Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Acquisition et traitement de potentiels évoqués

Glowacki, Anne; Meurisse, Hubert

Award date:
1991

Awarding institution:
Universite de Namur

Link to publication

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

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

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

Download date: 04. Jul. 2025

Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique

Rue Grandgagnage, 21
B-5000 NAMUR (Belgium)

Acquisition et traitement de potentiels évoqués

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

Anne Glowacki
Hubert Meurisse

Promoteurs : J-P. Leclercq
J.P. Peters

Année académique 1990-1991

RESUME

Depuis quelques années, un grand nombre d'appareillages scientifiques évoluent vers la conversion digitale de l' information. Il est devenu pratiquement nécessaire de donner à ces techniques les moyens informatiques actuels: traitement mathématique et logique des données, centralisation du contrôle des processus , facilité de stockage et de restitution de l'information, possibilité de banque de données scientifiques, affichage et impression de qualité...

L'objet de ce mémoire est d'effectuer, pour le problème médical particulier des "Potentiels Evoqués", le transfert d'une technologie en grande partie analogique vers une technologie digitale.

L'originalité du travail réside dans la volonté d'adapter les méthodes de développement de projet, classiquement réservées à la gestion, à une application en temps réel.

L'implantation physique est réalisée sur Personnal Computer accompagné d'une carte d'interface prototype. Ceci permet la communication entre l'unité d'acquisition des mesures et le système d'information.

ABSTRACT

For the last few years a large amout of scientific devices have evolved towards digital conversion of information. It has now become almost necessary to give these techniques the present computer means : mathematical and logical data processing, centralization of process control, the opportunity of storing and giving back information, the feasibility of a scientific data bank, high quality display and printing...

The purpose of this dissertation is, for this particular medical problem of evoked potentials, to pass from a mainly analogical technology to a digital one.

The originality of this work lies in the will of adapting methods used for developing projects to real time implementation, these methods being usually kept for management.

the physical setting up is carried out on PC with a prototype interface card, which allows communication between the acquisition unit of measurement and the system of information.

Nous tenons ici à exprimer nos plus vifs remerciements à nos promoteurs messieurs J.P. Peters et J.P. Leclercq pour l'assistance qu'ils nous ont fournie ainsi que pour leur grande disponibilité tout au long de cette année d'étude.

Nous tenons également à remercier le professeur E. Dubois pour ses conseils ainsi que monsieur R. Mairesse pour le soutien logistique dont il a toujours bien voulu nous entourer.

Enfin, nous remercions toutes les personnes qui ont participé de près ou de loin à l'achèvement de ce travail.

TABLE DES MATIERES

I. LES POTENTIELS EVOQUES. 1

I.1. Introduction..... 1

I.2. Comment représenter ces potentiels évoqués. 2

I.3. Les techniques de moyennage : Détection de signal noyé dans un bruit..... 2

I.4. Rejet d'artefacts. 5

I.5. Nomenclature des potentiels évoqués..... 6

I.6. Types de potentiels évoqués..... 6

I.6.1. Potentiels évoqués auditifs..... 6

I.6.2. Potentiels évoqués somesthésiques 6

I.6.3. Potentiels évoqués visuels 7

I.6.4. Variations de contingences négatives:..... 7

1.7. L'interprétation des Potentiels Evoqués..... 7

I.7.1. Anomalies du temps de latence..... 8

I.7.2. Anomalies d'amplitude des réponses 8

I.7.3. L'altération de la topographie des réponses..... 8

I.8. Conclusion..... 10

II. ETUDE DU PROJET 11

II.1.Les aspects généraux de l'acquisition. 11

II.2. Cahier général des charges..... 14

II.2.1 Système de stimulation..... 14

II.2.1.1.Le module des stimulations auditives..... 15

II.2.1.2. Le module des stimulations visuelles. 15

II.2.1.3.Le module des stimulations somesthésiques..... 16

II.2.2. Le système de captation d'amplification et de filtrage de signaux..... 16

II.2.3 Un système de conversion analogique-digital..... 17

II.2.4 Un système de traitement du signal afin de produire des résultats interprétables 17

II.2.4.1. Le moyennage. 17

II.2.4.2 Processus en parallèle..... 17

II.2.4.3. Le déclenchement ("triggering")..... 18

II.2.4.4. La fenêtre d' analyse..... 18

II.2.4.5. La récurrence de stimulation..... 18

II.2.4.6. Le nombre de stimulations..... 18
II.2.5 Les artefacts. 18
II.2.6. Le traitement graphique. 19
II.2.7. Le traitement mathématique..... 20
II.2.8. Impression des résultats..... 20
II.3.Première structuration du projet..... 24
II.3.1. L'acquisition.(figure 2.4)..... 24
II.3.2. Les traitements différés..... 27

III. ANALYSE FONCTIONNELLE.....29
III.1. Le schéma conceptuel des informations..... 29
III.1.1.Représentation de la structure du système d'information - Schéma Entité/Association 29
III.1.2: le schéma entité/association 32
III.2. Elaboration du schéma conceptuel des traitements..... 33
III.2.1 Acquisition..... 34
III.2.1.1 Phase d'initialisation de l'unité d'acquisition. 34
III.2.1.2 Phases de conversion digitale et transfert en mémoire..... 38
III.2.1.3 Phase de rejet des artefacts..... 39
III.2.1.4 Phase de moyennage des signaux..... 39
III.2.1.5 Phase d'affichage courant..... 41
III.2.1.6 Phase d' affichage terminal..... 42
III.2.1.7.Phase d' enregistrement des spectres..... 43
III.2.2.Traitements différés. 46
III.2.2.1. La phase d'exploitation des spectres..... 46
III.2.2.2. La phase d'impression du rapport d'examen. 50
III.2.2.3. La phase de chargement de SPECTREs..... 51
III.2.2.4. La phase d' archivage 53
III.3 Spécifications fonctionnelles..... 55
III.3.1 Introduction..... 55
III.3.2 Spécification..... 56

IV.CONCEPTION LOGIQUE.....66
IV.1 Spécification des modules du niveau supérieur..... 66
IV.2. Spécification des objets..... 67

V. SPECIFICATION DE L'INTERFACE 75

V.1. Généralités 75

V.2. Notre architecture 77

VI.IMPLANTATION PHYSIQUE.....92

VI.1. Implantation de l'unité d'acquisition.....92

VI.1.1. Les ressources (Schéma bloc n°1).....92

VI.1.2. La communication entre ces ressources 95

VI.1.3.Commande et contrôle du processus d'acquisition de P.E 95

VI.2 Implantation du système de coordination.....108

VI.3. Implantation du système d'information.....110

CONCLUSION..... 1 1 2

Introduction

Ce travail de fin d'études a pour but d'analyser, concevoir et réaliser la commande et le contrôle par software d'un appareil médical mesurant l'activité cérébrale liée à la perception sensorielle. Une gestion minimum des données recueillies fait partie des objectifs.

La tâche est vaste, tant les secteurs touchés sont nombreux : pilotage d'une carte d'interface prototype, traitement mathématique et graphique des données et gestion d'un petit système d'information.

Le choix est fait de suivre globalement les méthodologies de développement des projets[1] enseignées pendant ces deux ans d'études. La démarche est intéressante car elle démontre la possibilité d'adapter ces méthodes à des applications ne portant pas strictement sur la gestion des informations. En effet, grande part est ici faite à l'unité d'acquisition et de traitement des données. Cela nécessite évidemment une adaptation des démarches d'analyse citées.

Le premier chapitre présente l'activité cérébrale liée à la perception sensorielle. Cette activité, de nature électrique, est appelée "Potentiels Evoqués". Tout est dit sur ses origines, ses formes, ses représentations. Ce chapitre entrevoit aussi les stratégies techniques couramment employées pour recueillir de tels signaux.

Dans le second chapitre, nous entrons dans le vif du sujet : l'élaboration, étape par étape, d'un cahier des charges solide, complet et cohérent.
La première partie expose les étapes techniques nécessaires pour réaliser l'acquisition des Potentiels Evoqués: étape analogique, conversion digitale et moyennage des données.
Par raffinements successifs, un appareillage plus précis se dessine et, avec lui, le cahier des charges apparaît.
Une première structuration du projet met en évidence ses deux aspects essentiels : le temps réel et les traitements différés.

Le troisième chapitre constitue l'analyse fonctionnelle. Le profil du projet se précise avec l'élaboration du schéma des données et du schéma des traitements. A ce stade de l'analyse, les "fonctionalités" de base du futur logiciel sont soulignées et spécifiées.

Avec le quatrième chapitre, nous entrons dans la phase de réalisation proprement dite: la conception logique globale. On y trouve une architecture du logiciel sous forme d'une hiérarchie de modules. Ceux-ci possèdent une cohésion informationnelle ou fonctionnelle selon leur relation avec le système d'information et le système d'acquisition.

---

[1]F.BODART: "Conception des systèmes d'information des organisations"
P.DUBOIS: "Méthodologie de développement de logiciels"

Le chapitre cinq aborde la question importante de l'interface utilisateur. Son architecture donne une idée précise de la façon dont les différentes "fonctionalités" seront coordonnées et dont les informations seront saisies et émises. Sa structure est conçue en s'inspirant également de ce qui nous a été enseigné[2].

Le sixième et dernier chapitre expose la réalisation physique d'une partie du projet cadre. Il souligne notamment l'implantation de l'unité d'acquisition sur un PC accompagné d'un carte prototype . Il traite également de l'implémentation de l'interface et de la base de données.

---

[2] F.BODART: "Interface homme/machine "

CHAPITRE I

Les potentiels évoqués

I. LES POTENTIELS EVOQUES.

Par quel processus nerveux l'homme perçoit-il et comprend-il le monde ?

Une approche expérimentale consiste à mesurer l'activité électrique de groupes localisés de neurones dans le cerveau durant l'accomplissement de tâches liées à la perception. Cette activité neuronale engendre des courants électriques [1] qui traversent la boîte cranienne et donnent naissance à de petites différences de potentiel, dits potentiels évoqués.

I.1. INTRODUCTION.

L'activité électrique globale des milliards de neurones du cerveau est enregistrée au moyen de l'électroencéphalogramme ou EEG. De tels enregistrements donnent peu de renseignements sur la perception sensorielle, parce qu'en général ils ne se sont pas correlés à des stimulations spécifiques.

Ce n'est pas le cas de la technique dite des **potentiels évoqués** (P.E).
Soit un flash lumineux atteignant la rétine. Des signaux partent des neurones de la rétine et aboutissent, après relais, dans l'aire visuelle corticale située à l'arrière du cerveau. Là, ils activent des neurones qui traitent l'information reçue. Ces neurones engendrent alors des différences de potentiel qui peuvent être détectées par des électrodes placées sur le cuir chevelu en regard du cortex visuel: on les appelle potentiels évoqués visuels (PEV). De même, on peut enregistrer des potentiels évoqués dans l'aire auditive du cortex cérébral à la suite d'un stimulus auditif. On parlera de potentiels évoqués auditifs (PEA). Suite à des stimulations tactiles, des potentiels évoqués somesthésiques sont également mesurés (PES).

La technique des P.E. a trois applications principales:

- elle contribue à la mise en évidence d'activités cérébrales spécifiques qui traitent simultanément différents types d'information dans des canaux distincts
- elle constitue un indicateur objectif des fonctions sensorielles là où les tests de perception sont impraticables ou peu sûrs (ex : nouveau-né).
- elle permet de préciser les maladies organiques et de les distinguer des troubles psychogéniques.

[1] voir annexe 1

I.2. COMMENT REPRESENTER CES POTENTIELS EVOQUES.

Comme nous l'avons vu, les P.E. peuvent être définis comme la réponse d'une population neuronale consécutive à l'application d'une stimulation sensorielle. Ils sont calculés sur une période comprise entre une milliseconde et 5 secondes suivant la présentation du stimulus, selon le type de P.E. envisagé (P.E. visuels, auditifs ou somesthésiques).

Le P.E. se présente donc sous forme d'un graphique dont l'ordonnée est exprimée en microvolts (μV) et l'abscisses en millisecondes (ms). L'origine de l'axe des abscisses coïncide avec l'instant de présentation du stimulus, de telle sorte que l'abscisse de tout événement repéré sur le graphique correspond au délai d'apparition, ou "**temps de latence**" de cet événement par rapport à la présentation du stimulus.

I.3. LES TECHNIQUES DE MOYENNAGE : DETECTION DE SIGNAL NOYE DANS UN BRUIT.

principe:

Le problème principal posé par l'enregistrement des P.E est celui de leurs détection. Comme ces potentiels sont très faibles, ils sont complètement masqués par l'activité électrique globale et spontanée du cerveau. Alors que le signal enregistré par l'électroencéphalogramme atteint souvent une amplitude de 1OO à 2OO mV, les potentiels évoqués ne dépassent pas le plus souvent quelques dizaines de mV et peuvent même tomber jusqu'au dixième de mV. Dans l'étude des P.E., le signal de l' EEG est ainsi un bruit de fond aussi indésirable qu'encombrant.
Pour résoudre ce problème majeur, nous aurons recourt à la technique dite du **moyennage.**

Comme cela vient d'être énoncé, un signal (le P.E.) survenant en réponse à la présentation de stimuli successifs est "noyé" dans un autre signal (Dû en grande partie à l'électroencéphalogramme (EEG)) d'amplitude plus importante, empêchant l'analyse directe du P.E. Le P.E. survient cependant toujours avec le même temps de latence par rapport à la présentation des stimuli alors que l'EEG est aléatoire par rapport à leur survenance. Il s'ensuit que si l'on calcule la moyenne d'un grand nombre de traces EEG synchronisées par la présentation de stimuli successifs, tout ce qui constitue le P.E. se dégage progressivement de l'EEG dont l'espérance statistique est nulle.

amélioration du rapport "signal/bruit"

La technique de moyennage permet une augmentation du rapport signal/bruit extrêmement importante.
Considérons ,à la figure 1.1, la séquence somme et la séquence moyennée de 16 essais sans qu'aucun stimulus n'ait été émis.

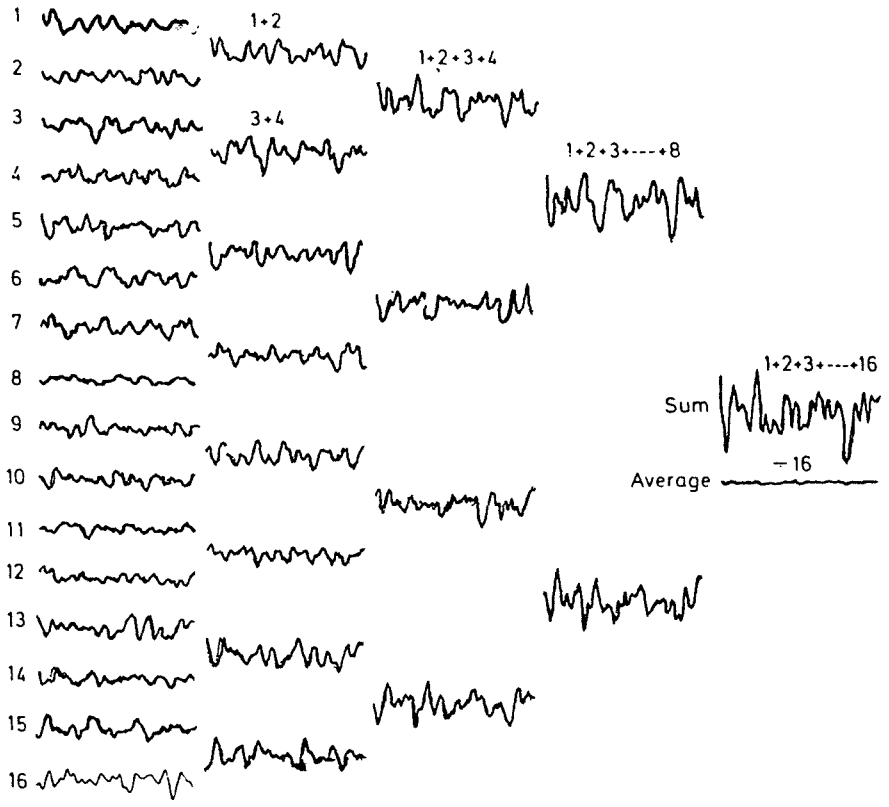


figure 1.1

On remarque en analysant les séquences sommes progressives que leur amplitude crête à crête augmente avec le nombre d'essais. Cependant, l'amplitude crête à crête de la séquence somme finale, par rapport à celle d'un essai individuel n'est pas augmentée d'un facteur égal à 16 essais mais plutôt d'un facteur égal à 4. En réalité, l'amplitude crête à crête de cette séquence somme est augmentée d'un facteur égal à la racine carrée du nombre d'essais et dans notre exemple $\sqrt{16} = 4$. C'est ainsi, qu'après moyennage, on retrouve un signal dont l'amplitude crête à crête a diminué d'un facteur $\sqrt{N}$. En effet, l'amplitude de la séquence somme vaut l'amplitude crête à crête d'un essai $\sqrt{N}$, tandis que l'amplitude crête à crête de la séquence moyennée vaut

$$\frac{\text{l'amplitude d'un essai} * \sqrt{N}}{N} = \frac{\text{l'amplitude d'un essai}}{\sqrt{N}}$$

Etant donné qu'il n'y a pas eu génération de stimuli, les séquences représentent l'activité de base du cerveau. On conclut qu'après moyennage, cette activité a diminué d'un facteur égal à la racine carrée du nombre d'essais.
Considérons maintenant à la figure 1.2 des séquences d'électroencéphalogramme avec émission d'un stimulus au début de chaque essai.

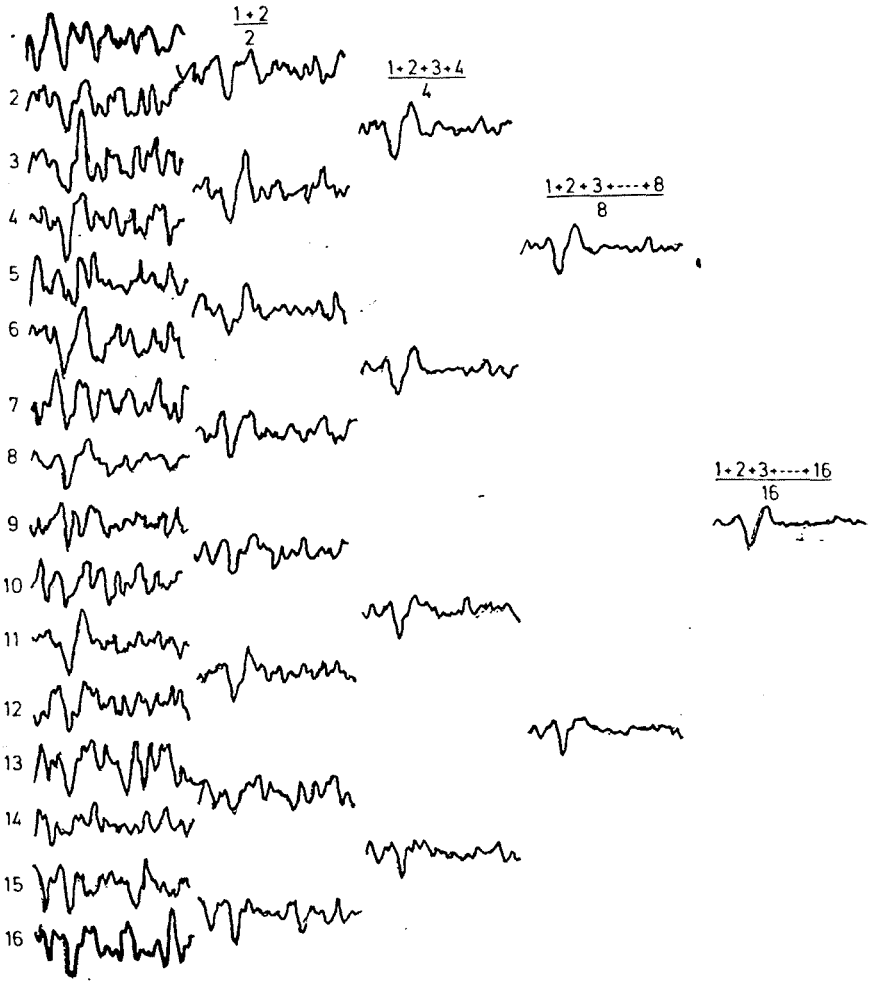


figure 1.2

Chaque stimulus produit normalement un potentiel évoqué identique qui pourrait être visualisé immédiatement si l'activité électrique de base du cerveau n'existait pas. Etant donné que cette dernière existe, il faut considérer le potentiel évoqué noyé dans le bruit de fond qui varie d'un essai à l'autre. Lorsque les séquences d'électroencéphalogramme sont additionnées et moyennées, le potentiel évoqué est resté identique à celui d'un essai individuel mais l'activité de base cérébrale a diminué de $\sqrt{N}$.

En conclusion, le rapport signal sur bruit s'accroît d'un facteur égale à la racine carrée du nombre d'essais.

Nombre d'essais.

Le nombre d'essais est déterminé par le rapport de l'amplitude de la réponse attendue à l'amplitude du bruit de fond de l'électroencéphalogramme, c'est-à-dire au rapport signal/bruit.
Si la réponse est faible (1µV) et si l'amplitude du bruit de fond est élevée (50µV crête à crête), 2500 essais sont nécessaires pour réduire le bruit de fond à 1 µV ($\sqrt{2500} = 50$).
Ainsi, on a ramené l'amplitude du bruit de fond à un ordre de grandeur similaire à l'amplitude du potentiel évoqué.
Cependant, pour détecter avec sécurité un potentiel évoqué faible, il est nécessaire de faire plus d'essais que ce nombre théorique.
Il faut donc que l'amplitude du bruit de fond soit inférieure à l'amplitude du potentiel évoqué. Ainsi, le nombre d'essais est donc toujours supérieur au carré de l'inverse du rapport signal/bruit présumé.

I.4. REJET D'ARTEFACTS.

La qualité du moyennage ne dépend pas seulement du nombre de signaux moyennés mais également du rejet d'artefacts. Un artefact est un phénomène d'origine accidentelle (clignement des yeux, mouvement du patient) qui se traduit par une suramplitude de l'EEG normal.
Ces artefacts peuvent être constants (modification lente de l'impédance d'électrode par la transpiration) ou transitoire. En général, les artefacts transitoires sont tellement importants par rapport aux potentiels évoqués qu'il est impossible de les éliminer par le moyennage. Parmi ces artefacts transitoires on trouve :

- *Les artéfacts physiologiques :*

Ce sont les plus importants. Il peut s'agir de mouvements oculaires ou de potentiels dûs à des mouvements musculaires. Les mouvements oculaires doivent être réduits au minimum en demandant au sujet de ne pas cligner des yeux. Les mouvements musculaires sont réduits en demandant au patient de rester immobile.
Une façon d'éliminer les artefacts est le rejet automatique de tout essai contaminé. Cette méthode devrait pouvoir éliminer automatiquement tous les essais où il y a suramplitude

du signal. Parfois, on enregistre le canal ENG (mouvements oculaires) de façon à éliminer les essais pour lesquels une activité significative apparaît sur ce canal.

- *Les artefacts électroniques* :

Ces artefacts proviennent d'une dérive des caractéristiques du système d'acquisition et de traitement. Afin de connaître cette dérive, on peut introduire un signal de calibration dans chaque essai.

I.5. NOMENCLATURE DES POTENTIELS EVOQUES.

Les résultats sont exprimés selon la nomenclature habituelle. On numérote l'onde suivant qu'elle est positive (P) ou négative (N) par rapport à la ligne de base iso-électrique (là où la différence de potentielle est nulle) et en fonction de la latence d'apparition (ex: P300, N100).

I.6. TYPES DE POTENTIELS EVOQUES.

Les potentiels évoqués se répartissent en trois grandes classes :

- Les composantes précoces de 1 à 10 ms.
- les composantes semi-précoces entre 10 et 50 ms.
- Les composantes tardives entre 50 et 500 ms.
- Les variations extrêmement longues : variations de contingence négative (CNV) de 500 à 5000 ms.

I.6.1. Potentiels évoqués auditifs.

Les potentiels évoqués auditifs sont de très faible amplitude (inférieure à 5µV) et de haute fréquence. Ils exigent une bande passante d'au moins 1000 Hz (annexe 1)(souvent de 10 à 1600 Hz)..

Le nombre d'essais est de l'ordre de 1OOO, mais la période réfractaire après chaque réponse est extrêmement courte, et les stimuli peuvent être présentés à la vitesse de 1O clics par seconde.

Ces potentiels sont très sensibles à l'intensité du stimulus et leur apparition est retardée d'autant plus que l'intensité du stimulus est faible.

I.6.2. Potentiels évoqués somesthésiques

Les P.E. somesthésiques évoquent une gamme de réponses allant des latences brèves aux latences longues. Ils sont générés par des chocs électriques au niveau du nerf médian du poignet et provoquent une réponse au niveau du cortex.

I.6.3. Potentiels évoqués visuels

La technique des P.E. visuels repose soit sur l'utilisation de flash lumineux à partir d'un stroboscope, soit d'un pattern réversible à diodes, de type échiquier. Les réponses évoquées par une variation de pattern d'échiquier sont plus stéréotypées chez les sujets normaux et se prêteraient plus à des comparaisons avec les patients. Il faut se rappeler toutefois que la latence et l'amplitude des composantes principales du potentiel évoqué sont dépendantes de l'intensité du stimulus et de la façon dont est généré le pattern. De plus, la réponse dépend de la taille de l'échiquier. En général, les patterns sont renversés toutes les demi-secondes et 1OO essais sont nécessaires. A cause de la variabilité de la brillance, de la taille de l'échiquier et de la vitesse de renversement, le système tout entier doit être calibré sur une population normale avant d'être utilisé chez le patient. On peut considérer que la réponse évoquée par un renversement d'échiquier ne contient pas de composantes à haute fréquence et qu'on ne peut compter sur des potentiels bien résolus que si on utilise une bande passante de 7O Hz (commençant à 0 Hz), et que l'on ne désire étudier que des potentiels de latence moyenne et longue.
Chez les individus normaux, l'amplitude varie de 3 à 30 µV et la latence de la composante principale est de 1OO msec.

I.6.4. Variations de contingences négatives:

La variation de contingence négative serait liée à des variables physiologiques telles que l'attention, l'activation générale, la motivation, la réaction d' orientation ou d'attente.
D'un point de vue instrumental, elle nécessite l' utilisation d' amplificateurs DC et d'électrodes très stables. De plus, elle exige une mesure de l' électroencéphalogramme sur des époques assez longues (elles peuvent dépasser plusieurs secondes).

1.7. L'INTERPRETATION DES POTENTIELS EVOQUES.[GUERI 91]

Il est intéressant d'esquisser les principes généraux de l'interprétation de P.E pour mettre en évidence les traitements nécessaires après l'acquisition complète d'une mesure.

On distingue trois types d'anomalies observables sur le graphique d'un P.E : des anomalies de temps de latence, d'amplitude et de topographie.

I.7.1. Anomalies du temps de latence.

Le temps de latence des réponses n'est jamais réellement diminué car aucun processus pathologique ne donne lieu à une accélération des vitesses de conduction nerveuse. Par contre une augmentation du temps de latence d'un pic (c'est-à-dire un maximum d'amplitude) peut correspondre à trois situations : une diminution des vitesses de conduction nerveuse, l'activation d'un sous-système de la modalité sensorielle étudiée ou l'exécution trop lente d'une tâche nécessaire à l'apparition d'un P.E.

I.7.2. Anomalies d'amplitude des réponses

Deux types de facteurs peuvent modifier l'amplitude des P.E. : des facteurs physiologiques et des facteurs techniques.
Les facteurs physiologiques sont :

- le nombre de fibres stimulées dans la population neuronale dont dépend le P.E.
- le degré de synchronisation de ces fibres (déterminant à la fois amplitude et durée de la réponse).
- l'orientation du dipôle générateur par rapport à l'électrode d'enregistrement (On entend par dipôle générateur, le groupe de neurones responsable de l' activité cérébrale reçue).

Les facteurs techniques sont liés au caractère bipolaire des enregistrements pour une amplitude donnée de l'activité captée au niveau de la cathode, l'amplitude du P.E. global peut être plus ou moins importante selon que l'anode capte simultanément une activité de polarité opposée ou identique.
On pourrait donc, par exemple diagnostiquer une lésion directe du générateur de P.E. suite à une diminution d'amplitude.

Il importe enfin de distinguer parmi les activités de grande amplitude, celles qui sont d'origine nerveuse et celles qui correspondent à des artefacts d'origine musculaire ou liés à des mouvements des yeux.

I.7.3. L'altération de la topographie des réponses.

Outre une variation de l'amplitude, la perte d'une ou plusieurs composantes liées à un canal sensoriel particulier donne lieu à des réponses distordues. On parlera d'altération topographique des réponses. L'interprétation des P.E. peut être particulièrement difficile dans ces situations car des confusions entre activités anormales topographiques mais normales quant à leur temps de latence et des activités normales retardées ou trop peu amples peuvent être enregistrées.

Ayant pris connaissance des différents points importants pour l'analyse d'un graphique de P.E., il reste à déduire les paramètres à recueillir : la présence ou l'absence des pics, leur latence et leur amplitude .

Quant à l'interprétation de ces paramètres, elle sera fonction de références normatives établies dans un groupe de patients comparable au patient étudié.
Nous entrons dès lors dans un domaine d' interprétation clinique qui dépasse nos objectifs. Il s'agit d'un domaine réservé à l'utilisateur.
On notera cependant que les paramètres des P.E. sont intimement liés aux caractéristiques du système d'acquisition.
Dès lors un système normatif propre à l'appareil de mesure sera souvent nécessaire.

I.8. CONCLUSION.

D'une façon générale, on peut définir un potentiel évoqué comme la réponse d'une population neuronale survenant avec une latence fixée après une stimulation sensorielle. Ceci l'oppose à l'activité électrique cérébrale continue appelée bruit de fond qui n'est pas synchronisée avec la stimulation.
Tous les P.E ont une amplitude bien inférieure à celle du bruit de fond électro-encéphalographique, électromyographique ou électrocardiographique recueilli par les électrodes cutanées utilisées pour l'enregistrement.

Le système d'acquisition et de traitement des P.E. devra donc améliorer considérablement le rapport signal/bruit.
Dans ce but, on fera la moyenne d'une série de réponses électrophysiologiques codées obtenues par répétition d'une même stimulation.
Le bruit de fond, par définition aléatoire et non synchronisé avec la stimulation tendra à disparaître alors que le P.E. croîtra avec le nombre de réponses moyennées.

Quant à l'analyse des données recueillies, elle sera du ressort de l'expert (le neurophysiologiste) aidé toutefois par des outils de présentation et de traitement qui permettront d'apprécier l'aspect de la courbe et de calculer facilement le temps de latence et l'amplitude des pics .

CHAPITRE II

Etude du projet

II. ETUDE DU PROJET

II.1.LES ASPECTS GENERAUX DE L'ACQUISITION.

Nous avons vu que l'activité électrique des structures nerveuses donnait lieu à l'apparition d'un potentiel électrique à la surface de la peau. Le placement d'électrodes en deux points mesure une différence de potentiel. Les variations temporelles de cette mesure reflètent les variations temporelles du signal nerveux. Notons que ces mesures ne sont pas exemptes de parasites extérieurs (réseau, champs électromagnétiques ...)

Nous savons également que le P.E. n'est qu'une infime partie du signal reçu, la grande partie émanant de l'activité globale des neurones. Notre but est donc d'extraire de ces mesures, une information sur les P.E. Autrement dit, l'acquisition suppose un travail de "nettoyage" indispensable. Ce traitement du signal se fera en trois étapes : l'étape analogique, la conversion analogique-digitale et le traitement mathématique du signal digital. A la sortie d'un tel traitement, le signal doit ressortir nettement du bruit dans lequel il était plongé.

- Etape analogique.

Les électrodes recueillent un signal analogique. Tel qu'il se présente, aucune analyse valable du P.E. ne pourrait être réalisée : l'amplitude est faible (µV) et il s'y superpose d'autres signaux (EEG, ECG...) d'amplitudes beaucoup plus grandes.
Il faut donc l'amplifier et le séparer des signaux parasites.
La chaîne d'amplification servira à multiplier l'amplitude du signal mesuré d'un facteur important (10^6). Ce facteur est le gain d'amplification.
La technique de l'amplification différentielle permet en outre d'éliminer une grande partie des signaux influençant les deux électrodes de façon identique (ex : parasites extérieurs) .

Les filtres quant à eux vont effectuer un nettoyage du signal en fréquence. En effet les activités parasites (EEG par exemple) peuvent différer des activités utiles non seulement par leur amplitude mais aussi par leur contenu fréquentiel (figure 2.1).
Il est possible dès lors de rejeter, par des filtres certaines activités inutiles.

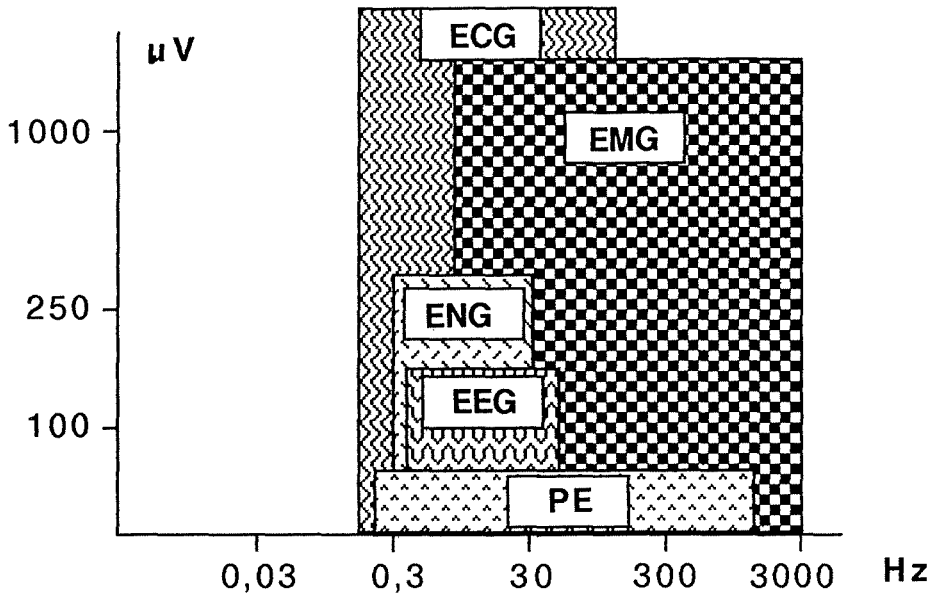


figure 2.1

(d'après une idée de L. Sovèges): le signal capté par les électrodes est constitué non seulement de P.E mais également d'autres activités biologiques parasites (EEG: ElectroEncéphalogramme; EMG: ElectroMyogramme ; ECG: Electrocardiogramme; ENG: clignement des yeux.)

- **La conversion analogique - digitale.**

Filtrage et amplification constitue donc les grandes fonctions de l'étape analogique de l'information. Son acquisition proprement dite passe obligatoirement par une conversion analogique-digitale. Cette étape correspond à la transformation d'un signal en temps continu en un signal en temps discret par le processus d'échantillonnage : prise de mesure à grande fréquence de l'amplitude du signal analogique, codage digital de la valeur et transfert en mémoire.
La fréquence d'échantillonnage sera telle que les données discrètes contiennent toute l'information du signal original.

- **Traitement informatique :le moyennage.**

Le signal analogique a été transformé en une suite de nombres représentant l'évolution de l'amplitude du signal en fonction du temps. Si aucune autre opération n'est réalisée, nous aboutissons à une longue séquence de nombres représentant un mélange de diverses informations provenant de l'activité cérébrale.
Le moyennage, comme nous l'avons déjà dit, va améliorer considérablement le rapport signal (P.E) sur bruit (parasites + EEG).

Sa réalisation pratique passe par une stimulation périodique et une synchronisation des données par rapport aux stimuli.(figure 2.2)

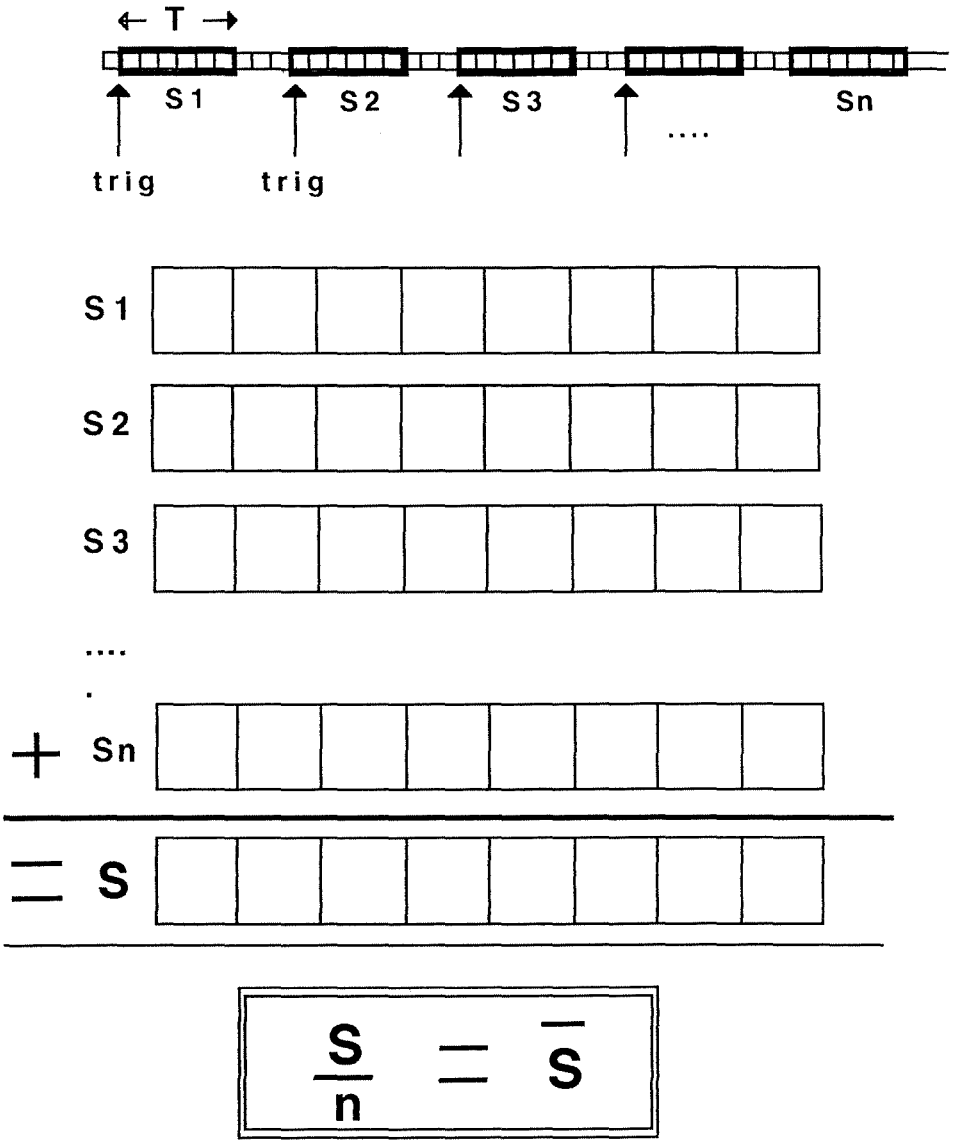


figure 2.2 -

Principe du moyennage. Le trigger survient simultanément à la présentation du stimulus et déclenche la conversion analogique-digitale du signal pendant une période de temps T (fenêtre d'analyse). Les traces suivant l' ensemble des stimuli, synchronisées par la présentation des stimuli, sont alors moyennées. (une idée de Jean Michel Guérit).

