

719-004-143-1

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

**Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.**
Option : Intelligence Artificielle

Sujet :

**Cartographie en 2D des signaux
EEG dans une architecture Client
Serveur**

Présenté par : DJEBRANI Abdenour
MAZOUZI Abdelghani

Promoteur : M^r ALLAM

Organisme d'accueil : Centre de développement des technologies avancées (CDTA).

Soutenu le: date soutenance, devant le jury composé de :

Nom. président du jury, grade, organisme

Président

Nom examinateur 1, grade, organisme

Examineur

Nom examinateur 2, grade, organisme

Examineur



- 2005/2006-

MIG-004-143-1

REMERCIEMENTS



Nous tenons à remercier ici toutes les personnes à qui nous devons d'avoir pu réaliser ce projet de fin d'étude.

Il nous vient naturellement à l'esprit notre promoteur M^r ALLAM qui nous a beaucoup aidé à réaliser ce modeste travail.

Nous remercions vivement BELKAID Ismail, qui par ses idées, ses commentaires et ses remarques constructives, nous a apporté une aide dans la rédaction de ce mémoire.

Les diverses connaissances que nous avons pu acquérir grâce à nous enseignants de primaires à l'université nous ont fourni la matière à ce projet. Pour cela, nous les en remercions chaleureusement.

De nombreuse conversation avec nos collègues nous ont permis d'enrichir et d'améliorer ce travail si intéressant, nous leur sommes très reconnaissants.

Les encouragements de nos amis nous ont donnés la force pour dépasser toutes les obstacles et aller devant, pour eux tous nous disons merci.

Nous souhaitons également remercier nos parents pour leur soutien et leurs encouragements.

Enfin, nous dédions ce travail à toute la famille DJEBRANI et MAZOUZI.

Abdeghani et Abdenour

إهداء

الحمد لله و هو المستحق للحمد و الثناء، نستعين به في السراء و
الضراء و نستغفره و نستهديه لما يقربنا إليه، و نؤمن به و نتوكل عليه
في جميع حالاتنا، و الصلاة و السلام على أفضل مبعوث للعالمين و أول
مشقوع يوم الدين، سيدنا و نبينا محمد و على آله و صحبه أجمعين.
بفضله تعالى و عونـه و بتوفيق منه، و تتويجاً لمسيرة سنوات
عديدة و أطوار كثيرة، بداية بالطور الأساسي فالإكمالي فالثانوي وصولاً
إلى الجامعي، و ثمرةً لجهـد كل من ساهم في هذه المسيرة بدايةً بالوالدين
الكريمين اللذان يعود لهما الفضل الأكبر فيها و الحظ الأوفر من التقدير و
العرفان، مروراً بكل أفراد العائلة دون إستثناء، و بكل زملاء الدراسة
دون ذكر الأسماء، و بكل الأساتذة الذين أمدونا بهذا الفضل دون بخل و
إنكفاء، و بكل الأصدقاء في شتى المجالات و الأنحاء.
بفضلهم جميعاً جاء هذا العمل المحتشم، و إليهم نهدى هذا العمل
المقدم و باسمهم، نسأل الله تعالى أن يعلمنا ما ينفعنا و ينفعنا بما علمنا
و أن يزدنا علماً.

SOMMAIRE

Introduction générale	1
-----------------------------	---

CHAPITRE I : Chaîne d'acquisition de données

I.1 Introduction	3
I.2 Les différents types de signaux	3
I.2.1 Signal analogique.....	3
I.2.2 Signal numérique.....	3
I.3 Analyse fréquentielle	4
I.4 Echantillonnage des signaux	5
I.5 Les fenêtres d'analyse	7
I.6 Les Filtres	8
I.6.1 Définition.....	8
I.6.2 Classification des filtres.....	9
I.7 Acquisition de données.....	9
I.7.1 Introduction.....	9
I.7.2 La chaîne d'acquisition de données.....	9
I.7.3 Description de dispositif du EEG	10
18 Conclusion	12

CHAPITRE II Electroencéphalogramme

II.1 Introduction	13
II.2 Historique	13
II.3 Présentation	14
II.4 Utilisation de l'EEG	14
II.5 EEG analogique et EEG numérisé	15
II.6 Les activités EEG	16
II.7 Placement des électrodes et montages.....	17
II.8 Représentation graphique d'EEG	18
II.8.1 Amplification	19
II.8.2 Traitement des données	19
II.8.3 Représentation des résultats.....	19
II.9 Cartographie de cerveau d'EEG	19
II.9.1 Les inconvénients de tracé EEG.....	20
II.9.2 L'ordinateur et le cerveau.....	21
II.9.3 Cartographies en 2D de cerveau.....	21
II.9.4 Techniques d'interpolation pour la cartographie d'EEG.....	22
II.9.4.1 L'interpolation.....	23
II.9.4.2 Classification des méthodes d'interpolation.....	23
II.9.4.2.1 Méthode discrète « Polygones de Thiessens »	24
II.9.4.2.2 Méthode continue « Moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance ».....	25
II.9.5 Applications médicales	26
II.10 Conclusion	27

CHAPITRE III:Architecteur client /serveur

III.1 Introduction	28
III.2 La notion d'intranet	28
III.2.1 Principe.....	28
III.2.2 L'utilité d'un intranet.....	29
III.3 Architecture client/serveur.....	27
III.3.1 Présentation.....	30
III.3.2 Avantages de l'architecture client/serveur	30
III.3.3 Fonctionnement d'un système client/serveur	31
III.4 Les protocoles	31
III.4.1 Le protocole TCP/IP.....	33
III.4.1.1 L'adresse IP	34
III.4.1.1.1 Le routage IP	34
III.4.1.1.2 Les ports	34
III.4.1.2 UDP	34
III.4.1.3 TCP.....	35
III.5 Les sockets	37
III.6 Conclusion	39

CHAPITRE IV: La description de l'application

IV.1 Introduction	40
IV.2 Spécification des besoins	41
IV.2 .1 Les cas d'utilisations	42
IV.2.1.1 Les acteurs	42
IV.2.1.2 Description textuelle des cas d'utilisation	42
IV.2.1.3 Les diagrammes de cas d'utilisation	43
IV.2.1.3.1 Le diagramme global des cas d'utilisation	44
IV.2.1.3.2 Le diagramme de cas d'utilisation acquisition et restitution des données	44
IV.2.1.3.3 Le diagramme de cas d'utilisation «Analyse et traitement des données»	45
IV.2.1.3.4 Le diagramme de cas d'utilisation « Visualisation des signaux»..	46
IV.2.1.3.5 Le diagramme de cas d'utilisation « Cartographie en 2D».....	46
IV.2.1.3.6 Le diagramme de cas d'utilisation « Stockage des données» ...	47
IV.2.1.3.7 Le diagramme de cas d'utilisation « Exportation des données» ..	48
IV.2.1.3.8 Le diagramme de cas d'utilisation « consultation».....	49
IV.3 Analyse.....	49
IV.3.1 Les paquetages (paquages).....	49
IV.3.1.1 Le paquetage « Application ».....	50
IV.3.1.2 Le paquetage « client-serveur ».....	50
IV.3.2 Les diagrammes de classe	50
IV.3.2.1 Le diagramme de classe « application ».....	50
IV.3.2.2 Le diagramme de classe « client-serveur ».....	52
IV.3.3 Les diagrammes d'activité	53
IV.3.3.1 Le diagramme d'activité globale	53
IV.3.3.2 Diagramme d'activité pour le cas«Acquisition et restitution des données»	55
IV.3.3.3 Diagramme d'activité pour le cas «Analyse et traitement des données»	56

IV.3.3.4 Diagramme d'activité pour le cas « Visualisation des données ».....	58
IV.3.3.5 diagrammes d'activité pour le cas « Cartographie en 2D ».....	59
IV.3.3.6 diagrammes d'activité pour le cas « Stockage des données ».....	61
IV.3.3.7 diagrammes d'activité pour le cas « Exportation des données ».....	63
IV.3.3.8 diagramme d'activité pour le cas « Consultation ».....	64
IV.4 Conception	65
IV.4.1 Les diagrammes de séquence	65
IV.4.1.1 Diagramme de séquence « Acquisition et restitution des données ».....	65
IV.4.1.2 diagramme de séquence « Cartographie en 2D ».....	68
IV.4.1.3 diagramme de séquence « Analyse et traitement des données ».....	70
IV.4.1.4 diagramme de séquence « Stockage des données ».....	72
IV.4.1.5 diagramme de séquence « exportation des données »	74
IV.4.2 Diagramme de composants	76
IV.5 Implémentation et test	77
IV.5.1 Langage de programmation LabVIEW.....	78
IV.5.2 Description de l'application réalisée.....	79
IV.5.2.1 La face avant	79
IV.5.2.2 Le diagramme.....	84
Conclusion.....	91

Annexe : Le langage de programmation graphique LabVIEW

1 Introduction.....	92
2 Le principe de LabVIEW	93
3 Programmation graphique flux de données	95
4 L'environnement de LabVIEW	96
4.1 La boîte à outils.....	97
4.2 La fenêtre Controls.....	98
4.3 La fenêtre Fonctions.....	99
5 Conclusion.....	100

LISTE DES FIGURES

Figure I.1 : Signal analogique et Signal numérique.....	4
Figure I.2 : Signal continu.....	6
Figure I.3 : signal représenté par une suite de valeurs ponctuelles.....	6
Figure I.4 : signal échantillonné.....	6
Figure I.5 : exemple de fenêtre d'analyse.....	7
Figure I.6 : La chaîne d'acquisition et de restitution de données.....	10
Figure I.7 : La chaîne d'acquisition des signaux EEG.....	11
Figure II.8 : La carte BNC 2090.....	11
Figure II.9 : La carte PCI MOI-16 E 4.....	12
Figure II.1 : les tracés EEG.....	14
Figure II.2 : EEG analogique et EEG numérisé.....	16
Figure II.3 : l'emplacement des électrodes selon le système international 10/20.....	18
Figure II.4 : exemple d'un tracé EEG.....	20
Figure II.5 : topographie de cerveau d'EEG.....	22
Figure II.6 : exemple de l'interpolation en utilisant le polygone de Thiessens.....	24
Figure II.7 : exemple de l'interpolation en utilisant les moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance.....	26
Figure III.1 : Fonctionnement d'un système client/serveur.....	31
Figure III.2 : les couches de modèle OSI.....	32
Figure III.3 : exemple d'un message envoyé selon le protocole TCP.....	36
Figure III.4 : schéma de protocole TCP/IP en mode connecté.....	38
Figure III.5 : schéma de protocole TCP/IP en mode non connecté.....	39
Figure IV.1 : cycle de vie en cascade.....	41
Figure IV.2 : Le diagramme global des cas d'utilisation.....	44
Figure IV.3 : Le diagramme de cas d'utilisation « acquisition et restitution des données ».....	45
Figure IV.4 : Le diagramme de cas d'utilisation « Analyse et traitement des données ».....	45
Figure IV.5 : Le diagramme de cas d'utilisation « Visualisation des signaux ».....	46
Figure IV.6 : Le diagramme de cas d'utilisation « Cartographie en 2D ».....	47
Figure IV.7 : Le diagramme de cas d'utilisation « Stockage des données ».....	47
Figure IV.8 : Le diagramme de cas d'utilisation « Exportation des données ».....	48
Figure IV.9 : Le diagramme de cas d'utilisation « consultation ».....	49
Figure IV.10 : les paquets application et client/serveur.....	49
Figure IV.11 : Le diagramme de classe « application ».....	51
Figure IV.12 : Le diagramme de classe « client-serveur ».....	52
Figure IV.13 : diagramme d'activité globale.....	54
Figure IV.14 : diagramme d'activité pour le cas « Acquisition et restitution des données ».....	55
Figure IV.15 : diagramme d'activité pour le cas « Analyse et traitement de données ».....	57
Figure IV.16 : diagramme d'activité pour le cas « Visualisation des données ».....	58
Figure IV.17 : diagramme d'activité pour le cas « Cartographie en 2D ».....	59
Figure IV.18 : diagramme d'activité pour le cas « Stockage des données ».....	62
Figure IV.19 : diagramme d'activité pour le cas « Exportation des données ».....	63
Figure IV.20 : diagramme d'activité pour le cas « Consultation ».....	64
Figure IV.21 : diagramme de séquence « Acquisition et restitution des données ».....	68
Figure IV.22 : diagramme de séquence « Cartographie en 2D ».....	69
Figure IV.23 : diagramme de séquence « Analyse et traitement des données ».....	71
Figure IV.24 : diagramme de séquence « Stockage des données ».....	73

Figure IV.25 : diagramme de séquence « exportation des données ».....	76
Figure IV.26 : diagramme de composants pour les différents modules du système.....	77
Figure IV.27 : la face avant de l'application.....	80
Figure IV.28 : Le menu «Type de visualisation ».....	80
Figure IV.29 : Les paramètres de réseau de l'application client.....	81
Figure IV.30 : Les paramètres de réseau de l'application serveur.....	81
Figure IV.31 : visualisation individuelle.....	82
Figure IV.32 : comparaison.....	82
Figure IV.33 : tracée EEG.....	82
Figure IV.34 : cartographie en 2D.....	83
Figure IV. 35 : panneau de traitement des signaux.....	83
Figure IV. 36 : Type d'acquisition «lecture».....	84
Figure IV. 37 : Type d'acquisition «simulation».....	85
Figure IV. 38 : Type d'acquisition «par réseau».....	85
Figure IV. 39 : Type d'acquisition «acquisition par carte».....	85
Figure IV. 40 : Visualisation individuelle.....	86
Figure IV. 41 : La cartographie en 2D.....	86
Figure IV.42 : Comparaison.....	87
Figure IV.43 : Tracé EEG.....	87
Figure IV.44 : Le programme de stockage des données.....	88
Figure IV.45 : Le sous VI «file».....	88
Figure IV. 46 : Le partage de données.....	89
Figure IV.47 : Le sous VI pour l'envoi de données.....	89
Figure IV. 48 : Le sous VI «Tcp» pour la création d'une connexion.....	90
Figure IV.49 : Le sous VI pour la conversion en chaîne de caractère.....	90
Figure1 : Fonctions de base proposées par l'environnement LabVIEW.....	93
Figure2: face-avant	94
Figure3: diagramme.....	94
Figure 4 : l'icône	94
Figure 5 : un diagramme Flot de données	96
Figure 6 : encapsulation d'un diagramme flot de données.....	96
Figure 7: face-avant et Diagramme d'un VI	96
Figure 8 : la barre d'exécution/édition	97
Figure 9 : La boîte à outils.....	98
Figure 10 : La fenêtre Controls.....	99
Figure 11 : la fenêtre Fonctions et ses sous-menus	100

Introduction générale

Le développement technique dans le domaine de la mesure électrique à la fin du 19^{ème} siècle a rendu possible un des plus grands triomphes de la neurologie moderne, la découverte faite par un psychiatre allemand appelé Hans Berger, qui a montré que le cerveau humain a une activité électrique continue, et qui peut être enregistrée grâce à l'encéphalogramme (EEG).

L'électroencéphalographie est l'enregistrement graphique, au moyen d'électrodes qui sont placées sur la surface du cuir chevelu, des différences de potentiel électrique produites au niveau de l'écorce cérébrale. Le terme électroencéphalogramme (EEG) désigne quant à lui la courbe qui est obtenue par l'électroencéphalographie.

Depuis le temps de Berger, il est connu que les caractéristiques de l'activité d'EEG changent dans différentes situations, en particulier avec le niveau de la vigilance. La fréquence et les amplitudes des signaux EEG changent, et ils ont été marqués avec des noms tels que alpha, bêta, thêta et delta. Des électroencéphalogrammes sont également employés en neurologie et psychiatrie pour aider principalement à diagnostiquer les maladies du cerveau, telles que l'épilepsie, des perturbations de sommeil et des tumeurs de cerveau.

L'utilisation des ordinateurs pour traiter des signaux de cerveau ouvre un nombre infini de manières d'extraire l'information utile. Une fois que les signaux acquis sont traités, analysés et stockés dans la mémoire d'ordinateur, des applications informatiques peuvent être développées pour visualiser les signaux sur écran et les représenter graphiquement sous forme d'une cartographie de cerveau qui permet de comprendre l'activité du cerveau d'un individu. Cette approche est différente de l'approche traditionnelle, qui ne fait aucune mesure sur les tracés EEG, au lieu de cela, elle se fonde sur l'évaluation qualitative ou l'apparence global des modèles des signaux. Aussi, grâce au développement de l'informatique, le partage des données d'EEG est devenu non seulement possible mais une nécessité car il permet de gagner du temps et de partager une même information entre plusieurs utilisateurs.

Grâce à tous ce qu'on vient de voir on a pensé à notre projet qui à pour finalité principale la cartographie en 2D des signaux de cerveau issus d'un appareil EEG analogique dans le domaine temporel et fréquentiel développée sur une architecture client\serveur.

Le contenu et l'organisation de ce mémoire de fin d'étude sont développés en quatre chapitres, les trois premiers sont dédiés à la partie théorique et le dernier est consacré à la partie pratique.

Le premier chapitre présente quelques notions fondamentales de traitement de signal et l'acquisition de données qui nous sont nécessaires pour réaliser l'acquisition et le traitement des signaux EEG.

Le deuxième chapitre étudie d'une part le fonctionnement de l'encéphalogramme en commençant par son histoire et les domaines d'utilisation. Puis on détermine la différence entre un EEG analogique et un EEG numérique et la nature des activités de cerveau. On passe par la suite aux placements et montages des électrodes. En fin, on arrive à la représentation graphique des résultats d'EEG. D'une autre part on s'intéresse à la cartographie en 2D de cerveau en déterminant ses avantages et ses applications médicales ainsi que les techniques d'interpolation utilisées pour calculer les points non mesurés sur le scalp.

Le troisième chapitre présente le principe de l'architecture client\serveur qui est développée à base de protocole de communication TCP/IP qui permet le partage des données EEG d'une façon fiable.

Dans le dernier chapitre nous abordons pour de bon le développement de notre projet en suivant les étapes décrites par le cycle de vie en cascade et modélisées par le langage de modélisation UML pour terminer par une description de notre application implémentée sous LabVIEW.

Le mémoire est clôturé par une conclusion qui comprend les objectifs atteints par notre travail et les perspectives, suivie d'une annexe qui présente succinctement le logiciel d'instrumentation LabVIEW utilisé dans notre projet.

CHAPITRE I :

Chaîne d'acquisition de données

I.1 Introduction :

Le traitement de signal et l'acquisition de données sont deux disciplines couvrant un très large champ d'applications de mesure, de tests ou de contrôle de procédé. Et notre projet fait partie de ces applications car il s'agit d'acquérir, de traiter, d'analyser et en fin de visualiser les signaux issus d'un EEG analogique. Dans ce chapitre on va donner quelques notions fondamentales de traitement de signal et on passera par la suite à la chaîne de réalisation d'une application d'acquisition de données et on termine par une description du dispositif d'EEG.

I.2 Les différents types de signaux :

I.2.1 Signal analogique:

Le mot signal désigne le résultat de la mesure d'une grandeur physique, le résultat obtenu sera vu comme une fonction du temps. On dira alors que le signal est analogique ou à temps continu.

I.2.2 Signal numérique:

Pour obtenir un signal numérique, on va échantillonner un signal analogique. Les échantillons sont alors transformés, afin de prendre des valeurs discrètes (les valeurs autorisées sont en nombre limitées et situées entre des intervalles de temps égaux). Il y a donc quantification en temps (échantillonnage) et en amplitude. On va alors coder l'amplitude de chaque échantillon transformé en un mot binaire, la suite des mots binaires obtenus va représenter numériquement le signal.

La figure I.1 montre les deux types de signal.

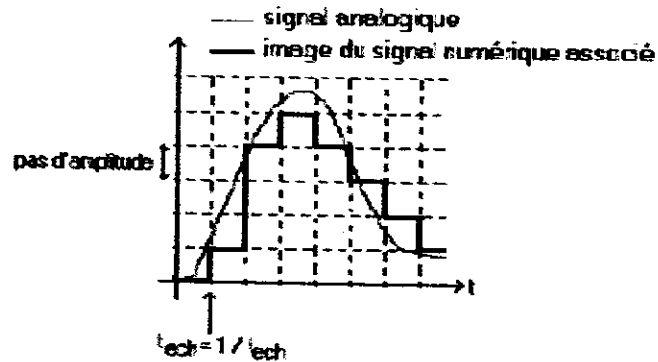


Figure I.1 : Signal analogique et Signal numérique.

I.3 analyse fréquentielle :

Il y a deux domaines importants de description du signal selon la nature de la variable indépendante :

- Le domaine de description temporelle de la forme $x(t)$ dans lequel la variable indépendante est le temps t . Il s'agit de description usuelle des signaux x . Dans ce domaine de représentation, le signal peut être caractérisé par sa durée, sa période fondamentale, ou son amplitude.
- Le domaine des fréquences de la forme $x(f)$ dans lequel la variable indépendante est la fréquence f dont la dimension est l'inverse du temps. Dans ce domaine de représentation le signal peut être caractérisé par sa bande passante, sa fréquence fondamentale, ou sa phase.

Cette dualité temps-fréquence est le fondement de la plupart des méthodes du traitement de signal. Ces deux domaines de description du signal sont reliés entre eux par la transformation de Fourier. Cette transformation de Fourier joue un rôle fondamental en traitement de signal.

Transformée de Fourier rapide (fft) :

Si on connaît une forme analytique du signal, par exemple, si on sait que le signal est une exponentielle, on peut essayer de calculer analytiquement la transformation de Fourier, c'est à dire de faire le calcul avec les formules

mathématiques. Dans le cas contraire, on est forcé de faire un calcul numérique, si possible grâce à un ordinateur.

Un ordinateur ne sait travailler que sur des fonctions discrètes (échantillonnées), c'est à dire sur une liste finie de nombres correspondant aux échantillons du signal. Plusieurs méthodes sont possibles:

- Certains programmes vont faire le calcul en suivant simplement la formule, mais c'est assez lent.
- D'autres programmes vont optimiser le calcul grâce à des algorithmes perfectionnés qui donnent un calcul de TF rapide (ou FFT = Fast Fourier Transform). Le plus courant est l'algorithme de Cooley-Tuckey.

Ceci dit, il n'est en général pas absolument nécessaire de savoir comment la machine procède, en général, on lui donne les valeurs numériques, et on récupère les valeurs calculées en sortie.

I.4 Echantillonnage des signaux :

Lorsqu'on désire numériser un signal analogique, il faut le représenter au préalable par une suite de valeurs numériques ponctuelles, c'est-à-dire le coder à l'aide d'une suite finie de nombres. Un tel prélèvement est appelé échantillonnage.

Cette opération consiste à prendre la valeur instantanée du signal à des instants séparés par un temps constant T_e . Une fois l'opération effectuée, on ne peut plus connaître la valeur du signal à tout instant, mais seulement toutes les T_e secondes.

Afin de garantir la restitution fidèle du signal, le théorème de l'échantillonnage [2] stipule que la fréquence d'échantillonnage doit être supérieure au double de la fréquence maximale à reproduire.

Sinon, on observe un phénomène dit de repliement [2], qui veut que les fréquences les plus élevées, en plus d'être reproduites à leurs justes valeurs, se voient inversées et décalées pour se superposer aux fréquences plus basses du signal.

Soit un signal continu (figure I.2):

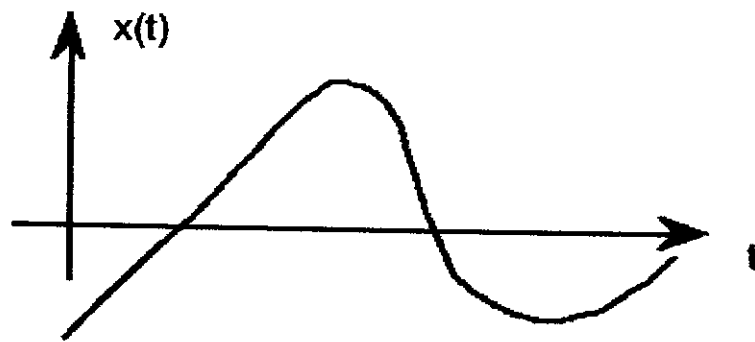


Figure I.2: Signal continu.

Un échantillonnage représente un signal par une suite de valeurs ponctuelles (figure II.3) :

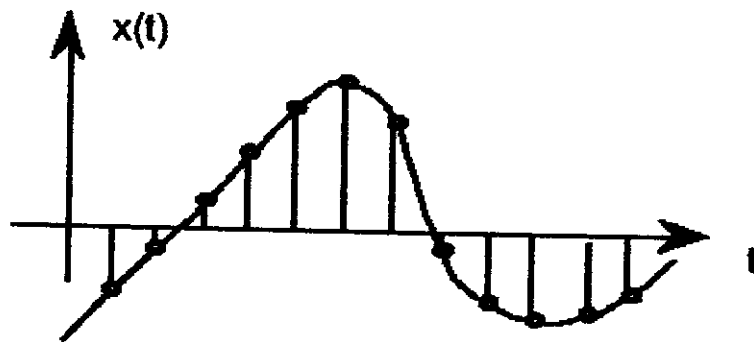


Figure I.3 : signal représenté par une suite de valeurs ponctuelles.

La représentation numérique des échantillons requiert une opération complémentaire de quantification et de codage. L'ensemble réalise une fonction de conversion analogique numérique (figure II.4).

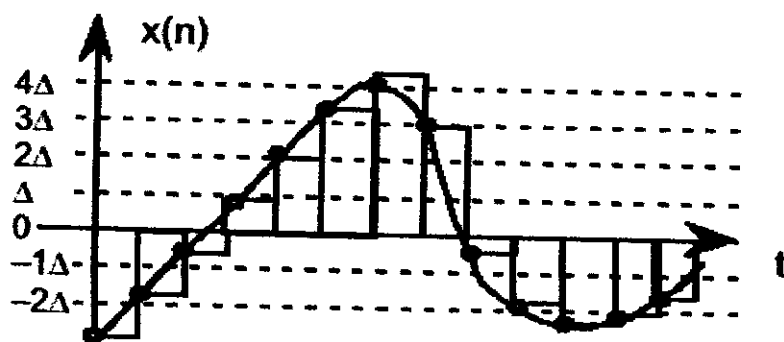


Figure I.4 : signal échantillonné.

I.5 Les fenêtres d'analyse :

Les fenêtres d'analyse ou de pondération jouent un rôle très important dans l'analyse de Fourier. Suivant la forme de spectre que l'on cherche à obtenir, on peut choisir des fenêtres de différentes formes. Pour observer un signal sur une durée finie, on le multiplie par une fonction fenêtre d'analyse. La plus simple est la fenêtre rectangulaire (ou porte), Quelques exemples sont donnés sur la figure suivante.

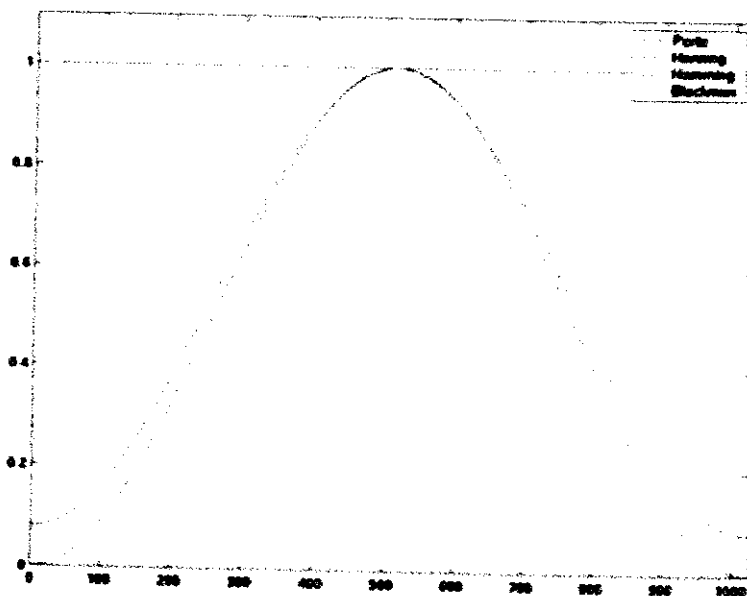


Figure I.5 : exemple de fenêtre d'analyse.

Dans le cas de fenêtres non rectangulaires, le spectre aura des lobes latéraux affaiblis (atténuation du bruit). Cependant, toute atténuation des lobes latéraux entraîne un élargissement du pic centrale et donc une perte de résolution en fréquence. On a donc le choix entre des raies fines mais parfois difficiles à séparer du « bruit » (fenêtre rectangulaires) et des raies nettement séparables mais grossières (fenêtre de Kaiser Bessel, Blackman-Harris).

Les fenêtres de Hanning et de Hamming constituent un cas intermédiaire entre les deux précédents (les secondes sont préférables aux premières pour la résolution en fréquence).

On peut se poser la question : pourquoi ces différentes fenêtres ? Le choix d'une fenêtre se fait en fonction de l'écart en fréquence entre les deux composantes

de fréquence dont on veut mesurer les amplitudes. Suivant le type de phénomène observé, on va réaliser des troncations de formes différentes [2].

- Si on étudie un phénomène transitoire court (dont la durée d'évolution est inférieure à la largeur de la fenêtre), la fenêtre rectangulaire permet de conserver tout le signal sans la moindre altération. Les deux autres types de fenêtres vont, en revanche, conduire à la perte d'une grande partie de l'information.
- Si on étudie un phénomène transitoire long (dont la durée est supérieure à la largeur de la fenêtre), la fenêtre rectangulaire conduit à des discontinuités temporelles et donc une forte altération du spectre, alors que les deux autres types de fenêtres évitent les discontinuités mais altèrent fortement le signal temporel et donc son spectre. Pour éviter ce problème, on préfère utiliser une nouvelle fenêtre, association d'une fenêtre rectangulaire dans la première partie et d'une fenêtre de Hanning dans la seconde.
- Si on étudie un phénomène évoluant continûment (par exemple un phénomène périodique ou un transitoire extrêmement long), il est préférable d'éviter la fenêtre rectangulaire en prenant par exemple, une fenêtre de type Hanning (on peut essayer les autres fenêtres disponibles pour voir la plus intéressante) ce qui évite les discontinuités et leurs conséquences sur le spectre qu'introduit la troncation rectangulaire.

I.6 Les Filtres :

I.6.1 Définition :

Les filtres représentent un outil essentiel en traitement de signal. Un filtre est un circuit électronique qui réalise une opération de traitement du signal. Autrement dit, il atténue certains signaux et en laisse passer d'autres. Un exemple connu du grand public est l'égaliseur audio.

Un filtre modifie certaines parties d'un signal d'entrée dans le domaine temporel et dans le domaine fréquentiel. D'après la théorie de Fourier, tout signal réel peut être considéré comme composé d'une somme de signaux sinusoïdaux (en nombre infini si nécessaire) à des fréquences différentes, le rôle du filtre est de modifier la phase et l'amplitude de ces composantes.

I.6.2 Classification des filtres :

On peut classer les filtres à partir de la forme de leur fonction de transfert. Les filtres les plus courants sont de l'un des quatre types suivants : passe-bas, passe-haut, passe-bande ou réjecteur de bande.

- **Un filtre passe-haut** : ne laisse passer que les fréquences au-dessus d'une fréquence déterminée, appelée fréquence de coupure. Il atténue les autres (basses fréquences). Autrement dit, il laisse passer ce qui est haut. C'est un atténuateur de graves pour un signal audio. On pourrait aussi l'appeler coupe-bas.
- **Un filtre passe-bas** : ne laisse passer que les fréquences au-dessous de sa fréquence de coupure. C'est un atténuateur d'aiguës pour un signal audio. On pourrait l'appeler coupe-haut.
- **Un filtre passe-bande** : ne laisse passer qu'une certaine bande de fréquences (et atténue tout ce qui est au-dessus ou en-dessous). Il est très utilisé dans les récepteurs radio, TV... pour isoler le signal que l'on désire capter.
- **Un filtre réjecteur** : aussi appelé filtre trappe, cloche ou coupe bande, est le complémentaire du passe-bande. Il atténue une plage de fréquences. Cela peut être utile pour diminuer certains parasites par exemple.

I.7 Acquisition de données:

I.7.1 Introduction :

Après avoir étudié les principales techniques de traitement de signal, nous allons nous intéresser à la réalisation d'une application d'acquisition, de traitement et de restitution de données. Les micro-ordinateurs sont actuellement des plates-formes privilégiées pour ce type d'application. Ils offrent en effet une très grande variété d'outils logiciels pour le développement des programmes d'acquisition et de traitement des données.

I.7.2 La chaîne d'acquisition de données :

Le point le plus délicat dans la réalisation d'une application d'acquisition de données est le choix de la solution matérielle et sa mise en œuvre. Le matériel le

plus répandu consiste en une carte d'entrées/sorties numérique et/ou analogique qui vient s'insérer dans le bus d'extension du micro-ordinateur (figure II.6). La sélection de cette carte, point déterminant de l'application, fait apparaître de nombreux critères (nombre de voies d'entrées/sorties, fréquence maximale d'acquisition, précision, etc.).

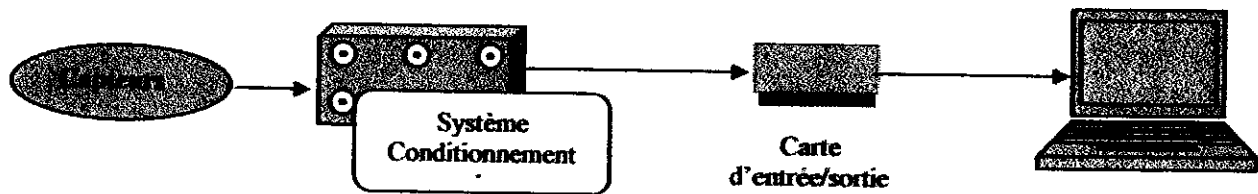


Figure I.6 : La chaîne d'acquisition et de restitution de données

Les principaux composants fonctionnels constituant la chaîne d'acquisition de données sont [2] :

- Les capteurs et les actionneurs pour transformer les grandeurs ou phénomène physiques à mesurer et à contrôler en signaux électriques.
- Le câblage et le conditionnement des signaux.
- Le matériel d'acquisition et de restitution de données.
- Les techniques de traitement des données (logiciel et/ ou matériel).

De façon plus globale, la mise en place d'une application d'acquisition de données doit obligatoirement suivre plusieurs étapes [2] :

- Identification et caractérisation des signaux d'entrées/sorties.
- Sélection du matériel d'acquisition et de restitution de données, avec une partie conditionnement de signaux si nécessaire.
- Réalisation du logiciel de l'application.

I.7.3 Description de dispositif du EEG :

Notre travail se fait sur le dispositif constitué de deux cartes de la société NATIONAL INSTRUMENT (figure II.7). la première carte est la BNC2090, sert

d'interface entre les connecteurs BNC et la deuxième est la PCI-MOI-16 E4, qui permet d'acquérir et de numériser le signal, elle est placée sur bus PCI de PC.

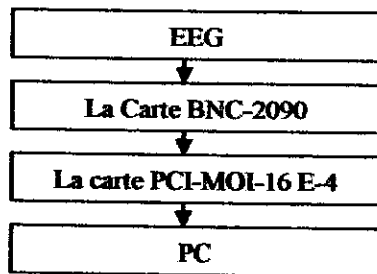


Figure I.7 : la chaîne d'acquisition des signaux EEG.

- La carte BNC 2090 [6]: elle peut être utilisée de deux manières soit comme étant un conditionneur de signaux ou comme un interface entre l'EEG et la carte d'acquisition connectée à un PC. Elle dispose de 16 entrées analogiques différentielles et 8 différentielles et une entrée connecteur à 68 pins (figure II.8).



Figure II.8 : La carte BNC 2090.

- La carte PCI MOI-16 E 4 (figure II.9) [6]: elle possède des spécifications techniques suivantes qui sont nécessaires lors de la réalisation de l'application d'acquisition des signaux EEG :
 1. 16 entrées différentielles et 8 différentielles.
 2. résolution de 12 bits.
 3. fréquence maximale d'échantillonnage par canal est 500 KHz.
 4. le gain possible est 0,5.
 5. domaine de tension possible est entre -11V et +11V.
 6. taille de buffer est 512 échantillons.

