Rapport d'avancement

BMONS

Beehive Monitoring System

Rédigé par :

Alice Danckaers
Benoît Raymond
Etiene Dalcol
Nicolas Van-Nhân Nguyen
Tao Zheng
Armand Sellier

Sous la direction de :

Olivier Reynet

Option Systèmes Perception Information Décision

Première impression, décembre 2014

Sommaire

Première partie

Introduction au projet

Chapitre 1

Formulation initiale du projet

1.1 Contexte

BeeHive Monitoring System (BMONS) est un projet qui a pour but d'aider les apiculteurs. Il s'agit de leur proposer un système de surveillance et de détection peu onéreux afin de prodiguer les meilleurs soins au meilleur moment aux ruches qui en ont besoin et d'éviter les vols.

En effet, les abeilles sont vitales à l'équilibre écologique. Einstein avait même dit : " Si l'abeille disparaît, l'humanité en a pour quatre ans à vivre ". Sans elles 84 % des espèces végétales cultivées pour l'alimentation disparaitraient. Or les abeilles sauvages sont aujourd'hui rares et l'espèce ne survivrai pas sans l'aide des apiculteurs. Ainsi le travail de ces derniers est crucial non seulement pour assurer la production de miel mais aussi pour la sauvegarde de l'environnement. Cependant, ces dernières années, les apiculteurs ont été confrontés à de nombreux problèmes et nous sommes aujourd'hui face à une diminution du nombre d'abeilles telles que la production annuelle européenne de miel est quatre fois moindre que celle qu'il y a vingt ans.

Pour aider à la résolution de ce problème, nous voulons donc créer un système capable d'aider l'apiculteur dans son travail et de ce fait combattre la disparition des abeilles.

1.2 Expression initiale du besoin

Après avoir discuté avec plusieurs apiculteurs, nous avons pu identifier leurs besoins et déterminer de quelle manière nous pouvons les aider. Ainsi l'objectif de ce système est tout d'abord de donner accès à l'apiculteur à des informations clés sur la ruche sans que celui-ci n'ai à se déplacer, ni à ouvrir les ruches. En effet l'ouverture de la ruche perturbe les abeilles et elle n'est pas possible en hiver à cause des températures trop basses. De plus les ruches sont souvent disposées dans des ruchers éloignés les uns des autres, ce qui complique le travail de l'apiculteur. Les informations nécessaires seraient : la température dans et en dehors de la ruche, le poids, l'humidité et les sons de la ruche. Mais le système devra aussi alerter l'apiculteur quand la sécurité de la ruche est compromise, pour permettre une action rapide destinée à sauver la colonie.

Le système BMONS est donc composé de deux parties distinctes. La première consiste en un élément embarqué dans la ruche qui consomme un minimum d'énergie et qui mesure les paramètres clés. Les données de cet élément embarqué sont transmises via un transmetteur sans fils à un serveur qui constitue la deuxième partie du système. Il donne accès à l'apiculteur aux différentes mesures effectuées dans et autour des ruches. Il envoie également des alertes de sécurités à l'apiculteur si besoin.

Chapitre 2

État de l'art

En effectuant nos recherches sur le sujet nous avons trouvé beaucoup d'informations sur les abeilles et le travail des apiculteurs en général, ainsi que des systèmes "maison" développés par des particuliers pour surveiller un peu mieux leurs ruches. Cependant nous avons également découvert l'existence de quatre projets similaires au notre : trois projets en cours ayant une approche OpenSource et un projet commercial déjà développé. Ce dernier appartient à la société anglaise Arnia. Ce système est décrit [2] comme permettant à l'utilisateur d'avoir des informations sur une ou plusieurs ruches telles que la température, l'humidité et l'intensité acoustique dans la ruche ainsi que la température du couvain. Les apiculteurs peuvent ensuite visualiser ces informations sur une partie sécurisée du site internet d'arnia. Ils peuvent également comparer les informations et évolution d'une ou plusieurs ruches, comme on peut le voir sur la figure 2.1.

FIGURE 2.1 – Interface du système d'arnia : comparaison des données d'une ruche

L'un des projets OpenSource est développé par Ken Meyer sur le site hackaday [8] et consiste à mesurer la température, l'humidité et le poids d'une ruche. Ce projet est encore en développement et plusieurs prototypes ont déjà été testés.

Il existe également un autre projet OpenSource sur le sujet. Il s'agit de Bzzz [7], développé par le Fablab de Lannion. Ce système propose une supervision de la température intérieure, de la luminosité extérieure et la masse d'une seule ruche via un envoi de données périodique par SMS et par visualisation des données sur

un portail en ligne. L'utilisateur pourra également configurer des alertes via le portail.

Enfin le dernier système existant que nous avons trouvé a été développé conjointement par le Fablab de Barcelone et Open Tech Collaborative, Denver, USA [3]. Ce projet OpenSource, appelé Open Source BeeHive, ne s'adapte pas aux ruches classiques mais propose une architecture simple qui permet de construire sa propre ruche entièrement, comme on peut le voir sur les figures 2.3 et 2.2.

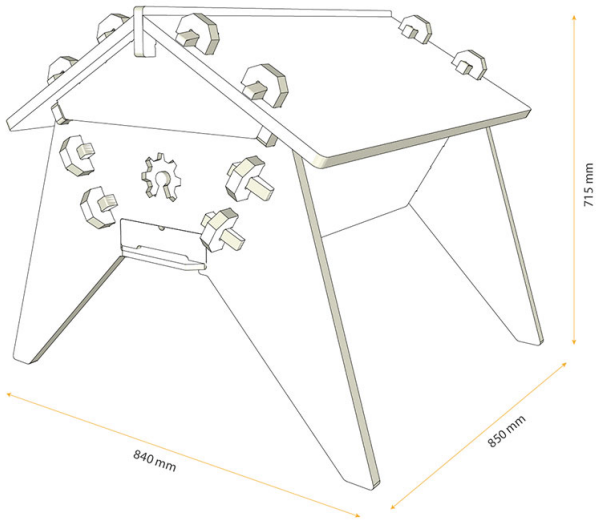

FIGURE 2.2 – Modèle de ruche Open Source Beehive

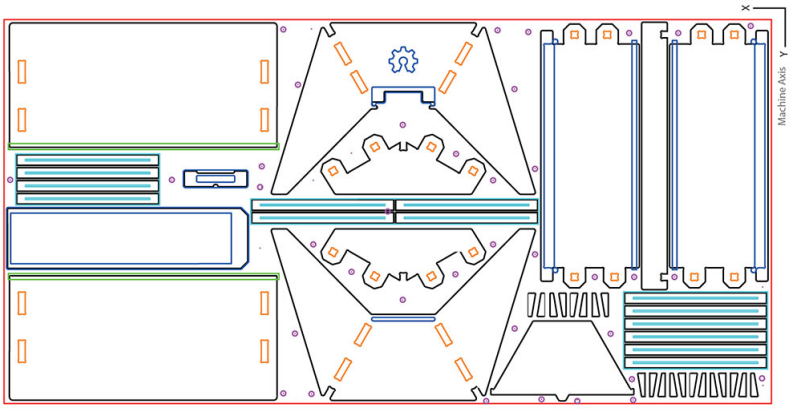

FIGURE 2.3 – Plan de la ruche Open Source Beehive

Ensuite un kit de capteurs à installer permet de mesurer la température, l'humidité, l'intensité acoustique et le nombre d'abeilles via un capteur infrarouge. Les données seront ensuite visibles de tous sur la plateforme Smart Citizen.

Nous n'avons pas détaillé ici tous les projets que nous avons trouvés du fait de leur grand nombre. Cependant nous nous sommes intéressés à ceux qui présentaient un intérêt pour le système que nous voulons développer.

Concernant le traitement sonore, le site internet Beesource comporte une description complète du système Apidictor [1]. Ce système permet de filtrer le son émit par une ruche pour prévoir un essaimage. Un schéma du montage électrique ainsi qu'un texte expliquant quelles fréquences sont surveillées et quelles méthodes ont été utilisées pour vérifier le bon fonctionnement de ce système sont également présents. Nous ne réutiliserons pas le montage proposé, car le traitement du signal sonore se fera de manière informatisée, en revanche le travail effectué pour savoir quelles fréquences sont à surveiller nous sera utile.