Le trigger est un signal électrique synchronisé à la présentation de chaque stimulus, avertissant l'ordinateur de sa survenue. Celui-ci peut dès lors effectuer la segmentation correcte des données en entrée et constituer la moyenne de tous les segments constitués pendant la mesure. Bien entendu, cette opération s'effectue au fur et à mesure de l'acquisition des signaux.

Lors du moyennage, un certain nombre de traces peuvent être brouillées par des parasites de grande amplitude (mouvements, EEG, clignements des yeux). En l'absence de mesures appropriées, ces parasites risquent de modifier la moyenne. Pour éviter cet écueil, le système de moyennage doit être équipé d'un système de réjection tel que toute trace contenant un signal d'une amplitude supérieure à un certain seuil fixé d'avance soit éliminé du décompte final.

II.2. CAHIER GENERAL DES CHARGES.

Un système sophistiqué d'acquisition de P.E doit , non seulement générer les stimuli, qu'ils soient auditifs, visuels ou somesthésiques, mais aussi réceptionner et coder l'électroencéphalogramme pour enfin traiter les composantes du P.E mathématiquement et graphiquement.
La mise au point d'une base de données liant patients aux mesures effectuées n'est pas à négliger .

Dans cette section, nous donnons une représentation schématique et générale de l'appareillage nécessaire à l'acquisition et au traitement des Potentiels Evoqués. Aucun choix n'est encore fait quant à l'implantation mais nous dressons l'inventaire des fonctions importantes au sein d'un tel système.

Cet appareillage se compose de quatre parties distinctes :

- un système de stimulation ,composé de générateurs, qui produit et contrôle les stimulations .
- un système d'amplification et de filtrage des réponses.
- un système de traitement mathématique et graphique du signal afin de produire des résultats interprétables.
- une base de données contenant les informations nécessaires à l'élaboration de rapports médicaux complets.

II.2.1 Système de stimulation.

principes :

Les stimuli, quelque soit le mode sensoriel choisi, doivent être reproductibles tant dans leur forme, leur intensité, qu'au niveau de leur durée d'apparition et de l'intervalle de temps qui les séparent.
Cet intervalle de temps doit être programmable entre deux limites; on les appellera "low latency" (LLA) et "hight latency" (HLA). Entre ces deux limites, les stimuli doivent

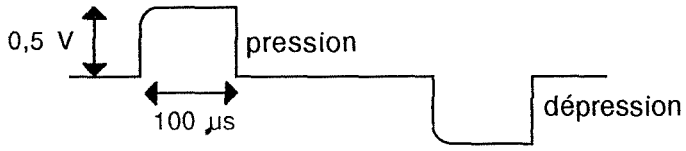
pouvoir être générés selon des algorithmes spécifiques, notamment la génération aléatoire.

Le type de stimulus utilisé a une grande importance car ses caractéristiques conditionneront les paramètres de la réponse électrophysiologique.
Etudiant principalement trois types de P.E. différents, PEV, PEA et PES, on distingue trois modules de stimulation : un module visuel, un module auditif et un module somesthésique.
Chacun des modules regroupe différents types de stimulations définis par des caractéristiques telles que la durée, l'intensité, la fréquence et la forme .

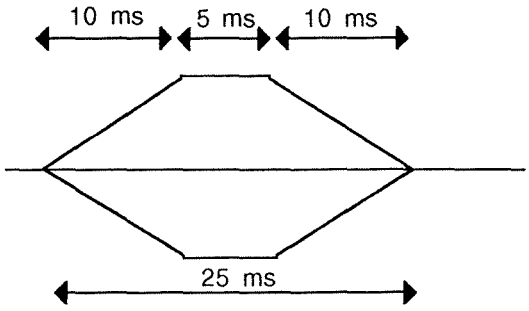
II.2.1.1.Le module des stimulations auditives.

On distingue deux types de stimulations :

le clic:



le burst :



L'intensité du son doit être un paramètre ajustable.
Les stimulations se feront en monoauriculaire ou biauroculaire, grâce à un casque d'écoute.

II.2.1.2. Le module des stimulations visuelles.

On distingue principalement :

- le flash lumineux de courte durée (lampe stroboscopique).
- le flash lumineux de longue durée (intensité variable).
- le pattern à damier : apparition d'un motif à damier pendant une fraction de secondes puis disparition ou inversion du motif

Pour les flashs, les stimulations se feront soit en binoculaire, soit en monoculaire grâce à des lunettes.

Pour les damiers , le pattern se situera à une distance pouvant varier et portera un point à fixer par le patient.

II.2.1.3.Le module des stimulations somesthésiques.

Il s'agira de l'application d'un courant de l'ordre du milliAmpère sur la peau du patient. L'intensité du courant et la durée du stimulus seront variables.

II.2.2. Le système de captation d'amplification et de filtrage de signaux

L'amplification physiologique a comme fonction principale d'augmenter les potentiels électriques dérivant du crâne et des nerfs tout en améliorant sensiblement le rapport signal/bruit.
Rappelons que la réponse évoquée est de l'ordre du µV, parfois moins. Le convertisseur Analogique-Digital, quant à lui manipule des différences de potentiels de l'ordre du Volt. Un gain de 10^6 se déduit aisément.

Une première manière d'améliorer le rapport signal/bruit se fait par l'emploi d'un système à trois électrodes :

- deux électrodes actives qui déterminent la différence de potentiel et la polarité de la réponse .
- l'électrode de masse place le patient à une tension identique à celle de la chaîne de mesure.

L'ensemble forme un **canal d' acquisition** .

Le préamplificateur est constitué de deux chaînes identiques couplées et chacune reçoit le potentiel d'une électrode du dipôle biologique. Par ce système, une tension parasite parvient de façon identique aux deux entrées de l'amplificateur et est ainsi éliminée.

Un système de filtrage du signal en fréquence permet également, et très proprement d'améliorer le rapport signal/bruit.
Quand on sait que le potentiel électrophysiologique couvre une gamme de fréquences très large (O à 5OOO Hz selon l'origine), certaines de celles-ci sont sans intérêt dans l'étude menée et n'apporte que des "parasites" supplémentaires.
L'ajustage d'une bande passante autour du domaine des fréquences intéressantes éliminera quantité de bruits.
Les valeurs les plus courantes sont les suivantes :

- PEV : de 0,3 à 600 Hz,
- PEA : de 10 à 1600 HZ,
- PES : de 1 à 1000 Hz (certains vont même jusque 3000 Hz)

II.2.3 Un système de conversion analogique-digital.

Nyquist a démontré que, "si un signal quelconque était appliqué à l'entrée d'un filtre passe-bas ayant une bande passante **H**, le signal ainsi filtré pouvait être entièrement reconstitué à partir d' un échantillonnage de ce signal effectué à une cadence précise et égale à **2H** par seconde . Une vitesse d'échantillonnage supérieure à 2H par seconde ne présente aucun intérêt car les fréquences supérieures qu'elle permettrait d'isoler pour reconstituer le signal, ont été éliminées lors du passage dans le filtre".
En bref , la fréquence de cadencement du convertisseur, programmable, sera choisie en fonction du choix des filtres.

II.2.4 Un système de traitement du signal afin de produire des résultats interprétables .

II.2.4.1. Le moyennage.

Rappelons qu'il s'agit d'une analyse séquentielle tendant à présenter le résultat moyen d'un certain nombre de balayages succédant chacun à un stimulus.
Pour que les réponses aux stimuli s'additionnent, elles doivent présenter deux caractéristiques indispensables :

- la synchronisation de la séquence d'analyse avec le stimulus.
- l'existence d'une chronologie constante de la réponse par rapport aux stimuli. C'est l'hypothèse essentielle sans laquelle les P.E sont indétectables (le bruit n'est pas synchronisé à la stimulation).

En bref, des phénomènes en phase s'ajoutent et se dégagent peu à peu du bruit de fond pour faire apparaître le signal épuré après un certain nombre de stimulations.
Pour peu qu'un parasite soit synchronisé au stimulus, il va être majoré tout comme le signal. L'artefact de stimulation en est un exemple. L'artifice technique consiste à inverser la phase de stimulus à chaque séquence.

II.2.4.2 Processus en parallèle.

Jusqu' ici nous avons considéré le moyennage d'un signal provenant d'un seul canal. Il est clair que certains examens peuvent réclamer plusieurs canaux de réception . Un

multiplexage des entrées est indispensable et repose évidemment la question de la fréquence d'échantillonnage du convertisseur Analogique-Digital.
Si le nombre de canaux est doublé, il faudra également doubler la fréquence du convertisseur car celui-ci n'échantillonne chaque voie que chacune à tour de rôle.

II.2.4.3. Le déclenchement ("triggering").

Pour réaliser la moyenne, il importe que chaque stimulus soit synchronisé à l' unité d'analyse.
Le triggering du moyennage débute soit à la fin de la stimulation, soit quelques msec avant l' apparition de celle-ci, dans le but de réaliser une calibration de la ligne de base .
Ce triggering peut être réalisé à partir d'une programmation interne (lorsque le stimulateur est interne) ou à partir d'un stimulus externe (par exemple lors de tests psychométriques de vigilance).

II.2.4.4. La fenêtre d' analyse.

Il importe que le triggering déclenche un échantillonnage de valeurs pendant un temps compatible avec la latence du phénomène étudié:

- fenêtre de 0 à 10 ms pour les P.E précoces (P.E.P)
- fenêtre de 10 à 50 ms pour les P.E moyens (P.E.S.P)
- fenêtre de 50 à 500 ms pour les P.E tardifs (P.E.T)
- fenêtre juqu' à 5000 ms pour les V.C.N.

II.2.4.5. La récurrence de stimulation.

Elle fait partie intégrante du mode de stimulation sensorielle que l'on désire pratiquer lors d' une mesure , cependant sa valeur est limitée par la fenêtre d'analyse . On ne pourrait en effet étudier une réponse tardive (500 ms) avec une récurrence de 10 stimuli par seconde .

II.2.4.6. Le nombre de stimulations.

Dans le cadre de la technique de moyennage, on comprend la nécessité de pouvoir choisir la valeur de ce paramètre.
Il est le résultat d'un compromis : tendre à diminuer la longueur de l'examen tout en apportant un bon rapport signal/bruit.

II.2.5 Les artefacts.

Les artefacts de toutes origines (physiologiques, électroniques) doivent être supprimés en fixant un seuil de potentiel au delà duquel une valeur est considérée comme fausse . Dès cet instant, si un signal électrophysiologique contient une valeur dépassant ce seuil, tout le signal est rejeté. Pour aboutir au nombre de stimuli voulu, il faudra prolonger l'examen du nombre de traces rejetées .
La diminution du seuil de réjection augmente la probabilité de rejet de traces parasitées et dès lors augmente la qualité du P.E recueilli . En contre-partie, l'augmentation de la durée de l'examen peut devenir une source d'erreurs (signes d'impatience, moins bonnes réactions aux stimulations ...) . Habituellement, un taux de réjection de 20% est considéré comme bon.

II.2.6. Le traitement graphique.

Généralités:

- Les P.E se représentent sous forme d'un graphe dont l' ordonnée est exprimée en µV et l' abscisse en ms.
- Le signal oscille autour d'une ligne de base iso-électrique .
- Par convention, les pics positifs pointeront vers le bas et les pics négatifs pointeront vers le haut.

En particulier,
les traces entrantes doivent apparaître à l'écran de façon continuelle ce qui permet une certaine surveillance

- de l'amplitude du signal
- de la stabilité des électrodes
- de phénomènes de saturation

- Dans le cas de dépassement du seuil de rejet, la mention "ARTEFACT" doit apparaître ainsi que le taux de rejet courant.
- De temps à autre, on passera en mode moyennage pour voir afficher sur l'écran la réponse évoquée en formation.
- A la fin de l'examen le graphique doit visualiser la ligne de base iso-électrique et les potentiels évoqués moyennés;
- La présentation de traces superposées s'explique par le fait qu'un artefact pourrait, malgré les diverses protections, contaminer la réponse évoquée . Il est conseillé d'effectuer, au minimum, deux mesures complètes . La superposition de deux mesures, normalement identiques, permet de résoudre cette dernière difficulté.

- La modification du gain du signal traité doit être possible, pour permettre une meilleure occupation de l'écran et une meilleure vision du graphique final tout en évitant la saturation. On notera cependant que le changement d'échelle reste limité aux phénomènes étudiés. Une saturation peut ainsi signifier une anomalie dans la mesure :
  - une électrode mal placée
  - perturbation du réseau
  - ...

II.2.7. Le traitement mathématique.

Amplitude et temps de latence

La sélection visuelle des pics sera obtenue par l'utilisation de deux curseurs se déplaçant le long de la courbe.
A tout instant, les positions des curseurs indiqueront :
- en abscisse : la latence (en ms),
- en ordonnée : l' amplitude **absolue** c-à-d par rapport à la ligne de base (en μV).

Pour les amplitudes **relatives** deux curseurs sont utilisés .
La mesure de l'amplitude se fait par rapport à celle d'un autre pic antérieur ou ultérieur de polarité opposée.

II.2.8. Impression des résultats.

En fin de moyennage, le système est capable de transcrire les courbes une à une. Une fois ce transfert réalisé sur papier, toutes les courbes affichées à l'écran, on ajoutera différents paramètres tels que type de P.E., fenêtre d'analyse, coordonnées du patient, date de l'examen, pour composer un rapport médical identifiable et complet.
Chaque type d'examen implique un rapport particulier(annexe 2).
Résumons maintenant l'ensemble de moyens à mettre en oeuvre pour réaliser un appareil permettant l'acquisition et le traitement de P.E.

Les entrées :

- Un type de stimulation
  Cette donnée est structurée car elle comprend une partie qui sélectionne un module de stimulation (PEU, PEA, ou PES).

Une partie qui définit plus précisément la stimulation responsable des réponses évoquées (définition en terme de forme, de fréquence, de durée...)

- Des signaux analogiques (activité cérébrale)

  Ceux-ci sont de l'ordre du mV.

  Ils peuvent être pris à différents endroits simultanément (1,2,3, ... canaux)

- Deux fréquences

  Ces paramètres définissent un filtre passe-haut et un filtre passe-bas.

  Ils composent la bande passante nécessaire au filtrage du signal d'entrée.

- Un seuil de rejet

  C'est la valeur au delà de laquelle, un signal doit être éliminé pour conserver une réponse évoquée cohérente.

- Un nombre de stimulations

  Indique le nombre de mesures dont on souhaite faire la moyenne. Chaque mesure rejetée (artefact) n' est pas comptabilisée.

- La récurrence de stimulation

  Indiquée en un certain nombre par seconde, elle définit le rythme dans la succession des stimuli.

- La fenêtre d'analyse.

  Exprimée en milliseconde, ce paramètre indique le délai choisi après une stimulation pour recueillir les informations évoquées. Ce temps est d'au moins 1O ms et peut aller jusqu'à 5000ms .

Un nombre de canaux de réception.

Ceci influence la fréquence d'échantillonnage des données.

On notera finalement que, les indications sur les conditions d'examens, les commentaires généraux et toute autre information intéressante à mémoriser lors d'un tel examen, ne sont pas des données directement nécessaires à la programmation du système. Elles seront toutefois prises en compte pour faciliter l'élaboration d'un rapport médical complet.

Les sorties et résultats :

Les sorties sont partiellement superposables aux entrées :

- l' ajustement des paramètres tels que les filtres .
- l' ajustement du seuil de rejet .
- la modification des caractéristiques de la stimulation .
- la modification du nombre et de la récurrence des stimulations.
- la programmation de la fenêtre d'analyse.
- la détermination d'une fréquence d'échantillonnage.

Les résultats sont :

- Les mesures successives
  Comme nous l'avons déjà dit, leur affichage est un excellent moyen de contrôler le bon déroulement de l'examen.
- Les courbes évoquées
  De temps à autre, ainsi qu'à la fin de l'examen, apparaissent le résultat des moyennes.
- L'amplitude d'un pic (mode absolu ou relatif)
- Le temps de latence d'un pic.
- L'impression d'un rapport complet sur l'examen.

Les principales phases de traitement.

- phase d' initialisation du système
- phase de stimulation
- phase d'amplification et de filtrage
- phase de synchronisation entre stimuli et fenêtre d'analyse
- phases de conversion digitale et transfert en mémoire
- phase de rejet des artefacts
- phase de moyennage des signaux
- phase d'affichage courant
- phase d'affichage terminal
- phase d' enregistrement de la mesure
- phase d' exploitation des réponses évoquées
- phase d'impression du rapport d'examen

Les ressources.

- les unités de stimulations

- les canaux de prise de donnée analogique
- la chaîne d'amplification et de filtrage
- le convertisseur analogique-digital
- l' unité de synchronisation
- l'unité arithmétique et logique
- les unités d'affichage et d'impression
- la mémoire de travail
- **Les dispositifs de stockage.**

  Il s'agit d'une composante particulièrement importante du système dans la mesure où l'analyse différée des informations qu'ils permettent, détermine non seulement les possibilités de travaux scientifiques ultérieurs, mais également la rapidité de l'examen (en évitant de perdre du temps, en présence du patient, à des analyses qui pourraient être réalisées ultérieurement.).

  Cette composante n'est pas toujours à la disposition des systèmes actuels. C'est une des raisons à l'origine de ce mémoire.

II.3.PREMIERE STRUCTURATION DU PROJET.

Ayant ainsi identifié les frontières du projet d'acquisition et de traitement des potentiels évoqués, découpons-le en deux applications autonomes . L'une, l'acquisition proprement dite, l'autre , mettant en valeur l'aspect de traitements différés(figure 2.3).

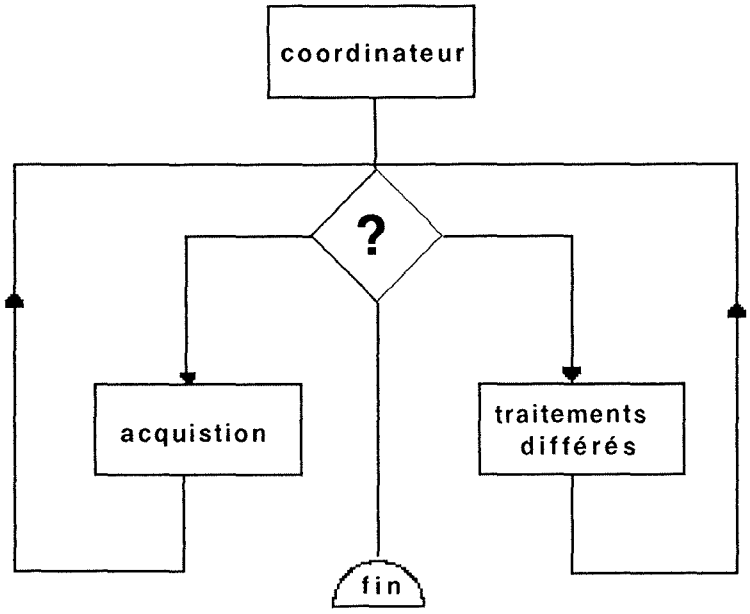


figure 2.3: découpe du projet en applications

Ces deux applications sont fondamentalement différentes.
La première se compose de plusieurs traitements en temps réels, contenant synchronisations , travaux en parallèle et emploi de multiples ressources.
La seconde rentre dans le cadre général des applications de gestion, même si les traitements effectués sont d'orientation plutôt mathématique et graphique.

II.3.1. L'acquisition.(figure 2.4)

Spécifications:

définition:

Cette application se déroule en temps réel. Ses objectifs sont l'initialisation des différents paramètres destinés à la mesure et la production d' une courbe de P.E. de bon rapport signal/bruit. Elle fournira également, tout au long de la mesure, un contrôle visuel à l'utilisateur.

reçoit:

- un message de demande d' acquisition
- tous les attributs nécessaires à l' enregistrement d' un **PE:**
- message: "Start acquisition"
- Des signaux analogiques (activité cérébrale)

génère:

.pour chaque canal d' acquisition :

- les **traces** successives
- les **sommes** successives des différentes traces
- les **moyennes** successives .

.un **nr_pe,** un **PE** enregistré

.les **spectres** du **PE** identifié par **nr_pe** enregistrés

.un **nr_dos** si le patient n'existait pas dans le S.I.

.un **patient** enregistré si il n'existait pas dans le S.I.

messages possibles:

- messages indiquant des paramètres invalides
- messages d'erreur concernant le système d'acquisition
- message de confirmation en vue du début d'acquisition
- message d'enregistrement d'une mesure

action sur l'U.A:

- l'ajustage des paramètres des filtres
- l'ajustage du seuil de rejet
- la modification des caractéristiques de la stimulation
- la modification du nombre et de la récurrence des stimulations
- la programmation de la fenêtre d'analyse
- la programmation de la fréquence d'échantillonnage

action sur le S.I:

un examen de P.E. dans la base de données :

- création d'un **PE**
- création d' un **patient** associé s'il n'existait pas dans la base de données
- création des **spectres** associés
- création des associations **prescription** et **production**

découpe en phases de traitement:

- phase d'initialisation du système d'acquisition
- phase de stimulation
- phase d'amplification et de filtrage
- phase de conversion digitale
- phase de transfert en mémoire
- phase de rejet des artefacts
- phase de moyennage des signaux
- phase d'affichage courant
- phase d'affichage terminal
- phase d'enregistrement de l'examen

synchronisations:

.la synchronisation **synchro_début**

est réalisée à l'arrivée du message "Start acquisition", lorsque l'initialisation de l'unité d'acquisition (U.A.) est terminée.

.la synchronisation **fin_fenêtre**

est réalisée lorsque le temps d'analyse, déterminé par le paramètre **fenêtre_analyse** est écoulé. Elle détermine donc la fin d'une **trace.**

.la synchronisation **constitution d'une série de stimuli**

est réalisée lorsque le paramètre nbre_stimulation est atteint. Elle marque la fin de l'acquisition du P.E.

.le paramètre de duplication **pour chaque stimulus**

a une valeur indiquant le nombre de mesures dont on souhaite tenir compte pour la mesure globale du P.E.

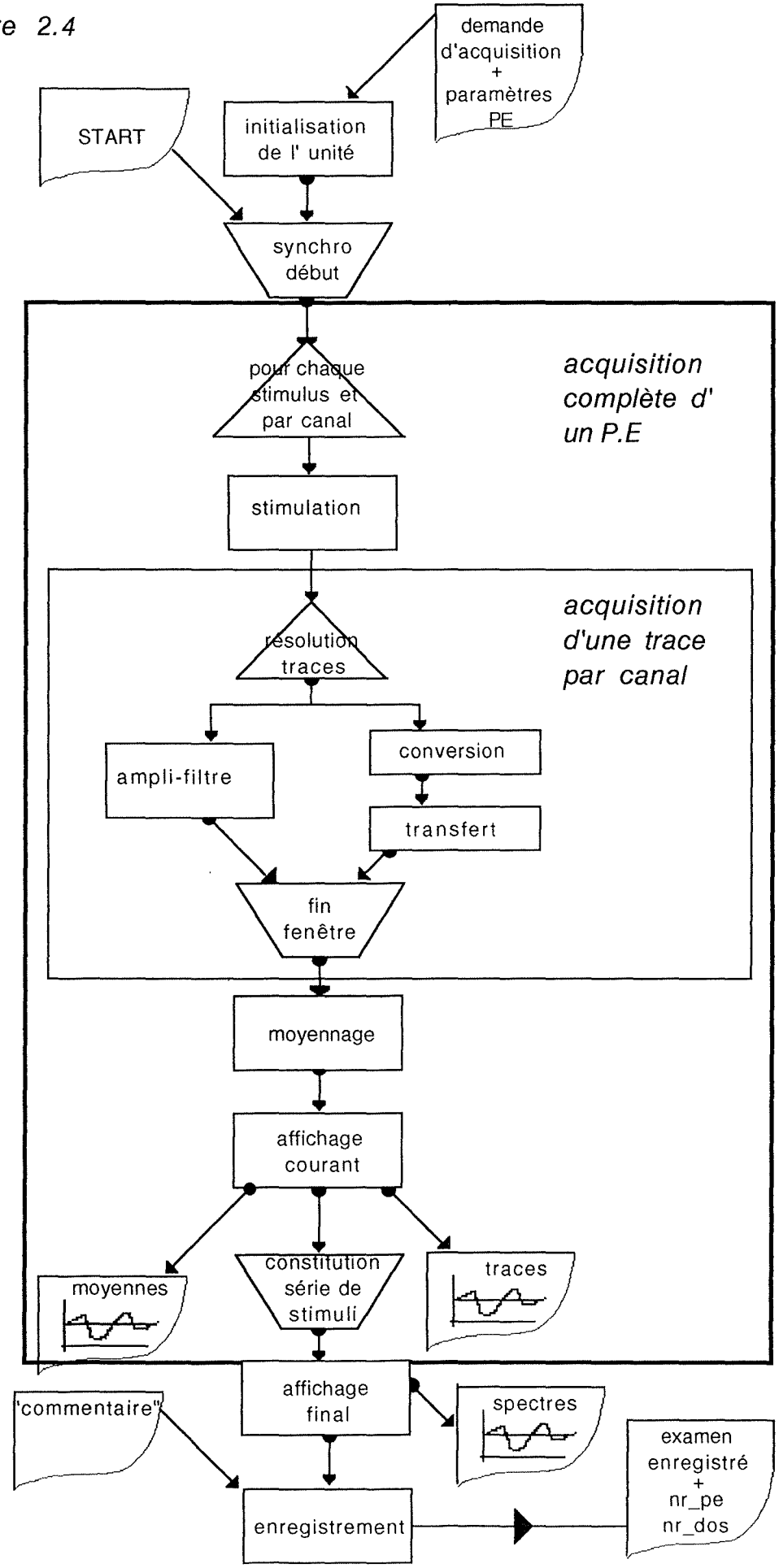
.le paramètre de duplication **résolution des traces**

a une valeur indiquant le nombre de données pour la constitution d'une trace. Ce paramètre se déduit de la fréquence d'échantillonnage et de la fenêtre d'analyse.

ressources:

- les unités de stimulations
- les canaux de prise de donnée analogique
- la chaîne d'amplification et de filtrage
- le convertisseur analogique-digital

figure 2.4


demande d'acquisition + paramètres PE
START
initialisation de l' unité
synchro début
pour chaque stimulus et par canal
acquisition complète d' un P.E
stimulation
résolution traces
acquisition d'une trace par canal
ampli-filtre
conversion
transfert
fin fenêtre
moyennage
affichage courant
moyennes
constitution série de stimuli
traces
affichage final
spectres
"commentaire"
examen enregistré + nr_pe nr_dos
enregistrement

- l'unité de transfert vers la mémoire
- l'unité de synchronisation
- l'unité arithmétique et logique
- l'unité d'affichage
- la mémoire de travail
- l'unité de stockage

II.3.2. Les traitements différés.

Spécification:

définition:

Cette application est l'outil d'analyse des P.E. L'autonomie temporelle qui lui est donnée par rapport à l'outil d'acquisition permet d'interpréter non seulement l'examen courant mais également un examen effectué précédemment.
Elle contient donc les traitements graphiques et mathématiques souhaités pour l'interprétation des mesures affichées.

reçoit:

- une demande de traitement différé
- un ou plusieurs **nr_pe**: ils identifient les examens que l'on désire exploiter.

génère:

.les **spectres** associés au **PE**
.le traitement différé effectué:
les informations souhaitées sur ces **spectres** ; **amplitude** et **latence** des pics.
les transformations graphiques souhaitées :
- changement de **gain**
- superposition graphique de **spectres** de différents **PE**.

.un rapport médical .

messages possibles:

- message de PE demandé inexistant
- message d'erreur à l'impression

Découpe en phases: (figure 2.5)

- phase d'impression du rapport d'examen.
- phase de chargement d'un P.E.
- phase d'exploitation (amplitude, latence) des spectres.
- phase d'archivage .

rmq: cette découpe dévoile l'apparition de deux nouvelles phases, non directement déduites du cahier des charges. Il s'agit du chargement d'un P.E et de l'archivage du système d'information.
Leur présence est nécessaire pour réaliser tous les objectifs de traitement des P.E. en "différé".

ressources:

- l'unité arithmétique et logique
- l' unité d'affichage
- la mémoire de travail
- l'unité de stockage

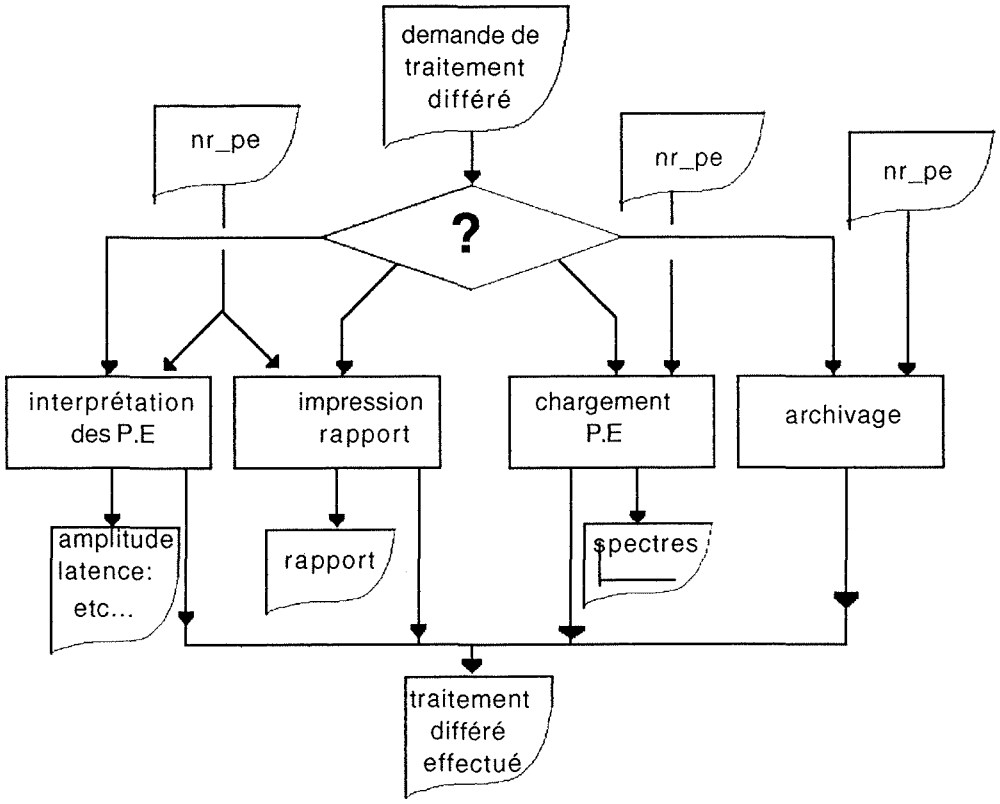


figure 2.5.: cette application est destinée à être répétée un grand nombre de fois.

CHAPITRE III

Analyse fonctionnelle

III. ANALYSE FONCTIONNELLE

III.1. LE SCHEMA CONCEPTUEL DES INFORMATIONS

Nous avons établi le schéma Entité/Association et surtout les domaines de valeurs des divers attributs sur base d'interview et de consultation de documentation.Une mise en ordre finale et une sélection des informations furent donc aussi nécessaires que délicates. Il faut, en effet, savoir que la technique des potentiels évoqués est relativement récente et que les renseignements que l'on peut trouver dans la littérature actuelle sont parfois peu cohérentes voire contradictoires.

III.1.1.Représentation de la structure du système d'information - Schéma Entité/Association

Quand on analyse le cahier de charge, on se rend compte que l'information qui sera utilisée et/ou mémorisée concerne trois concepts distincts : PATIENT, PE (Potentiel Evoqué) et SPECTRE. Ainsi, on réalise sur un PATIENT une action spécifiée (un PE) et l'on recueille de cette action les spectres caractéristiques.

Ces trois concepts mis en évidence constituent donc, chacun, une entité à part entière qui sera décrite par le sous-ensemble de ses attributs. Ainsi, nous aurons l'entité PATIENT, l'entité PE et l'entité SPECTRE.

ENTITE : PATIENT.

DEFINITION.

Toute personne physique ou morale qui a subi au moins un examen de PE.

IDENTIFIANTS.

Nom : le nom du patient.
Prénom : son prénom.
dnaiss : sa date de naissance.
nr_dos : le numéro de dossier du patient, ce numéro est un identifiant attribué une fois pour toute lorsque le paient vient pour la première fois subir un examen de PE.

ATTRIBUTS.

1) adresse : l'adresse du patient.
2) téléphone : son numéro de téléphone.
3) sexe : M ou F.

4) nom-médecin : nom du médecin qui demande l'examen.
5) taille : la taille du patient.

CONTRAINTES.

Tous les attributs sont obligatoires

ENTITE : PE.

DEFINITION.

Toute référence à un potentiel évoqué possible.

IDENTIFIANT.

nr_pe : numéro d'ordre identifiant un PE et attribué lors de la création de cette entité.

ATTRIBUTS.

1) type_pe : type de Potentiel Evoqué.
2) FH : fréquence du filtre passe-haut.
3) FB : fréquence du filtre passe-bas.
4) nbre_stimulation : nombre de stimulations d'un type donné à réaliser pour recueillir le spectre associé.
5) type_stimulation : type de stimulation à réaliser (flash, clic...)
6) récurrence : nombre de stimulations effectuées par seconde
7) fenêtre_analyse : durée de balayage (en milliseconde)
8) nbre_canaux : nombre de canaux utilisés pour un potentiel évoqué particulier.
9) seuil_rejet : le seuil de rejet avec lequel l'examen a été réalisé.
10)canal1 : position des électrodes (2) qui recueillent le Potentiel évoqué de l'hémisphère droit du cerveau.
11) canal2 : position des électrodes (2) qui recueillent le potentiel évoqué de l'hémisphère gauche du cerveau.
12) auriculaire : caractérise le fait qu'une des 2 oreilles est "bouchée" ou non.
13) oculaire : caractérise le fait qu'un des 2 yeux est occulté ou non.
14) durée : durée d'une stimulation.(en microseconde)
15) intensité : intensité sonore (en décibel).

16) fréquence : fréquence d'une stimulation (en Hertz).
17) compteur : nombre de spectres associés au PE.
18) date : date où le PE a été réalisé.

19) Commentaire : commentaire éventuel du médecin réalisant l'examen.

CONTRAINTES

- Les attributs (1O,11,12,13,14,15,16,19) sont facultatifs.
- Les attributs (1,2,3,4,5,6,7,8,11,12,13,14,15,16,17) possèdent un ensemble de valeurs spécifiques.

type_pe ∈ {P3OO, PEV,PEAP,PEASP,PEAT,PES,VCN}
FH ∈ {600,3000}
FB ∈ {0.3,10}
nbre_stimulation ∈ {2O , 5OOO}
type_stimulation ∈ {flash, damier, clic, clic, filtré, tone-burst, tone-pipe, tone-pipe-alterné, logon, son}
récurrence ∈ {o.75, 2O} (en nombre de stimuli par seconde)
fenêtre_analyse ∈ {1O, 5OOO}
nbre_canaux ∈ {1,4}
auriculaire et oculaire ∈ {mono et bi}
durée ∈ { , } (Encore indéterminée)
intensité ∈ {O, 11O}
fréquence ∈ { , } (Encore indéterminée)
compteur ∈ {1,2,3,4}

ENTITE : SPECTRE.

DEFINITION.

Tout résultat d'une acquisition d'un PE.

IDENTIFIANT.

nr_spectre : numéro d'ordre donné lors de la création de cette entité.
Son rôle dans l'association PRODUCTION.

ATTRIBUTS.

1) moyenne : ensemble des points obtenus suite au processus d'acquisition et qui représente la courbe caractéristique du PE recueilli.

2) Hémisphère : spécifie l'hémisphère (droit ou gauche) source du PE recueilli.
3) Résolution: nombre de points appartenant à une moyenne
4) gain : spécifie le gain
5) pic : spécifie l'amplitude, la latence et le nom attribué à un maximum du spectre.

CONTRAINTES.

Les attributs 2 et 3 sont facultatifs. (pas vraiment une contrainte).
L'attribut 5 est facultatif et répétitif.

Ces entités peuvent être regroupées pour exprimer certains faits :
1) PATIENT et PE pour décrire le fait qu'un patient doit subir un PE.
2) PE et SPECTRE pour décrire le fait qu'un spectre donné est produit par la réalisation d'un PE défini.

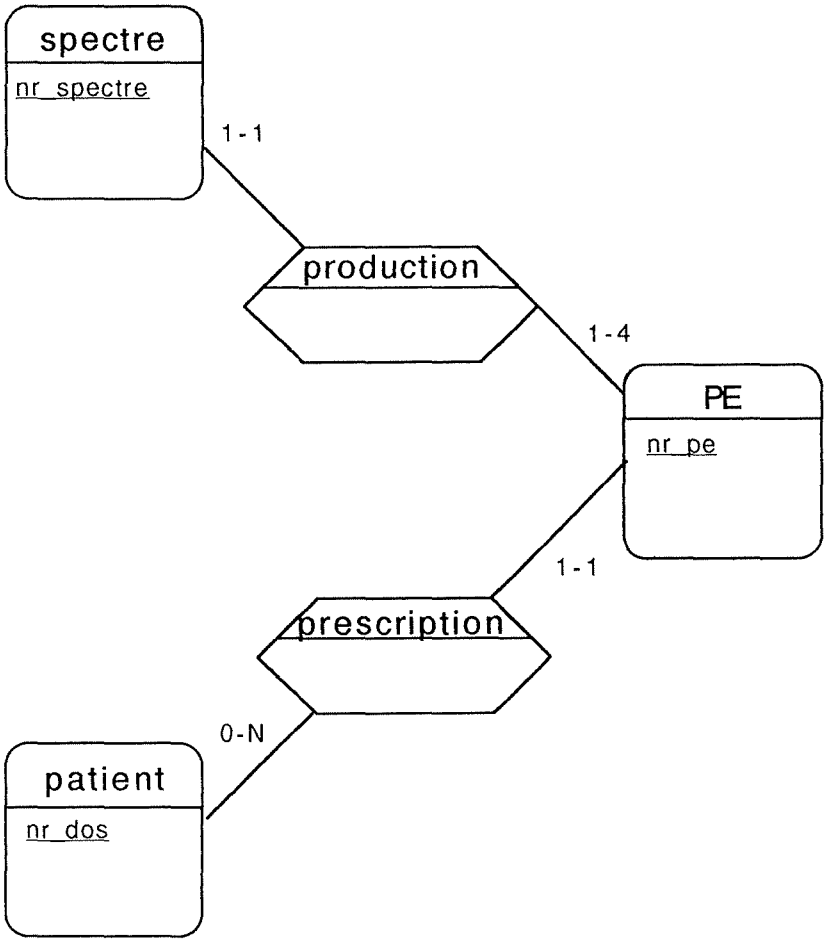
On a ainsi deux associations : PRESCRIPTION et PRODUCTION.

