



UNIVERSITÉ TOULOUSE III - PAUL SABATIER
M2 IM - MASTER INFORMATIQUE IMAGE ET MULTIMÉDIA

Rapport de "Conception détaillée"

Débruitage Doppler

Équipe :

Benjamin AZZINI
Lucien MAHOT
Aline LEPAILLEUR
Kevin LEPAN

Client et superviseur :

M. Denis KOUAMÉ

3 Janvier 2016

Table des matières

1	Introduction	2
2	Vue d'ensemble détaillée du système	3
2.1	Diagramme de cas d'utilisation	3
2.2	Diagramme d'activité	4
3	Structures de données et décomposition en classe	5
3.1	Structure de donnée	5
3.2	Diagramme de classe	5
4	Tests unitaires	7
5	Planning prévisionnel et analyse de risque	9

1. Introduction

Ce troisième rapport entre dans le cadre du chef d'œuvre du Master Informatique Image et Multimédia. Notre sujet concerne le développement d'une application permettant de débruiter le signal correspondant au flux sanguin capté par un système d'acquisition basé sur l'effet Doppler.

Le rapport de « Conception détaillée » a pour objectif de décrire comment nous allons procéder pour résoudre le problème d'un point de vue architecture logicielle. Nous présenterons en détails les modules cités dans notre précédent rapport de « Spécifications » avec différents diagrammes, ainsi que les tests permettant de vérifier le bon fonctionnement de chaque modules. Nous montrerons notre planning prévisionnel et les risques mis à jour d'après notre dernier rapport (« Spécifications »).

2. Vue d'ensemble détaillée du système

2.1 Diagramme de cas d'utilisation

Comme nous l'avons vu au rapport de spécification, nous avons à notre disposition 6 modules bien distincts plus ou moins denses en quantité de travail. Les modules de récupération du son et de filtrage sont les modules piliers de notre application qui donnent accès aux modules restants. Les autres modules permettent à l'utilisateur d'accéder directement aux fonctionnalités de l'application.

Le diagramme de cas d'utilisation résume tous les cas exprimant l'interaction entre le système et les acteurs. Dans notre cas l'acteur qui va interagir avec le système est unique et est représenté par la femme enceinte utilisant notre application. Les différents cas d'utilisation sont les suivants :

- Ecouter les battements du fœtus
- Obtenir le rythme cardiaque du fœtus
- Visualiser un spectre explicite
- Sauvegarder le son net du battement de cœur

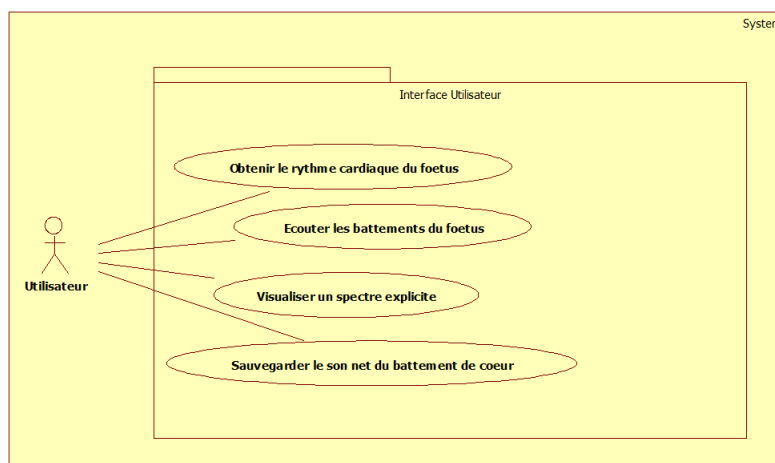


FIGURE 2.1 – Diagramme de cas d'utilisation

2.2 Diagramme d'activité

Dans les quelques cas d'utilisations vus précédemment, certains possèdent des actions en commun comme le module de filtrage. Afin d'obtenir une sauvegarde d'un son sans bruit et le rythme cardiaque du fœtus, il est obligatoire de passer par un filtrage du son initial. Le diagramme d'activité suivant reprend en détails le cas d'utilisation "Sauvegarder le son net du battement du coeur du fœtus" :

