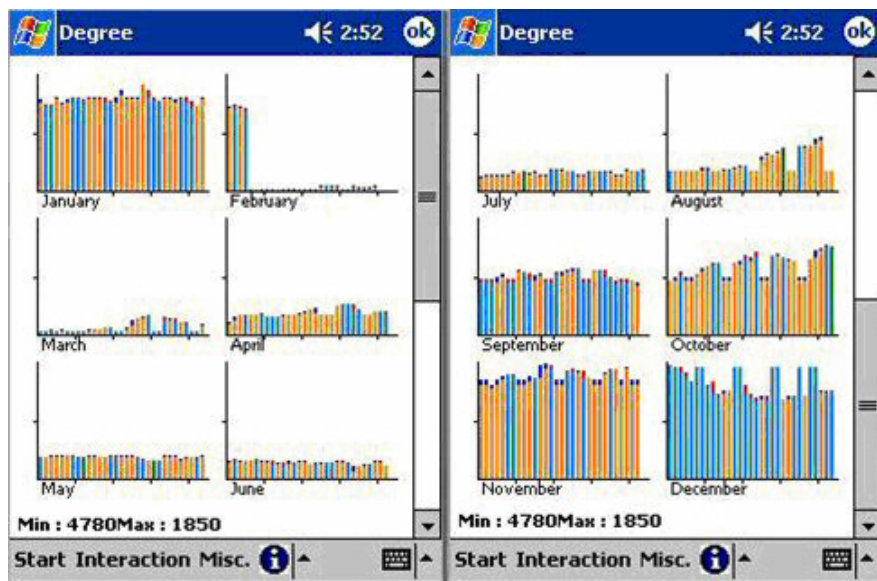


Figure 6-3 : Stacked Bar Chart¹² (Janvier – Juin / Juillet – Décembre).

La fonction de « Focus » permettant d'obtenir une vision d'un mois en particulier est également disponible pour cette visualisation. Ainsi nous obtenons le Bar Chart. Cette vision détaillée s'obtient simplement en appuyant avec le stylet sur un des mois proposés à l'écran (i.e. un des « calendriers »).

¹² Basée sur le travail de Frédéric Randolet (Mémoire de fin d'études, 2004).

6.1.3 Bar Chart

La dernière visualisation principale est appelée « Bar Chart ». Elle a deux fonctions au sein de l'application :

- Elle sert de visualisation à part entière (c'est-à-dire que l'utilisateur peut tout à fait décider de l'utiliser au départ des données de base de la série temporelle).
- Ou bien elle peut servir de résultat à un « focus » exécuté à partir des deux visualisations précédentes. Au préalable, l'utilisateur avait choisi de visionner « Brick Wall Chart » ou « Stacked Bar chart » et puis opté pour une analyse plus détaillée d'un mois en particulier.

La visualisation en « Bar Chart » est constituée de petits rectangles verticaux qui représentent chacun un jour du mois et peuvent être de deux couleurs (vert ou rouge). La couleur d'un rectangle est déterminée selon le rapport entre deux paramètres de la série temporelle.

En effet, cette visualisation effectue une comparaison entre ces deux paramètres pour permettre à l'utilisateur de voir lequel des deux est (ou a été) supérieur à l'autre et cela, jour après jour.

- Si le 1er paramètre est supérieur au 2e paramètre : le rectangle sera de couleur verte.
- Si le 1er paramètre est inférieur au 2e paramètre : le rectangle sera de couleur rouge.

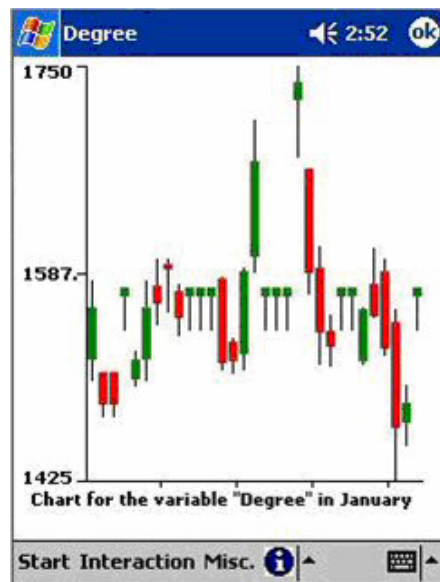


Figure 6-4 : Bar Chart¹³.

Cependant, du fait de ses caractéristiques et de son objectif de comparaison, pour pouvoir afficher cette visualisation, la condition minimale est que la série temporelle contienne au moins deux paramètres et qu'ils aient été sélectionnés par l'utilisateur.

6.2 Visualisations secondaires

Les visualisations secondaires développées lors de cette application sont également au nombre de trois. Dans ce qui suit, nous détaillerons les caractéristiques, avantages et inconvénients de chacune de ces visualisations.

¹³ Basée sur le travail de Frédéric Randolet (Mémoire de fin d'études, 2004).

La caractéristique commune des visualisations secondaires est d'être basée sur la forme, contrairement aux visualisations principales qui sont basées sur la couleur et les analogies qu'elles peuvent susciter auprès des utilisateurs.

6.2.1 Mountain

La première visualisation secondaire est la « Mountain ». Elle fait référence à une montagne renversée (la rotation de 90° est due à l'écran du PDA dont la hauteur est plus importante que la largeur). Elle représente l'évolution d'un paramètre pendant une année.

La taille (dans ce cas, il s'agit de la hauteur) nécessaire pour afficher tous les jours de l'année est supérieure à celle offerte par l'écran du PDA, c'est pourquoi une barre de défilement verticale a été ajoutée à la visualisation ainsi que le montrent les images suivantes.

Le principe de représentation de cette visualisation est le suivant : l'utilisateur peut aisément distinguer les points qui reflètent les valeurs maximales prises par la série temporelle en repérant les points situés à l'extrémité droite de l'image. En effet, de la même façon que nous observerions le point culminant d'une montagne en cherchant le point le plus élevé, ici, il s'agit de faire la même opération moyennant une rotation de 90°.

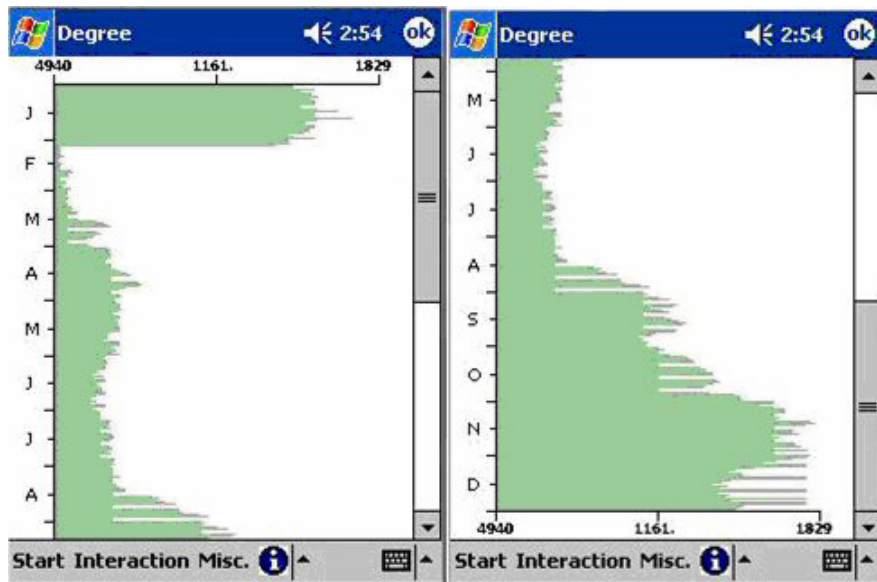


Figure 6-5 : Mountain.

Chronologiquement, les mois (et par conséquent les jours) sont ordonnés du haut vers le bas de l'écran.

Pour cette visualisation, il est possible d'obtenir un véritable « zoom » d'un mois en particulier. L'opération de « zoom » consiste en un agrandissement de la zone sélectionnée par l'utilisateur. Cette vision agrandie s'obtient simplement en appuyant avec le stylet sur la zone rectangulaire représentant un des mois proposés à l'écran (il suffit de se positionner entre les traits délimitant chaque mois sur l'axe).

Cela signifie que le même modèle de représentation sera utilisé pour la portion agrandie : il faudra toujours chercher le point se situant à l'extrême droite pour déterminer la valeur maximale.

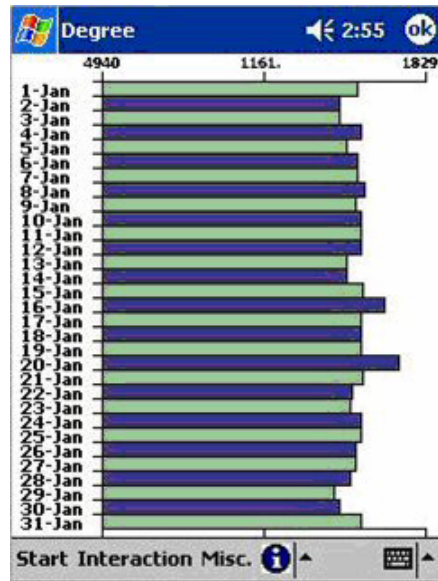


Figure 6-6 : Mountain (avec zoom sur un mois).

Pour un souci de lisibilité, la couleur des jours est en alternance verte et bleue mais ceci n'a aucune influence sur l'information représentée.

6.2.2 Multi-Lines

« Multi-Lines » constitue la deuxième visualisation secondaire. C'est une représentation dérivée de « Mountain » car elle est basée sur la même analogie que la précédente visualisation mais la forme pleine a été remplacée par une courbe. La courbe représente l'évolution d'un paramètre pendant une année.

L'avantage principal de cette visualisation par rapport à la précédente est qu'elle autorise l'utilisateur à visionner simultanément autant de paramètres qu'il le désire. Toutefois, pour un souci de lisibilité, il est conseillé de ne pas afficher trop de paramètres sur la même image (au-delà de huit paramètres, la visualisation perd de sa clarté de façon importante).

Le principe de représentation de cette visualisation est le suivant : l'utilisateur peut aisément distinguer les points qui reflètent les valeurs maximales prises par la série temporelle en repérant les points situés à l'extrémité droite de l'image. Chronologiquement, comme pour la visualisation précédente, les mois (et par conséquent les jours) sont ordonnés du haut vers le bas de l'écran.

Pour les mêmes raisons que précédemment, la taille (dans ce cas, il s'agit de la hauteur) nécessaire pour afficher tous les jours de l'année étant supérieure à celle offerte par l'écran du PDA, une barre de défilement verticale a été ajoutée à la visualisation comme le montrent les images suivantes.

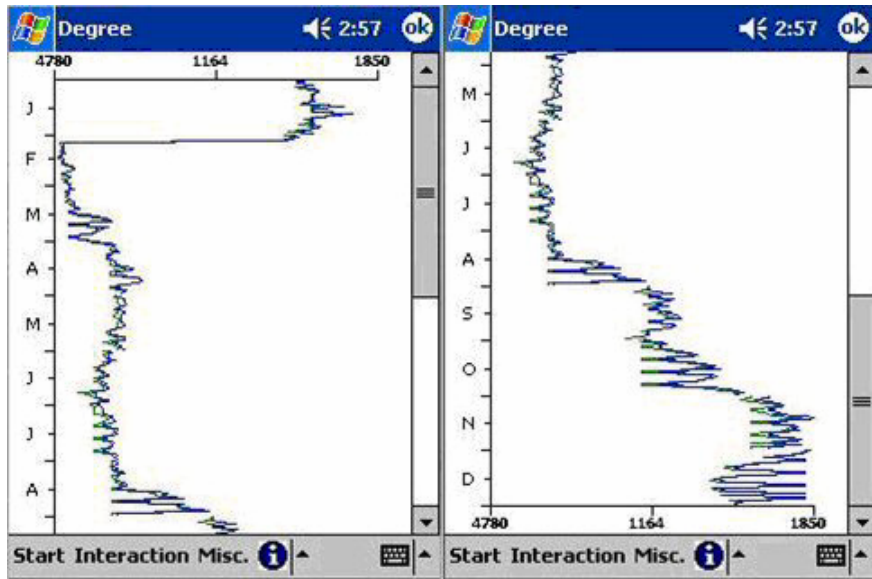


Figure 6-7 : Multi-Lines.

Pour cette visualisation, il est possible d'obtenir un véritable « zoom » d'un mois en particulier. L'opération de « zoom » consiste en un agrandissement de la zone sélectionnée par l'utilisateur. Cette vision agrandie s'obtient de la même façon que pour la visualisation précédente (i.e. « Mountain »), simplement en appuyant avec le stylet sur la zone rectangulaire représentant un des mois proposés à l'écran (il suffit encore de se positionner entre les traits délimitant chaque mois sur l'axe).

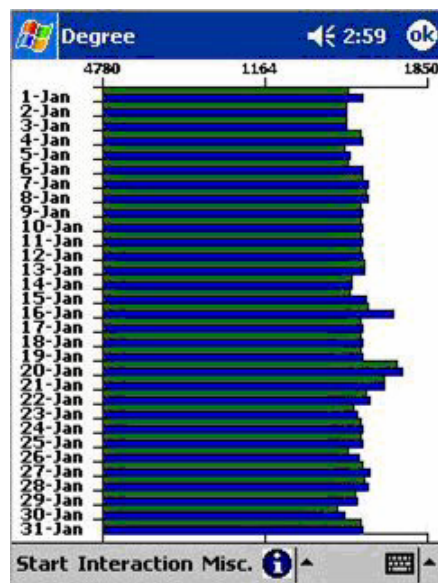


Figure 6-8 : Multi-Lines (avec zoom sur un mois).

Cela signifie que le même modèle de représentation sera utilisé pour la portion agrandie : il faudra toujours chercher le point se situant à l'extrême droite pour déterminer la valeur maximale.

Dans ce cas ci, l'alternance des couleurs ne sert plus à indiquer les jours mais à permettre à l'utilisateur de distinguer les différents paramètres sélectionnés.

Plus il aura sélectionné de paramètres, plus le nombre de couleurs pour chaque jour sera élevé.

6.2.3 Dashboard

« Dashboard » est la dernière visualisation créée. Cette visualisation n'en est pas une au sens premier du terme dans le sens où elle ne met en jeu aucune notion de couleur, de forme ou de représentation analogique. Cette représentation a été faite pour les personnes ayant une préférence pour effectuer des analyses ou des comparaisons de valeurs se trouvant au sein de tableaux.

Il s'agit en quelque sorte d'un tableau récapitulatif dans lequel se trouvent toutes les valeurs des paramètres analysés. Avec cette visualisation, autant de paramètres que possible peuvent être affichés par l'utilisateur pour une analyse comparative.

Le tableau est disponible sous deux formes :

- Le tableau des valeurs brutes (par mois), il suffit de changer le mois affiché pour obtenir ses informations.
- Le tableau des extrema (globaux, par mois et par paramètre).

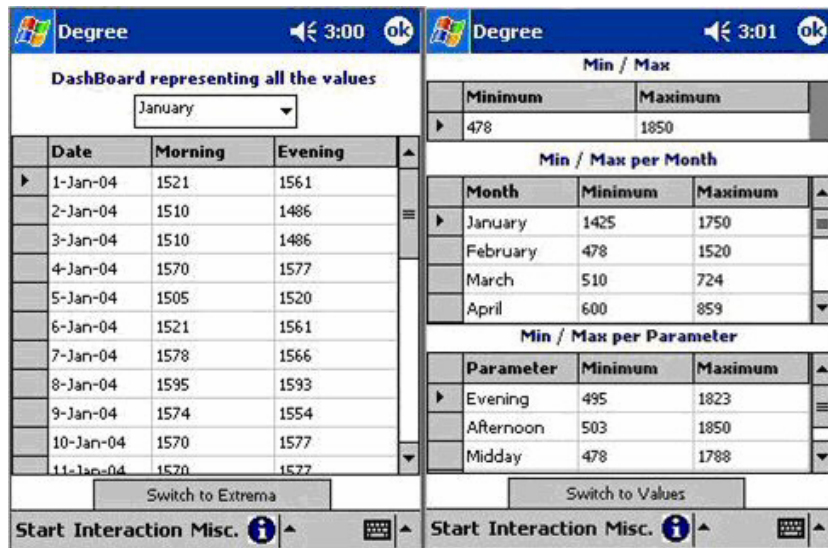


Figure 6-9 : Dashboard (Valeurs générales et extrêmes).

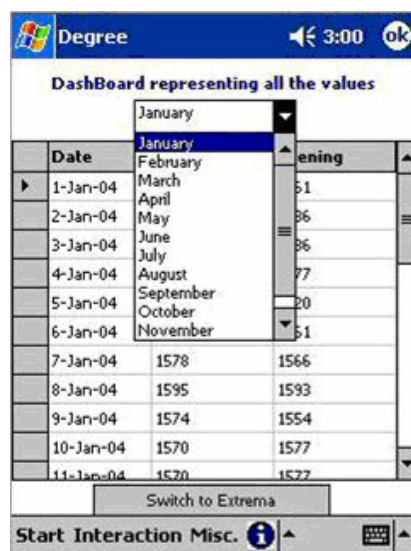


Figure 6-10 : Dashboard (Sélection du mois à visualiser).

Contrairement aux autres visualisations (qu'elles soient principales ou secondaires), « DashBoard » ne possède ni fonctionnalité « focus » ni fonctionnalité « zoom » que l'utilisateur pourrait utiliser pour une analyse plus détaillée. Pour changer de mois, il suffit de sélectionner un autre mois dans la liste déroulante qui est proposée. Les colonnes s'adaptent aux nombres de paramètres sélectionnés par l'utilisateur.

Bien entendu, aucune fonction de « Focus » ou de « Zoom » n'est disponible pour cette visualisation.

6.3 Interactions avec les utilisateurs

Lorsqu'une application est développée, nous devons tenir compte des interactions que les utilisateurs peuvent avoir avec celle-ci. Ces interactions peuvent répondre à différents principes ou concepts ergonomiques cités plus haut, notamment ceux qui préconisent de mettre en place des dispositifs afin d'aider l'utilisateur. Ils l'informent, le rassurent, le font patienter lorsqu'un traitement demande un certain laps de temps. Ils l'avertissent que l'application n'a pas subi de crash pendant ce laps de temps.

En ce qui nous concerne, ces différents dispositifs ont été mis en place à divers moments de l'application pour remplir ces fonctions.

De plus, l'application est développée sur PDA. Cela signifie que la perception de l'utilisateur lors de la manipulation d'un PDA (et plus précisément la combinaison stylo – écran tactile) sera tout à fait différente de celle qu'il aura en manipulant un ordinateur. En effet, l'écran du PDA est un écran tactile, l'utilisateur ne peut jamais être certain que la pression exercée sur cet écran a bien été détectée et a mené à une action.

Les dispositifs qui ont été mis en place sont les suivants :

- Boîte de dialogues

Les boîtes de dialogues ont été utilisées dans un but de confirmation de la part d'un utilisateur lorsqu'il s'apprêtait à effectuer une action de sélection, de suppression ou toute autre action qui pourrait se révéler irréversible.

Exemple : lorsque l'utilisateur souhaite supprimer un paramètre (ou un profil) de l'application ou lorsqu'il souhaite quitter l'application elle-même.

- Infos-bulles d'information / d'aide

La meilleure gestion que nous pourrions avoir au niveau des erreurs est de les empêcher. Pour cela, il faut prévenir et guider l'utilisateur en cas de doute ou de manque de clarté des options affichées à l'écran.

Exemple : Indiquer par une info-bulle quelle est la marche à suivre dans le cadre de la sélection de paramètre (que ce soit le nombre d'opérations, la séquence d'opérations à effectuer ou l'endroit où ces opérations se réalisent)

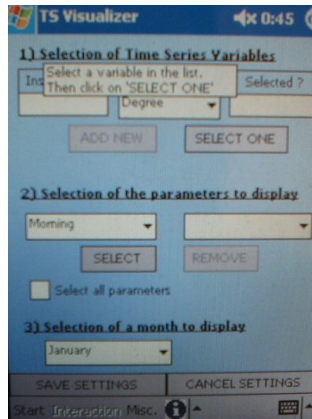


Figure 6-11 : Informations supplémentaires.

- Fenêtres d'état d'avancement des traitements

Lorsqu'un traitement a lieu, il est important d'indiquer à l'utilisateur que les opérations demandées sont en cours d'exécution et qu'elles n'ont pas été la cause d'un crash de l'application. Qui plus est, lorsque le temps de traitement de ces opérations devient long, l'utilisateur peut s'impatienter et effectuer d'autres opérations susceptibles de créer des problèmes.

Exemples : Une fenêtre avec une barre de défilement semble donc être une solution appropriée pour remplir les objectifs d'informations de l'utilisateur. Notamment dans les cas de sauvegarde, de calculs de données et de chargement d'une visualisation.

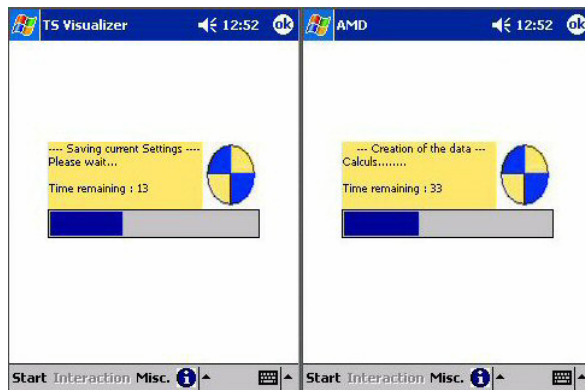


Figure 6-12 : Sauvegarde de la configuration.

- Sons

Comme il a été dit plus haut, l'utilisateur n'est jamais sûr que la pression exercée sur l'écran tactile a bien engendré une action au niveau de l'application. Pour pallier à ce problème, le son pourrait être une solution imaginable.

En effet, les sons peuvent tout à fait être utilisés dans le cadre d'information, de confirmation ou de prévention. Ils peuvent même indiquer à l'utilisateur qu'il a commis une erreur.