ASSOCIATION : PRESCRIPTION.
DEFINITION.
Associe à un PATIENT le PE qu'il doit subir.
ATTRIBUTS.
/

ASSOCIATION : PRODUCTION.
DEFINITION.
Associe à un PE le SPECTRE que son acquisition produit.
ATTRIBUTS.
/

III.1.2: le schéma entité/association

On aboutit finalement au schéma Entité/Association suivant :



III.2. ELABORATION DU SCHEMA CONCEPTUEL DES TRAITEMENTS.

Nous entrons ici dans la deuxième étape de la structuration: la découpe en fonctions de chaque phase.
Quoique ce travail pourrait être fait pour chaque phase des applications, nous ne développerons ici que les traitements dans lesquels l'automatisation par informatique intervient.
En plus de la découpe en fonction , nous accompagnerons, quand cela est nécessaire, un schéma d'enchaînement de ces fonctions.

Glossaire:

Comme certains concepts importants ne se trouvent pas définis dans le schéma E/A, nous commençons par les définir.

1- une **trace**: tout ensemble de données représentant un signal analogique digitalisé.
2- une **moyenne**: tout ensemble ordonné de données représentant la moyenne arithmétique de plusieurs **traces**.

3- une **somme**:tout ensemble ordonné de données représentant l'addition de plusieurs **traces**.
4- **amplitude**: valeur ,exprimée en "microvolt", associée à une donnée d'une **moyenne**, exprimant la différence de potentiel mesurée.
5- **latence**: valeur, exprimée en milliseconde, associée à une donnée d'une **moyenne**, exprimant le retard entre la première donnée de la moyenne et celle-ci.
6 **fréquence**: nombre de conversions "analogique/digitale" faites par seconde.
7- **rapport médical**: ensemble des informations nécessaires (PE, PATIENT, SPECTRE) pour établir un bilan médical.
8- **examen** : correspond à un **PE**, un **PATIENT** associé et les **SPECTREs** associés.
9- **graphe**: toute visualisation de **traces** ou **moyennes** sous forme graphique.
10- **rapport_médical:** ensemble des informations recueillies dans la base de données pour établir un rapport complet sur la mesure effectuée.

Nous utiliserons également les abréviations suivantes:

S.I : le système d'information

U.A: l'unité d'acquisition

III.2.1 Acquisition.

III.2.1.1 Phase d'initialisation de l'unité d'acquisition.

Spécifications

nom:

Init_acquisition

définition:

Cette phase initialise l'unité d'acquisition et programme le type de P.E. dont on fera l'acquisition.

reçoit:

.un message de demande d' acquisition

.tous les attributs nécessaires à l' enregistrement d' un **PE:**

type_pe, FH, FB, nbre_stimulation, type_stimulation, récurrence, fenêtre_analyse, nbre_canaux, seuil_rejet,plus les attributs facultatifs suivant le type de P.E. souhaité.

génère:

l'unité d'acquisition initialisée

action sur l' U.A:

L'ajustage des différents paramètres

Ces paramètres ne sont pas nécessairement identiques à ceux d'entrée car certains contrôles de validité sont effectués avant l'ajustage proprement dit.

règle:

Chaque ressource contribuant à l'acquisition est controlée.

Chaque paramètre est vérifié suivant les contraintes indiquées dans le schéma des informations.

Le réglage suit.

découpe en fonctions:

1. contrôle de l'appareillage

   nom: Init_hardware

   reçoit une demande d'acquisition

   Action sur l'U.A: le setup et le test de l'unité d'acquisition

   exception: "Erreur hardware"

2. réglage des paramètres de stimulation:

   nom: Réglage_stimulation

   reçoit: un **type_stimulation** valide, une **récurrence** valide

   un **nbre_stimulation** valide

   action sur l'U.A: le setup de l'unité de stimulation

3. réglage des paramètres du filtre

   nom: Réglage_filtre

   reçoit une **FH** et une **FB** valides

   action sur l' U.A: le setup des filtres analogiques

4. réglage de la fréquence d' échantillonnage

   nom: Réglage_convertisseur

   reçoit une **FH** et une **FB** valides

   un **nbre_canaux** valide

   une **fenêtre_analyse** valide

   génère une **résolution**

action sur l'U.A: réglage de la **fréquence** d'échantillonnage
règle : la fréquence sera **2*FH*nbre_canaux** en tenant compte de la loi de Nyquist (voir plus haut) et du multiplexage des canaux d'acquisition. La **résolution** sera fonction de la **fréquence** et de la **fenêtre_analyse**.

5. réglage du seuil de rejet

nom: Réglage_seuil
reçoit un **seuil_rejet** valide
génère un **seuil_rejet** d'artefacts.

6. réglage de la fenêtre d' analyse

nom: Réglage_temps
reçoit **fenêtre_analyse** valide
action sur l' U.A: setup de la fenêtre d'analyse

7. initialisation des fenêtres graphiques

nom: Init_graph
reçoit **fenêtre_analyse**
gain
type_pe
génère les fenêtres graphiques nécessaires au contrôle visuel de la mesure à effectuer.
exception: " Erreur unité graphique"

Les informations saisies sont nécessairement contrôlées avant toute exécution des fonctions de réglage en vue de respecter l'intégrité du système. Ces contrôles sont regroupés dans les fonctions suivantes.

8. vérification des paramètres de stimulation

nom: Stimulation_valide
reçoit un **type_stimulation**, une **récurrence**,un **nbre_stimulation** .
génère un **type _stimulation** valide, une **récurrence** valide, un **nbre_stimulation** valide
exception: " Nombre de stimulation invalide"
" Récurrence invalide "
" Type_stimulation invalide "

9. vérification des paramètres des filtres

nom: Filtres_valides
reçoit une **FB** et une **FH**
génère une **FB** et une **FH** valides
exception: " Bande passante invalide"

10. vérification du nombre de canaux

nom: nr_canaux_valide
reçoit un **nbre_canaux**
génère un **nbre_canaux** valide
exception: " Nombre de canaux invalides "

11. vérification du seuil de rejet

nom: Seuil_valide
reçoit un **seuil_rejet**
génère un **seuil_rejet** valide
exception: " Seuil invalide "

12. vérification de la fenêtre d'analyse

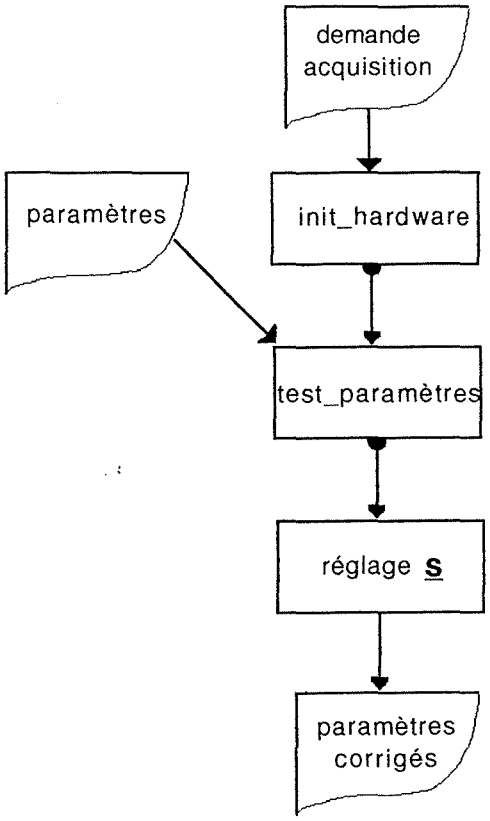
nom: Fenêtre_valide
reçoit une **fenêtre_analyse**
génère une **fenêtre_analyse** valide
exception: "Fenêtre invalide"

Rappelons que la fenêtre d'analyse est aussi fonction de la récurrence des stimulations, paramètre inclu dans le type de stimulations. Il faut donc recourir à une fonction de validation supplémentaire.

13. vérification de la cohérence récurrence-fenêtre

nom: Fenêtre_cohérente
reçoit une **fenêtre_analyse** valide
une **récurrence** valide
génère une **fenêtre_analyse** cohérente
exception: "Fenêtre incohérente avec la récurrence des stimulations"
règle : l'intervalle entre deux stimuli doit être supérieur à la fenêtre_analyse.

Nous rassemblerons ces fonctions de validation en une fonction globale appelée Test_paramètres.



III.2.1.2 Phases de conversion digitale et transfert en mémoire

Spécifications

Ces deux phases sont réunies car elles collaborent de façon très étroites.

nom:

Conversion_transfert

définition:

L'une échantillonne le signal analogique d'entrée, l'autre transfert la donnée en mémoire pour constituer petit à petit une trace par canal d'acquisition.

reçoit:

un signal analogique par canal d'acquisition

génère:

une **trace** par canal

action sur l' U.A:

création d'une trace par canal

règle:

S'il y a plusieurs canaux d'acquisition , le convertisseur échantillonne ceux-ci à tour de rôle.

Nous ne prolongeons pas la spécification de ces phases car elles sont en grande partie de nature hardware.

III.2.1.3 Phase de rejet des artefacts.

Cette phase sera appliquée à chaque trace avant d'effectuer la moyenne. On peut donc résumer cette phase en une simple fonction de validation de données et la regrouper avec la phase de moyennage.
Notons qu'on aurait pu choisir, comme beaucoup de systèmes actuels, une toute autre façon de résoudre ce problème. Celle-ci prévoit plutôt un circuit actif de rejet des artéfacts qui élimine le signal avant même qu'il ne soit codé. Cette solution diminue évidemment la partie informatique.

III.2.1.4 Phase de moyennage des signaux.

Spécifications

nom:

Moyennage

définition:

Cette phase a pour objectif d'additionner et moyenner des traces.

reçoit:

une **trace** par canal d'acquisition
un **seuil_rejet** valide

génère:

la **somme** et la **moyenne** des traces déjà accumulées pour chaque canal

exception:

"Artefact "

action sur l' U.A:

modification des **sommes**

modification des **moyennes**

règles:

Aucune trace contenant une donnée supérieure à **seuil_rejet** ne fait partie d'une somme ou d'une moyenne.

découpe en fonctions:

1. rejet des artefacts

nom: Rejet_artefacts

reçoit une **trace**

un **seuil_rejet**

génère une **trace** valide

exception: " ARTEFACT "

règle : il y a artefact lorsque un point de la trace, au moins, est supérieur à **seuil_rejet.**

2. addition des données

nom: Additionneur

reçoit une **somme** initiale

trace valide

génère une **somme**

action sur l' U.A: modification des sommes

modification des moyennes

règle: la somme initiale est additionnée, point par point, à la **trace** valide; le résultat est la somme finale.

3. mise à jour de la moyenne

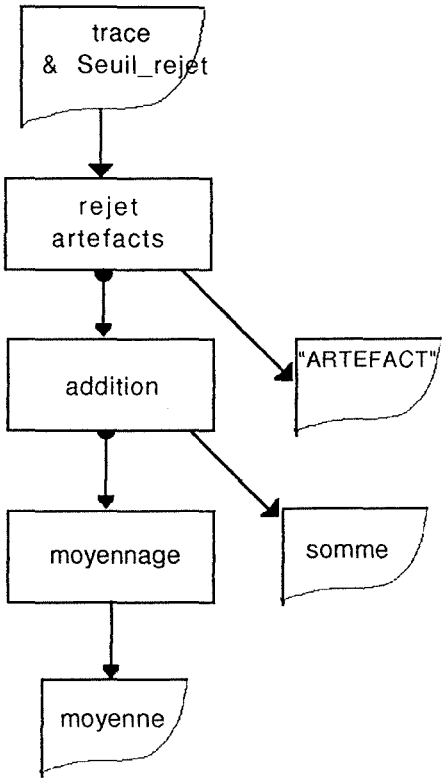
nom: Moyenneur

reçoit une **somme**

génère une **moyenne**

action sur l' U.A: modification des moyennes

règle: chaque point de la somme est divisé par le nombre de traces déjà accumulées.



III.2.1.5 Phase d'affichage courant.

Spécifications:

nom: Affichage_courant

définition:

Cette phase a pour objectif de transformer chaque trace et chaque moyenne en un graphe que l'on affichera pour contrôler l'acquisition.

reçoit:

par canal d'acquisition:

- une **trace**
- une **moyenne**

si pas de message de rejet d'artefact.

génère:

par canal:

- un **graphe** représentant la **trace**
- un **graphe** représentant la **moyenne**

exception:

"Artefact "

règle:

-Lorsqu'un artefact intervient , il n'y aura pas d'affichage, mais la mention "Artefact".

-Le graphe aura une allure interprétable ; le choix des axes sera dicté par la fenêtre d'analyse et l'amplitude du plus haut pic.

-Le graphe de la moyenne ne sera pas présenté à chaque fois; lorsqu'il apparaîtra, le graphe de la trace n'est plus affiché.

-Les conventions sont : les pics(maxima du graphe) négatifs pointent vers le haut, les pics positifs,vers le bas.

découpe en fonctions:

1. affichage des traces courantes

nom: Affichage_trace
reçoit une **trace**
génère un **graphe**

2. affichage des moyennes courantes

nom: Affichage_moyenne
reçoit **moyenne**
génère **graphe**

III.2.1.6 Phase d' affichage terminal.

Spécifications

Cette phase a pour objectif de transformer les moyennes obtenues en graphes correspondant à la visualisation du résultat final.

nom:

Affichage_final

définition:

Il s'agit , ici, de présenter la réponse évoquée finale.

reçoit:

par canal d' acquisition:

- une **moyenne**

génère:

par canal:

- un **graphe** représentant la **moyenne** terminale

règles:

-Le graphe aura une allure interprétable ; le choix des axes sera dicté par la fenêtre d'analyse et l'amplitude du plus haut pic.

-Les conventions sont : les pics(maxima du graphe) négatifs pointent vers le haut, les pics positifs,vers le bas.

découpe en fonctions:

1. affichage des réponses évoquées

nom:	Affichage_PE
reçoit	une **moyenne**
génère	un **graphe**

III.2.1.7.Phase d' enregistrement des spectres.

Spécifications

nom:

enregistrement_examen

définition:

Cette phase a pour objectif d'insérer un examen de P.E. dans la base de donnée, c-à-d enregistrer le **PE** ainsi que le **PATIENT** , s'il n'existe pas déjà, et les **SPECTREs** associés

reçoit:

- tous les attributs nécessaires à l'enregistrement d'un P.E.:
  type_pe, FH, FB, nbre_stimulation, type_stimulation, récurrence, fenêtre_analyse, nbre_canaux,seuil_rejet, plus les attributs facultatifs suivant le type de P.E. souhaité.
- un **commentaire** "à chaud" sur l'examen.
- les coordonnées du patient:
  nom, prénom, dnaiss, adresse, téléphone, sexe, nom_médecin, taille.
- associé à chaque **moyenne** recueillie:
  canal, gain, résolution

génère:

un **nr_pe,** un **nr_dos** si le PATIENT n'existait pas encore dans le S.I.
message possible: "Le patient est déjà dans la B.D"

action sur le S.I:

un examen de P.E. dans la base de données :

- création d'un **PE**
- création d'un **PATIENT** associé s'il n'existait pas dans la base de données.
- création des **SPECTREs** associés
- création des associations **prescription** et **production.**

règles:

Les attributs de **PE** seront les paramètres de réglage ainsi qu'un commentaire ("à chaud") sur l'examen.
L'identifiant **nr_pe** est attribué par incrémentation plus concaténation avec la date du jour.
Si le PATIENT existe déjà, il ne sera pas enregistré.
Les spectres sont les moyennes finales accompagnées de paramètres d'affichage tel que le gain ou la résolution (nombre de points par spectre).

découpe en fonctions:

1. mémorisation de la réponse évoquée

nom: PE_enregistré
reçoit **type_pe, FH, FB, nbre_stimulation, type_stimulation, récurrence, fenêtre_analyse, nbre_canaux,seuil_rejet,** plus les attributs facultatifs suivant le type de P.E. souhaité.
un **commentaire** " à chaud " sur l' examen.
génère un **nr_pe**
action sur le S.I:création d'un **PE**

2. mémorisation des spectres

nom: Spectres_enregistrés.
reçoit: associé à chaque **moyenne** recueillie:
canal, gain, résolution
génère les **SPECTREs** enregistrés
action sur le S.I: des **SPECTREs** dans la base de données

3. mémorisation du patient

nom: Pat_enregistré

reçoit

- un message "Nouveau patient dans la B.D"
- les coordonnées du PATIENT:

nom, prénom, dnaiss, adresse, téléphone, sexe, nom_médecin, taille.

génère un **nr_dos**

action sur le S.I: un **PATIENT** dans la base de données

4. vérification de l' existence du patient

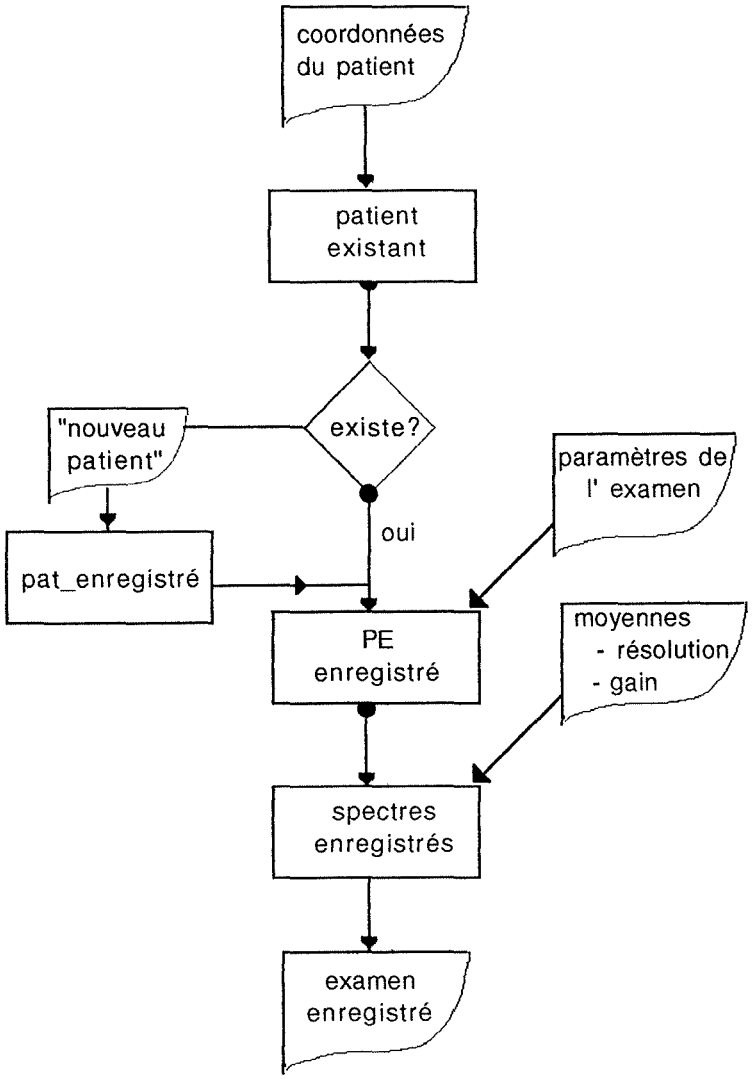
nom: Patient_existant

reçoit **nom, prénom, dnaiss**

génère "Le patient est déjà dans la B.D "

exception: "Patient inconnu dans la base de données"

règle : le triplet <nom,prénom,dnaiss> identifie un PATIENT éventuellement dans le S.I.



III.2.2.Traitements différés.

<u>III.2.2.1. La phase d'exploitation des spectres.</u>

spécifications:

nom: Travail_sur_spectres

définition:

Cette phase a pour objectif l'interprétation d' une mesure de P.E. par des outils de traitement mathématiques et graphiques.

reçoit:

.les **SPECTREs** d'un **PE** et les **graphes** associés

.un **nr_pe**

.un **nom** de **pic**

génère:

- l'**amplitude** des points des **SPECTREs**
- la **latence** des points des **SPECTREs**
- la superposition des **SPECTREs** d'entrée avec les **SPECTREs** issus de **nr_pe**
- la modification de l'allure des graphes ,représentant les SPECTREs, due au changement de **gain.**
- .un ou des pics enregistrés

action sur le S.I:

crée ou modifie l'attribut **pic** de l'entité **SPECTRE**

règle:

Les actions entreprises ne modifient pas la base de données.excepté pour l'enregistrement de l'attribut **pic** de **SPECTRE .**
Elles sont réalisées en interaction avec l'utilisateur.

découpe en fonctions:

1. superposition de réponses évoquées

nom:	Superposition
reçoit:	**nr_pe** valide les **graphes** associés aux **SPECTREs** d'un **PE**
génère:	la superposition des **graphes** d' entrée avec les **graphes** issus de **nr_pe**

2. calcul de l'amplitude absolue

nom:	Amplitude_absolue
reçoit:	un point du **graphe** représentant la **moyenne** du **SPECTRE**
génère:	une **amplitude** absolue pour ce point
règle:	l'**amplitude** est la valeur absolue de l' ordonnée du point convertie en "µV"

3. calcul de l'amplitude relative

nom:	Amplitude_relative
reçoit	deux points du **graphe** représentant la **moyenne** du **SPECTRE**
génère	une **amplitude** relative pour ce point

règle: l' **amplitude** est la valeur absolue de la différence entre les ordonnées des points convertie en "µV"

4. calcul de la latence

nom: Calcul_latence

reçoit: un point du **graphe** représentant la **moyenne** du **SPECTRE**

génère une **latence** pour ce point

règle: la **latence** est l'abscisse du point convertie en "ms"

5. enregistrement des caractéristiques d' un pic

nom: Pic_enregistré

reçoit: une **amplitude** absolue
une **latence**
un **nom** valide de pic

génère l'attribut pic de **SPECTRE** modifié

action sur le S.I: crée ou modifie l'attribut **pic** de l'entité **SPECTRE**

règle: amplitude et latence correspondent aux coordonnées d'un même point du graphe représentant le SPECTRE.

6. modification des gains

nom: Modification_gain

reçoit un **graphe**
un **gain** valide

génère la modification de l'allure des graphes ,représentant les SPECTREs, due au changement de **gain.**

Et les fonctions de validation sont:

7. validation d'un potentiel évoqué

nom: PE_existant

reçoit un **nr_pe**

génère un **nr_pe** valide

exception: "Potentiel évoqué invalide"

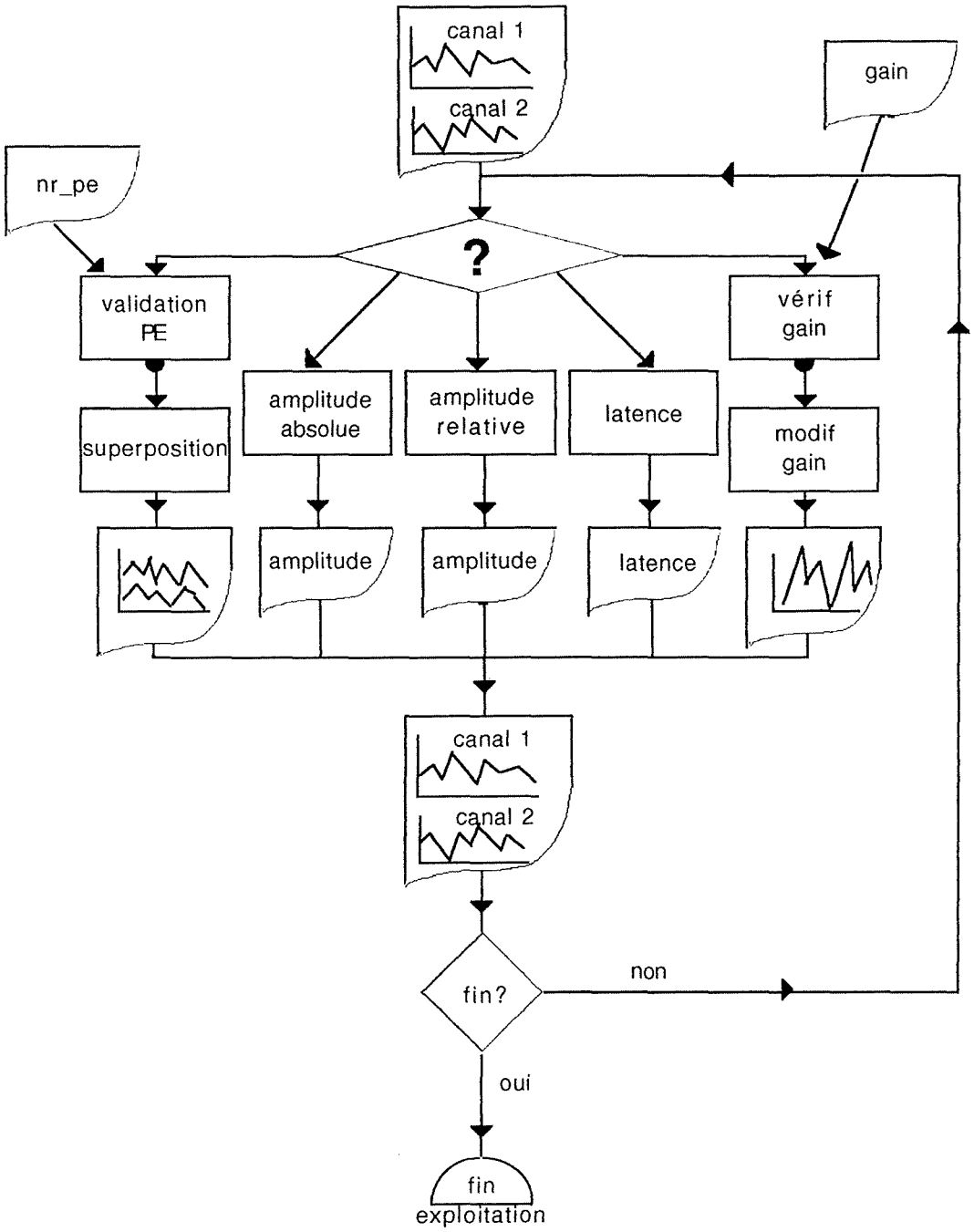
8. validation du gain

nom: Gain_valide

reçoit un **gain**

génère un **gain** valide
exception: "Gain invalide "

9. validation du nom de pic
nom: Pic_existant
reçoit un **nom**
génère un **nom** valide
exception:" Ce pic a déjà été enregistré "



III.2.2.2. La phase d'impression du rapport d'examen.

Spécifications

nom: Impression_rapport_examen

définition:

Cette phase a pour objectif d'imprimer un rapport complet sur un examen de P.E.

reçoit:

une demande d'impression

un **nr_pe**

génère:

un rapport imprimé

règle:

le rapport contiendra des informations sur l'examen: paramètres du **PE**, coordonnées du **PATIENT** et **SPECTRE**s réalisés.

découpe en fonctions:

1. rassemblement des informations

nom:	Création_du_rapport
reçoit	un **nr_pe** valide
génère	**rapport_médical**

2. impression du rapport

nom:	Création_du_rapport
reçoit	**rapport_médical**
génère	impression du rapport

L'accomplissement de cette fonction nécessite un contrôle préalable de l'état de la ressource d'impression .

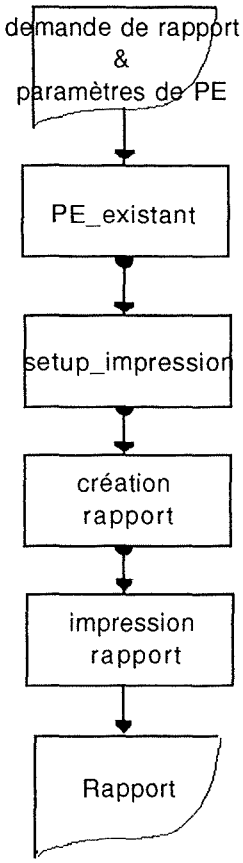
3. initialisation de l'impression

nom:	Setup_impression
reçoit	demande d' impression

génère le setup de l'unité d'impression
exception: " Erreur à l'unité d'impression"

Un contrôle de validité sur les données à imprimer est également nécessaire.

4. validation d'un potentiel évoqué
nom: PE_existant
reçoit un **nr_pe**
génère un **nr_pe** valide
exception: " Potentiel évoqué invalide "



III.2.2.3. La phase de chargement de SPECTREs.

Spécifications

nom: charger_PE

définition:

Cette phase a pour objectif d'accéder à un examen de P.E. dans la base de donnée, c-à-d de charger les **SPECTRE**s associés à ce P.E . Ceux-ci seront affichés .

reçoit:

un **nr_pe**

génère:

les graphes des **SPECTREs** associés au **PE.**

règle:

tout affichage de P.E par charger_PE efface automatiquement les informations présentes .

découpe en fonctions:

1. chargement de SPECTRE

   nom: Charger_SPECTRE

   reçoit un **nr_pe** valide

   génère les **SPECTREs** associés

2. affichage de SPECTREs

   nom: Affichage_SPECTREs

   reçoit **SPECTREs**

   génère les graphes associés

Et la fonction de validation est :

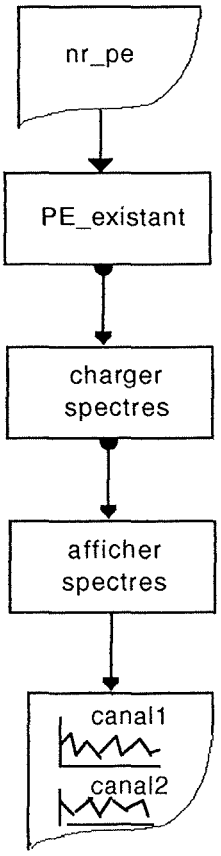
3. validation d'un potentiel évoqué

   nom: PE_existant

   reçoit un **nr_pe**

   génère un **nr_pe** valide

   exception: " potentiel évoqué invalide "



III.2.2.4. La phase d' archivage .

Spécifications

nom: Archivage

définition:

Cette phase a pour objectif d'archiver la base de données, en éliminant les enregistrements à partir d'une certaine date.

reçoit:

un message de demande d'archivage
une **date_archivage**

génère:

un ou plusieurs **PE** transférés
les **SPECTRE**s transférés
des **PATIENT**s dupliqués et transférés
message possible: "Il n'y a rien à archiver "

action sur le S.I:

Le transfert d'entités **PE** hors du S.I.
le transfert d'entités **SPECTREs** hors du S.I.

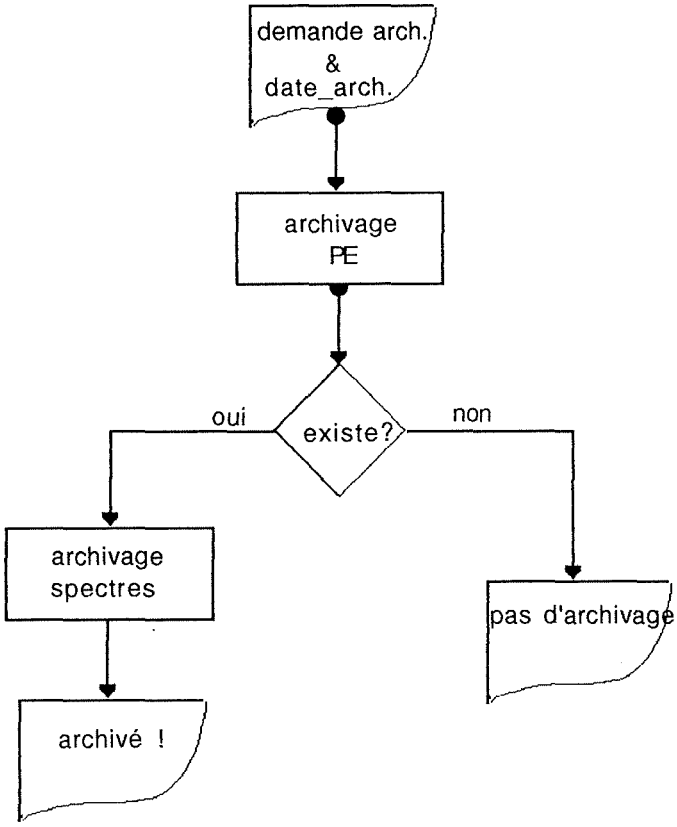
règle:

Les **SPECTREs** transférés sont ceux associés aux **PE** transférés.
Les **PE** transférés sont déterminés par la **date_archivage:**
seront compris dans l' archivage, tous les **PE** dont la **date** est inférieure à **date_archivage.**
Les **PATIENTs** concernés ne seront pas transférés, mais cependant les informations les concernant seront dupliquées et transférées.

découpe en fonctions:

1. archivage de P.E.
   nom: Archivage_PE
   reçoit une demande d'archivage
   une **date_archivage**
   génère un ou plusieurs **PE** archivés
   des **SPECTREs** à archiver
   des **PATIENTs** dupliqués et transférés
   exception: " Il n' y a rien à archiver "
   action sur le S.I: suppression de **PE**
   règle: il est nécessaire que la **date** du ou des **PE** soit inférieure à **date_archivage.**

2. archivage des SPECTREs associé à un P.E.
   nom: Archivage_SPECTREs
   reçoit des **SPECTREs** à archiver
   génère des **SPECTREs** transférés
   action sur le S.I: suppression de **SPECTREs**



III.3 SPECIFICATIONS FONCTIONNELLES.

III.3.1 Introduction

A la lumière du schéma des données et du schéma des traitements,cinq fonctionnalités de base se dégagent pour la construction du logiciel.
Il nous reste à définir pour chacune, les propriétés et type de leurs arguments et résultats. C'est l'étape de spécification fonctionnelle qui sera réalisée à l'aide du langage des prédicats du premier ordre.(Voir annexe 8 pour plus de renseignements)

La formulation des fonctions et des données du système portera uniquement sur les fonctions et données qui nous intéressent dans ce travail. On trouvera, ainsi détaillées, les 5 fonctionnalités de base suivantes :

1) **ACQUISITION**
2) **CHARGER _PE**
3) **INTERPRETATION_PE**
4) **IMPRESSION_RAPPORT_EXAMEN**
5) **ARCHIVAGE**

Le langage de spécification utilisé est celui qui a été introduit dans le cours de "Méthodologie de Développement de Logiciel"[1] .

Les fonctionnalités seront spécifiées suivant un canevas standard ;

[1] E. Dubois, année académique 90-91

- Arguments et résultats de la fonction.
- Exceptions éventuelles.
- Précondition et postcondition.
- Type de données.

Rappelons que les préconditions recouvrent les propriétés qui doivent être satisfaites pour que la fonctionnalité s'exécute correctement tandis que les postconditions explicitent les propriétés des résultats exprimées à l'aide de relations entre arguments et résultats.

III.3.2 Spécification

Fonctionnalité : ACQUISITION

Interface.

Arguments

paramètres, caract_pat

Résultats

un **nr_pe**

les SPECTREs du **PE** identifié par **nr_pe**

un nr_dos si le PATIENT n'existait pas dans le S.I

.

Exception

mes1: " Type de stimulation invalide"

mes2: " Filtres invalides "

mes3: " Nombre de canaux invalides"

mes4: " Fenêtre invalide"

mes5: " Récurrence invalide"

mes6: " Erreur hardware "

* Les arguments se présentent sous la forme d'un ensemble de valeurs qui vont fixer
*..directement ou indirectement le **PE** à réaliser.

Règle de traitement

Précondition :

Init_acquisition(paramètres) AND

Start_acquisition()

Postcondition :

Affichage_terminal(moyennes) AND Enregistrement_examen()

Excepté :

mes1 si (Test_paramètres()=1)
mes2 si (Test_paramètres()=2)
mes3 si (Test_paramètres()=3)
mes4 si (Test_paramètres()=4)
mes5 si (Test_paramètres()=5)
mes6 si Arrêt_acquisition()

Structure de données

paramètres : PC [**type_pe**,STRING; **FH**, INT; **FB**, INT; **stimulation**,PC; **fenêtre_analyse**,INT ; **seuil_rejet**, INT ; **canal1**,STRING; **canal2**,STRING; **nbre_canaux**,INT; **auriculaire**,STRING; **oculaire**,STRING; **durée**,INT; **intensité**,INT; **compteur**,INT; **date**,STRING ; **pic** , PC]

stimulation : PC [**nbre_stimulation** ,INT; **type_stimulation**,STRING; **récurrence**,INT]

nr_pe : INT

nr_SPECTRE : INT

nr_dos : STRING

caract_pat : PC[coord, PC; argument, PC]

argument : PC[**adresse**, STRING; **téléphone**, INT; **sexe**, CHAR; **nr_med**, STRING; **taille**, INT]

coord : :PC[**nom**, STRING; **prénom**, STRING; **dnaiss**, INT]

pic : PC[amplitude, INT; latence, INT; nom, STRING]

Init_acquisition(paramètres)

Interface

Arguments paramètres

Résultats paramètres

Règles de traitement

Précondition: /

Postcondition:

Init_hardware() AND
(Test_paramètres (paramètre)
AND Réglage_stimulation(type_stimulation_valide)
AND Réglage_temps(fenêtre_analyse_valide)
AND Réglage_filtre(FH_valide,FB_valide)
AND Réglage_convertisseur(FH_valide,nr_canaux_valide))

AND Allocation_par_défaut()

Structure de données

Paramètres : voir fonctionnalité 1

Enregistrement_examen()

Interface

Arguments paramètre, caract_pat, caract_SPECTRE, caract_pe

Résultats res

Règle de traitement

Précondition : /

Postcondition :

Spectres_enregistrés (liste_nr_SPECTRE,liste_ caract_spect)

AND Pe_enregistré (**nr_pe**, paramètres)

AND Pat_enregistré (**nr_dos**, caract_pat)

AND Allocation _nr_dos

AND Allocation_nr_pe

AND Allocation_nr_SPECTRE

Structure de données

paramètres : voir fonctionnalité 1

caract_pat : PC [coord, PC; argument, PC]

coord : PC [**nom**, STRING; **prénom**, STRING; **date_naiss**, INT]

argument : PC [**adresse**, STRING; **téléphone**, INT; **sexe**, CHAR;
nr_med,STRING; **taille**, INT]

liste_caract_spect : SEQ[PC [**moyenne**, BYTE;
résolution, INT;
gain, INT;
hémisphère,CHAR]]

res : PC[**nr_dos**, STRING; liste_nr_spect, SEQ of INT; **nr_pe**, INT]

Fonctionnalité CHARGER_PE

Interface

Arguments **nr_pe**

Résultats SPECTREs_à_l'écran

Exception mes

Règle de traitement

Précondition : pe_existant (**nr_pe**)

Postcondition : Affichage_pe(**nr_pe**)

Structure de données

nr_pe : INT

SPECTREs_à_l'écran : GRAPHE

Fonctionnalité TRAVAIL_SUR_SPECTRE

Interface

Arguments

SPECTREs_à_l'écran, **nr_pe**

Résultats

amplitude, latence

SPECTREs_superposés,SPECTRE_travaillé

Règle de traitement

Précondition : Pe_existant (**nr_pe**) AND Affichage_pe

Postcondition : ((amplitude_absolue OR amplitude_relative)AND Calcul_latence)

OR Modification_gain

OR Superposition_pe

Structure de données

nr_pe : INT

SPECTREs_à_l'écran :,SPECTREs_superposés ,SPECTRE_travaillé:GRAPHE

amplitude : INT

latence : INT

Fonctionnalité ARCHIVAGE

Interface

Arguments date_archivage

Résultats Pats_arch, Spects_arch, Pes_arch

mes "Archivage effectué"

Exceptés mes7 si ((date_archivage >date_jour) OR

(date_archivage<date_ancien_archivage))

mes8 si Supports_physiques_pas_près

Règle de traitement

Précondition : /

Postcondition : Pats_arch = Archivation_patient()

AND Spects_arch = Archivation_SPECTRE()

AND Pes_arch = Archivation_pe()

AND Archivage_Patient (Pats_arch)

AND Archivage_Spectre (Spects_arch)

AND Archivage_Pe(Pe_arch)

Exceptions : mes7 : "Date d'archivage invalide"
mes8 : "Supports physiques non près"

Structure de données

pats_arch : LISTE_PATIENT

pes_arch :

spects_arch :

Fonctionnalité IMPRESSION_RAPPORT_EXAMEN

Interface

Arguments **nr_pe**

Résultats Rapport

Exceptés mes9 si le **PE** de **nr_pe** n'existe pas
mes10 si imprimante non en état

Règle de traitement

Précondition : Spectre_existant (**nr_SPECTRE**) L test_état_imprimante

Postcondition : création_du_rapport L impression_rapport

Exception : mes9 : "Potentiel évoqué inexistant"
mes10 : "Imprimante non prête"

Structure de données

rapport : TEXTE_IMPRESSION (Voir annexe 2)

ETAT Etat_courant_UA

Opérations de réglage

Ces opérations ne sont pas spécifiées ici vu leur caractère particulier. Pour de plus amples informations, nous renvoyons le lecteur au chapitre 3.