7. bus de connexion est PCI.

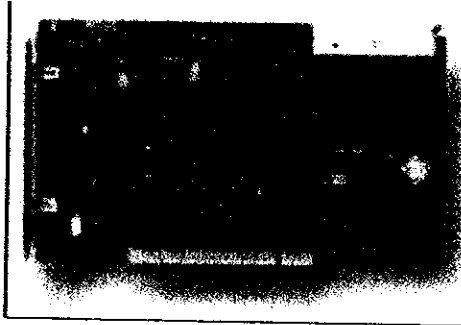


Figure II.9 : La carte PCI MOI-16 E 4.

Conclusion :

Dans ce chapitre nous avons tiré quelques notions fondamentales de traitement numérique de signal y compris la transformation de Fourier qui constitue la base du traitement de signal et sa célèbre transformation de Fourier rapide, poursuivi par un petit passage sur les notions de filtrage, de fenêtrage et d'échantillonnage. Donc, l'objectif principal est de donner une base fondamentale pour la conception de systèmes d'acquisition de données, qui constituent une phase préliminaire de traitement numérique de signal ce qui va nous faciliter la tâche dans le développement de notre système d'acquisition des signaux EEG.

CHAPITRE II :

Electroencéphalogramme

II.1 Introduction :

L'activité électrique du cerveau humain a été mesurée pour la première fois par l'électroencéphalogramme (EEG) qui nous a permis d'avancer considérablement sur notre connaissance du cerveau. Cette technique nous est encore aujourd'hui d'un précieux concours. Au cours des 15 dernières années, de nouvelles techniques de l'imagerie médicale ont fait leur apparition. Mais l'EEG est néanmoins toujours très utilisé en milieu hospitalier et reste indispensable dans bien des cas.

Il nous semble bon, dans un premier temps, de regarder quels aspects physiologiques L'EEG met en jeu, et aussi de présenter ses utilisations, qui ont contribué à l'évolution de la médecine. Nous allons aussi décrire les principes de L'EEG.

Dans un deuxième temps, nous allons voir l'apport de l'ordinateur pour la représentation graphique de l'EEG et plus précisément la topographie de cerveau qui constitue la finalité du projet présenté.

II.2 Historique [5]:

L'existence de l'électricité cérébrale est démontrée en 1875 par Caton. Le premier enregistrement de cette activité a été réalisé sur un cerveau de chien, en 1913, par Prawdicz-Neminsky, et sur l'homme, Hans Berger, professeur de psychiatrie à Iéna en Allemagne, est à l'origine des premiers enregistrements EEG chez l'homme en 1924. Le premier électroencéphalogramme est installé le 23 novembre 1956. Et ce n'est que 1960 que l'EEG a été développé en clinique grâce à l'apparition des amplificateurs différentiels.

II.3 Présentation :

L'électroencéphalographie est l'enregistrement graphique qui correspond à des différences de potentiel électrique (ddp) entre deux électrodes placées sur le scalp (cuire chevelu). Le terme électroencéphalogramme (EEG) désigne quant à lui la courbe qui est obtenu par l'électroencéphalographie.

Les signaux EEG sont de faible amplitude et alternatifs, entrant dans la bande des fréquences basses et très basses (de 0,5 à 60 Hz)

La figure suivante illustre le tracé EEG :

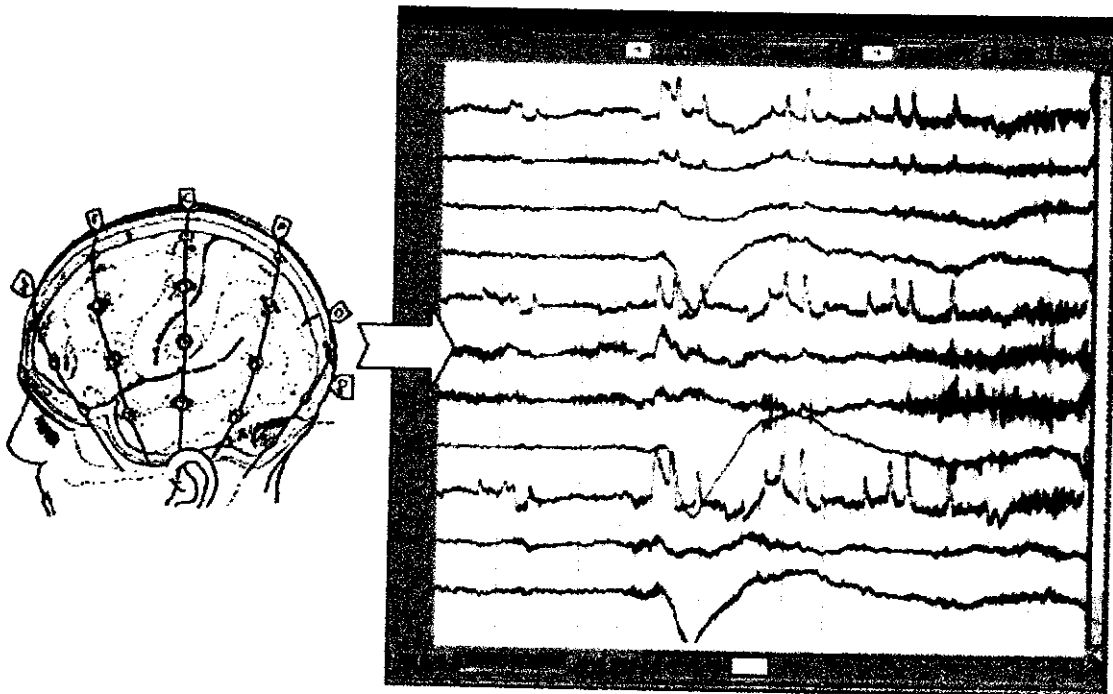


Figure II.1 : les tracés EEG [5].

II.4 Utilisation de l'EEG :

Grâce au EEG il est possible d'obtenir une orientation diagnostique à propos de certains affections neurologiques (maladies de système nerveux) tels que l'épilepsie qui constitue sans doute la pathologie la mieux et la plus étudiée par

l'électroencéphalogramme.

D'autres affections du système nerveux central, de plus en plus rarement, utilisent l'électroencéphalogramme comme moyen d'investigation. Il s'agit des pathologies suivantes [7] :

- Diagnostic de fin de vie (mort) .
- Accident vasculaire cérébrale.
- Tumeur.
- Traumatisme crânien.
- Maladie de Creutzfeldt Jakob.
- Encéphalite et méningo-encéphalite.
- Désorientation temporo-spatiale.
- Troubles de la compréhension.
- Troubles de la mémoire.
- Agitation.
- Démence d'origine infectieuse.

II.5 EEG analogique et EEG numérisé :

L'EEG analogique (ou standard ou conventionnel) est un signal figé sur papier et ne pouvant pas être traité a posteriori. L'exploitation des tracés EEG par le médecin est manuelle et il n'y a pas la possibilité d'interconnecter avec un PC. Par contre, l'EEG numérisé permet :

- Une visualisation sur un écran.
- Un traitement a posteriori des signaux.
- Un stockage et un archivage des données.
- Une édition ultérieure sur papier de seules séquences informatives.
- Une transmission a distance, en temps réel ou différé par télécommunications.

La différence entre l'EEG analogique et EEG numérisé est montrée dans la figure qui suit :

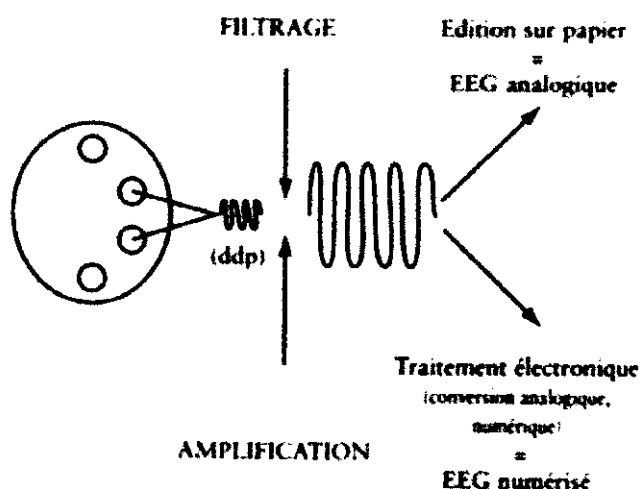


Figure II.2 : EEG analogique et EEG numérisé

II.6 Les activités EEG :

Les activités cérébrales sont caractérisées par leur fréquence et leur amplitude. Elles sont classées en fonction de leur bande de fréquence [5].

Chez l'adulte sain éveillé, l'activité EEG est constituée principalement de 2 rythmes alpha et bêta. On peut noter de manière physiologique des activités thêta et delta, dont l'importance varie en fonction de l'âge et du niveau de vigilance.

- **Alpha** : fréquences comprises en 8 et 13Hz. Elles caractérisent un état de conscience apaisé, et sont principalement émises lorsque le sujet a les yeux fermés.
- **Bêta** : correspond aux fréquences supérieures à 14 Hz (et généralement inférieures à 30 Hz). Elles apparaissent en période d'activité intense, de concentration ou d'anxiété.
- **Delta** : fréquences jusqu'à 5 Hz, normales chez le très jeune enfant, elles peuvent ensuite caractériser certaines lésions cérébrales.
- **Theta** : fréquences entre 4 et 7Hz. On les observe principalement chez l'enfant, l'adolescent et le jeune adulte. Elles caractérisent également certains états de somnolence ou d'hypnose, ainsi que lors de la mémorisation d'information.

Ces ondes sont de très faible amplitude, elles sont de l'ordre du microVolt. Elles ne suivent pas toujours une sinusoïde régulière. Pendant le sommeil, ou lors du diagnostic de l'épilepsie, on observe des pics d'activité et des formes d'ondes complexes.

II.7 Placement des électrodes et montages:

Les électrodes sont de 3 types :

- **L'électrode tampon** : les électrodes tampons sont les plus utilisées, pour des enregistrement dits standards, de l'ordre de l'heure. Mal fixées sur le crâne, elles risquent d'être déplacées en cas de survenue d'une crise épileptique.

- **Les électrodes cupules** : elles sont recommandées pour tout enregistrement de longue durée et lorsque la probabilité d'enregistrement d'une crise épileptique est forte.

- **Les électrodes aiguille** : elle sont réservées aux situations d'urgence, en réanimation, au bloc opératoire.

L'emplacement est standardisé par une nomenclature internationale appelée « system 10/20 ». Chaque électrode a un descripteur alphanumérique qui lui est assigné. Chaque descripteur se compose d'une lettre qui se rapporte à la zone du cerveau : le « O » pour Occipal, « C » pour la bande Centrale, « T » pour Temporale et « F » pour Frontale ; alors que le composant numérique indique l'orientation de devant à l'arrière et la médiane indiquant à l'extérieur une valeur croissante, les chiffres pairs indiquent le côté droit et les chiffres impairs pour le côté gauche. La figure II.3 nous donne une idée sur l'emplacement des électrodes selon le système international 10/20.

Un montage EEG correspond à la combinaison de couple d'électrodes :

- Si les deux électrodes sont actives, l'une reliée à l'entrée positive, l'autre à l'entrée négative, le montage est dit bipolaire. La ddp correspond à la somme algébrique des signaux recueillis sous chaque électrode active.

- Si l'une des deux électrodes est active (reliée au pôle négatif) l'autre neutre, (reliée au pôle positif) le montage est référentiel (et non monopolaire car l'électrode neutre

n'est véritablement nulle). Le ddp correspond à la valeur absolue du signal recueilli sous la seule électrode active.

L'intérêt des deux types de montages est complémentaire.

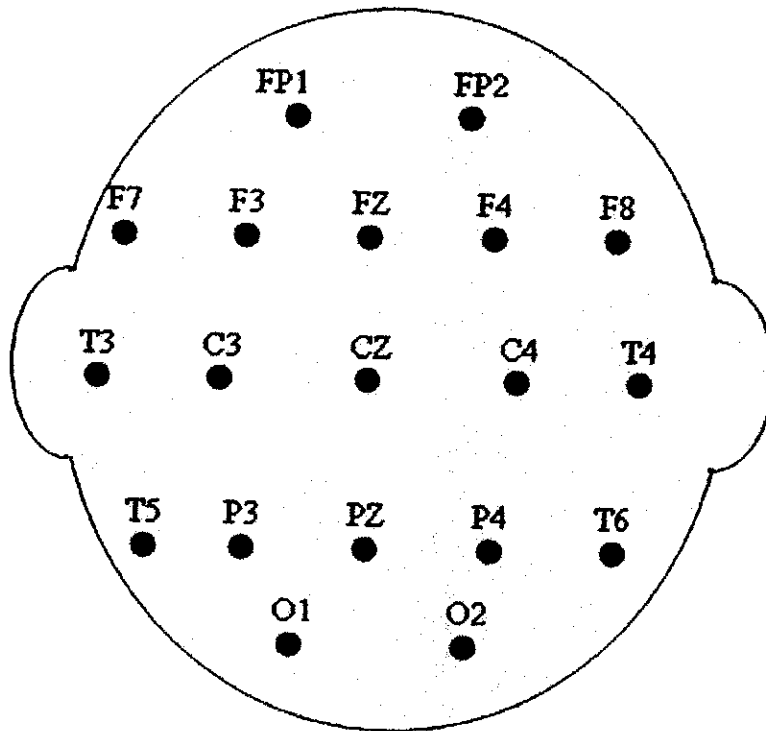


Figure II.3 : l'emplacement des électrodes selon le système international 10/20.

II.8 Représentation graphique d'EEG :

Il y a eu dans ces dernières années un très grand développement en ce qui concerne la représentation graphique de l'EEG, puisque il était auparavant limité au traçage de ce dernier sur papier, ce qui est le cas dans la plupart de nos hôpitaux. Les nouvelles techniques de visualisation de ces signaux ont vu l'introduction de l'ordinateur comme un moyen d'ouverture des possibilités pour extraire des informations utiles et pour travailler plus aisément.

Le signal EEG doit subir un traitement adéquat pour pouvoir le visualiser sur l'écran, les éléments essentiels d'un système de cartographie d'EEG ont les rôles suivants :

1. Acquisition des signaux EEG.
2. Traitement des données.
3. Stockage des informations.
4. Représentation des résultats.

II.8.1 Amplification :

Les signaux EEG sont de faible amplitudes ce qui demande une amplification qui sera suivi d'un filtrage qui permet d'adapter la bande passante du signal à la fréquence d'échantillonnage du convertisseur.

II.8.2 Traitement des données :

Il existe plusieurs techniques pour traiter les données, dans notre cas on utilise la transformation de Fourier rapide (fft).

II.8.3 Représentation des résultats :

Une fois l'acquisition est faite, les données seront stockées dans un fichier. En utilisant des algorithmes adéquats on peut visualiser les signaux sur écran.

Le développement de la représentation des signaux EEG ne cesse d'évoluer, grâce à l'utilisation des ordinateurs pour traiter des signaux de cerveau qui ouvre un nombre infini de manières d'extraire l'information utile. Une fois que les signaux EEG sont stockés dans la mémoire d'ordinateur, des techniques mathématiques puissantes peuvent être développées pour démêler la signification de ses signaux.

II.9 Cartographie de cerveau d'EEG :

La cartographie (ou la topographie) de cerveau d'EEG est une technique qui consiste à former une image de cerveau qui nous permet de comprendre l'activité et la fonction électrique du cerveau d'un individu.

II.9.1 Les inconvénients de tracé EEG:

Une application importante d'EEG est d'essayer de trouver l'endroit d'un foyer épileptique (une petite tache dans le cerveau où l'activité anormale commence et puis s'écarte à d'autres parties du cerveau) ou d'une tumeur, même lorsqu'elles ne sont pas évidentes.

Chaque traçage horizontal correspond à une paire d'électrode placée sur un secteur particulier du cuir chevelu du patient, formant un régulier grille comme le montre la figure suivante.

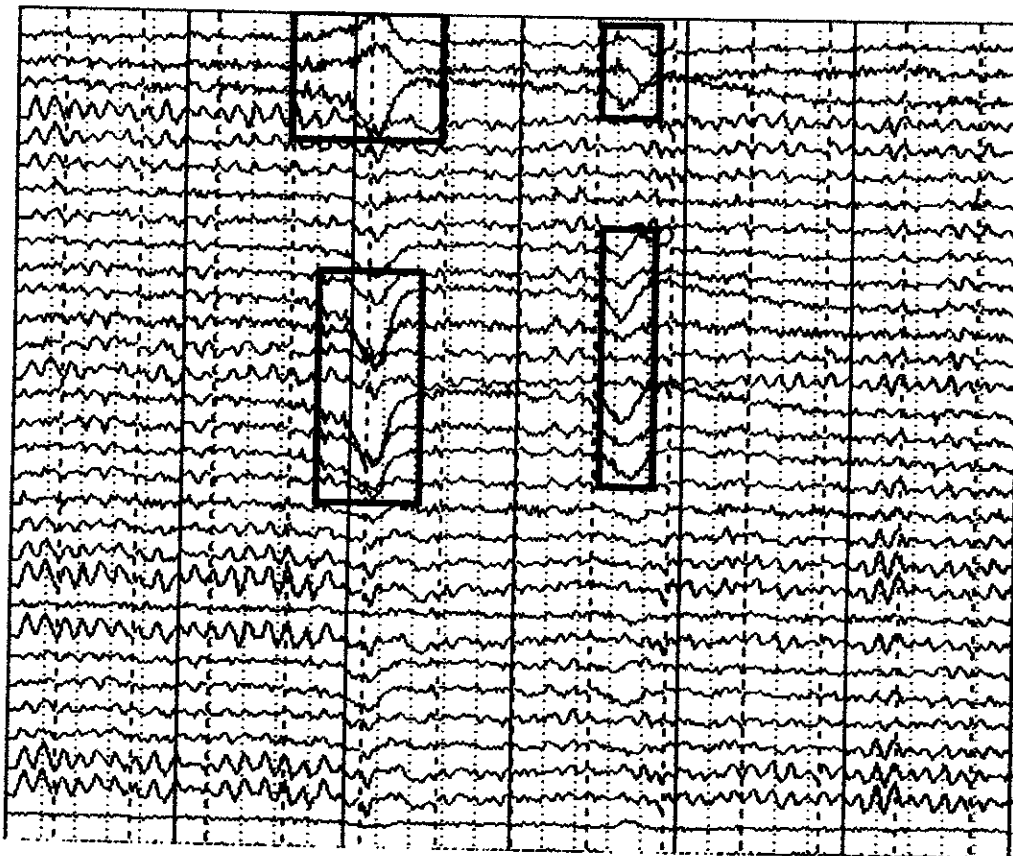


Figure II.4 : exemple d'un tracé EEG [5].

En notant l'ensemble de canaux où l'événement anormal de signaux (comme ceux marqués dans le rouge), le neurologue peut impliquer les parties du cerveau où l'anomalie est localisée. Il est très difficile de balayer et d'interpréter tout le tracé, et ceci est sujet de beaucoup d'erreurs, cependant, quand le nombre de canaux

anormaux est grand, ou la nature des changements est complexe. Un endroit précis d'un foyer ou d'une tumeur est impossible à localiser.

Ainsi, quelle est la solution ? L'ordinateur, naturellement.

II.9.2 L'ordinateur et le cerveau :

Afin d'employer la puissance et la flexibilité supérieures de l'ordinateur pour stocker et analyser les signaux EEG, une invention était d'importance fondamentale : le convertisseur analog-digital (ou DAC). Essentiellement, c'est un dispositif électronique qui prend un signal continu et le transforme en échantillons chaque échantillon étant une mesure de l'amplitude du signal à des intervalles réguliers de temps qui sera codé en binaire, ainsi le processus entier est également appelé prélèvement. Les résultats sont stockés sur le disque de l'ordinateur. Chaque canal d'EEG a son propre processus, parallèlement aux autres, et ceci procède en temps réel.

Une fois que tous les échantillons qui ont été enregistrés le long d'une période sont à l'intérieur de l'ordinateur, des programmes spéciaux sont employés pour montrer les signaux dans l'écran visuel, les imprimer, etc. des techniques mathématiques sont employées pour faire le filtrage, l'analyse de fréquence et d'amplitude et la topographie de cerveau. Cette approche s'appelle l'EEG quantitatif, parce qu'elle est différente de l'approche traditionnelle, qui ne fait aucune mesure sur les tracés ; au lieu de cela, elle se fonde sur l'évaluation qualitative ou l'apparence global des modèles des signaux.

II.9.3 Cartographies en 2D de cerveau:

La topographie d'EEG est une technique dans laquelle un grand nombre d'électrodes d'EEG sont placées sur la tête, suivant un choix géométrique. Un logiciel spécial à l'intérieur de l'ordinateur, montre l'activité cérébrale sur un écran de couleur, en codant la quantité d'activité dans plusieurs tonalités de couleur (par exemple, le noir et le bleu pourraient dépeindre les basses amplitudes d'EEG, alors que le jaune et le rouge pourraient dépeindre les plus grandes amplitudes). Les points spatiaux se trouvant entre les électrodes sont calculés par des techniques mathématiques de l'interpolation, et une gradation douce de couleurs est réalisée ainsi. (Figure II.5)

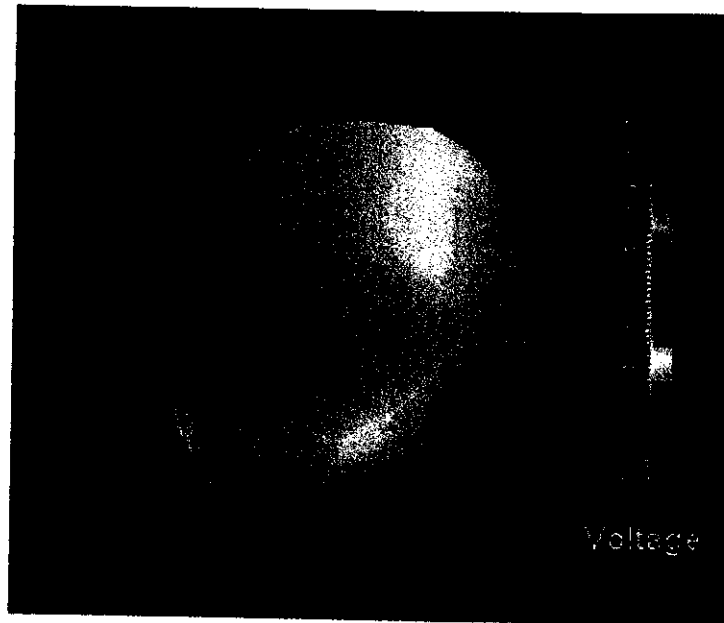


Figure II.5 topographie de cerveau d'EEG [5].

Cette approche donne un avis beaucoup plus précis et plus représentatif de l'endroit des changements de rythme, d'amplitude, etc, par rapport à la surface du crâne. Les neurologues travaillant avec le système topographique de cerveau d'EEG peuvent différencier plusieurs genres de diagnostics (quelques maladies mentales). Indiquer l'endroit exact des changements d'EEG a été également facilité. En outre, l'utilisation des animations de plusieurs photos séquentielles prises des cartographies de cerveau a rendu possible l'étude dynamique de la fonction de cerveau.

II.9.4 Techniques d'interpolation pour la cartographie d'EEG :

Le recueil des potentiels se fait en un nombre limité de points sur le scalp correspondant aux électrodes. Cependant, en chaque point du scalp il existe un potentiel électrique qui varie au cours du temps. Par des techniques d'interpolation, il est possible de reconstituer la distribution sur tout le scalp à partir des potentiels enregistrés sur chaque électrode. La distribution du potentiel sur l'ensemble de la surface du scalp peut être interpolée, à chaque instant. La distribution interpolée étant d'autant plus proche de la distribution réelle du potentiel que le nombre d'électrodes est élevé. De telles cartographies représentant la distribution des

potentiels sur toute la surface du scalp permettent de suivre, avec une précision, l'évolution temporelle des réponses évoquées.

II.9.4.1 L'interpolation :

L'interpolation est la procédure qui consiste à estimer la valeur d'attribut pour des sites inconnus situés à l'intérieur des limites définies par les positions de sites connus. L'interpolation repose sur un principe important appelé, l'autocorrélation spatiale. Selon ce principe, des objets rapprochés dans l'espace tendent à posséder des caractéristiques similaires. L'interpolation est très utile dans tous les cas où il n'a pas été possible de connaître tous les endroits pour lesquels des valeurs doivent être connues.

II.9.4.2 Classification des méthodes d'interpolation :

Les méthodes d'interpolation sont classifiées selon qu'elles sont :

- Discrètes ou continues
- Globales ou locales
- Exactes ou approximatives

Discrète : Ce type de méthode repose sur le principe que les variations importantes d'un phénomène se produisent aux frontières. Ce phénomène est supposé homogène et isotrope.

Continue : Ce type de méthode repose sur le principe que les variations importantes d'un phénomène se produisent de manière graduelle pouvant être estimées par un modèle mathématique.

Globale : Toutes les observations disponibles sont utilisées par l'interpolation.

Locale : Les valeurs inconnues sont estimées à partir d'un voisinage de points connus.

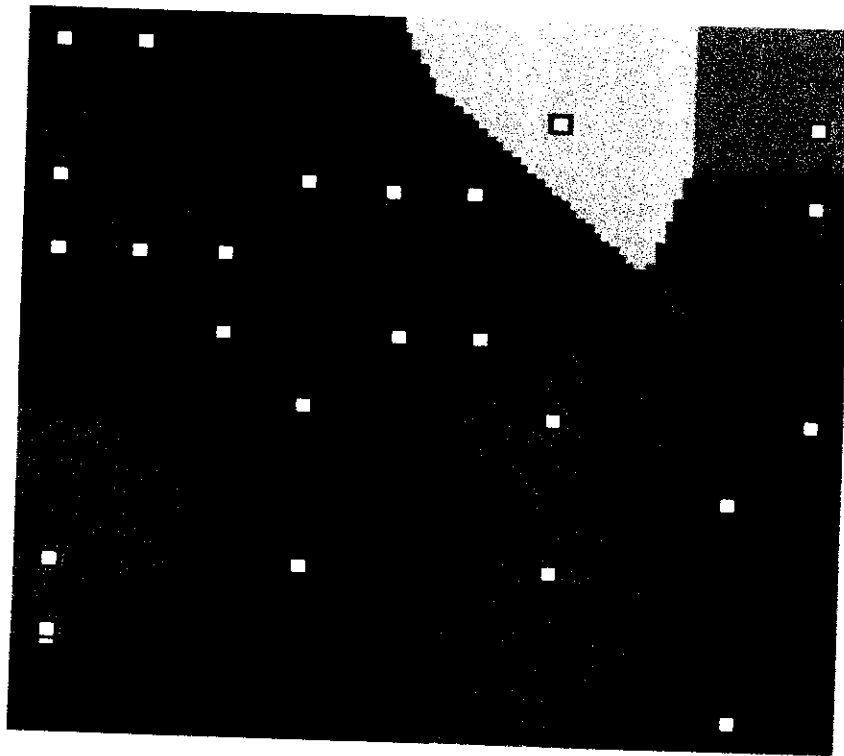
Exacte : Les valeurs originales sont préservées.

Approximative : Les valeurs originales ne sont pas préservées.

Pour la représentation cartographique d'EEG on a opté pour deux méthodes d'interpolation, l'une est le polygone de Thiessens utilisé pour la représentation fréquentielle et l'autre est moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance pour la représentation temporelle.

II.9.4.2.1 Méthode discrète « Polygones de Thiessens » :

Polygones de Thiessens est une méthode locale et exacte. Aussi connue sous le nom de cellules de Dirichlet et de polygones de Voronoï. Les surfaces sont calculées par la méthode du plus proche voisin. La valeur de chaque cellule de la surface est égale à la valeur de la donnée située le plus près. La taille et la forme des polygones ne dépendront donc que de la distribution des points calculés sur la surface (figureII.6).



FigureII.6 : exemple de l'interpolation en utilisant le polygone de Thiessens.

Avantages:

- Méthode rapide d'exécution.
- Méthode simple et très utile.
- Permet de conserver les données originales.

Inconvénients:

- La dimension et la forme des polygones dépendent de la localisation des points d'échantillonnage.
- Les polygones peuvent ne pas correspondre à la distribution spatiale du phénomène d'intérêt.

- La valeur de chaque polygone est estimée à partir d'un seul point, ce qui ne permet pas d'estimer l'erreur.

II.9.4.2.2 Méthode continue « Moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance » :

C'est une méthode locale approximative dans laquelle, Pour chaque point (cellule) de la surface on recherche les N plus proches voisins, puis on calcule la moyenne de leurs attributs en pondérant chacune des valeurs impliquées par l'inverse de la distance élevée à la puissance P, on attribue par la suite cette valeur moyenne au point en question, en fin on passe au point suivant et on recommence l'opération (figure II.7).

L'exposant P généralement utilisé est 2.

On utilise donc l'équation:

$$Z(x_j) = \frac{\sum_{i=1}^n Z(x_i) \cdot d_{ij}^{-2}}{\sum_{i=1}^n d_{ij}^{-2}}$$

où :

$Z(x_i)$ sont les valeurs de Z utilisées pour calculer la valeur manquante de Z.

d_{ij} est la distance séparant une cellule échantillonnée de la cellule pour laquelle on veut interpoler une valeur de Z.

n est le nombre de cellules utilisées pour calculer notre valeur Z manquante.

Influence de la valeur de l'exposant

- Lorsque P, l'exposant ou la puissance, approche de 0, la décroissance de l'influence d'un point tend vers une fonction linéaire.
- Plus P est grand, plus la décroissance augmente rapidement

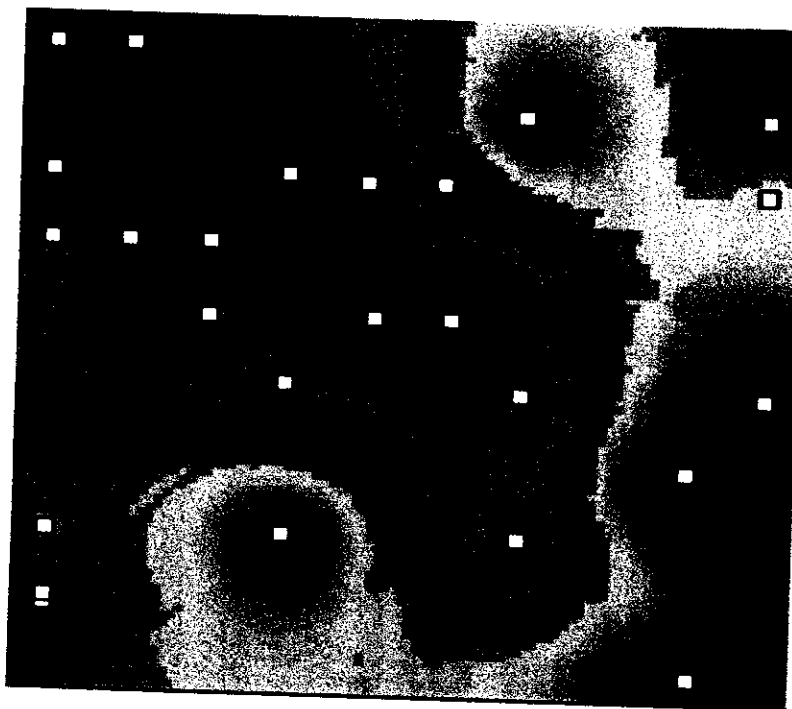


Figure II.7 : exemple de l'interpolation en utilisant les moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance.

Avantages:

- Méthode facile à utiliser et à comprendre.
- Représente bien les discontinuités dans les données.

Inconvénient:

- Méthode approximative qui modifie les valeurs utilisées pour l'interpolation.
- Choix arbitraire des paramètres taille du voisinage et exposant.

II.9.5 Applications médicales:

La topographie de cerveau d'EEG n'est pas exécutée dans tous les cas exigeant un enregistrement de l'activité de cerveau. Au départ, son indication principale était de déterminer la présence des tumeurs et quelques maladies du cerveau, mais aujourd'hui d'autres technologies de formation d'image ont remplacé l'EEG pour décrire des changements de structure de cerveau. Il y a plusieurs procédures neurochirurgicales qui se fondent sur EEG topographique pour localiser des problèmes électro-fonctionnels, tels que l'épilepsie, avant et pendant la chirurgie. Il est également approprié quand les perturbations dans la conscience et la vigilance sont présentes, comme narcolepsie (le début brusque du sommeil), le coma, etc. en

outre, la topographie de cerveau d'EEG est de plus en plus employée pour surveiller les effets du retrait des drogues psycho actives, et dans les maladies infectieuses du cerveau, aussi bien qu'aux suivi des patients qui ont soumis aux opérations de cerveau. En psychiatrie, la topographie de cerveau d'EEG a été de valeur en identifiant des désordres d'origine biologique, tels que la schizophrénie, des démences, hyperactivité et dépression, atrophie de cerveau et des désordres de déficit d'attention chez les enfants.

Aujourd'hui, il y a beaucoup de systèmes commerciaux de topographie de cerveau d'EEG en service. Ils sont généralement installés sous Windows ou les plateformes Macintosh, et peuvent être facilement actionnés par des techniciens ou des médecins. Le logiciel utilisé est fortement flexible, permettant la programmation de beaucoup de configurations et paramètres d'enregistrement, aussi bien que pour établir une base de données de référence des images. En outre, plusieurs cartes prises dans différents instants peuvent être montrées côte à côte [5].

II.10 Conclusion :

Du fait de l'évolution des ordinateurs vers le multimédia et le réseau, surtout dans le domaine médicale, une représentation graphique des signaux EEG analogique sous réseaux est devenu non seulement possible mais nécessaire pour avoir un enregistrement numérique des données sur des écrans plutôt que sur des rouleaux de papier.

CHAPITRE III:

Architecteur client /serveur

III.1 Introduction :

Les réseaux locaux sont nés au début des années 80 avec l'apparition des micro-ordinateurs. Le but de tel réseau est de faciliter l'accès à des périphériques partagés (imprimants, mémoires, etc.) et permet aussi le partage des applications informatiques. Le modèle client/serveur est l'une des architectures de réseau local la plus utilisée et la plus recommandée grâce à ses nombreux avantages.

Dans ce chapitre nous allons aborder, en plus de l'architecture client/serveur, la notion d'intranet, l'architecture des réseaux y compris le protocole TCP/IP et les sockets.

III.2 La notion d'intranet :

III.2.1 Principe :

Un intranet est un ensemble de services Internet (par exemple un serveur web) internes à un réseau local, c'est-à-dire accessibles uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisible de l'extérieur. Il consiste à utiliser les standards client-serveur de l'Internet (en utilisant les protocoles TCP/IP), comme par exemple l'utilisation de navigateurs Internet (client basé sur le protocoles HTTP) et des serveurs web (protocole HTTP), pour réaliser un système d'information interne à une organisation ou une entreprise.

Un intranet repose généralement sur une architecture à trois niveaux, composée:

- de clients (navigateur Internet généralement).
- d'un ou plusieurs serveurs d'application.

- d'un serveur de bases de données.

De cette façon les machines clientes gèrent l'interface graphique, tandis que les différents serveurs manipulent les données. Le réseau permet de véhiculer les requêtes et les réponses entre clients et serveurs.

Un intranet possède naturellement plusieurs clients (les ordinateurs du réseau local) et peut aussi être composé de plusieurs serveurs. Une grande entreprise peut par exemple posséder un serveur web pour chaque service afin de fournir un intranet composé d'un serveur web fédérateur liant les différents serveurs gérés par chaque service.

III.2.2 L'utilité d'un intranet :

Un intranet dans une entreprise permet de mettre facilement à la disposition des employés des documents divers et variés; cela permet d'avoir un accès centralisé et cohérent à la mémoire de l'entreprise. De cette façon, il est généralement nécessaire de définir des droits d'accès pour les utilisateurs de l'intranet aux documents présents sur celui-ci, et par conséquent une authentification de ceux-ci afin de leur permettre un accès personnalisé à certains documents.