Exemple : dans l'application qui a été développée, divers sons sont utilisés tels que

- *Un son pour avertir l'utilisateur qu'il a commis une erreur dans la sélection des paramètres.*
- *Un son pour l'avertir qu'il vient de trouver la valeur minimale (maximale) d'une visualisation.*
- *Des sons pour l'avertir qu'il s'apprête à quitter une partie de l'application ou l'application elle-même.*

6.4 Relations entre les visualisations

Lorsque l'utilisateur veut passer d'une visualisation à l'autre dans le cadre de son analyse d'une série temporelle, il faut déterminer dans quelle mesure il sera possible de réaliser cette opération. Il faut déterminer, à partir d'une visualisation donnée, vers quelle autre visualisation il est autorisé à « switcher ».

Différents cas de figure sont envisageables en fonction des caractéristiques propres à chaque visualisation. En effet, certaines visualisations n'ont aucune restriction quant aux nombres de paramètres qu'elles peuvent traiter tandis que d'autres ne peuvent en prendre qu'un, deux ou quatre.

Les différents cas de figure possibles sont ceux où :

- L'utilisateur peut passer directement d'une visualisation à l'autre sans aucune modification de paramètres.
- L'utilisateur peut passer partiellement d'une visualisation à l'autre sans aucune modification de paramètres.
- L'utilisateur peut passer d'une visualisation à l'autre moyennant une modification de paramètres.
 - Dans ce cas, il s'agit de l'ajout d'un ou de plusieurs paramètres.
- L'utilisateur peut passer d'une visualisation à l'autre moyennant une modification de paramètres.
 - Dans ce cas, il s'agit du retrait d'un ou de plusieurs paramètres.

Cette notion de « switch » d'une visualisation à l'autre est importante pour l'utilisateur car il ne lui est pas possible, au départ des mêmes données de base sélectionnées, d'afficher et d'analyser toutes les visualisations.

De ce fait, il devra passer par cette fonctionnalité s'il désire traiter selon différentes approches, les informations que lui offre la série temporelle.

Rappel des différentes caractéristiques des visualisations au niveau du nombre de paramètres qu'elles peuvent traiter :

VISUALISATION	NOMBRE DE PARAMETRES	FOCUS	ZOOM
BRICK WALL CHART	1, 2 ou 4 paramètres	Oui	Non
STACKED BAR CHART	2 paramètres	Oui	Non
BAR CHART	2 paramètres	Non	Non
MOUNTAIN	1 paramètre	Non	Oui
MULTI-LINES	Nombre indéfini	Non	Oui
DASHBOARD	Nombre indéfini	Non	Non

Tableau 6-1 : Caractéristiques des visualisations.

6.4.1 Cas 1 : Passage complet

L'utilisateur peut passer directement d'une visualisation à l'autre sans aucune modification de paramètres. Ceci correspond au cas où un certain nombre de paramètres sont sélectionnés par l'utilisateur. En utilisant la fonctionnalité « switch », il pourrait afficher toutes les visualisations les unes après les autres. Il formerait ainsi une boucle lui permettant de revenir à la visualisation initiale.

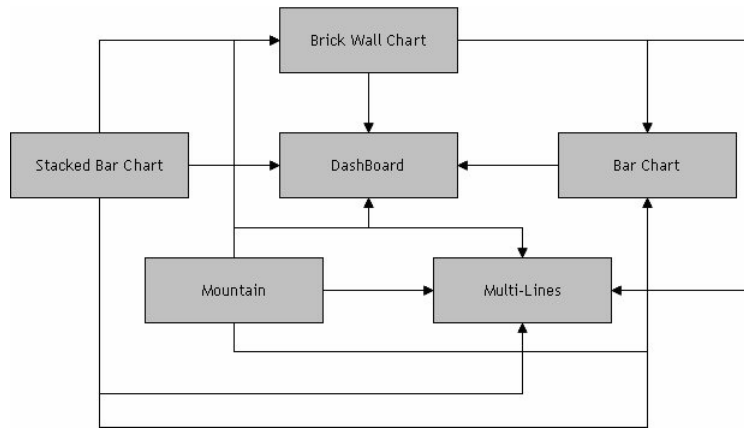


Figure 6-13 : Passage sans aucune modification (dans tous les cas).

Exemple : supposons que nous nous trouvions en « Brick Wall Chart » avec 1, 2 ou 4 paramètre(s) sélectionné(s), il serait possible (sans rien changer) de passer à « Dashboard », « Bar Chart » et « Multi-Lines » car ces visualisations sont réalisables avec ce nombre de paramètres sélectionnés.

6.4.2 Cas 2 : Passage partiel

L'utilisateur peut passer partiellement d'une visualisation à l'autre sans aucune modification de paramètres. En fonction du nombre de paramètres affichés (i.e. sélectionnés), l'utilisateur ne peut pas passer à toutes les visualisations.

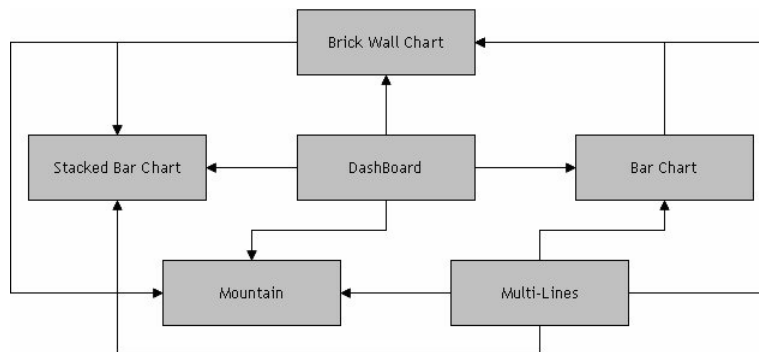


Figure 6-14 : Passage sans modification de paramètres (partiel).

Exemple : supposons que nous nous trouvions en « Dashboard » avec 1 ou 2 paramètre(s) sélectionné(s), il serait possible (sans rien changer) de passer à « Brick Wall Chart », « Stacked Bar Chart », « Bar Chart » et « Mountain » car ces visualisations sont réalisables avec ce nombre de paramètres sélectionnés. Cependant, si initialement nous nous trouvons avec la même visualisation mais en ayant sélectionné 4 paramètres, « Stacked Bar Chart » ne serait plus accessible.

6.4.3 Cas 3 : Passage avec ajout de paramètres

L'utilisateur peut passer d'une visualisation à l'autre moyennant une modification de paramètres. Dans ce cas, il s'agit de l'ajout d'un ou de plusieurs paramètres. Il faut passer à une sélection plus élargie des paramètres relatifs à la série temporelle.

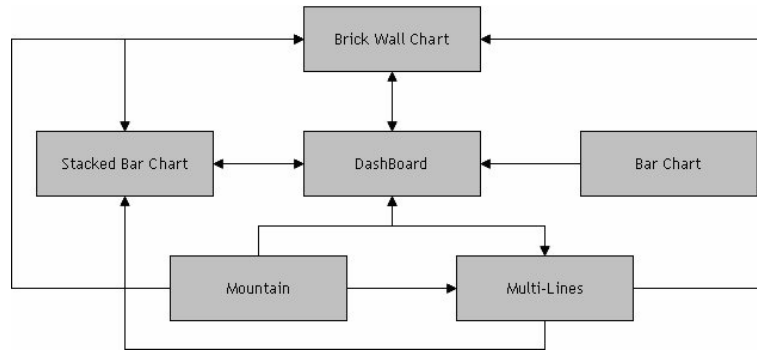


Figure 6-15 : Passage avec ajout de paramètres.

Exemple : supposons que nous nous trouvons en « Mountain » avec 1 paramètre sélectionné, il serait possible (sans rien changer) de passer à « Brick Wall Chart », « Stacked Bar Chart », « Dashboard » et « Multi-Lines » car ces visualisations sont réalisables avec ce nombre de paramètres sélectionnés. Cependant, pour afficher « Stacked Bar Chart » il faudrait alors ajouter un paramètre à la sélection.

6.4.4 Cas 4 : Passage avec suppression de paramètres

L'utilisateur peut passer d'une visualisation à l'autre moyennant une modification de paramètres. Dans ce cas, il s'agit du retrait d'un ou de plusieurs paramètres. Il faut passer à une sélection plus restreinte des paramètres relatifs à la série temporelle.

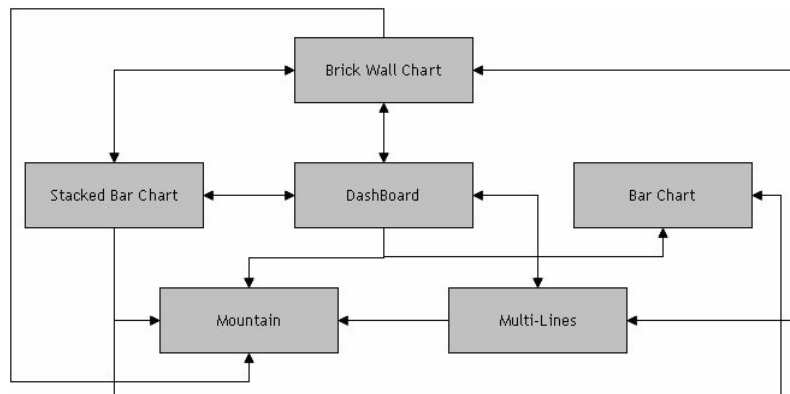


Figure 6-16 : Passage avec suppression de paramètres.

Exemple : supposons que nous nous trouvons en « Brick Wall Chart » avec 4 paramètres sélectionnés, il serait possible (sans rien changer) de passer à « Dashboard » et « Multi-Lines » car ces visualisations sont réalisables avec ce nombre de paramètres sélectionnés. Cependant, pour afficher « Stacked Bar Chart » il faudrait alors supprimer 2 paramètres de la sélection. Pour afficher « Mountain », il faudrait en supprimer 3.

7 Implémentation

7.1 Choix de l'outil de développement

Lors du chapitre 5, nous avons présenté et détaillé les caractéristiques des deux plateformes de développement envisagée (« Microsoft .NET » et « J2ME ») pour la réalisation de ce travail. Pour diverses raisons, notre choix s'est porté sur la plateforme « Microsoft .NET ».

Du point de vue de l'architecture

- « .NET » est un standard proposé par la société Microsoft, pour le développement d'applications. Une grande majorité des applications actuelles sont développées sous ce label, il est donc intéressant d'utiliser un outil standardisé et répandu.
- Le nouveau modèle de programmation, basé sur SOAP (Simple Object Access Protocol) et les Web Services change fondamentalement la façon de concevoir ses applications, et ouvre la voie vers un nouveau métier : la fourniture de services Web en ligne. Ces « Web Services » sont basés sur le protocole XML.
- Les applications compilées sous la forme de code intermédiaire se présentent sous la forme de binaires exécutables portables (PE, Portable Executable).

Du point de vue de son environnement de développement

- L'environnement de développement unifié qui permet d'écrire aussi bien des programmes Windows (Winform) que des sites Web dynamiques (ASP.NET) s'appelle « Visual Studio .NET ».
- De plus, « Visual Studio .NET » offre la possibilité de travailler avec un émulateur intégré, ce qui facilitera grandement l'implémentation.

Du point de vue des langages

- « .NET » est multi-langage et met à disposition une vaste gamme de langages de programmation.

7.2 Choix du langage de programmation

Grâce au CLR, la plateforme « .NET » est indépendante de tout langage de programmation et supporte nativement un grand nombre de langages de programmation.

7.2.1 Il existe de nombreux langages, lequel choisir ?

Tous les langages dans « .NET » sont égaux. Le choix du langage n'aura donc aucune influence sur ce qu'il sera possible de faire, ni sur les performances. VB.NET et C# sont toutefois privilégiés par « Visual Studio .NET » car celui-ci sait générer automatiquement du code VB.NET ou C#.

Ils sont égaux car, quelque soit le langage utilisé, ils permettent de faire les mêmes choses et ce, avec la même rapidité. Lors de la compilation d'un programme, celui-ci est traduit en pseudo code MSIL (Microsoft Intermediate Language) (cfr Section 5.5.2.3). Sur des algorithmes simples, il y a de fortes chances que la version VB.NET, C# ou C++ produise le même code MSIL et donc offre les mêmes performances.

7.2.2 Common Language Specification

Pour qu'un langage soit éligible au rang de langage supporté par la plateforme « .NET », il faut qu'il fournisse un ensemble de possibilités, de constructions qui sont recensées dans un contrat de service dénommé CLS : Common Language Specification. Pour ajouter un langage à « .NET », il suffit, a priori, de satisfaire aux exigences de la CLS, et que quelqu'un développe un compilateur depuis ce langage vers MSIL.

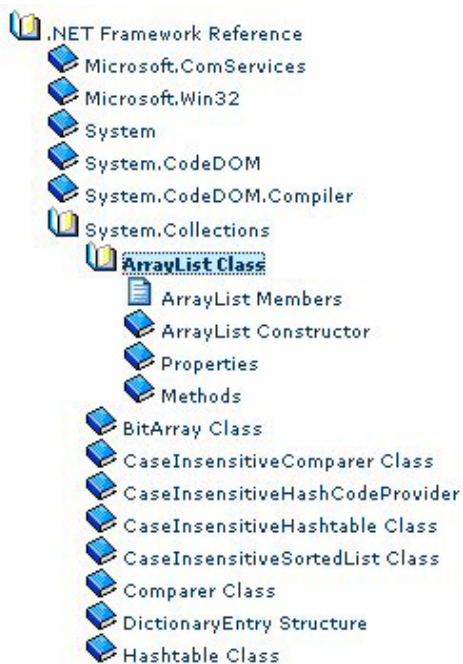
Ceci semble anodin en première lecture, mais les contraintes imposées par le respect de la CLS aux différents langages de « .NET » font, par exemple, que Visual Basic .NET soit finalement un nouveau langage qui ne garde pratiquement de Visual Basic 6 que la syntaxe.

Une autre conséquence directe de la compilation de tous les langages de « .NET » sous la forme d'un même code intermédiaire, est qu'une classe écrite dans un langage peut être dérivée dans un autre langage. Nous pouvons instancier dans un langage un objet d'une classe écrite dans un autre langage.

7.2.3 Le langage C#

Le langage C# créé par Microsoft dans le but de concurrencer Java, est devenu un standard approuvé par la spécification ECMA-334 de l'association européenne ECMA (European Computer Manufacturer's Association) en décembre 2001. Il permet de développer des applications Windows et Web en se basant sur l'architecture « .NET ». Ce langage a été conçu dans le but d'être indépendant de la plateforme et de l'environnement d'exécution. Normalement, le compilateur C# devrait pouvoir fonctionner partout où les types et fonctionnalités spécifiés dans l'ECMA-334 sont correctement pris en charge par l'environnement d'exécution.

Actuellement, l'environnement d'exécution du langage commun (CLR : Common Language Runtime) de la technologie « .NET » est évidemment, le domaine de prédilection du langage C#. Le mécanisme de traitement du code source écrit en C# par la plateforme « Microsoft.NET », produit lors de la phase de compilation, un code binaire particulier dénommé MSIL (MicroSoft Intermediate Language). Celui-ci sera interprété ensuite par le CLR chargé de la transformation en instructions exécutable par la machine.



La gestion de la mémoire par l'utilisation des pointeurs a été maintenue tout en veillant à y intégrer un dispositif d'appel explicite par l'utilisation de mots-clés spécifiques. Le concept d'héritage multiple dans la programmation orientée objet a été abandonné au profit de l'utilisation d'interfaces. Le langage C# permet une programmation orientée objet simple et moderne. Dérivé des langages de programmation C et C++, C# semble familier aux habitués du développement en C, C++ et Java. Il ne sera certainement pas plus difficile à apprendre pour les autres. Le langage C# ne comporte que 77 mots-clés. De plus, la programmation en C# se révèle intuitive, familière et moins verbeuse. Elle produit du code se démarquant par sa lisibilité et sa concision.

La technologie « .NET » met également à la disposition du programmeur de nombreuses classes du « .NET Framework », comprenant une pléthore de méthodes et propriétés exploitables dans des programmes C#. Le langage C# est une savante combinaison des langages C, C++ et Java en expurgant la plupart des sources d'erreurs et en améliorant certaines fonctionnalités. Il reprend les majeures caractéristiques, tout en simplifiant son utilisation. Il s'agit donc d'un langage orienté objet. Certaines notions utilisées par le C#, telles que le Garbage Collector, sont par ailleurs empruntées au Java.

7.3 Détails techniques de l'implémentation

7.3.1 Chargement et sauvegarde des données.

Lors du développement de l'application sur le PDA, une des exigences étaient d'utiliser des fichiers XML pour charger et stocker les données, qu'elles soient relatives à la configuration de l'application elle-même (profils, images, sons, fichiers d'aides,...) ou propres aux utilisateurs (séries temporelles).

7.3.1.1 Définition et utilité

Le format XML (eXtensible Markup Language) est un format de stockage des données élaboré récemment par consortium W3C (groupe de pilotage des normes pour le Web). Ce format garantit une parfaite compatibilité avec les outils informatiques contemporains (bases de données, langages de programmation). La syntaxe d'un fichier XML peut être décrite dans un fichier DTD (Déclaration des Types de Documents). Employé avec un programme informatique adéquat (parseur validant), ce fichier DTD contrôle la conformité du fichier XML à la syntaxe attendue par une application et permet des échanges de données parfaitement sûrs.

XML, à l'aide de son langage et de sa syntaxe, offre la possibilité de décrire de l'information structurée indépendamment du médium de diffusion. Dans le jargon XML cela s'appelle

« write once, publish many », cela va sans doute rappeler le fameux « write once, run anywhere » du langage Java.

L'avantage du fichier XML par rapport aux fichiers de format TXT est situé dans sa structure. Là où les fichiers textes contiennent l'information en vrac ligne après ligne, les fichiers XML permettent de décrire une structure dans les informations grâce aux balises. Grâce à ces balises, l'ordinateur est également capable d'en traiter le contenu (et de bien séparer les informations).

C'est un des avantages du XML : c'est l'un des rares formats qui peut être lu aussi bien par un homme que par un ordinateur. De plus, en se mettant d'accord sur les balises à utiliser, nous pouvons employer XML pour échanger des informations entre différentes personnes et logiciels. XML, grâce à l'utilisation de l'encodage d'UTF-8, supporte très bien tous les alphabets du monde.

7.3.1.2 Exemples

Voici quelques exemples de la structure des fichiers XML utilisés par l'application :

Fichier « index.xml »

Ce fichier constitue le fichier de base du chargement de l'application. Nous y trouvons les informations permettant de savoir quelles sont les séries temporelles ainsi que les visualisations présentes dans l'application.

```
<?xml version="1.0" encoding="UTF-8"?>
<INDEX>
  <NBR_TS>3</NBR_TS>
  <TS_LIST>
    <TS>Police</TS>
    <TS>Stock Market</TS>
    <TS>Weather</TS>
  </TS_LIST>
  <NBR_VIZ>6</NBR_VIZ>
  <VIZ_LIST>
    <VIZ>DashBoard</VIZ>
    <VIZ>Mountain</VIZ>
    <VIZ>Multi-Lines</VIZ>
    <VIZ>Bar Chart</VIZ>
    <VIZ>Stacked Bar Chart</VIZ>
    <VIZ>Brick Wall Chart</VIZ>
  </VIZ_LIST>
</INDEX>
```

Fichier « Profile List.xml »

Ce fichier contient la liste des profils créés par les utilisateurs. A chaque création, modification ou suppression par un (nouvel) utilisateur, ce fichier est mis à jour.

```
<?xml version="1.0" encoding="UTF-8"?>
<PROFILE_LIST>
  <USER>Archibald</USER>
  <USER>Colombo</USER>
  <USER>Stephane</USER>
</PROFILE_LIST>
```

Fichier « Stephane Profile.xml »

Exemple d'un profil créé par un utilisateur. Nous pouvons constater qu'il contient les informations dont l'application a besoin pour charger la partie visualisation. Nous pouvons y retrouver la série temporelle, la visualisation par défaut, la langue utilisée par l'aide contextuelle mais également les dates de création et modification du profil.

```

<PROFILE>
  <USER_NAME>Stephane</USER_NAME>
  <USER_HAND>Right</USER_HAND>
  <USER_LANGUAGE>French</USER_LANGUAGE>
  <USER_TIMESERIES>Weather</USER_TIMESERIES>
  <USER_VISUALIZATION>Brick Wall Chart</USER_VISUALIZATION>
  <USER_CREATION>12-12-2004</USER_CREATION>
  <USER_MODIFICATION>14-12-2004</USER_MODIFICATION>
</PROFILE>

```

Fichier « Stock Market.xml »

Une fois que l'utilisateur a choisi dans son profil la série temporelle qu'il veut analyser, l'application va aller chercher dans le fichier *Nom_de_la_serie_temporelle.xml* les renseignements dont elle va avoir besoin. Nous y retrouvons les informations suivantes :

- Le nombre de jours que comporte la série temporelle
- Le nombre de variables qu'elle contient ainsi que la liste
- Le nombre de paramètres qu'elle contient ainsi que la liste

```

<?xml version="1.0" encoding="UTF-8"?>
<TIME_SERIE>
  <TS_NAME>Stock Market</TS_NAME>
  <TS_SIZE>253</TS_SIZE>
  <NBR_VARIABLES>2</NBR_VARIABLES>
  <VARIABLES_LIST>
    <VARIABLE>IBM</VARIABLE>
    <VARIABLE>AMD</VARIABLE>
  </VARIABLES_LIST>
  <NBR_PARAMETERS>4</NBR_PARAMETERS>
  <PARAMETERS_LIST>
    <PARAMETER>Closing Price</PARAMETER>
    <PARAMETER>Highest Price</PARAMETER>
    <PARAMETER>Lowest Price</PARAMETER>
    <PARAMETER>Opening Price</PARAMETER>
  </PARAMETERS_LIST>
</TIME_SERIE>

```

Fichier « IBM 0.xml »

Voici un exemple typique de la structure d'un fichier contenant les valeurs d'une série temporelle. Pour chaque balise représentant un jour, nous retrouvons l'intitulé de la date mais également les valeurs de chaque paramètre pour ce jour. Le nombre de balise <DAY> dépend du nombre de jours que contient la série temporelle.

```

<?xml version="1.0" encoding="UTF-8"?>
<VARIABLE>
  <NAME>IBM</NAME>
  <DATASET>
    <DAY>
      <DATA>2-Jan-04</DATA>
      <DATA>2011,08</DATA>
      <DATA>2022,37</DATA>
      <DATA>1999,77</DATA>
      <DATA>2006,68</DATA>
    </DAY>
    <DAY>
      <DATA>5-Jan-04</DATA>
      <DATA>2020,78</DATA>
      <DATA>2047,36</DATA>
      <DATA>2020,78</DATA>
      <DATA>2047,36</DATA>
    </DAY>
  </DATASET>
</VARIABLE>

```

7.3.2 Classes principales de l'application

Dans le paragraphe qui suit, une brève présentation des différentes classes de l'application considérées comme principales, soit parce qu'elles occupent une position centrale dans l'architecture de l'application, soit parce qu'elles représentent une visualisation.