1) Init_hardware()
2) Test_paramètre()
3) Réglage_stimulation(type_stimulation_valide)
4) Réglage_temps(fenêtre_analyse_valide)
5) Réglage_filtres(FH_valide,FB_valide)
6)Réglage_convertisseur(FH_valide,nr_canaux_valide)

Opérations de commande

1)Init_hardware()

Précondition:/

Postcondition: Carte initialisée.

2) Start_acquisition()

Précondition: Init_hardware() AND Test_paramètres ()

Poscondition: Acquisition lancée

3) Arrêt_acquisition()

Précondition: /

Postcondition: arrêt_Acquisition.

4) Fin_acquisition()

Précondition: /

Postcondition: Acquisition terminée.

ETAT Etat_courant_BD

Invariants : /

Structure du contenu: (voir modèle E/A)

Opérations

Opérations de consultation

1) Patient_existant_1 (coord)

* Teste l'existence d'un patient de coordonnées 'coord' dans la base de
*données(BD)

Précondition : /

Postcondition : In (Patient (coord))

Structure de données intermédiaires : coord = PC [**nom**, STRING; **prénom**, STRING; **date_naiss** : STRING]

2) Patient_existant_2(**nr_dos**)

* Teste l'existence d'un patient de numéro de dossier nr_dos dans la BD.

Précondition : /

Postcondition : In(**Patient (nr_dos)**)

Structure de données intermédiaires : **nr_dos** :STRING

3) Pe-existant (**nr-pe**)

*Teste l'existence d'un potentiel évoqué du numéro de nr_pe dans la BD.

Précondition :

Postcondition : In(**Pe(nr_pe)**)

Structure de données intermédiaires : **nr_pe** : INT

4) Allocation_nr_dos(coord)= **nr_dos**

* Fonction retournant le numéro de dossier pour un patient

* Si il existe ou en crée un sinon.

Précondition : /

Postcondition : if In (Patient(coord))

then **nr_dos** = **nr_dos(Patient**(coord))

else not In(**Patient(nr_dos)**)

Structure de données intermédiaires

coord : PC

nr_dos : STRING

5) Allocation_nr_pe() = **nr_pe**

* Fonction retournant le numéro de PE pour un PE que l'on veut enregistrer.

Précondition : /

Postcondition : not in(**Pe(nr_pe)**)

Structure de données intermédiaires **nr_pe** : INT

Opérations de mise_à_jour.

1) Pat-enregistré (**nr_dos**, caract_pat1)

* Enregistre dans la BD le patient de caractéristiques caract_pat1 et de numéro de dossier nr_dos.

Précondition : /

Postcondition : In (**Patient(nr_dos)**)

AND Caracpat (**Patient(nr_dos)**) = caract_pat1

Structure de données intermédiaires

nr_dos : STRING

caract_pat , caract_pat1: (voir fonctionnalité 2)

2) Pe_enregistré (nr-pe, paramètres1)

Précondition : /

Postcondition : In (**Pe(nr_pe)**
AND In(**prescription(nr_pe, nr_dos)**)
AND Paramètres (**Pe(nr_pe)**) = paramètres1

Structure de données intermédiaires

nr_pe : INT; **nr_dos** : STRING

paramètres, paramètres1: voir fonctionnalité 1.

3) Spectre-enregistré (**nr_SPECTRE**, caract_SPECTRE 1)

Précondition : /

Postcondition : In (**SPECTRE(nr-SPECTRE)**)
AND (caract_spect (**SPECTRE(nr-SPECTRE)**) = caract-spect1)

Structure de données intermédiaires

nr-SPECTRE : INT

caract-spect1 : caract_spect (voir fonctionnalité 2)

4)Archivation_patient()

Précondition : /

Postcondition : Pour tout **nr_dos** : not Patient (**nr_dos**)
⇒ il existe élém : élém appartienne à pat_arch
AND **nr_dos**(élém) = **nr_dos**
AND arg(élém) = arg(**PATIENT** (**nr_dos**))
AND coord(élém) = coord (**PATIENT(nr_dos)**)

Structure de données intermédiaires

nr-dos : STRING

pat-arch : PC [coord; arg; **nr_dos**, STRING]

coord : PC[**nom**, STRING; **prénom**, STRING; **dnaiss**, STRING]

5) Archivation_SPECTRE ()

Précondition : /

Postcondition : Pour tout **nr_SPECTRE** : not **SPECTRE** (**nr_SPECTRE**)
⇒ il existe élém : élém appartient à SPECTRE-arch
AND **nr-SPECTRE** (elem) = **nr-SPECTRE**
AND caract-spect(elem) = caract-spect(**nr-SPECTRE**))
AND **nr-pe**(elem) = est-produit-par$_{PRE}$ (**Spectre(nr-SPECTRE)**)
AND not in(**production(nr-SPECTRE, nr-pe**(elem))

Structure de données intermédiaires

nr_SPECTRE : INT

SPECTRE_arch :PC[coord, PC; arg, PC; **nr_dos**, STRING]

6) Archivation-Pe()

Précondition : /

Postcondition : Pour tout **nr-pe** : not **PE(nr-pe)**

⇒ qu'il existe elem ∈ pe-arch

AND **nr-pe**(elem) = **nr-pe**

AND paramètre(elem) = paramètre (**PE(nr-pe)**)

AND **nr-dos**(elem) = a-été-prescrit $_{PRE}$(**PE(nr_pe)**)

AND not in (**Prescription(nr_pe, nr_dos**(elem))

Structure de données intermédiaires

nr_pe : INT

nr-dos :STRING

paramètre : voir fonctionnalité 1

ETAT : ETAT_archives

---

1) Archivage_patient(Pats-arch)

Précondition : /

Postcondition : état-archives

=(état-archives $_{PRE}$, Archi-pats ← Archi-pats Uni à pats_arch)

2) Archivage_SPECTRE(spects-arch)

Précondition :

Postcondition : état_archives

=état_archives$_{PRE}$, Archi-spect ← Archi_spects Uni à spects-arch)

3) Archivage_Pe (Pes_arch)

Précondition :

Postcondition : état_archives

=état_archives $_{PRE}$, Archi-Pes ← Archi_Pes Uni à Pes_arch)

Structure du contenu

état_archives :

PC [Archi_pat : SET [PC[coord : PC

arg : PC

nr-dos : STRING

Archi-spects : SET [PC [nr_pe : STRING

caract_spect : PC

nr_SPECTRE : INT

Archi_pe : SET [PC [nr-pe : INT

paramètre : PC

nr-dos : STRING]]

CHAPITRE IV

Conception logique

IV.CONCEPTION LOGIQUE

L'architecture du logiciel est constituée d'une hiérarchie de modules reliés par la relation "utilise" . Ces modules regroupent des fonctions selon des critères tantôt fonctionnels pour l'application en temps réel, tantôt informationnels pour l' application traitements différés.
Nous respecterons une découpe en niveau dont voici les grandes lignes. Au niveau supérieur nous aurons deux modules distincts : le module ACQUISITION et le module TRAITEMENTS DIFFERES, le premier réalisant l'aspect 'temps réel', le second, l'aspect 'différé'.
Le niveau suivant sera composé de sept objets constituant le noyau fonctionnel. (figure 5.1) Une architecture classique donne le troisième niveau à l'interface utilisateur. Nous lui consacrerons le chapitre suivant. Vient alors le niveau comprenant les modules réalisant les accès à la base de données. On trouve enfin un dernier niveau (niveau physique) contenant trois objets permanents : CARTE, IMPRIMANTE et CARTE_GRAPHIQUE.

Remarque: nous avons mis dans les différents modules les fonctions permettant l'extension du logiciel sans toutefois en donner une quelconque spécification.

IV.1 SPECIFICATION DES MODULES DU NIVEAU SUPERIEUR.

Fonction : Acquisition
Uses: UNITE_ACQUISITION, GRAPHE.

Fonction : Travail_sur_spectre
Uses: Smooth, Amplitude_absolue, Amplitude_relative, Calcul_latence, Modification_gain, Pic_enregistré, Pic_existant.

Fonction : Impression_rapport_examen
Uses: TEXTE_IMPRESSION

Fonction : Charger_PE
Uses: PE_existant, Affichage_PE.

Fonction : Achivage
Uses: Archivage_PE, Archivage_spectre, Archivage_patient.

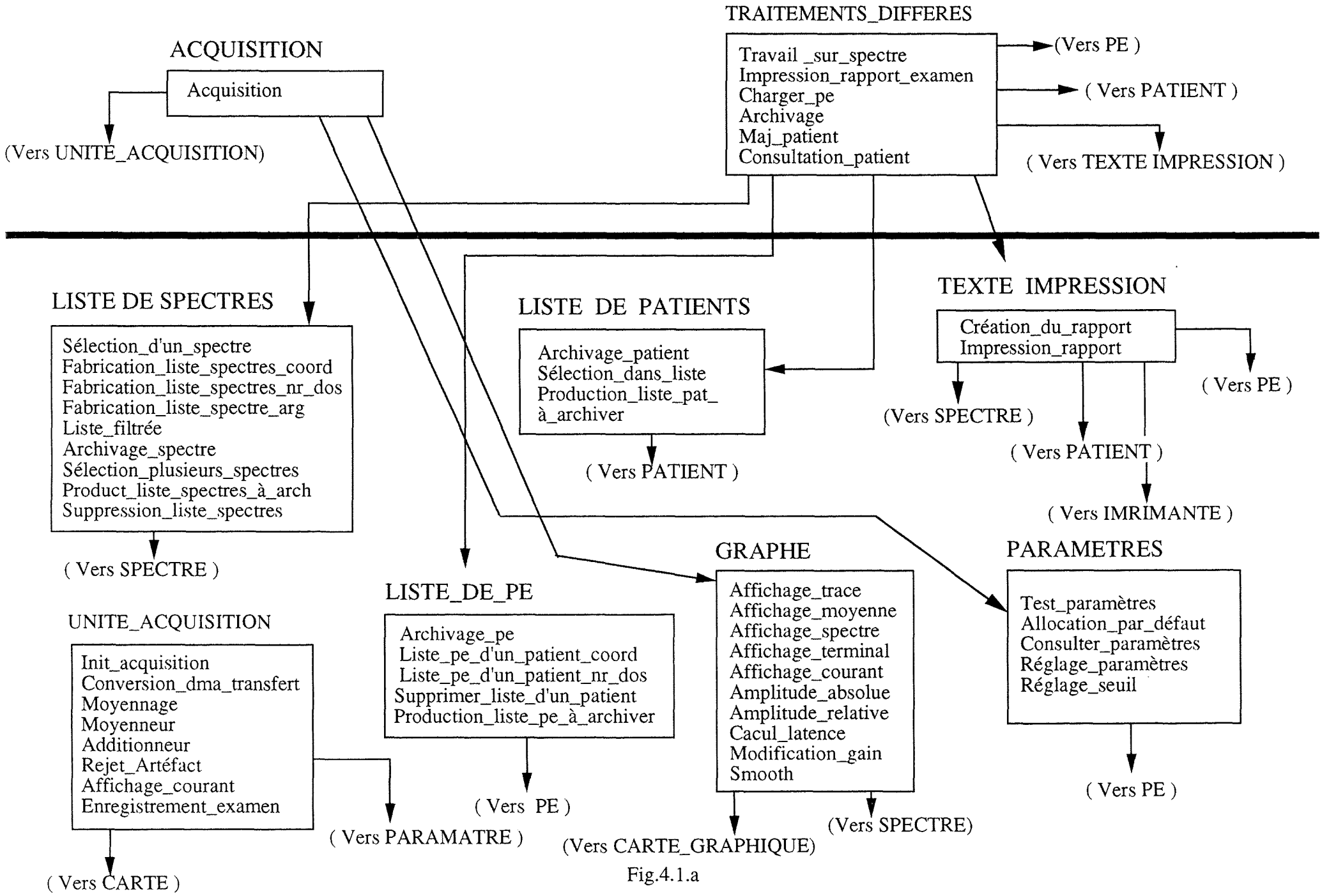

TRAITEMENTS_DIFFERES
Travail _sur_spectre
Impression_rapport_examen
Charger_pe
Archivage
Maj_patient
Consultation_patient
(Vers PE)
(Vers PATIENT)
(Vers TEXTE IMPRESSION)
ACQUISITION
Acquisition
(Vers UNITE_ACQUISITION)
LISTE DE SPECTRES
Sélection_d'un_spectre
Fabrication_liste_spectres_coord
Fabrication_liste_spectres_nr_dos
Fabrication_liste_spectre_arg
Liste_filtrée
Archivage_spectre
Sélection_plusieurs_spectres
Product_liste_spectres_à_arch
Suppression_liste_spectres
(Vers SPECTRE)
LISTE DE PATIENTS
Archivage_patient
Sélection_dans_liste
Production_liste_pat_
à_archiver
(Vers PATIENT)
TEXTE IMPRESSION
Création_du_rapport
Impression_rapport
(Vers PE)
(Vers SPECTRE)
(Vers PATIENT)
(Vers IMRIMANTE)
GRAPHE
Affichage_trace
Affichage_moyenne
Affichage_spectre
Affichage_terminal
Affichage_courant
Amplitude_absolue
Amplitude_relative
Cacul_latence
Modification_gain
Smooth
(Vers CARTE_GRAPHIQUE)
(Vers SPECTRE)
PARAMETRES
Test_paramètres
Allocation_par_défaut
Consulter_paramètres
Réglage_paramètres
Réglage_seuil
(Vers PE)
LISTE_DE_PE
Archivage_pe
Liste_pe_d'un_patient_coord
Liste_pe_d'un_patient_nr_dos
Supprimer_liste_d'un_patient
Production_liste_pe_à_archiver
(Vers PE)
UNITE_ACQUISITION
Init_acquisition
Conversion_dma_transfert
Moyennage
Moyenneur
Additionneur
Rejet_Artéfact
Affichage_courant
Enregistrement_examen
(Vers PARAMATRE)
(Vers CARTE)

Fig.4.1.a

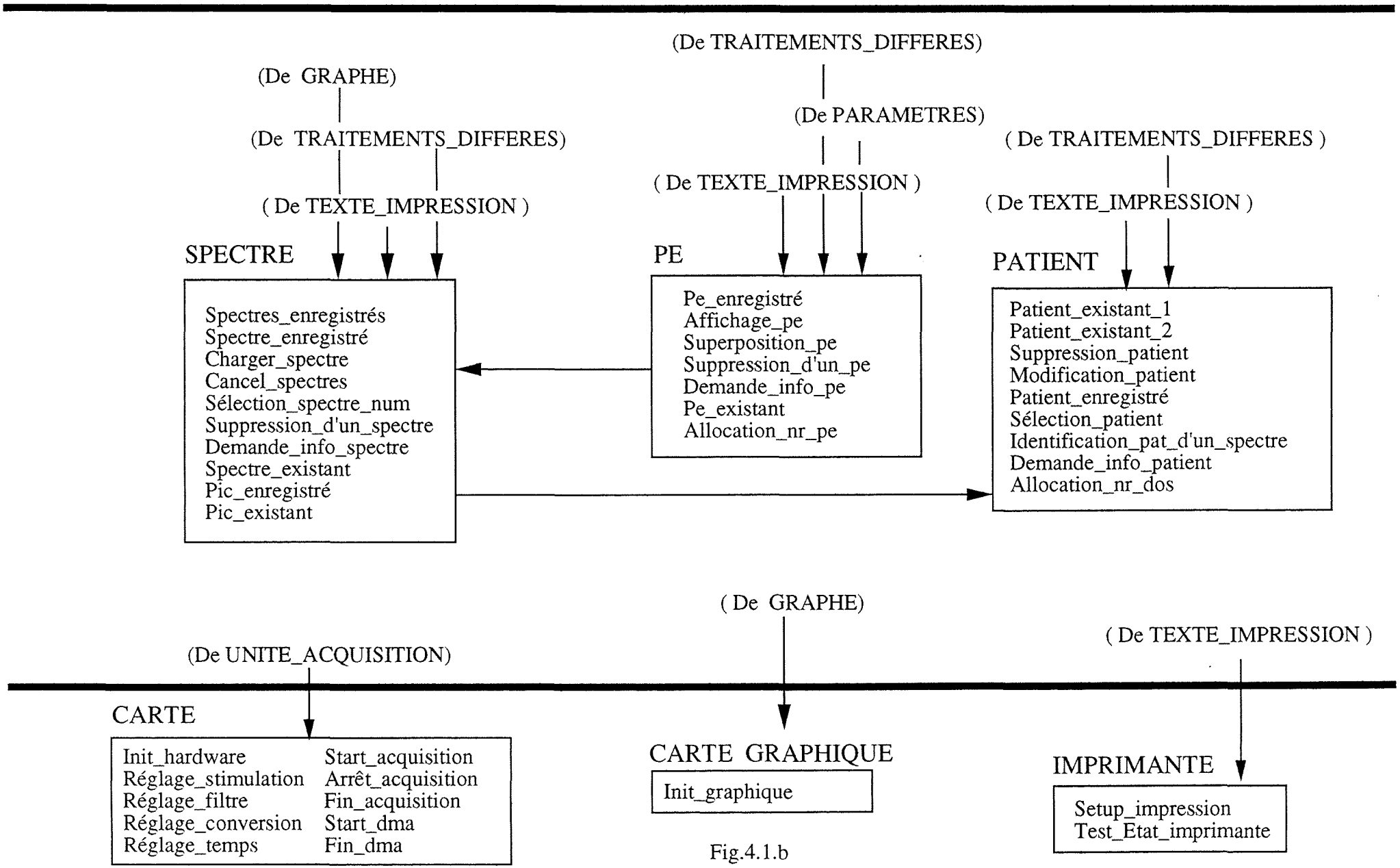

(De TRAITEMENTS_DIFFERES)
(De GRAPHE)
(De PARAMETRES)
(De TRAITEMENTS_DIFFERES)
(De TRAITEMENTS_DIFFERES)
(De TEXTE_IMPRESSION)
(De TEXTE_IMPRESSION)
(De TEXTE_IMPRESSION)
SPECTRE
Spectres_enregistrés
Spectre_enregistré
Charger_spectre
Cancel_spectres
Sélection_spectre_num
Suppression_d'un_spectre
Demande_info_spectre
Spectre_existant
Pic_enregistré
Pic_existant
PE
Pe_enregistré
Affichage_pe
Superposition_pe
Suppression_d'un_pe
Demande_info_pe
Pe_existant
Allocation_nr_pe
PATIENT
Patient_existant_1
Patient_existant_2
Suppression_patient
Modification_patient
Patient_enregistré
Sélection_patient
Identification_pat_d'un_spectre
Demande_info_patient
Allocation_nr_dos
(De GRAPHE)
(De TEXTE_IMPRESSION)
(De UNITE_ACQUISITION)
CARTE
Init_hardware
Réglage_stimulation
Réglage_filtre
Réglage_conversion
Réglage_temps
Start_acquisition
Arrêt_acquisition
Fin_acquisition
Start_dma
Fin_dma
CARTE GRAPHIQUE
Init_graphique
IMPRIMANTE
Setup_impression
Test_Etat_imprimante

Fig.4.1.b

IV.2. SPECIFICATION DES OBJETS.

1) Objet : LISTE_DE_SPECTRES;

Type:

SEQ [PC [**nr_pe** ,INT; **moyenne** ,SEQ[**point** , BYTE];**Hémisphère** ,STRING;**résolution**,INT;**gain** ,INT;**pic** ,PC[**amplitude**, INT; **latence**, INT; **nom**, STRING]]]

Interface

Invariant

Fonctions : Archivage-spectre.

Règles de traitement

Archivage_spectre

IN : liste_spectre : LISTE_DE_SPECTRE

OUT : /

2) Objet : LISTE_DE_PE

Type

SEQ[PC [**nr_pe**,INT ; **type_pe** , STRING ;**paramètres** , PARAMETRES ; **canal1**, STRING; **canal2** , STRING; .**auriculaire** , STRING; **oculaire** , STRING; **compteur** ,INT; **date** , STRING; **commentaire** ,STRING]]

Interface

Invariant

Fonctions : Archivage_pe

Règles de traitement

Archivage_pe

IN : liste_pe : LISTE_DE_PE

OUT :/

3) Objet : LISTE DE PATIENTS

Type:

SEQ [PC [**Nom** ,STRING; **Prénom** ,STRING; **dnaiss** ,STRING;**nr_dos** ,INT ;**adresse** ,STRING;**téléphone** ,STRING.;**sexe** , STRING;**nom-médecin** ,STRING;**taille** ,REAL]]

Interface

Invariant

Fonctions : Archivage_pat.

Règles de traitement

Archivage_pat

IN :SEQ **[nr_dos**: STRING]

OUT :

4) Objet : TEXTE_IMPRESSION

type: à déterminer suivant le type de PE [1]

Interface

Invariant

Fonctions : Création_du_rapport, Impression_rapport

Règles de traitement

Création_du_rapport

IN : **nr_pe**: INT

OUT : rapport_médical: TEXTE_IMPRESSION

Impression_rapport

IN : rapport_médical: TEXTE_IMPRESSION

OUT : impression d'un rapport [2]

5) Objet : PARAMETRES

Type:`

PC [**FH** :,INT; **FB** ,INT ; **nbre_stimulation** ,INT; **type_stimulation** ,STRING; **récurrence**,INT; **fenêtre_analyse**, INT; **nbre_canaux**,INT;**durée** ,INT; **intensité** ,INT; **fréquence** ,INT; **seuil_rejet**, INT]

Interface

Invariant

Fonctions : Test_paramètres, réglage_paramètres, réglage_seuil,Allocation_par_défaut,Modification_paramètres.

Règles de traitement

Test_paramètres

IN : **paramètres**: PARAMETRE

OUT : **status_erreur** : INT

Allocation_par_défaut

---

[1] voir annexe 2

[2] voir annexe 2

IN : **type_pe**: PE

OUT : **paramètres**: PARAMETRE

Modification_paramètre

IN : **paramètre**: PARAMETRE

OUT : **paramètre**: PARAMETRE

réglage_paramètres

IN : **paramètre**: PARAMETRE

OUT :

réglage_seuil

IN : **seuil_rejet**: INT

OUT :

6) Objet : SPECTRE

Type:

PC [**nr_pe** ,INT; **moyenne**,SEQ[**point** , BYTE];**Hémisphère** , STRING;**résolution**,INT;**gain**, INT;**pic**, PC[**amplitude**, INT; **latence**, INT; **nom**, STRING]]

Interface

Invariant

Fonctions :

Save_spectre, Cancel_spectre, Selection_spectre, Smooth, Suppression_d'un_spectre, Spectre_existantSpectre_enregistré, charger_spectres, affichage_spectres, spectre_enregistré, spectres_enregistrés, amplitude_absolue, amplitude_relative, calcul_latence, modification_gain, pic_enregistré.

Règles de traitement,

Spectres_enregistrés.

IN : SEQ [**moyenne** : SEQ[**point**: BYTE],**canal**: INT,**gain**:INT,**résolution**:INT)]

OUT : **Spectres_enregistrés**: SPECTRE

Spectre_enregistré

IN : **moyenne, canal, gain, résolution**

OUT : **Spectre**: SPECTRE

Cancel_spectre

IN : SEQ **[moyenne,canal,gain,résolution]**

OUT : /

Selection_spectre

IN : **nr_spectre**

OUT : **spectre**: SPECTRE

Suppression_d'un_spectre

IN : **nr_spectre** : INT

OUT : (message)

Spectre_existant

IN : **nr_spectre** : INT

OUT : BOOL

charger_spectres

IN : **nr_pe**, SEQ[**nr_spectre**: INT]

OUT : SEQ[**Spectre]**

pic_enregistré

IN : **amplitude**: INT, **latence**: INT, **nom_pic**: STRING

OUT : **spectre_modifié**: SPECTRE

pic_existant

IN : **nom_pic**: STRING

OUT : BOOL

7) Objet : GRAPHE

Type:

Interface

Invariant

Fonctions :

affichage_spectre, affichage_courant, affichage_terminal, affichage_moyenne, affichage_trace,amplitude_absolue, amplitude_relative, calcul_latence, modification_gain, pic_enregistré, Smooth.

Règles de traitement,

affichage_spectre

IN : **nr_spectre, nr_pe** : INT

OUT : **spectre_à_l'écran** : GRAPHE

affichage_courant

IN : **moyenne, trace** : SEQ[point,INT]

OUT : **graphe** : GRAPHE

affichage_terminal

IN : **moyenne** : SEQ[point,INT]

OUT : **graphe** : GRAPHE

affichage_trace

IN : **trace** : SEQ[point,INT]

OUT : **graphe** : GRAPHE

affichage_moyennee

IN : **moyenne** : SEQ[point,INT]

OUT : **graphe** : GRAPHE

amplitude_absolue

IN : **Spectre_à_l'écran** : GRAPHE

OUT : **spectre_à_l'écran** : GRAPHE, **amplitude** : INT

amplitude_relative

IN : **Spectre_à_l'écran** : GRAPHE

OUT : **spectre_à_l'écran** : GRAPHE, **amplitude**: INT

calcul_latence

IN : **Spectre_à_l'écran** : GRAPHE

OUT : **spectre_à_l'écran** : GRAPHE, **latence**: INT

modification_gain

IN : **Spectre_à_l'écran** : GRAPHE

OUT : **spectre_à_l'écran** : GRAPHE

Smooth

IN : **spectre_à_l'écran**: GRAPHE

OUT : **spectre_àl'écran**: GRAPHE

8) Objet PE

type:

PC [**nr_pe**, INT ; **type_pe**, STRING ;**paramètres**, PARAMETRE ; **canal1**, STRING; **canal2**, STRING; **.auriculaire**, STRING; **oculaire** , STRING;**compteur**, INT; **date**, STRING; **Commentaire**, STRING]

PARAMETRES :

PC [**FH**, INT; **FB**, INT ; **nbre_stimulation** , INT; **type_stimulation**, STRING; **récurrence**, INT; **fenêtre_analyse**, INT; **nbre_canaux**, INT;**durée**, INT; **intensité**, INT; **fréquence**, INT ; **seuil_rejet**, INT]

<u>rmq:</u> certains attributs sont facultatifs (déterminé par le type de PE demandé)

<u>Interface</u>

<u>Invariant</u>

<u>Fonctions</u> : Pe_enregistré, Pe_existant, Pe_enregistré

<u>Règles de traitement:</u>

Pe_enregistré

IN : **paramètre**:PARAMETRE, **type_pe**: STRING
OUT : **nr_pe**: INT

Pe_existant
IN : **nr_pe**: INT
OUT : BOOL

Suppression_pe
IN : **nr_pe**: INT
OUT : /

9) Objet PATIENT

type:

PC [**Nom**, STRING; **Prénom**, STRING; **dnaiss**, STRING;**nr_dos**, INT ;**adresse**, STRING;**téléphone**, STRING.;**sexe** , STRING;**nom_médecin**, STRING;**taille**, REAL]

Interface

Invariant

Fonctions : Patient_existant _1,Patient_existant_2,Patient_enregistré

Règles de traitement

Patient_existant_1
IN : **nr_dos**: INT
OUT : BOOL

Patient_existant_2
IN : **Nom**: STRING, **Prénom**,: STRING **dnaiss**: STRING
OUT : BOOL

Patient_enregistré
IN : **Nom** :STRING,**Prénom** :STRING,: **dnaiss**:STRING,**adresse**:STRING ,**téléphone**:STRING,**sexe**:STRING ,**nom_médecin**:STRING, **taille**: REAL
OUT : **nr_dos**: INT

10) Objet permanent CARTE

Le type et les primitives de cet objet dépendent directement des choix en matière d'implantation physique.Ils seront donc détaillés ultérieurement.

Nous nous donnons ici qu'un aperçu qualitatif du comportement de cet objet.

Interface

Invariant

Fonctions

Init_hardware,Start_acquisition,Arrêt_acquisition,Fin_acquisition,Start_dma,Fin_dma.,Réglage_stimulation,Réglage_filtre,Réglage_conversion,Réglage_temps.

Règles de traitement

Init_hardware: concerne les différentes actions nécessaires à l' initialisation et au contrôle de la carte d' interface, avant tout action de réglage .

Start_acquisition : commande de début d' acquisition au biveau de la carte.

Arrêt_acquisition : commande d' interruption d' acquisition au niveau de la carte.

Fin_acquisition : commande de fin normal d' acquisition.

Start_dma: commande de transfert de données vers la mémoire de travail.

Fin_dma.: signale la fin d' un transfert des données

Réglages: réglage par software des différentes unités déterminant le type d' un P.E à recueillir.

11) Objet Permanent IMPRIMANTE
(voir CARTE)

Interface

Invariant

Fonctions :

Test_état_imprimante: renvoit le status de l'imprimante.

setup_impression : connecte l'imprimante ou envoit les messages nécessaires à sa connexion.

11) Objet Permanent CARTE GRAPHIQUE
(voir CARTE)

Interface

Invariant

Fonctions :

Init_graphique : initialise le mode graphique de l'écran.

Acces à la base de données - SAP

Acces à la BD

PATIENT

Chercher le **PATIENT** de **coord** donné.

chercher le **PATIENT** de **nr_pat** donné.

chercher le **PATIENT** suivant le **patient** de **nr_pat** donné.

Chercher le **PATIENT** suivant le **PATIENT** de **coord** donné.

Chercher le **PE** du **PATIENT** de **nr_pat** donné.
Chercher le premier **PE** du **PATIENT** de **coord** donné.
chercher le **PE** suivant le **PE** du **PATIENT** de **nr_pat** donné.
Chercher le **PE** suivant le **PE** du **PATIENT** de **coord** donné.

PE

Chercher le **PE** de **nr_pe**.
Chercher le **PE** suivant le **PE** de **nr_pe** donné.
Chercher le **PATIENT** d'un **PE** de **nr_pe** donné.
Chercher le(s) **SPECTRE**(s) d'un **PE** de **nr_pe** donné.

SAP

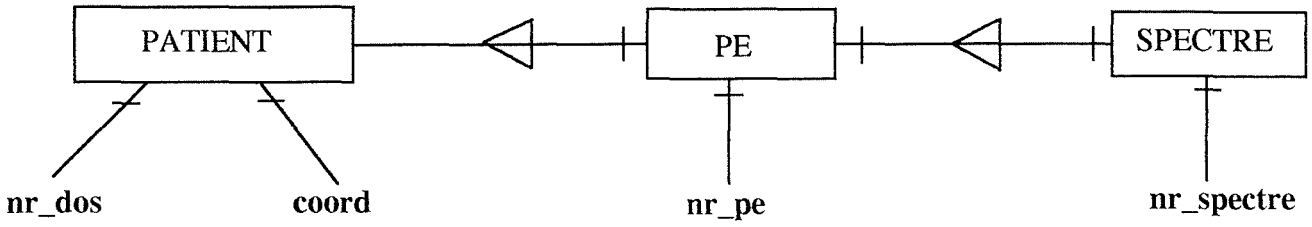


fig.4.3

Rmq : 1)Ne sont représentés pour chaque article que les attribus identifiant de ces articles.
2)L'attribut 'coord' de l'article PATIENT est attribut décomposable remplaçant sur le schéma le premier identifiant de PATIENT à savoir le triplet (nom,prénom,dnaiss).

CHAPITRE V

Interface

V SPECIFICATION DE L'INTERFACE

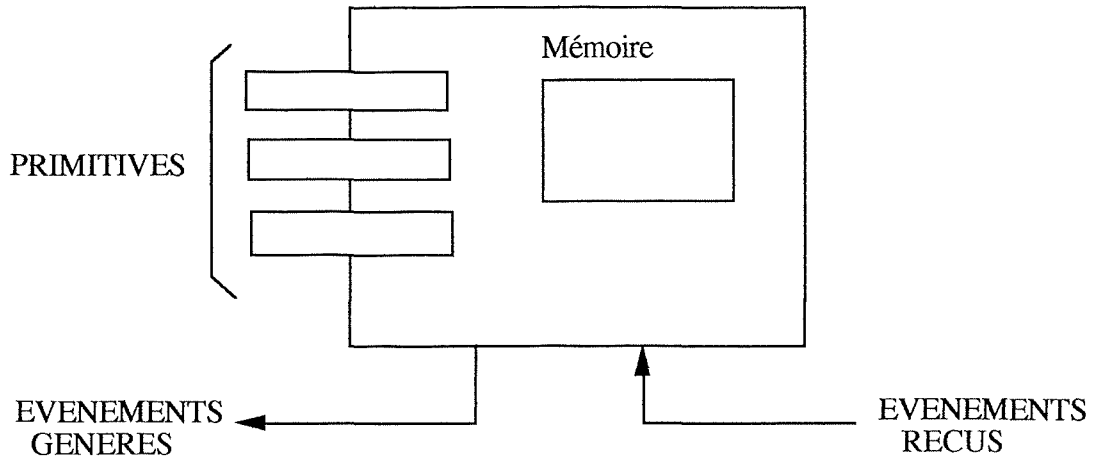
V.1. GENERALITES

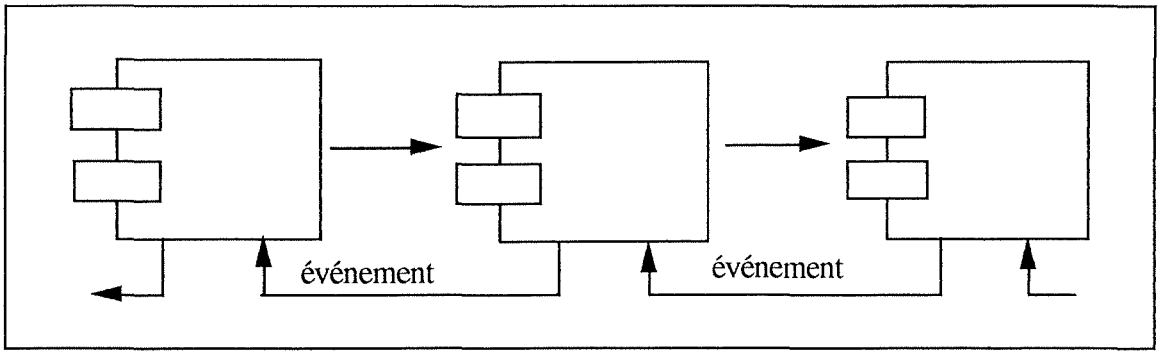
Au sens classique, les niveaux supérieurs de l'architecture logique utilisent des modules d'interface pour assurer la partie interactive du logiciel.

Aujourd'hui cependant, les concepteurs souhaitent renforcer la convivialité des applications informatiques. De ce fait, de plus en plus, on trouve l'interface au "top" de la hiérarchie. Il joue un rôle de coordination entre toutes les "fonctionnalités".