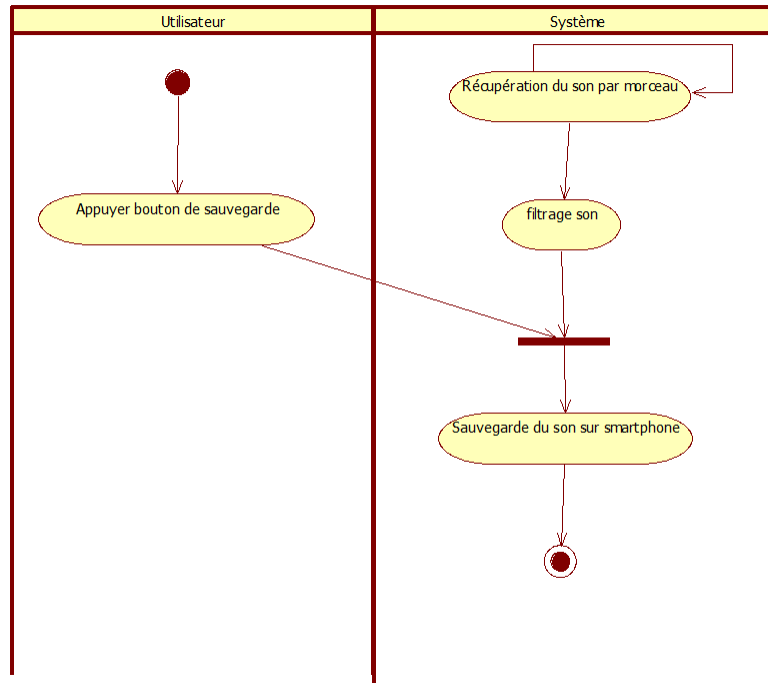


FIGURE 2.2 – Diagramme d'activité

3. Structures de données et décomposition en classe

3.1 Structure de donnée

La principale structure de donnée qui sera manipulée dans notre projet est un vecteur de float représentant notre signal 1D, chaque module du système traite ce vecteur, que ce soit pour la partie filtrage ou affichage.

3.2 Diagramme de classe

Le diagramme de classe suivant nous permet de visualiser l'ensemble du système plus précisément ainsi que les communications entre celles-ci.

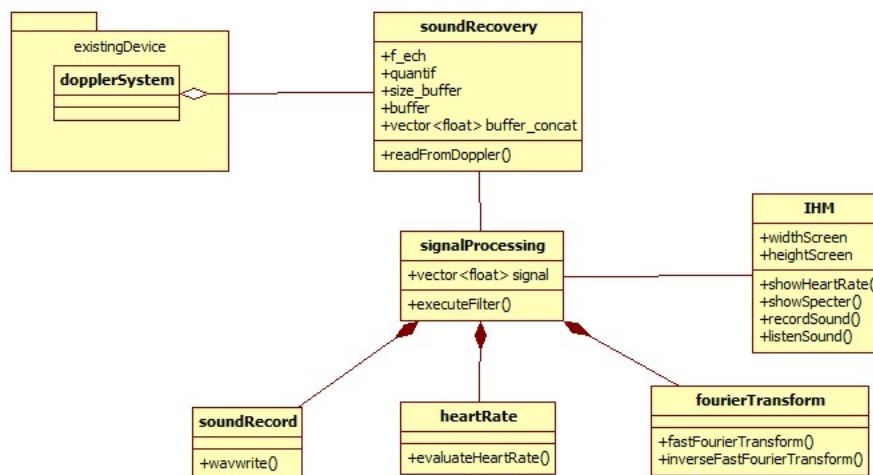


FIGURE 3.1 – Diagramme de classe

1. Classe ExistingDevice

Réprésente le système existant mis à notre disposition, à savoir l'appareil nous permettant d'écouter le flux sanguin.

2. Classe SoundRecovery

Va nous permettre de récupérer le son de l'appareil de manière continue en utilisant un buffer, une fois récupéré, le buffer est concaténé pour former le signal que l'utilisateur récupère pour pouvoir passer dans la chaîne de traitement.

3. Classe SignalProcessing

Cette classe contient le vecteur représentant le signal reçu, une fois le filtre appliqué sur ce dernier il pourra être traité convenablement par les classes internes à SignalProcessing

4. Classe SoundRecord

Cette classe permet d'enregistrer la session d'écoute courante au format .wav sur le smartphone de l'utilisateur

5. Classe FourierTransform

Cette classe a pour but de calculer la transformer de Fourier ainsi que l'inverse du signal, nous permettant donc d'afficher à l'utilisateur le spectre du signal écouté à l'écran pour la classe IHM.

6. Classe HeartRate

Le rythme cardiaque foetal est calculé par le biais de cette classe, à partir du signal débruité.