7.3.2.1 MenuFrame

Cette classe est utilisée pour gérer le menu principal de l'application qui donne accès à la gestion des profils, à l'explorateur de fichiers ainsi qu'à la partie visualisation de l'application.

7.3.2.2 TSFrame

Voici la classe clé de l'application. Elle se trouve au centre de toutes les autres classes et elle permet la gestion de la partie visualisation (création des menus, gestion des événements, affichage des visualisation, gestion des interactions avec l'utilisateur).

7.3.2.3 Calculs

La classe « Calculs » est appelée chaque fois qu'une configuration est réalisée par un utilisateur. Elle va lire les fichiers XML contenant les données relatives à la série temporelle afin de répartir les données brutes et créer des listes pour chaque paramètre ainsi que des listes pour les extrema.

7.3.2.4 CrossRoad

Cette classe sert de point intermédiaire entre la classe principale « TSFrame » et les classes de création de visualisations. En effet, c'est lors de l'exécution de cette classe que sera « redirigée » l'instruction de création d'une visualisation en provenance de « TSFrame »

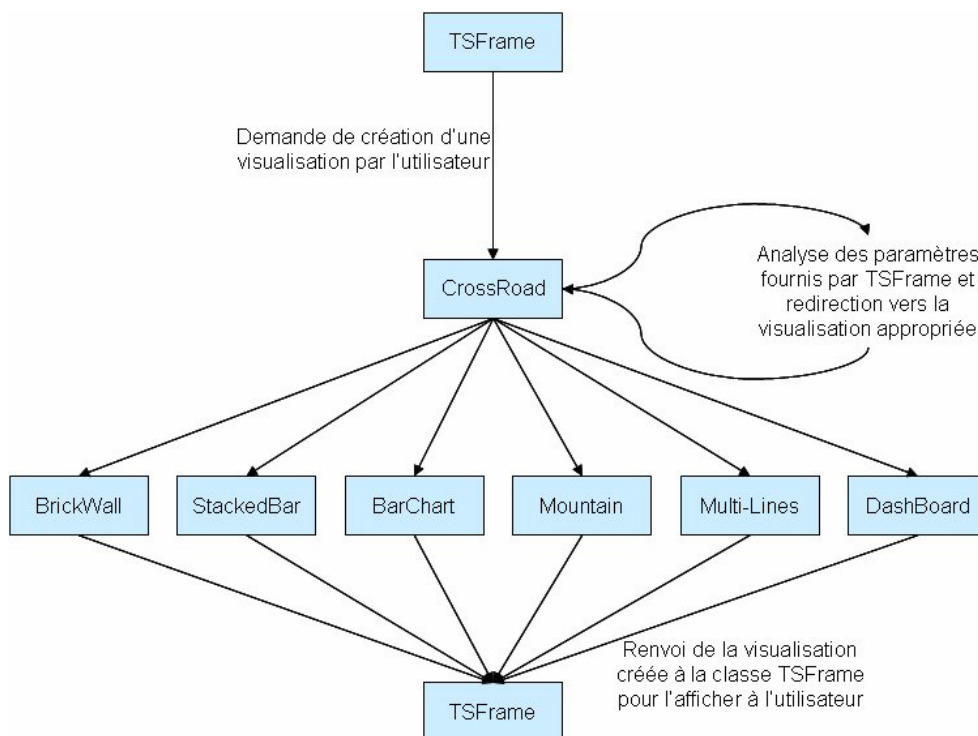


Figure 7-1 : Aiguillage de l'instruction de création d'une visualisation selon son type.

7.3.2.5 NodeGraph

Cette classe a été créée dans le but d'implémenter les relations entre les visualisations décrites dans la section 5.6. Son fonctionnement est très simple : lorsque cette classe est appelée, elle analyse la visualisation qui est actuellement en cours d'utilisation et va rechercher dans la configuration de l'utilisateur le nombre de paramètres qui ont été sélectionnés. Elle va ensuite calculer (selon les schémas possibles) les différentes options qui seront offertes pour switcher d'une visualisation à l'autre.

7.3.2.6 BrickWall, StackedBar, BarChart, Mountain, Multi-Lines, DashBoard

De façon à obtenir une architecture plus simple, chacune des visualisations possède sa propre classe avec son code. De cette façon, il est aisé de rediriger l'instruction de création d'une visualisation vers la classe adéquate (et cela, grâce à la classe « CrossRoad » décrite ci-dessus). Les classes ont quasiment toutes la même structure (excepté « DashBoard ») dans les étapes de conception, à savoir :

- Calcul des paramètres à afficher
- Calcul de l'échelle de valeurs
- Création et affichage des axes
- (Création et affichage de l'échelle de couleur)
- Création des formes et ajout de la couleur.
- Affichage de l'échelle sur la visualisation

7.3.3 Codage des couleurs

L'échelle de couleur et l'algorithme, permettant de construire cette échelle, ont été tirés du travail de Frédéric Randolet [RANDOLET04] :

L'échelle de couleur est constituée respectivement de blanc-jaune-rouge-noir. Comme une couleur est un triplet d'entiers compris entre 0 et 255 (conversion RGB d'une couleur), nous pouvons y faire correspondre les triplets suivants : (255,255,255)-(255,0,255)-(255,0,0)-(0,0,0).

Nous aurons donc trois intervalles de couleurs :

- (255,255,255) → (255,0,255)
- (255,0,255) → (255,0,0)
- (255,0,0) → (0,0,0)

Il suffit dès lors de remettre les valeurs de l'historique sur une échelle compatible avec l'échelle de valeur. Les valeurs oscillent entre une borne supérieure (bs) et une borne inférieure (bi). Nous pouvons alors convertir les valeurs en pourcent :

$$x \in [bi, bs]: \textit{pourcent} = x * 100 \div (bs - bi)$$

Comme il existe trois intervalles de valeurs différentes pour les couleurs, les pourcentages obtenus devront être divisés en trois parties également. Ensuite, il sera facile de faire la correspondance avec un triplet.

```

/* i étant compris entre 0 et 767*/
switch(i/256)
{
  case 0 :
    return Color.FromArgb(255, 255-(i%255), 255) ;
  case 1 :
    return Color.FromArgb(255, 0, 255-(i%255)) ;
  case 2 :
    return Color.FromArgb(255-(i%255), 0, 0) ;
}

```

De cette manière, nous obtenons pour toute valeur, une couleur qui sera affichée à l'écran.

7.4 Limites et améliorations possibles

7.4.1 Au niveau du paramétrage de l'application

Une amélioration possible de l'application serait son paramétrage. Il serait intéressant de permettre à l'utilisateur de décider d'un nombre beaucoup plus important de paramètres au démarrage de l'application. Ce paramétrage pourrait s'effectuer en fonction de différentes situations, que ce soit au niveau du PDA en lui-même ou de certaines préférences de l'utilisateur. Plusieurs propositions méritent d'être soulevées :

- Les PDA's qui sont faits aujourd'hui possèdent une résolution différente de celle des PDA's plus anciens. Il serait intéressant de permettre à l'utilisateur de sélectionner la résolution qu'il désire utiliser pour l'application (tel que ce qui se fait pour un ordinateur) et ainsi adapter les visualisations afin qu'elles offrent une meilleure qualité visuelle.
- Bien qu'elles soient utilisées dans le même but, à savoir des informations au cours d'une période donnée, chaque série temporelle (identique dans la forme) est différente dans le contenu. Suite à cela, les informations que nous souhaiterions extrapoler de ces séries peuvent être tout à fait divergentes. Bref, il faudrait proposer à l'utilisateur de cocher des indices parmi une liste et de choisir les paramètres qu'il voudrait voir apparaître sur la visualisation.

En effet, voici deux exemples illustrant ces propos :

- En météorologie, il est fréquent de présenter les informations en utilisant des moyennes réalisées sur certaines périodes. Il faudrait pouvoir afficher ces informations calculées à partir des données brutes fournies à l'application.
- Dans le secteur boursier, savoir combien d'échanges se sont produits tel jour en fonction du cours de l'action affiché, serait une information très utile.

→ Nous pouvons constater que ces deux informations sont tout à fait différentes, non seulement leur signification n'est pas la même mais la façon dont l'information est représentée est aussi différente. Il est donc impossible de les programmer toutes les deux de façon identique.

7.4.2 Au niveau du son

L'objectif de ce mémoire ne consistait pas à utiliser le son comme un élément d'information à part entière. En effet, il ne représente pas une particularité du PDA (contrairement à d'autres appareils mobiles tels que le téléphone portable).

Toutefois, il doit être possible de d'utiliser le son afin de renseigner plus efficacement l'utilisateur. De ce point de vue, différentes variables acoustiques comme le volume, le timbre

et la position peuvent être utilisées pour représenter des aspects qualitatifs, quantitatifs et catégoriques de l'information. Ces variables sont aussi bénéfiques dans les circonstances où les représentations visuelles ne sont pas possibles pour afficher des données ou bien pour enrichir des réalisations graphiques.

A partir de cela, nous pourrions très bien envisager de nous servir du son pour certaines visualisations telles que Brick Wall chart ou Mountain. Il permettrait à l'utilisateur à voir l'évolution d'un mois, grâce à un son continu qui serait joué lorsque l'utilisateur consulte un mois de l'année. Ce son augmenterait en cas de valeur élevée et diminuerait en cas de valeur faible.

Une autre idée serait d'enregistrer un son, émis par un automate, prononçant chaque valeur de paramètre. Dès lors, l'utilisateur pourrait faire « défiler » les valeurs de chaque jour en les écoutant et nous pourrions, ainsi, gagner de la place pour les visualisations. Cependant, cette option est fortement déconseillée suite aux caractéristiques de la mémoire à court terme. Bien entendu, il est tout à fait possible d'afficher les informations sur l'écran et de faire jouer le son (ou une voix prononçant les différentes valeurs).

Le son peut également être utile dans la navigation ainsi que dans la gestion des menus. Celle-ci peut être améliorée par le son. Nous pourrions imaginer un son propre à chaque action sur un menu. Cela aiderait l'utilisateur à déterminer où il se situe dans les différents menus hiérarchiques. Néanmoins, les utilisateurs ont tendance à diminuer (voire couper) le son des menus car ils les trouvent inappropriés et agaçant à la longue. Il faut donc trouver le juste milieu entre le côté informatif d'une part et, le côté lassant et répétitif d'autre part.

Notre canal visuel est limité, et ceci, à différents niveaux. Tout d'abord, nous avons, plus ou moins, la possibilité de vision à 180°. Par contre, entendre est non-directionnel [DEM&SCHO01] : nous ne pouvons pas voir ce qui est derrière nous, mais nous pouvons l'entendre et localiser sa position de façon précise [BREWSTER94] et [BREWSTER99]. Ensuite, quand une personne regarde son écran d'ordinateur, la plupart des charges de détails sont brouillées par le secteur particulier sur lequel elle se concentre. Le canal visuel est en effet très sélectif. Si un détail change soudainement (une icône de plateau de système commençant à clignoter par exemple), cette personne ne le remarquera pas forcément. Mais si un petit signal d'alarme est joué, il attirera probablement plus son attention [BREWSTER94] et [BREWSTER99].

Donc, comme nous le constatons, les personnes ont besoin d'utiliser leurs yeux pour réaliser des tâches ou, devrions-nous dire, pour réaliser « une tâche à la fois ». Mais grâce à leur audition, ils peuvent faire plusieurs choses simultanément, car l'audition ne surcharge pas la vue [NOIRHOMME00]. Etant donné ses caractéristiques, ajouter le bruit aux représentations visuelles offre la dimensionnalité additionnelle : quand une information complexe est présentée simultanément sous une forme visuelle et auditive, elle exige de manière significative moins d'effort pour l'interpréter [DEM&SCHO01].

D'ailleurs, la mémoire auditive à court terme est plus efficace que la mémoire visuelle à court terme [NOIRHOMME00]. Particulièrement avec les PDA's, le bruit est une manière intéressante de recevoir de l'information et du feedback. Cela, tout en se concentrant sur une tâche spécifique, qui exige toutes les capacités visuelles, ou l'utilisation des deux mains. Cependant, l'audition humaine ne perçoit pas facilement des valeurs absolues telles la hauteur et l'intensité. Elle distingue plutôt des changements de paramètres. Et quand elle représente des données, les bruits non vocaux sont limités par la perception auditive. La résolution des

bruits non vocaux dépendra de la gamme des valeurs pour représenter l'information, qui ne s'applique pas à la parole [NOIRHOMME00].

De plus, le bruit est passager tandis que les graphiques sont persistants. En d'autres termes, une fois le bruit produit, il disparaît tandis que les graphiques restent. Un autre élément à garder à l'esprit est le réglage du volume. Il est nécessaire de surveiller le niveau sonore ambiant autour du dispositif et d'adapter son volume de rendement convenablement. La réduction des problèmes de l'ennui dus au volume excessif ou insuffisant du bruit est une étape très importante en augmentant l'acceptabilité des interfaces auditives [BREWSTER00].

8 Evaluation de l'application

8.1 Objectifs

Après avoir débuté par une analyse des besoins pour s'assurer de la nécessité de la nouvelle application, après avoir élaboré et développé une architecture, il est tout à fait logique de terminer le processus de développement par une évaluation de l'application par les utilisateurs. Cette évaluation va permettre de vérifier si tous les objectifs imposés aux départs ont été respectés et si d'autres améliorations peuvent y être apportées.

L'application ayant pour but de développer plusieurs visualisations aux caractéristiques différentes, il sera intéressant de les comparer entre elles, cela permettra d'observer les différences qui existent entre ces visualisations, que ce soit à un niveau objectif (leur performance) ou à un niveau subjectif (les préférences des utilisateurs).

Les tests d'évaluation ont permis d'apporter une réponse aux questions suivantes :

1. Les visualisations permettent-elles une analyse correcte des séries temporelles.
2. Les préférences des utilisateurs quant à (aux) la (les) visualisation(s) qu'ils préfèrent utiliser pour analyser une série temporelle.
3. Obtenir de la part des utilisateurs des remarques sur l'application, que ce soit au niveau de l'utilisabilité, de l'utilité de l'application ou encore au niveau d'améliorations possibles.

8.2 Echantillon d'utilisateurs

L'échantillon d'utilisateurs dont nous nous sommes servis pour réaliser l'évaluation est composé de 23 personnes dont une majorité d'hommes (environ 2/3 de l'échantillon). Toutes les personnes qui ont participé à ces tests ont été volontaires pour les réaliser. Dans cet échantillon, la moyenne d'âge des participants est de 27,6 ans et la tranche d'âge s'étend de 17 à 50 ans. 9 personnes parmi l'échantillon ont entre 22 et 25 ans.

Les domaines de profession des utilisateurs sont variés. Parmi les utilisateurs, nous retrouvons une informaticienne, des infirmiers et infirmières, un policier, une institutrice, une journaliste, un chercheur en biologie, un assistant/chercheur en mathématique, un ouvrier en bâtiment, un comptable, un géographe mais également des étudiant(e)s en informatique, kinésithérapie, logopédie et interprétariat. Toutefois, la branche dominante est celle de l'informatique. En effet, 6 personnes sont issues de cette branche. Les utilisateurs issus de l'informatique vont permettre d'avoir une appréciation plus technique sur l'application tandis que les autres pourront certainement apporter une évaluation plus centrée vers l'aspect pratique.

L'expérience en informatique varie fortement (pour ceux en possédant une), mais nous pouvons estimer que l'utilisation d'un ordinateur peut être considérée comme familière. Pour un grand nombre des utilisateurs, l'expérience dans les Mobiles Devices correspond à l'expérience du GSM. Chacun d'entre eux possédant un téléphone portable. Quant à leur expérience avec les PDA, hormis quelques brèves manipulations pour certains, elle est inexistante pour les autres. Il est donc question d'une initiation lors de cette évaluation.

En ce qui concerne le background sonore et graphique, 15 personnes n'ont absolument aucune expérience dans le domaine musical mais par contre, l'expérience des jeux électroniques est beaucoup plus répandue parmi les utilisateurs.

8.3 Présentation du questionnaire

Pour permettre une simplification du test pour les utilisateurs et une comparaison optimale, une série temporelle spéciale a été créée afin que chaque utilisateur travaille sur les mêmes données et que les résultats ne soient pas faussés.

De ce fait, grâce aux utilisateurs travaillant sur les mêmes données, les comparaisons de résultats seront possibles et il sera permis de tirer des conclusions à la fin de ce chapitre.

Les tests d'évaluations vont comporter plusieurs parties afin de s'assurer que l'application est bien acceptée par les utilisateurs et qu'elle permet une analyse correcte des séries temporelles.

8.3.1 Informations personnelles

Premièrement, il sera demandé aux volontaires quelques informations personnelles (nom, âge, genre, titre/position, domaine d'activité et adresse e-mail). Celles-ci seront utilisées pour identifier les sujets et valider leurs réponses.

Nom – Prénom			
Age		Genre	M - F
Titre / Position			
Domaine d'activité / d'études			
Adresse e-mail			

Tableau 8-1 : Informations personnelles.

8.3.2 Expériences précédentes dans les services mobiles

La seconde partie consiste à analyser plus précisément le profil graphique et mobile des utilisateurs. Il leur sera demandé quelle est leur expérience en informatique, des GSM, des PDA, ainsi que ce qu'ils possèdent comme expérience musicale et graphique et enfin, quelle est leur expérience avec les jeux vidéo et si ceux-ci utilisent des visualisations (ou des graphiques).

Pour mieux interpréter les termes utilisés dans cette partie du profil, voici quelques précisions :

- Expérience en informatique :

Le terme « informatique » n'est pas à prendre au sens des études dans le domaine de l'informatique mais plutôt de la familiarisation avec cet outil (utilisation d'ordinateurs, de logiciels, ...). Cela permet de savoir si l'utilisateur se lance dans l'évaluation sans acquis préalables ou s'il maîtrise les principes fondamentaux de la manipulation d'un ordinateur.

- **Expérience des GSM :**
L'expérience des GSM nous permet de savoir si les personnes sont familiarisées à la manipulation de petites touches ainsi qu'à l'utilisation d'un écran de petite taille.
- **Expérience des PDA :**
Un PDA possède plus de fonctionnalités qu'un GSM et il est intéressant de savoir si les utilisateurs sont habitués (ou pas) à utiliser les fonctionnalités de base de ces appareils mobiles.
- **Expérience des jeux électroniques (avec graphique ou visualisation) :**
L'expérience dans le domaine des jeux peut se révéler intéressante à connaître. Plus particulièrement si ces jeux utilisent des graphiques ou des visualisations pour représenter les informations.

Expérience en informatique	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Si oui, depuis combien d'années ?					
Expérience des GSM	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Expérience des PDA	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Expérience musicale	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Expérience graphique	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Expérience avec les jeux électroniques	<input type="checkbox"/> Aucune	<input type="checkbox"/> Rare	<input type="checkbox"/> Occasionnelle	<input type="checkbox"/> Fréquente	<input type="checkbox"/> Quotidienne
Si oui, certains utilisent-ils des graphiques ou visualisations ?					

Tableau 8-2 : Expériences précédentes dans les graphismes et les services mobiles.

8.3.3 Scénarii

Les utilisateurs devront se soumettre à divers scénarii qu'il leur faudra, pour certain, réaliser en un minimum de temps ou de tentatives. Ces scénarii devraient permettre de cerner les points forts (et faibles) des visualisations présentées.

8.3.3.1 Phase 1

La première phase des tests consiste à fournir à l'utilisateur un aperçu global du mode de fonctionnement de l'application. Cela servira surtout pour toutes les fonctionnalités existantes autres que les visualisations. Il devra notamment pouvoir se créer un profil (ou le cas échéant en modifier un selon des paramètres donnés), parcourir l'arborescence des fichiers ou encore utiliser les fonctionnalités d'aide. Cette première phase devrait permettre à l'utilisateur une prise en main suffisante de l'application afin d'être capable d'effectuer correctement les tests suivants.

8.3.3.2 Phase 2

Lors de cette phase, l'utilisateur doit déterminer le jour qui, selon lui et ce qu'il voit à l'écran, correspond à la valeur la plus élevée ainsi qu'à la valeur la plus basse. Il s'agit clairement de la phase la plus objective de toutes celles présentes dans l'évaluation de l'application. Les tests sont élaborés de telle façon qu'il n'existe qu'une solution possible, excepté pour le dernier test

de chaque série. L'utilisateur n'est pas limité dans le nombre d'essais mais plutôt dans le temps car une réponse est demandée en général après un laps de temps de 30s de réflexion.

L'utilisateur sera aidé dans son évaluation par des sons. Deux sons interviendront dans l'application lorsque la valeur maximale (ou minimale) d'un jour est sélectionnée par l'utilisateur et auront pour but de lui faciliter ses recherches de maximum et de minimum.

8.3.3.3 Phase 3

La troisième étape consiste à permettre à l'utilisateur d'appréhender les couleurs et de les utiliser. La meilleure méthode, pour vérifier si l'assimilation des couleurs est réalisée, est de comparer plusieurs couleurs (deux en l'occurrence). L'utilisateur devra prendre connaissance de la valeur à une date choisie arbitrairement.