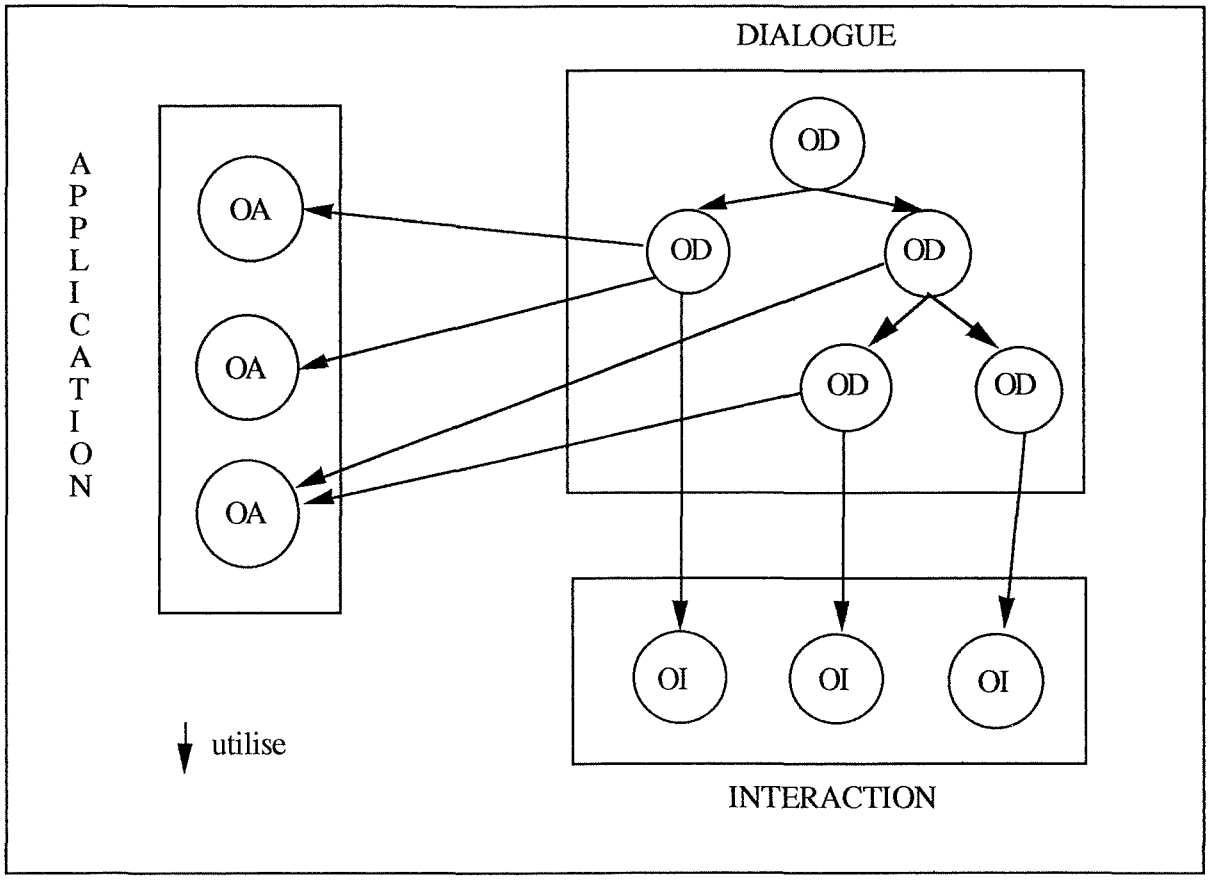
C'est l'orientation que nous adopterons et c'est pourquoi un chapitre lui est consacré.

Pour spécifier l'interface, nous choisirons une méthode vue au cours d' "interface Homme/Machine" de F.Bodart (année académique 90/91). Cette méthode consiste à mettre en évidence un ensemble d'objets possédant un comportement dicté par des demandes qui lui sont adressées et/ou par des événements extérieurs. Chaque objet possède un ensemble de services qu'il peut offrir et est donc distinct de tout autre. Tous les objets suivent le même modèle :





On distingue les Objets de Dialogue (OD) et les Objets Interactifs (OI). Un OD est entité abstraite qui est susceptible d'informer d'autres objets d'événements significatifs pour eux tandis qu'un OI est un bouton poussoir, une fenêtre... . Ces deux types d'objets sont reliés entre eux selon une architecture propre à une application mais qui sera toujours du type :



OA = Objet de l'Application (PATIENT,...)

L'interface, lui-même, est donc vu comme une architecture de modules décomposée globalement en trois niveaux :

- un niveau de coordinateur dont le but est d'ordonnancer les modules de l'application et ceux de l'interface
- un niveau d'OD
- un niveau d'OI

V.2. NOTRE ARCHITECTURE

Nous avons réuni aux figures 5.1,5.2,5.3 et 5.4 l'architecture des divers OD et OI qui constituent notre interface

Nous avons aussi fixé certains choix quant aux types d'OI utilisés, leur agencement et enchaînement.

1) Mise en évidence des différentes fonctionnalités réalisables (dès maintenant ou par le futur) à l'aide d'une barre de menu principal et de menus déroulants.

2) Saisie d'attributs au clavier, que ce soit pour un patient, un potentiel évoqué ou un spectre, à l'aide de boîte dialogue.

3) Les boîtes de dialogue pour la saisie sont agencées de sorte que dans la réalisation d'une fonctionnalité, l'utilisateur sache toujours où il en est. Par exemple, l'enregistrement d'un potentiel évoqué requiert après son acquisition, la saisie des coordonnées du patient et du commentaire sur le spectre recueilli. La fenêtre de dialogue PATIENT viendra d'abord occulter une partie du spectre et sera ensuite elle-même occultée par la fenêtre de dialogue POTENTIEL EVOQUE.

4) La saisie à l'intérieur d'une boîte de dialogue est telle que l'utilisateur peut à tout moment modifier n'importe quel champ.

5) La validation des champs d'une boîte de dialogue se fait en sélectionnant le champ OK qui se trouve toujours situé au même endroit dans une boîte quelconque (ainsi d'ailleurs que le champ ANNULER). L'annulation de toute la fonctionnalité se fait en sélectionnant le champ ANNULER.
La possibilité de retourner à l'affichage et état précédant est offerte par la touche ESCAPE du clavier.

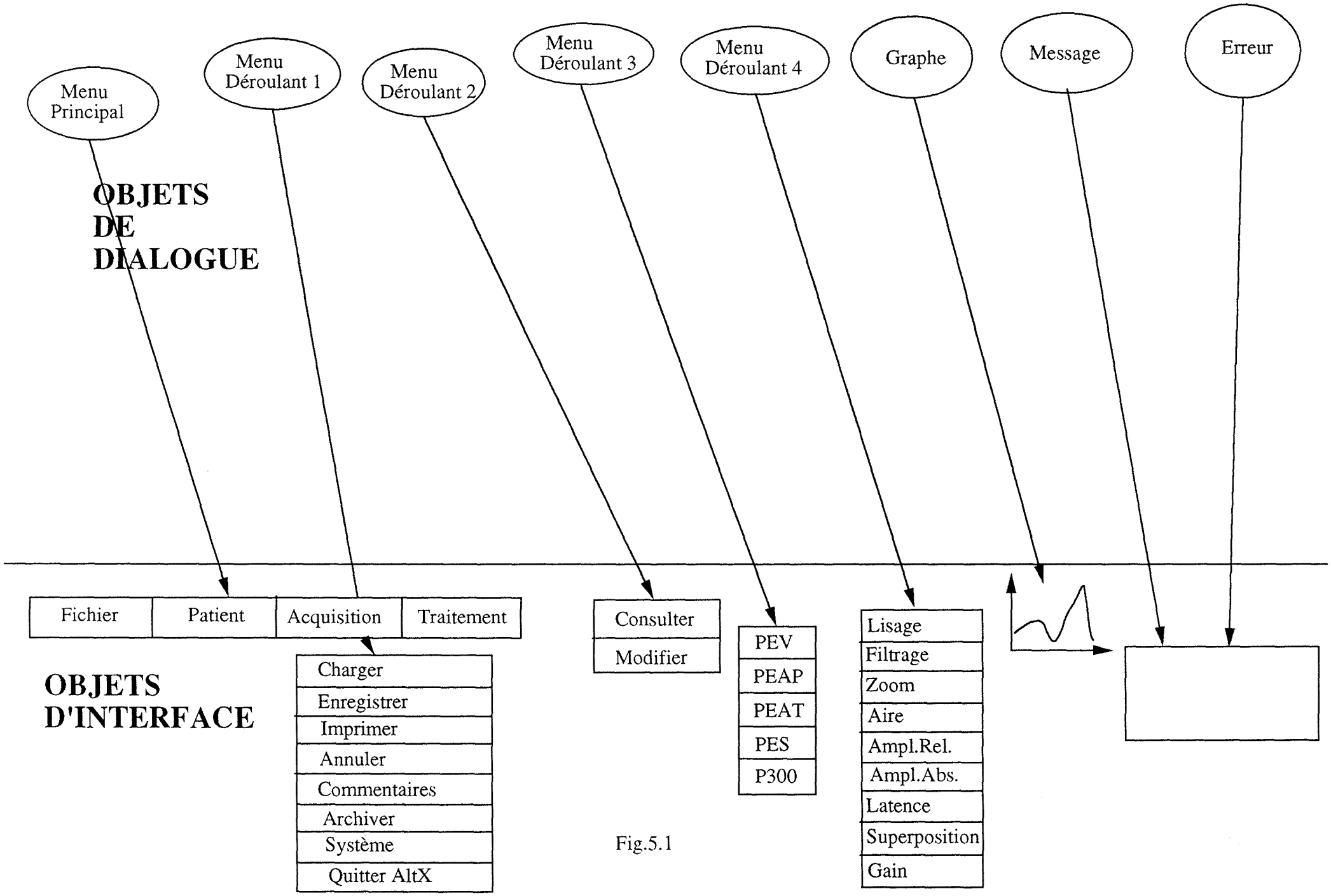

Menu Principal
Menu Déroulant 1
Menu Déroulant 2
Menu Déroulant 3
Menu Déroulant 4
Graphe
Message
Erreur
OBJETS DE DIALOGUE
Fichier
Patient
Acquisition
Traitement
Consulter
Modifier
PEV
PEAP
PEAT
PES
P300
Lisage
Filtrage
Zoom
Aire
Ampl.Rel.
Ampl.Abs.
Latence
Superposition
Gain
OBJETS D'INTERFACE
Charger
Enregistrer
Imprimer
Annuler
Commentaires
Archiver
Système
Quitter AltX

Fig.5.1

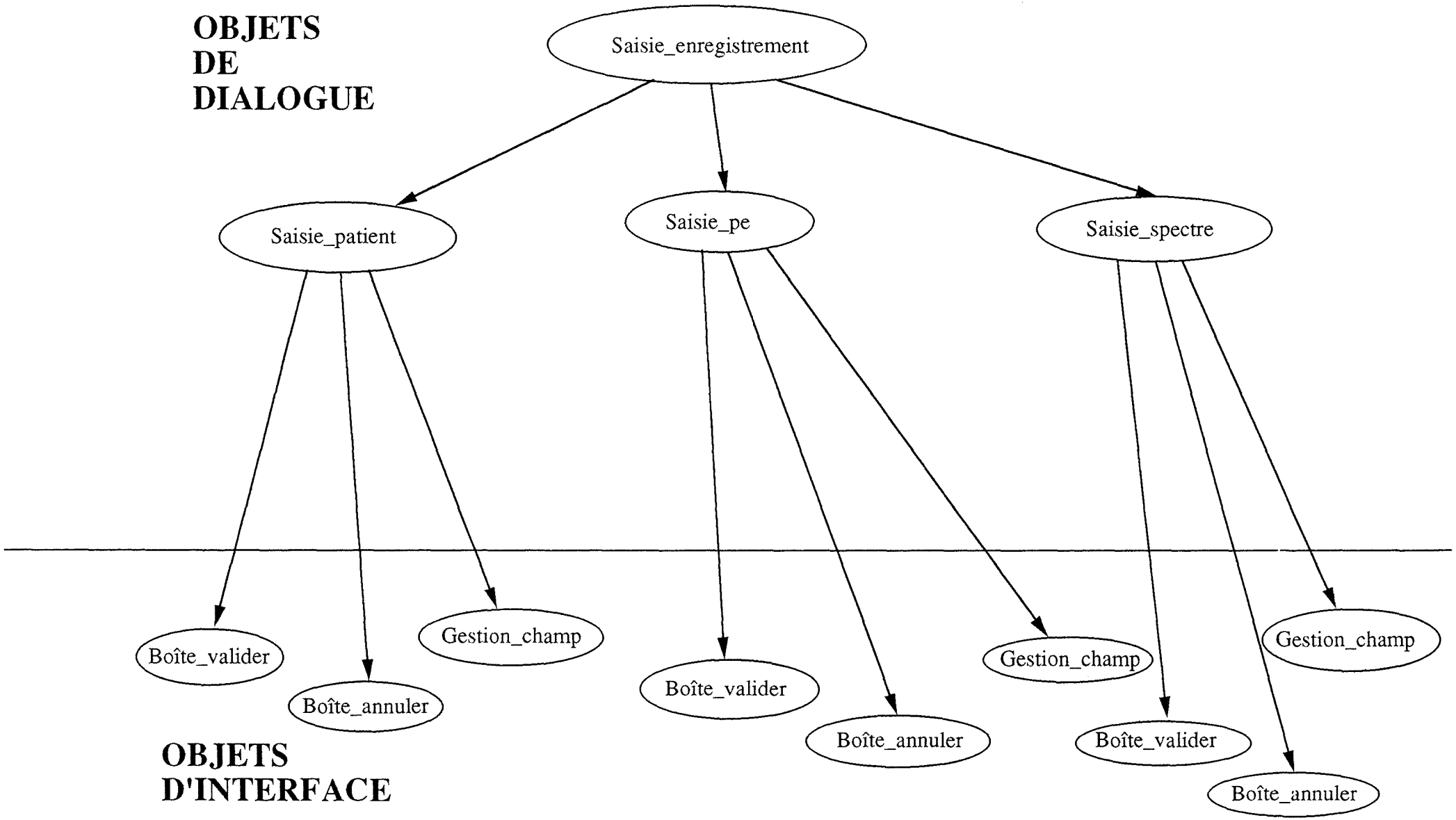

OBJETS
DE
DIALOGUE
Saisie_enregistrement
Saisie_patient
Saisie_pe
Saisie_spectre
Boîte_valider
Boîte_annuler
Gestion_champ
Boîte_valider
Boîte_annuler
Gestion_champ
Boîte_valider
Boîte_annuler
Gestion_champ
OBJETS
D'INTERFACE

Fig.5.2

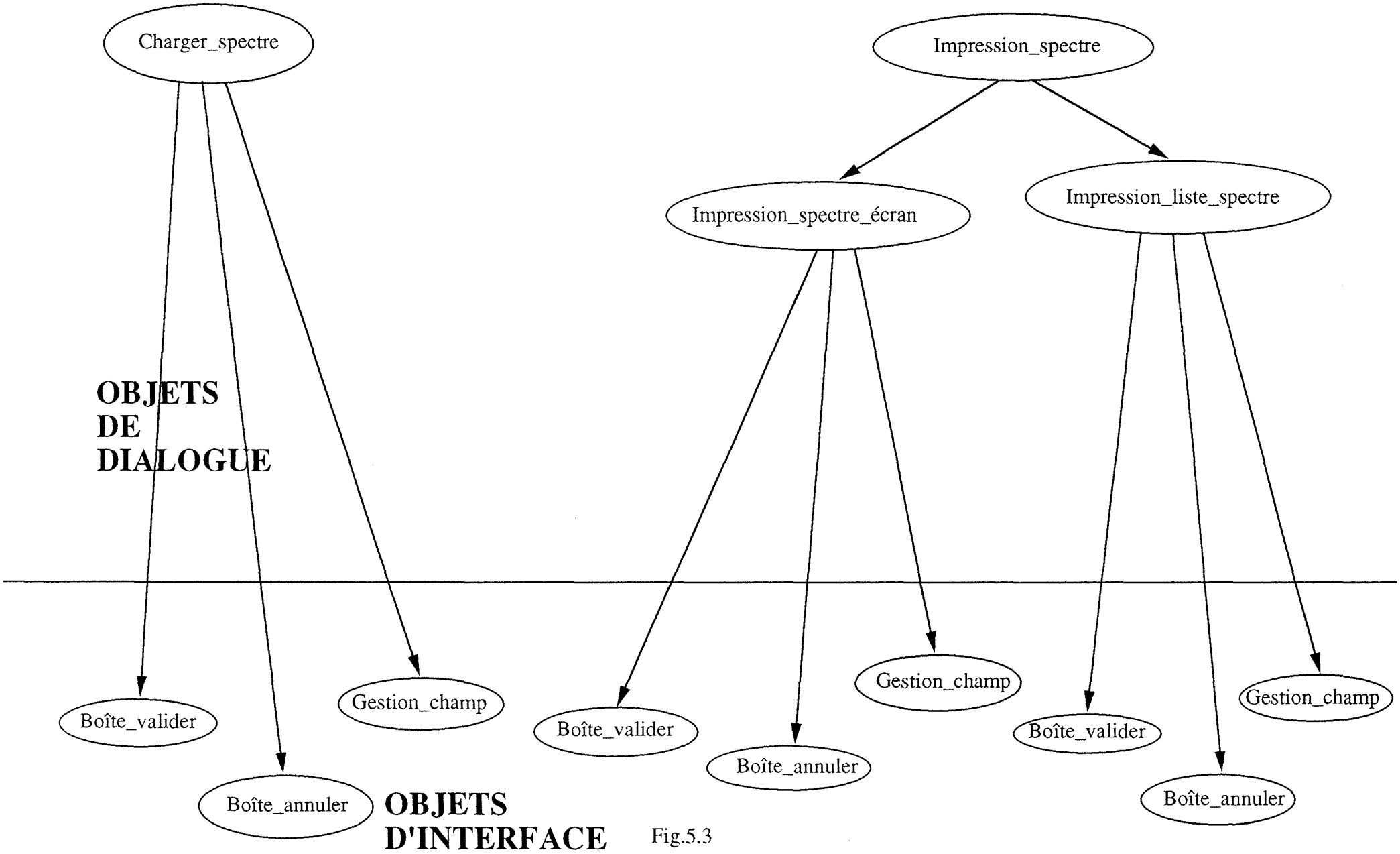

Charger_spectre
Impression_spectre
Impression_spectre_écran
Impression_liste_spectre
OBJETS
DE
DIALOGUE
Boîte_valider
Boîte_annuler
Gestion_champ
Boîte_valider
Boîte_annuler
Gestion_champ
Boîte_valider
Boîte_annuler
Gestion_champ
OBJETS
D'INTERFACE

Fig.5.3

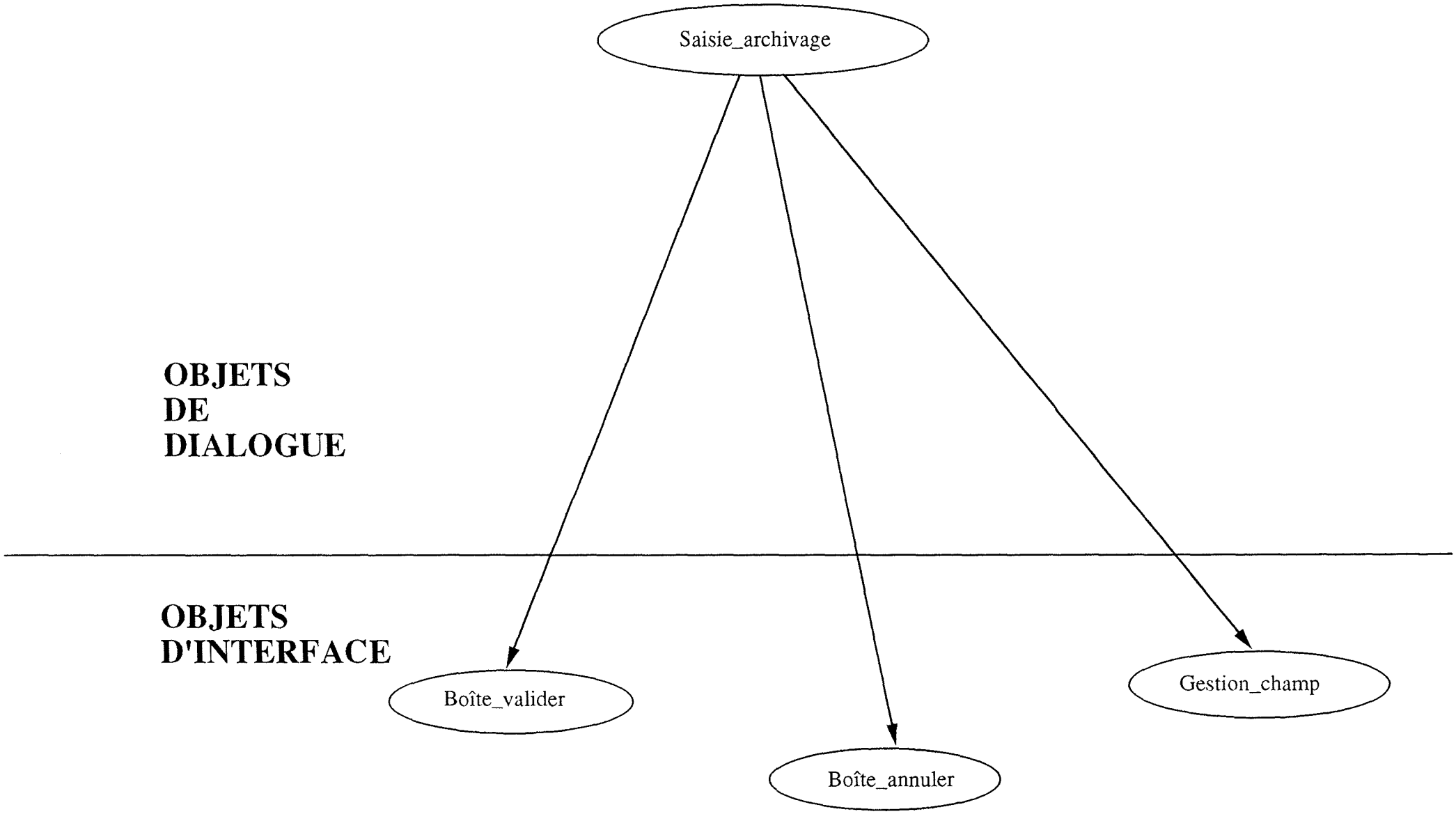

Saisie_archivage
OBJETS
DE
DIALOGUE
OBJETS
D'INTERFACE
Boîte_valider
Boîte_annuler
Gestion_champ

Fig.5.4

6) Dès qu'un spectre aura été affiché, il y en aura toujours un à l'écran : soit le dernier qui a été acquis , soit le dernier que l'on a demandé d'afficher et sur lequel l'utilisateur pourra toujours travailler (calcul de hauteur relative ou absolue de pics,...)

7) Afin d'éviter de réafficher sans cesse tout l'écran, on a choisi l'optique de sauver la zone écran qui serait recouverte par une boîte de dialogue.

Interressons-nous maintenant à la spécification des objets les plus importants de cet interface.

Pour bien comprendre le formalisme utilisé, il est bon de définir quelques notions:

1) Le comportement d'un objet est l'ensemble des actions effectuées lors d'un changement d'état .

2) Une primitive est une opération que l'on peut effectuer sur un objet (ou un service que l'on peut demander à l 'objet).

3) Un événement correspond à un changement d'état d'un objet de dialogue significatif pour un autre objet de dialogue.

I.4.3 Spécifications

Objet Menu_principal (MP)

Description de l'interface

Primitives :

.Création_MP

.Réinit-MP

Evénements

.ev_select_M1 L'utilisateur sélectionne le menu 1

.ev_select_M2 L'utilisateur sélectionne le menu 2

.ev_select_M3.L'utilisateur sélectionne le menu 3

.ev_select_M4.L'utilisateur sélectionne le menu 4

Classe d'OI utilisés

Barre de menus

Description du comportement

Déclaration des variables

N : nombre de colonnes : INT

COL 1..N : Array of string

Réaction aux primitives

IF création_MP
THEN Le MP est renvoyé à l'écran
Initialisation de N
Initlabel MP (col. 1..N)
Affichage_barre_menu

IF Réinit_MP
THEN le MP est en situation initiale

Réaction aux événements
IF(ev_ESC_menu_déroulant OR ev_annuler_fenêtre OR
ev_fin_impression OR ev_fin_archivage)
THEN réinit_MP

Objet Menu n

Description de l'interface
Primitives
.pr_création_MD(n) crée le menu déroulant n
.pr_réinit_MD(n)
Evénements
.ev_select_ligne 1
.ev_select_ligne m
Classe d'OI utilisées : menu déroulant
Classe d'OA utilisées : spectre

Description du comportement
Déclaration des variables
M : Nombre de lignes dans Menu : INT
LIGNE 1..M array of STRING
Réaction aux primitives
IF pr_création_MD(n)
THEN Initialisation de m
Initlabel MD (ligne 1..M)
Affiche Menu_déroulant
ECRAN2:=SCREEN

IF pr_réinit_MD(n)
THEN Affichage (ECRAN2)

Réaction aux événements
IF ev_select_n
THEN IF not test_spectre_save
THEN réunit MP
création MD(n)
IF ev_ESC_graph THEN Reinit MD(n)
IF ev_ESC_Ipression THEN Reinit MD(n)
IF ev_ESC_Archivage THEN Reinit MD(n)

IF ev_sélect_charger THEN IF NOT test_spectre_save
THEN création_mess_err
IF ev_ok_mess_err
THEN save_spectre
IF ev_non_mess_err
THEN pr_réinit MP
pr_création_MD(n)

Objet Graphe

Description de l'interface
Primirives
Créationgraph
Réinitgraph
Installgraph
Installstick
Activation_stick
Position_stick
Déplacement_gauche_stick
Déplacement_droite_stick
Affichage_info; m_o_j info
Mise à jour_échelle
Classes d'OI utilisées : graphique

Description du comportement

Déclaration des variables

ABSCISSE 1.. MAXA : graduation de l'axe des abscisses
ORDONNEE 1..MAXO :graduation de l'axe des ordonnées
Aire, POS1,POS2,POSITION1, POSITION2 : INT

Réaction aux primitives

```
IF créationgraph THEN init_Axes
                      Affiche_Axes
                      génère ev_Select_Main_Menu

IF réinitgraph THEN affiche le dernier graphe à l'écran avant
                    transformation ou sélection de menus.

IF installgraph THEN dessine le graphe à l'écran
IF instalstick THEN dessine les sticks à l'origine du graphe

IF activation_stick THEN désactive le stick activé
                         et active le stick désactivé

IF Positionstick (STICK)
THEN IF STICK est Stick1 THEN return POSITION1
     IF STICK est Stick2 THEN return POSITION2

IF Déplacement _gauche_stick
THEN IF Stick1_activé
   THEN IF position1>O
        THEN effacer_stick
             installe_stick (POSITION1_1); position1 : = position1-1
   ELSE rien
ELSE IF (position 1<position 2)
THEN Effacer_stick
     Instal_stick (POSITION2-1);position2 := position 2-1
ELSE rien

IF Déplacement_droite_stick
IF Stick1_activé THEN IF (POSITION1<MAXA
                          AND POSITION1<POSITION2)
THEN Effacer_Stick
```

```
            Install_stick(POSITION +1)
            POSITION1:=POSITION1+1
        ELSE rien
        ELSE IF (POSITION2<MAXA)
        THEN Effacer_stick; Instal_stick(POSITION2+1)
        POSITION2:=POSITION2+2
        ELSE rien

        IF Initstick THEN POSITION1:=O; POSITION2:=O
        IF Affichage_info
        THEN Modification_aire (POSITION1,POSITION2,AIRE)
            Afficher aire(LABEL_AIRE,AIRE)

        IF Modification_hauteur1
        THEN Afficher_hauteur1(LABEL_HAUTEUR1,HAUTEUR1)
            Calcul_hauteur2
            afficher_hauteur2(LABEL-HAUTEUR2,HAUTEUR2)

        IF Mise_à_jour_échelle THEN Nouvelles_échelles

        IF Mise_à_jour_info THEN Calcul_Aire(POS1, POS2,AIRE)
                                 Afficher (Aire)
                                 Calculer_hauteur(POS1,POS2)
                                 Afficher hauteur(Hauteur1,Hauteur2)

    Réaction aux événements
        IF ev_select_Main Menu THEN creationgraph
        IF ev_annuler_acquisition THEN Réinitgraph
        IF ev_'→'_ THEN Déplacement_droite_stick
        IF ev_'←'THEN Déplacement_gauche_stick
        IF ev Activation_stick THEN Activation_stick
        IF ev_ouvrir_nr_pe_OK THEN Install_graphe(nr_pe)
```

Objet Info

Description de l'interface

Primitives

Création_info
Modification_aire (POSITION1,POSITION2)
Modification_hauteur(POSITION1,POSITION2)
Modification_latence
Evénements
Afficher_aire, Affiche_hauteur, affiche_latence

Description du comportement
Réaction aux primitives
IF Création info2 THEN Afficher_label_Aire
Afficher_label_Hauteurs
Afficher_label_latence
Init_champ_latence
Init_champ_Aire
Init_champ_hauteur

IF Modification_aire (POS1,POS2)
THEN Calcul_aire (POS1,POS2)

IF Modification_hauteur (POS1,POS2)
THEN calcul_hauteur (POS1,POS2)

IF Modification_latence THEN Calcul_latence
IF Afficher_aire (AIRE)
THEN Mise à l'écran de la valeur de Aire

IF Affiche_latence THEN mise à l'écran de la latence

IF Affiche_hauteur (HAUTEUR1, HAUTEUR2)
THEN Mise à l'écran des nouvelles hauteurs.

Objet : enregistgrement_pe

Description de l'interface
Primitives
Création_enregistrement
Réinit_enregistrement

Obt_patient
Obt_spectre
Classes d'OD utilisées
Saisie_patient, Saisie_spectre

Classes d'OI utilisées
boîte de dialogues + bouton de confirmation
Classes d'OA utilisées
patient (via Patient_existant et demande_info_pat)

Evénements
ev_ok_enregistrement
ev_ESC_enregistrement
ev_annuler_enregistrement
Description du comportement
Déclaration des variables
PATIENT, INFO_PE

Réactions aux primitives
IF Création_enregistrement THEN Création_saisie_patient
Création_saisie_spectre
ECRAN4 :=SCREEN

IF Réinit_enregistrement
THEN Affichage (ECRAN4)

IF Obt_patient THEN Réinit_saisie_pat
Remplissage_champ(PATIENT)
Générer ev_ok_patient

IF Obt-pe
THEN Réinit_saisie_spectre
Remplissage_champ (INFO-PE)
Générer ev_ok_pe

Réaction aux événements
IF ev_ESC THEN Affichage (ECRAN1)
IF ev_select_enregistrement THEN Création_enregistrement

IF ev_spectre_non_encore_enregistre THEN Création_enregistrement

Objet Saisie_patient

Description de l'interface

Primitives

.Création_saisie_pat

.Réinit_saisie_pat

.Affichage_pat_existant

.Saisie_identifiant_pat

Saisie_pat_reste

Evénements

.ev_annuler_saisie_pat

.ev_ok_saisie_pat

.ev_identifiant_saisie

Classes d'OI utilisées Boîte à champ d'entrée

Classses d'OA utilisées Patient

Description du comportement

Déclaration des variables

LABELNOM, LABELPRENOM, LABEL...

NOM, PRENOM,.....

Réaction aux primitives

IF Création_saisie_pat

THEN Initlabel(LABEL_NOM,...)

Afficher_label

Initchamp

Initcurseur

IF Réinit_saisie_pat THEN Initchamp

Initcursuer

IF Affichage_pat_existant (Identifiant_pat)

THEN IF Patient_existant (Identifiant_pat)=true

THEN Remplir_champs (attributs_pat)

IF Saisie_identifiant_pat
THEN Saisie_nom
Saisie_prénom
Saisie_date_naiss

IF Saisie_pat-reste
THEN Saisie_autre_attributs

Réaction aux événements
IF ev_ESC_Saisie_pe THEN Reinit_Saisie_pat
IF ev_ESC THEN Reinit_Saisie_pat

Objet Saisie_pe

Description de l'interface
Primitives
.Création_saisie_pe
.Réinit_saisie_pe
.Affichage_nr_pe_calculé
.Saisie_commentaires

Evénements
.ev_annuler_saisie_pe
.ev_ok_saisie_pe
Classes d'OI utilisées Boîte de labels et champs
Classes d'OA utilisées spectre

Description du comportement
Déclaration des variables
COMMENTAIRE : Texte
NR_SPECTRE : INT
Réactions aux primitives
IF Création_saisie_pe THEN Initlabel
Afficher_label
Init-champ

IF Réinit_saisie_pe THEN Remettre le curseur au début du commentaire qui est remis à blanc

IF Affichage_nr_pe_calculé
THEN Affichage (**nr_pe**)
IF saisie_commentaires
THEN tant que pas (ev_annuler_saisie_pe ou ev_ok_saisie_pe)
saisir_ligne_champ

Réaction aux événements
IF ev_ok_saisie_pat THEN Création_saisie_pe

Objet Saisie_date_archive

Description de l'interface
Primitives
.Création_datre_arch
.Affichage_dernière_date
.Saisie_date
.Contrôle_format_date
.Réinit_date_arch
Evénements
.ev_ok_arch
.ev_annuler_arch

Classe d'OI utilisées label et champ

Description du comportement
Déclaration des variables
date : INT
LABELDATE : STRING
Réaction aux primitives
IF Création_date_arch_THEN Initlabel
Afficher_label
Initchamp

IF Affichage_dernière_date THEN date=Date_dernier_archivage

afficher(date)

IF Saisie_date THEN Entrer (date)
Afficher(date)

IF Réinit_date_arch THEN Initchamp2

Réaction aux événements
IF ev_ESC THEN retour ECRAN2
IF ev_F1 THEN return ECRAN2

Objet Affichage_message_archivage

Description de l'interface
Primites
.Création_message_confirmation
Evénements
.ev_annuler_arch
Classes d'OI utilisées boîte de dialogue
Description du comportement
Déclaration des variables
MESSAGE_CONFIRM String
Réaction aux primitives
IF Création_message_confirmation
THEN Création_message_confirmation
Intlabel(MESSAGECONFIRM)
Affichage (MESSAGECONFIRM)
Réaction aux événements
IF ev_ESC THEN retour graphe
IF ev_ok_date_arch THEN Création_message_confirmation

Objet Archivage_en_cours

Description de l'interface
Primitives
.Création_arch_cours

Evénements

.ev_fin_arch_cours

Classes d'OI utilisée boîte de dialogue

Description du comportement

Déclaration des variables

MESSAGE_EN_COURS : STRING

Réaction aux primitives

IF création_arch_cours

THEN Initlabel (MESSAGE_EN_COURS)

Affichage (MESSAGE_EN_COURS)

Réaction aux événements

IF ev_ESC THEN retour graphe

IF ev_ok_confirm_arch THEN Création_arch_en_cours

Objet Archivage_fin

Description de l'interface

Primitive

.Création_archivage_fin

Evénements

.Evénements ev_archivage_fin

Classe d'OI utilisées boîte de dialogue

Description du comportement

Déclaration des variables

MESSAGE_FIN_ARCH : string

Réaction aux primitives

IF Création_archivage_fin THEN Inittlabel

Affichage

Réaction aux événements

IF ev_fin_arch_cours THEN création_archivage_fin

Objet Affichage_erreurs

Description de l'interface

Primitives

.Création_mess_err

Evénements

.ev_ok_mess_err

.ev_non_mess_err

Classes d'OI utilisée

Boîte de dialogue

Description du comportement

Déclaration des variables

Mess : string

Réaction aux primitives

IF Création_mess_err THEN Init(Mess)
Affichage(Mess)

Objet Menu_imprimer

Description de l'interface

Primitives

.Création_impression

.Réinit_impression

Evénements

.ev_Select_imprimer

Classes d'OI utilisées

Boîte de dialogue et champs

Description du comportement

Déclaration des variables

LIGNE1 : String

LIGNE2 :String

Réaction aux primitives

IF Création_impression THEN Init(Ligne1,Ligne2)
Affichage (Ligne1,Ligne2)
ECRAN6=rien

IF Réinit_impression THEN Affichage (ECRAN6)

Réaction aux événements

IF ev_select_imprimer THEN création_impression

IF ev_ESC THEN retour ECRAN2

Objet Confirm_impression

Description de l'environnement

Primitives

.Création

Evénements

.ev_impression

.ev_ESC (on annule impression)

.Description du comportement

Déclaration des variables

MESSAGE : string

Réaction aux primitives

IF Création

THEN Init(MESSAGE :"confirm_impression")

Affichage (MESSAGE)

Réaction aux événements

IF ev_select_impression_écranTHEN Création

IF ev_select_list_spectre_ok THEN Création

Objet message impression en cours

Description de l'interface

Primitives

.Création : Init(MESSAGE) Affichage

Evénements

.ev_fin_impression

Description du comportement

Réaction aux primitives

IF Création THEN Initlabel

Affichage

Réactions aux événements

IF ev_impression THEN création

IF ev_sélect_liste_spectre_ok THEN création

A l'annexe 7 on trouvera une visualisation des divers écrans constituant l'interface.

CHAPITRE VI

Implémentation physique

VI.IMPLANTATION PHYSIQUE

Des étapes précédentes, nous héritons d'une architecture de projet indépendante de tout choix physique. Nous arrivons maintenant au stade ultime du développement : la conception physique.

Un système d'acquisition de P.E. doit non seulement générer des stimuli, mais également réceptionner l'activité cérébrale ainsi déclenchée pour la traiter mathématiquement et graphiquement et mémoriser les résultats obtenus, pour des traitements ultérieurs.
La dualité "temps réel/temps différé" mise en évidence lors de l'étape de conception nous amène à nouveau à séparer l'implantation de l'unité d'acquisition et l'implantation du système d'information. Cependant , ne perdons pas de vue que les deux applications communiquent entre elles par l'échange d'informations; l'une recueille le P.E., l'autre l'exploite.
Dans ce chapitre, nous insisterons particulièrement sur l'implantation des unités d'acquisition (U.A) et de coordination (interface), et dirons quelques mots sur l'implantation du système d'information (S.I). Les traitements différés sont laissés au stade des spécifications.
Rappelons que la démarche de conception physique conserve en tout point la découpe en phases et fonctions . Celles-ci deviennent en fait des "procédures". Nous n'en donnerons qu'une implémentation globale. Le formalisme utilisé sera le langage Pascal, dont les qualités sont appropriées à une présentation claire et structurée[1].