2.1 Choix des capteurs

Après avoir réalisé l'état de l'art pour notre système de surveillance d'une ruche, nous nous sommes ensuite intéressés aux capteurs que nous allons employer.

Capteur de température

Nous avons choisis un capteur de température identique pour recueillir la température interne et externe de la ruche. Il s'agit d'une thermistance NTC boitier Goutte Radial 1000 ohms sortie fil de cuivre émaillé 1 pc(s). Ce dernier possède une gamme de mesure comprise entre -40 C et 100 C ce qui correspond bien aux exigences discutées avec le client (Voir tableau des exigences).

Capteur d'humidité

On a choisit d'inclure ce capteur dans notre système compte tenu des résultats de l'Etat de l'art. Néanmoins, après discussion avec le client, cette option n'est pas primordiale pour un apiculteur.

Capteur de pression Les capteurs de pression vont nous permettre de récupérer le poids de ruche et surtout celui des hausses pour avertir l'apiculteur de la quantité de miel produite. Pour se faire, le projet Bzzz développé par le Fablab de Lannion a prévu d'utiliser deux sachets de Pompote remplis d'eau sucrée pour éviter les variations de pression atmosphérique, le gel et l'évaporation. Cependant, après avoir pris conscience de l'importance de la localisation de la grappe, nous avons pensé installer deux capteurs de pression par cadre (au niveau de chaque extrémité) soit 20 au total mais cette solution s'est avérée être difficile à mettre en place à cause de la surface sur laquelle repose les cadres (simple lamelle en métal). Après discussion avec le client, nous avons finalement opté pour la confection d'un cadre en bois aux même dimensions de la ruche dans lequel se trouverons les capteurs de pression. L'utilisateur décidera de l'endroit où le placer en fonction des données qu'il veut récupérer.

Tilt sensor

Ce capteur

Microphone

Deuxième partie

Dossier fonctionnel

Chapitre 3

Ingénierie des exigences

3.1 Approche Top-Down

Dans cette partie nous allons analyser notre système avec une approche Top-Down. Cela signifie que nous adopterons une démarche de conception descendante. Pour cela nous avons tracé le diagramme "bête à cornes", que l'on peut voir sur la figure 3.1. Il permet de représenter graphiquement l'expression du besoin. Comme on peut le voir sur le diagramme, le système BMONS rend service aux apiculteurs en agissant sur une ou plusieurs ruches. Il a pour but d'aider la surveillance d'un rucher et d'avertir l'apiculteur en cas de problème.

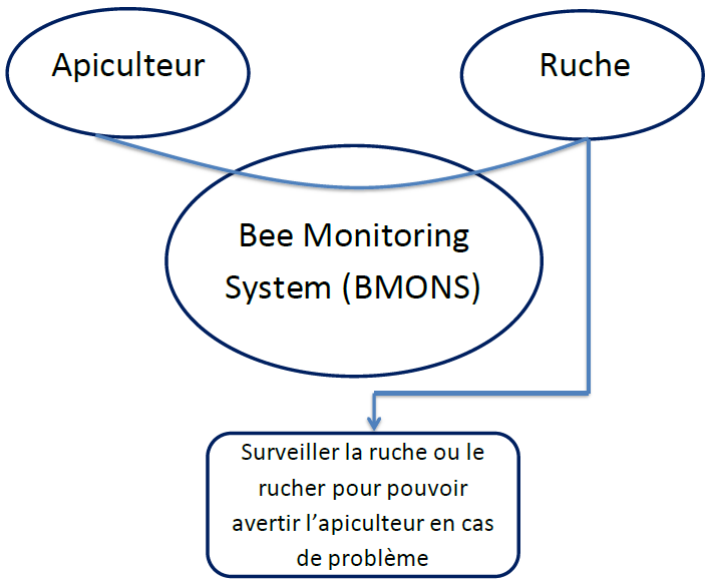

FIGURE 3.1 – Diagramme "bête à cornes" du système BMONS

Le diagramme pieuvre, 3.2 et 3.3, nous permet ensuite de faire apparaître les fonctions principales du système. On peut aussi y retrouver les fonctions de

services et de contraintes.

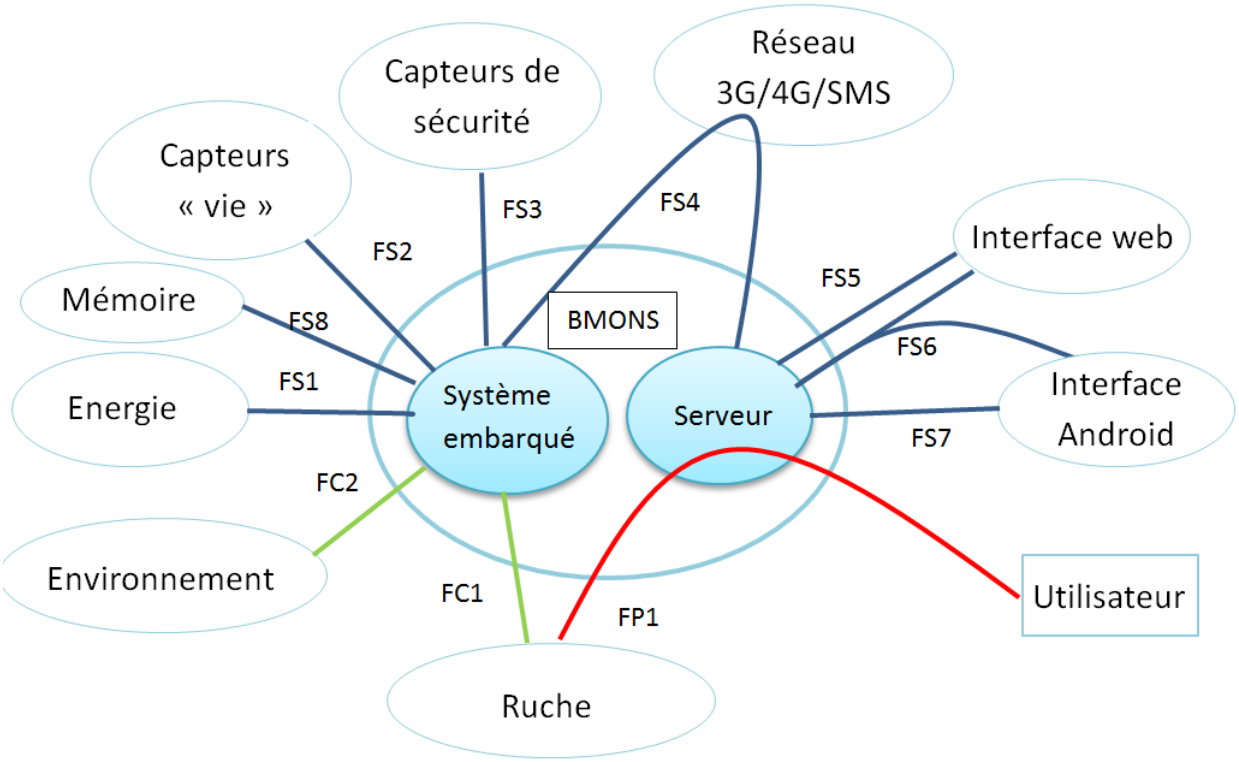

FIGURE 3.2 – Diagramme pieuvre du système BMONS

FP1 : Informer l'apiculteur de l'état de la ruche

FC1 : Etre adapté aux dimensions de la ruche
FC2 : Résister à l'environnement de la ruche

FS1 : Etre autonome en énergie
FS2 : Acquérir les données relatives à la vie de la ruche
FS3 : Acquérir les données relatives à l'intégrité de la ruche
FS4 : Communiquer avec le serveur via le réseau 3G/4G/SMS
FS5 : Proposer une IHM web
FS6 : Proposer un accès via l'API
FS7 : Proposer une IHM Android
FS8 : Conserver les données localement

FIGURE 3.3 – Légende diagramme pieuvre du système BMONS

3.2 Approche Bottom-Up

Nous allons maintenant adopter la démarche inverse, mais néanmoins complémentaire, de l'approche Top-Down. Il s'agit de l'approche Bottom-Up. C'est une démarche de conception ascendante qui va nous permettre d'avoir une vision plus globale du système. On peut voir sur 3.4 et 3.5 les exigences issues de cette analyse.