Il devra apprécier l'évolution des valeurs à partir de cette date, d'abord par rapport à un mois, ensuite une semaine, et finalement un jour auparavant. Cette phase permet de constater si les couleurs mises en valeur par les visualisations sont appréciables et si celles-ci sont adaptées à la comparaison des différentes variables des séries temporelles.

8.3.3.4 Phase 4

La quatrième partie regroupe les phases 2 et 3 en même temps. L'intérêt de cette phase est de déterminer si l'application permet à l'utilisateur d'obtenir un aperçu correct de la série après un rapide coup d'œil. Il n'est nullement question ici de chercher une valeur bien précise ou un jour précis mais de percevoir des variations.

Dans ce cas, il est demandé à l'utilisateur de signaler le mois qui, selon lui, possède la variation la plus forte. Par variation, il est entendu la plus forte différence entre la valeur au premier jour de chaque mois par rapport au dernier jour de ce même mois. L'utilisateur sera donc amené à juger les différences entre les couleurs. La réponse doit être donnée suivant une impression, « quel est le mois qui semble subir la plus grosse variation au premier coup d'œil ».

L'utilisateur peut consulter les valeurs exactes via les info-bulles mais ce n'est pas nécessaire étant donné que la réponse doit être donnée suivant une impression. Ce test permet de voir que les couleurs représentent bien la situation réelle.

8.3.3.5 Phase 5

Dans cette dernière phase, l'utilisateur doit simplement pointer avec le stylo un jour précis. Cela permet de calculer la précision des visualisations et de les comparer. En effet, le nombre de tentatives est comptabilisé pour chaque visualisation. Le jour à trouver est toujours le même, ce qui permet de comparer les visualisations entre elles au niveau de la précision.

Il y a en tout 5 séries de questions. Les séries suivent le même scénario et posent les mêmes questions à chaque fois. Afin de pouvoir comparer l'efficacité des visualisations, les questionnaires correspondent aux mêmes variables (d'une série temporelle) et ce, pour chaque utilisateur.

En réalité, chaque variable est associée à une visualisation pour ces tests. Chaque utilisateur utilisera toutes les visualisations mais dans des ordres différents. Les scénarii seront donc organisés de la manière suivante :

1. Utilisateur 1 :
 - Valeur de ... avec Brick Wall Chart.
 - Valeur de ... avec Stacked Bar Chart.
 - Valeur de ... avec Mountain.
 - Valeur de ... avec Multi-Lines.
 - Valeur de ... avec une visualisation au choix.
2. Utilisateur 2 :
 - Valeur de ... avec Multi-Lines.
 - Valeur de ... avec Brick Wall Chart.
 - Valeur de ... avec Stacked Bar Chart.
 - Valeur de ... avec Mountain.
 - Valeur de ... avec une visualisation au choix.
3. Utilisateur 3 :
 - Valeur de ... avec Mountain.
 - Valeur de ... avec Multi-Lines.
 - Valeur de ... avec Brick Wall Chart.
 - Valeur de ... avec Stacked Bar Chart.
 - Valeur de ... avec une visualisation au choix.
4. Utilisateur 4 :
 - Valeur de ... avec Stacked Bar Chart.
 - Valeur de ... avec Mountain.
 - Valeur de ... avec Multi-Lines.
 - Valeur de ... avec Brick Wall Chart.
 - Valeur de ... avec une visualisation au choix.
5. Utilisateur 5 : ...

Scénario 0 (en sec)		Scénario 3.d (en sec)	
Scénario 1 (en sec)		Scénario 3.e (en sec)	
Scénario 2.a (en sec)		Scénario 3.f (en sec)	
Scénario 2.b (en sec)		Scénario 4 (en sec)	
Scénario 3.a (en sec)		Scénario 5.a (en #)	
Scénario 3.b (en sec)		Scénario 5.b (en #)	
Scénario 3.c (en sec)		Scénario 5.c (en #)	

Tableau 8-3 : Résultats des scénarii.

8.3.4 Evaluation générale de l'interface

Après avoir interagi avec l'application, il sera demandé aux utilisateurs leurs impressions à propos de l'interface générale de l'application. Nous essaierons de déterminer quelles parties de l'interface se sont avérées faciles, agréables à manipuler et lesquelles ne l'ont pas été.

Lisibilité	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Excellente
Exploration des visualisations	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Excellente
Navigation des menus	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Excellente
Gestion utilisateurs (profils, explorateurs de fichiers)	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Excellente

Tableau 8-4 : Evaluation générale de l'interface.

8.3.5 Evaluation des visualisations

Afin de pouvoir réaliser des tests de préférences, les utilisateurs devront fournir une appréciation pour chacune des visualisations. Cela nous permettra de savoir comment ces dernières sont perçues par les utilisateurs au niveau des analogies, des couleurs et des formes.

Brick Wall Chart	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne
Stacked Bar Chart	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne
Bar Chart	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne
Mountain	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne
Multi-Lines	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne
DashBoard	<input type="checkbox"/> Très mauvaise	<input type="checkbox"/> Mauvaise	<input type="checkbox"/> Moyenne	<input type="checkbox"/> Bonne	<input type="checkbox"/> Très bonne

Tableau 8-5 : Evaluation des visualisations.

8.3.6 Visualisation préférée

Après avoir fourni une appréciation pour chacune des visualisations, l'utilisateur devra indiquer quelle est la visualisation qu'il préfère.

<input type="checkbox"/> Brick Wall Chart	<input type="checkbox"/> Stacked Bar Chart	<input type="checkbox"/> Bar Chart	<input type="checkbox"/> Mountain	<input type="checkbox"/> Multi-Lines	<input type="checkbox"/> Dashboard
---	--	---------------------------------------	--------------------------------------	---	---------------------------------------

Tableau 8-6 : Visualisation préférée.

8.3.7 Commentaires généraux

Finalement, leurs commentaires sur l'application seront collectés, dans un souci de correction par rapport aux objectifs ou d'améliorations par rapport à des utilisations futures.

8.4 Résultats et analyses

8.4.1 Test de Mc Nemar

8.4.1.1 Méthode

Sur base des résultats obtenus via les tests d'évaluation de l'application, plusieurs tests ont été menés. Il s'agit des tests de Mc Nemar car les échantillons utilisés ne sont pas indépendants. Pour ce test, il ne faut pas commettre l'erreur d'appliquer les formules des échantillons indépendants. Il faut connaître, pour chaque individu, son appréciation pour les 2 visualisations, que nous pouvons résumer par un tableau de contingence 2x2 croisant les effectifs des deux variables.

Nous allons comparer les visualisations 2 à 2 de manière à tenter de découvrir si l'appréciation fournie par un utilisateur sur une visualisation est liée à celle qu'il a fournie pour une autre visualisation.

V1 /V2	-	+	Total
-	x_{11}	x_{12}	$x_{1.}$
+	x_{21}	x_{22}	$x_{2.}$
Total	$x_{.1}$	$x_{.2}$	

L'hypothèse de départ (H_0) est la suivante : $x_{12} = x_{21}$

Or, comme nous avons $x_{11} = x_{11} + x_{12}$ et $x_{21} = x_{11} + x_{21}$, cela revient à poser comme hypothèse de départ H_0 que $x_{12} = x_{21}$

En utilisant le test du Chi carré (χ^2), nous avons l'hypothèse nulle $x_{12} = x_{21}$ qui est estimée par $(x_{12} + x_{21})/2$ et la statistique de test (T) s'écrit de la manière suivante :

$$T = \frac{\left(x_{12} - \frac{x_{12} + x_{21}}{2}\right)^2 + \left(x_{21} - \frac{x_{12} + x_{21}}{2}\right)^2}{\frac{x_{12} + x_{21}}{2}} \quad (8.1)$$

Après simplification, l'équation 8.1 peut s'écrire :

$$\frac{(x_{12} - x_{21})^2}{x_{12} + x_{21}} \quad (8.2)$$

Nous pouvons alors comparer l'équation 8.2 à un χ_1^2 ou, sa racine carrée à une variable normale centrée réduite.

➔ Si la valeur trouvée pour T est supérieure à la valeur du χ_1^2 , c'est-à-dire si nous avons $T > Q_{\chi_1^2}(1 - \alpha)$, alors nous pouvons rejeter l'hypothèse de départ H_0

Initialement, nous avons un tableau 6x6 comportant les 6 visualisations ainsi que les 6 appréciations possibles.

	Très mauvaise	Mauvaise	Moyenne	Bonne	Très bonne	Total
Brick Wall Chart	0	1	1	10	11	23
Stacked Bar Chart	0	1	1	14	7	23
Bar chart	0	3	18	1	1	23
Mountain	0	1	4	17	1	23
Multi-Lines	0	1	2	8	12	23
Dashboard	0	6	11	3	3	23
Total	0	13	37	53	35	138

Cependant, si nous voulons observer les critères « négatif » et « positif » des visualisations, nous allons devoir regrouper plusieurs colonnes de façon à nous retrouver avec un tableau de taille 6x2.

Les colonnes à regrouper sont les colonnes « Très mauvaise », « Mauvaise » et « Moyenne » pour le critère « négatif » d'une part et « Bonne » et « Très bonne » pour le critère « positif » d'autre part.

Nous obtenons le tableau suivant de taille 6x2 :

	-	+	Total
BWC	2	21	23
SBC	2	21	23
BC	21	2	23
M	5	18	23
ML	3	20	23
DB	17	6	23
Total	50	88	138

Le but de ce test est de comparer les appréciations positives et négatives des visualisations et cela, deux à deux. Parmi les six visualisations, nous en avons quatre qui sont considérées comme « annuelles » (Brick Wall Chart, Stacked Bar Chart, Mountain et Multi-Lines) et deux comme « mensuelles » (Bar Chart et DASHBOARD).

Une démarche intéressante serait donc de comparer une visualisation dite « annuelle » par rapport à une visualisation dite « mensuelle ». Nous aurons alors les couples suivants :

BC / BWC (Tableau 1) BC / SBC (Tableau 2) BC / M (Tableau 3) BC / ML (Tableau 4)
 DB / BWC (Tableau 5) DB / SBC (Tableau 6) DB / M (Tableau 7) DB / ML (Tableau 8)

Tableau 1			
BC / BWC	-	+	Total
-	23	42	65
+	4	23	27
Total	27	65	92
$T = (42-4)^2 / (42+4) = 31,39$			

Tableau 2			
BC / SBC	-	+	Total
-	23	42	65
+	4	23	27
Total	27	65	92
$T = (42-4)^2 / (42+4) = 31,39$			

Tableau 3			
BC / M	-	+	Total
-	26	39	65
+	7	20	27
Total	33	59	92
$T = (39-7)^2 / (39+7) = 22,26$			

Tableau 4			
BC / ML	-	+	Total
-	24	41	65
+	5	22	27
Total	29	63	92
$T = (41-5)^2 / (41+5) = 28,17$			

Tableau 5			
DB / BWC	-	+	Total
-	19	38	57
+	8	27	35
Total	27	65	92
$T = (38-8)^2 / (38+8) = 19,57$			

Tableau 6			
DB / SBC	-	+	Total
-	19	38	57
+	8	27	35
Total	27	65	92
$T = (38-8)^2 / (38+8) = 19,57$			

Tableau 7			
DB / M	-	+	Total
-	22	35	57
+	11	24	35
Total	33	59	92
$T = (35-11)^2 / (35+11) = 12,52$			

Tableau 8			
DB / ML	-	+	Total
-	20	37	57
+	9	26	35
Total	29	63	92
$T = (37-9)^2 / (37+9) = 28,17$			

8.4.1.2 Conclusion sur le test de Mc Nemar

Lorsque nous consultons la table de Chi-carré (cf. Figure B.1), nous constatons que la valeur de $Q_{\chi^2}(1-\alpha)$ pour $ddl = 1$ et $\alpha = 0,05$ vaut 3,841.

A partir de là, si nous comparons chacun des T que nous avons obtenu avec cette valeur $Q_{\chi^2}(1-\alpha)$, nous constatons que toutes les valeurs de T sont supérieures. Cela signifie que nous devons rejeter toutes les hypothèses que nous avons posées au début du test. Il faut donc rejeter l'hypothèse d'égalité des proportions de changement. Nous pouvons conclure qu'il y a bien une différence d'appréciation pour les visualisations choisies. En d'autres termes, le fait que les utilisateurs aient une appréciation positive pour les visualisations dites « annuelles » implique qu'ils auront une appréciation fort différente en ce qui concerne les visualisations dites « mensuelles ».

8.4.2 Tests ANOVA : Analyse de la variance à deux facteurs

Les tests statistiques mesurent la probabilité que les différences mesurées entre les observations soient dues au seul fait du hasard (hypothèse nulle H_0).

L'analyse de la variance nous sert à mesurer la dissimilarité qu'il peut exister entre deux facteurs. Dans le cas présent, les deux facteurs seront les différentes visualisations et les utilisateurs. Grâce à l'analyse, nous pourrions dire si les résultats d'un test varient d'une visualisation à l'autre et, si ceux-ci changent d'une personne à l'autre.

L'analyse de la variance à deux critères de classification à pour but la comparaison des moyennes de nA populations, à partir d'échantillons aléatoires et indépendants prélevés dans chacune d'elles. Ces populations sont en général des variantes (ou niveaux n_a) d'un facteur contrôlé (ou facteur A) de variation.

- Conditions
 1. le paramètre étudié suit une distribution normale.
 2. les variances des populations sont toutes égales.
 3. les échantillons sont prélevés aléatoirement et indépendamment dans les populations.

Cette méthode permet donc de déterminer l'effet de deux facteurs et leur interaction. L'interaction, dans le sens de l'ANOVA, mesure l'influence que l'état d'un des facteurs a sur l'effet exercé par l'autre facteur sur la variable dépendante.

Les calculs suivant seront effectués comme s'il s'agissait d'un modèle fixe avec échantillons de plusieurs observations. Cependant, nous pourrions considérer que le modèle dans lequel nous nous trouvons serait un modèle mixte à échantillons de plusieurs observations. En effet :

- Dans le cas du modèle mixte, nous considérons une simple infinité ou quasi-infinité de populations. Nous avons un des critères de classification qui possède un nombre fini p de modalités et l'autre, qui possède un nombre infini, ou au moins un très grand nombre de modalités.
- D'une part, les moyennes théoriques, qui sont liées au critère de classification aléatoire, sont des variables aléatoires. D'autre part, les moyennes qui ne dépendent que du critère de classification fixe sont des constantes, tout comme la moyenne générale.

8.4.2.1 Méthode

L'hypothèse de départ choisie, soit H_0 , sera qu'il n'y a pas de différence significative entre les différentes observations, suivant un facteur ou l'autre c'est-à-dire que toutes les moyennes sont identiques. L'hypothèse H_1 suppose qu'au moins une des moyennes est différentes des autres.

Soit un tableau à double entrée suivant deux facteurs de la forme suivante :

Variable 1	Variable 2			
	i=1	2	3	...
j=1	X_{11}	X_{21}	X_{31}	
2	X_{12}	X_{22}	X_{32}	
3	X_{13}	X_{23}	X_{33}	
4	X_{14}	X_{24}	X_{34}	
5	X_{15}	X_{25}	X_{35}	
...

Afin de calculer la somme des carrés (noté SC), il faut calculer le carré des différences entre la moyenne des éléments d'un facteur et la moyenne globale, ce qui est donné par les formules 8.3 et 8.4.

$$\forall j : (\bar{X}_{ij} - \bar{\bar{X}})^2 \quad (8.3)$$

$$\forall i : (\bar{X}_{ij} - \bar{\bar{X}})^2 \quad (8.4)$$

A partir de là, il est possible de calculer la valeur prévue pour tout les X_{ij} en fonction des deux facteurs par la formule 8.5.

$$\hat{X}_{ij} = \bar{\bar{X}} + (\bar{X}_i - \bar{\bar{X}}) + (\bar{X}_j - \bar{\bar{X}}) \quad (8.5)$$

- $\bar{\bar{X}}$ représente la moyenne globale,
- $(\bar{X}_i - \bar{\bar{X}})$ représente l'ajustement par rapport au premier facteur
- $(\bar{X}_j - \bar{\bar{X}})$ représente l'ajustement par rapport au premier facteur

Ainsi, un tableau de la même taille que le tableau initial est obtenu et contient les valeurs prévues pour chaque élément.

A partir de ce point, les valeurs résiduelles sont données par la différence entre les valeurs observées X_{ij} et les valeurs prévues (i.e. valeurs ajustées) comme il est indiqué à la formule 8.6.

$$R\acute{e}sidual_{ij} = X_{ij} - \hat{X}_{ij} \quad (8.6)$$

Dès lors, les différents SC peuvent être trouvés de la façon suivante :

- Premier facteur : différence entre les a moyennes des éléments du premier facteur \bar{X}_i

$$SC_A = b \times \sum_{i=1}^a (\bar{X}_i - \bar{\bar{X}})^2 \quad (8.7)$$

- Second facteur : différence entre les b moyennes des éléments du premier facteur \bar{X}_j

$$SC_B = a \times \sum_{j=1}^b (\bar{X}_j - \bar{\bar{X}})^2 \quad (8.8)$$

- SC résiduelle : différence entre les observations réelles X_{ij} et les valeurs ajustées

$$\hat{X}_{ij} = \bar{\bar{X}} + (\bar{X}_i - \bar{\bar{X}}) + (\bar{X}_j - \bar{\bar{X}})$$

$$SC_E = \sum_{i=1}^a \sum_{j=1}^b (X_{ij} - \bar{X}_i - \bar{X}_j + \bar{\bar{X}})^2 \quad (8.9)$$

Les degrés de liberté pour les deux facteurs sont respectivement $a-1$ et $b-1$, le degré de liberté résiduel sera donc de $(a-1) \times (b-1)$.

Ensuite, le carré des moyens (CM) d'un facteur est obtenu en divisant le SC par le degré de liberté correspondant au facteur. CM est en réalité la variance expliquée d'un facteur alors que le CM total est la variance inexpliquée.

C'est le rapport entre la variance expliquée et la variance inexpliquée qui est intéressant pour calculer la valeur de Fisher (notée F) et donc donné par

$$F_A = \frac{\text{Variance expliquée pour le facteur A}}{\text{Variance inexpliquée}} = \frac{CM_A}{CM_{R\acute{e}sidual}} \quad (8.10)$$

Il faut maintenant comparer F trouvé en 8.10 avec le quantile de Fisher (voir Figure B.2).

Par rapport à un degré d'erreur α (souvent 5% ou 1%), le degré de liberté par rapport à un facteur et le degré de liberté total, la table de Fisher permet de trouver le nombre théorique de Fisher, soit F_α .

Si le F trouvé par 8.10 est supérieur au F théorique, soit F_{α} , cela implique que l'on a réussi à mettre en évidence la présence d'au moins une moyenne différente des autres. Alors il existe une différence significative entre les différents éléments correspondant au facteur. Dans le cas contraire, il est impossible d'affirmer une telle différence.

Dans les cas des tests effectués dans ce travail, le premier facteur sera les visualisations, à savoir, si les visualisations possèdent des différences significatives entre elles. Le deuxième facteur sera les utilisateurs car il est possible que les utilisateurs aient des comportements différents les uns des autres.

Les visualisations sont au nombre de 6, le degré de liberté pour ce facteur sera donc de 5 (soit 6-1) ; l'échantillon d'utilisateur étant de 23, le degré de liberté de ce deuxième facteur sera de 22 (soit 23-1). Le degré de liberté total sera ainsi de $2*22 = 44$. Les tables de Fisher devront donc être consultées respectivement pour les degrés de liberté (2,44) et (22,44). Le tableau 8.7 montre les différentes valeurs du quantile de Fisher pour $\alpha = 0,01$ ou $\alpha = 0,05$.

Fisher	$\alpha = 0,01$	$\alpha = 0,05$
ddl 2,44	4,31	3,21
ddl 22,44	1,79	1,85

Tableau 8-7 : Valeurs de Fisher.

A partir de là, il reste à calculer les différentes valeurs de F pour les comparer avec celles du tableau.

8.4.2.2 Conclusion sur le test ANOVA

Bien que les conditions d'application des tests paramétriques ne soient pas toutes réunies ici (distribution normale des populations, variances des populations identiques et connues, observations indépendantes), une analyse de la variance (ANOVA) a été effectuée. En effet :

1. «La validité de l'analyse de la variance a été démontrée en supposant les populations normales et de même variance, et les échantillons aléatoires, simples et indépendants.»;
2. «En ce qui concerne l'hypothèse de normalité [...], il suffit en pratique d'éviter d'employer l'analyse de la variance lorsque les distributions des populations-parents sont très différentes des distributions normales (distribution en i ou en j par exemple) [...]. Dire que l'analyse de la variance est, dans les limites qui viennent d'être esquissées, peu sensible à la non-normalité des populations-parents, c'est dire que, dans ces limites, des risques réels de première espèce de l'ordre de 4 à 7% correspondent généralement à un risque théorique de 5% (niveau de signification 0,05), et des risques réels de l'ordre de 0,5 à 2% à un risque théorique de 1% (niveau 0,01).»;
3. «De même, l'hypothèse d'égalité des variances ou hypothèse d'homoscédasticité est d'importance relativement secondaire lorsque les effectifs des échantillons sont tous égaux. Dans ces conditions, l'analyse de la variance est, comme le test t, une méthode robuste, tant en ce qui concerne la normalité qu'au point de vue de l'égalité des variances. Il n'en est cependant pas ainsi quand les effectifs des échantillons sont variables. Le risque de première espèce peut alors être influencé considérablement par une inégalité de variance.» (Dagnelie, 1975). Ce risque de première espèce ou risque a consiste en un rejet de l'hypothèse nulle H_0 (différences entre les observations dues au hasard) alors que celle-ci est vraie.

Le protocole expérimental a été élaboré afin de pouvoir utiliser l'analyse de la variance qui est l'un des tests statistiques les plus puissants qui existent. Dans certaines situations, il est en effet possible que la troisième condition énoncée par Dagnelie (homoscédasticité) soit violée.