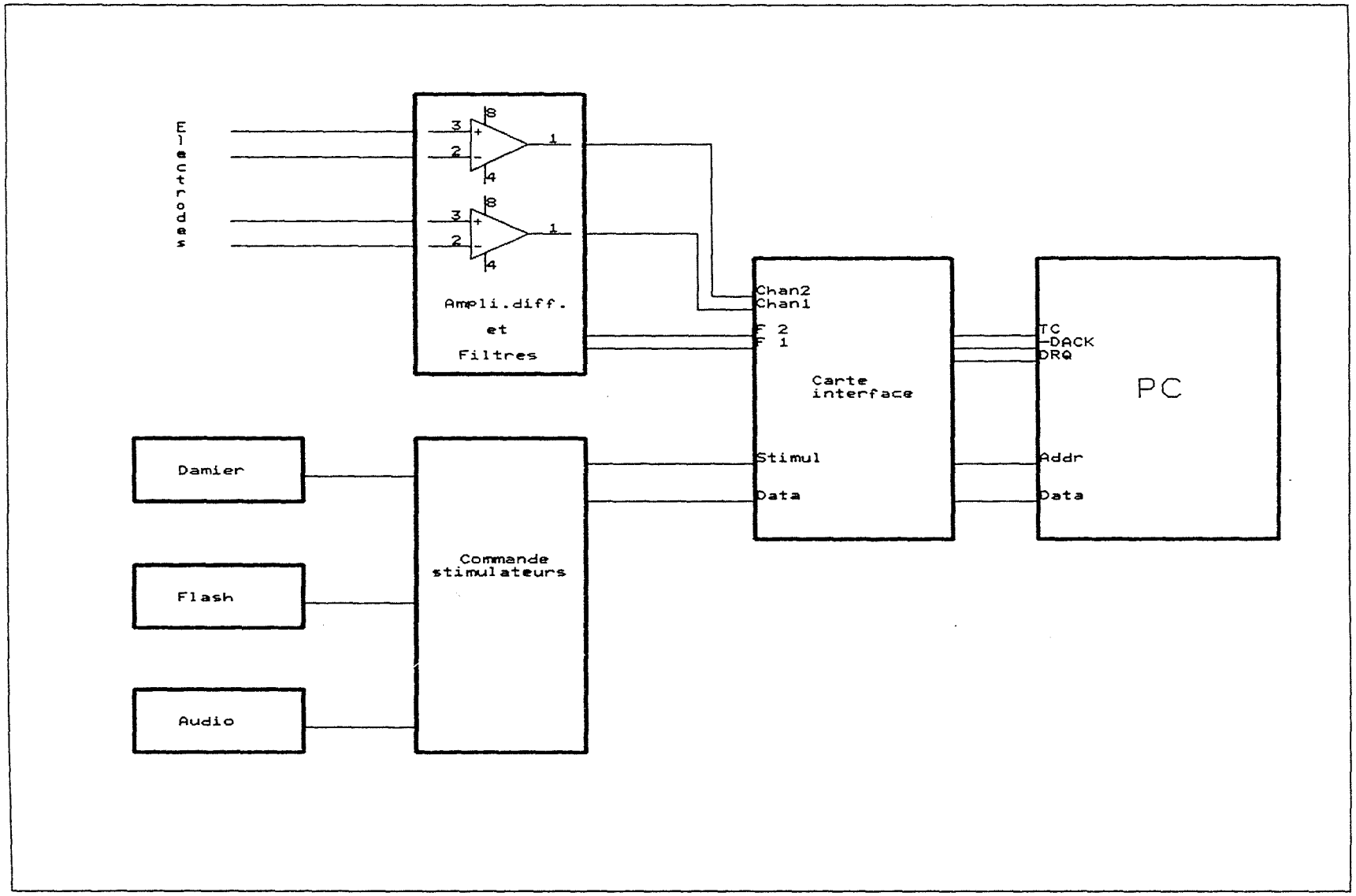
VI.1. IMPLANTATION DE L'UNITE D'ACQUISITION.

VI.1.1. Les ressources (Schéma bloc n°1)

Si les procédures venant des niveaux fonctionnels de notre architecture (niveaux supérieurs) doivent être indépendantes des choix physiques, il n'en est pas de même pour les fonctions de bas niveau qu'elles utilisent. La présentation des ressources physiques mises à notre disposition est donc une étape indispensable.
Rappelons les différentes ressources jugées nécessaires lors de l'analyse du problème:

- les unités de stimulations
- les canaux de prise de données analogiques
- la chaine d'amplification et de filtrage
- le convertisseur analogique-digital
- l'unité de transfert vers la mémoire
- l'unité de synchronisation

[1] Le langage Pascal est ici vu comme un langage formel permettant une certaine généralité dans l'explication des différentes procédures. Il n'en est pas pour autant le langage d'implémentation.



SCHEMA BLOC 1

- l'unité arithmétique et logique
- l'unité d'affichage
- l'unité de stockage

La ressource principale: le PC

Pour commander le processus d'acquisition , confiance est faite à un Personal Computer de la série AT. Il renferme plusieurs ressources utiles:

1.) L'unité arithmétique et logique
La mémoire de travail
L'unité de stockage

Le microprocesseur (80286 ou 80386) peut effectuer les opérations arithmétiques et logiques voulues et organiser la mémoire de travail tout en commandant l'ensemble des ressources périphériques (imprimante, disque dur, écran vidéo, carte prototype pour citer ce dont nous aurons besoin).

2.) L'unité de transfert vers la mémoire

Lorsqu'on souhaite transférer des données en mémoire , deux techniques peuvent être employées. La première sollicite le processeur; c'est le transfert normal. L'autre, est appelée "transfert en accès direct à la mémoire" ou **DMA**.
Le DMA, est une technique qui permet de transférer des données entre la mémoire et un organe périphérique, sans passer par le microprocesseur. On peut ainsi décharger celui-ci et accélérer l'exécution du programme..
Dans les PC on réalise l'accès direct à la mémoire par le contrôleur 8237A de chez Intel(le **DMAC**). Il fournit quatres canaux de transfert DMA , capables de transférer des blocs de 64K. L'utilisation de contrôleurs en cascade permet d'augmenter le nombre de canaux possibles.
Le DMAC sera donc utilisé dans notre application pour réaliser un gain de temps considérable. Un transfert DMA est 6 fois plus rapide qu'un transfert effectué par le processeur.
De plus amples informations concernant ce périphérique de Intel sont rassemblées en annexe[2] .

3.) L'unité d'affichage

[2] annexe 4: le DMA

Le choix d'un PC restreint les possibilités en matière de carte vidéo. Par ses capacités d'affichage graphique surclassant largement d'autres standards et sachant qu'elle fait presque partie de l'équipement "standard" indispensable d'un PC, la carte **EGA/VGA** sera notre ressource d'affichage.

Les ressources analogiques

Les unités de stimulations
Les canaux de prise de données analogiques
La chaîne d'amplification et de filtrage

Ces ressources sont réalisées par des modules purement hardware que nous ne détaillerons pas ici. Seul leur comportement est intéressant à connaître.
En toute généralité:
- la génération d'un stimulus est provoqué par l'activation d'un signal électrique à l'entrée d'un module de commande des stimulateurs.
- une fois les électrodes appliquées sur le patient, les sorties des canaux de prise de données génèrent continuellement des signaux analogiques amplifiés et filtrés.
- deux signaux horlogiques de fréquences réglables, envoyés vers ces canaux réalisent la bande passante souhaitée.

La carte interface

Elle permet la communication entre les ressources analogiques et le PC. Elle consiste essentiellement en:

1.) L'unité de conversion digitale

Le convertisseur A/D (**ADC**) est un AD7878, pouvant digitaliser des signaux analogiques bipolaires entre - 3 et +3 Volts avec une précision de 12 bits.
La digitalisation des données se fait en trois phases: l'ADC, recevant un signal de conversion, masque toute activité qui pourrait le perturber. Il fige alors une tension analogique, exécute sa routine de digitalisation, range la donnée dans une pile et enfin rend à nouveau possible toute activité extérieure. Tout est alors prêt pour le transport de la donnée vers la mémoire centrale.

2.) L'unité de synchronisation

Pour supporter les fonctions de synchronisation et de génération de signaux horlogiques, deux cadenceurs/compteurs programmables, les **Timers 8254** de la famille Intel, sont utilisés.
Le 8254 est un cadenceur/compteur d'intervalles capable de générer à la demande des signaux continus de durée déterminée ou des signaux oscillant de période déterminée. Cela permet notamment la génération de délais précis sous contrôle software, sans devoir implémenter des boucles d'attente qui monopolise le processeur. D'autres problèmes telles que la génération de rythme ou la génération de signaux horlogiques réglables sont résolus facilement par ce timer.
Une description détaillée se trouve en annexe[3].

Cette carte est commandée par une "puce" 74HCT75 (notée **Cmd**) et un bouton" Start/Stop".(74HCT08)

VI.1.2. La communication entre ces ressources

Le schéma bloc n°1 présente globalement l'interconnexion des différentes ressources.Voici quelques mots d'explication.

Les signaux analogiques en provenance des électrodes alimentent continuellement les entrées **chan1** et **chan2** de la carte. Celle-ci, respectant un "timing" convertit régulièrement ses signaux en données discrètes qu'elle achemine vers la PC via la ligne **Data**. Les échanges entre PC et carte doivent respecter un certain protocole . Dans ce but, les lignes **DREQ** (la carte demande le transfert de donnée), **DACK** (le CPU accepte une demande de transfert) et **TC** (Terminal Count: le PC signale la fin du transfert d'un groupe de données) sont utilisées. La carte peut également demander la génération d'un stimulus d'un type donné par ses sorties **Stimul** et **Data**.

VI.1.3.Commande et contrôle du processus d'acquisition de P.E

Nous devons maintenant réaliser physiquement le schéma de la dynamique des traitements élaboré au chapitre 2 (figure 2.4). Nous n'aborderons ici que l'aspect "temps réel" de l'application.
Nous disposons de ressources programmables. Le rôle du processeur et donc du code qu'il va exécuter, est d'organiser le travail en parallèle .
Distinguons trois grandes phases d'exécution :

[3] annexe 5: le timer 8254.

1°) le microprocesseur programme le comportement de chacun des périphériques et déclenche l'acquisition; c'est le rôle des procédures **Init_acquisition** et **Start_acquisition.**

2°) Le processeur lance les DMAs successifs, réalise les sommations et moyennes, commande les affichages. La carte organise la dynamique des traitements de stimulation, d'amplification et filtrage,de conversion digitale et de transfert en mémoire. N'oublions pas qu'une synchronisation entre les différentes unités est indispensable.

La procédure **Conversion_dma_transfert** est donc lancée par le processeur puis prise en charge par la carte tandis qu'il exécute les procédures **moyennage**.et **affichage_courant.**et organise la synchronisation.

3°) Lorsque le nombre de stimulations à moyenner est atteint, le processeur interrompt l'activité des périphériques et réalise les traitements de clôture.

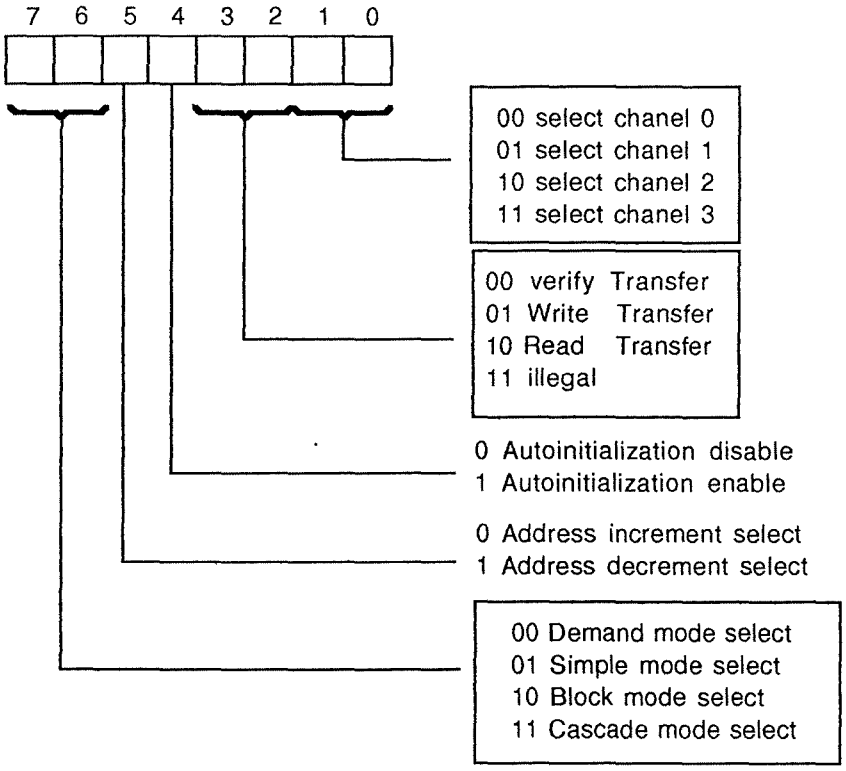
Phase 1: Init_acquisition et Start_acquisition

Cette phase requiert quelques notions générales sur la programmation des interfaces. Une interface ou un périphérique peut être défini comme étant n' importe quel circuit commandé par le CPU. Il répond donc électriquement à une ou plusieurs adresses de ports d'E/S. Nous entendons par "**port**" un emplacement spécial destiné à lire ou écrire des données et ne faisant pas partie de la mémoire du système[4].
Programmer une interface c'est envoyer à son adresse, une commande codée sur un octet. Décodant ce "**mot de commande**" l'interface agit en conséquence.

Exemple:
Pour pratiquer un transfert de données par accès direct à la mémoire, on utilise le périphérique 8237A . Avant de commencer les transferts, il est nécessaire de programmer son comportement. Il faut notamment lui demander un mode et un sens de transfert des données et choisir un des quatre canaux dont il dispose[5]. Ces trois sélections sont effectuées par l'envoi d'un mot de commande dans le registre de mode se situant à l'adresse 16H (en hexadécimal)[6].
Voici les commentaires sur le registre de mode permettant de savoir quel mot de commande envoyer.

---

[4] annexe 6: la programmation des interfaces
[5] annexe 4: le DMA
[6] en annexe 6 se trouve l'ensemble des ports d'E/S d'un PC/AT



L'envoi de la valeur 49H(en binaire 01001001) détermine un transfert simple (bits 6 et 7) en lecture (bits 2 et 3) et sur le canal DMA n°1 (bits 0 et 1).

L'instruction en code C serait : **"outport (16H,49H)".**

La phase d'initialisation de l'unité d'acquisition résulte donc dans la détermination des adresses d'E/S et dans l'exportation de mots de commande à ces adresses. Ces mots de commande vont déterminer le comportement de chacune des ressources.
Dans ce qui suit, nous donnons une ligne de conduite pour l'initialisation des ressources mais nous n'entrons pas dans les détails d'implémentation tels que valeurs d'adresse ou valeurs des mots de commande. Cependant, pour le lecteur intéressé, nous avons réuni en annexe la documentation et les références nécessaires à une programmation complète (annexe 4, annexe 5, annexe 6).

Initialisation des adresses d'E/S

Le but de cette initialisation est d'assigner les adresses et les commandes d'E/S à des variables du code .
Ainsi,dans l'exemple ci-dessus en écrivant une procédure :

procédure **Setup_mode** (var **reg_mode,transfert_simple**: integer)

```
#       begin
#               reg_mode := 16H;
#               transfert := 49H;
#       end;
```

on pourra écrire dans le programme principal,

```
#       Setup_mode( reg_mode, transfert_simple);
#       OUTPORT(dma.reg_mode,transfert_simple);
```

pour réaliser l'initialisation du mode de transfert.

En généralisant ce principe, on construit les grandes lignes de la procédure **Init_hardware** :

```
#       Type  timerchip = record
#                       cmdreg, c0reg, c1reg, c2reg : integer
#                       end;
#       Type  dma_reg = record
#                       basereg, comptreg, maskreg,
#                       modereg, pagereg, cmdreg, statusreg : integer;
#                       end;
#       procédure Init_hardware (canal : integer; var dma: dma_reg;var
#                       timer,timer2: timerchip; var adc,reset_7878, Cmd: integer);
#
#       begin
#               Setup_dma ( canal, dma );
#               Setup_timer(timer1);
#               Setup_timer(timer2);
#               Setup_Cmd ( Cmd);
#               Setup_ADC ( adc);
#               Setup_Reset_ADC  ( reset_7878);
#       end;
```

où les procédures **Setup_...** ne sont que des assignations de valeurs à des variables. La liste de toutes les valeurs des adresses d'E/S dont nous avons besoin sont consignée en annexe[7] ·

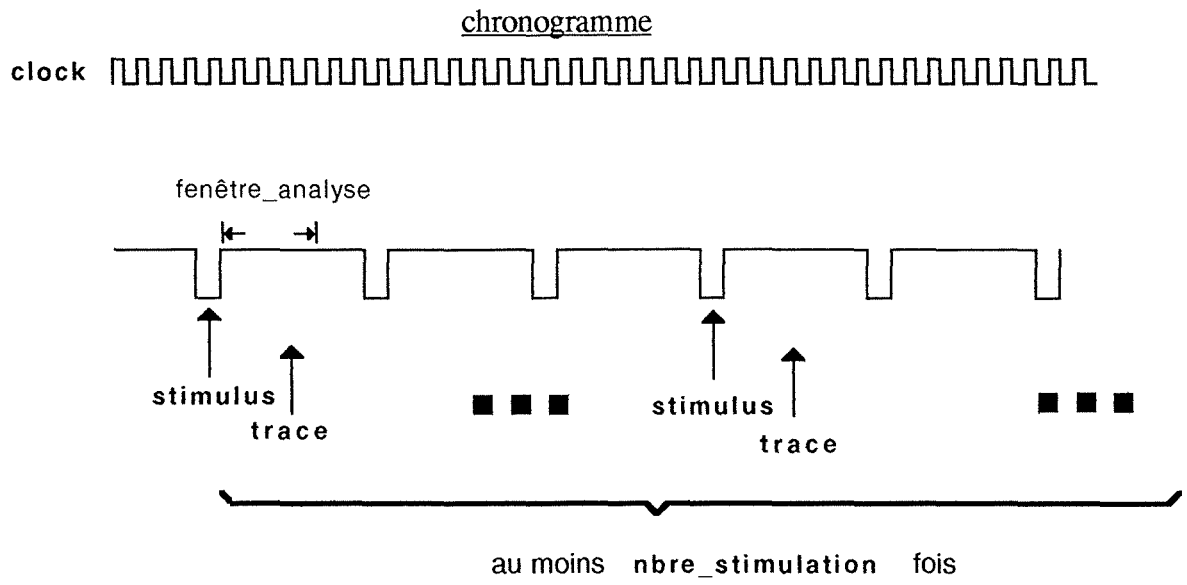
Ayant ainsi trié les adresses et les mots de commande dont nous avons besoin, la programmation de l'interface peut être étudiée.

Programmation des timers de la carte

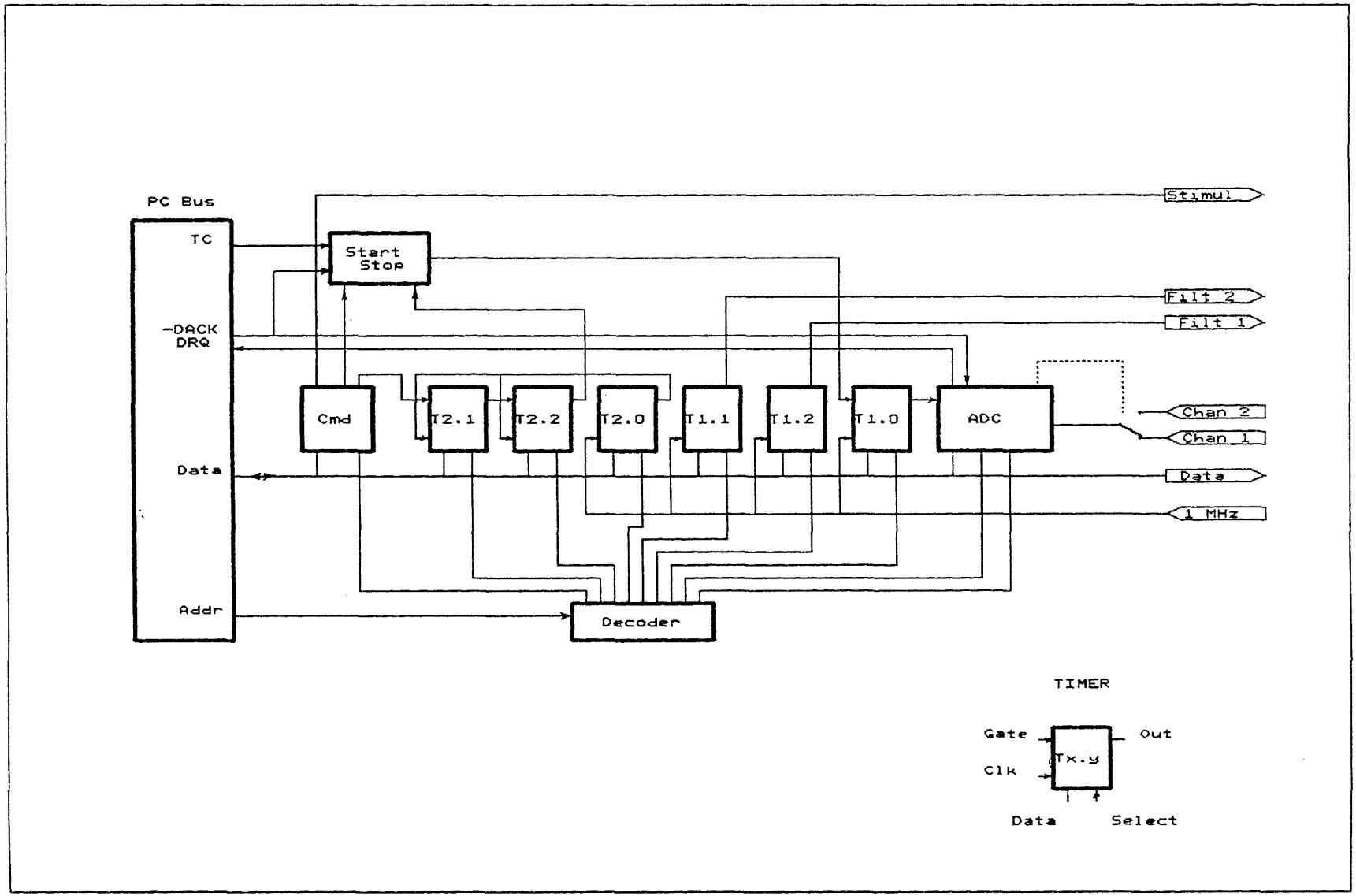
[7] annexe 6: la programmation des interfaces.

Pour initialiser les Timers, il faut savoir ce que l'on attend d'eux..
Reprenons le schéma de la dynamique des traitements, pour constater que l'application "acquisition" comporte plusieurs itérations imbriquées.
La première est une itération sur **nbre_stimulation.** La seconde , de période beaucoup plus faible, porte sur **fenêtre_analyse**; elle est rythmée par la **fréquence** d'échantillonnage et s'arrête lorsque la fin de la fenêtre est atteinte.
Etablissons le chronogramme des événements tels qu'ils doivent se dérouler (l'unité de temps est la période d'un signal horlogique).

La première itération:



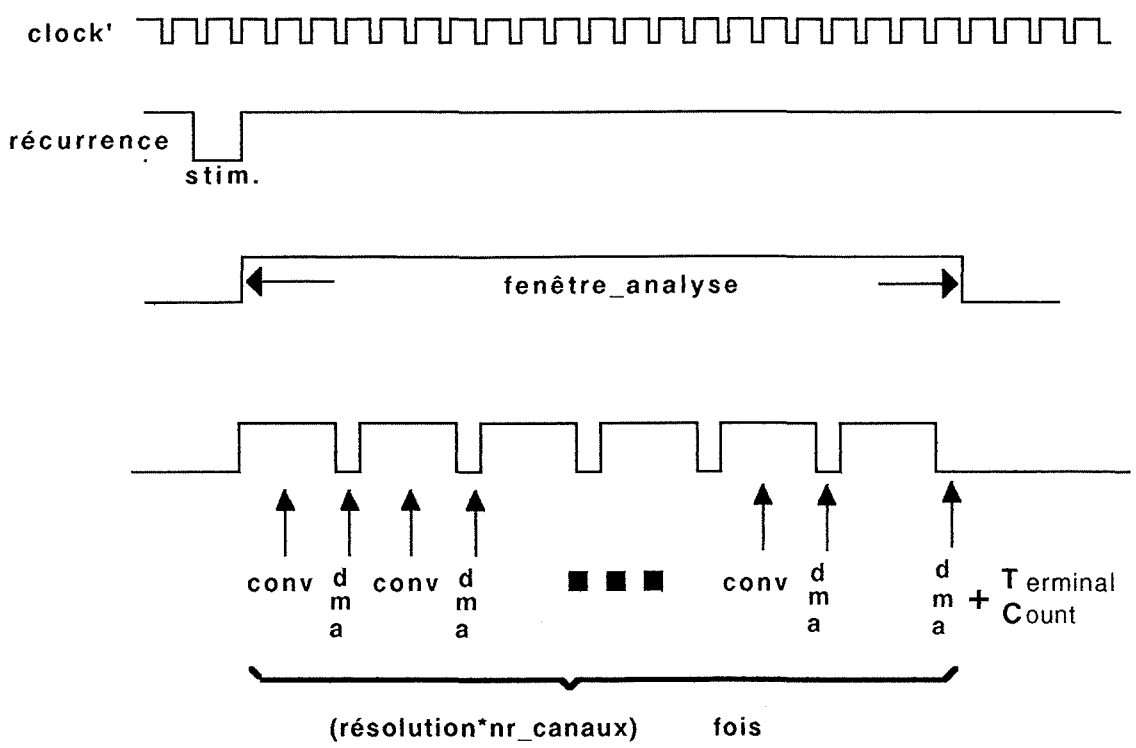
Chaque état bas représente la génération d'un stimulus. Il déclenche, à l'état haut, l'acquisition d'une trace.Ces deux opérations sont répétées périodiquement. La fréquence de répétition n'est autre que la **récurrence**. Rappelons que la fenêtre doit nécessairement être inférieure à la période entre deux stimulations.
Le contrôle sur **nbre_stimulation** sera implémenté classiquement par une boucle software.
Par contre la **récurrence** sera prise en charge par un Timer.
Sur le schéma bloc n°2, on peut voir le Timer2 (noté **T2**) entièrement voué à cette tâche:
Le premier compteur **T2.0** reçoit un signal horlogique de 1 MHz spécifique à la carte. Cela correspond à une unité temporelle de 1 microseconde. **T2.0** génère à son tour un rythme pour les compteurs **T2.1** et **T2.2**; Lorsque **T2.1** reçoit le feu vert de **Cmd**, il génère le rythme de récurrence. A la fin de chaque cycle, il envoie un signal à **T2.2** qui génère un délai correspondant à la **durée** d'une stimulation.



SCHEMA BLOC 2

La seconde itération:

chronogramme



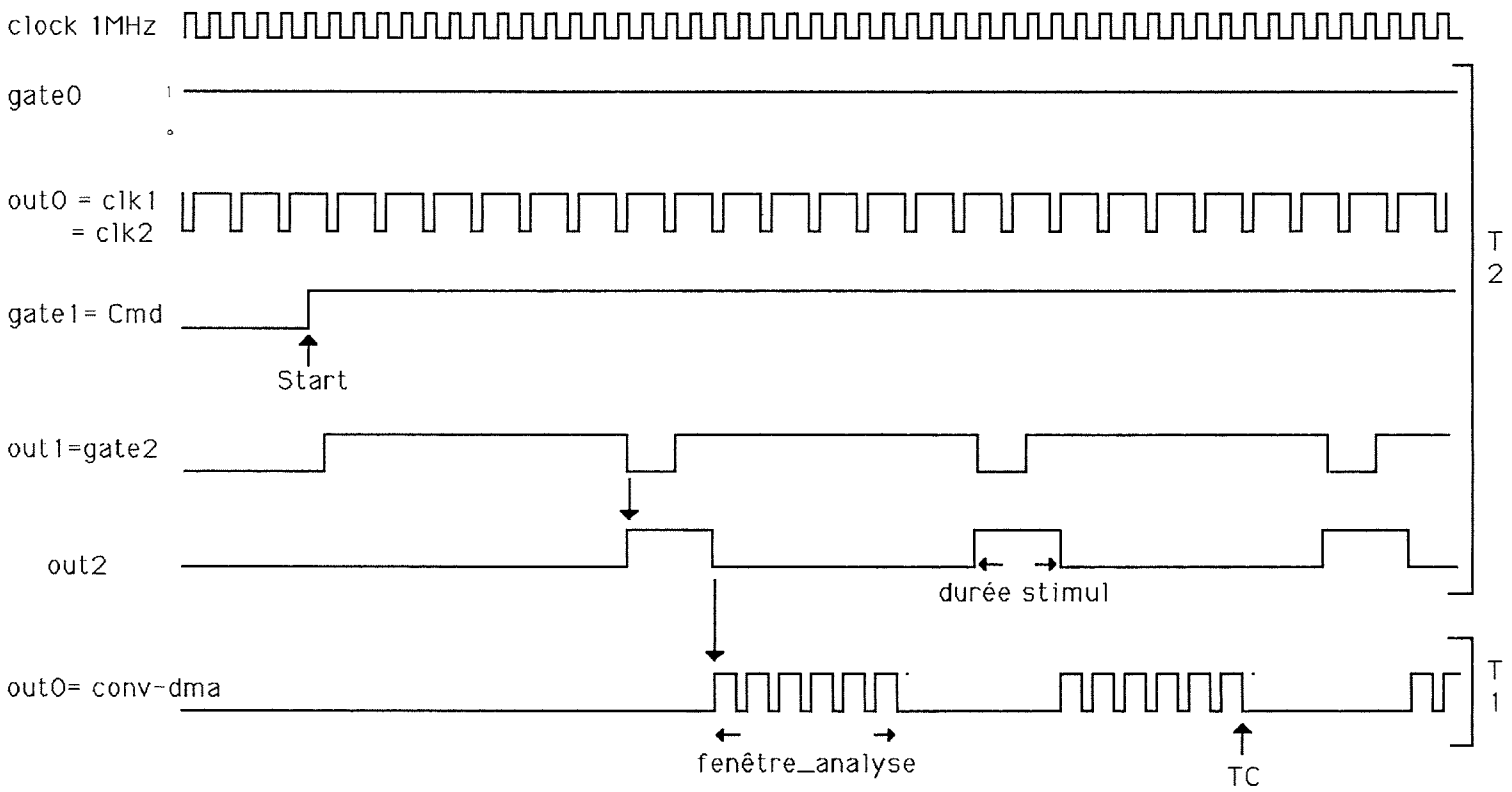
Sur le temps d'une fenêtre d'analyse, plusieurs centaines de conversions peuvent avoir lieu. Elles sont chaque fois suivies d'un transfert en mémoire. La génération d'un rythme par un timer sera également de mise. On peut voir sur le schéma bloc n°2 que le premier compteur du timer1, **T1.0**, est relié à l'**ADC**. L'échantillonnage est déclenché la première fois, par le **Cmd** et le **T2.2**.dont on a parlé ci-dessus; ce compteur intervient car le déclenchement d'une série de conversion doit toujours s'effectuer à la fin d'une stimulation. Ce rythme "conversion-dma" est interrompu par l'arrivée d'un signal "TC" (terminal count) agissant sur le bouton "Start/Stop". Ce signal est généré par le DMAC lorsque le nombre de transferts souhaités est réalisé.C'est donc en finalité, le DMAC qui contrôle la fenêtre d'analyse. Après chaque stimulation, le cycle reprend.

A la figure 6.1, on trouve un chronogramme résumant l'enchaînement des événements contrôlés par la carte.

Une dernière tâche incombe aux timers; le réglage des filtres. Nous avons signalé plus haut que les filtres se règlent par l'envoi de signaux horlogiques. Les timers peuvent également générer des signaux horlogiques (ondes carrées) de fréquence quelconque. Deux compteurs les **T1.1** et **T1.2**, seront choisis pour cette tâche.

Figure 6.1

Chronogramme résumant l'enchaînement
des traitements contrôlé par la carte



Voici la programmation résumée des deux timers 8254 :

```
#       const           ONESHOTSTROBE =1 /* générateur de délai      */
#                       RATEGENERATOR = 2 /* générateur de rythme   */
#                       SQUAREWAVE   = 3  /* générateur d'onde carrée */
#       /*ces constantes donnent le mode de fonctionnement des timers */
#
#       procédure Init_8254 (timer: timerchip,num,mode, délai:integer)
#       /* timer indique le numéro du timer ( 1 ou 2)/*
#       /* num indique le numéro du compteur (0,1 ou 2)/*
#       /* délai x  période de la Clock caractérise les délais ou les rythmes générés/*.
#
#       begin
#               Outport ( timer.cmdreg, mot_commande);
#       /* le mot de commande est fonction du mode choisi et et de num */
#               Outport ( timer.c0reg+num, LSB(délai));
#               Outport ( timer.c0reg+num, MSB(délai));
#       /* les adresses des compteurs se suivent;  en incrémentant c0reg de num */
#       /* ,on obtient l'adresse du bon compteur                          */
#
#       /*un entier sur 16 bits s'exporte en deux fois sur un bus 8 bits */
#       /*Les fonctions LSB et MSB séparent les 8 bits faibles des 8 bits forts */
#
#       end;

#       procédure réglage_filtres( FH,FB: integer);
#       var période:integer;
#       begin
#               période := 1/FB;
#               Init_8254( 1,1,SQUAREWAVE,période);
#               période:= 1/FH;
#               Init_8254( 1,2,SQUAREWAVE,période);
#       end;

#       procédure réglage_temps (fenêtre_analyse: integer);
#       begin
#               Init_8254( 2,1,RATEGENERATOR, fenêtre_analyse );
```

```
#       end;

#       procédure réglage_stimulation(durée_stimulation: integer);
#       begin
#               Init_8254( 2,2,ONESHOTSTROBE,durée_stimulation);
#       end;

#       procédure réglage_conversion( fenêtre_analyse,nbre_canaux, FH;
#                                     var résolution: integer);
#       begin
#               résolution := nbre_canaux * fenêtre_analyse * FH;
#               Init_8254( 1,0,RATEGENERATOR,résolution);
#       end;
```

Pour être complet nous ajoutons:

```
#       procédure Réglage_seuil(seuil:integer)
#       /*simple assignation de la variable Seuil */
```

Ces fonctions peuvent être rassemblées dans:

```
#       procédure Règlage_paramètres ( param: paramètres);
#       begin
#               réglage_filtre ( param.FH,param.FB);
#               réglage_temps ( param.fenêtre_analyse);
#               réglage_stimulation ( param.durée);
#               réglage_conversion (param.fenêtre_analyse,
#                           param.nbre_canaux,param.FH);
#               Réglage_seuil(param.seuil)
#       end;
```

Nous ne détaillons pas **Test_paramètres**(var **param**:paramètres), fonction de validation sans particularité.

Pour des renseignements plus précis sur la programmation des timers, notamment le contenu des mots de commandes ou les différents modes de fonctionnement nous renvoyons le lecteur aux annexes[8].

Voici résumée, la phase d'initialisation:

[8] annexe 5 : le timer 8254

```
#       procédure Init_acquisition( param : paramètres; var résolution: integer);
#       begin
#               Init_harware(canal_dma,dmac,T1,T2,adc,reset_adc,Cmd);
#               Test_paramètres(param);
#               Réglages_paramètres(param, résolution);
#       end;
```

Commande de la carte

Le déclenchement et l'arrêt de l'activité de la carte sont commandés depuis le processeur, par l'intermédiaire du 74HCT75 (**Cmd**). Son principe de fonctionnement est simple. Il reçoit un mot de commande et le transmet à différents composants de la carte notamment les timers, le sélecteur de stimulations et le bouton "Start/Stop" .
On distingue deux mots de commandes principaux pour cette carte: un mot pour inhiber la carte et un mot pour la déclencher.

Voici le schéma procédural qui en découle:

```
#       const   Start = 6; /* mot de commande pour déclencher la carte /*
#               Stop = 0;  /* mot de commande pour arrêter la carte /*
#
#       procédure Start_acquisition ;
#       begin
#               Outport( Cmd, Start)
#       end;

#       procédure Arrêt_acquisition;/* inhibe la carte  */
#       begin
#               Outport( Cmd,Stop)
#       end;
```

Phase 2: Conversion_dma_transfert , moyennage

Comme nous l'avons vu au point précédent, cette phase est itérée sur **nbre_stimulation.**
En voici le profil:

```
for i:= 1 to nbre_stimulation
begin
        Conversion_dma_transfert();
        moyennage();
end;
```

Programmation du DMA

Les détails de mise en oeuvre du DMAC sont consigné dans[9] .Nous n'en donnons ici que les grandes lignes.

Plusieurs informations doivent être fournies au DMAC:

- le nombre d'octets à transférer
- un mode de transfert
- le sens du transfert (vers la mémoire ou vers le périphérique)
- une adresse mémoire de début de transfert
- la sélection du canal de DMA

Il faut ensuite rendre le contrôleur capable d'accepter une demande de DMA.

```
#       procédure Start_dma ( buffer: array of integer,count: integer);
#       var addr: integer;
#       begin
#               addr := adresse( buffer );
#               Outport( maskreg, mot_mask );
#       /*cette instruction bloque le DMAC pendant le temps de programmation /*
#       /*le mot de masquage dépend du canal dma choisi  /*
#               Outport( modereg, mot_mode );
#       /*la valeur du mot de mode commandera un transfert simple, un transfert vers la
#       mémoire ( write) et choisit le canal dma  */
#               Outport( comptreg, count);
#               Outport( basereg, LSB (addr) );
#               Outport(basereg, MSB (addr) );
#               Outport( pagereg,PAG (addr) );
#       /* l'adressage se fait en trois phases ; octet de poids faible, octet de poids fort et #
        adresse de page*/
#               Outport( maskreg, no_mask);/*cette instruction débloque le  DMAC*/
#       end;
```

[9] annexe 4: le DMA

Le DMAC est lancé par le processeur, puis assure luimême le travail. Au niveau software, la procédure **Conversion_dma_transfert** se résume donc à **Start_dma.**

Le transit des données

. Lorsque le DMAC, programmé en "transfert simple"(donnée par donnée), importe les données,il n'occupe pas les bus en permanence. Le processeur peut effectuer, en parallèle, le moyennage de la série de mesures précédentes.
On procédera donc comme suit:
1° A l'initialisation, création et mise à zéro de deux zones tampons partagées par DMAC et CPU. Ces zones seront remplies alternativement;

type **buffer**: array[0..TAILLE]
var **tampon**:array[0..1] of buffer;
procédure **création_buffer**(buf: buffer); (non détaillée)

On ajoutera donc dans **Init_acquisition** les lignes suivantes:

création_buffer(tampon[0]);
création_buffer(tampon[1]);

avec **tampon[0]** et **tampon[1]** variables globales.

2° A chaque itération l'une est remplie, l'autre est traitée.
3° En terminaison, il reste à traiter une série de mesures.

. Comme le DMAC et le CPU vont travailler en parallèle, il est nécessaire de réaliser une synchronisation, à la fin de chaque itération:
le registre de status du DMAC indique, à chaque instant, l'arrivée des "TC" (Terminal Count"). Il suffit d'implémenter une surveillance de ce registre.

```
#       procédure Synchro_DMA_CPU( cmdreg, canal: integer)
#       /*le registre de status de DMAC indique l'arrivée des "TC"  */
#       begin
#               mot_status := 0;
#               while not (mot_status indique un "TC" à canal)
#               begin
#                       mot_status := Inport ( cmdreg);
#               end
#       end;
```

. Lorsque l'acquisition s'effectue sur plusieurs canaux (**nbre_canaux**>1: problème du multiplexage de canaux), la présence d'un unique ADC oblige à une conversion parallèle-série (le convertisseur balaie les différents canaux). Les données arrivent donc en mémoire, entrelaçées. C'est au processeur d'en faire le tri.