Identifiant	Type	Expression de l'exigence	Performance	Test	Groupe logique	Fonction
exi_01	contrainte	Supporter les variations de température	de -15 à 60°C		GLE_Capteur_info	Résister au milieu ambiant de la ruche
exi_02	contrainte	Ne pas être endommagé par la cire et la propolis		Plonger dans le miel		
exi_03	contrainte	Ne pas être endommagé par les abeilles		Mise en situation		
exi_04	contrainte	Ne pas être endommagé par l'apiculteur lors de ses interventions sur la ruche				
exi_05	contrainte	Supporter les variations d'humidité	de 0 à 100 %			
exi_06	contrainte	Ne pas nuire aux abeilles				Perturber au minimum
exi_07	contrainte	Etre adapté aux dimensions		implantation dans un prototype		
exi_08	contrainte	Respecter la reglementation alimentaire				Adapter à la legislation
exi_09	service	Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 2/jour			Mesurer les paramètres propres relatifs au fonctionnement de la ruche
exi_10	service	Relever le bruit dans la ruche				
exi_11	service	Mesurer la température dans la ruche	[-5 ; 50] résolution 1°C fréquence 1/heure			
exi_12	service	Mesurer l'humidité dans la ruche	[0 ; 100] résolution 1% fréquence 1/heure			
exi_13	service	Vérifier la présence d'abeilles	fréquence 1/heure			
exi_14	service	Relever la température extérieure	[-15 ; 50] résolution 1°C fréquence 1/heure			
exi_15	service	Détecter un choc			GLE_Capteur_sécu	Mesurer les paramètres liés à la sécurité et à l'intégrité de la ruche
exi_16	service	Détecter l'ouverture de la ruche				
exi_17	contrainte	Supporter les variations de température	[-15 ; 50] (° C)			Résister au milieu extérieur
exi_18	contrainte	Supporter les variations d'humidité	[0 - 100] %			
exi_19	contrainte	Résister aux chocs				
exi_20	contrainte	Etre étanche		plonger dans l' eau		

FIGURE 3.4 – Exigences issues de l'approche Bottom-Up (1/2)

Identifiant	Type	Expression de l'exigence	Performance	Test	Groupe logique	Fonction
exi_21	service	Transmettre les données par ondes radio	3G ou 4G		GLE_Boitier_Exterieur	Gérer les données
exi_22	contrainte	pouvoir stocker les données localement				
exi_23	contrainte	Etre relié aux capteurs extérieurs				Récupérer les données
exi_24	contrainte	Etre relié aux capteurs intérieurs				
exi_25	contrainte	Supporter les variations de température	[-15 ; 50] (° C)			Résister au milieu extérieur
exi_26	contrainte	Supporter les variations d'humidité	[0 - 100] %			
exi_27	contrainte	Résister aux chocs				
exi_28	contrainte	Etre étanche		plonger dans l' eau		
exi_29	service	Avoir 2 modes de fonctionnement (été, hivers)				S'adapter au besoins de l'apiculteur
exi_30	contrainte	Déclancher des mesures à intervalles réguliers				
exi_31	contrainte	être autonome en énergie				S'alimenter en énergie
exi_32	service	Avertir l'apiculteur en cas de diminution significative du poids de la ruche			GLE_software	Informer l'apiculteur
exi_33	service	Avertir l'apiculteur en cas de choc				
exi_34	service	Avertir l'apiculteur en cas de détection d'essaimage				
exi_35	service	Afficher les données (graphique)				
exi_36	service	Informer l'apiculteur sur l'alimentation électrique et l'état des capteurs				
exi_37	service	Donner la possibilité aux utilisateurs de configurer les alertes				
exi_38	service	Donner la possibilité aux utilisateurs de configurer les informations à afficher				
exi_39	service	Donner la possibilité aux utilisateurs de configurer le niveau de confidentialité				
exi_40	service	Pouvoir accéder rapidement à l'information				
exi_41	service	Déterminer la position de la grappe				Traiter les donnees reçues
exi_42	service	Reconnaitre le chant des futurs reines				
exi_43	service	Identifier les prémices de l'essaimage				
exi_44	service	Identifier les bruits caractéristiques dans la ruche				
exi_45	contrainte	Restreindre l'accès aux données				Gérer les données
exi_46	service	Extraire les données				
exi_47	service	Conserver un historique des données				

FIGURE 3.5 – Exigences issues de l'approche Bottom-Up (2/2)

3.3 Fonctions métiers du système

Chapitre 4

Spécification fonctionnelle 3 axes

4.1 Raffinement FAST

Le diagramme FAST regroupe les fonctions principales, techniques et contraintes globales définies dans lors de l'établissement des exigences et après leur discussion avec le client. Ainsi, certaines exigences que nous avons préalablement établit ont été acceptée mais d'autres ont écartée comme l'analyse des odeurs dans la ruche jugée finalement trop compliquée à mettre en place et difficile à exploiter. D'autres exigences ont vu le jours après les échanges avec l'apiculteur ce qui est venus enrichir et compléter notre tableau des exigences. Le raffinement des trois types de fonctions en sous fonctions et les solutions techniques associées a celles-ci apparaissent également. Il a évolué au cours du projet en fonction des autres documents d'ingénierie système et des solutions techniques retenues. On peut voir la version finale du FAST sur la figure 4.1