7. Classe IHM

Concerne l'application elle-même, l'interface graphique qui sera présentée à l'utilisateur lui permettant de visualiser le spectre du signal écouté, calculé grâce à la classe FourierTransform. Mais aussi le rythme cardiaque foetal par le biais de la classe HeartRate ainsi que la possibilité d'enregistrer la session tout en écoutant le signal. Les attributs de cette classe représentent la dimension de l'écran pour un affichage dépendant de cette dimension.

4. Tests unitaires

On peut tester et valider séparément chaque module. En effet, le schéma modulaire de notre application permet de tester les résultats des différents modules. Cependant, la vérification de l'exécution temporelle est nécessaire et plus complexe.

1. Filtrage/debruitage

La validation du filtrage pourra s'effectuer en comparant nos résultats issus de l'application avec ceux produits par matlab. De plus on pourra soustraire le son filtré au son original pour ne récupérer que le bruit et se rendre compte du bon fonctionnement du filtrage.

2. Récupération du son

Pour tester la bonne récupération on pourra mettre en oeuvre une fonction de lecture audio des données ou simplement vérifier leur cohérence.

3. Enregistrement

Pour tester le bon enregistrement du fichier, on pourra le lire avec le lecteur audio par défaut du téléphone Android. Si le formatage des données est erroné le fichier ne se lira pas. On pourra comparer à l'oreille que le fichier enregistré est correct pour vérifier la cohérence des données.

4. Fft

On pourra comparer les résultats après fft/iff t entre notre application et matlab sur les mêmes échantillons. On pourra aussi vérifier la cohérence des données.

5. Rythme cardiaque

La vérification du rythme cardiaque est impossible. En effet (et comme cela est précisé dans l'article[1]) il faudrait pouvoir comparer nos résultats avec ceux d'un matériel medical professionnel utilisé sur le même sujet et en même temps que notre appareil. On peut cependant vérifier que les données sont cohérentes : rythme de base entre 110 et 160 bpm en moyenne.

6. Affichage

Les tests de l'affichage seront essentiellement visuels. L'affichage doit être fluide et agréable. L'affichage du spectre doit être calé sur le son pour paraître cohérent.

5. Planning prévisionnel et analyse de risque

Au vu de l'avancé du projet, voici le diagramme de Gantt mis à jour. Et concernant l'analyse des risques, nous n'avons pas remarqué de nouveaux risques potentiels que ceux déjà énoncés dans le rapport de spécification.

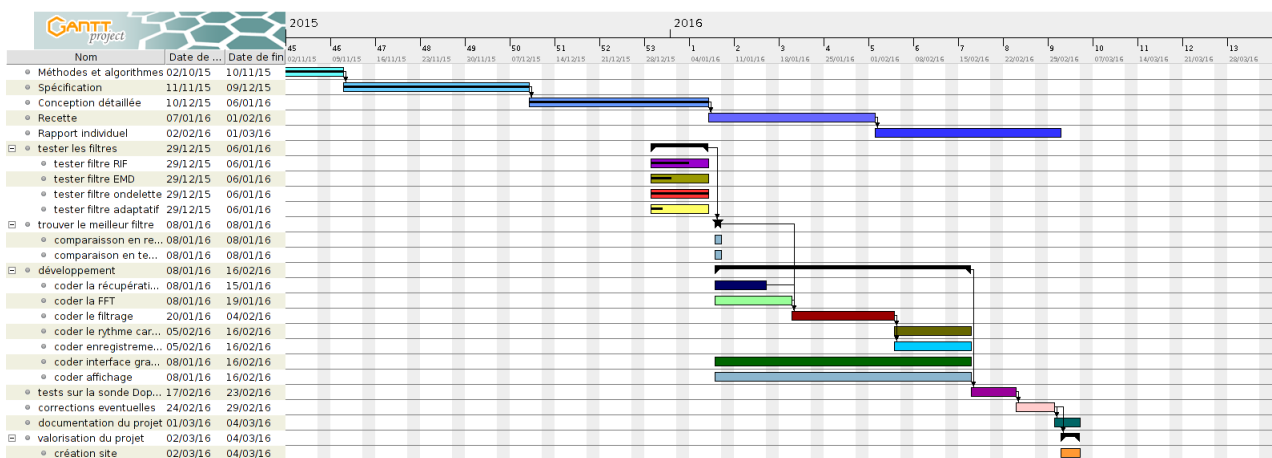


FIGURE 5.1 – Planning mis à jour

Bibliographie

- [1] Iulian Voicu, Jean-Marc Girault, Catherine Roussel, Aliette Decock, and Denis Kouame. Robust estimation of fetal heart rate from us doppler signals. *Physics Procedia*, 3(1) :691–699, 2010.