Des documents de tous types (textes, images, vidéos, sons, ...) peuvent être mis à disposition sur un intranet. De plus, un intranet peut réaliser une fonction de groupware très intéressante, c'est-à-dire permettre un travail coopératif. Voici quelques unes des fonctions qu'un intranet peut réaliser:

- Mise à disposition d'informations sur l'entreprise (panneau d'affichage)
- Mise à disposition de documents techniques
- Moteur de recherche de documentations
- Un échange de données entre collaborateurs
- Annuaire du personnel
- Gestion de projets, aide à la décision, agenda, ingénierie assistée par ordinateur
- Messagerie électronique
- Forums de discussion, listes de diffusions, chat en direct

- Portail vers Internet

De cette façon un intranet favorise la communication au sein de l'entreprise et limite les erreurs dues à la mauvaise circulation d'une information. L'information disponible sur l'intranet doit être mise à jour en évitant les conflits de version.

III.3 Architecture client/serveur :

III.3.1 Présentation :

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. Lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur.

Dans un environnement purement Client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est un des principaux atouts de ce modèle.

III.3.2 Avantages de l'architecture client/serveur :

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont:

- **Des ressources centralisées:** étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- **Une meilleure sécurité:** car le nombre de points d'entrée permettant l'accès aux données est moins important

- Une administration au niveau serveur: les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- Un réseau évolutif: grâce à cette architecture ont peu supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

III.3.3 Fonctionnement d'un système client/serveur :

Un système client/serveur fonctionne selon le schéma suivant:

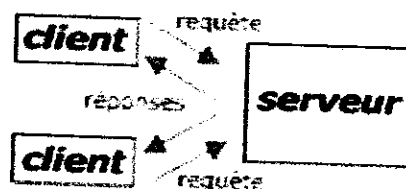


Figure III.1 : Fonctionnement d'un système client/serveur

- Le client émet une requête vers le serveur grâce à son adresse et le port, qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.

III.4 Les protocoles :

Le but des réseaux est de faire communiquer plusieurs ordinateurs ensemble. Si les hommes communiquent entre eux grâce aux différentes langues, les ordinateurs utilisent différents protocoles.

Les communications sont souvent internationales, et comme pour les hommes, il n'existe pas de protocole universel. Certains sont plus utilisés que d'autres, il en existe cependant un très grand nombre, chacun cherchant à imposer sa propre norme.

Un protocole est une description formelle de règles et de conventions à suivre dans un échange d'informations, que ce soit pour acheminer les données jusqu'au

destinataire ou pour que le destinataire comprenne comment il doit utiliser les données qu'il a reçues.

Pour s'y retrouver plus facilement dans l'ensemble des protocoles, l'International Standard Organisation (ISO) a défini un modèle de base appelé modèle OSI. Ce modèle définit 7 niveaux différents pour le transfert de données. Ces niveaux sont également appelés couches.

La figure suivant montre l'organisation de ces couches :

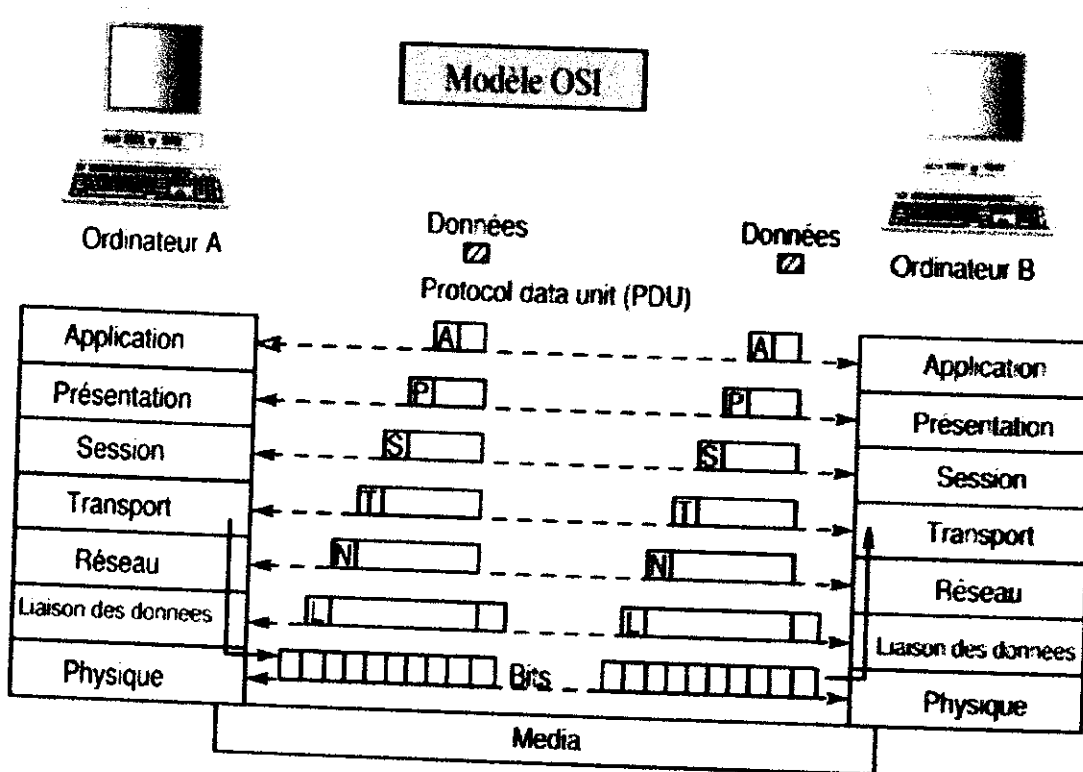


Figure III.2 : les couches de modèle OSI

- Le septième niveau, la couche Application, gère le transfert des informations entre programmes.

- Le sixième niveau, la couche Présentation, s'occupe de la mise en forme des textes et des conventions d'affichage.

- Le cinquième niveau, la couche Session, s'occupe de l'établissement, de la gestion et coordination des communications.
- Le quatrième niveau, la couche Transport, gère la remise correcte des informations.
- Vient ensuite le niveau trois, la couche Réseau, qui détermine les routes de transport et qui s'occupe du traitement et du transfert de messages.
- Le niveau deux, la couche Liaison de données, s'occupe du codage, de l'adressage, et de la transmission des informations.
- Le premier niveau, la couche physique, gère les connections matérielles.

A chacun de ces niveaux, on encapsule un en-tête et une fin de trame qui comporte les informations nécessaires en suivant les règles définies par le protocole utilisé.

Chaque couche N d'une machine de réseau gère la conversation (communication) avec la couche N d'une autre machine de ce même réseau.

Cette conversation est caractérisée par un certain nombre de règles et conventions qui sont connus sous le nom de protocole (ou protocole de la couche N).

En réalité, les données de la couche N d'une machine ne sont pas directement transférées à la couche N d'une autre machine, mais chaque couche passe les données et le contrôle à la couche immédiatement inférieure, jusqu'à la plus basse, et cette dernière couche est appelée la couche physique qui est responsable réellement de la communication.

III.4.1 Le protocole TCP/IP :

TCP/IP est une suite de protocoles. Le sigle TCP/IP signifie «Transmission Control Protocol/Internet Protocol». Il provient des noms des deux protocoles majeurs de la suite de protocoles, (c'est-à-dire les protocoles TCP et IP).

TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur Internet et se base sur la notion adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données.

III.4.1.1 L'adresse IP :

IP signifie Internet Protocol : littéralement "le protocole d'Internet". C'est le principal protocole utilisé sur Internet. Le protocole IP permet aux ordinateurs reliés à ces réseaux de dialoguer entre eux. L'adresse IP est une adresse unique attribuée à chaque ordinateur sur Internet. Chaque message (chaque petit paquet de données) est enveloppé par IP qui contient les différentes informations :

- l'adresse de l'expéditeur (votre adresse IP),
- l'adresse IP du destinataire,
- différentes données supplémentaires (qui permettent de bien contrôler l'acheminement du message).

L'adresse IP se présente le plus souvent sous forme de 4 nombres (entre 0 et 255) séparés par des points. Par exemple: 204.35.129.3.

III.4.1.1.1 Le routage IP :

Pour envoyer un message sur Internet vous déposez le paquet IP sur l'ordinateur le plus proche (celui de votre fournisseur d'accès en général). Le paquet IP va transiter d'ordinateur en ordinateur jusqu'à atteindre le destinataire.

III.4.1.1.2 Les ports :

Avec IP, nous avons de quoi envoyer et recevoir des paquets de données d'un ordinateur à l'autre. Mais que nous ayons plusieurs programmes qui fonctionnent en même temps sur le même ordinateur: un navigateur, un logiciel d'email et un logiciel pour écouter la radio sur Internet. Si l'ordinateur reçoit un paquet IP, comment savoir à quel logiciel donner ce paquet IP ?

L'adresse IP permet de s'adresser à un ordinateur donné, et le numéro de port permet de s'adresser à un logiciel particulier sur cet ordinateur.

III.4.1.2 UDP :

UDP/IP est un protocole qui permet justement d'utiliser des numéros de ports en plus des adresses IP (On l'appelle UDP/IP car il fonctionne au dessus d'IP). IP s'occupe des adresses IP et UDP s'occupe des ports.

Avec UDP/IP, on peut être plus précis: on envoie des données d'une application x sur l'ordinateur A vers une application y sur l'ordinateur B.

- Chaque couche (UDP et IP) va ajouter ses informations. Les informations de IP vont permettre d'acheminer le paquet à destination du bon ordinateur. Une fois arrivé à l'ordinateur en question, la couche UDP va délivrer le paquet au bon logiciel.
- Les deux logiciels se contentent d'émettre et de recevoir des données. Les couches UDP et IP en dessous s'occupent de tout.

III.4.1.3 TCP :

Le protocole TCP est défini dans le but de fournir un service de transfert de données de haute fiabilité entre deux ordinateurs "maîtres" raccordés sur un réseau de type "paquets commutés", et sur tout système résultant de l'interconnexion de ce type de réseaux.

TCP est capable:

- de faire tout ce que UDP sait faire (ports).
- de vérifier que le destinataire est prêt à recevoir les données.
- de découper les gros paquets de données en paquets plus petits pour que IP les accepte
- de numéroter les paquets, et à la réception de vérifier qu'ils sont tous bien arrivés, de redemander les paquets manquants et de les rassembler avant de les donner aux logiciels.

Pour envoyer le message "Salut, comment ça va ?", la figure III 3 montre ce que fait TCP (Chaque flèche représente 1 paquet IP):

A l'arrivée, sur l'ordinateur 204.66.224.82, la couche TCP reconstitue le message "Salut, comment ça va ?" à partir des 3 paquets IP reçus et le donne au logiciel qui est sur le port 80.

Avec TCP/IP, on peut maintenant communiquer de façon fiable entre logiciels situés sur des ordinateurs différents.

ordinateur 199.7.55.3

ordinateur 204.66.224.82

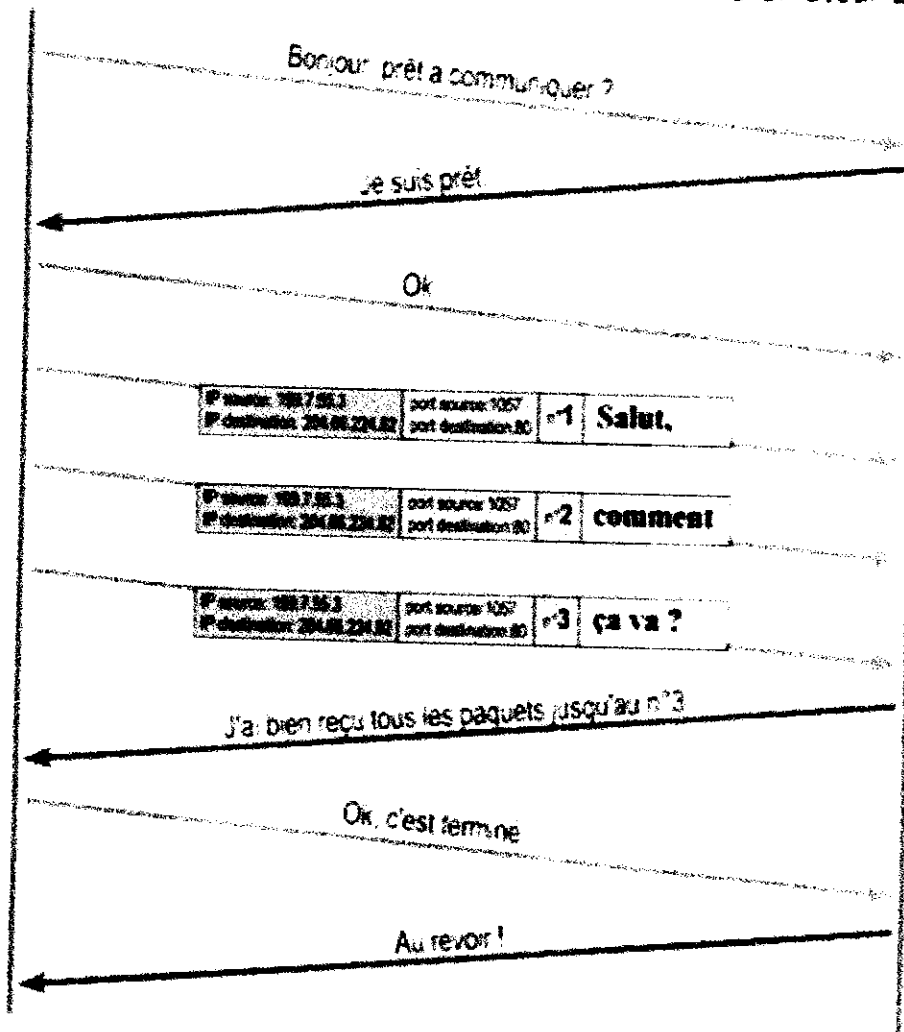


Figure III.3 : exemple d'un message envoyé selon le protocole TCP.

TCP/IP est utilisé pour des tas de choses:

- Dans votre navigateur, le protocole HTTP utilise le protocole TCP/IP pour envoyer et recevoir des pages HTML, des images GIF, JPG et toutes sortes d'autres données.
- FTP est un protocole qui permet d'envoyer et recevoir des fichiers. Il utilise également TCP/IP.

- Votre logiciel de courrier électronique utilise les protocoles SMTP et POP3 pour envoyer et recevoir des emails. SMTP et POP3 utilisent eux aussi TCP/IP.
- Votre navigateur (et d'autres logiciels) utilise le protocole DNS pour trouver l'adresse IP d'un ordinateur à partir de son nom (par exemple, de trouver 216.32.74.52 à partir de 'www.yahoo.com'). Le protocole DNS utilise UDP/IP et TCP/IP en fonction de ses besoins.

III.5 Les sockets :

Il s'agit d'un modèle permettant la communication inter processus afin de permettre à divers processus de communiquer aussi bien sur une même machine qu'à travers un réseau TCP/IP.

La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication:

- Le mode connecté (comparable à une communication téléphonique), utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.

Le schéma de protocole TCP/IP en mode connecté est illustré dans la figure III.4 :

- Le mode non connecté (analogue à une communication par courrier), utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné.

Le schéma de protocole TCP/IP en mode non connecté est illustré dans la figure III.5 :

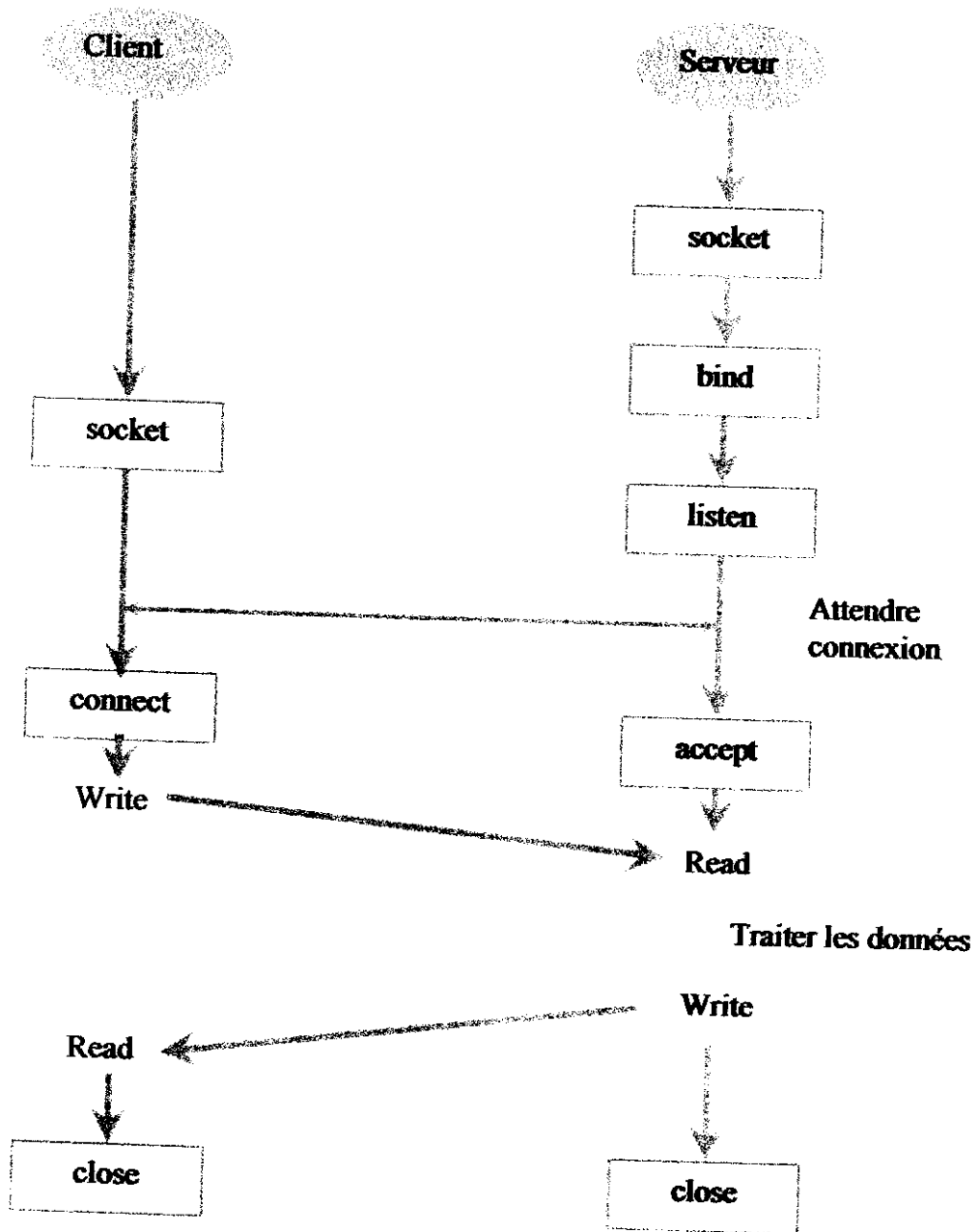


Figure III.4 : schéma de protocole TCP/IP en mode connecté.

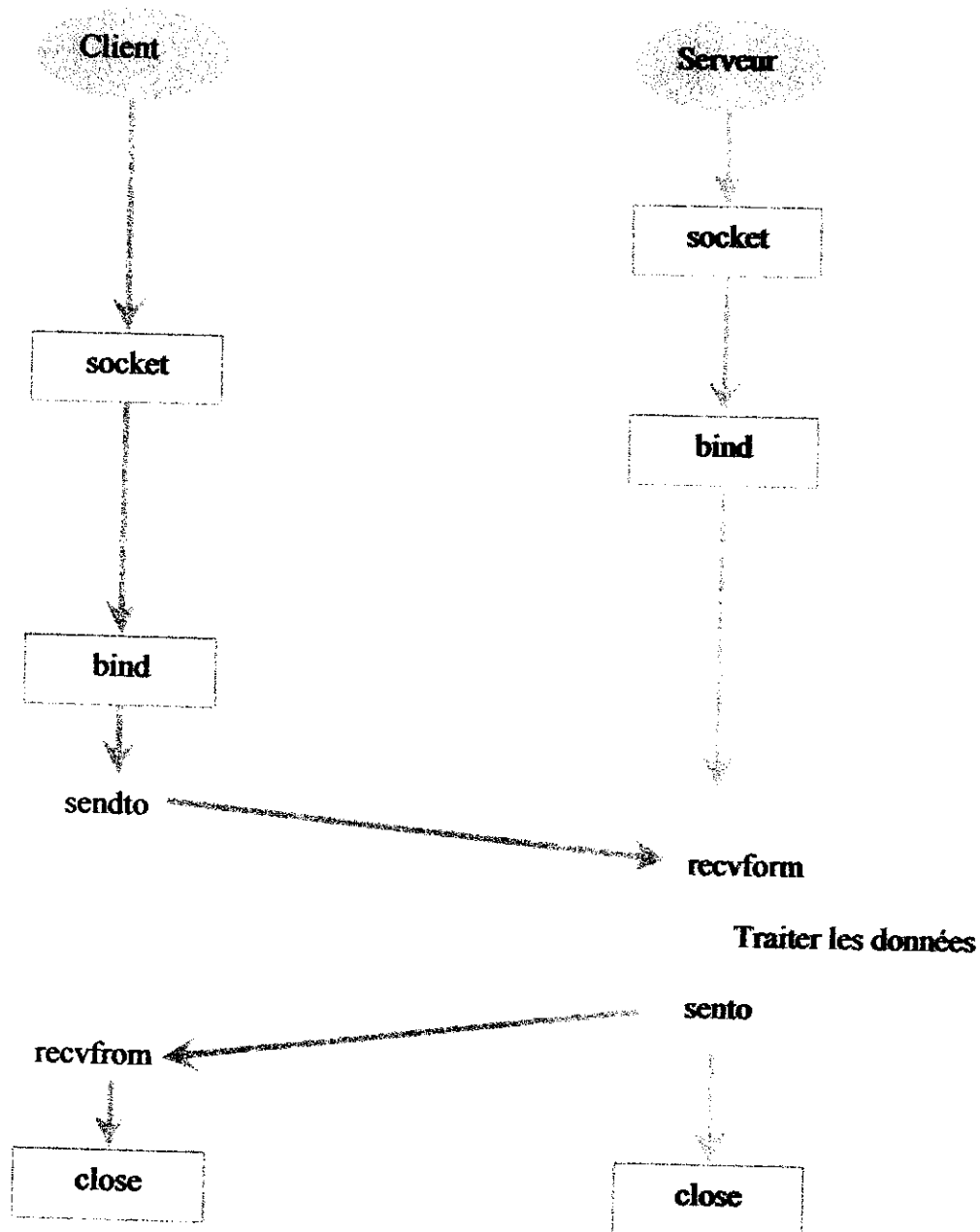


Figure III.5: schéma de protocole TCP/IP en mode non connecté.

III.6 Conclusion :

Développer une application informatique sous réseau est devenu une tâche très importante vu des avantages qu'offre le réseau en matière de communication et l'accès rapide et à distance à l'information.

Concernant notre projet, il fonctionne selon l'environnement client/serveur qui se trouve dans un réseau intranet, ce qu'il lui ouvre une fenêtre sur Internet et donne la possibilité aux médecins de consulter l'examen EEG effectué sur des patients en se trouvant n'importe où.

CHAPITRE IV:

La description de l'application

IV.1 Introduction :

Notre projet a pour objective de réaliser une cartographie en 2D des signaux EEG dans une architecture client/serveur et pour mieux comprendre le processus de développement de notre système on se réfère à un des modèles de cycle de vie, permettant de prendre en compte, en plus des aspects techniques, l'organisation et les aspects humains. Pour cela, on a opté pour le cycle de vie en cascade qui permet de décrire toutes les phases de développement d'un logiciel de l'établissement des besoins jusqu'à teste et implémentation. Le schéma de la figure IV.1 nous montre les déférentes étapes par lesquelles passe un logiciel selon le cycle de vie en cascade.

Pour permettre de formaliser les étapes préliminaires de développement d'un système et rendre ce développement plus fidèle aux besoins d'utilisateur, on a recours aux méthodes d'analyse et de conception. Parmi les méthodes les plus connues on peut notamment citer le langage de modélisation unifié UML.

Né de la fusion des méthodes objet dominantes (OMT, Booch et OOSE), puis normalisé par l'OMG en 1997, UML est rapidement devenu un standard incontournable. UML n'est pas à l'origine des concepts objet, mais il en donne une définition plus formelle et apporte la dimension méthodologique qui faisait défaut à l'approche objet.

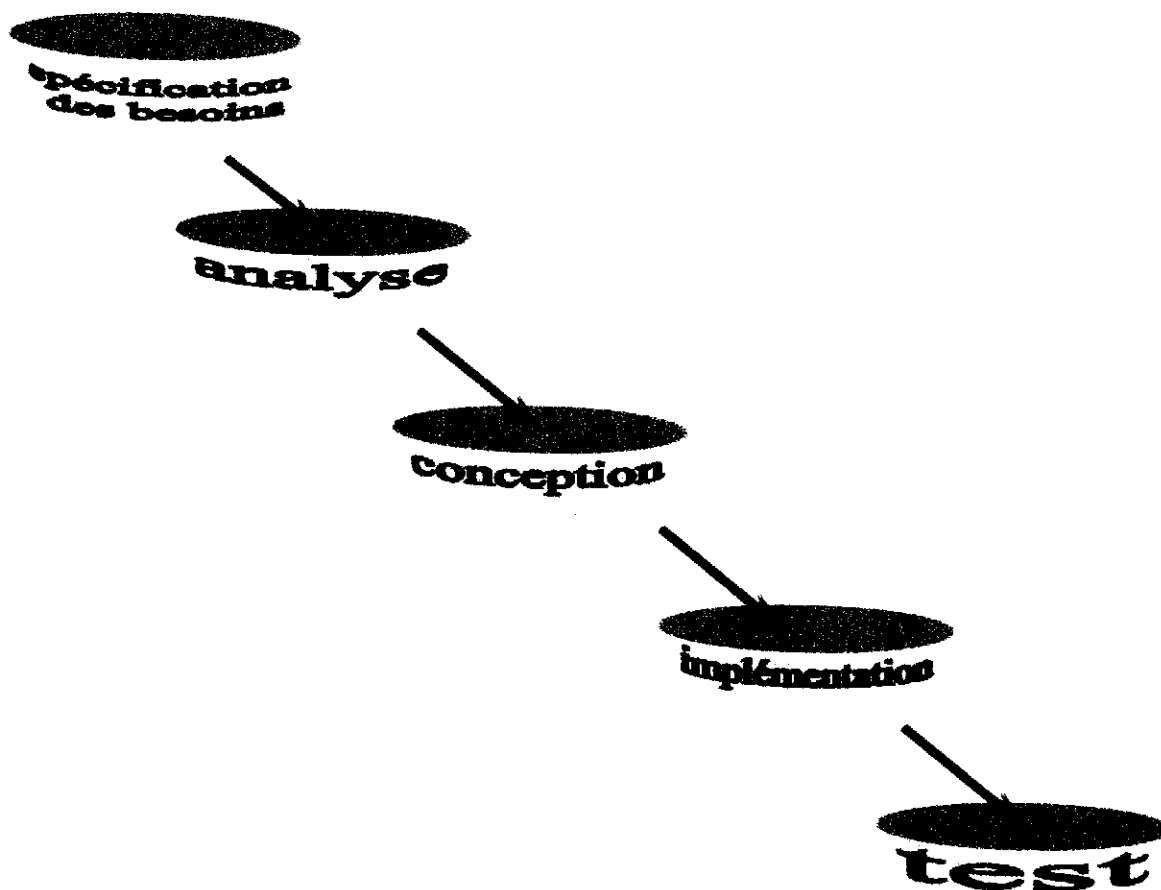


Figure IV.1 : cycle de vie en cascade.

IV.2 Spécification des besoins :

La démarche de développement de notre projet est pilotée par les besoins des utilisateurs. Avec UML ce sont les utilisateurs qui guident la définition des modèles et définissent ce que doit être le système. Les besoins des utilisateurs servent aussi de fil rouge, tout au long du cycle de développement.

- A chaque itération de la phase d'analyse, on clarifie, affine et valide les besoins des utilisateurs.

- A chaque itération de la phase de conception et de réalisation, on veille à la prise en compte des besoins des utilisateurs.
- A chaque itération de la phase de test, on vérifie que les besoins des utilisateurs sont satisfaits.

IV.2 .1 Les cas d'utilisations :

Les cas d'utilisation (en anglais *use case*) permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.

Un cas d'utilisation permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre, mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne.

L'objectif poursuivi par les cas d'utilisation est de permettre de décrire, la finalité des interactions du système et de ses utilisateurs.

IV.2.1.1 Les acteurs :

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. On distingue deux type d'acteurs : principal (idéalement il y en a un seul), et secondaires. Il est à noter qu'un utilisateur peut amené à jouer plusieurs rôles vis-à-vis du système et à ce titre être modélisé par plusieurs acteurs.

Dans notre système l'acteur principale est bien entendu le médecin qui a seul, le droit d'interagir avec le système.

IV.2.1.2 Description textuelle des cas d'utilisation :

Avant d'entamer les diagrammes de cas d'utilisation il est important de commencer par une description textuelle des différents cas d'utilisation composant le système qui sont :

- **Visualisation des signaux** : ce cas d'utilisation permet au médecin de visualiser les signaux EEG.
- **Cartographie en 2D** : c'est le cas le plus important car il constitue l'objectif de notre projet qui vise à représenter graphiquement les signaux EEG sous forme d'une topographie de cerveau dans les domaines temporel et fréquentiel.
- **Exportation des données** : ce cas d'utilisation permet de transmettre les données issus de l'appareil EEG vers d'autres applications dans d'autres stations à travers des connexion réseaux basées sur une architecture client/serveur et communiquant avec le protocole TCP/IP.
- **Stockage des données** : pour tout examen EEG il est recommandé de stocker les résultats obtenus pour une consultation ultérieure, c'est le rôle de cas stockage des données.
- **Consultation** : après le stockage des données, ce cas vient pour permettre au médecin de revoir les résultats des examens EEG obtenus précédemment. Comme il peut les traiter, les analyser et les visualiser.
- **Acquisition et restitution des données** : pour avoir les signaux EEG on doit commencer par l'acquisition qui permet d'extraire les échantillons.
- **Analyse et traitement des données** : après l'acquisition des données c'est ce cas qui va permettre de traiter le signal EEG en lui effectuant le filtrage, l'amplification et la transformée de fourier.

IV.2.1.3 Les diagrammes de cas d'utilisation :

Ils décrivent sous la forme d'actions et de réactions le comportement d'un système du point de vue de l'utilisateur et donc le caractère fonctionnel des objets. Les besoins de chaque acteur déterminent l'ensemble des besoins d'un système. La description des cas d'utilisation délimite le système, ils seront utilisés tout au long du cycle de vie du projet.

Le cas d'utilisation décrit le quoi et le quoi faire et non le comment qui relève de la conception.

IV.2.1.3.1 Le diagramme globale des cas d'utilisation :

Dans ce diagramme, nous allons décrire d'une façon générale les fonctionnalités globales du système en définissant grossièrement les cas d'utilisations les plus évidents comme le montre la figure suivante :

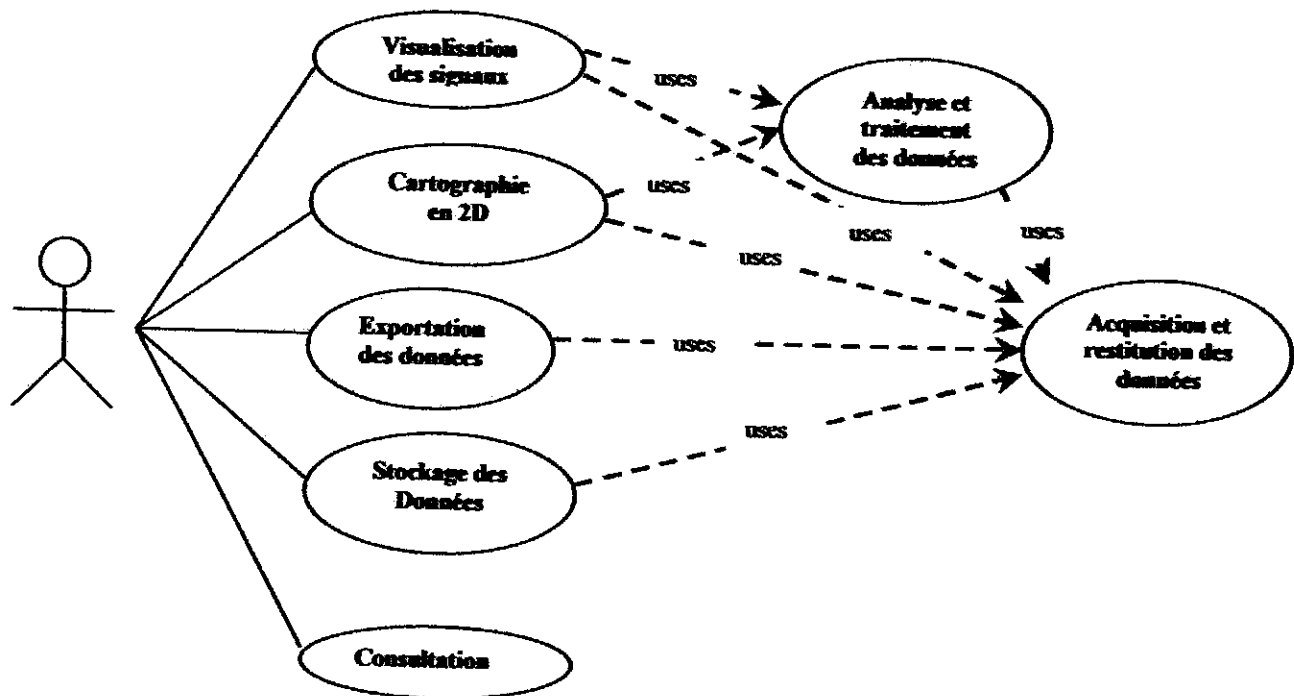


Figure IV.2 : Le diagramme global des cas d'utilisation.

IV.2.1.3.2 Le diagramme de cas d'utilisation « acquisition et restitution des données » :

Dans ce diagramme (figure IV.3) on va spécifier les types d'acquisition qui sont : lecture depuis un fichier, en temps réel grâce à la carte d'acquisition ou à distance en utilisant un réseau. Et tous ces types utilisent la lecture d'une carte d'acquisition.

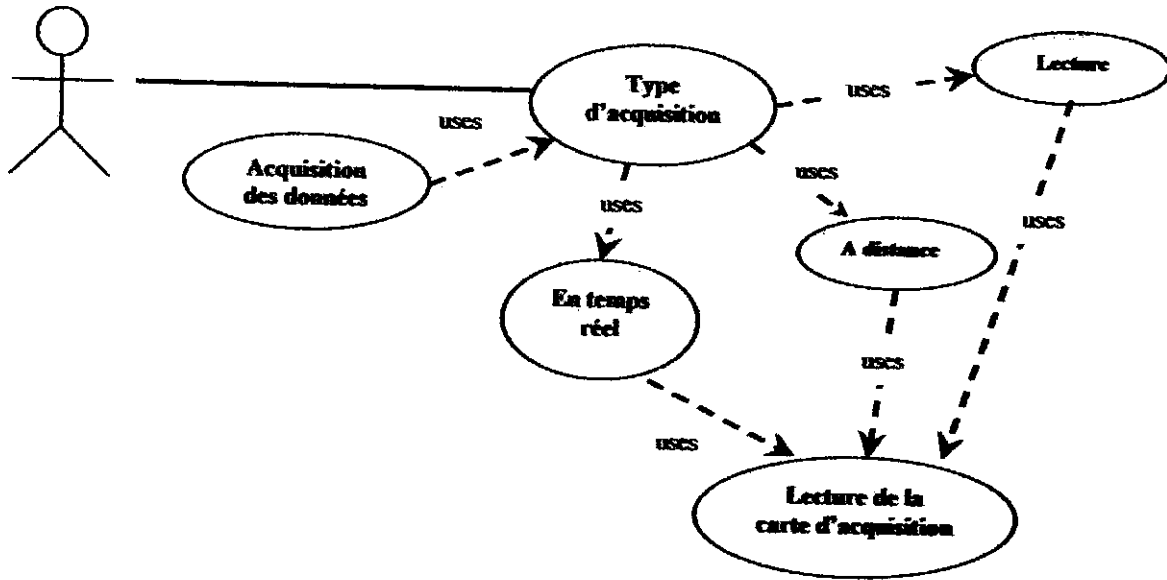


Figure IV.3 : Le diagramme de cas d'utilisation « acquisition et restitution des données ».