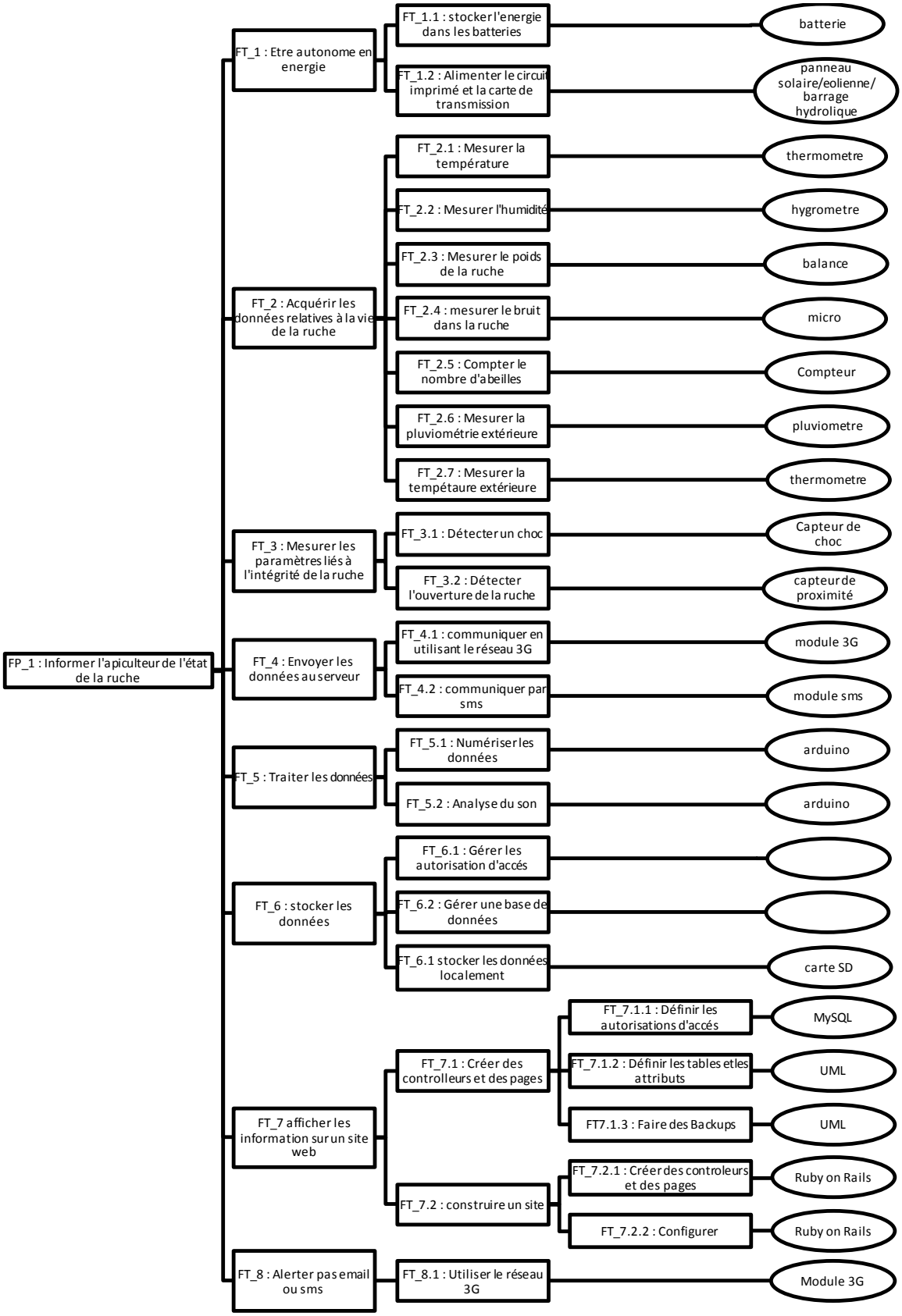

FIGURE 4.1 – Diagramme FAST du système BMONS

4.2 Spécification des données

La spécification des données permet de mettre à jour les différentes grandeurs et unités intervenant dans notre système. Grâce à cela, nous savons exactement quel type de donnée traiter et comment convertir les données numériques de sortie des capteurs en grandeurs physiques facilement compréhensible pour l'apiculteur. Il faudra ensuite envoyer ces données traitées au serveur qui les stockera dans une base de données. Cette étude a aussi permis d'établir les alertes qu'il va falloir prévoir afin d'avertir le propriétaire de l'état de son rucher. Ces dernières seront aussi stockées sur le serveur qui viendront compléter l'historique de la ruche. Il est possible qu'une alerte soit envoyée après avoir effectué un recoupement d'informations Par exemple, pour alerter l'apiculteur d'un essaimage en cours, il est nécessaire de détecter une agitation dans la ruche (chant de la reine) grâce au microphone et une chute importante de la masse grâce à la balance.

Il existera deux types d'alertes : les alertes actives qui préconiserons l'apiculteur à agir directement sur la ruche (par exemple lorsque le développement de la grappe est excentrée par rapport aux cadres) et des alertes passives qui encourageront le client à se rendre sur le serveur pour consulter les derniers relevés.

L'axe Data comportant la spécification des messages, des évènements et des alarmes est fourni en annexe. Le diagramme des données est décris dans la figure 4.2

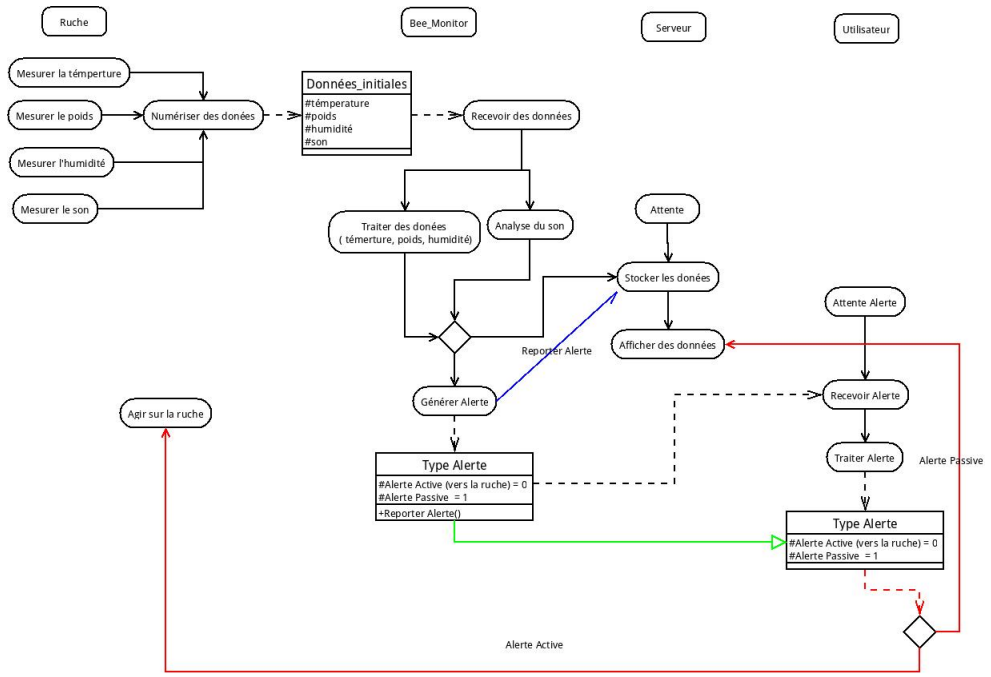

FIGURE 4.2 – Diagramme d'activité du flux de données

4.3 Modèle de Données

«««< HEAD

Le modèle de données est une façon abstraite de représenter les données du système et de modéliser les informations contenus dans une base de données. Le modèle constitue des ensembles possèdent d'un nom et des attributs nommés. La des relations est fait via des clés primaires (id, dans notre cas) et des clés étrangères. ======= Le modèle de données est une façon abstraite de représenter

les données du système et de modéliser les informations contenus dans une base de données. Le modèle constitue des ensembles possédant un nom et des attributs nommés. La des relations est fait via des clés primaires (id, dans notre cas) et des clés étrangères. »»»> 223e9576f5061e6f8a923b319f3dae6c3230f7d2

4.3

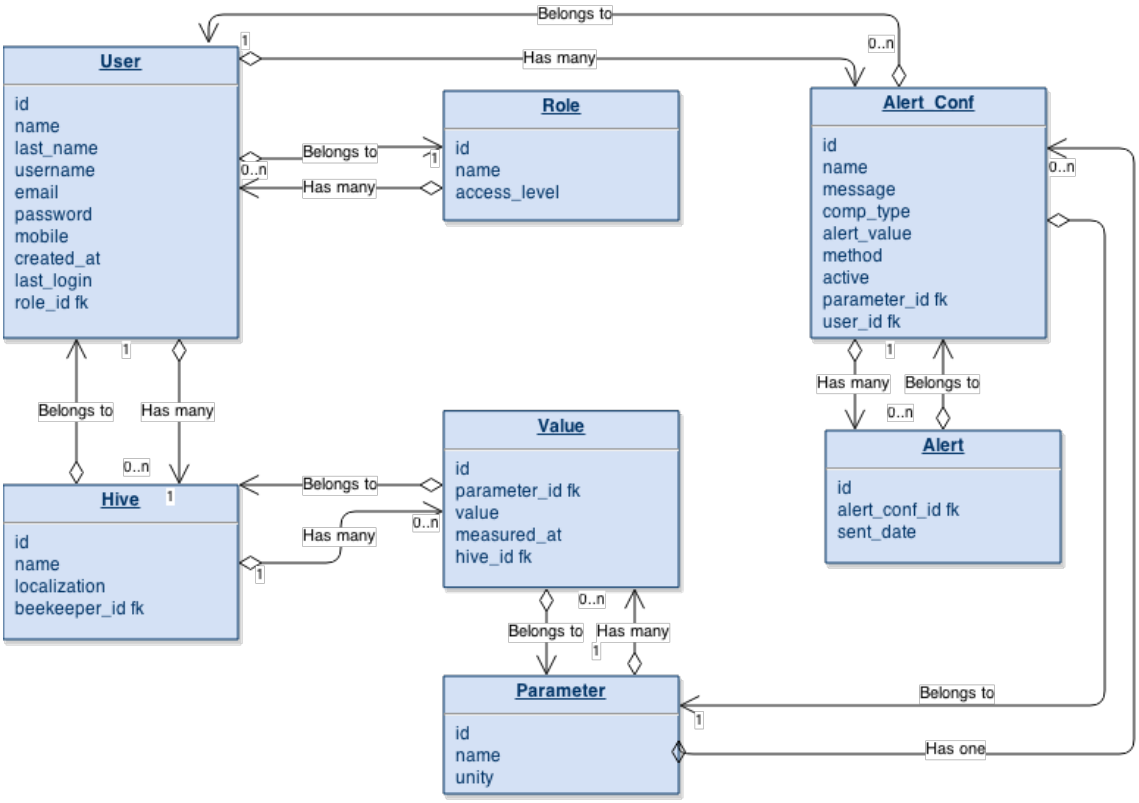

FIGURE 4.3 – Modèle de données pour le système BMONS

4.4 Spécification des comportements

Nous allons ici décrire le fonctionnement de notre système. Il est résumé dans la figure 4.9.