Le moyennage

Il s'agit d'abord de contrôler les données par rapport au **seuil_rejet**

```
#       procédure rejet_artefact( buf: buffer; var status: integer);
#       /* simple boucle et retourne 0 si pas d'artefact*/
```

Ensuite, on trie, somme et on moyenne les données du tampon:

```
#       var compteur_stimuli, résolution : integer;
#       procédure additionneur ( var somme: array[1..TAILLE] of integer;
#                                       num, nbre_canaux: integer);
#               begin
#                       j:= 1; i:=num ;
#                       for j:=1 to résolution
#                       begin
#                               somme[j] :=somme[j] + tampon[ 0 ou 1][i];
#                               i:= i + nbre_canaux;
#                       end
#               end;
#       procédure moyenneur ( somme: array[1..TAILLE] of integer;
#                               var moyenne: array[1..TAILLE] of integer);
#       /*compteur_stimuli donne le nombre de stimuli déjà additionnés ( globale) */
#       begin
#               for j:=1 to résolution
#               moyenne[j] := somme[j]/compteur_stimuli
#               end;
```

Les procédures d'affichage graphique ne seront pas détaillées ici. Un élément important est toutefois à signaler.: l'affichage des traces a lieu en fin de DMA sur les données courantes tandis que l'affichage des moyennes, est réalisé sur les données provenant de l'acquisition précédente. Un dédoublement est donc nécessaire.

Phase 3: Interruption de l'activité de la carte et traitements de clôture

La phase précédente a montré qu'il restait, à la terminaison de la boucle principale, une série de mesures à moyenner. On exécute donc une dernières fois la série de procédures de traitement: **rejet_artefact**(), **additionneur**(), **moyenneur**(). Avant cela, il faut évidemment interrompre le processus d'acquisition. C'est le rôle de **Arrêt_acquisition**().

Voici pour finir l'implémentation globale de l'acquisition:

```
Type  buffer: array[1..TAILLE] of integer;
      tableau: array[1..TAILLE] of integer;

var   tampon: array[0..1] of buffer;
      compteur_stimuli,compteur_artefact, status, j: integer;
      somme, moyenne: array[1..CANAUX_MAX] of tableau;
      /* canaux_max est le nombre de canaux maximum autorisé */
      résolution : integer;

procédure acquisition ( param: paramètres);
begin
      Init_acquisition( param, résolution );
      Start_acquisition;
      for compteur_stimuli:= 1 to param.nbre_stimulation
      begin
            Start_dma (buffer[1 ou 0],;
            rejet_artefact(tampon[0 ou 1], status);
            if status=0 then
            begin
                  for j:= 1 to param.nbre_canaux
                  begin
                        additionneur(somme[j],j);
                        moyenneur(somme[j],moyenne[j]);
                  end
            end;
            else
            begin
```

```
                    compteur_stimuli:=compteur_stimuli -1 ;
                    compteur_artefact:=compteur_artefact +1
            end;
            Synchro_DMA_CPU
        end;
        Arrêt_acquisition;
        rejet_artefact(tampon[0 ou 1], status);
        /* celui qui reste à traiter*/
        if status=0 then
        begin
            for j:= 1 to param.nbre_canaux
            begin
                additionneur(somme[j],j);
                moyenneur(somme[j],moyenne[j])
            end
        end;
        else
        begin
            compteur_artefact:=compteur_artefact +1;
        end;
    end;
```

Il reste à insérer les procédures de présentation et d'enregistrement que nous n'avons pas détaillées.

VI.2 IMPLANTATION DU SYSTEME DE COORDINATION

Bien qu'il soit hors de question de détailler toutes les fonctions et sous-fonctions participant à l'interface réalisé pour ce travail, il faut toutefois dire quelques mots sur certaines d'entre elles. En effet, il faut savoir que dans l'optique que nous avons choisie d'avoir constamment à l'écran une mesure (soit la dernière acquise, soit la dernière chargée) nous ne pouvions échapper au mode graphique pour l'élaboration de l'interface (Apparemment, il n'y a pas grand-chose de réalisé à ce sujet .). Comme avec le mode graphique les notions de curseur et de caractère disparaissent, il faut, pour réaliser une prise de données au clavier, la 'simuler'.

Décrivons maintenant trois outils primordiaux de l'interface.

Affichage d'une chaîne de caractères 'entre deux bornes'

Ceci est réalisé grâce à la procédure **Disp** qui reçoit:

.un pointeur vers un caractère (début de la chaîne à afficher)

.un entier qui va servir à définir la colonne où doit commencer l'affichage

.un entier qui va servir à définir la ligne où doit commencer l'affichage

Suivant le nombre de caractères composant la chaîne, la fonction Disp effectue elle-même le retour à la ligne (à la même colonne).

Saisie à l'écran d'une chaîne de caractères 'entre deux bornes'

Ceci est réalisé grâce à la procédure **EditString** qui reçoit:

.un pointeur vers un caractère(début du string à saisir)

.deux entiers spécifiant la colonne et la ligne où la prise de caractères doit commencer

.un entier spécifiant la colonne à droite que ne peut dépasser la prise de caractères (donc EditString gère également les retours à la ligne lorsque le nombre de caractères à saisir est supérieur à l'intervalle séparant les deux bornes.)

.un entier spécifiant le maximum de caractères que l'on veut saisir à l'écran

Avant toute saisie, EditString affiche des paramètres par défaut.
La visualisation(déplacement...) du curseur, primordiale pour l'utilisateur, est également gérée par EditString via les procédures:

. **HideCurs** va chercher le pixel actif. Cette procédure ne reçoit pas de paramètres.

. **DisplayCurs** dessine un curseur de la largeur de caractère choisie. Elle reçoit en paramètres : trois entiers. Les deux premiers spécifiant l'emplacement où le curseur doit être dessiné et le troisième la couleur d'affichage.

. **MoveCurs** déplace le curseur. Elle reçoit un entier spécifiant le nombre de caractères dont le curseur doit se déplacer ainsi que deux autres entiers spécifiant les bornes limites du déplacement permis.

Elle renvoie un entier indiquant la touche spéciale du clavier frappée. Cette dernière a une signification particulière pour la gestion des fenêtres. Pensons, par exemple, à la touche RETURN, à la flèche de déplacement vers le haut,...

Gestion d'une fenêtre de dialogue

La gestion d'une fenêtre de dialogue est prise en charge par la procédure **SaisirValider** qui reçoit comme seul paramètre le numéro de la fenêtre en question.
Elle réalise :

. l'affichage, s'il y en a des paramètres par défaut dans les divers champs de la fenêtre
. le calcul du nombre maximum de caractères à saisir.
. la gestion des déplacements de zones lumineuses.Ceci sert de point de repère à l'utilisateur.

Elle renvoie un entier indiquant la frappe d'une touche déterminante (ESC, le OK de la fenêtre,...).
L'utilisateur peut à tout moment revenir sur un champ mal encodé. Certaines "fonctionnalités" impliquent une saisie et un affichage important d'informations qui doivent se faire par étapes successives.Prenons l'exemple d'un enregistrement de PE. Il commence, au début de l'examen, par la prise des paramètres spécifiques; à la fin de l'acquisition, suivent alors l'affichage de la trace finale, la prise des coordonnées du patient, la saisie des commentaires éventuels du médecin et l'affichage du numéro de dossier attribué. Les trois dernières étapes sont mises en évidence en annexe 7[10]. Il est plus rassurant pour l'utilisateur d'avoir autant que possible le plus d'étapes de même finalité (par exemple ici l'enregistrement d'un PE) sous les yeux au même instant. Comme l'interface permet , dans une même fenêtre, de revenir en arrière, il permet également de revenir sur ses pas dans l'ordre des fenêtres grâce à la touche ESCAPE. Cela implique donc que l'enregistrement réel du PE dans la base de données ne s'effectue qu'à la toute dernière étape. Il est réalisé en sélectionnant le bouton "OK" de la fenêtre affichant le numéro de dossier attribué. On peut avoir un aperçu des divers écrans qui constituent l'interface en se reportant aux annexes[11].

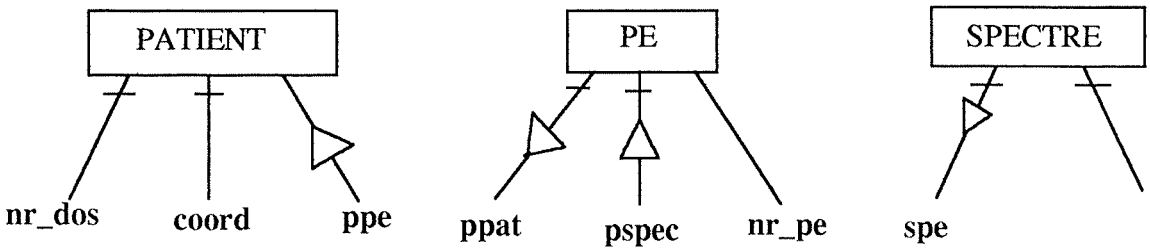
VI.3. IMPLANTATION DU SYSTEME D'INFORMATION

Nous n'aurions pas voulu conclure ce travail sans quelques mots sur la troisième et dernière brique de base du logiciel: le système d'information. C'est lui, en effet, qui réalise la communication nécessaire entre l'acquisition et l'application "temps différés".

Stockage des informations

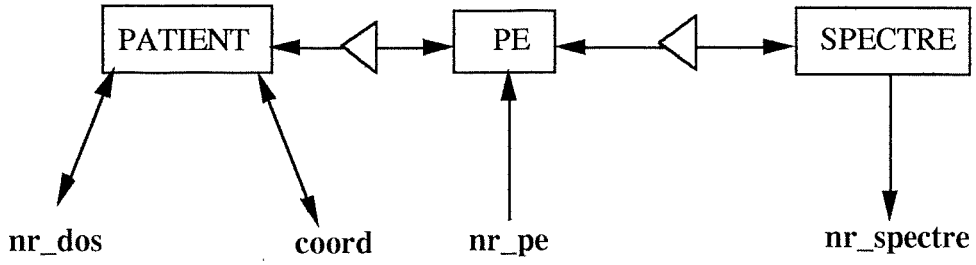
[10] page 2 de l'annexe 7
[11] annexe 7: les écrans

Etant donné le petit nombre de types d'articles, nous attribuerons un fichier par article. Nous rangerons dans un fichier de nom FPATIENT le type d'article PATIENT, dans un fichier de nom FPE le type d'article PE et dans un fichier de nom FSPECTRE le type d'article SPECTRE.
Sur base du SAN (Schéma des accès nécessaires) de la figure 6.2, on peut réaliser le schéma conforme de la même figure. Pour cela, il faut éliminer les deux types de chemin par rotation avec addition de nouveaux items (ppe,pspect et ppat) et contraintes d'intégrité.



Contraintes d'intégrité: ppe(:PATIENT) in nr_pe(:PE)
pspec(:PE) in nr_spectre(:SPECTRE)
ppat(:PE) in nr_dos(:PATIENT)
spe(:SPECTRE) in nr_pe(:PE)

Schéma conforme



SAN

Fig.6.2

Le fichier FPATIENT sera d'organisation séquentielle indexée car les patients resteront toujours dans la BD. Comme la copie sur disquette est répandue dans les milieux médicaux, les PEs et spectres enregistrés risquent de ne pas rester longtemps dans la BD nous choisirons pour les fichiers FPE et FSPECTRE une organisation séquentielle simple.

Conclusion

CONCLUSION

L'objet premier du mémoire était d'apporter des modifications à un système d'acquisition et de traitement de potentiels évoqués.
En fait de modifications, ce fut une totale refonte du projet. En effet, celui-ci se composait en tout et pour tout d'un très sommaire cahier des charges accompagné de lignes de programme confuses et d'une documentation très technique.
Notre formation de physicien nous a fortement aidé à aborder le sujet et à en comprendre les spécificités techniques, notamment le comportement de la carte d'interface sur base de son schéma bloc.
Mais avant toute implantation, un travail d'analyse et de conception devait être entrepris. Pour les réaliser, nous nous sommes aidés des diverses méthodes enseignées.
L'adaptation de celles-ci aux spécificités d'une application "temps réel" fut un des aspects particulièrement intéressant de ce travail. Nous avons été sans cesse obligés, pour échafauder une solution, de développer deux approches jugées souvent antagonistes: l'une basée sur une analyse abstraite du problème, l'autre élaborant une solution sur base des éléments techniques dont nous disposions.
Nous dirons pour conclure, qu'après deux années de formation à l'informatique de gestion, nous sommes contents d'avoir pu nous exercer dans un tout autre domaine : le contrôle de processus en temps réel.

ANNEXES

Annexe 1. Structure des cellules nerveuses

L'ensemble du système nerveux est constitué de cellules à caractères spécifiques que l'on appelle neurones et qui, comme les cellules d'autres tissus, sont séparées de leurs voisines par une membrane continue.
Des protubérances de deux types sont attachées à la partie centrale de la cellule (corps cellulaire) : des dendrites par lesquelles arrivent au corps cellulaire les stimuli et une structure longue et fine, appelée axone, par lequel se propagent les stimuli en provenance du corps cellulaire. A de très rares exceptions près, les membranes de neurones distincts ne fusionnent pas et les contacts interneuronaux s'effectuent par contiguité et non continuité. L'influence d'une cellule nerveuse sur d'autres s'effectue en leurs points de contiguité appelés synapses.

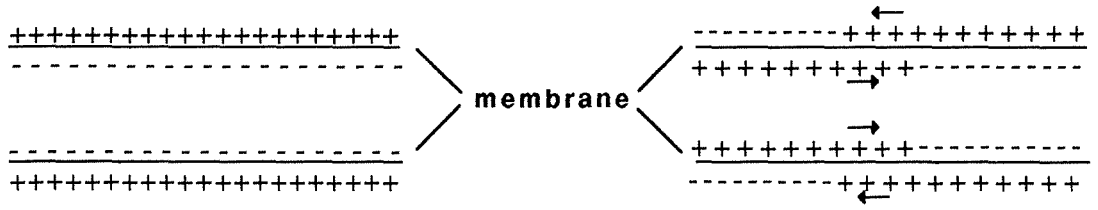
1.Concentration ionique et potentiel de repos.

La fibre nerveuse ou musculaire présente une différence de potentiel entre son intérieur et son extérieur telle que l'intérieur est négatif par rapport à l'extérieur. Par convention, on prend le potentiel électrique du fluide à l'extérieur comme zéro de potentiel et on trouve ainsi des différences de potentiels extérieur-intérieur de l'ordre de 9O mV. Cette différence de potentielle appelée potentiel de repos trouve son origine dans les différences de concentrations ioniques qui existent entre les milieux extra et intra - cellulaires. On trouve entre autres, les ions Cl-, K+,Na+ et de gros anions (A+). On a pu montrer que l'intérieur de la cellule renferme peu d'ions Cl- et Na+ mais beaucoup d'ions A- et K+. La situation inverse se présente à l'extérieur de la cellule. Dès lors, les passages ioniques transmembranaires sont gouvernés par deux gradients, l'un chimique, l'autre ionique mais aussi par les propriétés de perméabilité de la membrane à ces ions. Les concentrations hors équilibres sont maintenues à l'aide d'un processus encore mal connu appelé pompe à Na-K, qui transporte les ions Na+ et K+ en sens inverse à travers la membrane et est ainsi responsable de l'établissement et du maintien du potentiel de repos.

2.Le potentiel d'action.

Supposons que l'on désire examiner la réponse d'un axone à un stimulus faible. On constate qu'en deçà d'un seuil critique, la réponse de l'axone est quasi inexistante. Par contre dès que le stimulus est supérieur au seuil critique, il produit une impulsion de courant qui va se propager le long de l'axone et cela sans atténuation. En fait, le

dépassement du seuil provoque une augmentation brusque de la perméabilité au Na+ (d'un facteur 1OOO)
donc une entrée massive d'ions Na+ positifs qui renverse les polarités et rend l'intérieur de la cellule, à cet endroit, positif par rapport à l'extérieur (le potentiel de l'axone peut passer, par exemple, de 9O mV à 5O mV)



figures *(a)* *(b)*

Il en résulte qu'à l'intérieur de l'axone (cfr. fig.(b)), les ions positifs vont s'éloigner de la zone d'application du stimulus (vers la droite du dessin). A l'extérieur, les ions positifs vont se rapprocher de la zone d'application. La zone immédiatement adjacente va donc voir son potentiel augmenter. Dès qu'il aura atteint le seuil de potentiel d'action, la membrane, en cet endroit, verra à son tour, sa perméabilité à Na+ augmenter et ainsi de suite de proche en proche, l'impulsion de potentiel d'action se propagera le long de l'axone.

Cette activité électrique interne peut être appréhendée grâce à des électrodes, au niveau de la peau et conduit à toute une série de techniques telles que l'encéphalographie et les potentiels évoqués.

Annexe 2

Cette annexe renferme les différents types de rapports médicaux à effectuer après acquisition de potentiels évoqués.

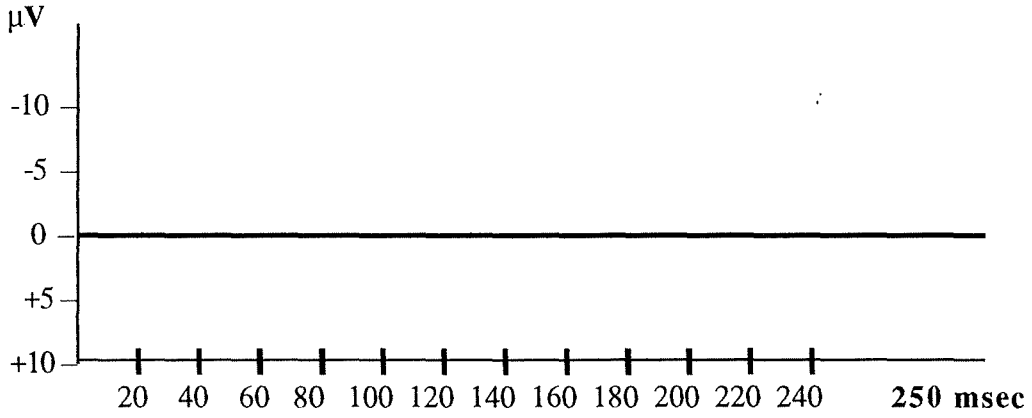
Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

Potentiels évoqués visuels

Stimulation binoculaire : Flash 1OO stimuli récurrence 1 x/sec
Electrode : 1° canal O1 - F3 (Trace rouge) Hémisphère gauche
2° canal O2 - F4 (Trace bleue) Hémiphère droit
Bande passante : 1 à 1OO Hz.



Nom et adresse du laboratoire :

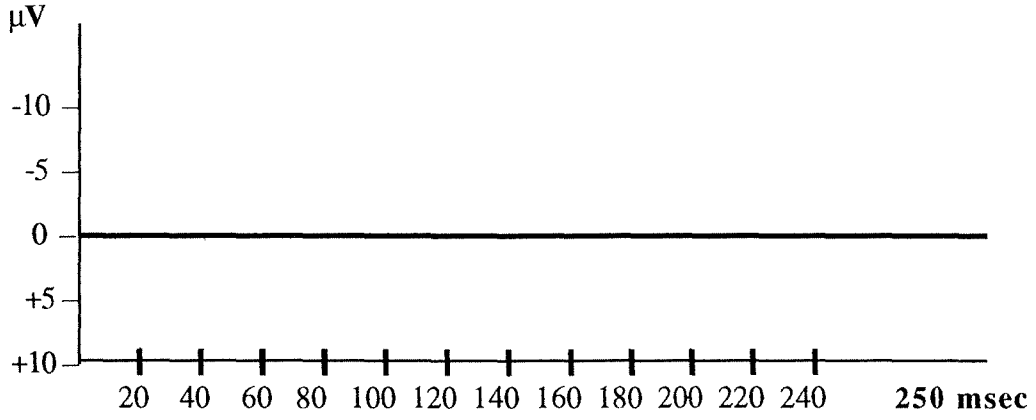
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

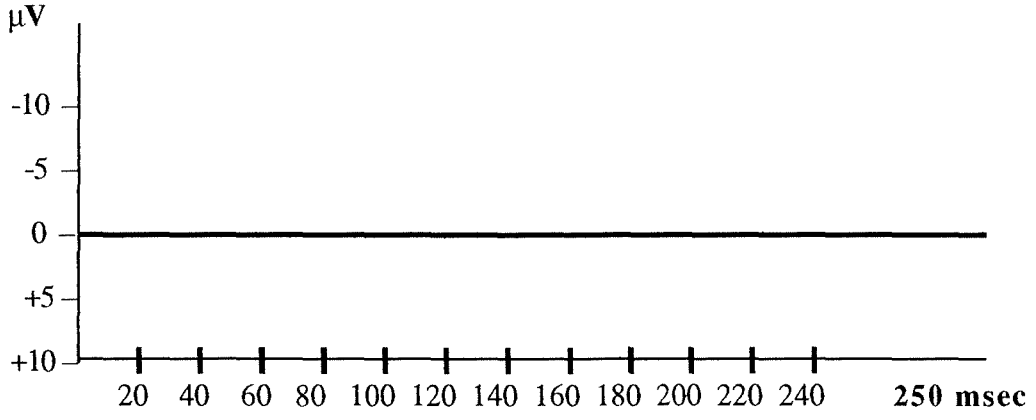
Potentiels évoqués visuels

Stimulation monoculaire : Flash 1OO stimuli.Récurrence 1 xsec.
Electrode : 1° canal O1 - F3 (Trace rouge) Hémisphère gauche
2° canal O2 - F4 (Trace bleue) Hémiphère droit
Bande passante : 1 à 1OO Hz

Stimulation oeil gauche.



Stimulation oeil droit.



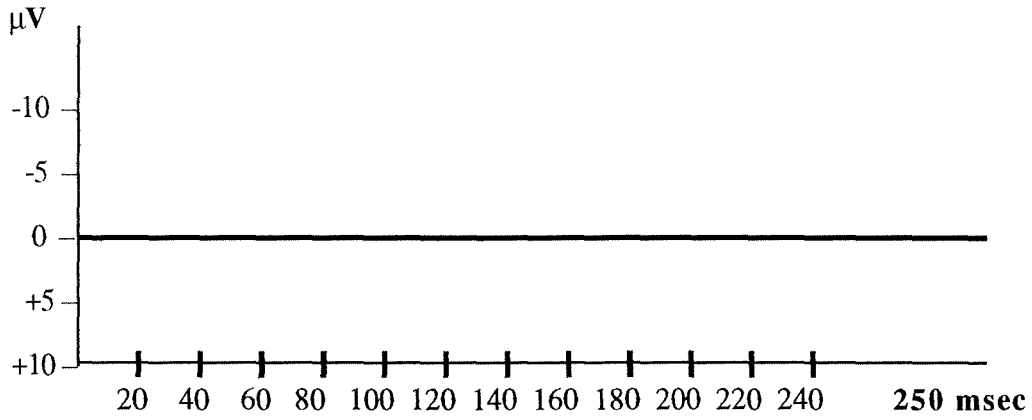
Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

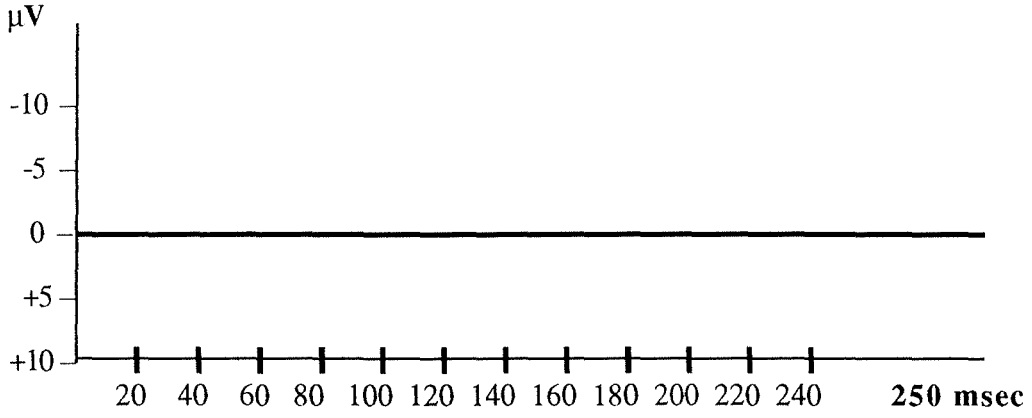
Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

Potentiels évoqués visuels

Stimulation binoculaire : damier 1OO stimuli récurrence O,75 x/sec.
Electrode : 1° canal O1 - F3 Trace rouge Hémisphère gauche
2° canal O2 - F4 Trace bleue Hémisphère droit
Bande passante : 1 à 1OO Hz
Dimension : Fréquence spectrale :



Dimension : Fréquence spectrale :



Nom et adresse du laboratoire :

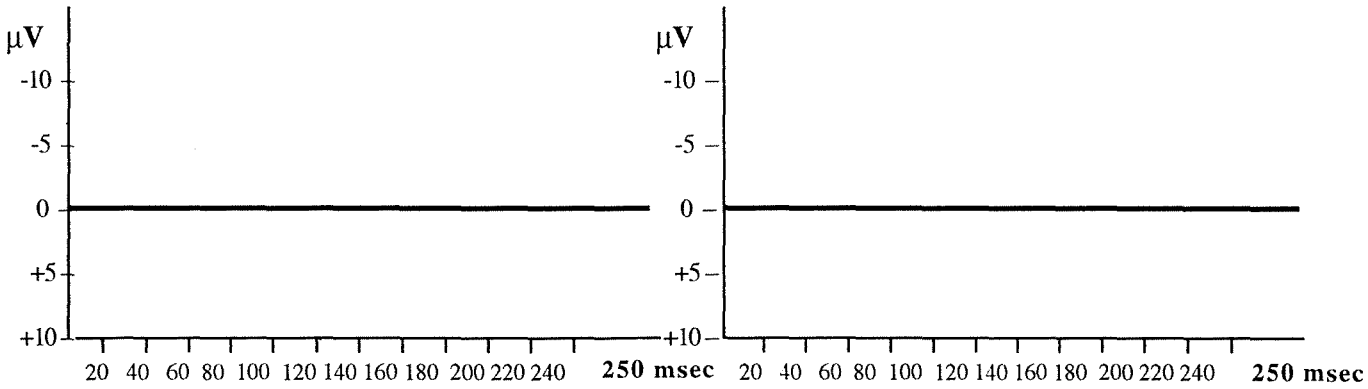
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

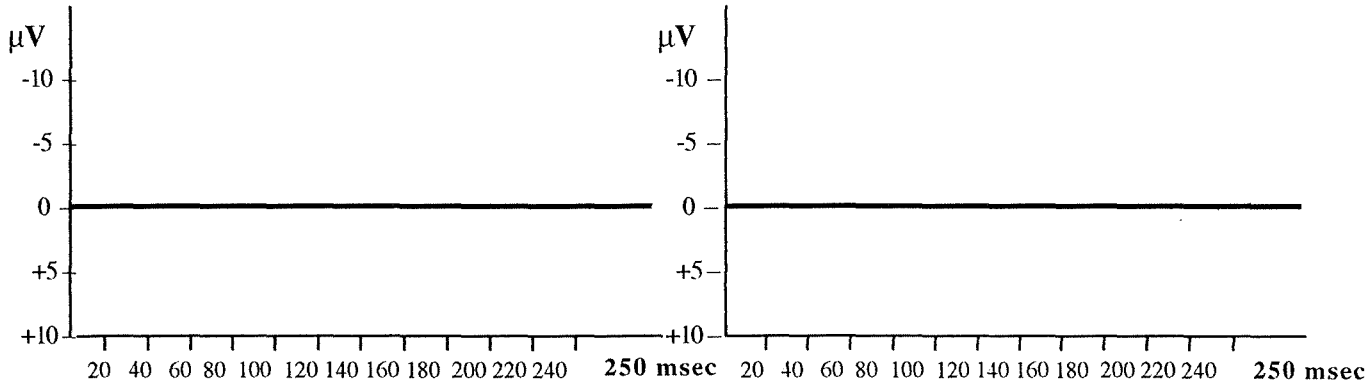
Potentiels évoqués visuels

Stimulation monoculaire : damier 1OO stimuli récurrence O,75 x/sec.
Electrode 1° canal O1 - F3 Trace rouge Hémisphère gauche
2° canal O2 - F4 Trace bleue Hémisphère droit
Bande passante : 1 à 1OO Hz

1 Dimension : Fréquence spatiale :
Stimulation oeil gauche Stimulation oeil droit



2 Dimension : Fréquence spatiale :
Stimulation oeil gauche Stimulation oeil droit



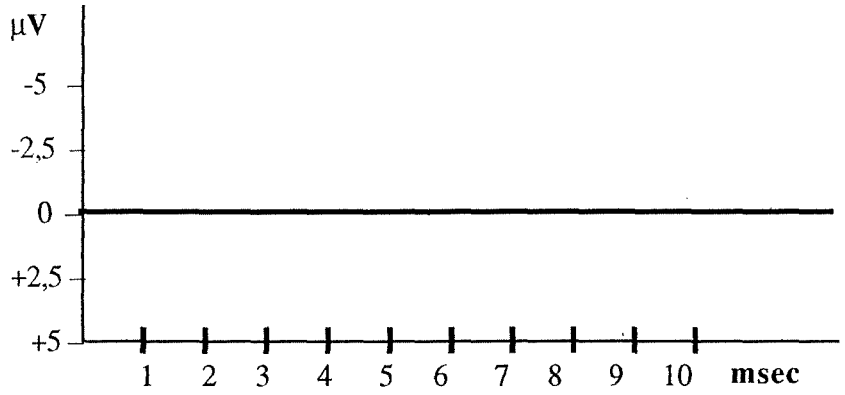
Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

Potentiels évoqués auditifs précoces.

Stimulation binoculaire : clic
tone-pipe : durée 1OO µsec 1.5OO stimuli
récurrence : 2O x/sec.
Electrode 1° canal T3 - CZ Trace rouge Hémisphère gauche
2° canal T4 - CZ Trace bleue Hémisphère droit
Bande passante : 1OO à 3.OOO Hz



Nom et adresse du laboratoire :

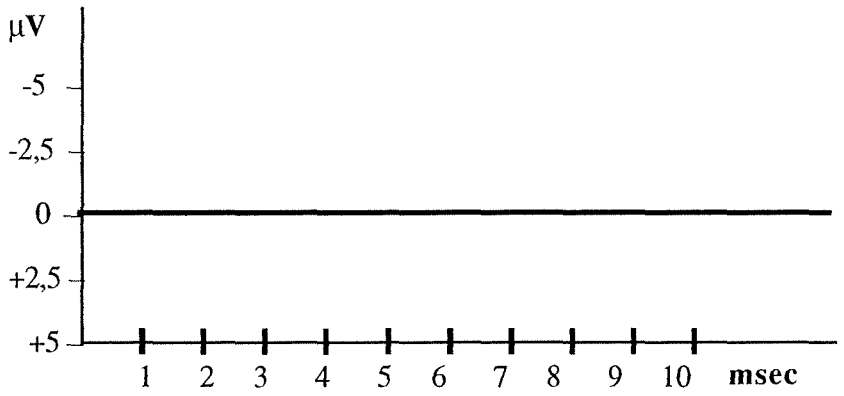
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

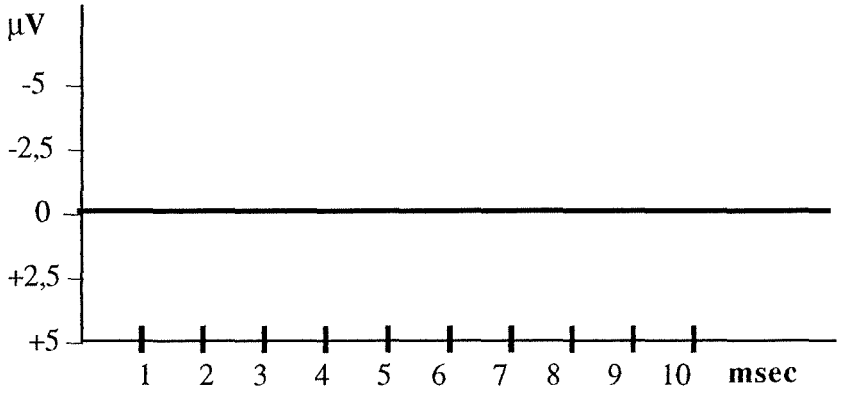
Potentiels évoqués auditifs précoces.

Stimulation mono-auriculaire : clic
Tone-pipe alterné durée 1OOµsec15OO stimuli
Récurrence : 2O x/sec.
Electrodes : 1° canal : T3 - CZ Trace rouge hémisphère gauche
2° canal : T4 - CZ Trace bleue hémisphère droit
Bande passade : 1OO à 3.OOO Hz

Stimulation oreille gauche



Stimulation oreille droite



Nom et adresse du laboratoire :

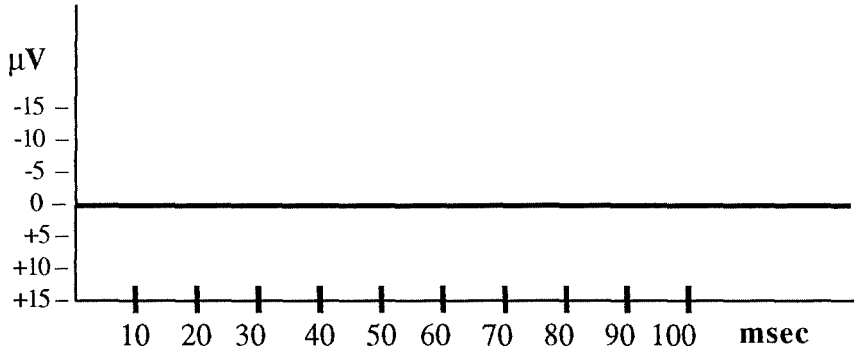
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

Potentiels évoqués auditifs semi-précoces

Stimulation biauriculaire : Tone-pipe alterné durée 1OOµsec
Clic
1OOO stimuli

Récurrence : 2O x/sec.
Electrodes : 1° canal : T3 - CZ Trace rouge hémisphère gauche
2° canal : T4 - CZ Trace bleue hémisphère droit
Bande passante : 5O à 2.OOO Hz



Nom et adresse du laboratoire :

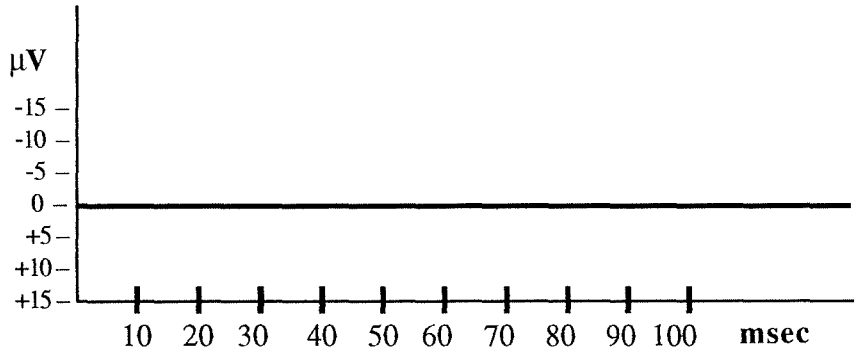
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non

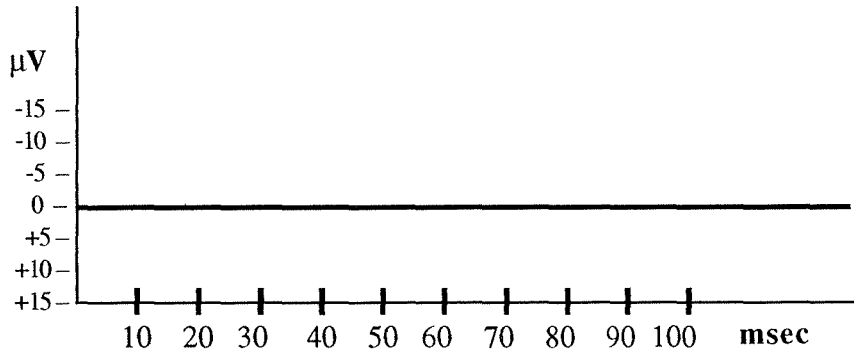
Potentiels évoqués auditifs semi-précoces

Stimulation mono-auriculaire : Tone-pipe altéré durée 1OOµsec
Clic
1OOO stimuli
Récurrence 2O x/sec.
Electrodes : 1° canal : T3 - CZ Trace rouge hémisphère gauche
2° canal : T4 - CZ Trace bleue hémisphère droit
Bande passante : 5O à 2OOO Hz

Stimulation oreille gauche



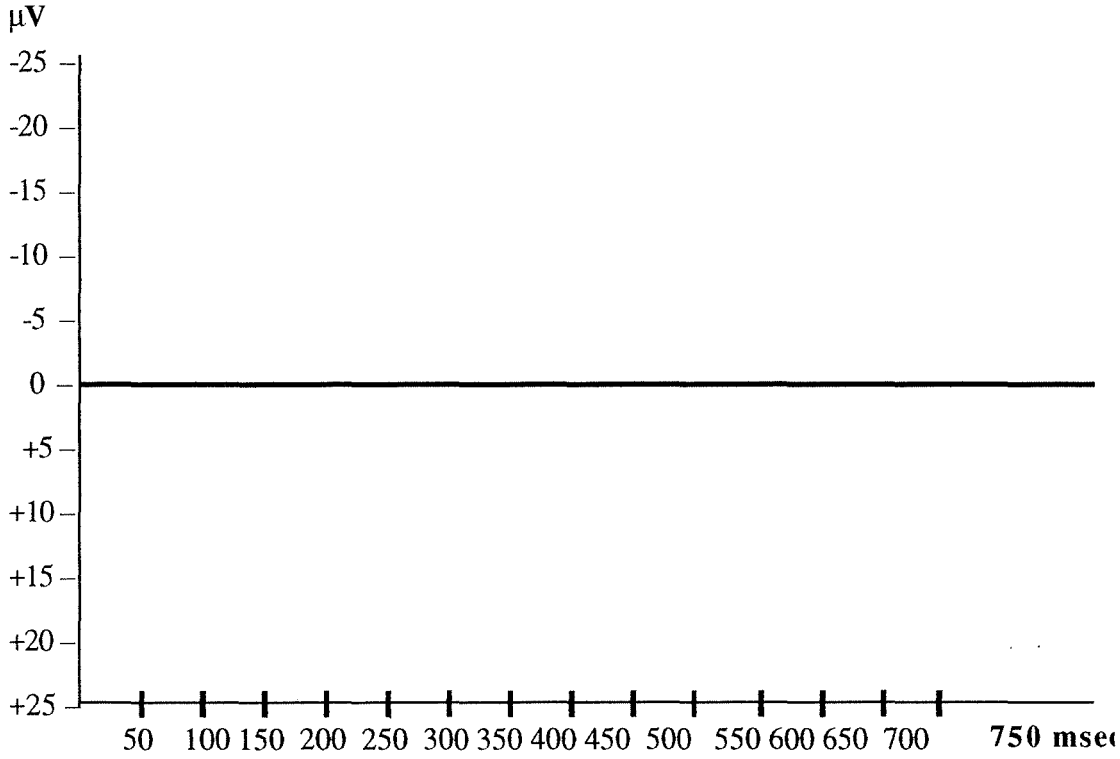
Stimulation oreille droite



Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non
Potentiels évoqués auditifs tardifs.
Stimulation biauriculaire : Tone burst : 1OOOHz durée 1OOµsec. 2OO stimuli
Récurrence : 1 x/sec.
Electrodes : 1° canal : T3 - CZ Trace rouge hémisphère gauhe
2° canal : T4 - CZ Trace bleue hémisphère droit
Bande passante : 1 à 1OO Hz