IV.2.1.3.3 Le diagramme de cas d'utilisation « Analyse et traitement des données » :

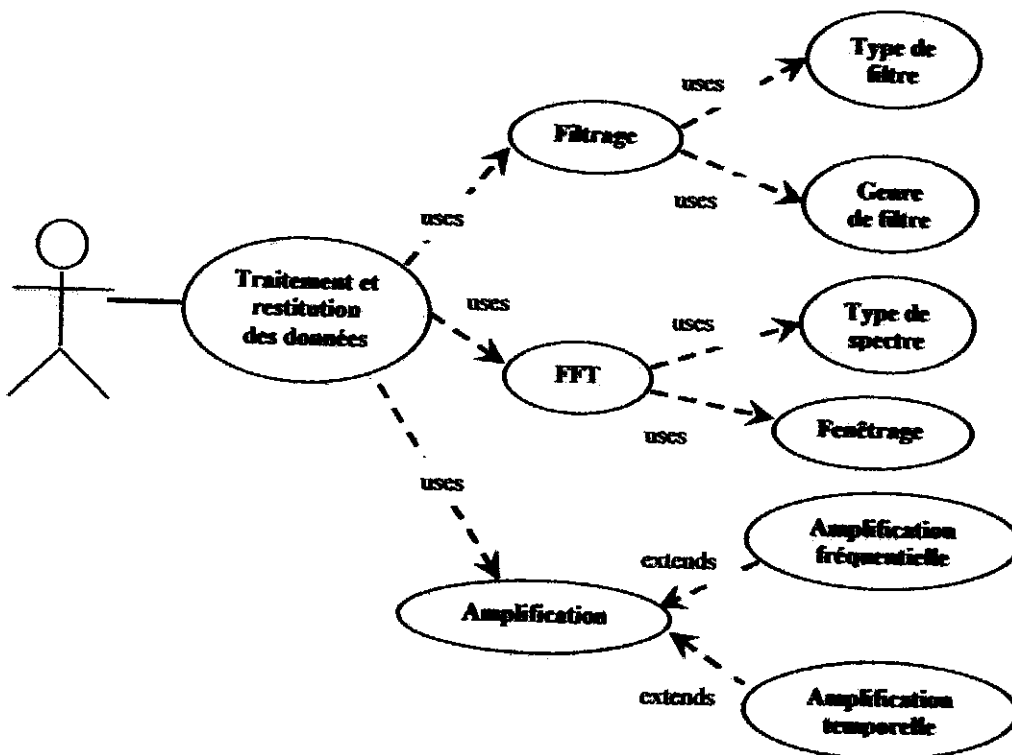


Figure IV.4 : Le diagramme de cas d'utilisation « Analyse et traitement des données ».

L'analyse et traitement des données à pour rôle d'effectuer les opérations de filtrages, transformée de fourier rapide et l'amplification tout en utilisant le type et le genre de filtre pour l'opération de filtrage et le type de spectre et le fenêtrage pour la fft. Et pour l'amplification il y a deux types : l'amplification temporelle et l'amplification fréquentielle. Le diagramme de la figure IV.4 illustre bien ce cas d'utilisation.

IV.2.1.3.4 Le diagramme de cas d'utilisation « Visualisation des signaux » :

Pour la visualisation des signaux issus de l'appareille EEG on a opté pour deux représentation : le tracé EEG qui consiste à représenter les seize signaux acquis sur un seul graphe, et la visualisation individuelle qui représente un seule signal sur un graphe. (figure IV.5)

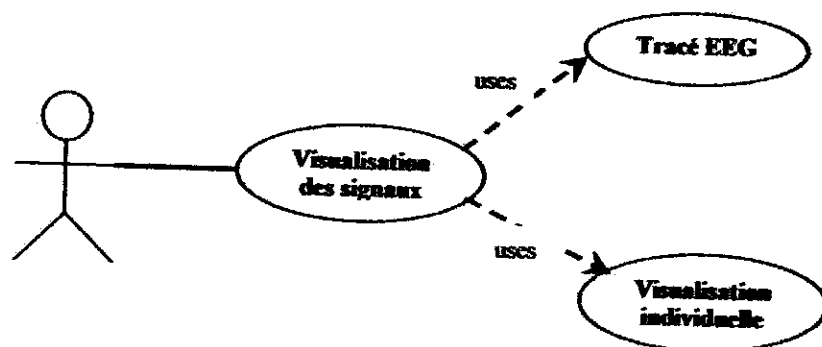


Figure IV.5 : Le diagramme de cas d'utilisation « Visualisation des signaux ».

IV.2.1.3.5 Le diagramme de cas d'utilisation « Cartographie en 2D » :

Pour représenter les signaux graphiquement en 2D on a opté pour deux représentations, l'une dans le domaine temporel en utilisant les valeurs de tension et l'autre dans le domaine fréquentiel en utilisant les fréquences des signaux EEG. La figure IV.6 illustre bien ce diagramme.

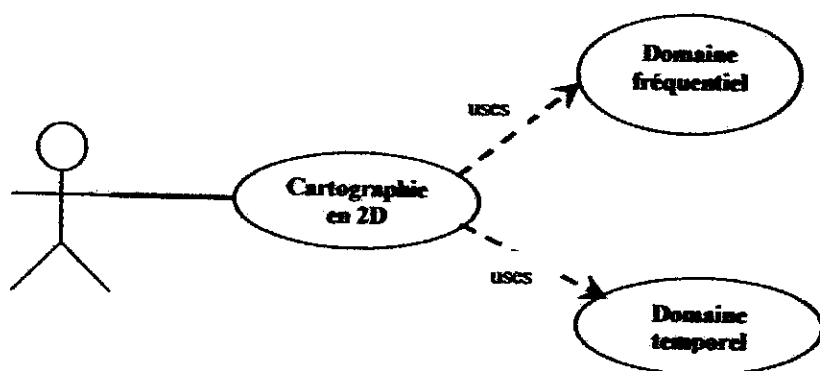


Figure IV.6 : Le diagramme de cas d'utilisation « Cartographie en 2D ».

IV.2.1.3.6 Le diagramme de cas d'utilisation « Stockage des données » :

Pour stocker les données on a besoins des opérations qui sont affectées aux fichiers comme l'ouverture qui s'effectue soit par la création d'un nouveau fichier soit par l'écrasement d'un fichier existant ou bien par la mise à jour d'un fichier déjà ouvert, et l'écriture dans un fichier qui nécessite une conversion en binaire des échantillons, en fin on termine par la fermeture du fichier. (Figure IV.7).

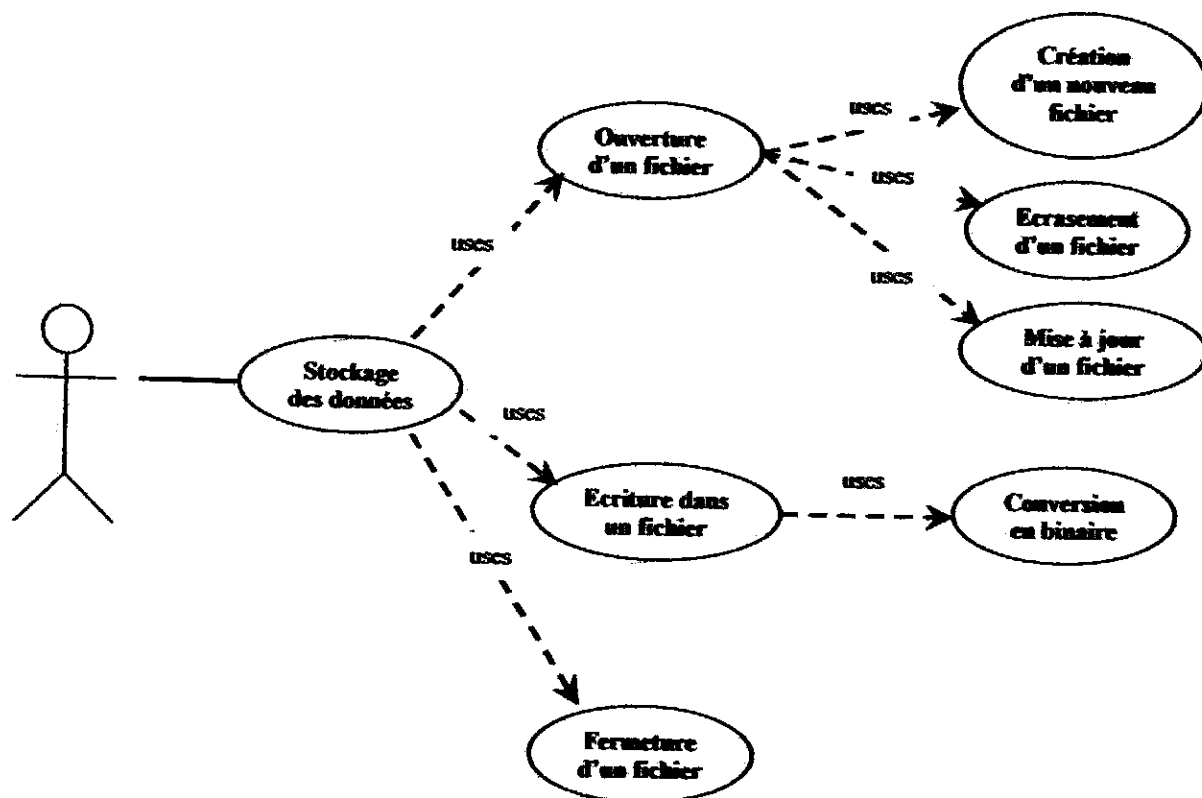


Figure IV.7 : Le diagramme de cas d'utilisation « Stockage des données ».

IV.2.1.3.7 Le diagramme de cas d'utilisation « Exportation des données » :

L'exportation des données vers d'autres applications dans d'autres stations commence par l'établissement de la connexion à la suite d'une demande de connexion de la part d'un client qui mène à la création de la connexion qui va être fermée suite à une demande de fermeture de connexion. En suite la gestion des messages en procédant aux opérations d'empilement et de duplication des messages. En fin on transmet les messages qui sont convertis en binaire. (Figure IV.8).

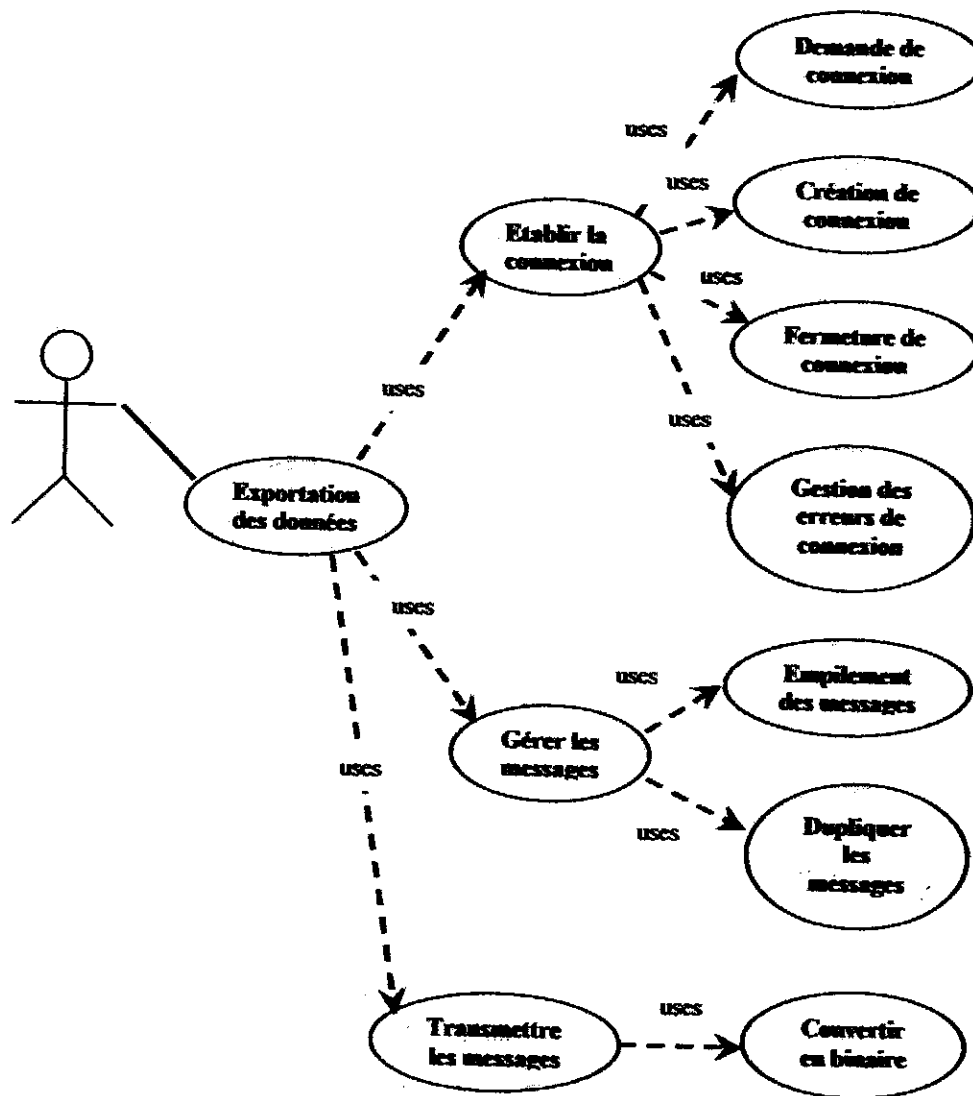


Figure IV.8 : Le diagramme de cas d'utilisation « Exportation des données ».

IV.2.1.3.8 Le diagramme de cas d'utilisation « consultation » :

Pour consulter les données on a besoin d'ouverture d'un fichier, de faire la conversion, de lecture du fichier et en fin de fermeture du fichier (figure IV.9).

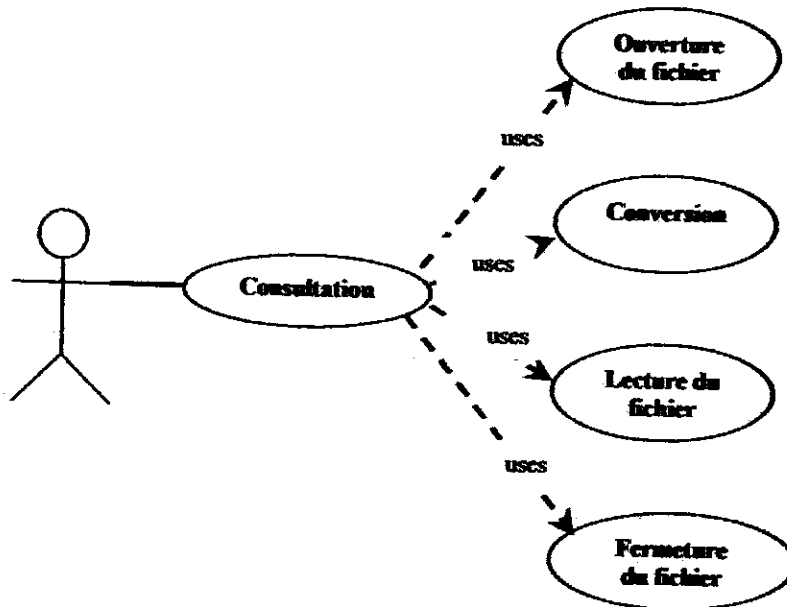


Figure IV.9 : Le diagramme de cas d'utilisation « consultation ».

IV.3 Analyse :

Après la spécification des besoins vient la phase d'analyse en passant de la vue fonctionnelle représentée par les cas d'utilisation vers une vue objet.

IV.3.1 Les paquetages (paquages):

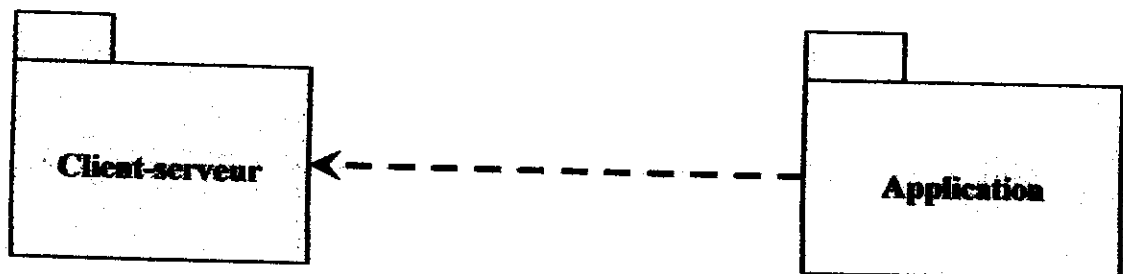
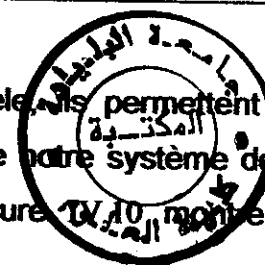


Figure IV.10 les paquetages application et client/serveur

Les paquetages sont des sous ensembles d'un modèle, ils permettent de structurer un système en sous systèmes. On peut dégager de notre système deux principaux paquetages : application et client-serveur. La figure IV.10 montre la relation entre les deux paquetages.



IV.3.1.1 Le paquetage « Application » :

Dans ce paquetage on a rassemblé les différentes classes nécessaires pour réaliser l'application qui a pour finalité l'acquisition et la représentation graphique des signaux EEG. Ces classes sont : traitement, cartographie, visualisation des signaux, échantillon, acquisition, réseau, fichier et carte d'acquisition.

IV.3.1.2 Le paquetage « client-serveur » :

Dans ce paquetage sont rassemblées les classes permettant de transmettre les données vers d'autres applications à travers un réseau basé sur l'architecture client/serveur. Ces différentes classes sont : client, serveur, connexion, réception, duplication, file d'attente et échantillon.

IV.3.2 Les diagrammes de classe :

Les diagrammes de classe montre la structure statique d'un système, il permet la visualisation des classes et les relations entre elles. On peut représenter notre système en deux diagrammes de classe application et client/serveur qui forme en réalité un seul diagramme de classe parce que les deux mènent à la réalisation d'un seul système. Dans ces diagrammes on va représenter les différentes classes constituant le système en déterminant leurs attributs, leurs méthodes et les relations entre elles.

IV.3.2.1 Le diagramme de classe « application » :

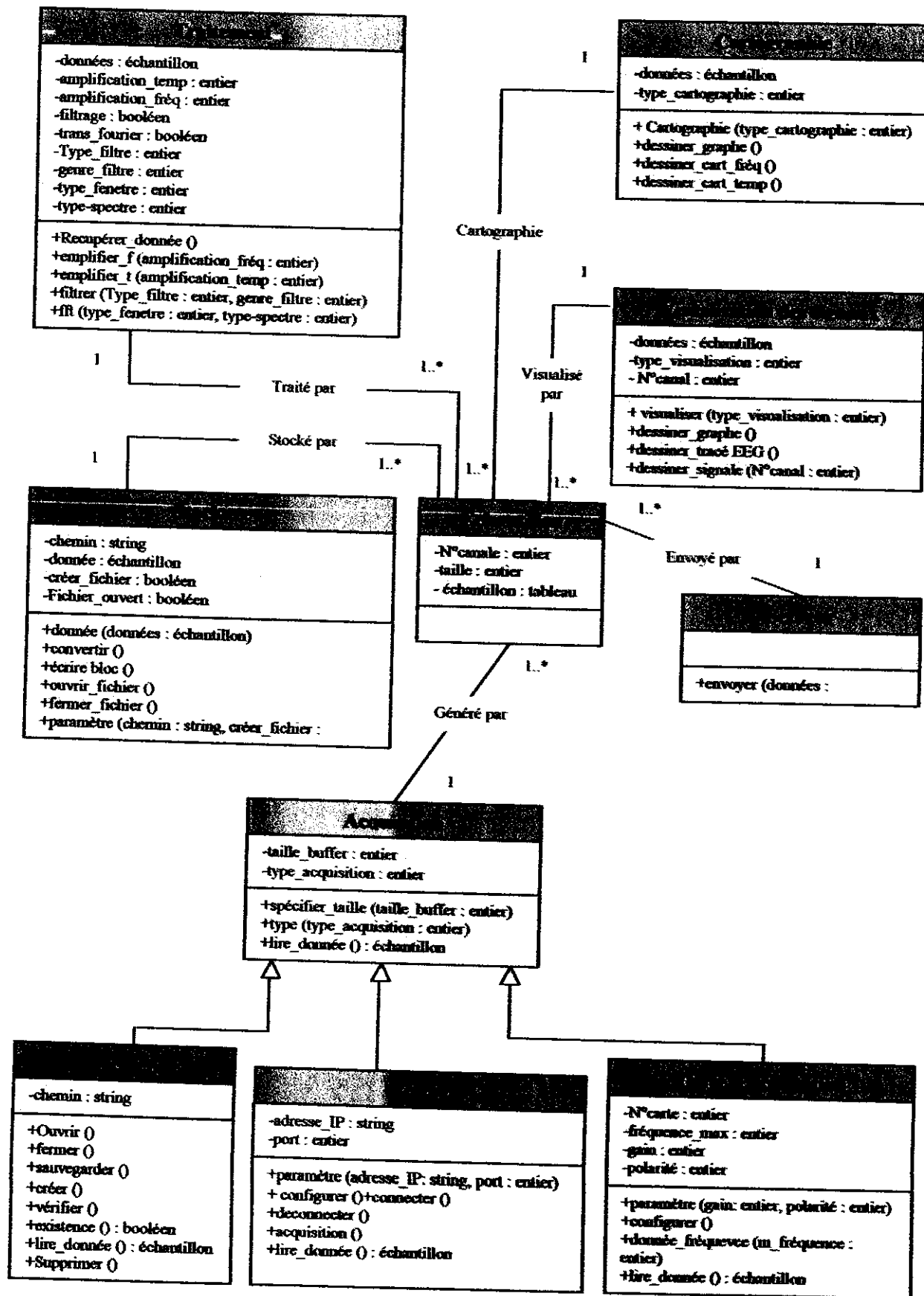


Figure IV.11 : Le diagramme de classe « application ».

IV.3.2.2 Le diagramme de classe « client-serveur »:

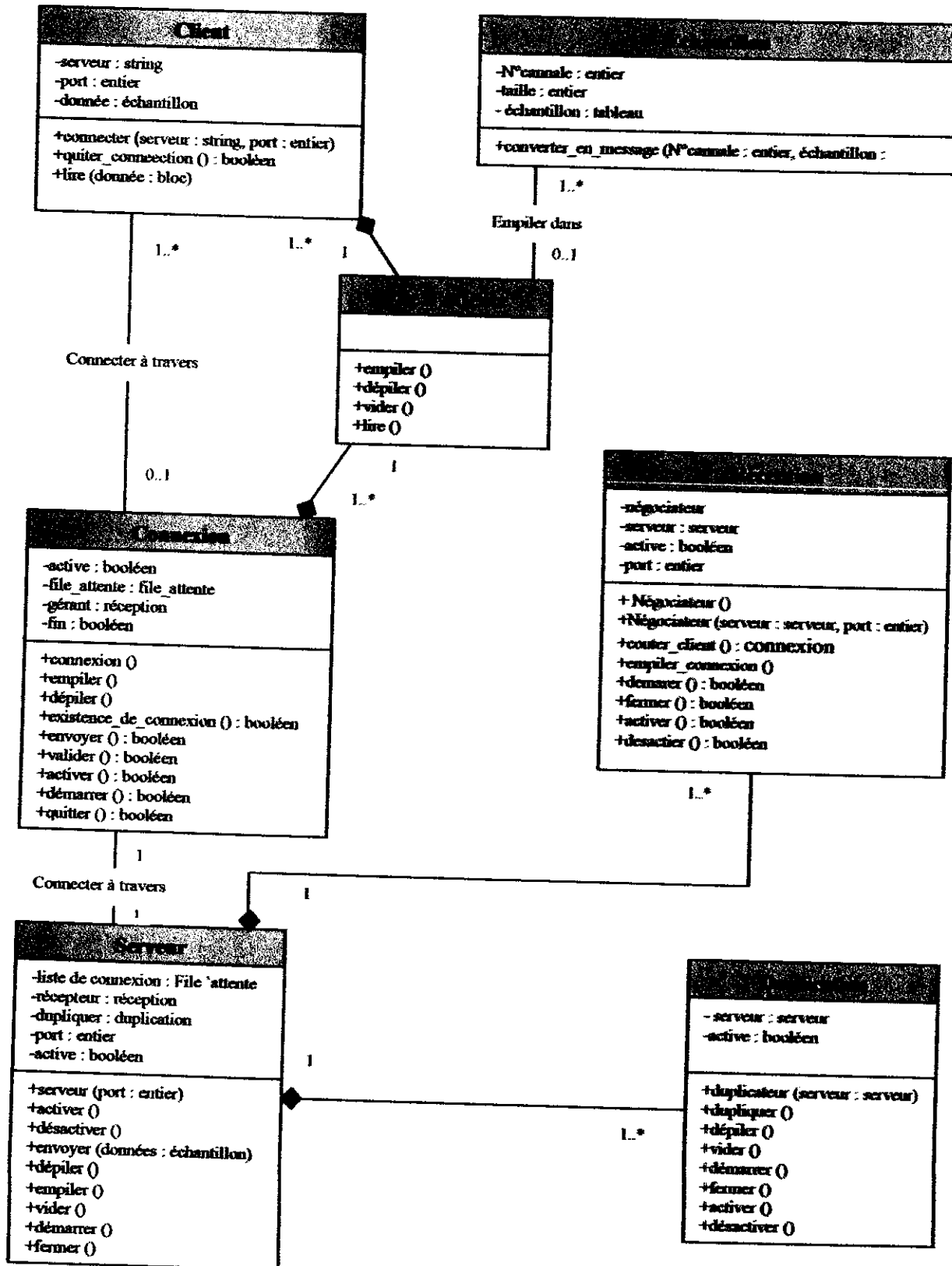


Figure IV.12 : le diagramme de classe « client-serveur ».

IV.3.3 Les diagrammes d'activité :

Les diagrammes d'activité montre la structure dynamique d'un système, ils permettent de représenter graphiquement le comportement interne d'une méthode ou d'un cas d'utilisation.

IV.3.3.1 Le diagramme d'activité global :

Le système commence par l'acquisition et restitution des données via l'EEG, en suit, c'est l'utilisateur qui décide quoi faire soit :

- d'analyser et de traiter les signaux acquis .en suit, les visualiser ou les représenter graphiquement en 2D.
- de stocker les données dans un fichier pour pouvoir les consulter ultérieurement. Et comme les données sont enregistrées dans le micro-ordinateur on pourrait les traiter et les visualiser.
- d'exporter les données vers d'autres applications à travers une connexion réseau en utilisant le protocole de communication TCP/IP pour garantir un partage des données très fiable,
- de consulter les résultats obtenus et enregistrés précédemment.

En fin, le système procède à la fermeture à la suit d'une demande de l'utilisateur (figure IV.13).

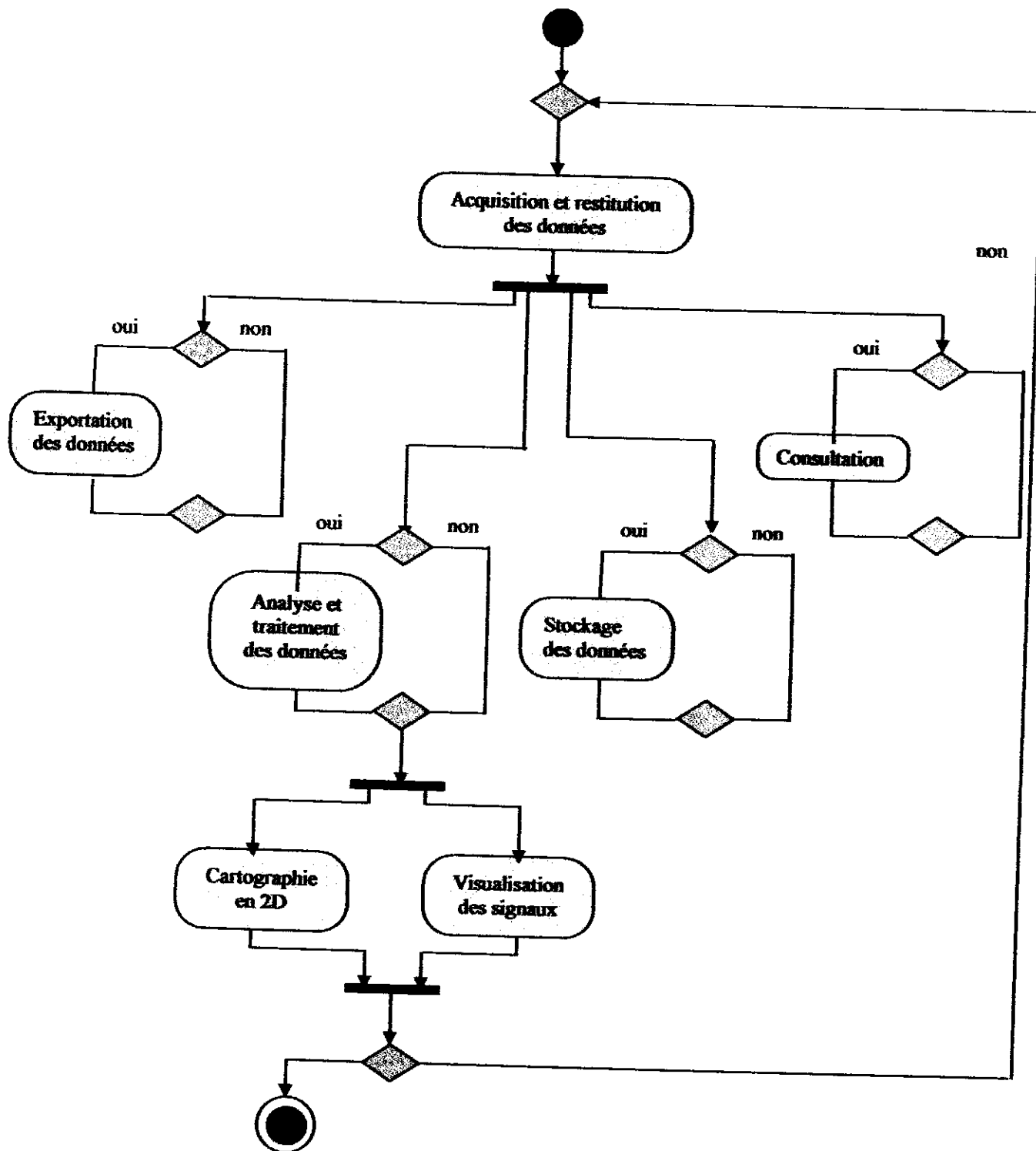


Figure IV.13 : diagramme d'activité global.

IV.3.3.2 diagramme d'activité pour le cas « Acquisition et restitution des données » :

L'acquisition et restitution des données s'effectue, selon le schéma IV.14, soit :

- en temps réel, ce qui exige une configuration de la carte.
- A distance, ce qui demande la configuration de serveur réseau et l'établissement d'une connexion.
- Par lecture d'un fichier.

Après la sélection de type d'acquisition on passe à la lecture d'échantillon.

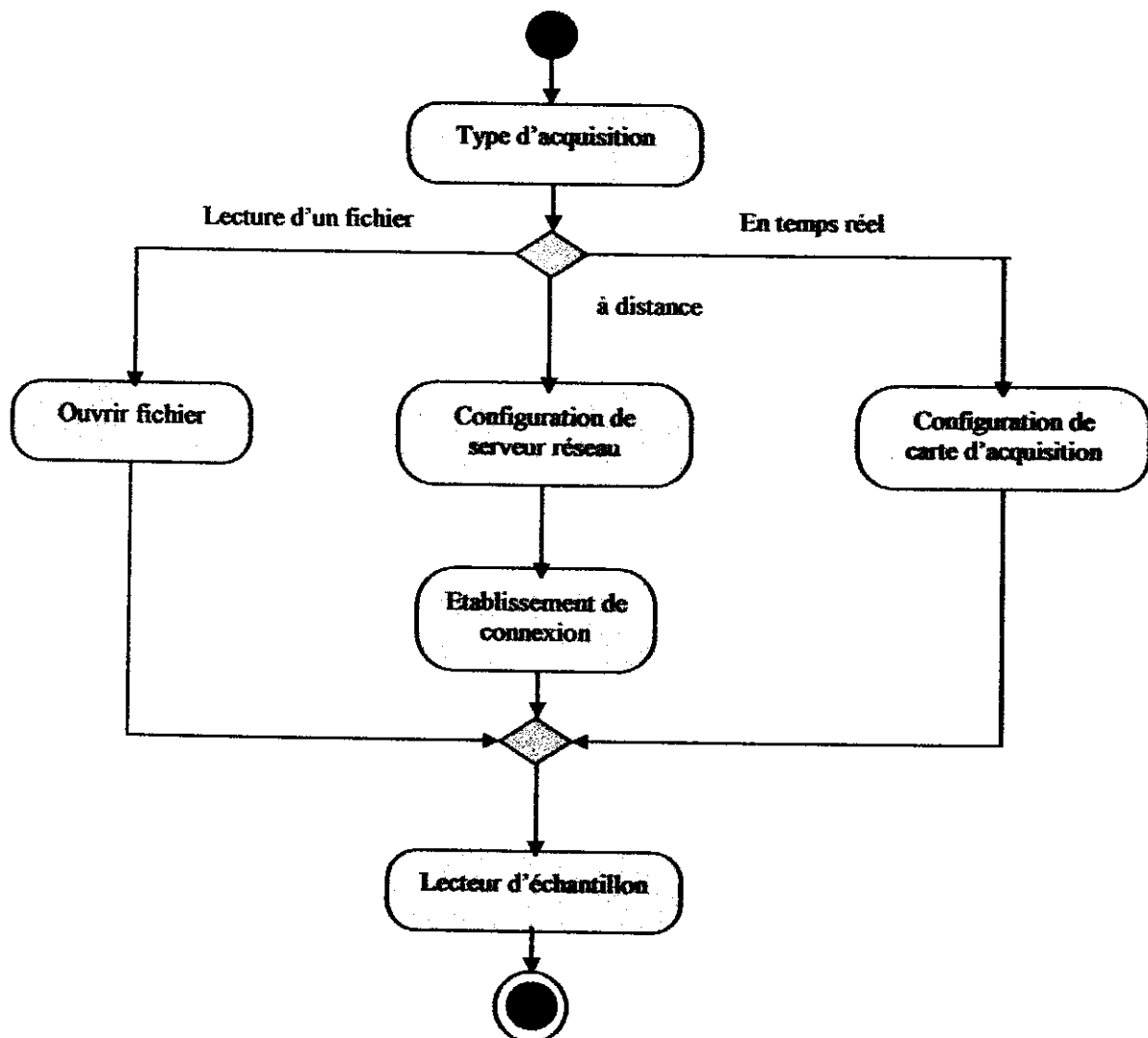


Figure IV.14 : diagramme d'activité pour le cas « Acquisition et restitution des données ».

IV.3.3.3 diagramme d'activité pour le cas « Analyse et traitement des données » :

Après que le système récupère les échantillons à partir d'une source d'acquisition (lecture d'un fichier, la carte d'acquisition, par réseau), L'analyse et traitement de donnée donne à l'utilisateur la possibilité de faire les fonctions suivantes qui reste à sa demande (Figure IV.15) :

- L'amplification temporelle pour permettre d'avoir une visualisation claire des signaux qui ont des amplitudes basses.
- Filtrage des données qui commence premièrement par un choix de genre et le type de filtre et un ordre bien déterminé à fin d'effectuer le filtrage.
- La transformation de Fourier : après le choix de fenêtre et de spectre l'utilisateur peut faire la transformation de Fourier rapide (fft) pour passer de la représentation temporelle vers la représentation fréquentielle des signaux.
- Après la transformation de fourier l'utilisateur a la possibilité de faire une amplification fréquentielle. Pour mieux visualiser les signaux de faibles fréquences.