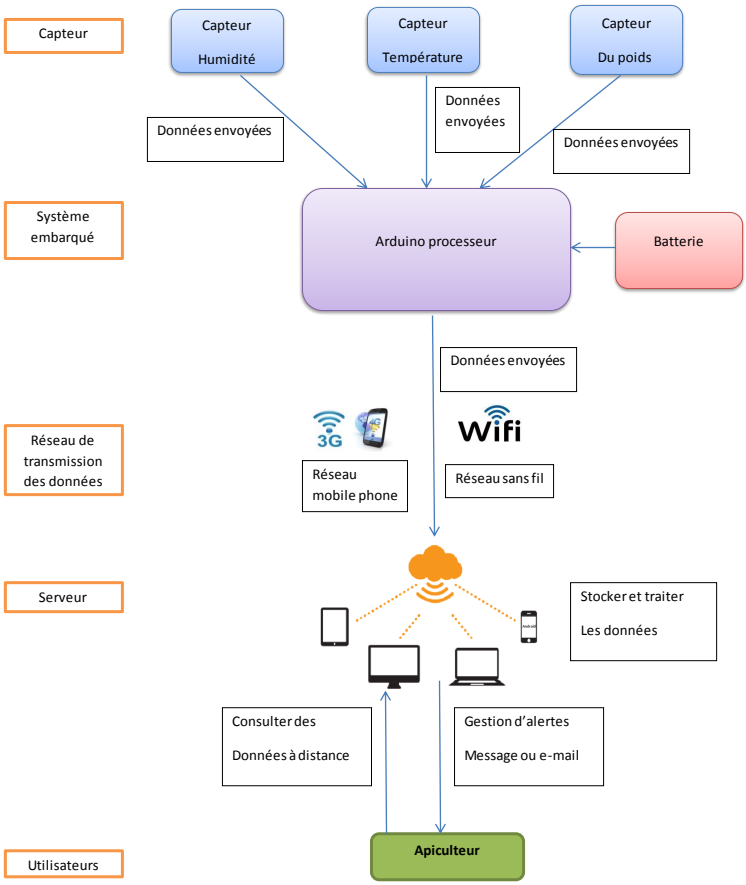

FIGURE 4.4 – Diagramme de spécification des comportements

Les capteurs mesurent plusieurs paramètres internes à la ruche : l'humidité, la température et le poids de la ruche. Les données sont ensuite transmises au système embarqué. Dès que le système embarqué les reçoit, il traite les données, sélectionne celles qui sont valides et les envoi au serveur soit par le réseau sans fil (2G ou 3G), soit par le réseau téléphone si besoin. Une fois que le serveur reçoit les données, elles sont traitées et stockées dans une base de données sécurisée. Une représentation sous forme de graphiques permet une vision pratique et exploitable

des informations par l'apiculteur.

Quand les mesures effectuées dépassent certain critères, par exemple, si la température est plus élevée que la température maximum pour la ruche, le serveur va générer un alerte qui sera envoyée à l'utilisateur, c'est-à-dire l'apiculteur, par SMS ou par e-mail. Par ailleurs, les apiculteurs peuvent consulter l'état de la ruche à distance afin de bien gérer la productivité de la ruche ou de limiter les situations problématiques.

La spécification du comportement ce fait également grace aux diagrammes de séquence. Ces diagrammes expliquent précisément comment les different parties du systeme interagisent dans le but d'executer une action. Ils détail l'ordre des actions ainsi que leur nature et leur durée d'execussion. A ce jour 5 diagrames de séquence ont été réalisé ils concernent l'action de récuperation de la mémoire flash, l'analyse des information par le serveur, le service permetant à l'apiculteur d'avertir le serveur d'une intervention sur une ruche, la fonction permetant la configuratiion des alertes et l'action de connecxion d'un apiculteur à son compte.