L'analyse de la variance utilisée ici est un modèle croisé d'ANOVA à deux critères de classification et à échantillons d'une seule observation.

Les tests effectués l'ont été de manière chronométrée. Dans ce cas ci, le temps sera le facteur déterminant, le premier facteur. (Les tableaux complets des calculs se trouvent en annexes)

Si nous nous reportons aux tableaux de calculs du premier et du second test de l'ANOVA à deux facteurs concernant l'efficacité des visualisations, nous pouvons constater que pour le facteur des utilisateurs, qu'il s'agisse d'un niveau de 1% ou de 5%, les hypothèses d'uniformité des facteurs sont rejetées. Par contre, pour le facteur du type de visualisation, qu'il s'agisse d'un niveau de 1% ou de 5%, les hypothèses d'uniformité des facteurs ne sont pas rejetées.

Nous pouvons donc conclure à une différence dans les chiffres du point de vue de temps d'utilisation, que ce soit une différence envers les visualisations ou envers les utilisateurs.

Premier test de l'ANOVA						
Source	SC	ddl	CM	F	p.c. 001	p.c. 005
Visualisations	855,107851	2	427,6	6,393	Reject	Reject
Utilisateurs	2087,31514	22	94,88	1,419	Accept	Accept
Résiduelle	2942,42299	44	66,87			

Dexuième test de l'ANOVA						
Source	SC	ddl	CM	F	p.c. 001	p.c. 005
Visualisations	905,881493	2	452,94	8,4774	Reject	Reject
Utilisateurs	1445,01012	22	65,682	1,2293	Accept	Accept
Résiduelle	2350,89162	44	53,429			

Nous ne pouvons pas affirmer qu'une visualisation manifeste un avantage de performance par rapport aux autres et cela, au niveau du temps nécessaire à la réalisation d'une tâche, de la précision ou de l'efficacité. Les mêmes résultats quantitatifs peuvent être obtenus à partir de n'importe quelle visualisation.

Toutefois, il est important de signaler que ces conclusions doivent être prises avec précaution car la distribution utilisée n'est pas formellement une normale

8.4.3 Test de la médiane

8.4.3.1 Méthode

Le test de la médiane est utilisé pour vérifier l'égalité des médianes provenant de deux populations (voire plus). Il constitue une alternative non paramétrique aux tests d'analyse de la variance.

Nous avons deux populations de n_1 et n_2 éléments. Si nous combinons ces 2 échantillons et déterminons la médiane de cet échantillon combiné, nous aurons encore une estimation appropriée de la population médiane sous l'hypothèse H_0 : les populations ont la même médiane, c'est-à-dire que les deux groupes devraient être séparés dans les mêmes proportions par cette valeur.

Nous savons que pour comparer des proportions, nous pouvons utiliser le test χ_1^2 ou le test exact de Fisher.

Nous allons travailler sur un tableau de contingence 2x2 où les éléments de ce tableau seront, pour chaque échantillon, le nombre d'éléments supérieurs et inférieurs à la Médiane (M).

Nous obtenons un tableau tel que

	Pop 1	Pop 2
Au dessus de M	a_1	a_2
En dessous de M	b_1	b_2

Il peut y avoir des observations juste à la médiane, cela est dû à la méthode de mesure utilisée lors des tests. Il a donc été décidé d'utiliser \leq versus $>$ plutôt que $<$ versus \geq . Toutefois, cela n'a aucune influence sur les résultats.

La valeur T du test s'obtient par l'équation :

$$T = \frac{(2a_1 - n_1)^2 \times (n_1 + n_2)}{n_1 \times n_2} \text{ et } n_i = a_i + b_i \quad (8.11)$$

Nous pouvons alors comparer l'équation 8.11 à un χ_1^2 ou, sa racine carrée à une variable normale centrée réduite.

→ Si la valeur trouvée pour T est supérieure à la valeur du χ_1^2 , c'est-à-dire si nous avons $T > Q_{\chi_1^2}(1 - \alpha)$, alors nous pouvons rejeter l'hypothèse de départ H_0

Dans la table du χ_1^2 , la valeur de $Q_{\chi_1^2}(1 - \alpha)$ pour $ddl = 1$ et $\alpha = 0,05$ vaut 3,841.

BWC / SBC

	BWC	SBC
Au dessus de M	13	6
En dessous de M	10	17

La médiane vaut 10. Nous avons $T = 8,69$, ce qui est supérieur à $Q_{\chi_1^2}(1 - \alpha)$. Il faut donc rejeter les hypothèses en ce qui concerne les médianes de ces 2 populations.

BWC / ML

	BWC	ML
Au dessus de M	11	9
En dessous de M	12	14

La médiane vaut 11. Nous avons $T = 3,13$, ce qui est inférieur à $Q_{\chi_1^2}(1 - \alpha)$. Nous pouvons donc accepter les hypothèses en ce qui concerne les médianes de ces 2 populations.

SBC / ML

	SBC	ML
Au dessus de M	10	13
En dessous de M	13	10

La médiane vaut 9,5. Nous avons $T = 1,39$, ce qui est supérieur $Q_{\chi_1^2}(1 - \alpha)$. Nous pouvons donc accepter les hypothèses en ce qui concerne les médianes de ces 2 populations.

8.4.3.2 Conclusion sur le test de la médiane

Le test de la médiane a été utilisé car il est basé sur l'idée que, lorsque deux groupes sont superposables, leurs médianes ont la même valeur qui peut être estimée par la médiane globale des deux échantillons réunis.

Contrairement au premier cas (BWC/SBC) où nous avons du rejeter l'hypothèse de départ, dans les deux autres cas (BWC/ML et SBC/ML), elle a été acceptée car la valeur de T était inférieure à celle de $Q_{\chi^2}(1-\alpha)$. Cela nous permet de dire que ces populations vérifient l'égalité en ce qui concerne leur médiane.

8.5 Remarques et améliorations

A la fin du questionnaire, les utilisateurs étaient invités à formuler des remarques générales sur l'application, aussi bien d'un point de vue positif que négatif. Ces remarques portaient sur leurs préférences, les défauts/qualités des visualisations, de l'application ou de l'outil utilisé.

Après une compilation de toutes les remarques émises, nous avons pu les rassembler sous plusieurs catégories.

8.5.1 Remarques négatives

1. Les couleurs.
 - La nuance de couleurs est difficilement perceptible sur certaines visualisations. Il n'est pas facile de déterminer si une valeur est légèrement inférieure ou supérieure à une autre uniquement avec la couleur.
2. Le stylo et l'écran.
 - Le stylo est très difficile à utiliser correctement pour une personne n'ayant pas d'expérience avec un PDA.
 - L'utilisateur ne sait pas toujours s'il a correctement appuyé sur l'écran.
 - L'écran nécessite une bonne vue de la part des utilisateurs. Tout est beaucoup plus petit, parfois trop ?
 - La taille de l'écran nécessite une trop grande précision pointant les objets avec le stylo.
3. Les visualisations
 - La représentation des jours est trop compacte sur certaines visualisations (Stacked Bar Chart, Mountain)
 - Le tableau (« Dashboard ») contient trop d'informations et est moins pratique que les autres visualisations car les différences de valeur ne se remarquent pas immédiatement.
4. Autres
 - Il faudrait pouvoir configurer la langue de l'application en français ou en anglais au début (et pas seulement les textes d'aide).
 - Les boutons devraient être de couleurs différentes pour mieux renseigner sur leur but. (Ex : « Vert = save » et « Rouge = cancel »)
 - L'ordre dans lequel il faut effectuer les opérations pourrait être plus explicite. (Profil, configuration, création et interactions).

8.5.2 Remarques positives

1. Les visualisations

- Les visualisations sont efficaces car elles offrent une vue globale tout en permettant d'en obtenir une plus détaillée.
 - Les lignes de segmentation d'une semaine à l'autre permettent de se retrouver plus facilement lorsqu'il faut chercher un jour précis.
2. Les interfaces
 - Les interfaces sont conviviales et agréables à utiliser.
 - Les menus sont clairs.
 - La lisibilité et la navigation sont assez intuitives.
 3. Autres
 - Les textes d'aide en français et en anglais sont une bonne idée.
 - L'idée du programme est intéressante. C'est un bon outil d'analyse clair et pratique.

8.6 Conclusion

Grâce aux tests réalisés (Mc Nemar, ANOVA et de la médiane), nous pouvons tirer quelques conclusions à propos des préférences des utilisateurs quant aux visualisations et leur efficacité.

Il est évident que les utilisateurs préfèrent des visualisations telles que Brick Wall Chart, Stacked Bar Chart ou Multi-Lines. Et cela, pour des raisons de facilité de manipulation. En effet, malgré les inconvénients rencontrés avec ces visualisations (nuance de couleurs difficilement perceptibles pour la première, jours trop compacts pour les deux autres), ce sont ces trois là qui permettent à l'utilisateur de retrouver, le plus rapidement possible, l'information recherchée.

Aussi bien Brick Wall Chart que Stacked Bar Chart permettent de retrouver de façon rapide et précise un jour en particulier, notamment grâce à leur forme et leur structure, tandis que Multi-Line permet de juger, en un bref coup d'œil, de l'évolution d'une série.

Ce qu'il est intéressant de remarquer, c'est que Brick Wall Chart et Multi-Lines ont été jugées « Très bonne » par respectivement 11 et 12 personnes mais c'est Brick Wall Chart qui est la visualisation préférée des utilisateurs avec 10 votes contre 7 pour Multi-Lines. Nous pouvons en déduire la constatation suivante : dans plus de la moitié des cas, lorsque l'appréciation de ces 2 visualisations est identique, Brick Wall Chart constitue la visualisation de prédilection.

Quant aux autres visualisations non plébiscitées par les utilisateurs, à savoir Bar Chart, Mountain et Dashboard, il faut reconnaître qu'elles n'offrent pas les mêmes possibilités que les précédentes. Dashboard étant même considérée comme la plus « mauvaise » des visualisations suivie par Bar Chart. Toutefois, parmi les rares utilisateurs qui ont qualifié Dashboard de « bonne », voire « très bonne », on retrouve surtout (à une exception près) des personnes plus âgées. Celles-ci ont plus l'habitude de travailler avec des tableaux et s'y retrouvent certainement mieux.

Les visualisations offrent deux possibilités de périodes pour l'analyse. Elles peuvent être soit « annuelles » soit « mensuelles ». Il était intéressant de vérifier si un lien existait entre ces deux critères de temps. Au vu des appréciations des utilisateurs, nous pouvons dire que ceux qui choisissent ou cotent favorablement les visualisations dites « annuelles » sont moins attirés par les visualisations dites « mensuelles ».

En résumé, c'est bien le confort d'utilisation et le caractère intuitif des visualisations qui ont fait que certaines visualisations ont été appréciées et d'autres moins.

9 Conclusion

Tout au long de ce mémoire, l'objectif était de poser les bases adéquates pour la création d'un outil générique permettant la visualisation de séries temporelles sur PDA. Pour cela, nous avons tiré profit de précédents travaux réalisés dans ce domaine. En effet, le domaine d'application de la visualisation sur les appareils mobiles n'est pas celui pour lequel ont été développées le plus grand nombre d'applications. Le problème majeur rencontré dans ce domaine est issu de la taille de l'ensemble des données qu'il faut représenter. Les applications, utilisant des techniques de visualisation d'informations, ne permettent pas de représenter de façon claire et lisible ces ensembles de données tel qu'il est possible d'en collecter pendant une période de temps plus ou moins longue.

L'analyse de l'état de l'art nous a permis de définir les notions d'information, de visualisation et de visualisation de l'information. Elle nous a également permis d'avoir un bref aperçu des applications déjà existantes et de nous rendre compte qu'il y avait beaucoup de manières d'aborder le problème et de tirer profit des technologies disponibles. Hélas, nous avons également vu que ces technologies sont demeurées relativement difficiles à instaurer pour un large usage de diffusion.

Les dispositifs mobiles, tels que le PDA nécessitant l'usage d'un stylet, souffrent toujours de certaines limitations, particulièrement lorsqu'il s'agit de l'affichage d'informations et de l'interaction avec l'utilisateur.

Pour pallier à ces limitations techniques inévitables, des visualisations ont été développées (conçues d'une nouvelle manière pour certaines). Tout en essayant de respecter au maximum les principes et règles ergonomiques élémentaires, ces visualisations tentent d'utiliser des analogies (calendrier, montagne, ...) avec le monde réel en se basant sur la forme et les couleurs. D'autres moyens ont été trouvés pour contourner les limitations imposées par ces appareils mobiles.

Une fois les visualisations élaborées, nous étions prêts à donner le contour du genre d'application de visualisation que nous avons voulu mettre en application pour PDA. Notre approche a consisté en considérant quelques scénarii. Ces scénarii ont permis de préciser les tâches principales que nous devions voir apparaître dans l'application.

A ce stade, il ne restait plus qu'à déterminer les différentes possibilités qui s'offraient à nous quant à l'environnement de développement. Du choix de cet environnement, nous avons examiné les langages de programmation existants et disponibles pour l'exécution d'applications mobiles.

Nous avons finalement opté pour l'environnement de développement « Microsoft .NET » car il offre, non seulement, un environnement visuel efficace et agréable (Visual Studio 2003) mais également un émulateur intégré de PC de poche, tandis que Java (avec son J2ME) ne semblait pas aussi « intuitif » pour concevoir des applications mobiles.

Bien que tout à fait semblable du point de vue des techniques de codage, les deux principaux langages disponibles, que sont Java et C#, diffèrent par leur rentabilité et leur distribution. Après le choix de l'environnement de développement, c'est finalement le C# (la figure de proue de « Microsoft .NET ») qui s'est avéré le plus adéquat.

Nous avons alors examiné les diverses possibilités et contraintes, qui nous ont menés à raffiner la définition et les dispositifs de notre application. Puisque le temps était limité, nous avons dû placer plusieurs limitations à notre projet initial. Nous avons exposé les étapes principales de la phase de conception, établi les principes et règles ergonomiques qu'il était bon de respecter lors du développement d'une application et ensuite, défini la méthode de stockage de données et les composants logiques. Nous sommes alors devenus plus pratiques, et avons donné une vue d'ensemble de la hiérarchie de classe de l'architecture prévue.

Comme cela pouvait être prévu, le processus d'implémentation a apporté son lot de problèmes. C'est pourquoi nous avons également discuté de certaines difficultés auxquelles nous avons dû faire face pendant la phase de codage. En conclusion, nous avons présenté quelques limitations inhérentes au langage d'exécution dont nous nous sommes servi.

Les annexes ont permis de dévoiler complètement l'application en présentant toutes les interfaces construites.

Nous concluons par le fait que ce travail ne représente qu'un commencement. A ce stade, des améliorations doivent encore être apportées afin de rendre cette application plus intéressante notamment par l'apport du son.

10 Bibliographie

- [ARAGON04] « Introduction aux Séries Temporelles », <http://www.vcharite.univ-mrs.fr/PP/lubrano/cours/ch1.pdf> (Dernière visite : 10/08/2005)
- [BARTHET88] Marie-France Barthelet , « Logiciels interactifs et ergonomie », <http://membres.lycos.fr/interaction/Ouvrages/Barthelet1988/barthelet1988.pdf> (Dernière visite : 10/08/2005)
- [BREWSTER94] S. A. Brewster, « Providing a structured method for integrating non-speech audio into human-computer interfaces », Thèse de doctorat, University of York, UK, 1994.
- [BREWSTER99] S.A. Brewster, and M.G. Crease, « Correcting Menu Usability Problems With Sound », Accepté à la publication dans *Behaviour and Information Technology 17*, 1999.
- [BREWSTER00] S.A. Brewster, A. Crossan, and M.G. Crease, « Automatic volume control for auditory interfaces », dans *Volume II Proceedings of BCS HCI 2000*, Sunderland, UK, 2000, pp 17-18.
- [CHEVILLON04] Guillaume Chevillon, « Pratique des séries temporelles », <http://guillaume.chevillon.free.fr/lecturenotes.pdf> (Dernière visite : 10/08/2005)
- [CCMa] Comment Ca Marche, « Synthèse additive et soustractive », <http://www.commentcamarche.net/video/lumiere.php3> (Dernière visite : 10/08/2005)
- [CCMb] Comment Ca Marche, « Le codage de la couleur », <http://www.commentcamarche.net/video/couleur.php3> (Dernière visite : 10/08/2005)
- [CCMc] Comment Ca Marche, « Introduction aux PDA », <http://www.commentcamarche.net/pc/pda.php3> (Dernière visite : 10/08/2005)
- [CCMd] Comment Ca Marche, « Introduction à .NET », <http://www.commentcamarche.net/dotnet/dotnet-intro.php3> (Dernière visite : 10/08/2005)
- [DEMAINE96] Jeff Demaine, « Visualisation de l'information », <http://www.collectionscanada.ca/9/1/p1-234-f.html> (Dernière visite : 10/08/2005)
- [DEM&SCHO01] C. Demoulin, O. Schöller, « Sonification of Time-dependant Data », Mémoire en informatique, FUNDP, Namur, 2001.
- [DUFOUR98] Jean-Marie Dufour, « Techniques de séries chronologiques : introduction », <http://www.fas.umontreal.ca/SCECO/Dufour> (Dernière visite : 13/08/2005)
- [ITTEN86] J. Itten, « L'art de la couleur », Ed. Dessain et Tolra, Ravensburg, 1986

- [KADDOUR99] C. Kaddour, « Compression des images fixes par fractales basée sur la triangulation de Delannay et la quantification vectorielle », Mémoire en informatique, Université des Sciences et de la technologie Hourai Boumedienne, Algérie, 1999,
<http://www.google.fr/url?sa=t&ct=res&cd=1&url=http%3A//www.kaddour.com/&ei=ZtP9QofcOIn4sgGE8MXXDg> (Dernière visite : 13/08/2005)
- [LAUER01] Christophe Lauer, « Introduction a Microsoft .NET », http://www.dotnet-fr.org/intro_dotnet_tmr.php3 (Dernière visite : 10/08/2005)
- [MOUCHART78] M. Mouchart, L. Simar, « Recyclage en Statistique : Méthodes non paramétriques », Comité de Statistique, Université Catholique de Louvain, 1978
- [NOIRHOMME00] M. Noirhomme-Fraiture, « Le son dans les interfaces IHM : Application à la représentation de données multivariées complexes », dans *2èmes Journées Multimédia*, Namur, Septembre 2000.
- [NOIRHOMME04] Monique Noirhomme, Anne de Baest, « Check-list d'identification du contexte d'utilisation », Cours d'informatique IHM, FUNDP Namur, 2004
- [POULINGEAS04] Patrick Poulingeas, « La visualisation de l'information », <http://www.msi.unilim.fr/~poulingeas/Visualisation.htm> (Dernière visite : 10/08/2005)
- [RANDOLET04] Frédéric Randolet, « Visualisation de séries temporelles sur PDA », Mémoire en informatique, FUNDP Namur, 2004.
- [SCHOFFER98] Nicolas Schöffler, « Définition de l'information », <http://www.olats.org/schoffer/definfo.htm> (Dernière visite : 10/08/2005)
- [STACKO00] John Stasko, Richard Catrambone, Mark Guzdial, and Kevin McDonald, « An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures », *International Journal of Human-Computer Studies*, Vol. 53, No. 5, November 2000, pp. 663-694.
- [SUN03] Sun Microsystems, « J2ME - Java 2 Micro Edition », http://www.3ie.org/nouvelles_technologies/fiche/fiche_J2ME.htm (Dernière visite : 10/08/2005)
- [THOMAS05] Rémi Thomas, « Débuter avec Microsoft Visual Studio .NET », <http://www.pixel-technology.com/rthomas/> (Dernière visite : 10/08/2005)

Autres :

- <http://www.icpsr.umich.edu/help/glossary.html> (Dernière visite : 10/08/2005)
<http://www.cogsci.princeton.edu/cgi-bin/webwn> (Dernière visite : 10/08/2005)
<http://cdss.state.co.us/glossary.asp> (Dernière visite : 10/08/2005)
<http://www.bridgefieldgroup.com/glos9.htm> (Dernière visite : 10/08/2005)

Annexes A : Manuel d'utilisation

A.1 Liste des fichiers

A.1.1 Fichiers à déployer

Pour utiliser et installer l'application « TSviz_2K5 » sur le PDA, il faut commencer par déployer le projet « TSviz_2K5 » en utilisant Visual Studio .NET 2003 (ou toute autre version supérieure). Le dossier du projet contient les fichiers suivants :

- Les sous-répertoires « bin » et « obj »
- Les fichiers sources « .cs » qui sont les fichiers C# de l'application
- Les fichiers sources « .resx » qui sont les fichiers liés aux fichiers C# utilisés pour les interfaces.
- Les fichiers icônes « .ico »
- Les fichiers TSviz_2K5 (C# Device Project File) et TSviz_2K5 (Microsoft Visual Studio Solution Object)
- Le fichier « TSviz_2K5.csdproj »

Si l'un de ces fichiers venait à manquer lors du déploiement, l'application ne pourrait fonctionner correctement.