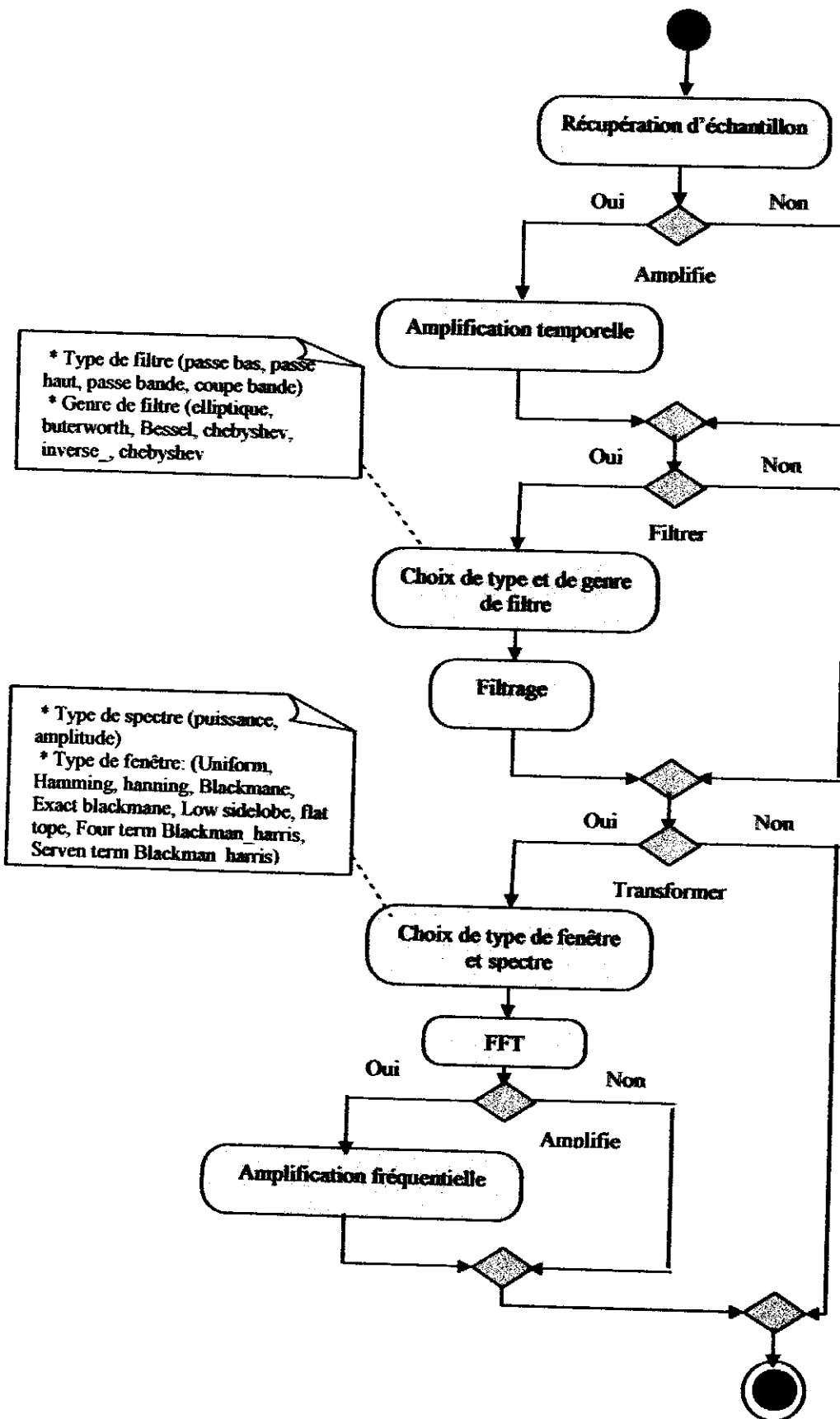


Figure IV.15 diagramme d'activité pour le cas « Analyse et traitement de données ».

IV.3.3.4 Diagramme d'activité pour le cas « Visualisation des données » :

Si l'utilisateur veut visualiser les signaux EEG il n'a que choisir la nature de la visualisation.

S'il veut visualiser un seul signal il n'a que donner le numéro de canal pour récupérer le signal et le dessiner par la suite.

Par contre, s'il veut visualiser les seize signaux à la fois il opte pour le tracé EEG qui lui permet de récupérer les signaux et les dessiner dans un même graphe (figure IV.16).

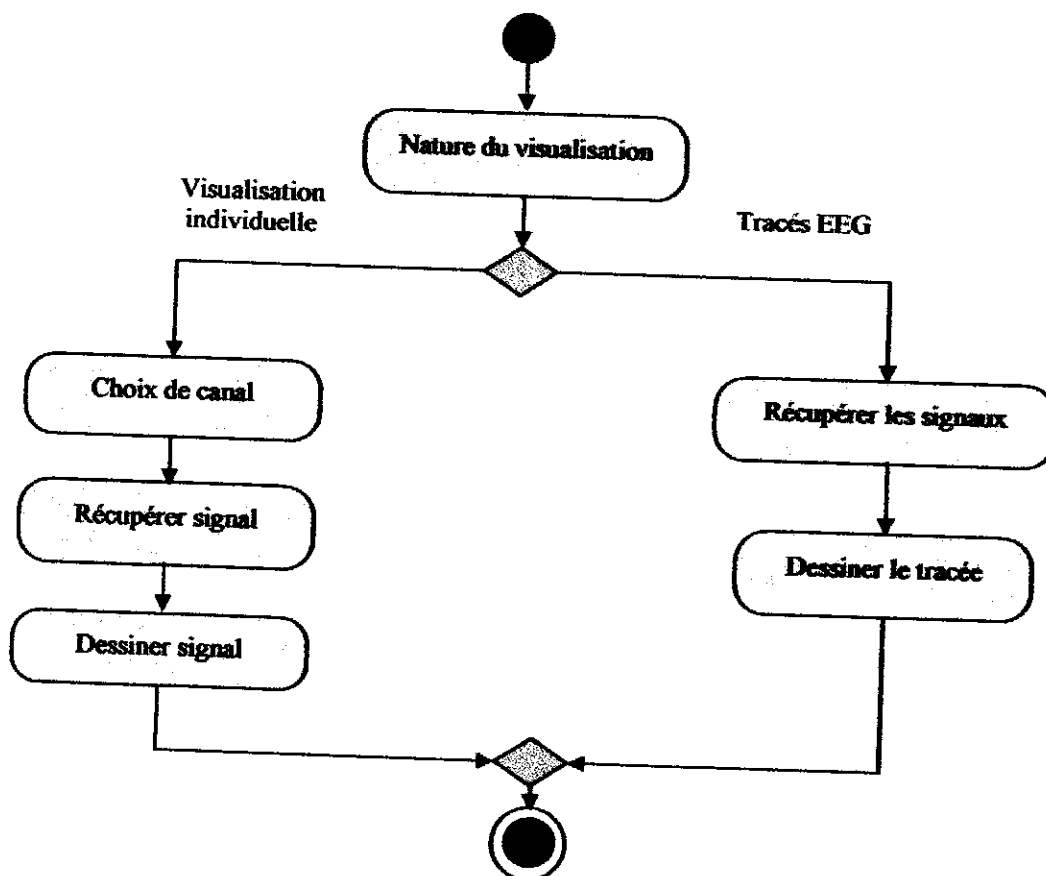


Figure IV.16 diagramme d'activité pour le cas « Visualisation des données ».

IV.3.3.5 diagrammes d'activité pour le cas « Cartographie en 2D » :

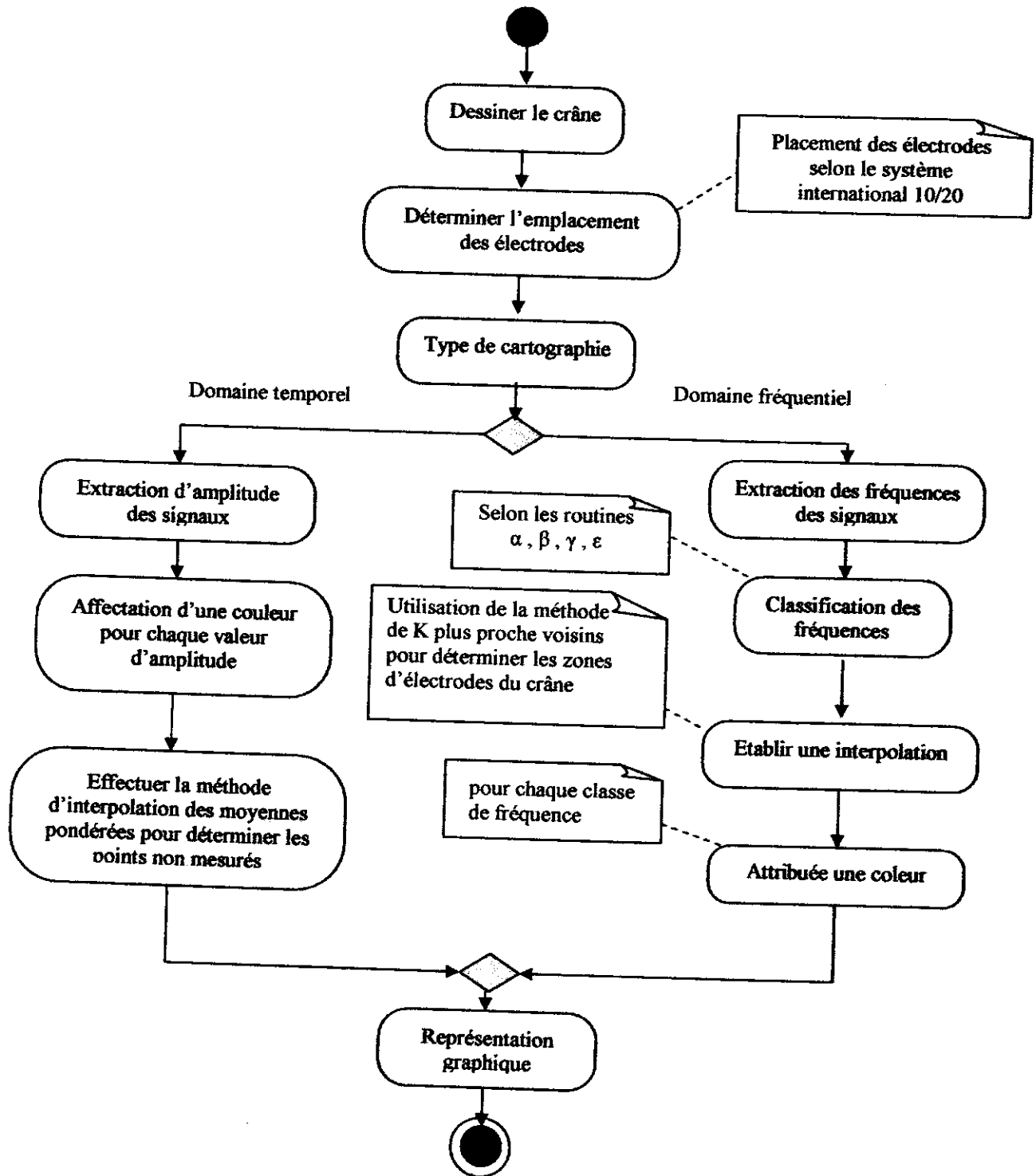


Figure IV.17 : diagramme d'activité pour le cas « Cartographie en 2D ».

Pour la cartographie en 2D des signaux EEG on doit d'abord dessiner le crâne, puis, on détermine l'emplacement des électrodes sur le crâne. En fin, l'utilisateur a la possibilité de choisir le type de cartographie :

Dans le domaine fréquentiel, on va extraire les fréquences à partir des signaux EEG, les classier selon les routines Alpha, Bêta, Thêta, gamma, et par la suite, on attribue une couleur à chaque classe. Et pour pouvoir déterminer les autres points du crâne non calculées on établie une approximation en utilisant la méthode de plus proche voisins selon l'algorithme suivant :

L'algorithme de recherche de plus proche voisin d'un point x consiste à passer sur l'ensemble des N points mesurés du crâne A et à regarder si ce point est le plus proche ou non qu'un de plus proche voisin déjà sélectionné.

Algorithme de recherche de plus proche voisin

$i = 1$

mettre le point $D[i]$ dans plus proche voisin

pour i allant de 2 à N

si la distance entre $D[i]$ et x est inférieure à la distance du point le plus proche voisin à x

supprimer le plus proche voisin de x

mettre dans plus proche voisin le point $D[i]$

fin si

fin pour

Dans le domaine temporel, on va extraire les amplitudes à partir des signaux EEG, et affecter une couleur pour chaque valeur d'amplitude et pour les valeurs intermédiaires non calculées on effectue une approximation pour déterminer ces valeurs et avoir une gradation des couleurs en utilisant la méthode des moyennes mobiles pondérées par l'inverse de la distance élevée à une puissance qui utilise la méthode des K plus proches voisins.

L'algorithme de recherche de voisinage consiste à passer sur l'ensemble des N points du crâne A et à regarder si ce point est plus proche ou non qu'un des plus proches voisins déjà sélectionnés, et si oui, l'insérer

Algorithme de recherche de K plus proches voisins

pour i allant de 1 à k

mettre le point $D[i]$ dans `proches_voisins`

fin pour

pour i allant de $k+1$ à N

si la distance entre $D[i]$ et x est inférieure à la distance d'un des points de `proches_voisins` à x

supprimer de `proches_voisins` le point le plus éloigné de x

mettre dans `proches_voisins` le point $D[i]$

fin si

fin pour

`proches_voisins` contient les k plus proches voisins de x

pour réaliser l'interpolation pour la cartographie en 2D dans le domaine temporel on a choisis $K=4$ pour les K plus proches voisins et $P=2$ pour la puissance.

Après le choix de type de cartographie le système va montrer la représentation graphique (Figure IV.17).

IV.3.3.6 diagrammes d'activité pour le cas « Stockage des données » :

Si l'utilisateur veut enregistrer les données acquises, il donne d'abord le chemin du fichier, si ce dernier n'est pas ouvert il va le créer s'il n'existe pas un autre fichier ouvert, sinon, si un fichier est ouvert il va soit l'écraser ou le mettre à jour. Ensuite, le système récupère l'échantillon à stocker, le convertit en chaîne de caractère et effectue l'opération de l'écriture du bloc. En fin, si l'utilisateur désire fermer le fichier le système va procéder à sa fermeture (figure IV.18).

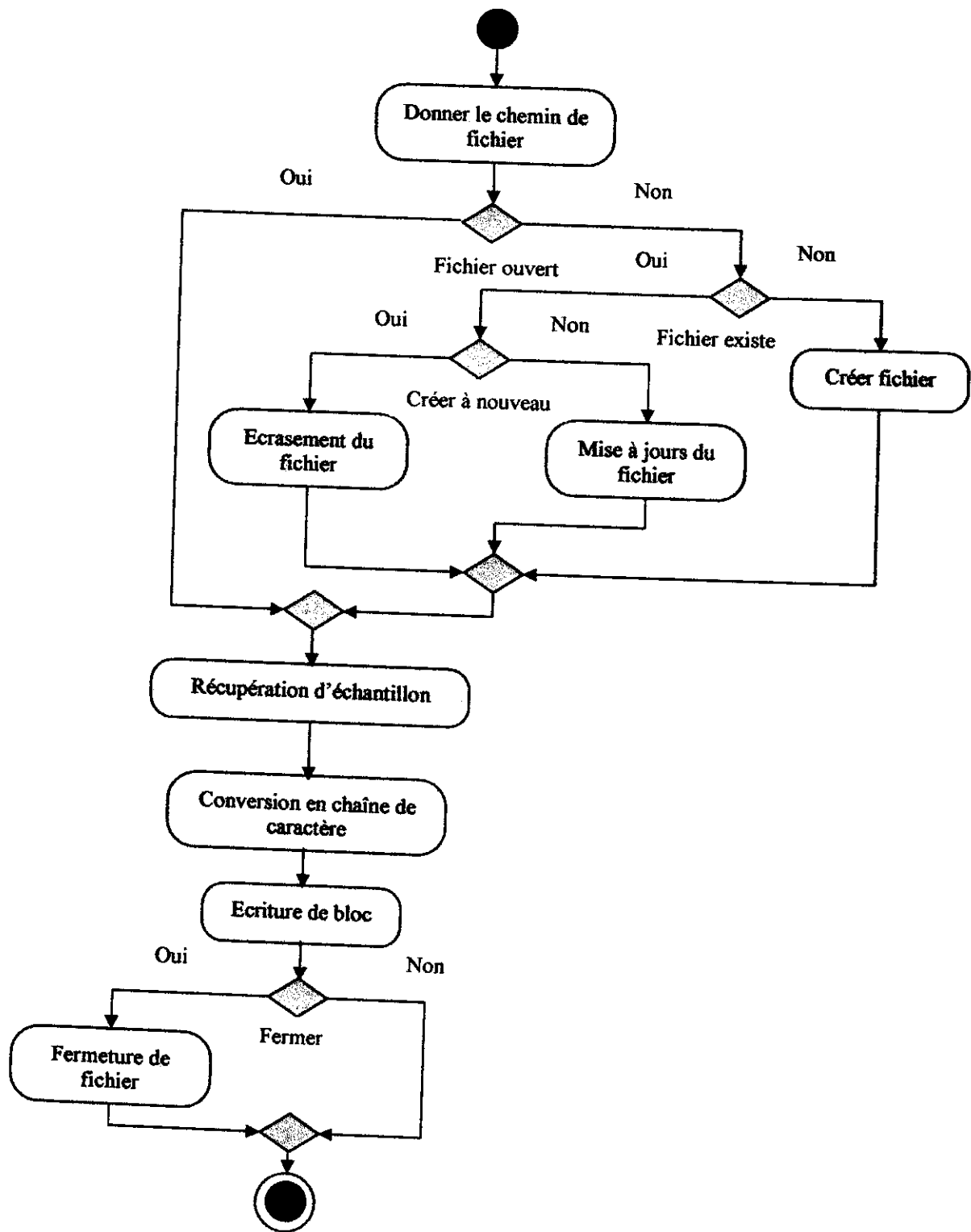


Figure IV.18 : diagramme d'activité pour le cas « Stockage des données ».

IV.3.3.7 diagrammes d'activité pour le cas « Exportation des données » :

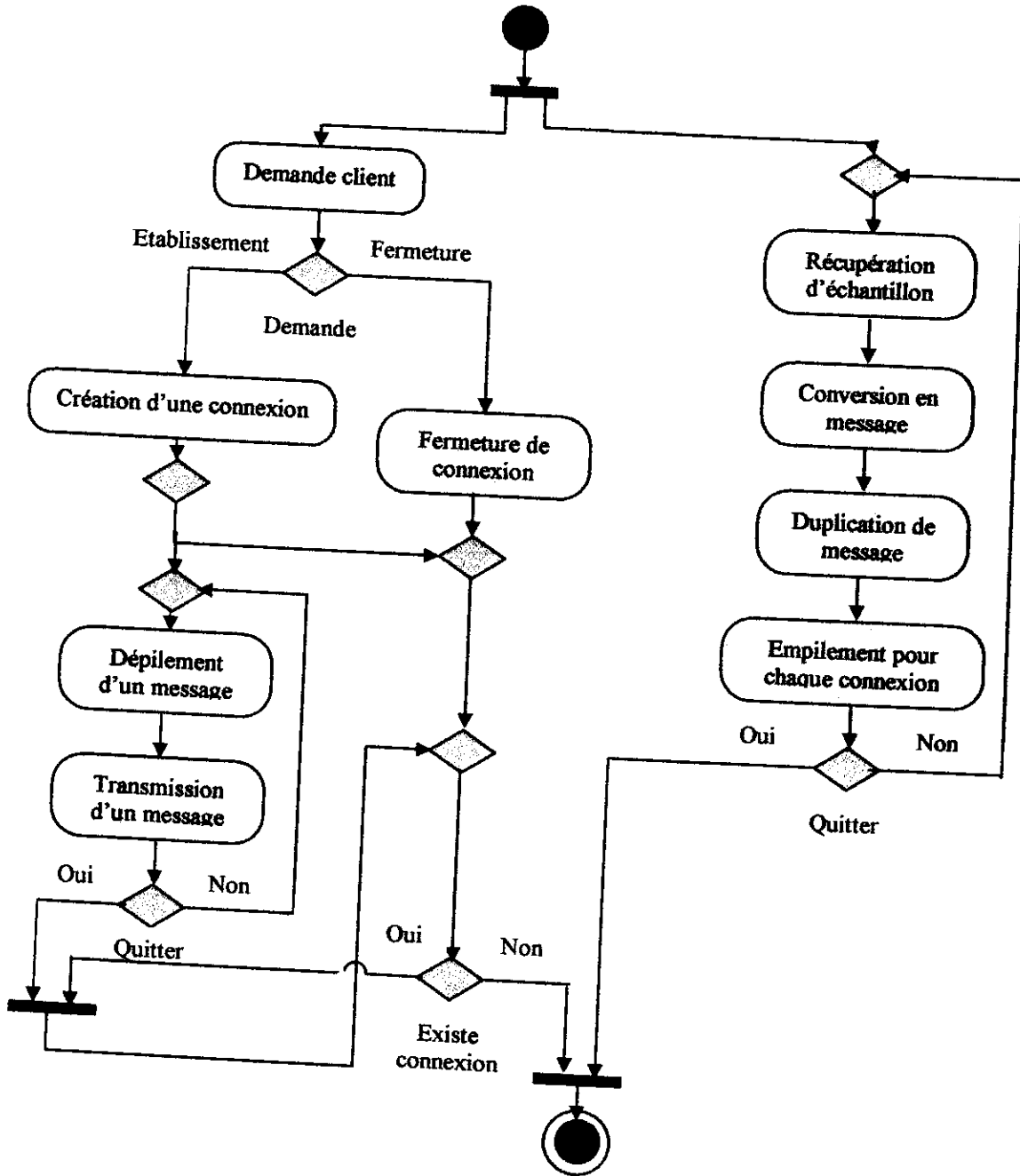


Figure IV.19 : diagramme d'activité pour le cas « Exportation des données ».

Pour mettre l'application « cartographie en 2D des signaux EEG » sous réseau. D'une part, le système récupère l'échantillon, le convertit en message. Par la suite, il va en faire des copies destinées aux applications client et va les empiler pour chaque connexion. D'une autre part, s'il arrive une demande d'établissement d'une

connexion de la part d'un client connecté au réseau, le système procède à sa création. Ainsi, la connexion dépile les messages et les transmet vers le client. Sinon, si le client demande une fermeture de la connexion le système va le faire. Le système se ferme s'il y a lieu une demande de la part de l'utilisateur (figure IV.19).

IV.3.3.8 diagramme d'activité pour le cas « Consultation » :

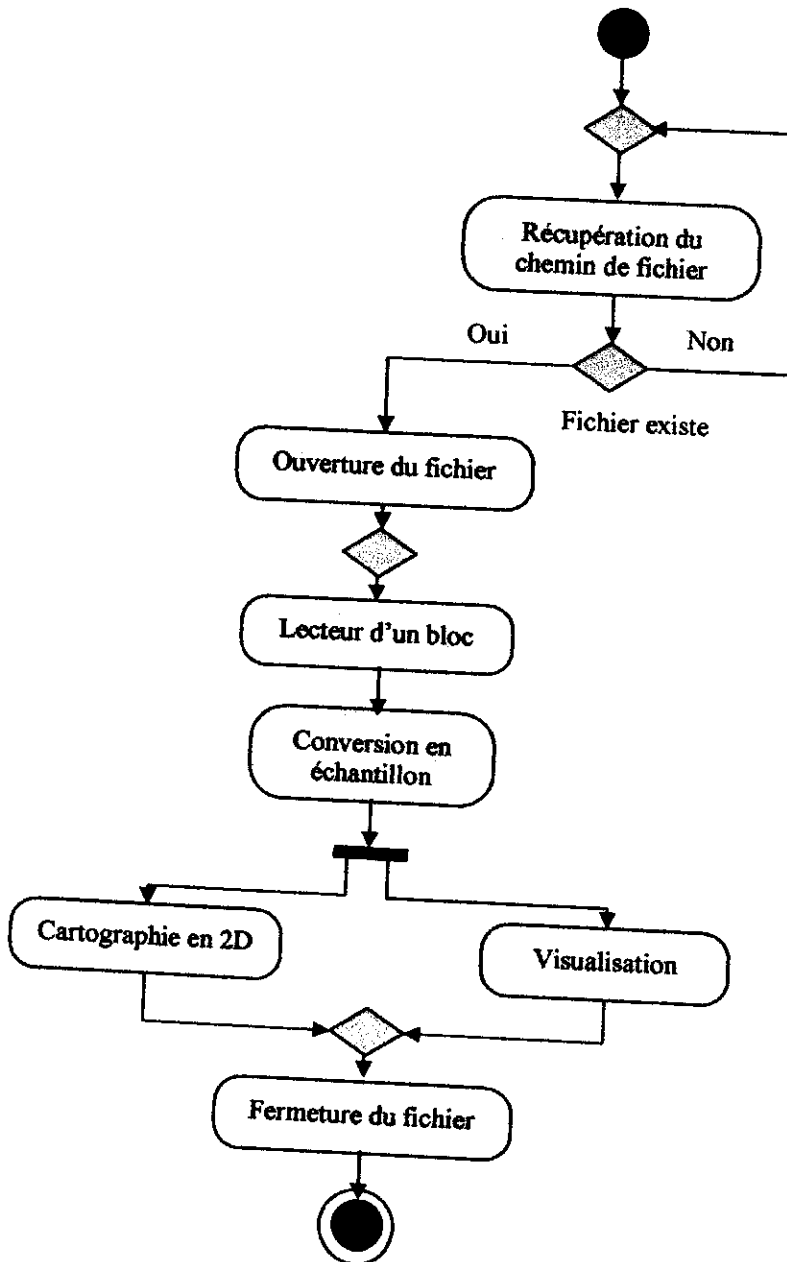


Figure IV.20 diagramme d'activité pour le cas « Consultation ».

Pour que l'utilisateur puisse consulter les données précédemment enregistrées, il introduit d'abord le chemin du fichier. Si le fichier n'existe pas, il n'a qu'introduire un autre chemin. Sinon, il va ouvrir le fichier que le système va lire en faisant la conversion en échantillon. En suite, c'est à l'utilisateur de choisir la façon de représentation d'échantillon soit de visualiser le signal ou d'établir une cartographie en 2D. En fin, si l'utilisateur désire quitter, le système va procéder à la fermeture du fichier (figure IV.20).

IV.4 Conception :

L'objective de la phase de conception est d'avoir une représentation des interactions temporelles entre objets dans la réalisation des opérations. D'avoir une description du déploiement des composants sur les dispositifs matériels. Et de déterminer l'architecture des composants physiques de l'application.

IV.4.1 Les diagrammes de séquence :

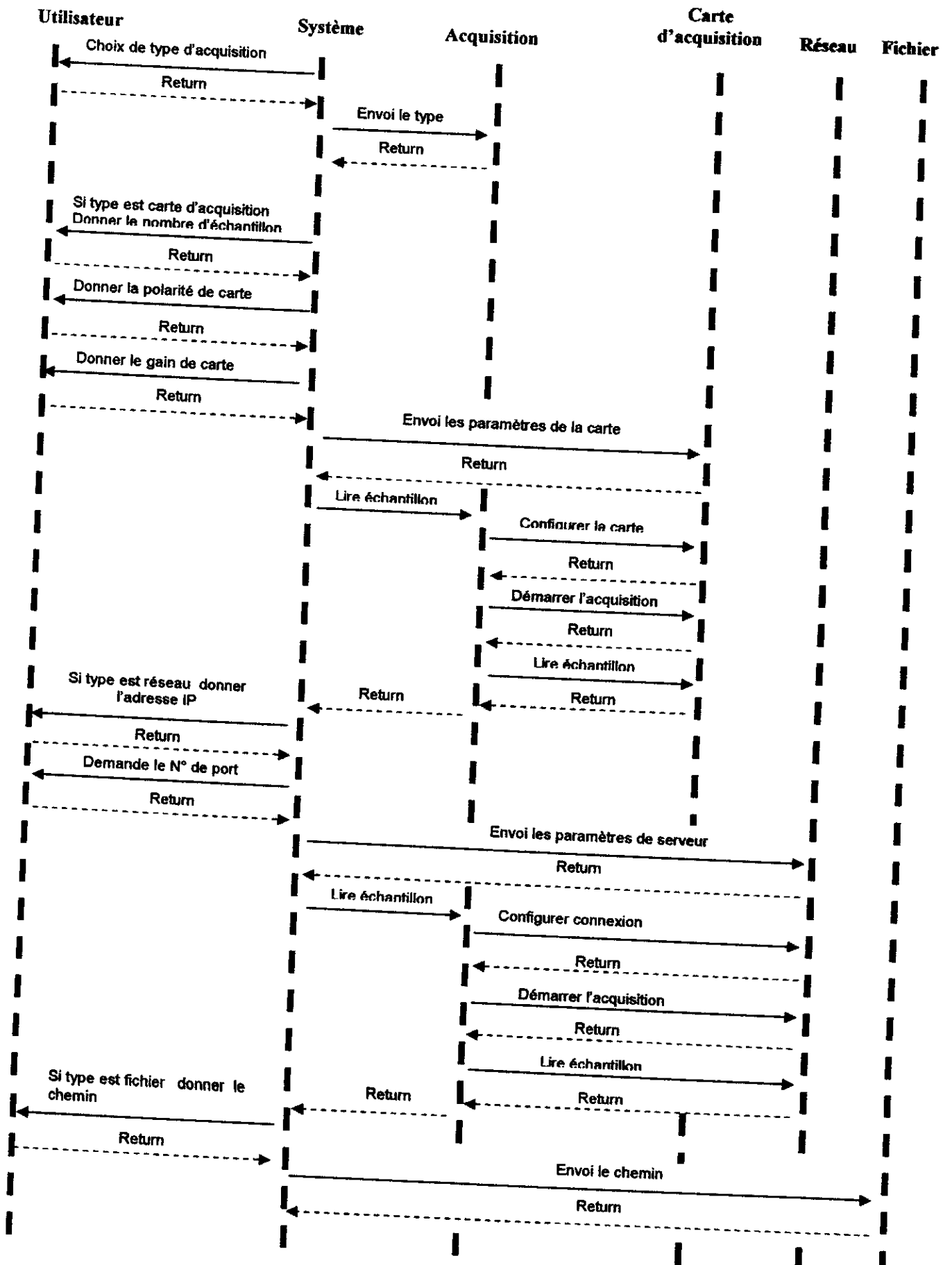
Un diagramme de séquence montre les interactions entre objets selon un point de vue temporel.

IV.4.1.1 Diagramme de séquence « Acquisition et restitution des données » :

Le déroulement de l'acquisition des données est réalisé selon les opérations illustrées dans la figure IV.21 :

- Le système demande le type de l'acquisition à l'utilisateur. Ce dernier va envoyer le type à l'acquisition.
- Si le type est carte d'acquisition le système demande à l'utilisateur le nombre d'échantillons, la polarité et le gain. Ces derniers vont être envoyés à la carte d'acquisition.
- Le système demande à l'acquisition de lire l'échantillon, l'acquisition de son part demande à la carte d'acquisition de configurer la carte, puis, de démarrer l'acquisition et en fin la lecture d'échantillon.

- Si le type est réseau le système demande à l'utilisateur l'adresse IP de serveur et le numéro de port. Ces derniers vont être envoyés au réseau.
- Le système demande à l'acquisition de lire l'échantillon, l'acquisition de son part demande au réseau de configurer la connexion, puis, de démarrer l'acquisition et en fin la lecture d'échantillon.
- Si le type est fichier le système demande à l'utilisateur le chemin du fichier. Ce dernier va être envoyé au fichier.
- Le système demande à l'acquisition de lire l'échantillon, l'acquisition de son part demande au fichier de charger le fichier, puis, de démarrer l'acquisition et en fin la lecture d'échantillon.



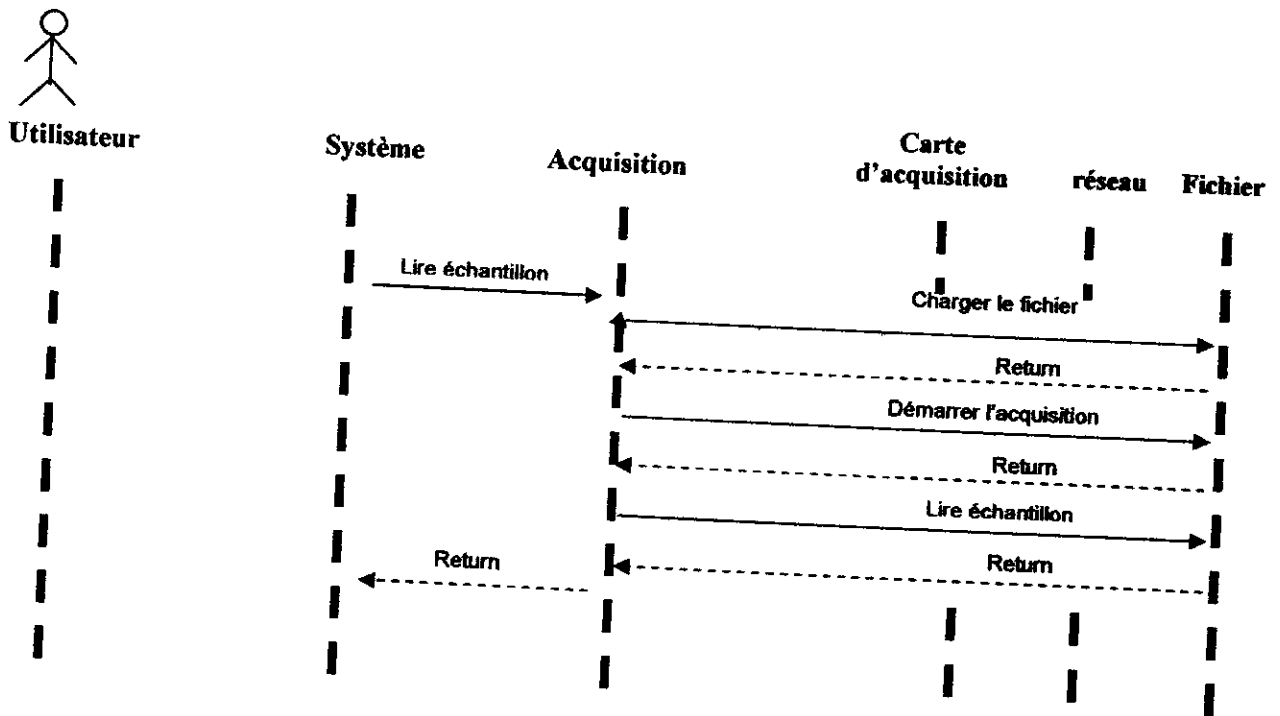


Figure IV.21 : diagramme de séquence « Acquisition et restitution des données ».

IV.4.1.2 diagramme de séquence « Cartographie en 2D » :

La figure IV.22 montre les opérations conduisant à la cartographie en 2D des signaux du cerveau.

- Le système récupère les données de l'acquisition.
- Le système demande le type de cartographie de l'utilisateur.
- Si le type est domaine temporel, le système envoie le type à la cartographie qui à son tour, demande au système d'extraire les amplitudes des signaux acquis.
- La cartographie affecte une couleur pour chaque intervalle bien déterminé d'amplitudes.
- Si le type est domaine fréquentiel, le système envoie le type à la cartographie qui à son tour, demande au système d'extraire les fréquences des signaux acquis.
- La cartographie classe les fréquences selon les routines et par la suite, elle attribue à chaque classe une couleur.