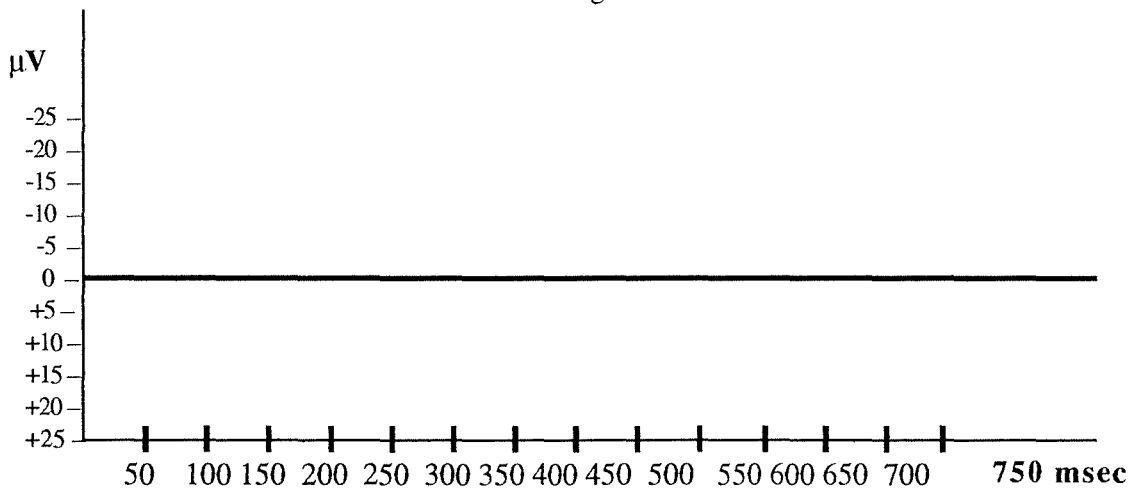


Nom et adresse du laboratoire :

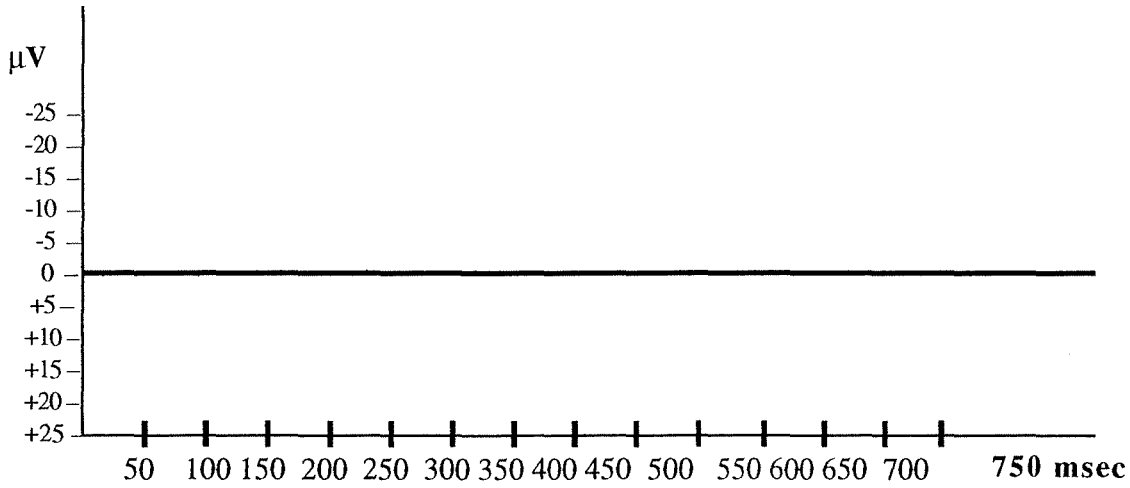
Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non
Potentiels évoqués auditifs tardifs
Stimulatin mono-auriculaire : Tone burst :1OOOHz durée 1OOµsec 2OO stimuli
Récurrence : 1 x/sec.
Electrodes :1° canal : T3 - CZ Trace rouge hémisphère gauche
2° canal : T4 - CZ Trace bleue hémisphère droit
Bande passante : 1 à 1OO Hz

Stimulation oreille gauche



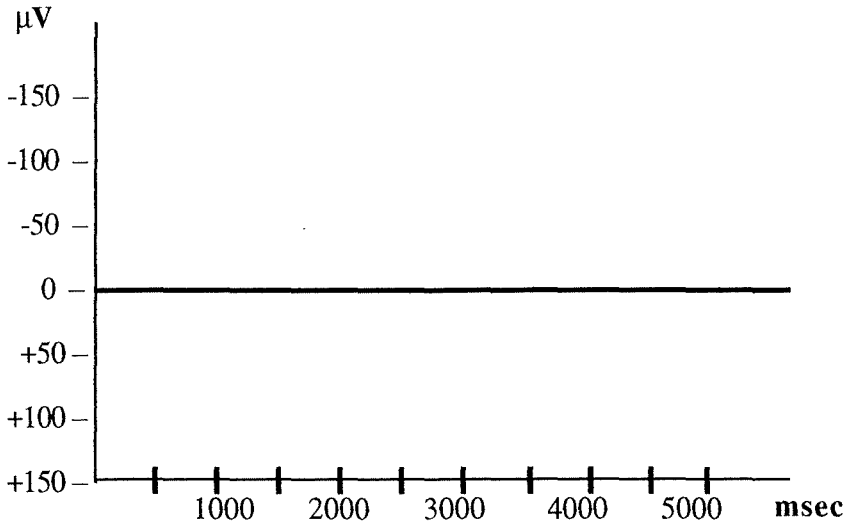
Stimulation oreille droite



Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

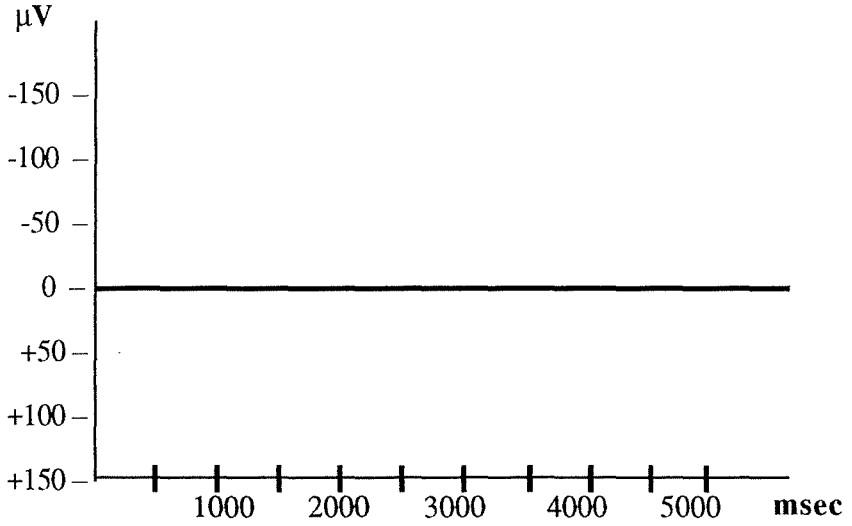
Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non
Variation contingente négative
Stimulation : Son 1OOO Hz durée 1OO µsec intensité : 7O db
suivi 1 sec après d'un flash (1O écl./sec.) (...11) 24 essais
Récurrence : O,1 x/sec.
Electrodes : 1° canal : FZ - PZ Trace rouge
2° canal : CZ - T3 Trace bleue
Bande passante : 1 à 3O Hz



Nom et adresse du laboratoire :

Exploration fonctionnelle du système nerveux.

Nom : Prénom : Sexe :
Date de naissance : Date de l'examen :
Dossier n° : Provenance :
Etat du sujet : collaborant ou non
Onde P 3OO
Stimulation : son d 15OO Hz durée ??? intensité 7O db
18O stimuli
entrecoupées de façon aléatoire par 2O stimuli sonores à 1OOO Hz et de durée ???
Recurrence : 1x/sec

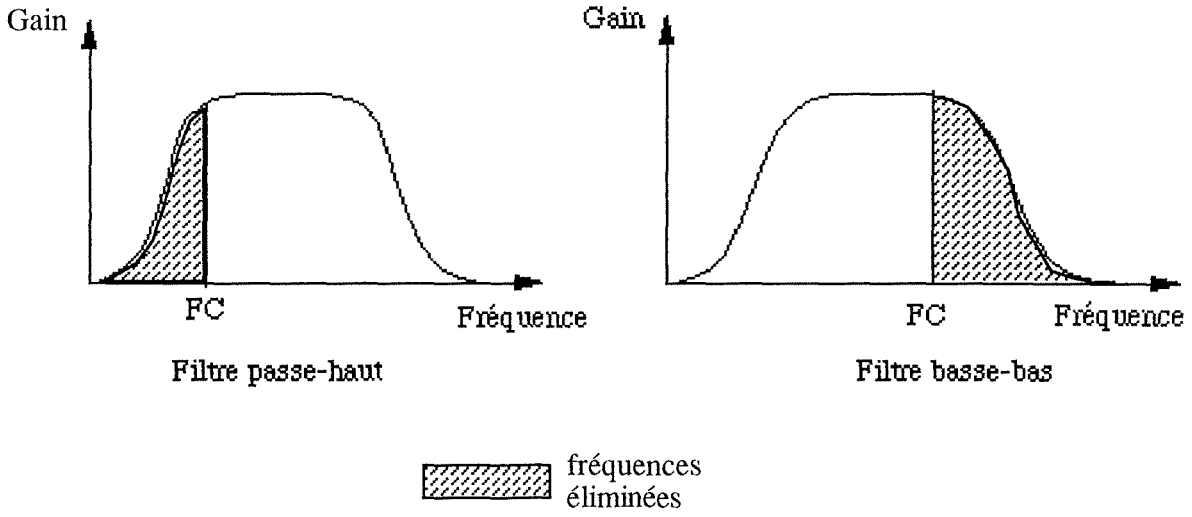


Annexe 3: La bande passante

La bande passante d'une chaîne d'amplification est l'intervalle de fréquences du signal à l'intérieur duquel cette chaîne d'amplification travaille avec un gain constant. L'intervalle de fréquences est défini par deux fréquences:

1) la plus basse est déterminée par un filtre passe-haut qui diminue le gain de la chaîne d'amplification pour une fréquence inférieure à sa fréquence de coupure (FC).

2) la plus haute est,elle, déterminée par un filtre passe-bas qui diminue le gain pour une fréquence supérieure à sa FC.



Annexe 4: L' acces direct à la mémoire

L'accès direct à la mémoire, le DMA, est une technique qui permet de transférer des données entre la mémoire et un organe périphérique, sans passer par le microprocesseur. On peut ainsi décharger celui-ci et accélérer l'exécution du programme..
Dans les "personal computers" on réalise l'accès direct à la mémoire par le contrôleur 8237A de chez Intel(le DMAC). Il fournit quatres canaux de transfert DMA capables de transférer des blocs de 64K. L'utilisation de contrôleurs en cascade permet d'augmenter le nombre de canaux possibles

Les opérations DMA (figure a):

1) Avant tout transfert , le contrôleur doit être initialisé. Plusieurs actions doivent donc être faites:

- introduire le nombre d'octets à transférer.
- choisir un type de transfert :
  - simple: permet, entre deux données transférées, de pouvoir libérer le bus pour le microprocesseur.
  - bloc: garde le bus tout le temps du transfert des données
- sélectionner la lecture ou l'écriture:
  - lecture: le transfert de la mémoire vers un périphérique.
  - écriture: le transfert d'un périphérique.vers la mémoire
- donner une adresse mémoire de début de transfert:
  - soit l'adresse de la première donnée à émettre, soit l'adresse où placer la première donnée à recevoir.
- choisir un canal de DMA et en fixer la priorité:
  - plusieurs canaux pouvant être actifs, en même temps, un ordre de priorité est important.
- rendre le contrôleur capable d'accepter une demande de DMA.

Cette initialisation est faite en envoyant des "mots de commande" aux registres du 8237 DMAC via les les ports d'E/S .

2) Un périphérique fait une demande de DMA . Cette demande est envoyée sur une ligne de"demande de dma" (DREQ). Il y a autant de ces lignes qu'il y a de canaux.(ex : pour le 80286: DREQ0 - DREQ7).

3) Le contrôleur gère la priorité de la demande et envoie une confirmation au périphérique via une des lignes DACK. Il envoie également une demande vers le microprocesseur sur sa ligne " Hold Request line" (HRQ).

4) Le microprocesseur achève son travail en cours, libère les ressources nécessaires au DMA (bus des adresses, le bus de contrôle et le bus des données), se place dans un état d'attente et envoie un signal sur la ligne "hold acknowledge" (HLDA) vers le 8237.

5) Le DMAC prend alors le contrôle du système: il génère simultanément l'adresse mémoire concernée par le transfert, sur le bus des adresses et envoie une impulsion READ ou WRITE demandant ainsi à la mémoire d'enregistrer une donnée déposée par un périphérique au moment de l'ACK ou de livrer une donnée. La valeur de cette adresse, contenue dans un registre, est automatiquement incrémentée ou décrémentée après chaque transfert. Les données ne transitent pas par le circuit DMA. Il ne faut dès lors que 5 cycles d'horloge pour transférer un octet, alors que si le microprocesseur effectuait ce transfert lui-même, il ne lui faudrait pas moins de 29 cycles (les transferts DMA sont donc 6 fois plus rapide).

Le registre de mot courant qui détient le nombre de transferts à effectuer, est décrémenté après chaque transfert. Remarquons qu'en cas de transfert "simple", le processeur est capable d'exécuter un cycle machine complet entre chaque transfert.

6) Le 8237 DMAC génère un signal ,appelé "terminal count" (TC), lorsque le registre de mot courant atteint la valeur FFFFH.

7) Le DMAC désactive alors le signal HOLD. Aussitôt que cela est détecté, le processeur désactive HLDA et retrouve le contrôle du système.

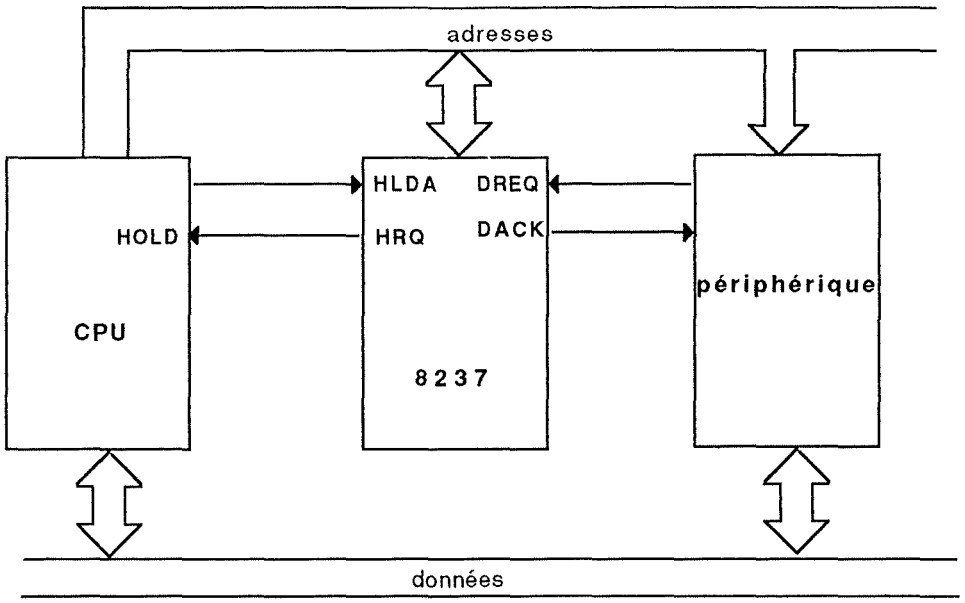


figure a

La programmation du DMAC:

Nous avons vu plus haut que les opérations faites par le DMAC étaient fonction d'une programmation initiale. Le processeur adresse les différents registres du DMAC par les ports d'E/S réservés à cet usage.
Les registres principaux sont:

- le registre de commande
  Programmé par le processeur, il contrôle les opérations du DMAC.
- les registres de mode
  Il y en a un par canal de DMA. Ils déterminent le mode de transfert pour chacun d'eux.
- le registre de demande
  Il permet de produire des demandes de DMA par software.
- les registres de masquage
  Ils permettent de masquer une demande de DMA pour un canal particulier.
- le registre de status
  Ce registre permet de lire l'activité des différents canaux de DMA; on peut contrôler, par exemple, l'arrivée d'un signal TC.

Pour plus de précisions, nous renvoyons le lecteur à la documentation Intel.

Le 8237A accepte de se faire programmer par le processeur, tant que le HLDA est inactif. En d'autres termes, toute demande de transfert doit être masquée pendant cette période. On procède donc comme suit:

1) masquage des demandes (via le registre de masque) ou inhibition totale de contrôleur
(via le registre de commande)
2) initialisation des registres
3) démasquage des demandes

Le DMAC peut dès lors et à tout instant entrer dans sa phase active.

Les spécifications détaillées du DMAC se trouvent dans la documentation Intel.

Annexe 5: le Timer 8254

Pour supporter les fonctions de synchronisation primordiales dans l'unité d'acquisition, deux cadenceurs/compteurs (" Timer/counter") programmables sont utilisés

Présentation du "Timer/counter" 8254

Le 8254 est un cadenceur/compteur d'intevalles, de la famille Intel, prévu pour résoudre des problèmes communs à tout système devant gérer des applications en temps réel. Prenons comme exemple, la génération de délais précis sous contrôle software. Au lieu de mettre au point, par software, des boucles d'attente, le programmeur configure le 8254 de façon à rencontrer ses désirs et programme l'un des compteurs à sa disposition pour un délai désiré. Une fois ce délai écoulé, le 8254 interrompera le CPU. La prise en charge par software est donc minimale et les délais sont facilement accommodables aux volontés de l'application.
D'autres problèmes tels que la génération de rythmes ou la génération d'ondes carrées sont également résolus par ce timer.

Description:

Le Timer 8254 est composé de 3 compteurs indépendants.
Chaque compteur possède une entrée horlogique ("**clock**"), une entrée réservée à un signal déclencheur appelé "**gate**"et une sortie ("**out**") Ils génèrent des signaux représentant des délais, des rythmes ou des ondes carrées .Ces signaux sont de période réglable par le biais d'un registre de comptage.

Le fonctionnement d'un compteur

L'utilisateur peut choisir le mode du signal de sortie - délai, rythme ou ondes carrées - en initialisant le registre de commande. Il peut aussi programmer sa durée -ou sa période, pour le rythme et l'onde carrée- par le biais du registre de comptage; la valeur ainsi introduite multipliée par la période de l'horloge donne le délai généré par le compteur; ainsi si l'on introduit la valeur 1000 dans un compteur rythmé à 1 MHz cela correspond à un délai de 1 ms ; la même valeur dans un compteur battant à 1KHz donne un délai de 1s.

Les modes de fonctionnement:

mode 0:"**interrupt on terminal count**"
Lorsque la **gate** devient active, le signal **out** est activé pendant le délai programmé. Si la gate se désactive pendant un moment, **out** reste actif mais le décompte est suspendu .Il reprendra en même temps que la **gate**. A la fin , **out** est désactivé.

mode1:" **hardware retriggerable one-shot**"
Tout se passe comme en mode 0. De plus, à chaque impulsion de la **gate**, le décompte programmé recommence.

mode2:" **rate generator**"
Une fois ce mode programmé, le décompte reprend continuellement (sans impulsion de la gate).La fin de chaque délai est marquée par la désactivation de **out** le temps d'un cycle d'horloge.
Chaque fois que la gate est inactive, le décompte est interrompu dans les mêmes conditions que pour le mode 0 (avec reprise là où le décompte était resté).

mode3 : "**square wave generator**"
Ici, le signal **out** est actif durant la moitié du décompte, inactif l'autre moitié, et ceci sans interruption.

Pour de plus amples informations, notamment sur les mots de commande à placer dans les registres, nous renvoyons le lecteur à la documentation Intel.

Annexe 6: Les interfaces

Cette annexe est destinée à étudier la structure des entrées/sorties (E/S) du CPU .

Architecture.

Le CPU communique avec une ROM, une RAM et un boîtier d'interface. La ROM et la RAM sont groupées pour constituer la mémoire du système. Cette mémoire a des adresses valides comprises entre 00000H et FFFFFH. Le boîtier d'interface ne fait pas partie de cet espace mémoire.(figure a) Il a son propre espace d'adressage. Une opération d'E/S peut être définie comme suit:

-Entrée: le CPU lit les données provenant d'une source qui n'est pas la mémoire du système.

-Sortie: le CPU écrit les données vers une destination qui n'est pas la mémoire du système.

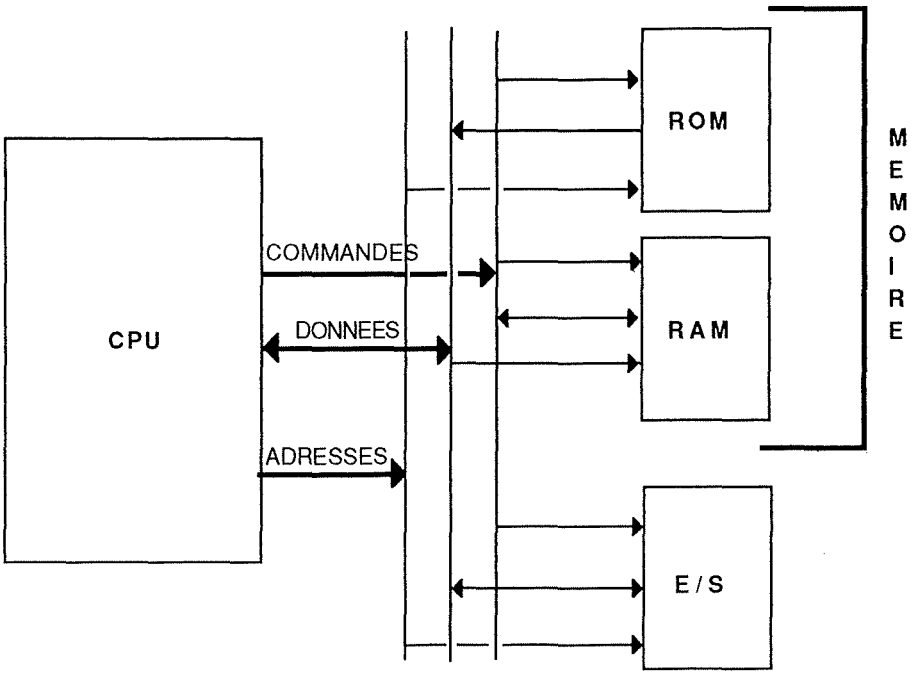


figure a.

Les commandes pour l'interface (E/S) doivent être distinguées des commandes pour la mémoire.En effet, bien que le système d'E/S et le système de mémoire soient complètement séparés, ils utilisent tous deux les mêmes lignes d'adresse. On peut donc avoir, par exemple, une adresse 00A2H mémoire et une adresse 00A2H d'E/S. C'est au niveau du bus des commandes que la distinction doit se faire: il y aura donc

des lignes de commande pour la mémoire et des lignes de commande pour l'interface (figure b).

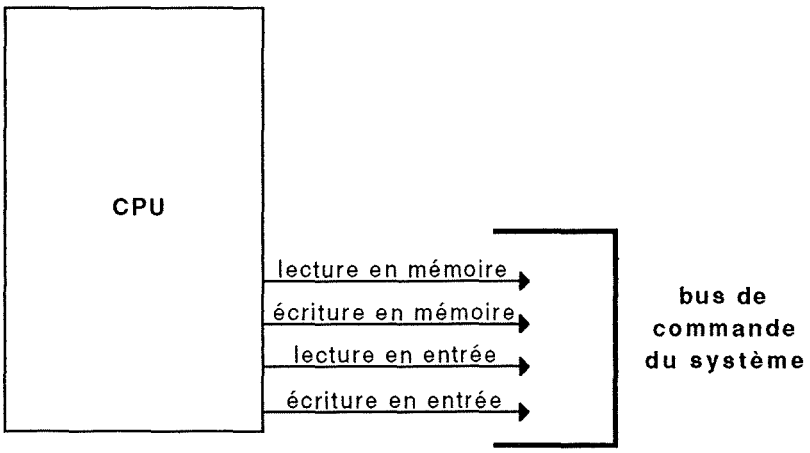


figure b

Dans un système classique, il existe plusieurs ports d'E/S. Nous entendons par "PORT" un emplacement spécial destiné à lire ou écrire des données et ne faisant pas partie de la mémoire du système.
A chaque port d'E/S est attribuée une adresse distincte, nommée code de sélection de port. Le CPU envoie une adresse sur le bus d'adressage, décodée par un circuit logique. Suivant la ligne de commande active, la valeur obtenue est une adresse en mémoire ou une adresse concernant un périphérique du CPU. (figure c).

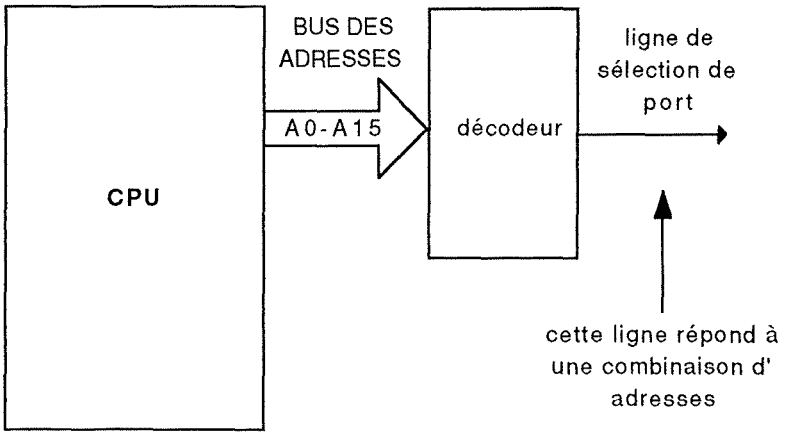


figure c.

Espace d'E/S réservé sur Personal Computer

Les Ports du PC

Circuit	PC/AT
Contrôleur de DMA (8237A)	**000-01F**
Contrôleur d'interruption	020-03F
Temporisateur	040-05F
Communication périphérique programmable	néant
Clavier	060-06F
Horloge en temps réel	070-07F
Registre de DMA	**080-09F**
Contrôleur d' interruption n°2	0A0-0BF
Contrôleur de DMA n°2	**0C0-0DF**
Coprocesseur mathématique	0F0-0F1
Coprocesseur mathématique	0F8-0FF
Contrôleur de disque dur	1F0-1F8
Adaptateur de jeux	200-207
Unité d' extension	néant
Connexion pour la 2nde imprimante parallèle	278-27F
Seconde connexion sérielle	2F8-2FF
Carte de prototype	**300-31F**
Carte de réseau	360-36F
Connexion pour la 1ère imprimante parallèle	378-37F
Carte d' écran monochrome et connexion d' imprimante parallèle	3B0-3BF
Catre d' écran couleur/graphique	3D0-3DF
Contrôleur de disquette	3F0-3F7
Première connexion sérielle	3F8-3FF

Nous allons maintenant dresser une liste des E/S qui nous intéressent dans le cadre de ce mémoire.

Les Ports du DMA SUR PC/AT

Circuit	adresses
Contrôleur de DMA (8237A)	**000-01F**
Registres d' adresse de base	
- canal 0	00
- canal 1	04
- canal 2	08
- canal 3	0C
Registres ce comptage	
- canal 0	02
- canal 1	06
- canal 2	0A
- canal 3	0E
Registre de commande	10
Registre de masquage	14
Registre de mode	16
Registre de page de DMA	**080-09F**
- canal 0	87
- canal 1	83
- canal 2	81
- canal 3	82
- canal 4	01
- canal 5	03
- canal 6	05
- canal 7	07
Contrôleur de DMA n°2	**0C0-0DF**
Registres d' adresse de base	
- canal 4	C0
- canal 5	C4
- canal 6	C8
- canal 7	CC
Registres ce comptage	
- canal 4	C2
- canal 5	C6
- canal 6	CA
- canal 7	CE
Registre de commande	D0
Registre de masquage	D4
Registre de mode	D6

Les Ports de la Carte d' acquisition

commande Start/Stop	300
ADC	301
Reset ADC	303
timer1	304-307
- registre de commande	304
- compteur 0	305
- compteur 1	306
- compteur 2	307
timer2	308-30B
- registre de commande	308
- compteur 0	309
- compteur 1	30B

Annexe7

BARRE DE MENU PRINCIPAL

FICHIER	PATIENT	ACQUISITION	TRAITEMENT

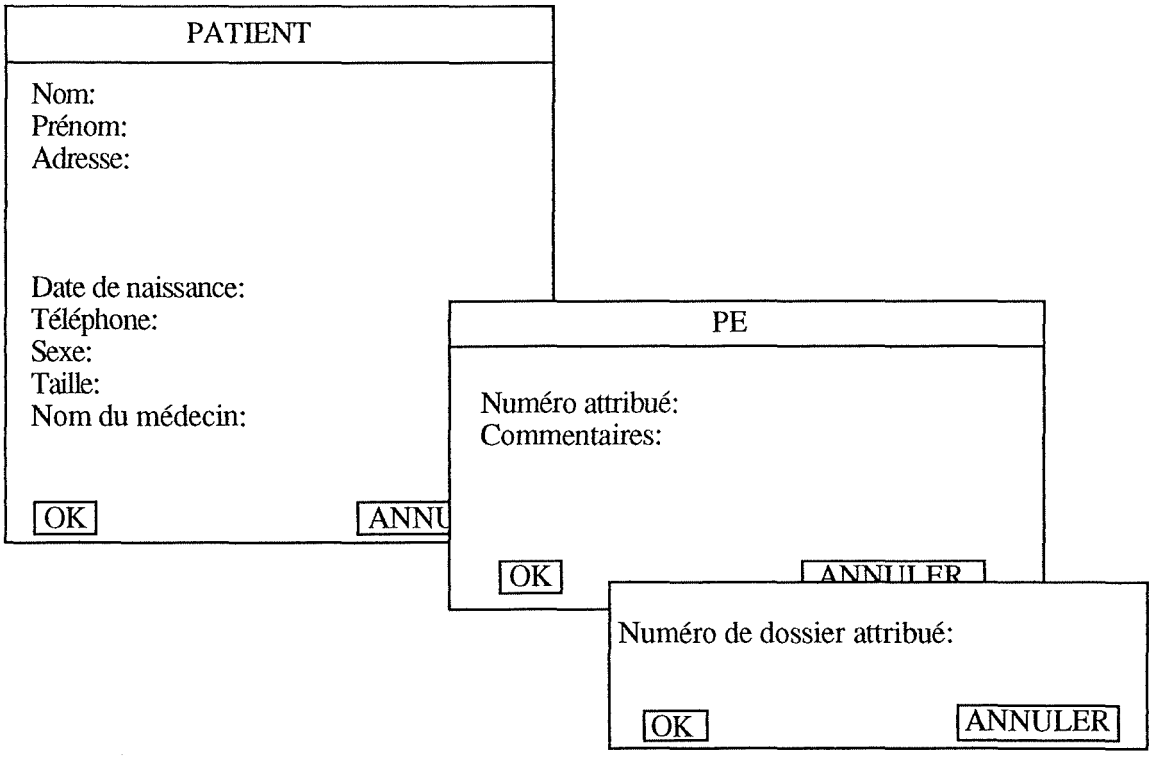
MENUS DEROULANTS

FICHIER	PATIENT	ACQUISITION	TRAITEMENT
CHARGER ENREGISTRER IMPRIMER ANNULER COMMENTAIRES ARCHIVER SYSTEME QUITTER ALTX	CONSULTER MODIFIER	PEV PEAP PEAT PES P300	LISAGE FILTRAGE ZOOM AIRE AMPL.REL. AMPL.ABS. LATENCE SUPERPOSITION GAIN

FENETRE CHARGER

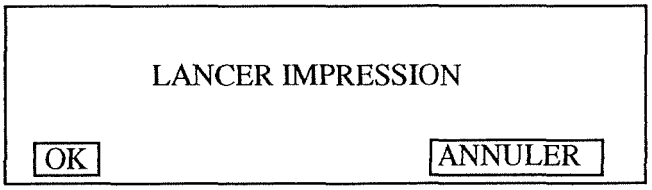
CHARGEMENT PE	
Numéro de PE:	
OK	ANNULER

FENETRES SUCCESSIVES DE ENREGISTRER



Il y a deux sous-choix possibles dans 'IMPRIMER' :
- Imprimer le spectre qui est à l'écran
- Imprimer une liste de spectres

IMPRESSION DU SPECTRE A L'ECRAN



IMPRESSION D'UNE LISTE DE PE

NUMERO DE PE POUR IMPRESSION
(MAX 10)

1)
2)
3)
4)
5)
6)
7)
8)
9)
10)

OK ANNULER

FENETRE ANNULER

ANNULER L'ACQUISITION

OK ANNULER

FENETRE ARCHIVAGE

ARCHIVAGE

Date dernier archivage:
Date nouvel archivage:

OK ANNULER

FENETRE PEV

PARAMETRES PEV

Date:
Etat du malade:
Stimulation
.Oculaire:
.Type:
.Nombre:
.Récurrence:

Filtre
.FH:
.FB:

Electrodes
.Canal1:
.Canal2:

Fenêtre d'analyse:
Seuil d'artéfact:

OK ANNULER

Le langage utilisé pour réaliser les spécifications fonctionnelles possède quelques spécificités. Nous tâcherons dans cette annexe d'en expliquer les grandes lignes. Il possède des types de données prédéfinis (BOOL, CHAR,INTEGER, STRING, DATE) pour lesquels il fournit des constructeurs de types. Nous définirons également la notion importante d'état permettant de conserver des informations entre deux exécutions de fonctions.

1. *Les constructeurs de types*

Nous ne détaillerons que ceux que nous utiliserons.

1.1. La séquence

Une séquence est une suite d'éléments de même type qui se trouvent dans un ordre précis.

On peut réaliser différentes opérations simples sur une séquence :

- construction d'une séquence : s = [0, 1, 2, 3]
- test d'appartenance d'un élément à une séquence : (2 ∈ s) renvoie TRUE
- test d'une séquence vide : Empty(s) renvoie FALSE
- longueur d'une séquence : Length(s) = 3
- obtention du premier élément d'une séquence : First(s) = 0
- obtention du dernier élément d'une séquence : Last(s) = 3
- extraction du début d'une liste : Head(s) = [0,1, 2]
- extraction de la fin d'une liste : Tail(s) = [1,2, 3]
- concaténation de deux séquences grâce à l'opérateur +
- accès au $i^{ème}$ élément d'une séquence : $s_1 = 0$
- ajout d'un élément à la fin d'une séquence : Append(s, 4) = [0, 1, 2, 3, 4]

Ce ne sont quelques exemples parmi d'autres (ajout d'un élément à une position précise de la séquence,...)

1.2. L'ensemble

L'ensemble réunit des données de même type pour lesquelles l'ordre n'a pas d'importance.

On peut réaliser sur ces ensembles un certain nombre d'opérations. En voici quelques unes :

- construire un ensemble : e = {0, 1, 2}

- réaliser des opérations classiques sur les ensembles : concaténation, soustraction, réunion, intersection,...
- tester l'appartenance d'un élément à un ensemble : (3 ∈ e) qui donne la valeur FALSE
- tester un ensemble pour voir s'il est vide : Empty(e) renvoit FALSE
- obtenir le nombre d'éléments d'un ensemble : Card(e) = 3
- ajout d'un élément à un ensemble : Add(e, 3) = {0, 1, 2, 3}
- la suppression d'un élément d'un ensemble : Remove(e, 1) = {0, 2, 3}

1.3. Le produit cartésien PC

Ce concept permet la définition de t-uples d'éléments de types différents.Les opérations permises sur les produits cartésiens sont les suivantes :

– création d'un objet du type : pc = <"Dupond", 56, "rue de la gare">
– accès à un des champs du t-uple : numéro(pc) = "56"
– comparaison de deux t-uples avec les opérateurs = et ≠
– modification d'un champ du t-uple : pc' = (pc, numéro→103)

2. *Les états*

Suivant le schéma Entité/Association on pourra appliquer un certain nombre de requêtes. Par exemple :

- In(Patient(id)) pour tester l'existence d'une occurrence du type d'entité Patient identifié par id
- In(Prescription(id_pat, id_pe)) teste l'existence d'une association Prescription entre l'occurrence du type d'entité **patient** identifiée par id_pat et l'occurrence du type d'entité **PE** identifiée par id_pe.

....

Bibliographie

[Benoi88] BENOIT M., MICHEL B., PC, XT et AT : *maintenance et améliorations*, BCM, 1988.

[BIBLE] *La bible du PC.*

[Bodar89] BODART F., *Conception assistée des systèmes d'information*, Masson, 1989.

[Borla] BORLAND, *Turbo C*

[Chiap83] CHIAPPA K.H., *Evoked potentials in clinical neurology*, NewYork, Raven Press, 1983.

[Coffr84] COFFRON J.W., *Programmation du 8086-8088*, Sybex, 1984

[Courj84] COURJON J., *Apport des potentiels évoqués visuels (PEV) à la neurologie*, Rev. E.E.G., Neurophysiol.

[Desme74] DESMEDT J.E., BRUNKO E., DEBECKER J., CARMELIET J., *The system bandpass requiered to avoid distorsion of early components when averaging somatosensory evoked potentials*, Electroencephalography and Clinical Neurophysiology.

[Gueri88] GUERIT J.M., MAYER M., *L'enregistrement et interprétation des PE : principes élémentaires*, Cliniques St Luc-Bruxelles, Hôpital St Vincent-de-Paul, Paris, 1988

[Gueri91] GUERIT J.M., Premier chapitre , réf.inconnue(sortie en 91)

[Haina86] HAINAUT J.-L., *Conception assistée des applications informatiques : 2. Conception de la base de données*, Masson, 1986.

[Intel90] INTEL, *Peripherals.*

[Lilen] LILEN H., *8088 et ses périphériques- Les circuits clés des IBM PC et compatibles*, Ed. Radio.

[Maugu] MAUGUIERE F., FISCHER C., *Les potentiels évoqués en neurologie*, Editions Techniques-Encyclo. Med. Chir. (Paris-France), Neurologie, 17031B10, 4-1990, 26p.

[Soveg88] SOVEGES L., *L'électroencéphalographie et les potentiels évoqués : les chaînes de mesures*, Note de cours UCL St LUC, 1988.

[Tubel84] TUBELLO P.,*Acquisition et traitement des potentiels évoqués*, Mémoire à l'institut supérieur industriel du Hainaut, Département Electronique, Année académique 1983-1984.

Réalisation, impression et reliure : SPRL Centre de Reprographie - Tél. : 081/22.87.72