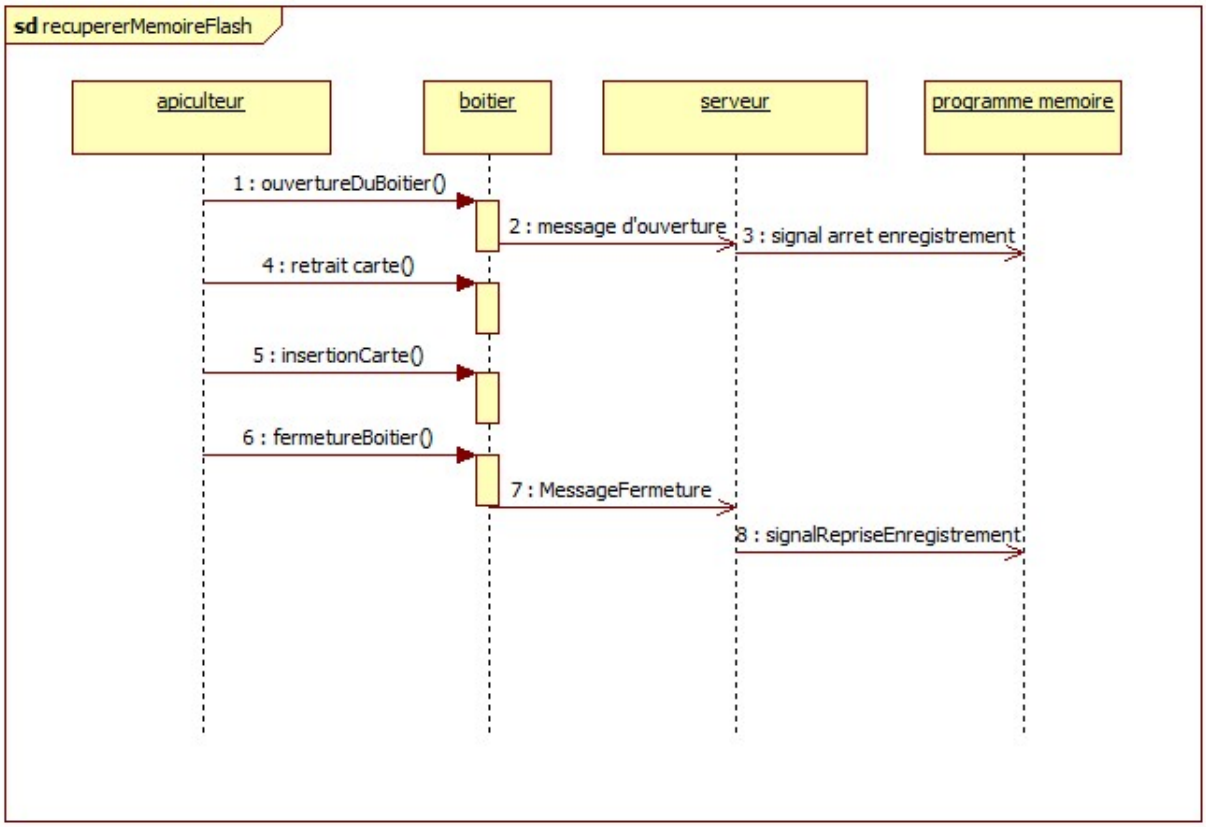

FIGURE 4.5 – Diagramme de spécification des comportements

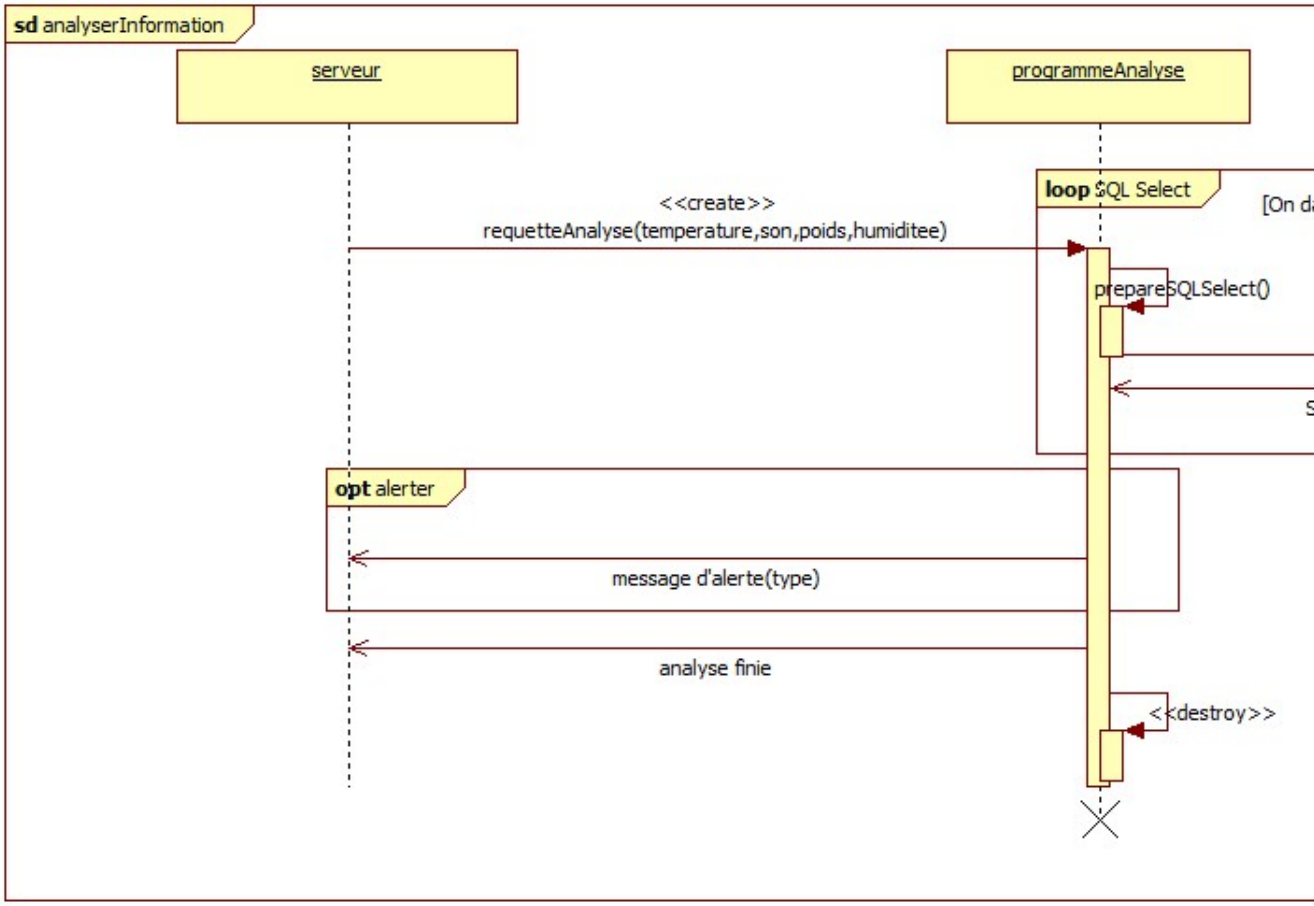

FIGURE 4.6 – Diagramme de spécification des comportements

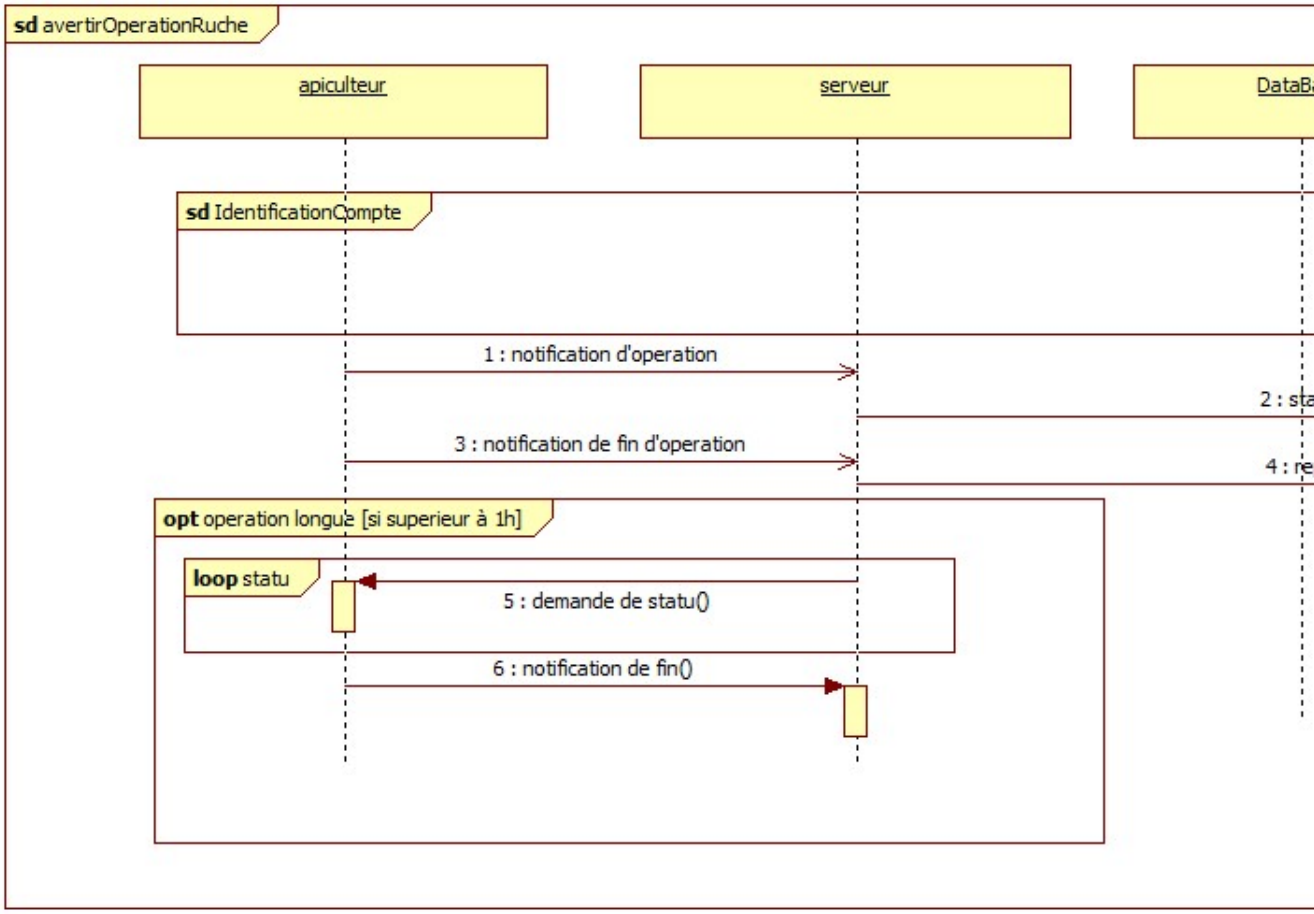

FIGURE 4.7 – Diagramme de spécification des comportements

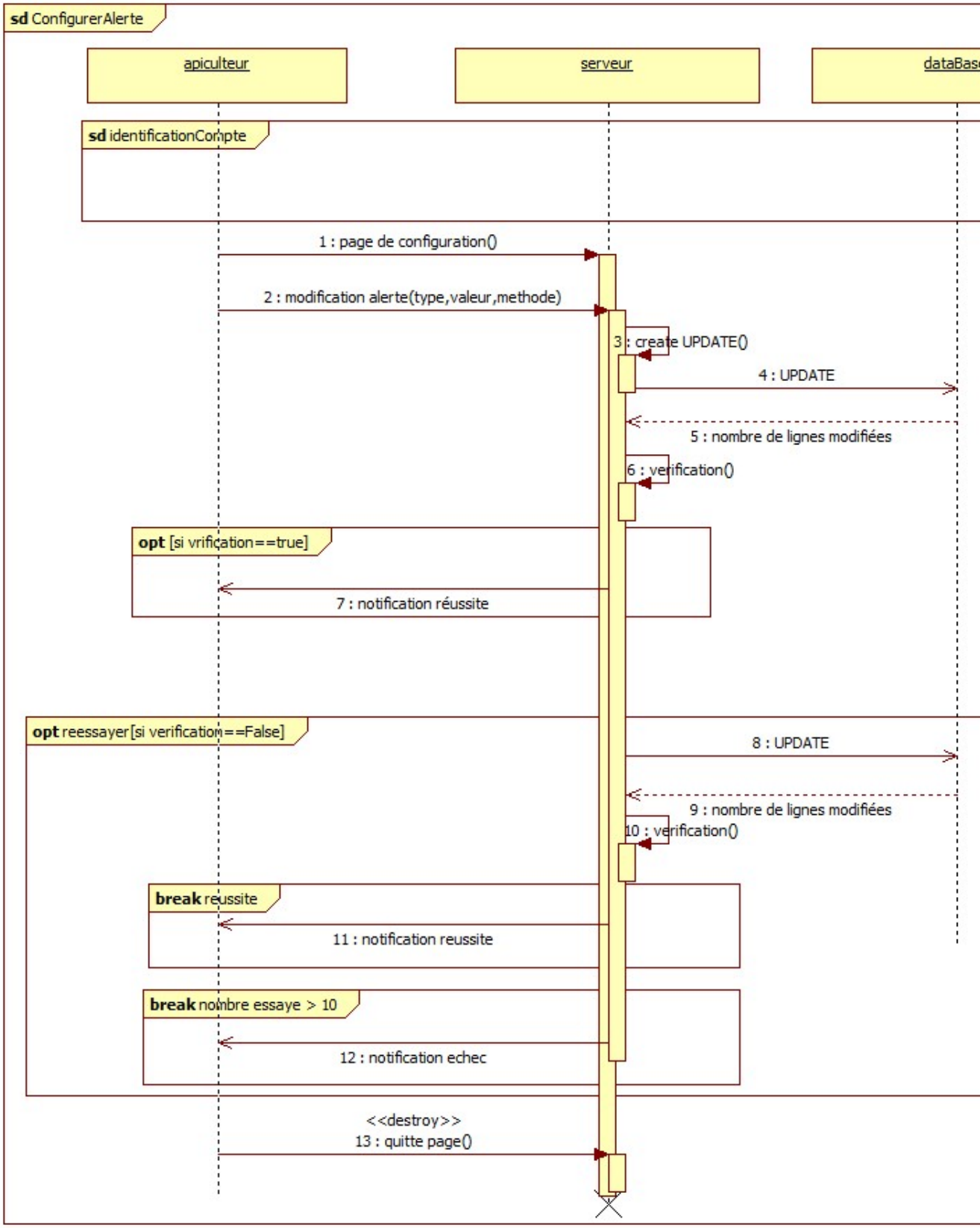

FIGURE 4.8 – Diagramme de spécification des comportements

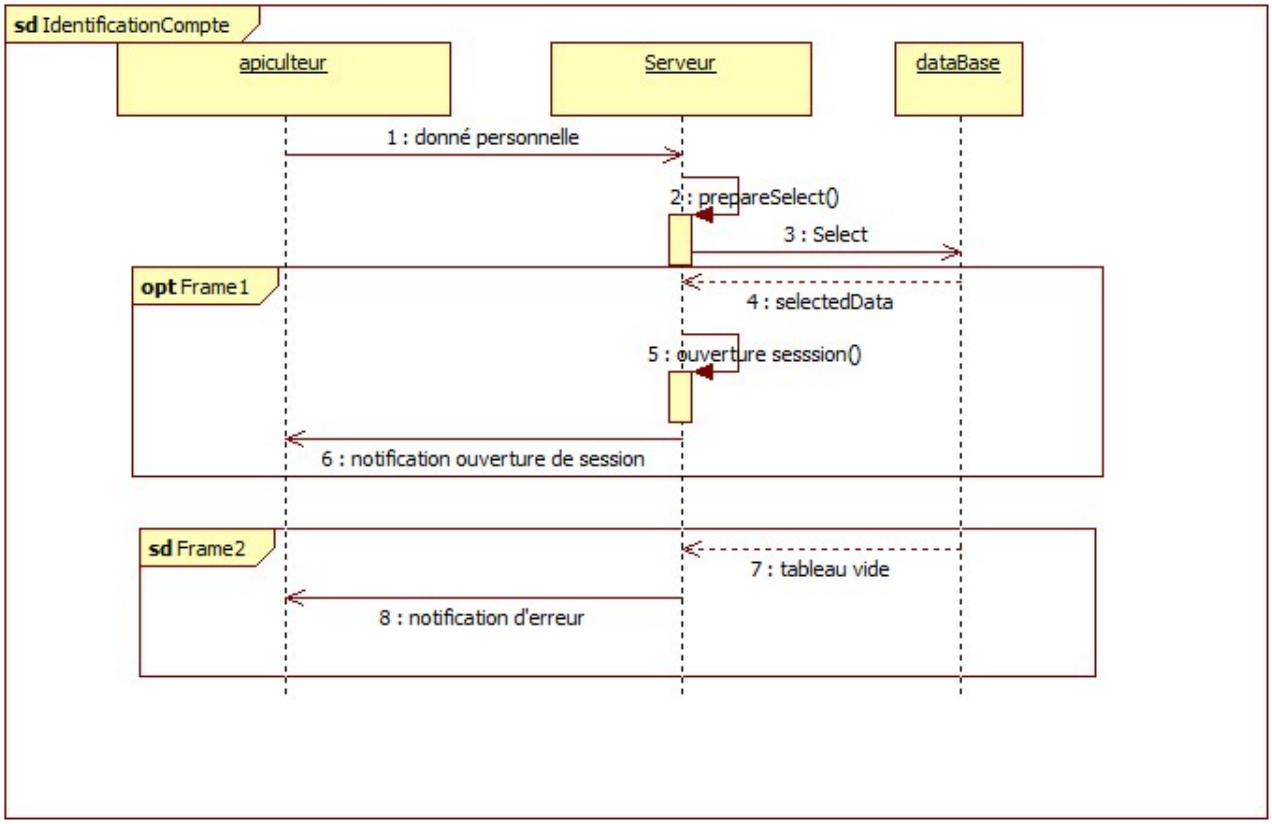

FIGURE 4.9 – Diagramme de spécification des comportements

Chapitre 5

Architecture fonctionnelle

L'étude de la spécification fonctionnelle trois axes a permis d'établir l'architecture fonctionnelle du système qui est représentée sur la figure 5.1. Ce schéma résume les interactions entre chaque partie : Bee Monitor qui regroupe l'ensemble des capteurs, la carte Arduino ainsi que la carte SSD pour l'enregistrement local des données, le module de transmission et le système d'alimentation rendant notre projet autonome en énergie et le serveur. Chaque acteur interagissant avec le système y sont également représentés : Les abeilles/ruche, l'apiculteur et l'environnement.