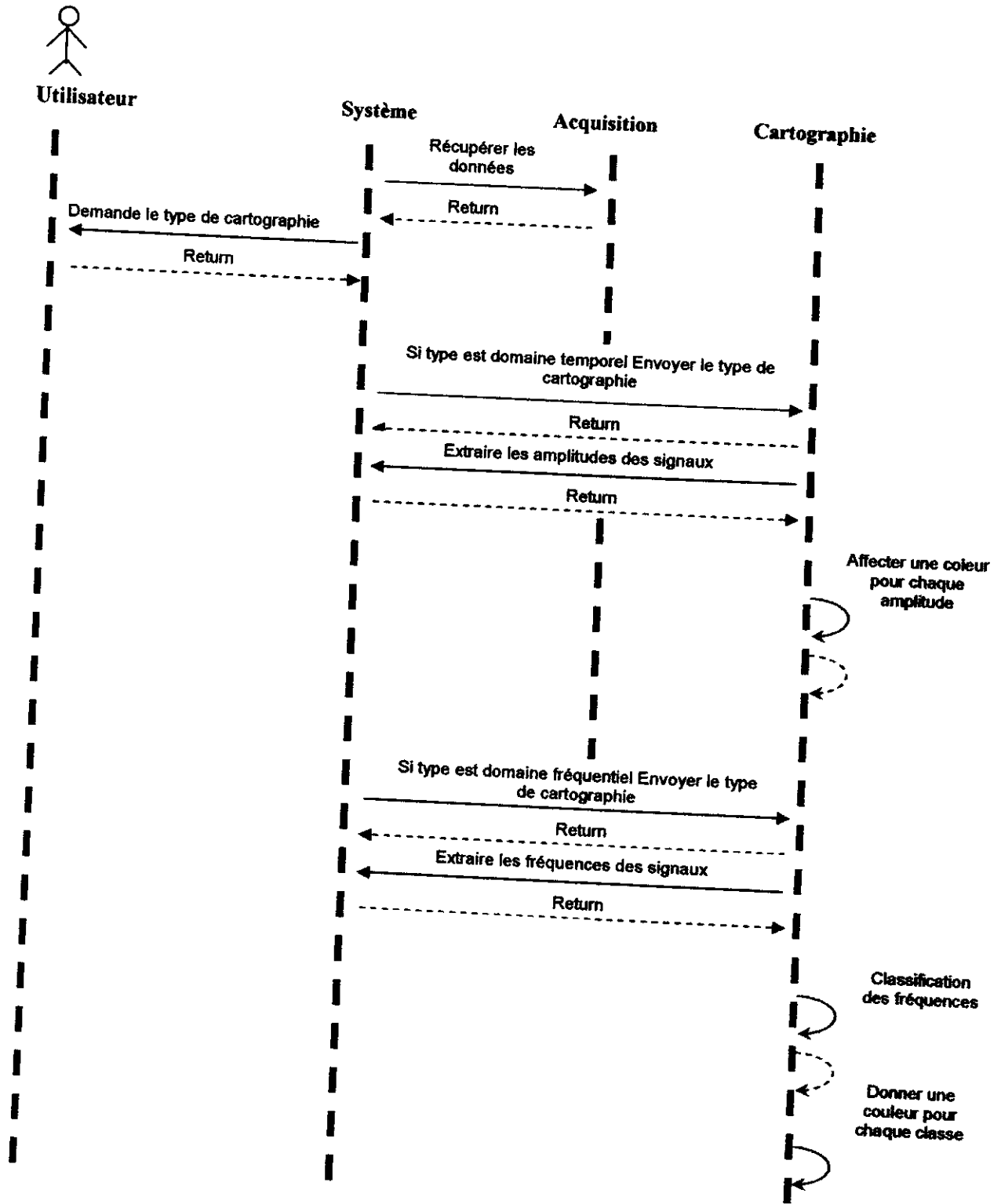


Figure IV.22 : diagramme de séquence « Cartographie en 2D ».

IV.4.1.3 diagramme de séquence « Analyse et traitement des données » :

L'analyse et traitement des données se déroulent entre l'utilisateur, le système, et traitement selon le scénario illustré par la figure IV.23:

- l'objet traitement récupère l'échantillon acquis par le système.
- Le système vérifie s'il y a une décision d'amplification temporelle de la part de l'utilisateur, si c'est le cas le système récupère la valeur de l'amplification et l'envoie au traitement qui va procéder à l'amplification.
- Le système vérifie s'il y a une décision de filtrage de la part de l'utilisateur, si c'est le cas le système récupère le type et le genre de filtre et les envoie au traitement qui va procéder au filtrage.
- Le système vérifie s'il y a une décision de transformation de fourier rapide de la part de l'utilisateur, si c'est le cas le système récupère le type de fenêtrage et celui de spectre les envoie au traitement qui va procéder à la transformation.
- Le système vérifie s'il y a une décision d'amplification fréquentielle de la part de l'utilisateur, si c'est le cas le système récupère la valeur de l'amplification et l'envoie au traitement qui va procéder à l'amplification.

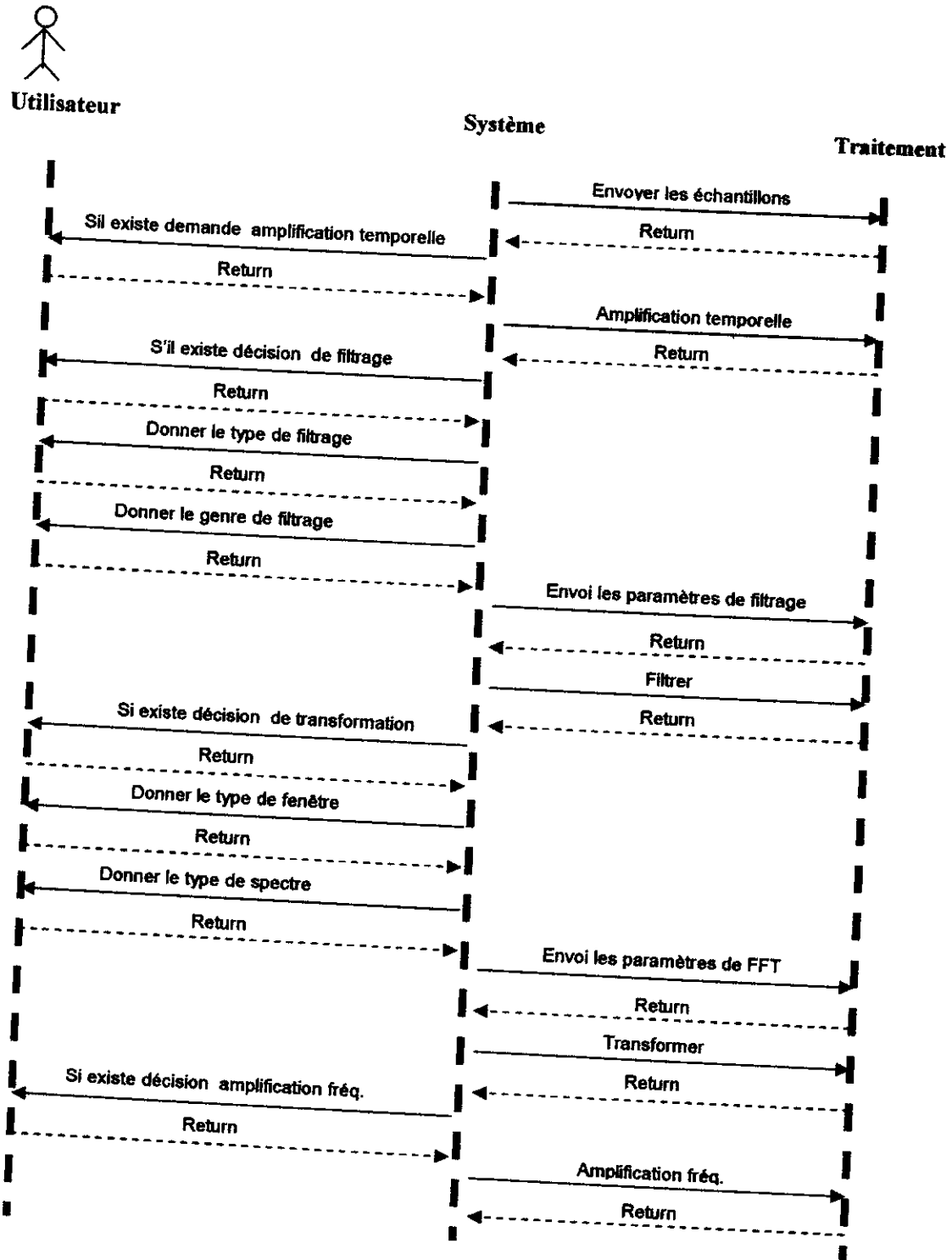


Figure IV.23 : diagramme de séquence « Analyse et traitement des données ».

IV.4.1.4 diagramme de séquence « Stockage des données »

L'opération de stockage des données s'effectue selon les étapes suivantes illustrées dans la figure IV.24 :

- l'utilisateur demande le stockage des données au système qui va la donner au stockage.
- Le système demande le chemin d'écriture de l'utilisateur. Puis, il l'envoie au stockage. Il vérifie par la suite l'état de fichier.
- Si le fichier n'est pas ouvert il vérifie si un fichier est ouvert.
- Si c'est le cas, l'objet stockage donne la décision au fichier soit d'écraser le fichier existant ou de lui faire une mise à jour.
- S'il n'existe pas un autre fichier ouvert, stockage demande de créer le fichier.
- Après vérification d'état, stockage demande au fichier de charger le fichier.
- En suite, le système va demander au stockage de convertir en chaîne de caractère des données acquises et d'écrire le bloc.
- Si l'utilisateur demande de quitter l'application, le système envoie la demande de fermeture de fichier au stockage, ce dernier demande au fichier de sauvegarder et de fermer le fichier.

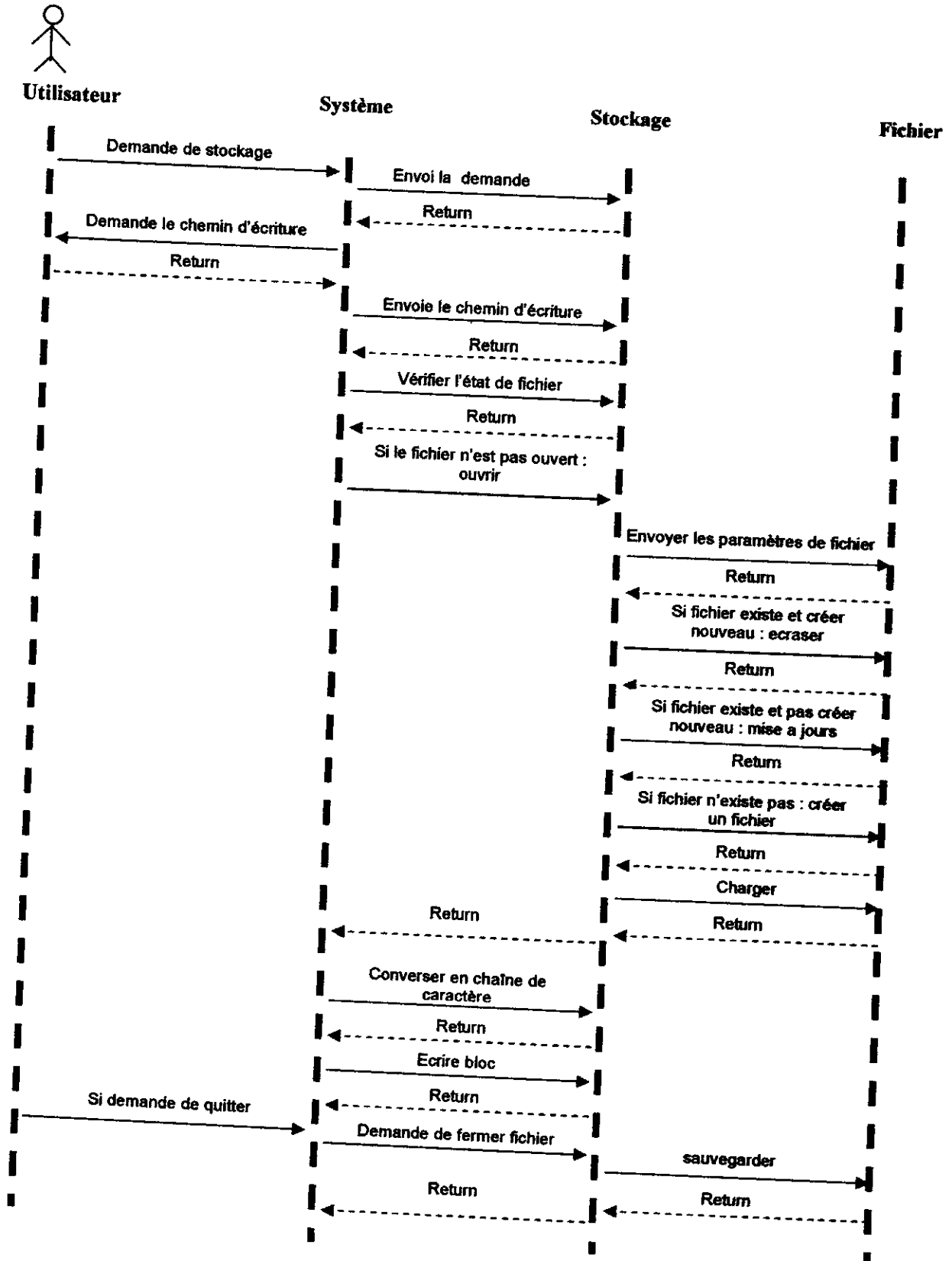
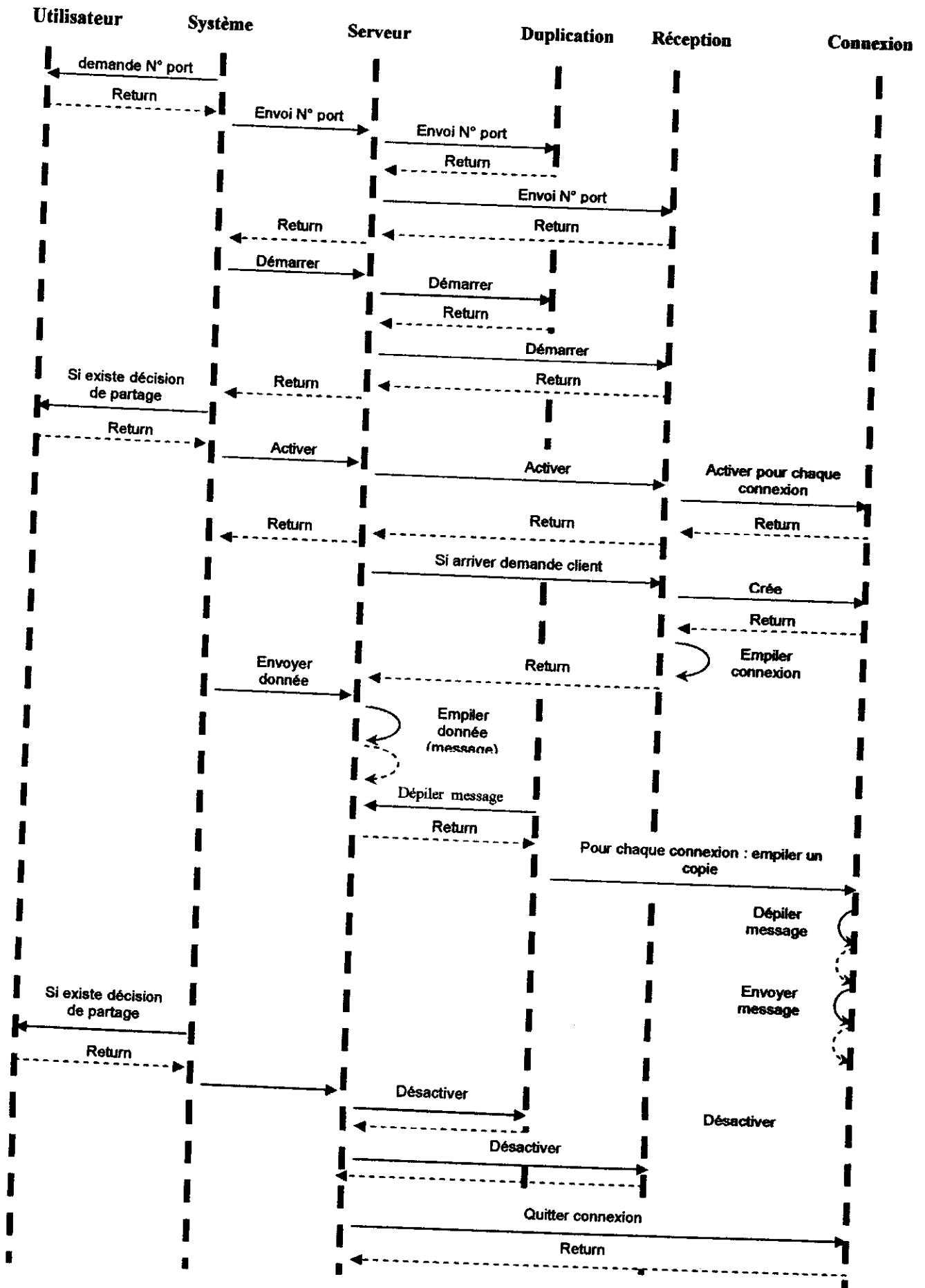


Figure IV.24 : diagramme de séquence « Stockage des données ».

IV.4.1.5 diagramme de séquence « exportation des données » :

Nous présenterons dans la figure IV.25 les différentes opérations effectuées pour transmettre les données.

- Le système demande à l'utilisateur le numéro de port. Ce dernier va être envoyé au serveur qui l'envoie par la suite à la duplication et la réception.
- Le système lance le serveur qui va, de son tour, lancer la duplication et la réception.
- A la suite d'une demande de partage de l'utilisateur le système active le serveur qui va activer la réception qui active les connexions existantes.
- S'il arrive une demande de connexion de la part d'un client, la réception crée une connexion et l'empile dans la liste des connexions.
- Le système envoie les données au serveur sous forme d'un message.
- La duplication procède au dépilement des messages. en suite, il en fait des copies pour chaque connexion, d'où, chaque connexion procède à l'empilement du message.
- La connexion a pour rôle le dépilement et l'envoi du message vers le client.
- Si l'utilisateur décide de désactiver le réseau le système désactive le serveur et par conséquent, la duplication et la réception seront aussi désactivées.
- Si l'utilisateur décide de quitter l'application le serveur sera fermé ainsi que la duplication et la réception.



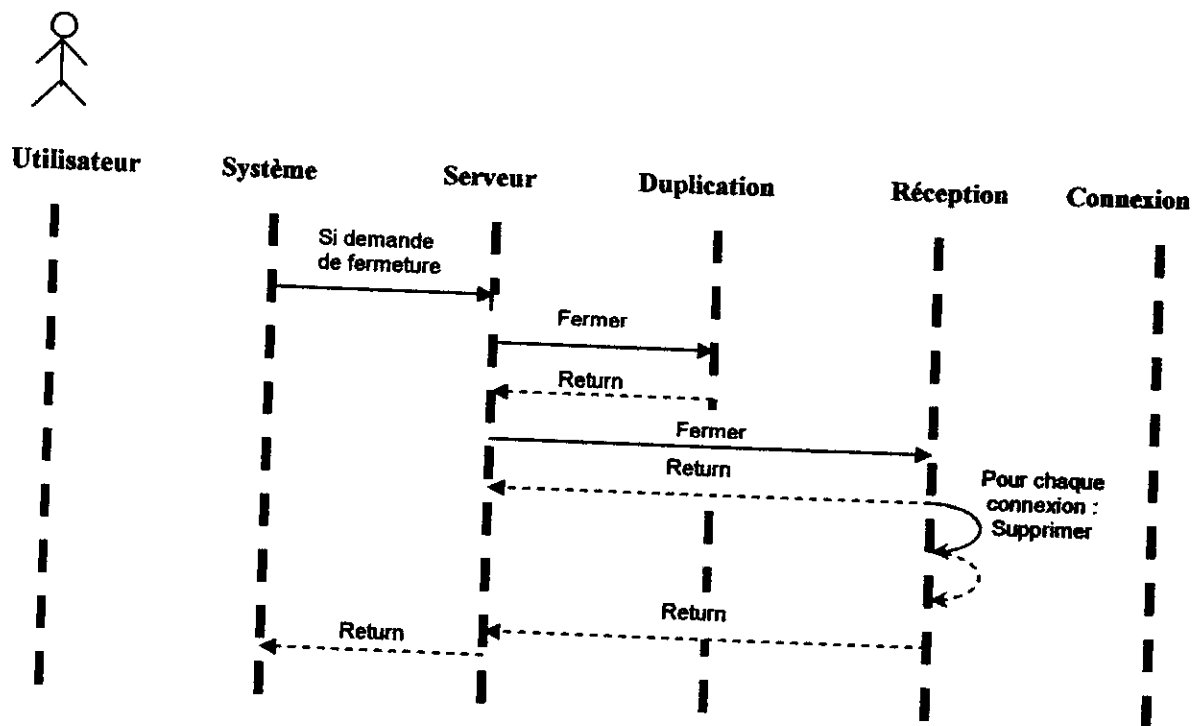


Figure IV.25 : diagramme de séquence « exportation des données ».

IV.4.2 Diagramme de composants :

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en terme de modules : fichiers sources, bibliothèques, exécutables, etc. Ils montrent la mise en oeuvre physique des modèles de la vue logique avec l'environnement de développement.

Les composants de notre application sont organisés en paquetages qui définissent des sous systèmes. Ils permettent de gérer la complexité, par encapsulation des détails d'implémentation.

L'architecture de notre système est composée de deux module ou sous système (figure IV.26) :

- **Cartographie en 2D des signaux EEG** : c'est une application développée sous LabVIEW 7 et sa sortie est de type exécutable. Ce module se charge des

opérations d'acquisition, de traitement, de visualisation et de représentation graphique des signaux issues d'un EEG analogique. Ce module peut fonctionner sur un poste locale. Mais pour qu'il puisse fonctionner sous réseau il a besoin d'un autre module qui se charge de partager les données, c'est le module client/serveur.

- **Client/serveur** : ce module est développé aussi sous LabVIEW 7. il se charge d'envoyer les données de serveur vers l'autre poste client connecté au serveur. les diagrammes de composants représentent l'architecture de notre système qui est composé en deux modules

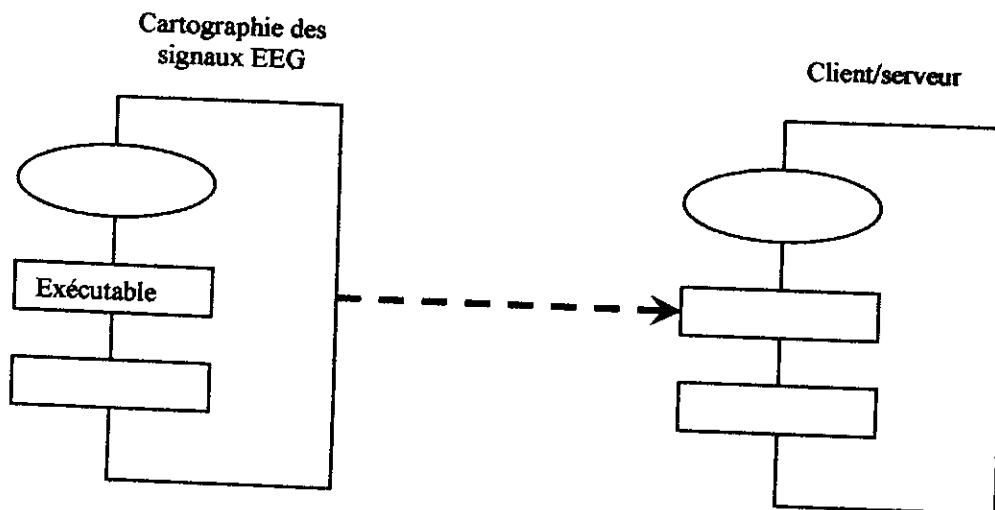


Figure IV.26 : diagramme de composants pour les différents modules du système.

IV. 5 Implémentation et test :

Après avoir détaillé le projet en réalisant différents diagrammes décrivant les phases : spécification des besoins, analyse et conception. On va arriver à la dernière phase qui consiste à implémenter sur PC ce qu'on vient de décrire en utilisant le langage de programmation graphique LabVIEW, et on fini par le test de l'application pour contrôler et améliorer la qualité su système.

IV. 5.1 Langage de programmation LabVIEW :

Le langage utilisé dans notre application est le langage de programmation graphique LabVIEW 7 (voir annexe) qui donne les différents outils nécessaires à la réalisation de la cartographie en 2D des signaux EEG dans une architecture client/serveur.

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) est un logiciel de développement d'applications d'instrumentation, il est plus particulièrement destiné à l'acquisition de données et au traitement du signal. En effet, ce logiciel offre de larges possibilités de communication entre l'ordinateur et le monde physique (par cartes d'acquisitions analogiques ou numériques, réseau, liaisons série et parallèles, etc.) ainsi que d'importantes bibliothèques mathématiques permettant de traiter les signaux mesurés.

LabVIEW est un langage de programmation graphique, il a été développé par la société National Instrument. Il propose un environnement destiné à l'instrumentation où l'on trouve les quatre fonctions de base nécessaires à ces applications industrielles.

1. acquisition et restitution de données :
2. analyse et traitement des données :
3. présentation et stockage des données :
4. exportation des données :

Puisque on travaille sur les cartes d'acquisition de National Instrument (BNC2090 et PCI-MOI-16 E-4), La bibliothèque de VIs DAQ de LabVIEW permet d'utiliser les cartes d'acquisition de données National Instruments.

LabVIEW possède plus de 400 fonctions intégrées spécialement conçues pour extraire les informations pertinentes qui proviennent des données acquises, pour analyser les mesures ou traiter les signaux. Ces fonctions comprennent les FFT, l'analyse de fréquence, la génération de signaux, des fonctions de mathématiques, l'ajustement de courbe et l'interpolation et l'analyse dans les domaines temporel et fréquentiel.

LabVIEW propose un large ensemble d'outils de visualisation pour présenter les données sur l'interface utilisateur de l'instrument virtuel (graphes, graphes déroulants, visualisation 2D et 3D). Vous pouvez instantanément reconfigurer des attributs pour la présentation des données comme les couleurs, la taille de la police, les types de graphes. Vous pouvez également utiliser la souris pour opérer des rotations, faire un zoom avant et arrière sur les graphes. Il ne s'agit pas de programmer vos graphiques et vos attributs en partant de zéro mais seulement d'aller chercher ces objets et de les déposer sur les faces-avant de vos instruments.

LabVIEW inclut des technologies réseau standard comme le TCP/IP et propose des protocoles de communication robustes pour publier et aller chercher les données.

IV.5.2 Description de l'application réalisée:

Un programme LabVIEW est appelé Instrument virtuel (VI : virtual instrument) qui est divisé en deux parties, la première est la face avant (front panel) qui représente l'interface de l'application et la deuxième est le diagramme (block diagram) qui décrit le fonctionnement interne du VI.

IV.5.2.1 la face avant :

L'interface de l'application est représentée dans la figure IV.27, elle permet d'interagir avec l'utilisateur grâce à un ensemble d'objets (boutons, menus, graphes, etc.) ; afin de mieux comprendre l'environnement de l'application on va donner une description des différents objets constituant l'interface.

La face avant est divisée en trois panneaux, le premier panneau se trouve à gauche et contient les objets suivants

1. **Le menu « type d'acquisition »** : qui permet de sélectionner la manière d'acquérir les signaux EEG, pour le serveur on trouve la simulation, la lecture depuis un fichier ou l'acquisition de la carte branchée au PC. Et pour le client on trouve en plus de la simulation et la lecture l'acquisition par réseau.

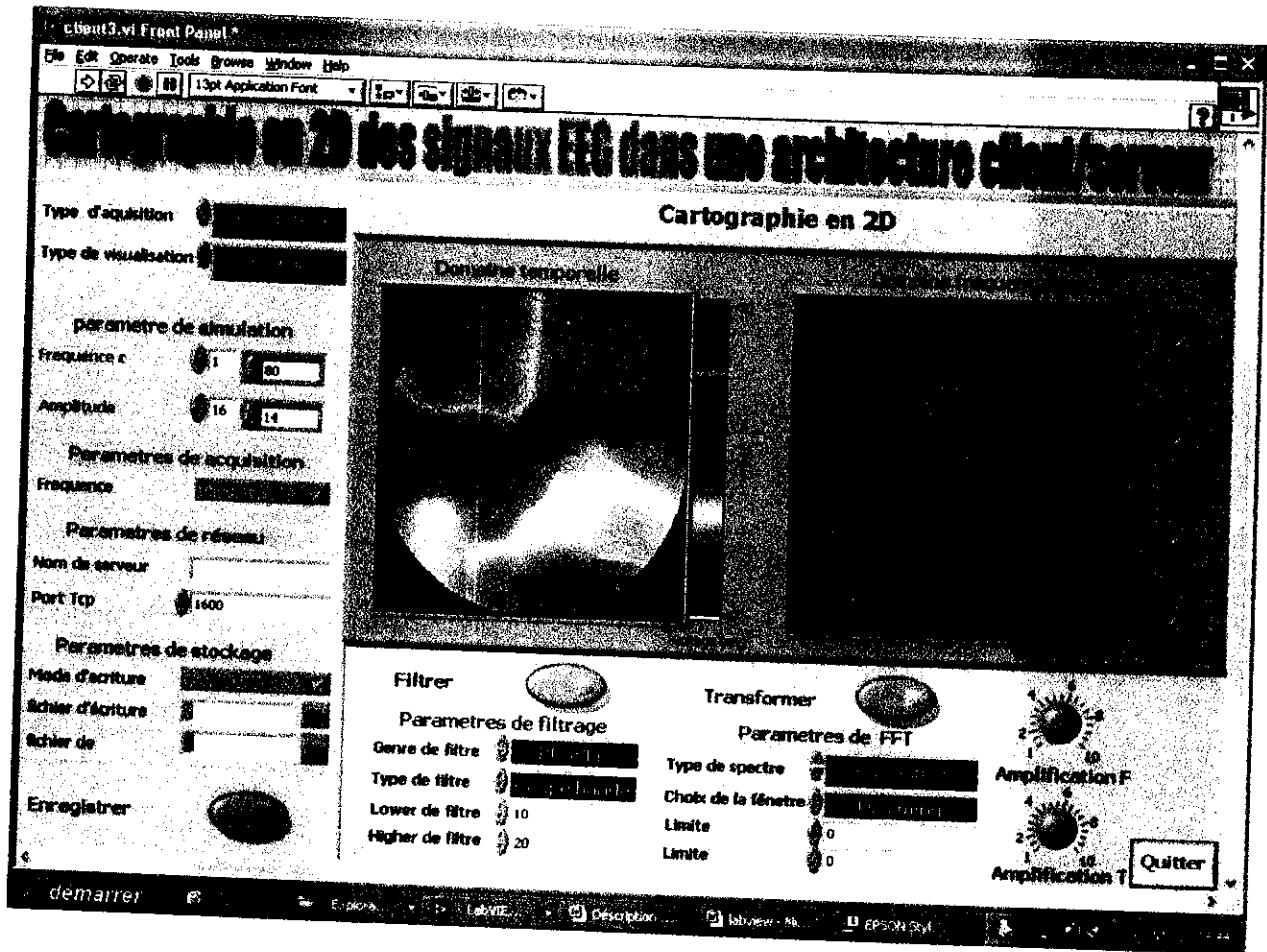


Figure IV.27 : la face avant de l'application.

2. Le menu «Type de visualisation » : qui donne le type de visualisation soit une visualisation individuelle en donnant le numéro de canal, soit une comparaison qui affiche deux signaux pour pouvoir déterminer la différence entre eux, soit le tracé EEG qui permet d'afficher tous les signaux en même temps ou bien la cartographie en 2D des signaux EEG dans les deux domaines fréquentiel et temporelle (figure IV.28).

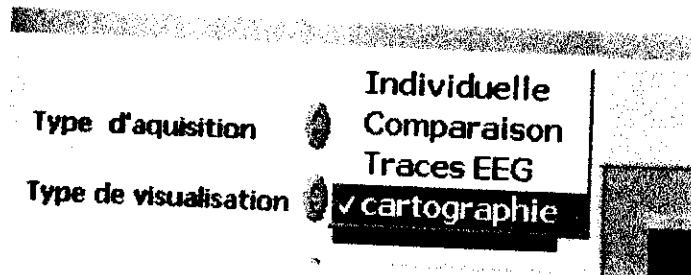


Figure IV.28 : Le menu «Type de visualisation ».

3. **Les paramètre de simulation** : qui nous donne la possibilité de générer des signaux grâce aux différents valeurs de fréquences et d'amplitudes.
4. **Les paramètres d'acquisition** : dans laquelle on donne le fréquence d'échantillonnage.
5. **Les paramètres de réseau** : pour le client on trouve le nom de serveur et le numéro de port (figure IV.29) et pour le serveur on trouve seulement le numéro de port. (figure IV.30).

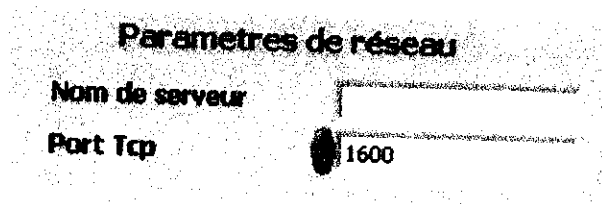


Figure IV.29 : Les paramètres de réseau de l'application client.



Figure IV.30 : Les paramètres de réseau de l'application serveur

6. **Les paramètres de stockage** : ou on trouve le bouton enregistrer qui permet d'effectuer le stockage, le menu « mode d'écriture » y compris ouverture et nouveau, le fichier d'écriture et le fichier de lecture.
7. En fin on trouve le bouton quitter pour quitter l'exécution de l'application.

Le deuxième panneau, en haut à droite, est dédié à l'affichage. Selon le type de visualisation, on a les quatre types d'affichage qui correspond à la visualisation individuelle (figure IV.31), comparaison (figure IV.32), tracée EEG (figure IV.33) et la cartographie en 2D (figure IV.34).

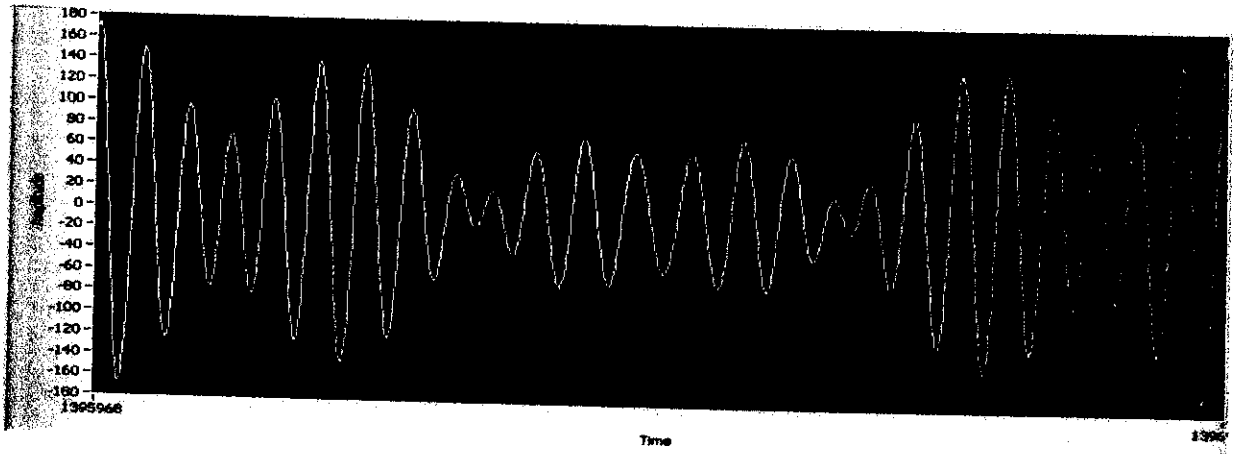


Figure IV.31 : visualisation individuelle.