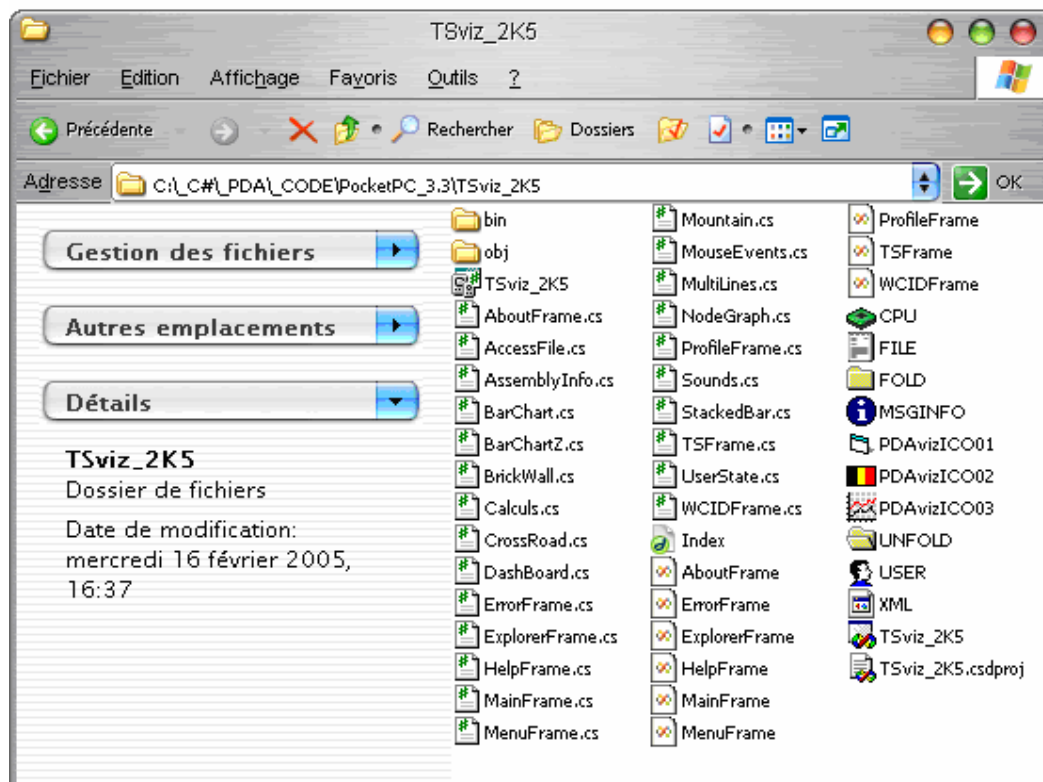


Figure A-1 : Fichiers sources de TSviz_2K5.

A.1.2 Fichiers à transférer

Pour utiliser pleinement l'application, il ne faut pas seulement déployer les fichiers sources via l'environnement de développement. Il faut également transférer sur le PDA les dossiers suivants et cela, avant le démarrage de l'application :

- HelpText
- Pictures
- Police, Stock Market, Weather
- Profile
- Sounds

Ces dossiers comprennent les fichiers nécessaires à l'exécution de l'application (HelpText, Pictures et Sounds) ainsi que ceux contenant les données relatives aux utilisateurs (Profile)

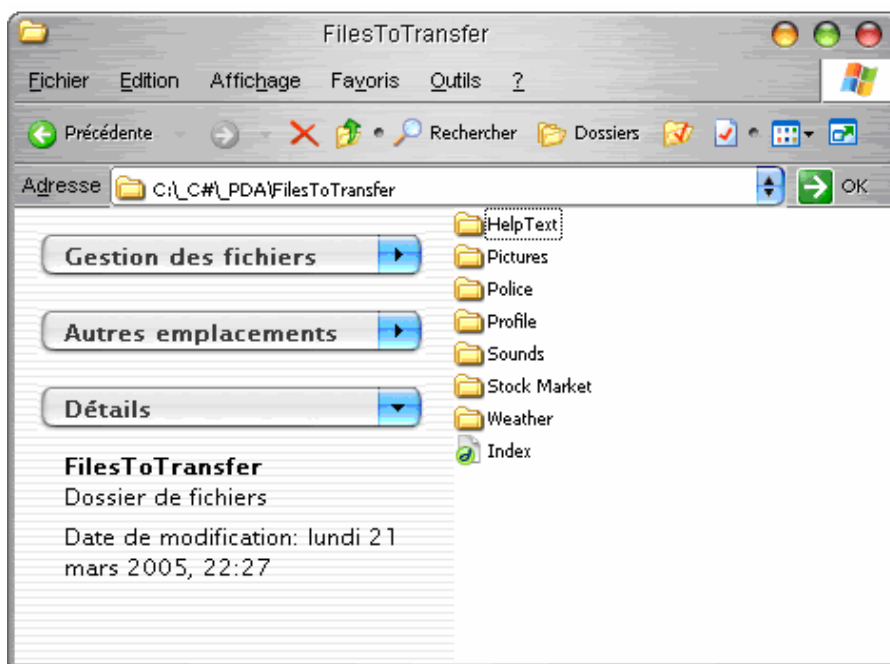


Figure A-2 : Fichiers à transférer sur le Pocket PC.

Les fichiers se trouvant dans les dossiers « Police, Stock Market et Weather » ne sont pas à proprement parler des fichiers de l'application. Ce sont les dossiers contenant les données des séries temporelles analysées par les utilisateurs.

Il est bien entendu certain que ces fichiers vont varier selon les utilisateurs, leur domaine d'application ou leurs besoins.

Une fois tous les fichiers nécessaires déployés et transférés sur le PDA, voilà à quoi devrait ressembler le dossier « PDAviz » :

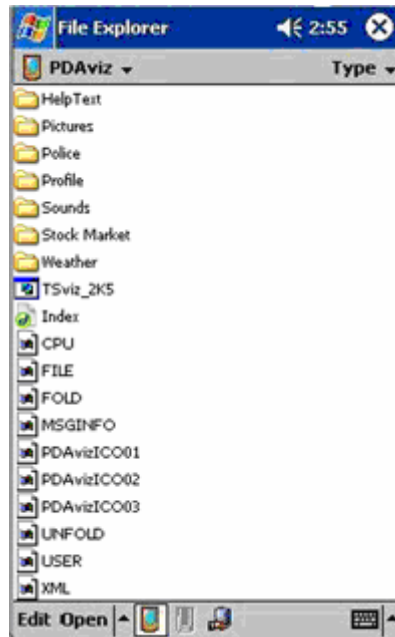


Figure A-3 : Fichiers présents sur le Pocket PC après transfert.

A.2 Démarrage de l'application

A.2.1 Ouverture de l'application

Lorsque l'utilisateur démarre l'application « TSviz_2K5 », il tombe sur l'écran principal de l'application qui lui permet d'entrer ou de sortir du programme. Il peut éventuellement appuyer avec le stylo sur l'image pour en obtenir un agrandissement partiel.

Lorsqu'il souhaite quitter l'application, il appuie sur le bouton correspondant et une boîte de dialogue lui demande une confirmation. S'il choisit de ne pas quitter le programme, il reviendra à l'écran de démarrage.



Figure A-4 : Ouverture de l'application.

Pour démarrer l'application et ainsi arriver dans le menu principal, l'utilisateur doit appuyer sur le bouton « Enter ». En cliquant sur le bouton « Quit », l'utilisateur recevra un message de confirmation.

Durant tout le déroulement de l'application, lorsque l'utilisateur devra introduire une donnée de configuration ou prendre une décision « irréversible » (par exemple la suppression d'une donnée), une boîte de dialogue apparaîtra afin de lui demander confirmation.

A.2.2 Ouverture du menu principal

Lorsque l'utilisateur décide d'entrer dans l'application, il verra apparaître le menu principal qui lui permettra de créer un nouveau profil, de sélectionner un profil dans une liste de profils existants, d'effacer un profil, de naviguer dans l'explorateur de fichiers, de lancer la partie de visualisation ou encore de quitter le menu (i.e. retourner à l'écran de démarrage).

Au cas où l'utilisateur voudrait obtenir quelques informations sur les profils utilisateurs déjà existants, les séries temporelles disponibles ou encore les visualisations, il peut lancer l'explorateur de fichiers (« Files Explorer ») qui lui permettra de naviguer parmi ces fichiers et d'obtenir de brèves descriptions pour chacun d'entre-eux.

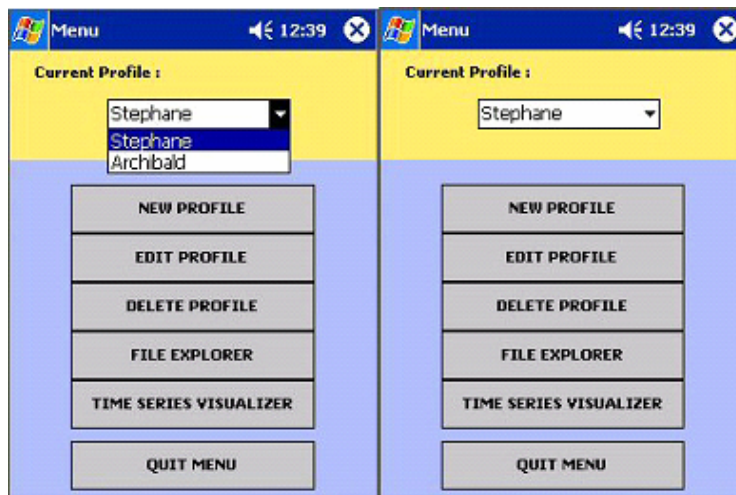


Figure A-5 : Ouverture du menu principal.

Dans la partie jaune de l'écran, l'utilisateur peut trouver la liste des profils actuellement enregistrés dans l'application (cela au moyen des fichiers XML transférés sur le PDA). Il doit alors sélectionner un profil (ou s'en créer un nouveau) pour procéder à l'étape suivante qui est de lancer la partie visualisation.

Si toutefois l'utilisateur appuie sur le bouton « Time Series Visualizer » et qu'aucun profil n'a été sélectionné au préalable, l'application utilisera, par défaut, le 1er profil se trouvant dans la liste et enverra un message d'erreur si la liste venait à être vide.

Dès que l'utilisateur effectue une opération pouvant impliquer une modification de la liste des profils, celle-ci est automatiquement rechargée afin de toujours présenter à l'utilisateur une liste mise à jour.

A.3 Manipulation des profils

Tout utilisateur qui veut se servir de l'application doit obligatoirement posséder un profil enregistré dans l'application. Sans la réalisation de cette étape, il ne sera pas possible de procéder à l'étape qui permet la consultation et l'analyse des séries temporelles.

A.3.1 Création d'un nouveau profil

Cette opération peut se réaliser dans 2 situations :

- La première situation possible pour se créer un profil, c'est lorsque l'utilisateur manipule l'application pour la toute première fois. Il faut donc qu'il enregistre ses préférences avant de l'utiliser.
- La seconde possibilité survient lorsqu'un utilisateur, possédant déjà un profil, souhaite modifier ses préférences sans pour autant effacer les anciennes. Il peut alors se créer un nouveau profil et ainsi conserver ses données précédentes.

En appuyant sur « New Profile » dans le menu principal, la fenêtre de création d'un profil va s'ouvrir. Il faudra alors introduire différentes informations relatives aux préférences de l'utilisateur, ses informations sont :

- Le nom de l'utilisateur (ou éventuellement un surnom), c'est ce nom qui apparaîtra dans la liste de tous les profils.
- La main qui va tenir le stylo.
- La langue pour le texte d'aide.
- La visualisation par défaut.

L'utilisateur peut ensuite sauvegarder son profil (« Save »), effacer sa sélection (« Reset ») ou revenir en arrière (« Back »).

Dans l'éventualité où le nom que l'utilisateur aurait introduit existerait déjà dans la liste des profils existants, il sera proposé à l'utilisateur de supprimer le profil existant et de le remplacer par le nouveau profil, soit de changer le nom d'utilisateur dans le profil récemment créé.

Figure A-6 : Création d'un profil.

Dans l'hypothèse où l'utilisateur ne saurait pas quelle série temporelle ou quelle visualisation par défaut il peut choisir, il lui est possible d'ouvrir l'explorateur de fichiers et de consulter les

brèves informations relatives à ces paramètres. Pour cela, il lui suffit d'appuyer sur le bouton « FE ».

A.3.2 Modification d'un profil existant



En sélectionnant un profil dans la liste et en appuyant sur « Edit Profile », l'utilisateur peut modifier les paramètres existants de son profil et ensuite le sauvegarder.

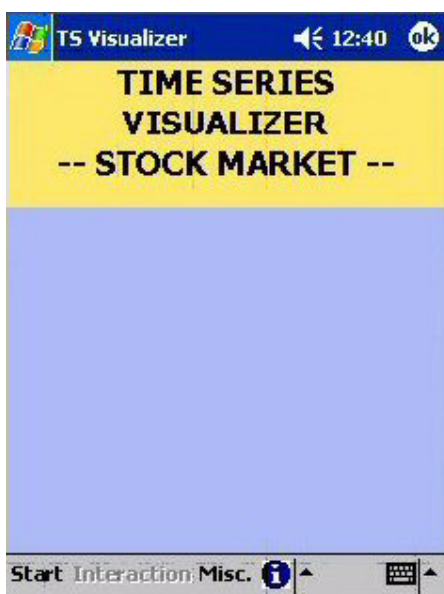
Dans l'hypothèse où l'utilisateur ne saurait pas quelle série temporelle ou quelle visualisation par défaut il peut choisir, il lui est possible d'ouvrir l'explorateur de fichiers et de consulter les brèves informations relatives à ces paramètres.

Pour cela, il lui suffit d'appuyer sur le bouton « FE », il ouvrira la fenêtre de l'explorateur de fichiers et il pourra ainsi consulter les textes explicatifs en rapport avec chaque série temporelle et chaque visualisation. À partir de ce moment là, il pourra faire son choix pour ce qui est de la série temporelle et de la visualisation par défaut.

A.3.4 Suppression d'un profil existant

En sélectionnant un profil dans la liste et en appuyant sur « Delete Profile », l'utilisateur peut effacer un profil existant. Lorsqu'il ne restera plus de profil dans l'application, il sera alors impossible de passer à l'étape de visualisation. Pour ce faire, il faudra repasser par l'étape de création d'un profil utilisateur.

A.4 Lancer la partie visualisation



Une fois que l'utilisateur a sélectionné un profil dans la liste et qu'il a appuyé sur « Time Series Visualizer », il peut alors commencer l'analyse d'une série temporelle.

L'écran de base de la partie visualisation possède 2 panels de couleur différente. Le panel supérieur jaune indique à l'utilisateur quelle série temporelle il est en train d'analyser. Cela lui permet de savoir à tout moment ce qu'il est en train de consulter. Ce panel se modifie automatiquement dès que la série temporelle analysée est changée.

Dans cette partie, quatre menus sont proposés à l'utilisateur. Tous les menus ne sont pas accessibles directement comme il est possible de le constater sur l'image. Certains ne s'activeront que lorsqu'une visualisation est affichée ou une configuration établie.

Les quatre menus proposés sont :

- « START »
- « INTERACTION »
- « MISC. » (Miscellaneous)
- « INFORMATIONS » (représenté par le symbole « (i) »)

A.4.1 Menu « start »

C'est dans le menu « start » que se trouvent les principaux sous menu permettant la création d'une visualisation.

En effet, la première étape par laquelle l'utilisateur doit passer est celle de la configuration des paramètres (« Settings») en vue de créer n'importe quelle visualisation.

Il est également possible pour l'utilisateur, dans la partie visualisation même, de modifier son profil ou d'en créer un nouveau, d'effacer l'écran (« Clear») ou de quitter la partie visualisation.

A.4.2 Menu « interaction »

Ce menu, inaccessible au départ, n'est rendu disponible qu'une fois que l'utilisateur a affiché une visualisation et cela, quelle qu'elle soit.

Ce menu permet d'obtenir des informations sur les valeurs dessinées à l'écran (date, valeurs de tous les paramètres), de « switcher » vers une autre visualisation à partir de la visualisation actuelle, d'obtenir un zoom ou un focus sur un mois en particulier selon une autre représentation.

A.4.3 Menu « misc. » (Miscellaneous)

Ce menu donne accès à deux fenêtres qui vont permettre à l'utilisateur d'obtenir des informations sur le mode de fonctionnement de l'application.

La fenêtre principale de ce menu est la fenêtre d'aide contextuelle où l'utilisateur pourra obtenir un texte explicatif en fonction d'un sujet sélectionné dans une liste. Les explications sont disponibles aussi bien en français qu'en anglais (selon la langue choisie dans le profil).

A.4.4 Menu « informations »

Ce menu, inaccessible au départ, n'est rendu disponible qu'une fois que l'utilisateur a effectué la configuration des paramètres de l'application.

Il lui permet d'obtenir des informations sur les différents paramètres principaux qu'il a sélectionné et lui permet également de réaliser une simulation de modification de sa configuration. De cette façon, il peut envisager d'autres sélections sans risquer de perdre l'actuelle configuration.

Finalement, une fois son analyse terminée, l'utilisateur peut quitter la partie visualisation. Comme pour chaque autre décision importante de l'application, une boîte de dialogue contenant un message de confirmation s'affichera.

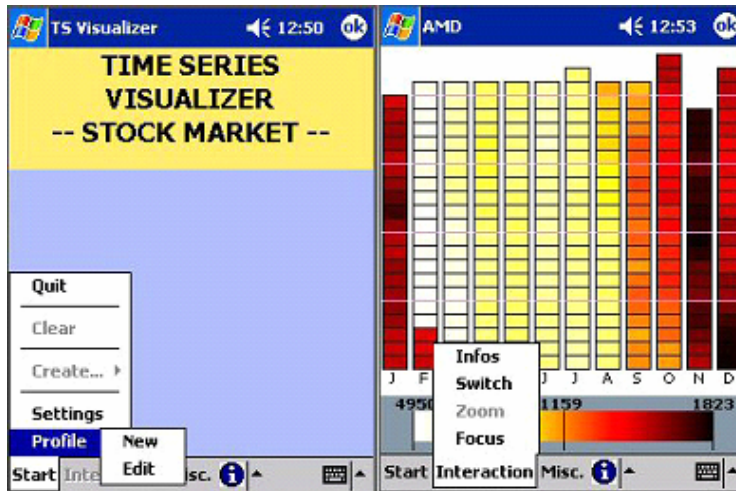


Figure A-7 : Menus présents dans la partie visualisation (1).

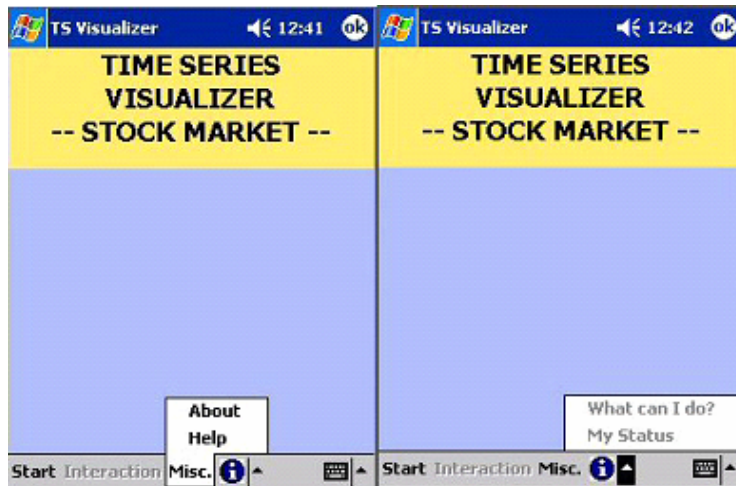


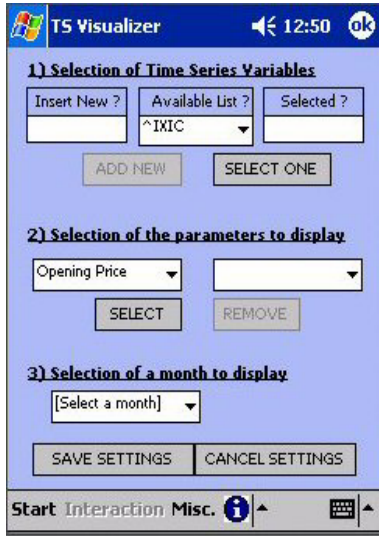
Figure A-8 : Menus présents dans la partie visualisation (2).

S'il décide de ne pas quitter, ses paramètres n'auront pas été supprimés et il pourra poursuivre son travail comme si de rien n'était.



Figure A-9 : Message de confirmation pour quitter le programme.

A.5 Configurer l'application



De façon à simplifier la procédure de configuration, tous les paramètres à configurer pour que l'application puisse fonctionner correctement sont réunis dans cette fenêtre.

Il est important de noter qu'une seule configuration ne permettra pas à l'utilisateur de créer toutes les visualisations en une seule fois à partir des mêmes données de base. Il lui faudra donc parfois sélectionner ou modifier des paramètres en fonction des visualisations qu'il désire afficher.

La procédure de configuration a été élaborée en trois étapes correspondant aux informations dont l'application a besoin pour créer une visualisation.

(a) La variable relative à la série temporelle à visualiser : Plusieurs variables relatives à une série temporelle sont proposées dans la liste « Available List ». La taille de cette liste dépend de la série temporelle elle-même. Il est important de noter qu'une seule variable à la fois doit être sélectionnée pour visualiser la série temporelle.

(b) Les différents paramètres en rapport avec la série temporelle : L'utilisateur choisit les paramètres à afficher parmi ceux de la liste de gauche et ils se retrouveront ainsi dans la liste de droite. Il lui sera toujours possible de « désélectionner » des paramètres s'il le souhaite. L'utilisateur peut décider de sélectionner le nombre de paramètres qu'il lui plaît. Cocher la case « All » lui permettra de gagner du temps et peut être utile dans le cas d'une visualisation acceptant autant de paramètres qu'il n'en existe (ex : « Multi-Lines »).

(c) Un mois de l'année : Pour certaines visualisation, il est nécessaire de sélectionner un mois en particulier. Toutefois, la sélection d'un mois dans la configuration n'aura pas d'influence si l'utilisateur décide de zoomer sur un autre mois à partir d'une visualisation.