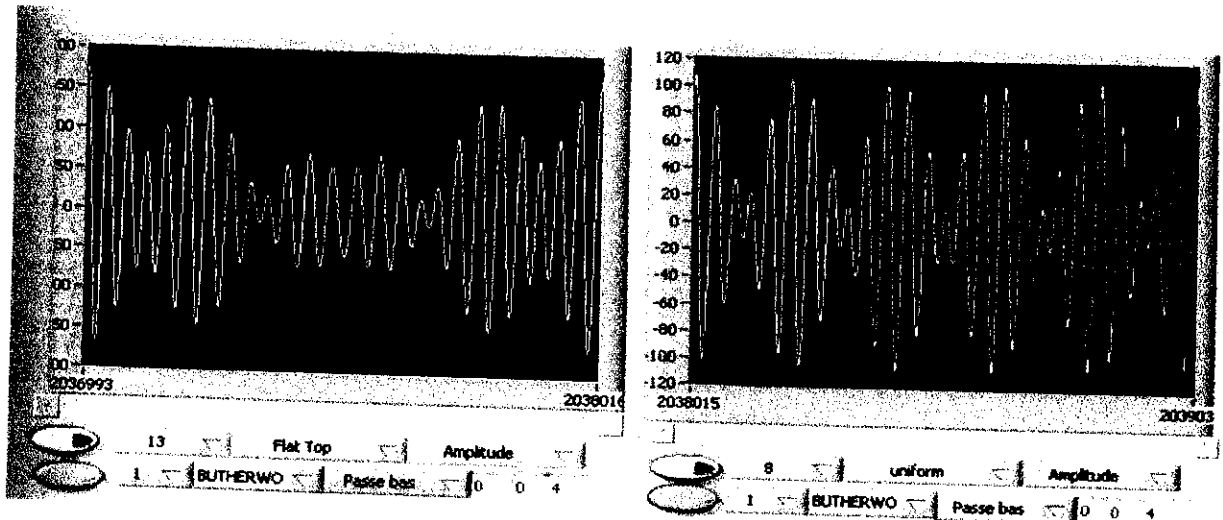


Figure IV.32 : comparaison.

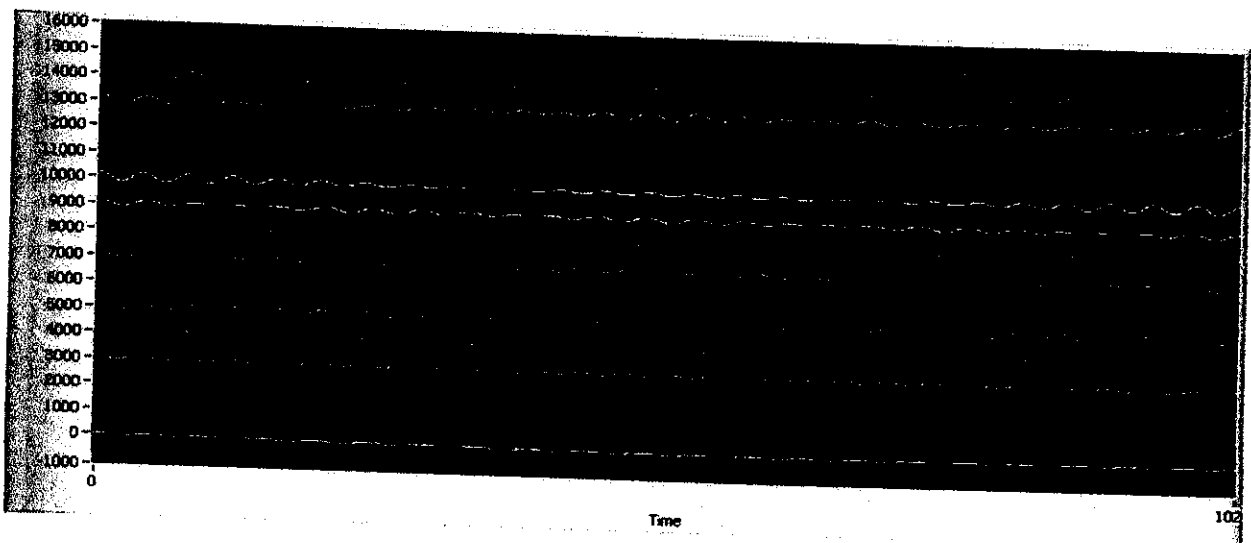


Figure IV.33 : tracée EEG.

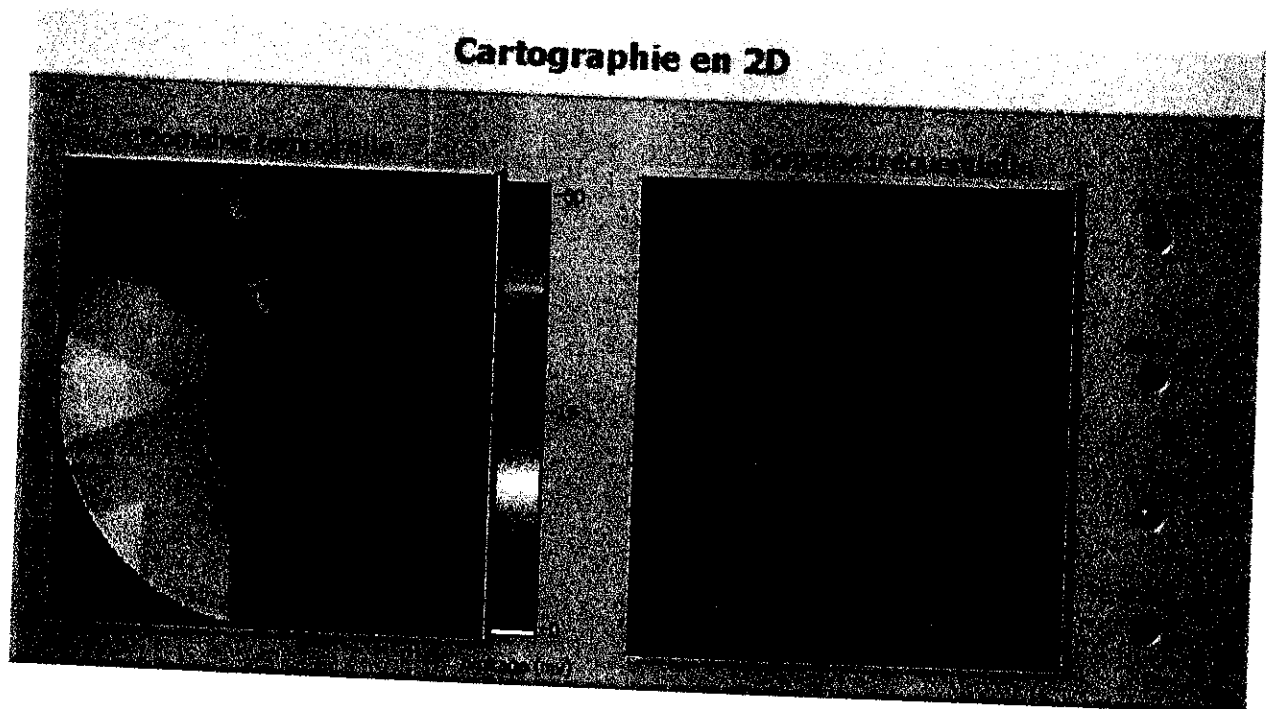


Figure IV.34 : cartographie en 2D.

Le dernier panneau, en bas à droite (figure IV. 35), est consacré au traitement des signaux, il contient les objets suivants :

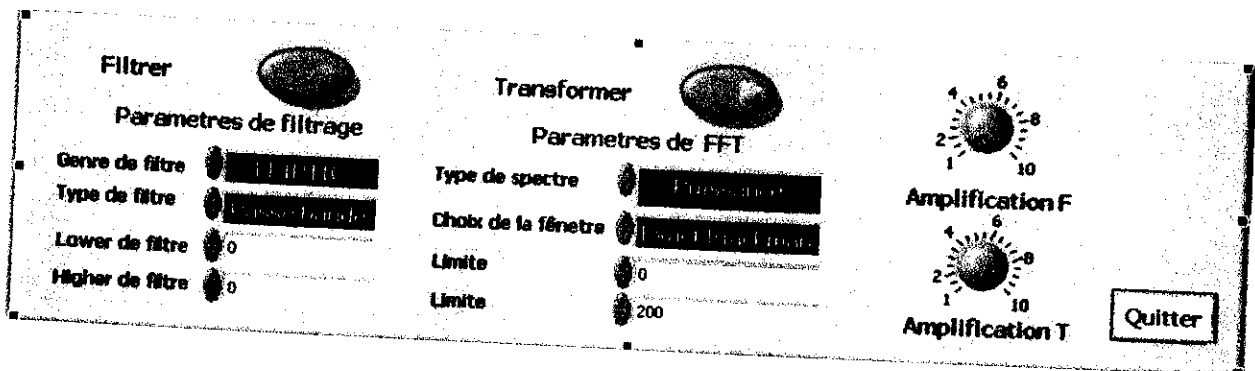


Figure IV. 35 : panneau de traitement des signaux.

1. Le bouton filtrer : il permet d'effectuer l'opération de filtrage d'un signale.
2. le bouton transformer : il permet d'effectuer l'opération de transformation de Fourier d'un signale.

3. Les paramètres vde filtrage : dans la quelle on trouve le menu genre de filtre, le menu type de filtre, lower de filtre et heigher de filtre.
4. Les paramètres de FFT : ou on trouve le menu « type de spectre », le menu « choix de la fenêtre » , limite inférieure et limite supérieur.
5. Deux règle de graduation circulaire : une pour l'amplification temporelle et l'autre pour l'amplification fréquentielle.

IV.5.2.2 le diagramme :

C'est le programme de l'application écrit sous la forme d'un diagramme de flux de donnée, l'application est écrite sous forme modulaire qui consiste en programme principale et un ensemble de sous-VI (sous programmes). Puisque le programme est représenté en image, il s'explique de lui-même et est donc facile à comprendre.

Pour mieux comprendre le corps de programme de l'application nous allons décrire, d'une façon globale, les différentes partis principales constituant le diagramme.

- **L'acquisition :** dans cette partie nous avons utilisé la structure de contrôle Séquence qui donne un ordre précis d'exécution à un nombre quelconque de sous diagrammes, telle qu'elle est représentée dans les figures suivantes.

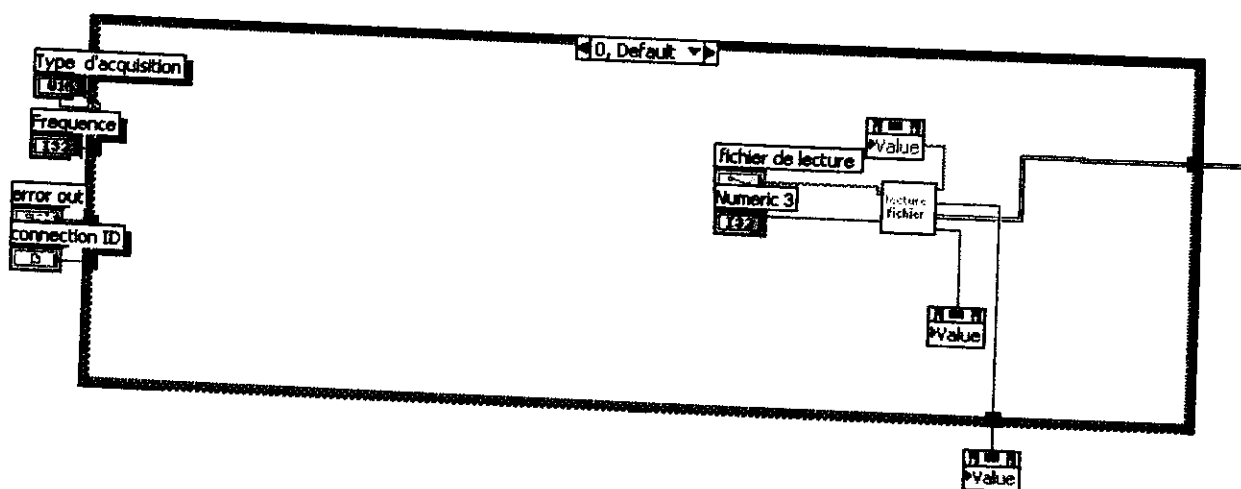


Figure IV. 36 : type d'acquisition « lecture ».

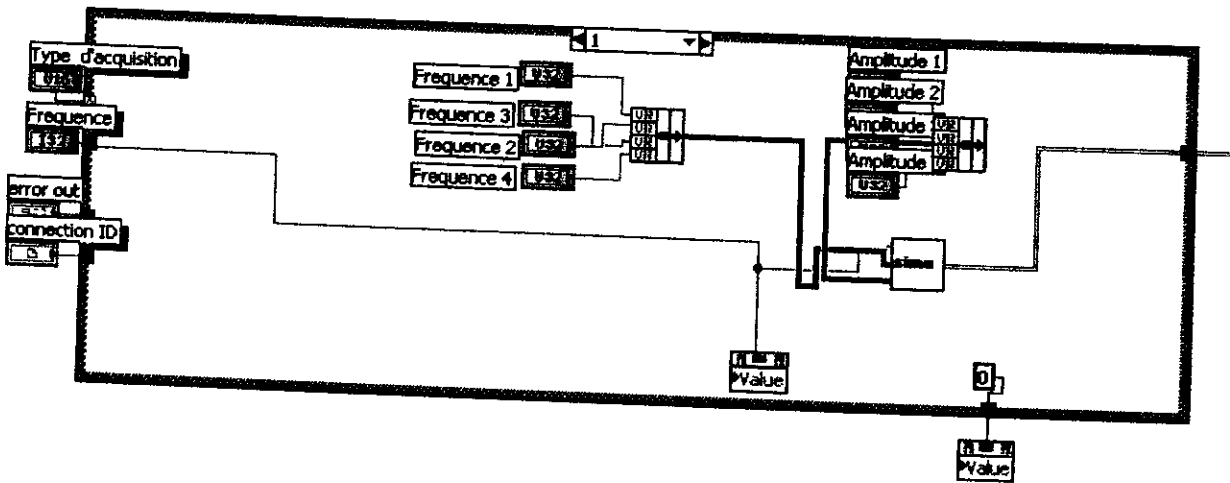


Figure IV. 37 : type d'acquisition « simulation ».

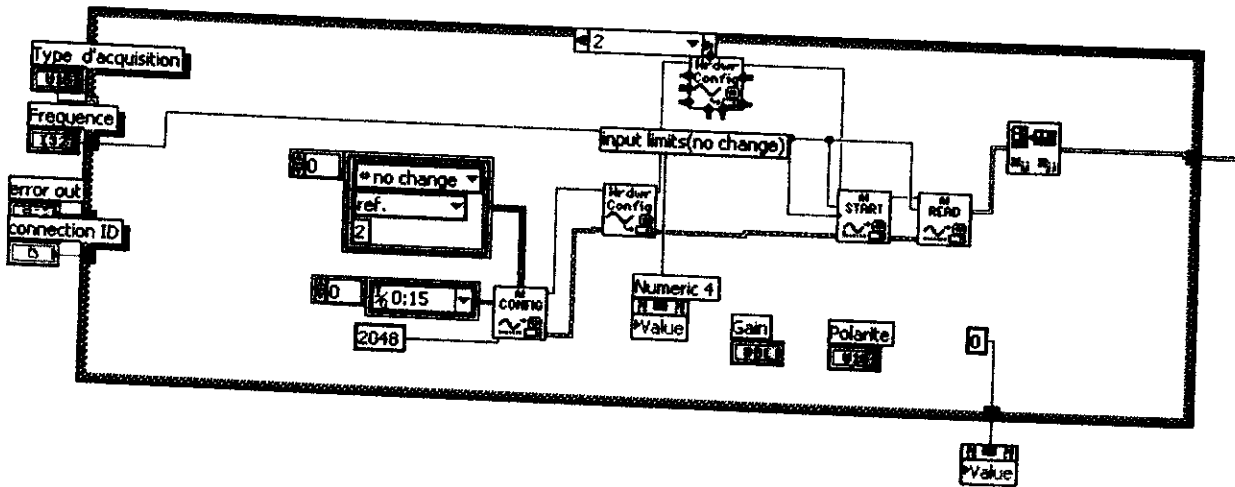


Figure IV. 38 : type d'acquisition « par réseau » pour l'application serveur.

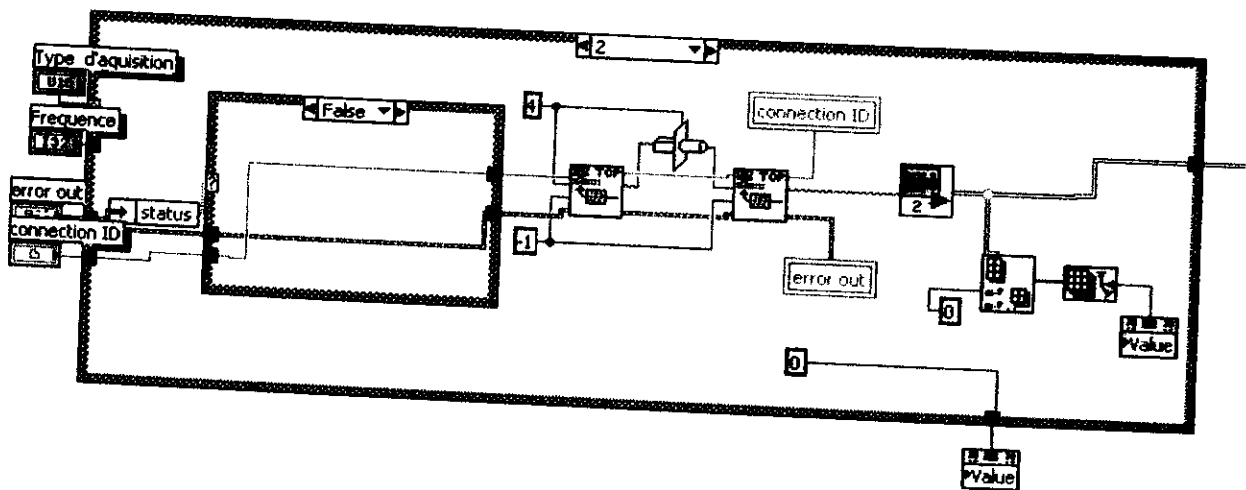


Figure IV. 39 : type d'acquisition « acquisition par carte » pour l'application serveur.

- Le traitement et la visualisation des signaux EEG :

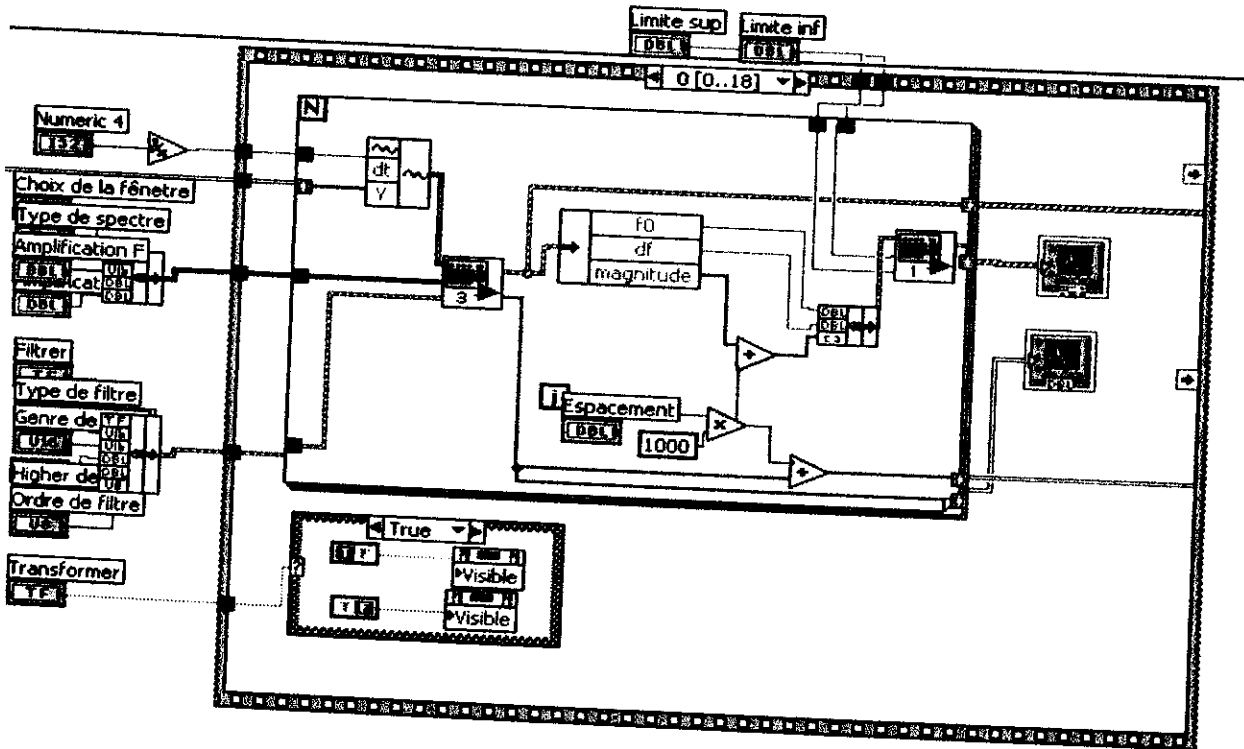


Figure IV. 40 : visualisation individuelle.

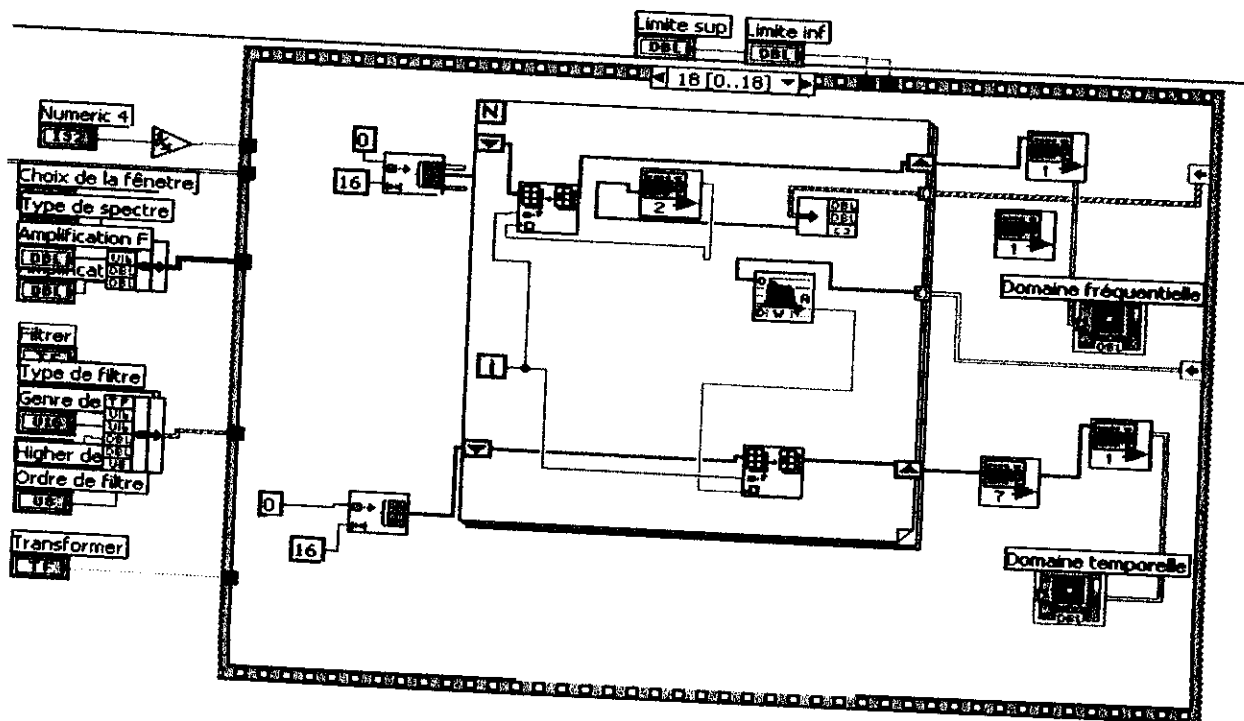


Figure IV. 41 : la cartographie en 2D.

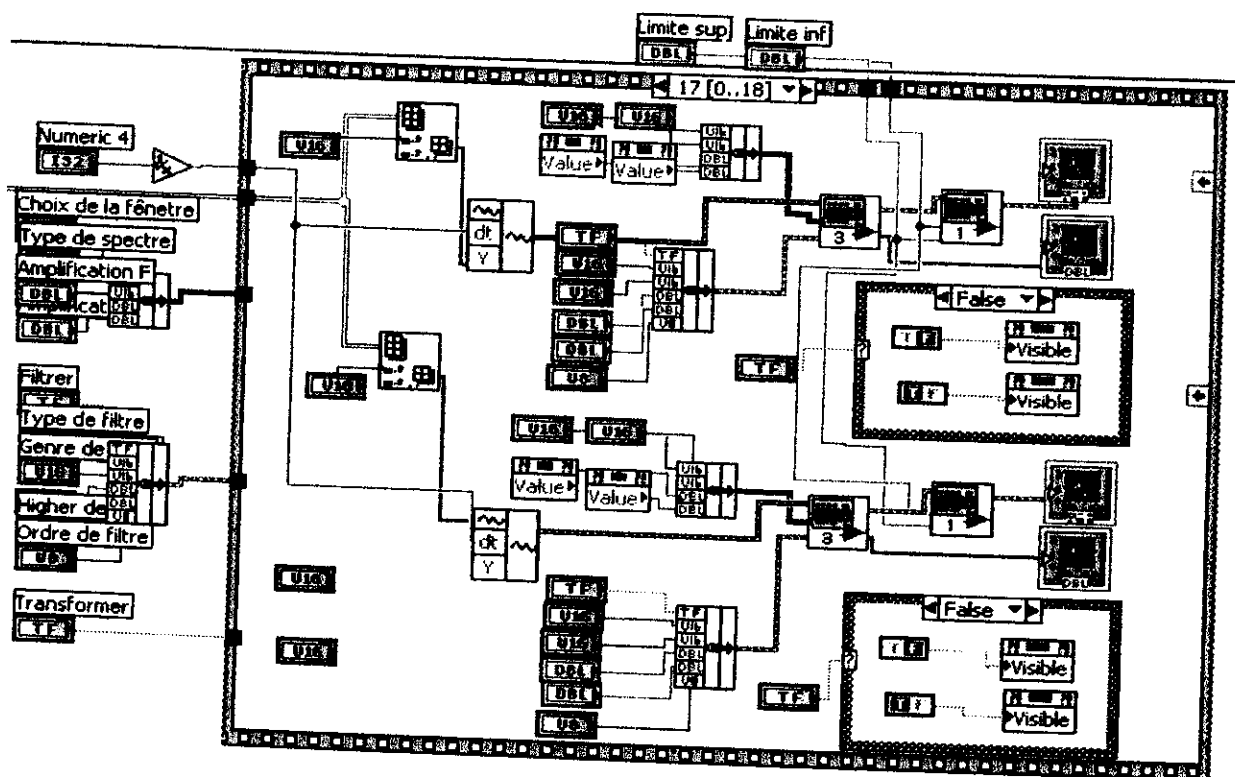


Figure IV. 42 : comparaison.

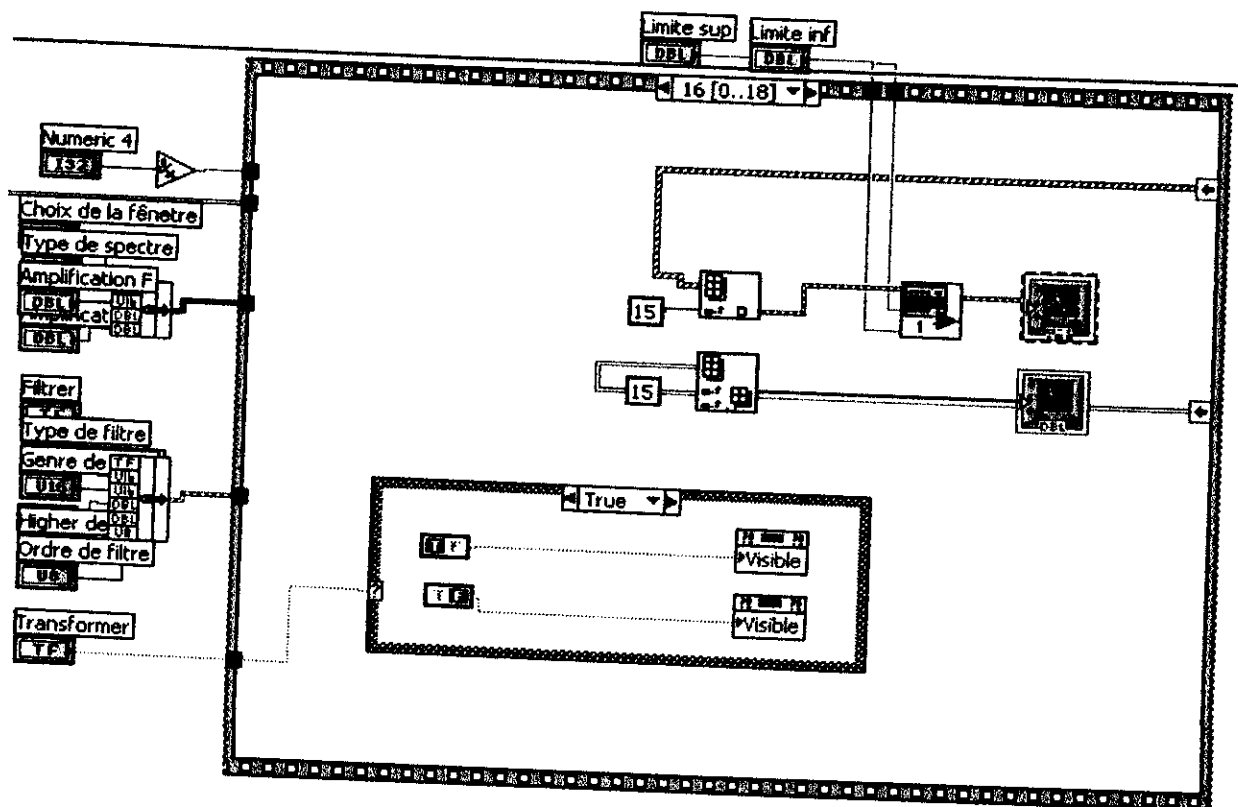


Figure IV. 43 : tracé EEG.

Dans les figures précédentes, nous avons présenté le programme de traitement et la visualisation des signaux EEG en utilisant la structure de contrôle « Séquence ». Ce programme utilise un ensemble de sous VI, qu'on n'a pas représenté, qui nous permet de réduire la taille du programme.

- **L'enregistrement** : dans la figure IV.44, nous avons le programme qui permet de stocker les données et créer, lire et écrire sur les fichiers. CE programme utilise un sous VI (figure IV.45).

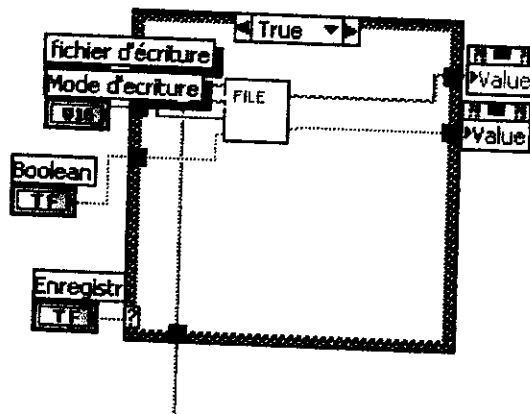


Figure IV. 44 : le programme de stockage des données.

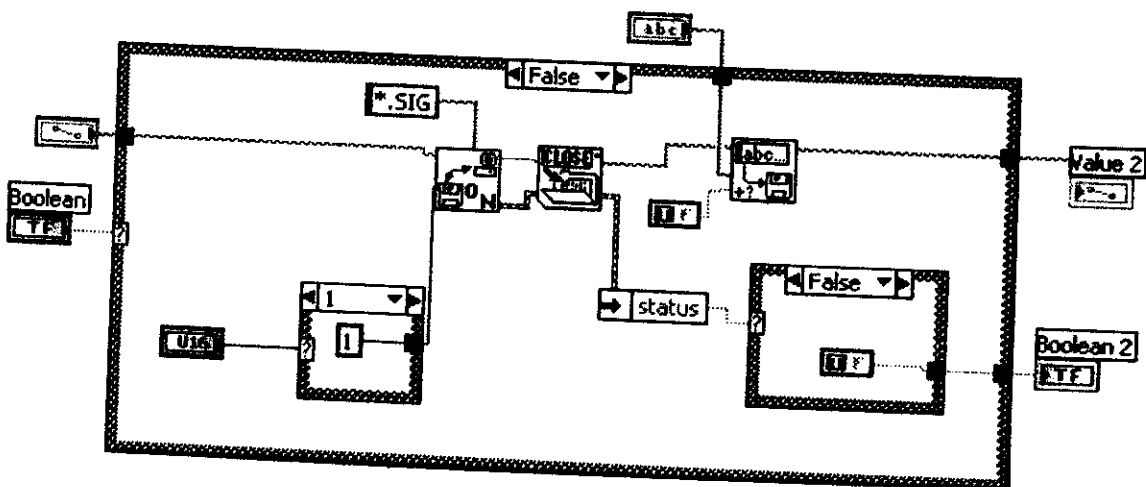


Figure IV. 45 : le sous VI « file ».

- Le partage des données: la parties du diagramme consacré au partage des données est montré dans la figure IV.46. dans laquelle sont utilisée trois sous VI.

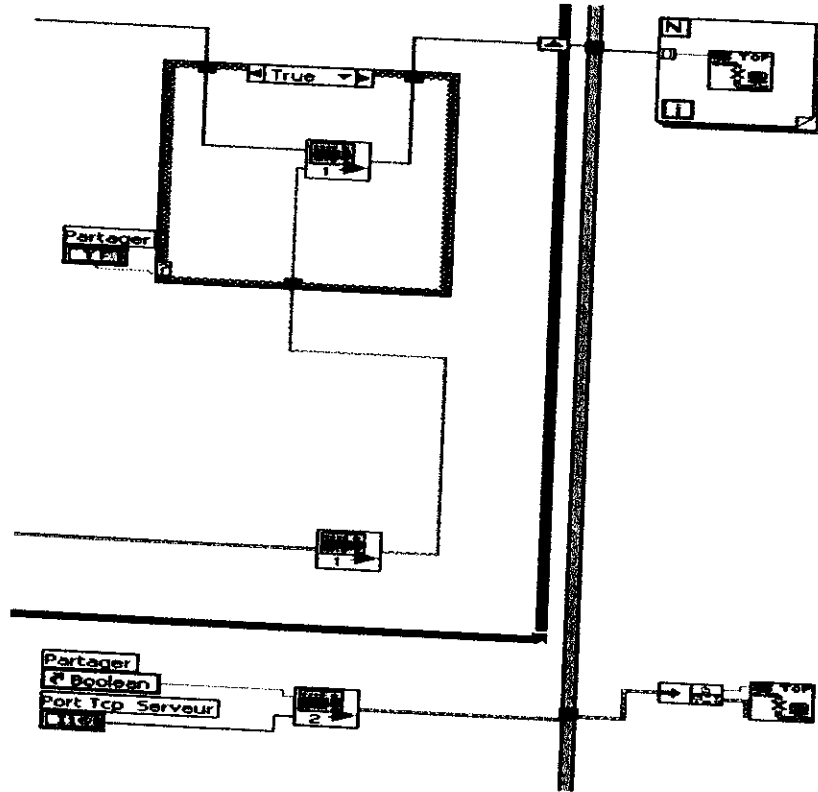


Figure IV. 46 : le partage des données.

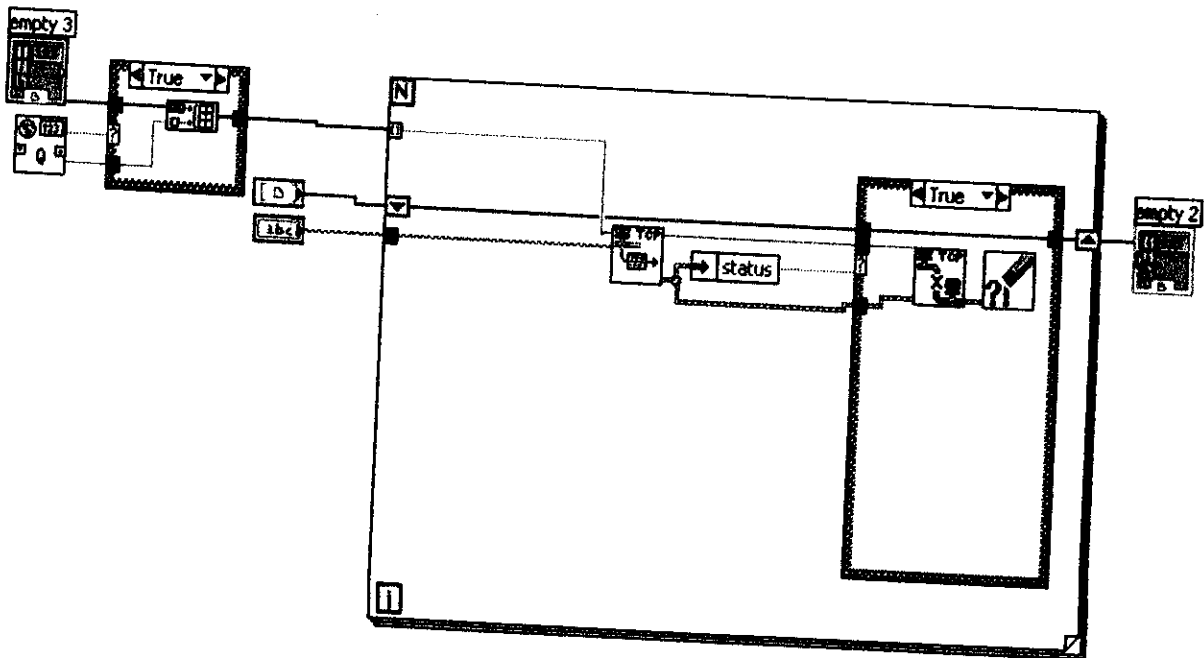


Figure IV. 47 : le sous VI pour l'envoi des données.

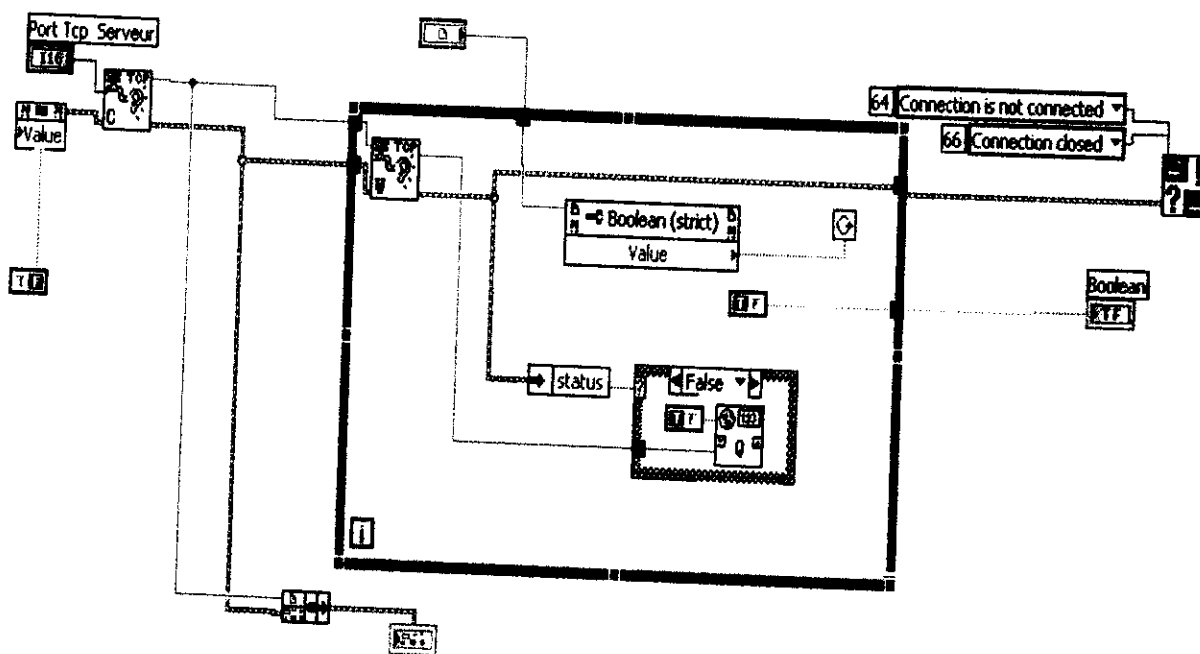


Figure IV. 48: le sous VI « Tcp » pour la création d'une connexion.

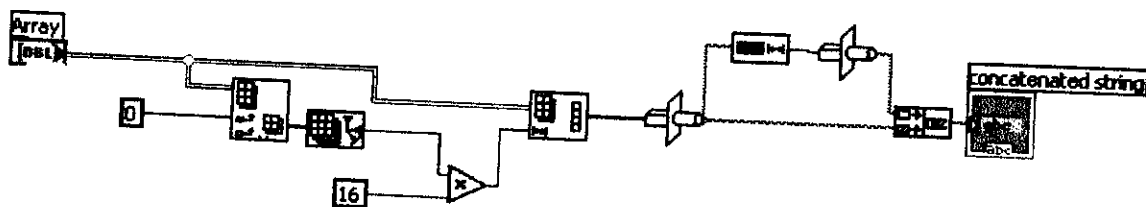


Figure IV. 49: le sous VI pour la conversion en chaîne de caractère.

Conclusion

Tout au long de ce mémoire, nous avons présenté notre projet en deux parties, nous avons commencé par l'aspect théorique qui nous donne les concepts de base de traitement de signale et d'acquisition des données, nous fait connaître le principe d'encéphalographie et nous décrit l'environnement client/serveur à base de protocole TCP/IP. Et pour l'aspect pratique, nous avons décrit les principales étapes de développement du projet grâce au langage de programmation unifié (UML), pour terminer par une brève description de l'application réalisée.

Dans ce projet de fin d'étude, nous sommes arrivés à notre objective qui consiste à réaliser une cartographie en 2D des signaux EEG dans une architecture client/serveur, en passant bien entendu, par toute la chaîne de réalisation d'une application d'acquisition des signaux EEG. Nous avons commencé par l'acquisition des signaux. En suit, le traitement de ces signaux en leur effectuant la fft, le filtrage, etc. Pour finir par une visualisation sur écran en choisissant différentes manières : individuelle, comparaison, tracé EEG, et le plus important la cartographie en 2D dans deux domaines temporel et fréquentiel. Et pour élargir l'utilisation de ce système, il a été développé sous réseau à base de protocole TCP/IP sur une architecture client /serveur.

Ce travail a été développé sous LabVIEW qui offre la souplesse et la puissance d'un langage de programmation sans les difficultés ni la complexité correspondantes puisque sa méthode de programmation graphique fonctionnelle est naturellement familière.

Grâce aux cartes d'acquisition offertes par le centre de développement des technologies avancées (CDTA), nous avons réalisé une acquisition réelle des signaux générés par ces cartes.

Comme perspectives, on aurait aimé implémenter cette application dans un hôpital pour pouvoir avoir des signaux d'EEG réels qui nous permettent d'en tirer les mesures pratiques de fréquences et de tensions qui nous mènent à réaliser une cartographie en 2D plus représentative.

Annexe :**Le langage de programmation graphique LabVIEW****1 Introduction :**

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) est un logiciel de développement d'applications d'instrumentation. Bien que tout à fait utilisable dans un grand nombre de domaines, LabVIEW est plus particulièrement destiné à l'acquisition de données et au traitement du signal. En effet, ce logiciel offre de larges possibilités de communication entre l'ordinateur et le monde physique (par cartes d'acquisitions analogiques ou numériques, réseau, liaisons série et parallèles, etc.) ainsi que d'importantes bibliothèques mathématiques permettant de traiter les signaux mesurés.

L'idée de LabVIEW est de remplacer les instruments de mesures et d'analyse d'un laboratoire par un ordinateur muni de cartes spécifiques et d'un logiciel approprié. Ainsi, un seul ordinateur muni d'une carte d'acquisition analogique et de LabVIEW est capable de remplacer un voltmètre, un fréquencemètre ou un oscilloscope. De plus, on pourra traiter, analyser et archiver sur disque automatiquement les mesures effectuées.

LABVIEW est un langage de programmation graphique, il a été développé par la société National Instrument. Il propose un environnement destiné à l'instrumentation où l'on trouve les quatre fonctions de base nécessaire à ces applications industrielles.

1. acquisition et restitution de données :
2. analyse et traitement des données :
3. présentation et stockage des données :
4. exportation des données :

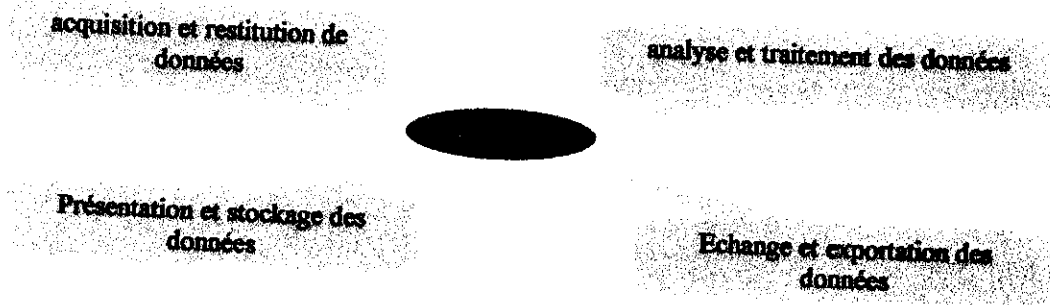


Figure1 : Fonctions de base proposées par l'environnement LabVIEW.

2 Le principe de LabVIEW :

LabVIEW permet de réaliser, entre autre, des instruments virtuels. Par extension on appellera VI (pour *Virtual Instrument*) toute application réalisée avec LabVIEW. Un VI est composé de trois parties liées :

- Une **face-avant (Front-Panel)** : c'est l'interface (ou moyen de communication) avec l'utilisateur. Cette face-avant, personnalisable à loisir, est composée d'objets graphiques comme des interrupteurs, des potentiomètres, des zones de graphismes, etc.. Elle représente la face-avant de l'instrument (figure2).
- Un **diagramme (Block-Diagram)** : cette partie décrit le fonctionnement interne du VI. On utilise le langage G pour décrire ce fonctionnement. Destiné à être utilisé par des ingénieurs et des scientifiques, non informaticiens de formation, LabVIEW utilise un langage de programmation graphique G (pour Graphique) afin de décrire les programmes dictant le comportement de l'application. Ainsi l'utilisateur est affranchi de la lourde syntaxe des langages de programmation textuels tels que C, Basic, etc.. (figure3).
- Une **icône (Icon)** : c'est la symbolisation de l'instrument virtuel qui permettra de faire appel à un instrument virtuel déjà créé à l'intérieur d'un autre instrument virtuel, c'est n principe de structure hiérarchique et de modularité (figure4).

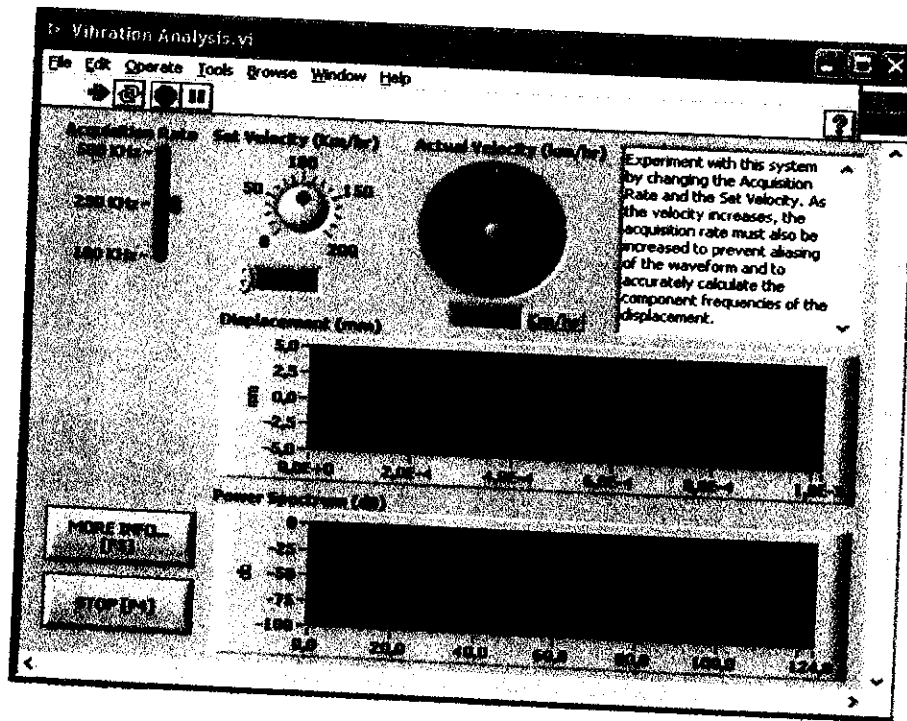


Figure2: face-avant.

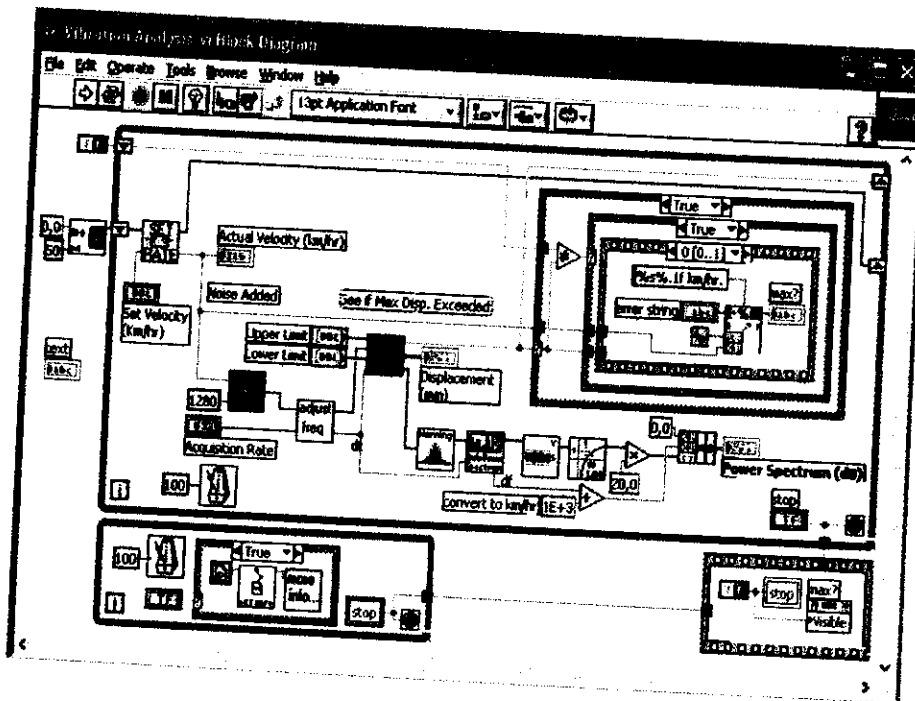


Figure3: diagramme.



Figure 4 : l'icône.

3 Programmation graphique flux de données :

La programmation sous LabVIEW est basée sur les diagrammes flux de données qui permet d'exprimer un programme de manière graphique. Un tel diagramme (Figure 5) est composé de :

- **Terminaux** : ils définissent les entrées (rectangle gras) et les sorties (rectangle fin) de la fonction.
- **Noeuds** : ils définissent les opérations à effectuer. Ils sont représentés par un carré pouvant éventuellement contenir une image illustrant leur fonctionnalité.
- **Arc orientés** : ils relient noeuds et terminaux et permettent d'indiquer le passage de données d'un noeud vers un autre. Par convention, ces arcs sont orientés implicitement de gauche à droite.
- **Jetons** : ils représentent les données transitant sur les arcs. Ils sont représentés par des points.

Des règles d'évolution permettent de faire évoluer ce diagramme :

- A l'initialisation, les terminaux d'entrée produisent chacun un jeton.
- Lorsqu'un noeud possède un jeton sur chacun de ces arcs entrant, le noeud peut être exécuté : chaque jeton en entrée est consommé et le noeud produit un jeton sur chacun de ces arcs sortants.

Un diagramme flot de données peut être encapsulé afin d'être réutilisé, en tant que noeud, par d'autres diagrammes flot de données. Les terminaux du diagramme deviennent alors les entrées/sorties du noeud.

Ainsi le diagramme de la Figure 4.4 exprime le calcul de " $d = a + b * (b - c)$ " et de " $e = b - c$ ". A l'instant considéré, le noeud "+" peut être tiré, le noeud "-" a été tiré. Le noeud "*" ne peut être tiré pour l'instant puisqu'un de ces arcs d'entrée ne possède pas de jeton. Cela sera possible lorsque le noeud "+" aura été tiré et qu'il aura produit un jeton sur son arc de sortie. La Figure 6 représente la forme encapsulée de ce diagramme.

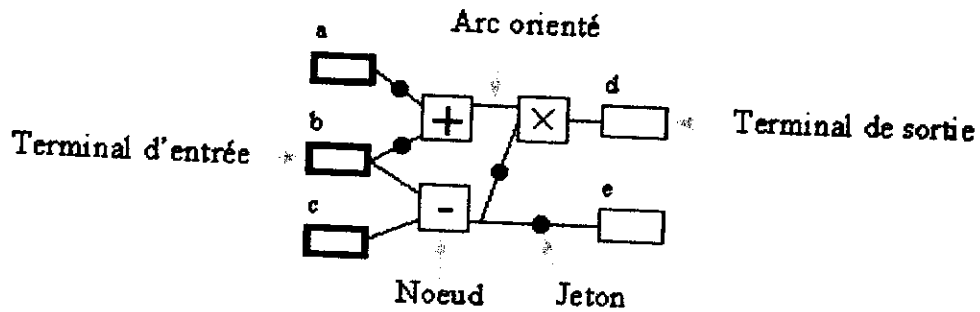


Figure 5 : un diagramme Flot de données.

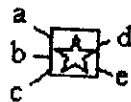


Figure 6 : encapsulation d'un diagramme flot de données.

4 L'environnement de LabVIEW :

Lorsque LabVIEW est lancé, celui-ci ouvre par défaut un nouveau VI. Deux fenêtres vierges apparaissent à l'écran : une face-avant ou *Panel* et un diagramme ou *Diagram* (Figure 7).

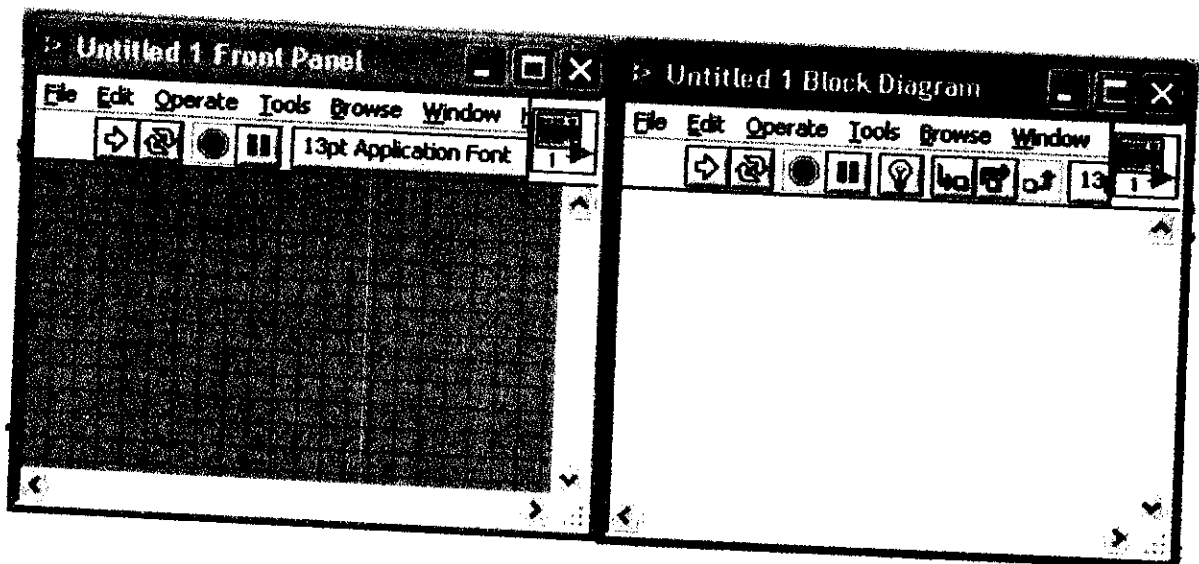


Figure 7: face-avant et Diagramme d'un VI.

Chacune de ces fenêtres est composée de :

- Une barre de titres, contenant le titre du VI. L'extension *Diagram* indique que la fenêtre considérée est le diagramme. La fenêtre dont la barre de titres est bleue indique que celle-ci a le contrôle, c'est-à-dire que c'est à elle que s'adressent les informations provenant du clavier ou de la souris. On donne le contrôle à une fenêtre en cliquant dessus.

- Une barre de menus déroulants. Ceux-ci permettent de gérer la sauvegarde des programmes, de gérer les différentes fenêtres, etc.
- Une barre d'exécution/édition (Figure 8) se retrouve dans chacune des fenêtres et permet de lancer l'exécution d'un VI, de l'interrompre, etc. Cependant, on ne retrouve les icônes de mise au point que dans la fenêtre du diagramme.

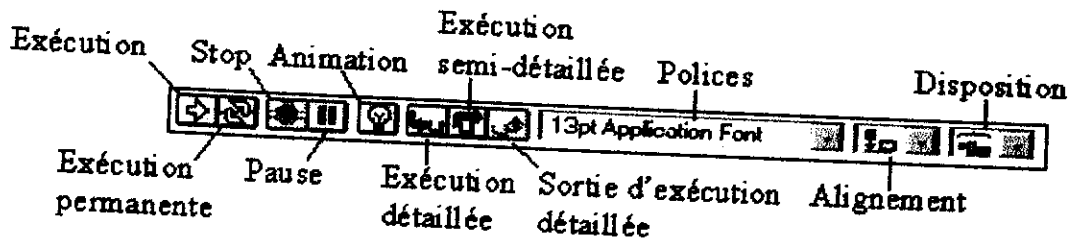


Figure 8 : la barre d'exécution/édition.

Différentes fenêtres permettent de créer des VIs :

- **La boîte à outils** : elle va permettre de manipuler les objets graphiques créés.
- **La fenêtre Controls** : elle va permettre de créer des objets graphiques sur la face-avant.
- **La fenêtre Functions** : elle va permettre de créer des objets graphiques dans le diagramme.

On appelle menu local, le menu qui apparaît lorsque l'on clique avec le bouton droit de la souris dans une fenêtre en maintenant le bouton enfoncé. Par exemple, lorsque l'on ouvre un menu local sur un objet, un menu contextuel apparaît permettant de changer les propriétés de cet objet. Lorsque l'on ouvre un menu local dans une face-avant (resp. diagramme) une fenêtre Controls (resp. Functions) apparaît.

4.1 La boîte à outils :

On fait apparaître la boîte à outils (Figure 9) permettant de manipuler les objets graphiques, en suivant la séquence de menu **Windows >> Show Tools Palette** ou bien en ouvrant un menu local dans la fenêtre avec la touche **SHIFT** enfoncée. Une fois apparue, la boîte à outils est toujours visible. On sélectionne l'outil désiré en

cliquant dessus. Il est à noter que la touche de tabulation permet une sélection rapide des outils les plus utilisés.

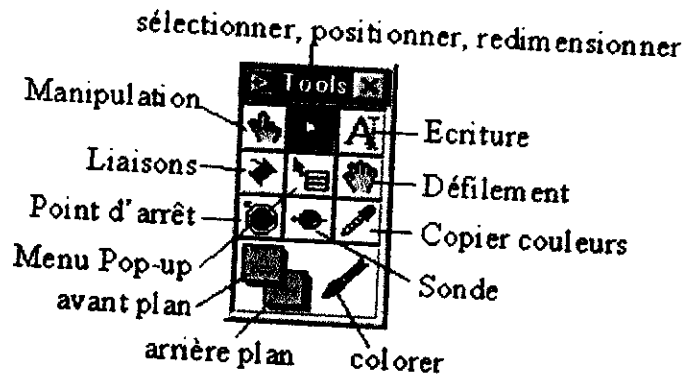


Figure 9 : La boîte à outils.

- outil Doigt : permet de positionner des éléments des palettes Controls et Functions sur la face-avant du diagramme.
- outil Flèche : permet de positionner, redimensionner et sélectionner les objets.
- outil Texte : permet de modifier du texte et d'en créer.
- outil Bobine : permet de câbler des objets entre eux dans le diagramme.
- outil Menu local : fait apparaître un menu local dans un objet.
- outil Main : fait défiler toute la fenêtre sans avoir recours aux barres de défilement.
- outil Point d'arrêt : permet de définir des points d'arrêt dans les VIs, les fonctions, les séquences et les structures.
- outil Sonde : permet de créer des sondes sur les fils.
- outil Pipette : copie les couleurs pour les coller à l'aide de l'outil Pinceau.
- outil Pinceau : permet de définir la couleur de l'arrière-plan et du premier plan.

4.2 La fenêtre Controls :

Lorsqu'une fenêtre de type face-avant est active, on peut faire apparaître une fenêtre nommée Controls (Figure 10) permettant de positionner des objets sur la

face-avant en suivant la séquence de menu Windows >> Show Controls Palette ou bien en ouvrant un menu local dans la fenêtre.

Une fenêtre de sous-menus contenant des objets graphiques apparaît lorsque l'on clique sur les icônes de cette fenêtre.

La sélection d'un objet dans un de ces sous-menus permet de le disposer dans la face-avant et fait automatiquement correspondre un terminal dans le diagramme.

Cette fenêtre n'apparaît que lorsqu'une fenêtre de type face-avant est active : si une fenêtre de type diagramme est active, cette fenêtre disparaît.

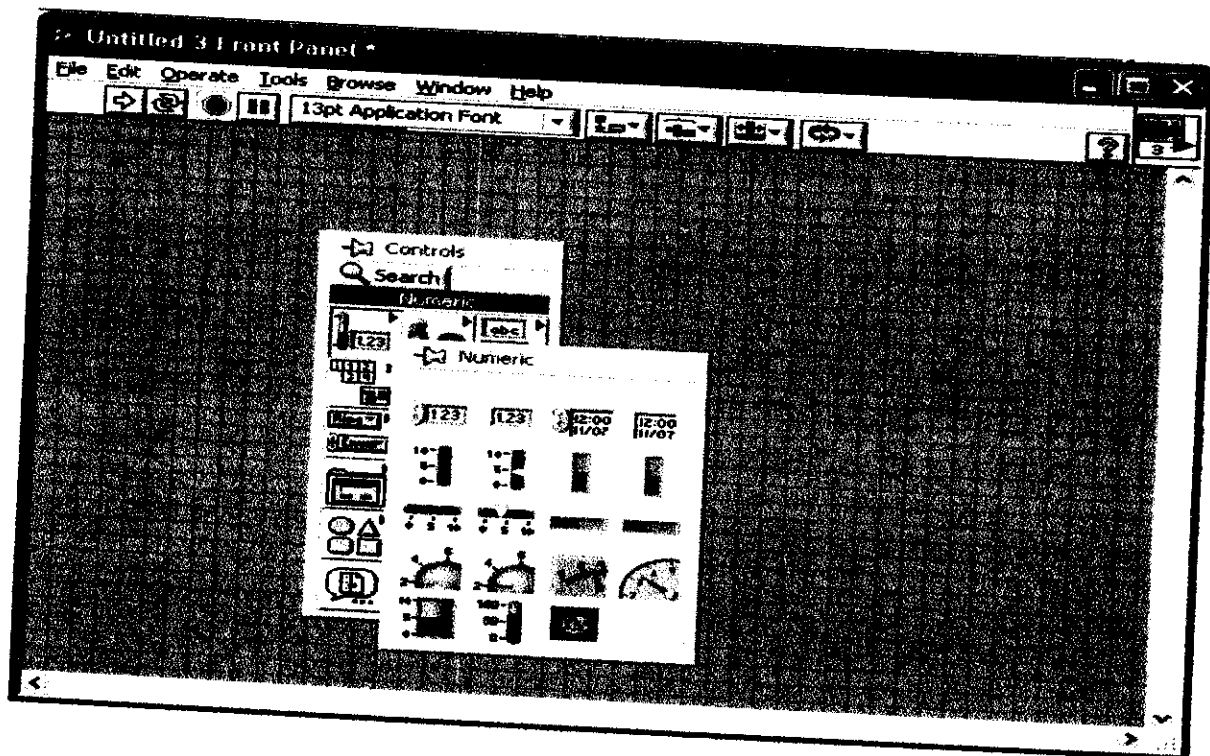


Figure 10 : La fenêtre Controls.

4.3 La fenêtre Functions :

De la même manière que pour la face-avant, lorsqu'une fenêtre de type diagramme a le contrôle, on peut faire apparaître une fenêtre nommée Functions (Figure 11) permettant de positionner des objets dans le diagramme (diagram) en suivant la séquence de menu Windows >> Show Functions Palette ou bien en ouvrant un menu local dans la fenêtre.

Une fenêtre de sous-menus contenant des noeuds (fonctions) apparaît lorsque l'on clique sur les icônes de cette fenêtre.

La sélection d'un objet permet de le disposer dans le diagramme.

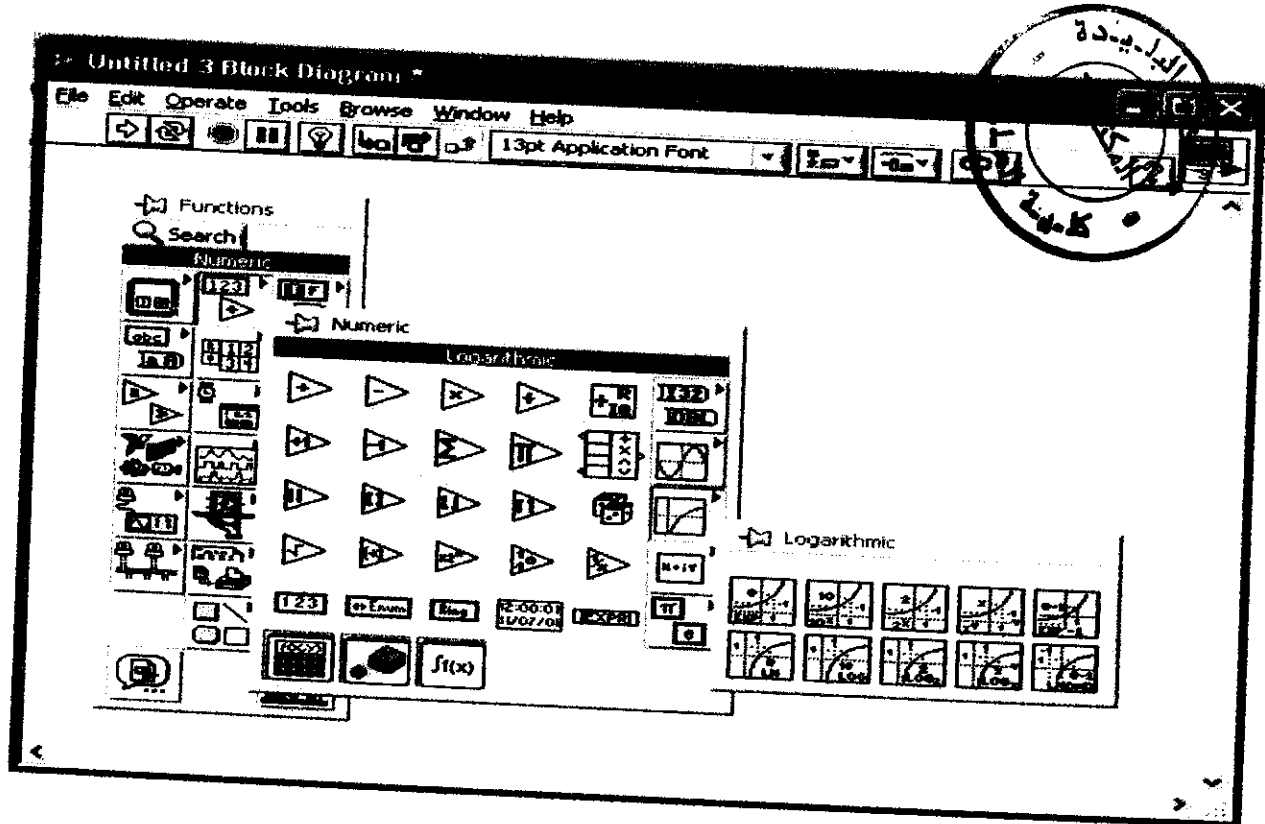


Figure 11 : la fenêtre Functions et ses sous-menus.

5 Conclusion :

La bibliothèque de VIs DAQ de LabVIEW permet d'utiliser les cartes d'acquisition de données National Instruments. LabVIEW peut aussi s'interfacer avec d'autres matériels industriels, des automates programmables, des enregistreurs, des régulateurs ainsi que les périphériques de conditionnement du signal et d'acquisition de données distribuée de chez National Instruments. L'Assistant Solutions DAQ est un utilitaire capable de générer automatiquement vos VIs de mesure et de contrôle en fonction des renseignements que vous avez fournis. Après cette étape, il ne reste en général qu'à personnaliser le VI généré pour qu'il réponde parfaitement à votre besoin. Notez que de nombreux exemples de mesure et de contrôle s'installent sur votre ordinateur en même temps que LabVIEW.

LabVIEW possède plus de 400 fonctions intégrées spécialement conçues pour extraire les informations pertinentes qui proviennent des données acquises, pour analyser les mesures ou traiter les signaux. Ces fonctions comprennent les FFT, l'analyse de fréquence, la génération de signaux, des fonctions de mathématiques, l'ajustement de courbe et l'interpolation et l'analyse dans les domaines temporel et fréquentiel. Présenter. LabVIEW possède aussi de puissantes bibliothèques d'analyse, aussi performantes que celles de logiciels dédiés. Ces bibliothèques complètes permettent par exemple de faire des statistiques, des calculs polynomiaux, de l'analyse fréquentielle, du fenêtrage, du filtrage numérique, ...

LabVIEW propose un large ensemble d'outils de visualisation pour présenter les données sur l'interface utilisateur de l'instrument virtuel (graphes, graphes déroulants, visualisation 2D et 3D). Vous pouvez instantanément reconfigurer des attributs pour la présentation des données comme les couleurs, la taille de la police, les types de graphes. Vous pouvez également utiliser la souris pour opérer des rotations, faire un zoom avant et arrière sur les graphes. Il ne s'agit pas de programmer vos graphiques et vos attributs en partant de zéro mais seulement d'aller chercher ces objets et de les déposer sur les faces-avant de vos instruments.

LabVIEW inclut des technologies réseau standard comme le TCP/IP et propose des protocoles de communication robustes pour publier et aller chercher les données.

BIBLIOGRAPHIE

1. Francis Cottet, " LabVIEW Programmation et application", DUNOD, 2002.
2. Francis Cottet, " traitement des signaux et acquisition de données", DUNOD, 2000.
3. Pascal Roques, "UML2 par la pratique Etude de cas et exercices corrigés", EYROLLES, 2004.
4. BUSBY Milk, "introduction à TCP-IP", INTEREdition, 1991.
5. Renato M.E. Sabbatini, Mapping the Brain. Brain & Mind Magazine, Septembre/Novembre 1997 téléchargeable sur le site www.cerebromente.org.br/n03/tecnologia/eeg.htm
6. DEVAUBLANC Olivier, " installez un petit réseau de PC", DUNOD, 2003, Paris.
7. l'encyclopédie Vulgaris-Médical, © 2000-2006 Vulgaris.
8. PENDERS Tom, "introduction à UML", OEM,2002.
9. HUNT Crang, "TCP-IP: Administrateur de réseau", O'REILLY, Paris, 2002.
10. Raïs El'hadi BEKKA, "fondement du traitement du signal", Office des publications universitaires, Alger, 1998.
11. BELKAID Ismail, "Conception et réalisation d'un moniteur EEG en mode Client/serveur", Mémoire PFE ingénieur d'état en informatique, Option système d'information, Département d'informatique, USDB, Session October 2005.
12. Mme KAOUA.YKRELEF.Ikram. "Représentations graphiques en trois dimensions des signaux EEG séparés par la méthode d'analyse en composantes indépendant", Mémoire de magister, Spécialité Communication, Département d'électronique, USDB, juillet 2005.