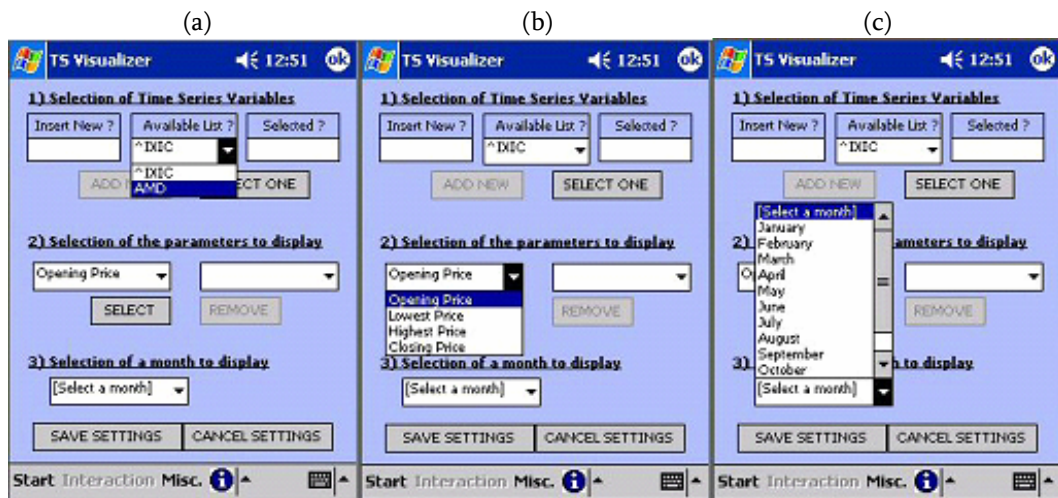


Figure A-10 : Configuration de l'application.

De petits messages d'aide peuvent apparaître au moment où l'utilisateur sélectionne la variable à utiliser par l'application. Pour cela, il doit cliquer sur l'un des trois textes au dessus des listes de variables.

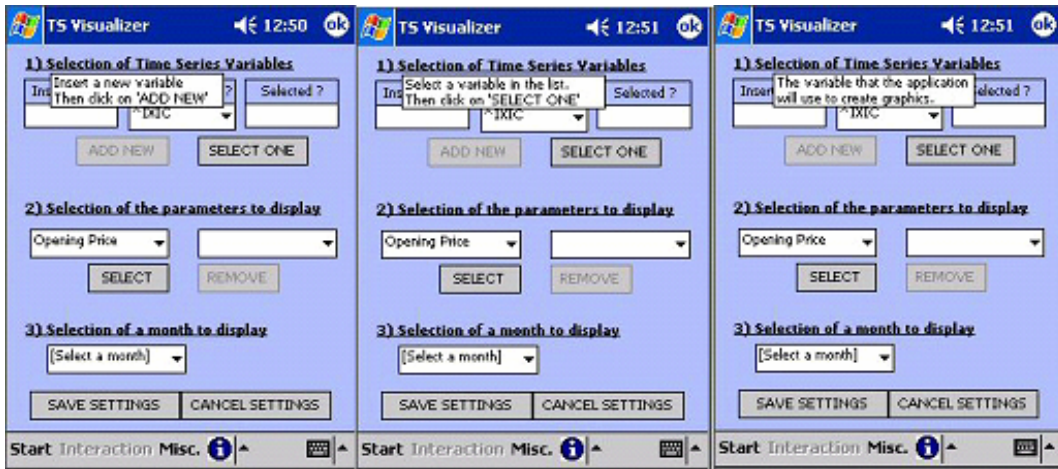


Figure A-11 : Messages d'aide.

(a) Enregistrement de la sélection : L'utilisateur doit ensuite sauvegarder sa sélection (« Save Settings ») s'il veut pouvoir créer une visualisation. En cas d'erreur, un message lui indiquant quel(s) paramètre(s) n'a (ont) pas été sélectionné(s) correctement.

(b) Enregistrement en cours : Si la sélection de l'utilisateur est correcte, sa configuration sera enregistrée et les informations seront utilisées pour créer les données.

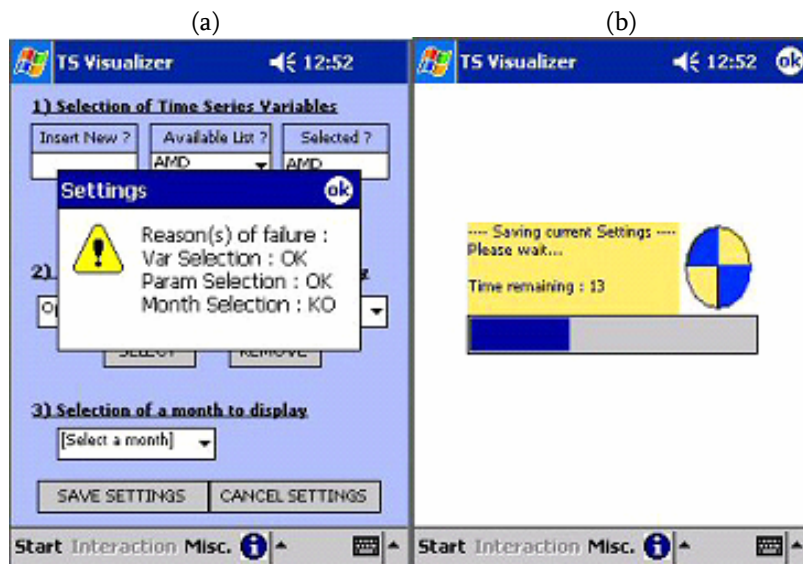


Figure A-12 : Enregistrement d'une configuration.

A.6 Créer une visualisation

Une fois que l'utilisateur a sauvegardé sa configuration, il lui est désormais possible de créer une visualisation. Il est important de noter que toutes les visualisations ne peuvent pas être

créées à partir d'une seule et même configuration. En effet, toutes les visualisations ne permettent pas de manipuler le même nombre de paramètres. Il faudra donc parfois modifier la configuration de façon à pouvoir utiliser toutes les spécificités des visualisations.

Pour créer une visualisation, il suffit d'aller dans le menu principal (menu « start ») et d'appuyer sur « Create » et ensuite choisir la visualisation souhaitée. En cas d'incompatibilité de la configuration avec la visualisation sélectionnée, un message d'erreur indiquera à l'utilisateur les motifs d'incompatibilité. A partir de ce message, il pourra éventuellement modifier la configuration s'il tient absolument à afficher la visualisation qu'il a choisie ou bien changer la visualisation à afficher.

Lorsqu'une visualisation est affichée sur l'écran, il est toujours possible de l'effacer (« Clear ») ou d'en créer une autre. Il suffit de reproduire la démarche initiale en sélectionnant une autre visualisation.

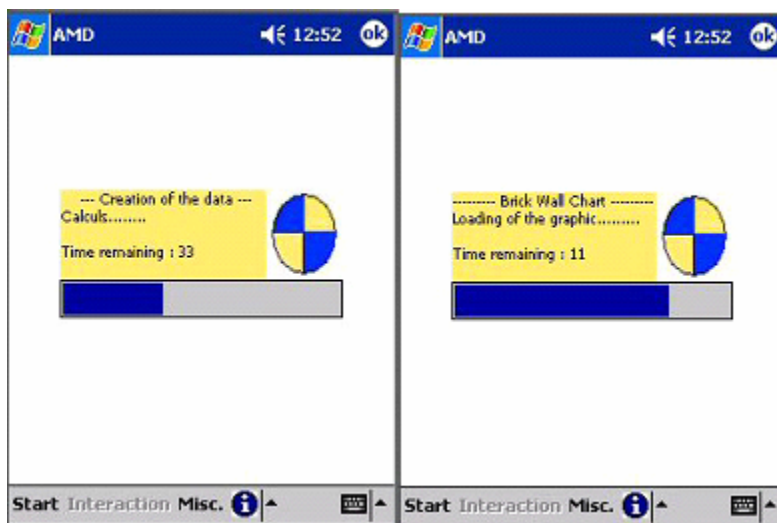


Figure A-13 : Chargement d'une visualisation.

Six visualisations peuvent être créées par l'utilisateur

- Visualisation « Brick Wall Chart »
- Visualisation « Bar Chart »
- Visualisation « Stacked Bar Chart »
 - 2 écrans : 6 premiers mois de l'année et 6 derniers mois de l'année
- Visualisation « Mountain »
 - avec un zoom possible sur chacun des mois de l'année.
- Visualisation « Multi-Lines»
 - avec un zoom possible sur chacun des mois de l'année.

Voici quelques captures d'écran des différentes visualisations obtenues :

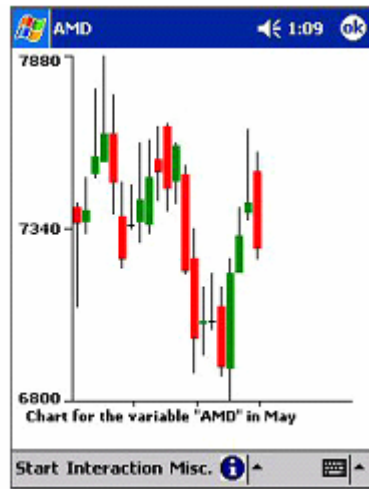
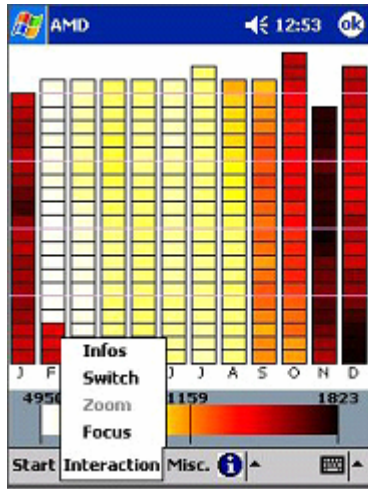


Figure A-14 : Brick Wall Chart - Bar Chart.

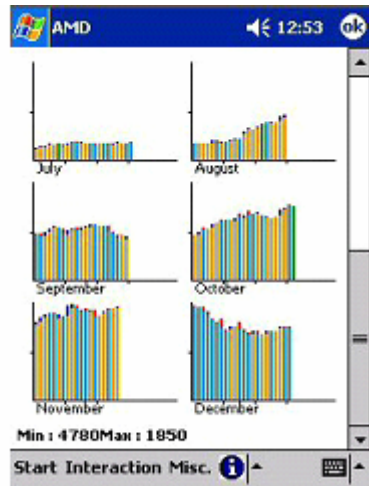
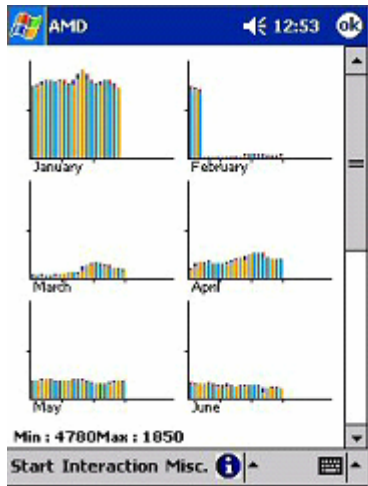


Figure A-15 : Stacked Bar Chart de Janvier à Juin et de Juillet à Décembre.

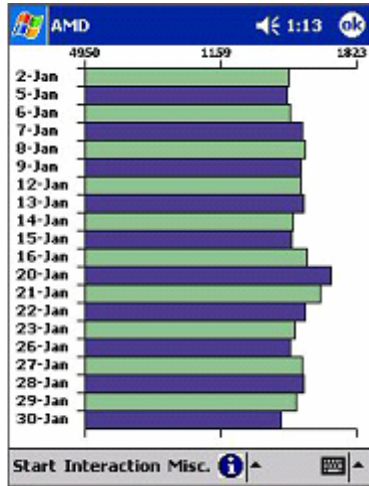
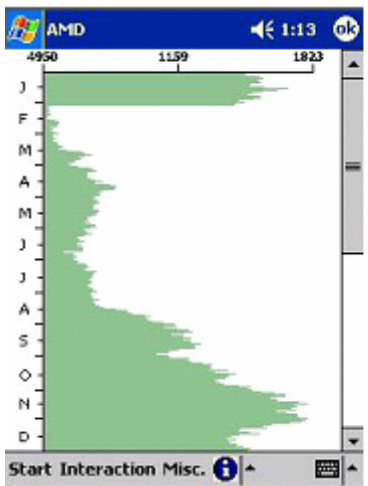


Figure A-16 : Mountain et zoom sur le mois de Janvier.

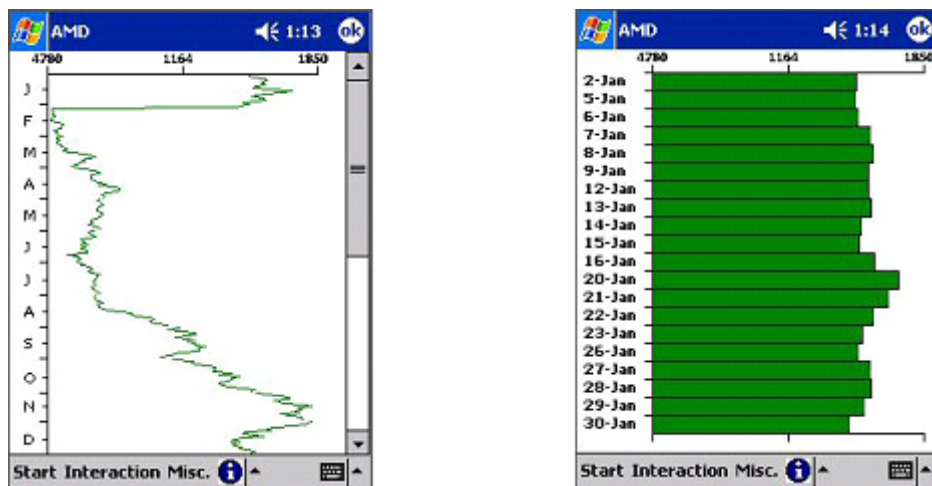


Figure A-17 : Multi-Lines et zoom sur le mois de janvier.

A.7 Interaction avec une visualisation

Une fois qu'il a complété sa configuration et créé une visualisation, l'utilisateur dispose de plusieurs interactions qui lui permettront d'effectuer au mieux son analyse de la série temporelle.

L'application propose quatre interactions qui seront rendues accessibles selon les visualisations que l'utilisateur a décidé d'afficher.

Les interactions sont les suivantes :

- Infos
- Switch
- Zoom
- Focus

A.7.1 Interaction « infos »

L'interaction « infos » permet à l'utilisateur, lorsqu'il appuie sur l'écran avec le stylo, d'obtenir des renseignements sur le point où il vient d'appuyer. Ces renseignements sont affichés dans un petit panneau et restent à la disposition de l'utilisateur tant qu'il laisse le stylo appuyé sur l'écran.

Quelle que soit la visualisation affichée, les informations se présenteront toujours à l'utilisateur de la même manière, à savoir, la date de la valeur touchée sur l'écran suivie par les valeurs de chaque paramètre en rapport avec la donnée pointée. Il se peut donc que, selon la variable, ce panneau soit plus ou moins grand.

A.7.2 Interaction « Focus »

Quoi de plus intéressant que de pouvoir passer d'une visualisation à l'autre pour avoir une représentation différente de l'information que nous sommes en train de visualiser ?

C'est exactement ce que permet de faire l'interaction « switch ». Cette opération permet de passer d'une visualisation à l'autre sans pour autant devoir modifier toute la configuration qui a

été réalisée quelques instants précédemment. En fonction des modifications de paramètres, certaines visualisations seront marquées comme accessibles et d'autres pas.

A.7.3 Interaction « Zoom »

Lorsque l'utilisateur a décidé d'afficher une visualisation qui représente l'entièreté des valeurs de toute l'année, il peut être intéressé par le fait d'observer plus en détail une partie des informations.

Dans cette optique, l'interaction « zoom » permet de zoomer sur un mois en particulier pour obtenir un agrandissement selon le même mode de visualisation.

Cette opération est disponible pour les visualisation « Mountain » et « Multi-Lines ».

A.7.4 Interaction « switch »

Parallèlement à l'interaction « zoom » qui permet à l'utilisateur d'observer plus en détail une partie des informations mais selon un mode de représentation similaire, l'interaction « focus » permet de représenter plus en détail un mois mais selon un mode différent de visualisation.

Cette opération est disponible pour les visualisation « Brick Wall Chart » et « Stacked Bar Chart ». Cependant, pour que l'interaction « focus » puisse être utilisée, il faut que pour la série temporelle sélectionnée et analysée, l'utilisateur ait choisi au minimum deux paramètres à afficher.

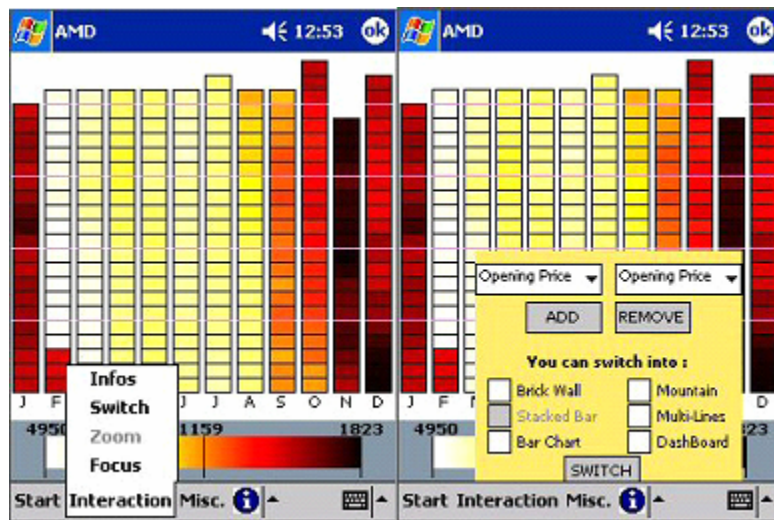


Figure A-18 : Interaction "Switch".

A.8 Autres fonctionnalités

A.8.1 Explorateur de fichiers

Lorsqu'il se trouve dans le menu principal, l'utilisateur a la possibilité d'accéder à un explorateur de fichiers dans lequel il pourra consulter brièvement les différents fichiers relatifs à l'application.

Cet explorateur est modifié « en temps réel » selon les opérations effectuées par l'utilisateur (et autorisées par le système). Par exemple, il lui est possible de supprimer un profil existant à

partir de l'explorateur et cela modifiera la liste de profils qu'il retrouvera dans le menu principal.

Cela dit, cet explorateur de fichiers, comme son nom l'indique, permet de naviguer à travers les fichiers mais ne permet pas d'en créer directement à partir de cette fenêtre.

Il pourra consulter :

- (a) Les profils des différents utilisateurs.
- (b) Les fichiers relatifs aux séries temporelles avec une brève description de chacune d'elles.
- (c) Les fichiers relatifs aux visualisations avec une brève description de chacune d'elles (caractéristiques et interactions disponibles).

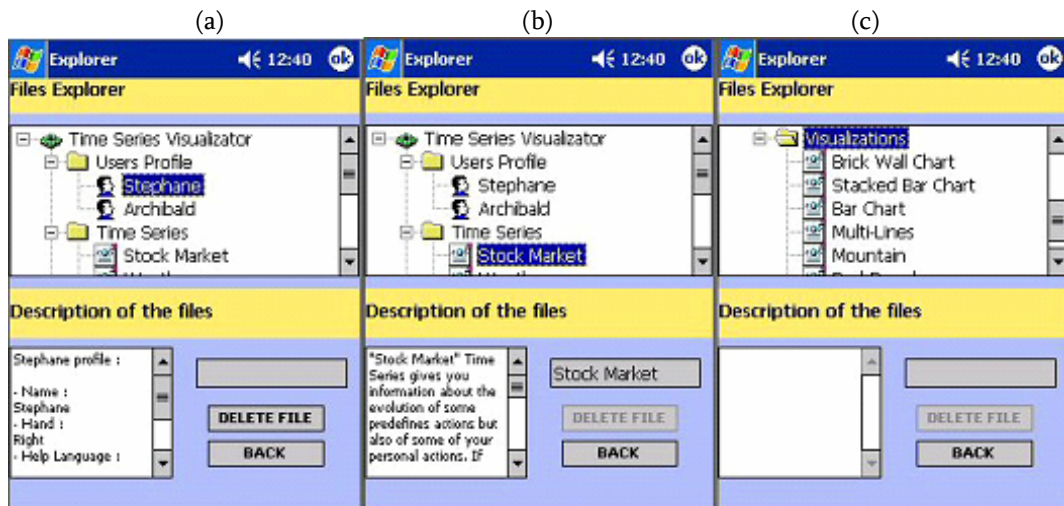


Figure A-19 : Explorateur de fichiers.

A.8.2 Statut de l'utilisateur

Une fois que l'utilisateur a configuré l'application, le menu « information » lui est désormais accessible.

Ce menu possède deux sous-menus qui permettent à l'utilisateur, à tout moment de son analyse, d'afficher des panneaux lui indiquant quel est son statut dans le programme.

Le premier panneau (a) lui fournit un récapitulatif des principaux paramètres qui le concernent :

- Son surnom (utilisé pour sauvegarder un profil)
- La visualisation par défaut qu'il a choisit
- La variable appartenant à la série temporelle qu'il a choisi d'analyser.
- Le mois sélectionné
- Le(s) paramètre(s) sélectionné(s) pour étudier la série temporelle.

L'avantage de ce système est que l'utilisateur peut également procéder à une simulation de modification de sa configuration (b) pour voir quelles sont les visualisations qu'il pourrait créer en ajoutant ou supprimant un ou des paramètre(s).

La partie supérieure de l'écran lui indique quelle est sa situation actuelle, c'est-à-dire les visualisations qu'il pourrait concrètement afficher s'il gardait sa configuration.

La partie centrale lui offre le choix entre ajouter ou retirer des paramètres. Cela le renseigne également sur le nombre de paramètres sélectionnés avant simulation mais également le nombre qui seront accessibles après la simulation.

Enfin, la partie inférieure de la fenêtre lui indique les résultats de sa simulation. Tout comme la partie supérieure, un système de cases cochées signale les visualisations réalisables.

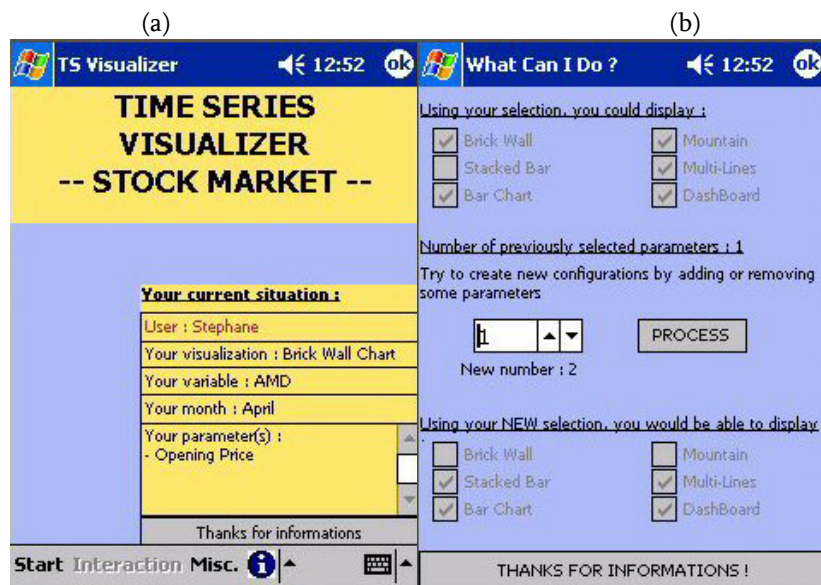


Figure A-20 : Statut de l'utilisateur.

A.8.3 Aide

L'utilisateur qui le désire peut afficher une fenêtre contenant une liste de sujets pour lesquels il existe un texte d'aide. Ces textes d'aide sont disponibles en français ou en anglais selon la langue sélectionnée dans le profil.

Comme il est possible de le constater sur les images, les sujets décrits par un texte sont les suivants :

- Introduction.
 - Une introduction qui renseigne l'utilisateur sur le but général de l'application.
- Configuration.
 - Ce texte explique très brièvement à l'utilisateur quelles sont les étapes à suivre pour compléter une configuration.
- Types de données.
 - Ce texte explique très brièvement à l'utilisateur la façon de manipuler les données qui lui seront proposées et qu'il choisira pour compléter sa configuration.

- Visualisations.
 - Ce texte explique très brièvement à l'utilisateur comment manipuler les visualisations et les différentes options qui lui sont offertes lorsqu'il a affiché un graphique.
- Brick Wall Chart, Stacked Bar Chart, Bar Chart, Mountain, Multi-Lines, DashBoard.
 - Les différentes caractéristiques des visualisations.

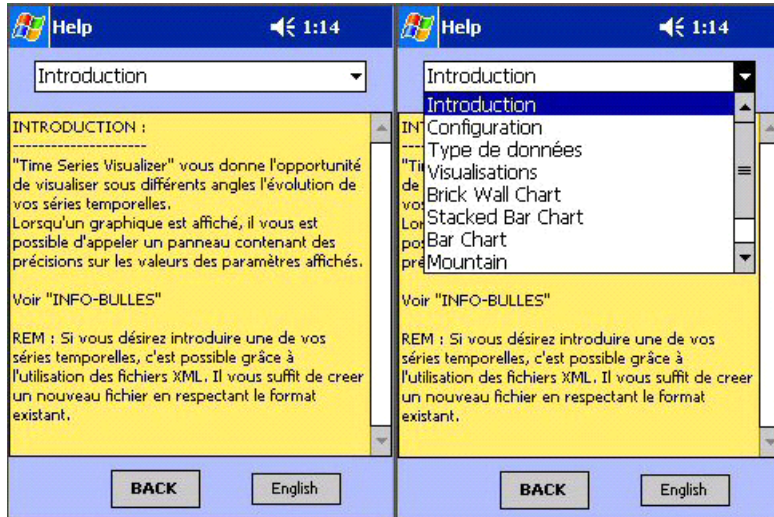


Figure A-21 : Topics d'aide.

Annexes B : Tables

B.1 Table de Chi carré (χ^2)

n	0,01	0,025	0,05	0,1	0,5	0,9	0,95	0,975	0,99
1	0,0002	0,001	0,004	0,02	0,45	2,71	3,84	5,02	6,63
2	0,02	0,05	0,10	0,21	1,39	4,61	5,99	7,38	9,21
3	0,11	0,22	0,35	0,58	2,37	6,25	7,81	9,35	11,34
4	0,30	0,48	0,71	1,06	3,36	7,78	9,49	11,14	13,28
5	0,55	0,83	1,15	1,61	4,35	9,24	11,07	12,83	15,09
6	0,87	1,24	1,64	2,20	5,35	10,64	12,59	14,45	16,81
7	1,24	1,69	2,17	2,83	6,35	12,02	14,07	16,01	18,48
8	1,65	2,18	2,73	3,49	7,34	13,36	15,51	17,53	20,09
9	2,09	2,70	3,33	4,17	8,34	14,68	16,92	19,02	21,67
10	2,56	3,25	3,94	4,87	9,34	15,99	18,31	20,48	23,21
11	3,05	3,82	4,57	5,58	10,34	17,28	19,68	21,92	24,73
12	3,57	4,40	5,23	6,30	11,34	18,55	21,03	23,34	26,22
13	4,11	5,01	5,89	7,04	12,34	19,81	22,36	24,74	27,69
14	4,66	5,63	6,57	7,79	13,34	21,06	23,68	26,12	29,14
15	5,23	6,26	7,26	8,55	14,34	22,31	25,00	27,49	30,58
16	5,81	6,91	7,96	9,31	15,34	23,54	26,30	28,85	32,00
17	6,41	7,56	8,67	10,09	16,34	24,77	27,59	30,19	33,41
18	7,01	8,23	9,39	10,86	17,34	25,99	28,87	31,53	34,81
19	7,63	8,91	10,12	11,65	18,34	27,20	30,14	32,85	36,19
20	8,26	9,59	10,85	12,44	19,34	28,41	31,41	34,17	37,57
21	8,90	10,28	11,59	13,24	20,34	29,62	32,67	35,48	38,93
22	9,54	10,98	12,34	14,04	21,34	30,81	33,92	36,78	40,29
23	10,20	11,69	13,09	14,85	22,34	32,01	35,17	38,08	41,64
24	10,86	12,40	13,85	15,66	23,34	33,20	36,42	39,36	42,98
25	11,52	13,12	14,61	16,47	24,34	34,38	37,65	40,65	44,31
26	12,20	13,84	15,38	17,29	25,34	35,56	38,89	41,92	45,64
27	12,88	14,57	16,15	18,11	26,34	36,74	40,11	43,19	46,96
28	13,56	15,31	16,93	18,94	27,34	37,92	41,34	44,46	48,28
29	14,26	16,05	17,71	19,77	28,34	39,09	42,56	45,72	49,59
30	14,95	16,79	18,49	20,60	29,34	40,26	43,77	46,98	50,89
40	22,16	24,43	26,51	29,05	39,34	51,81	55,76	59,34	63,69
50	29,71	32,36	34,76	37,69	49,33	63,17	67,50	71,42	76,15
60	37,48	40,48	43,19	46,46	59,33	74,40	79,08	83,30	88,38
70	45,44	48,76	51,74	55,33	69,33	85,53	90,53	95,02	100,43
80	53,54	57,15	60,39	64,28	79,33	96,58	101,88	106,63	112,33
90	61,75	65,65	69,13	73,29	89,33	107,57	113,15	118,14	124,12
100	70,06	74,22	77,93	82,36	99,33	118,50	124,34	129,56	135,81

Figure B-1 : Table de Chi carré.

B.2 Table de Fisher

	v_1	1	2	3	4	5	6	7	8	9	10	12	15	20
v_2														
1		161	199,5	215,7	224,6	230,2	234	236,8	239	240,5	241,9	243,9	245,9	248
2		18,5	19	19,16	19,25	19,3	19,33	19,35	19,4	19,38	19,4	19,41	19,43	19,45
3		10,1	9,55	9,28	9,12	9,01	8,94	8,89	8,85	8,81	8,79	8,74	8,7	8,66
4		7,71	6,94	6,59	6,39	6,26	6,16	6,09	6,04	6	5,96	5,91	5,86	5,8
5		6,61	5,79	5,41	5,19	5,05	4,95	4,88	4,82	4,77	4,74	4,68	4,62	4,56
6		5,99	5,14	4,76	4,53	4,39	4,28	4,21	4,15	4,1	4,06	4	3,94	3,87
7		5,59	4,74	4,35	4,12	3,97	3,87	3,79	3,73	3,68	3,64	3,57	3,51	3,44
8		5,32	4,46	4,07	3,84	3,69	3,58	3,5	3,44	3,39	3,35	3,28	3,22	3,15
9		5,12	4,26	3,86	3,63	3,48	3,37	3,29	3,23	3,18	3,14	3,07	3,01	2,94
10		4,96	4,1	3,71	3,48	3,33	3,22	3,14	3,07	3,02	2,98	2,91	2,85	2,77
11		4,84	3,98	3,59	3,36	3,2	3,09	3,01	2,95	2,9	2,85	2,79	2,72	2,65
12		4,75	3,89	3,49	3,26	3,11	3	2,91	2,85	2,8	2,75	2,69	2,62	2,54
13		4,67	3,81	3,41	3,18	3,03	2,92	2,83	2,77	2,71	2,67	2,6	2,53	2,46
14		4,6	3,74	3,34	3,11	2,96	2,85	2,76	2,7	2,65	2,6	2,53	2,46	2,39
15		4,54	3,68	3,29	3,06	2,9	2,79	2,71	2,64	2,59	2,54	2,48	2,4	2,33
16		4,49	3,63	3,24	3,01	2,85	2,74	2,66	2,59	2,54	2,49	2,42	2,35	2,28
17		4,45	3,59	3,2	2,96	2,81	2,7	2,61	2,55	2,49	2,45	2,38	2,31	2,23
18		4,41	3,55	3,16	2,93	2,77	2,66	2,58	2,51	2,46	2,41	2,34	2,27	2,19
19		4,38	3,52	3,13	2,9	2,74	2,63	2,54	2,48	2,42	2,38	2,31	2,23	2,16
20		4,35	3,49	3,1	2,87	2,71	2,6	2,51	2,45	2,39	2,35	2,28	2,2	2,12
21		4,32	3,47	3,07	2,84	2,68	2,57	2,49	2,42	2,37	2,32	2,25	2,18	2,1
22		4,3	3,44	3,05	2,82	2,66	2,55	2,46	2,4	2,34	2,3	2,23	2,15	2,07
23		4,28	3,42	3,03	2,8	2,64	2,53	2,44	2,37	2,32	2,27	2,2	2,13	2,05
24		4,26	3,4	3,01	2,78	2,62	2,51	2,42	2,36	2,3	2,25	2,18	2,11	2,03
25		4,24	3,39	2,99	2,76	2,6	2,49	2,4	2,34	2,28	2,24	2,16	2,09	2,01
26		4,23	3,37	2,98	2,74	2,59	2,47	2,39	2,32	2,27	2,22	2,15	2,07	1,99
27		4,21	3,35	2,96	2,73	2,57	2,46	2,37	2,31	2,25	2,2	2,13	2,06	1,97
28		4,2	3,34	2,95	2,71	2,56	2,45	2,36	2,29	2,24	2,19	2,12	2,04	1,96
29		4,18	3,33	2,93	2,7	2,55	2,43	2,35	2,28	2,22	2,18	2,1	2,03	1,94
30		4,17	3,32	2,92	2,69	2,53	2,42	2,33	2,27	2,21	2,16	2,09	2,01	1,93
40		4,08	3,23	2,84	2,61	2,45	2,34	2,25	2,18	2,12	2,08	2	1,92	1,84
60		4	3,15	2,76	2,53	2,37	2,25	2,17	2,1	2,04	1,99	1,92	1,84	1,75
120		3,92	3,07	2,68	2,45	2,29	2,17	2,09	2,02	1,96	1,91	1,83	1,75	1,66
infini		3,84	3	2,6	2,37	2,21	2,1	2,01	1,94	1,88	1,83	1,75	1,67	1,57

Figure B-2 : Table de Fisher.

B.3 Tableaux de calculs des tests de l'ANOVA à deux facteurs

Tableau de calculs du premier test de l'ANOVA à deux facteurs concernant l'efficacité des visualisations :

Utilisateur	BWC	SBC	ML	Moyennes
1	6	3	5	4,78
2	19	10	13	14,11
3	16	10	17	14,44
4	10	15	9	11,33
5	20	17	11	16,11
6	8	5	3	5,22
7	25	6	3	11,44
8	17	13	14	14,78
9	8	4	2	4,78
10	16	11	5	10,56
11	8	3	14	8,22
12	11	10	14	11,67
13	19	10	13	14,11
14	19	12	14	15,00
15	14	11	21	15,44
16	3	4	2	3,11
17	22	8	11	13,56
18	6	3	10	6,33
19	10	3	10	7,67
20	19	5	11	11,78
21	16	8	9	11,00
22	5	2	9	5,44
23	3	5	3	3,78
Moyennes	13,17	7,74	9,70	13,80

SC Simple	SC carré	Valeurs ajustées			Valeurs résiduelles		
-9,03	81,4712717	4,15	-1,29	0,67	2,19	4,29	4,33
0,31	0,09436542	13,48	8,05	0,67	5,85	1,95	12,33
0,64	0,41026955	13,81	8,38	0,67	2,52	1,62	16,33
-2,47	6,10380623	10,70	5,27	0,67	-0,70	9,73	8,33
2,31	5,32312358	15,48	10,05	0,67	4,85	6,95	10,33
-8,58	73,6455637	4,59	-0,84	0,67	3,07	5,84	2,33
-2,36	5,5671323	10,81	5,38	0,67	14,52	0,62	2,33
0,97	0,94839592	14,15	8,71	0,67	3,19	4,29	13,33
-9,03	81,4712717	4,15	-1,29	0,67	4,19	5,29	1,33
-3,25	10,5518818	9,93	4,49	0,67	5,74	6,51	4,33
-5,58	31,1553676	7,59	2,16	0,67	0,07	0,84	13,33
-2,14	4,56785852	11,04	5,60	0,67	-0,04	4,40	13,33
0,31	0,09436542	13,48	8,05	0,67	5,85	1,95	12,33
1,20	1,43060361	14,37	8,94	0,67	4,63	3,06	13,33
1,64	2,69131531	14,81	9,38	0,67	-0,48	1,62	20,33
-10,69	114,336195	2,48	-2,95	0,67	0,85	6,95	1,33
-0,25	0,06168568	12,93	7,49	0,67	8,74	0,51	10,33
-7,47	55,8096886	5,70	0,27	0,67	0,30	2,73	9,33
-6,14	37,6658977	7,04	1,60	0,67	2,96	1,40	9,33
-2,03	4,10525866	11,15	5,71	0,67	8,19	-0,71	10,33
-2,80	7,86197616	10,37	4,94	0,67	5,63	3,06	8,33
-8,36	69,8808578	4,81	-0,62	0,67	0,52	2,62	8,33
-10,03	100,523559	3,15	-2,29	0,67	0,19	7,29	2,33

SC Simple	-0,63	-6,06	-4,11
SC carré	0,39691074	36,7816915	16,8778774

Source	SC	ddl	CM	F	p.c. 001	p.c. 005
Visualisations	855,107851	2	427,6	6,393	Reject	Reject
Utilisateurs	2087,31514	22	94,88	1,419	Accept	Accept
Résiduelle	2942,42299	44	66,87			

Tableau de calculs du second test de l'ANOVA à deux facteurs concernant l'efficacité des visualisations :

Utilisateur	BWC	SBC	ML	Moyennes
1	8	8	10	8,56
2	17	9	11	12,44
3	19	5	9	11,11
4	18	4	11	11,00
5	20	10	11	13,56
6	6	10	13	9,78
7	20	5	3	9,33
8	19	9	5	11,11
9	25	7	9	13,67
10	19	10	6	11,78
11	6	8	13	9,11
12	16	9	12	12,44
13	16	9	12	12,44
14	16	11	16	14,22
15	13	10	11	11,33
16	5	5	3	4,44
17	16	9	12	12,44
18	10	2	8	6,67
19	6	8	2	5,33
20	16	6	10	10,56
21	15	9	10	11,44
22	6	3	8	5,67
23	6	6	4	5,33
Moyennes	13,93	7,48	9,09	13,75

SC Simple	SC carré	Valeurs ajustées			Valeurs résiduelles		
-5,20	26,9992311	8,73	2,28	3,89	-1,06	5,72	6,11
-1,31	1,7087445	12,62	6,17	3,89	4,71	2,83	7,11
-2,64	6,97236106	11,29	4,84	3,89	8,05	0,16	5,11
-2,75	7,5714896	11,18	4,73	3,89	6,82	-0,73	7,11
-0,20	0,03844675	13,73	7,28	3,89	5,94	2,72	7,11
-3,97	15,7915332	9,95	3,50	3,89	-3,62	6,50	9,11
-4,42	19,5213807	9,51	3,06	3,89	10,49	1,94	-0,89
-2,64	6,97236106	11,29	4,84	3,89	8,05	4,16	1,11
-0,08	0,00721945	13,84	7,39	3,89	11,16	-0,39	5,11
-1,97	3,89610833	11,95	5,50	3,89	7,38	4,50	2,11
-4,64	21,5344526	9,29	2,84	3,89	-2,95	5,16	9,11
-1,31	1,7087445	12,62	6,17	3,89	3,71	2,83	8,11
-1,31	1,7087445	12,62	6,17	3,89	3,71	2,83	8,11
0,47	0,22145329	14,40	7,95	3,89	1,27	3,05	12,11
-2,42	5,84817805	11,51	5,06	3,89	1,49	4,94	7,11
-9,31	86,6237772	4,62	-1,83	3,89	0,71	6,83	-0,89
-1,31	1,7087445	12,62	6,17	3,89	3,71	2,83	8,11
-7,08	50,1967619	6,84	0,39	3,89	3,16	1,61	4,11
-8,42	70,8677859	5,51	-0,94	3,89	0,49	8,94	-1,89
-3,20	10,2149173	10,73	4,28	3,89	4,94	1,72	6,11
-2,31	5,32312358	11,62	5,17	3,89	3,71	3,83	6,11
-8,08	65,3666966	5,84	-0,61	3,89	0,16	3,61	4,11
-8,42	70,8677859	5,51	-0,94	3,89	0,49	6,94	0,11

SC Simple	0,18	-6,27	-4,66
SC carré	0,0309416	39,3552103	21,7592159

Source	SC	ddl	CM	F	p.c. 001	p.c. 005
Visualisations	905,881493	2	452,94	8,4774	Reject	Reject
Utilisateurs	1445,01012	22	65,682	1,2293	Accept	Accept
Résiduelle	2350,89162	44	53,429			

B.4 Tableaux de calculs des tests de la médiane

BWC / SBC

Valeur	2	<u>3</u>	<u>3</u>	3	3	3	3	4	4	5	5	5
Rang	1	2	3	4	5	6	7	8	9	<u>10</u>	<u>11</u>	<u>12</u>
Valeur	5	<u>6</u>	<u>6</u>	<u>6</u>	<u>8</u>	<u>8</u>	<u>8</u>	8	8	<u>10</u>	<u>10</u>	<u>10</u>
Rang	13	<u>14</u>	<u>15</u>	<u>16</u>	17	18	19	20	21	<u>22</u>	<u>23</u>	<u>24</u>
Valeur	10	10	10	<u>11</u>	11	11	12	13	<u>14</u>	15	<u>16</u>	<u>16</u>
Rang	25	26	27	<u>28</u>	29	30	31	32	<u>33</u>	34	<u>35</u>	<u>36</u>
Valeur	16	17	17	<u>19</u>	<u>19</u>	<u>19</u>	<u>19</u>	20	<u>22</u>	25		
Rang	<u>37</u>	<u>38</u>	39	<u>40</u>	<u>41</u>	<u>42</u>	<u>43</u>	<u>44</u>	<u>45</u>	<u>46</u>		

	BWC	SBC
Au dessus de M	13	6
En dessous de M	10	17

BWC / ML

Valeur	2	2	<u>3</u>	<u>3</u>	3	3	3	5	5	5	6	<u>6</u>
Rang	1	2	<u>3</u>	<u>4</u>	5	6	7	<u>8</u>	9	10	<u>11</u>	<u>12</u>
Valeur	<u>8</u>	<u>8</u>	<u>8</u>	9	9	9	<u>10</u>	<u>10</u>	10	10	<u>11</u>	11
Rang	<u>13</u>	<u>14</u>	<u>15</u>	16	17	18	<u>19</u>	<u>20</u>	21	22	<u>23</u>	<u>24</u>
Valeur	11	11	13	13	<u>14</u>	14	14	14	14	<u>16</u>	<u>16</u>	<u>16</u>
Rang	25	26	27	28	<u>29</u>	30	31	32	33	<u>34</u>	<u>35</u>	<u>36</u>
Valeur	17	17	<u>19</u>	<u>19</u>	<u>19</u>	<u>19</u>	<u>20</u>	21	<u>22</u>	25		
Rang	<u>37</u>	38	<u>39</u>	<u>40</u>	<u>41</u>	<u>42</u>	<u>43</u>	44	<u>45</u>	46		

	BWC	ML
Au dessus de M	11	9
En dessous de M	12	14

SBC / ML

Valeur	2	<u>3</u>	<u>3</u>	3	3	3	3	4	4	5	5	5
Rang	1	2	3	4	5	6	7	8	9	<u>10</u>	<u>11</u>	<u>12</u>
Valeur	5	<u>6</u>	<u>6</u>	<u>6</u>	<u>8</u>	<u>8</u>	<u>8</u>	8	8	<u>10</u>	<u>10</u>	10
Rang	13	<u>14</u>	<u>15</u>	<u>16</u>	17	18	<u>19</u>	20	21	<u>22</u>	<u>23</u>	<u>24</u>
Valeur	10	10	10	<u>11</u>	11	11	12	13	14	15	<u>16</u>	<u>16</u>
Rang	25	26	27	<u>28</u>	29	30	31	32	<u>33</u>	34	<u>35</u>	<u>36</u>
Valeur	16	17	17	<u>19</u>	<u>19</u>	<u>19</u>	<u>19</u>	20	<u>22</u>	25		
Rang	<u>37</u>	<u>38</u>	39	<u>40</u>	<u>41</u>	<u>42</u>	<u>43</u>	<u>44</u>	<u>45</u>	<u>46</u>		

	SBC	ML
Au dessus de M	10	13
En dessous de M	13	10