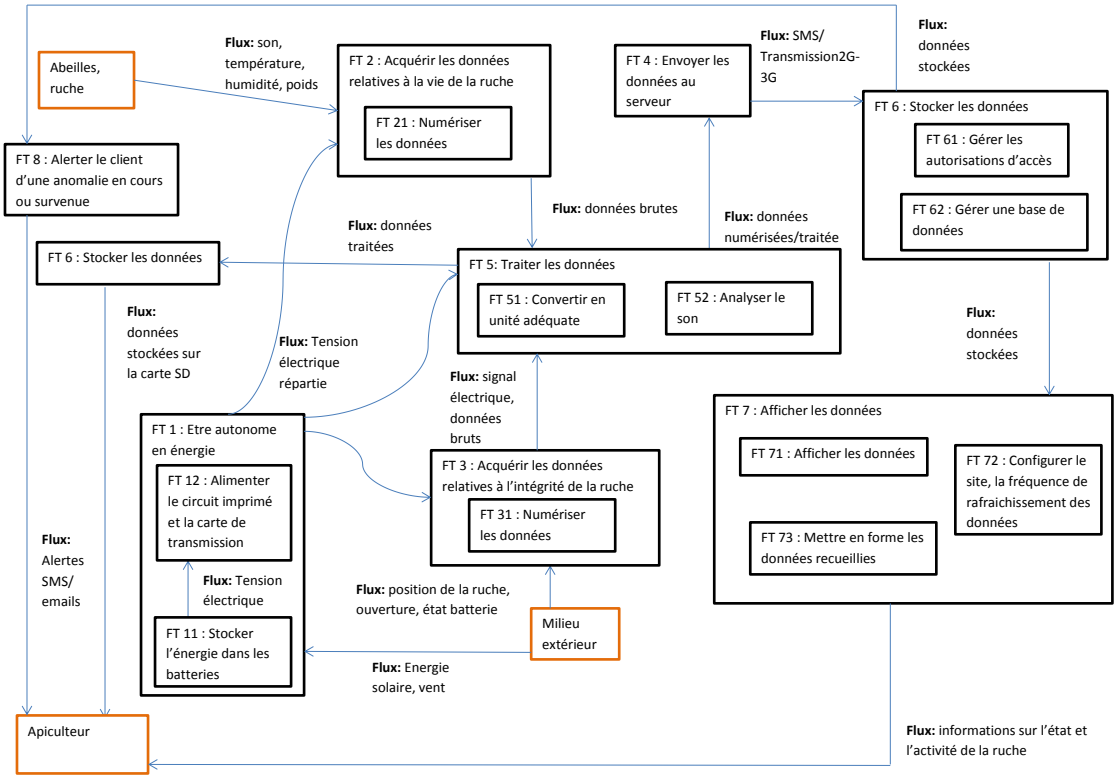

FIGURE 5.1 – Architecture fonctionnelle

Troisième partie

Implémentation

Chapitre 6

Architecture physique

6.1 Architecture physique

6.2 Structure de découpage du projet

Structure de découpage du projet, ou Work Breakdown Structure (WBS) en anglais, est un diagramme hiérarchique, axée sur les tâches et activités que l'équipe de projet doit exécuter pour atteindre les objectifs du projet.

Dans cette partie nous allons analyser nos tâches. Elles sont divisés en deux parties : Serveur et Arduino. La partie serveur comprend tout qu'est lié au développement du site et les scripts qui contrôlent les logiciels, les backups et l'obtention de données. La partie arduino comprend tous les capteurs et modules, l'énergie et les scripts de contrôle des capteurs et manipulation des données.

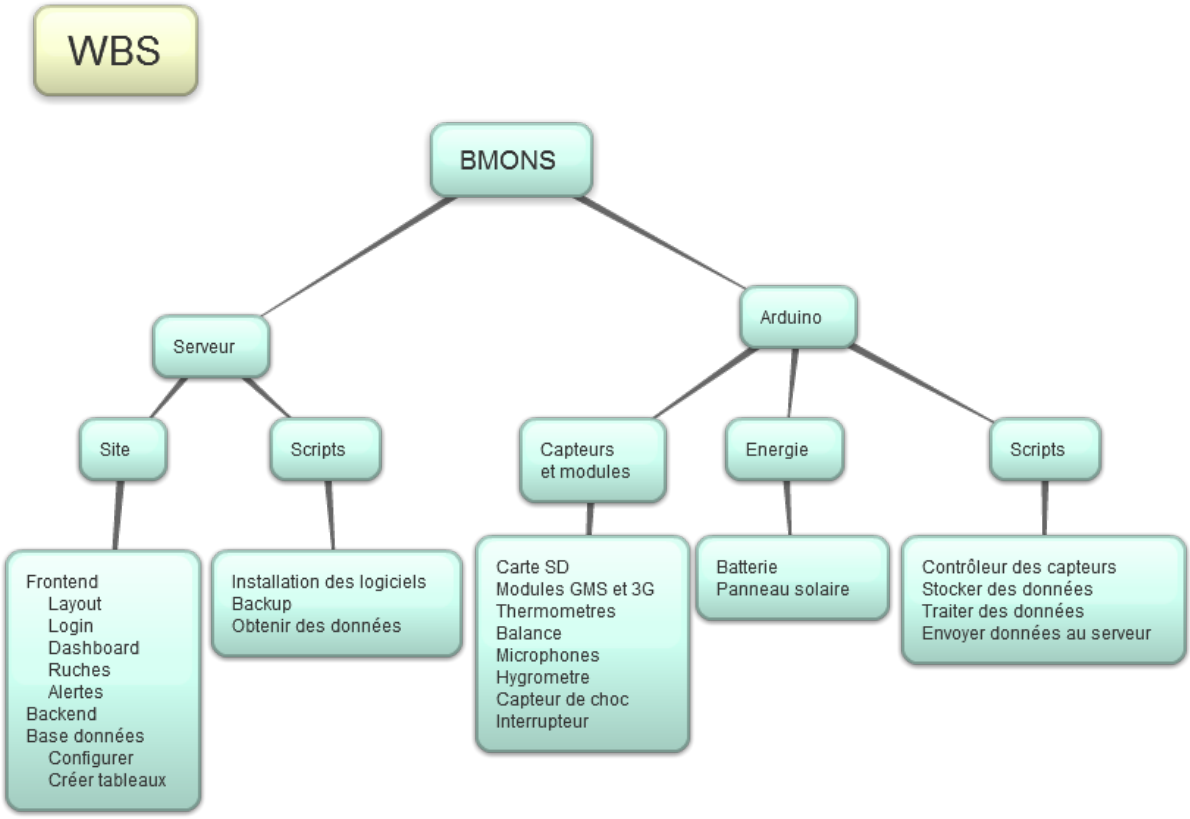

FIGURE 6.1 – Structure de découpage du projet du système BMONS

Quatrième partie

Organisation

Chapitre 7

Méthodes de travail

Méthodes de travail Organisation temporelle, spatiale, humaine interactions des membres de l'équipe projet interactions avec les encadrants interactions avec les tiers

Chapitre 8

Outils pour les échanges

Quels sont les outils qui nous permettent de travailler ensemble ?

Chapitre 9

Répartition des tâches dans le temps

WBS et diagramme de Gantt

Cinquième partie

Journal du projet

Chapitre 10

Choix et justifications

10.1 Choix des capteurs

Après avoir réalisé l'état de l'art pour notre système de surveillance d'une ruche, nous nous sommes ensuite intéressés aux capteurs que nous allons employer.

Capteur de température

Nous avons choisis un capteur de température identique pour recueillir la température interne et externe de la ruche. Il s'agit d'une thermistance NTC boitier Goutte Radial 1000 ohms sortie fil de cuivre émaillé 1 pc(s). Ce dernier possède une gamme de mesure comprise entre -40 C et 100 C ce qui correspond bien aux exigences discutées avec le client (Voir tableau des exigences). Le capteur placé à l'extérieur nous renseignera sur les conditions météorologique et expliquer une éventuelle diminution de la production de miel par exemple. Celui placé à l'intérieur nous donnera une idée de l'isolation de la ruche en hiver et de sa ventilation en été.

Capteur d'humidité

On a choisit d'inclure ce capteur dans notre système compte tenu des résultats de l'état de l'art. Néanmoins, après discussion avec le client, cette option n'est pas primordiale pour un apiculteur mais elle sera tout de même rajoutée au tableau de bord du serveur.

Capteur de pression

Les capteurs de pression vont nous permettre de récupérer le poids de ruche et surtout celui des hausses pour avertir l'apiculteur de la quantité de miel produite. Pour se faire, le projet Bzzz développé par le Fablab de Lannion a prévu d'utiliser deux sachets de "Pompote" remplis d'eau sucrée pour éviter les variations de pression atmosphérique, le gel et l'évaporation. Cependant, après avoir pris conscience de l'importance de la localisation de la grappe, nous avons pensé installer deux capteurs de pression par cadre (au niveau de chaque extrémité) soit

20 au total mais cette solution s'est avérée être difficile à mettre en place à cause de la surface sur laquelle repose les cadres (simple lamelle en métal). Après discussion avec le client, nous avons finalement opté pour la confection d'un cadre en bois aux même dimensions de la ruche dans lequel se trouverons les capteurs de pression. L'utilisateur décidera de l'endroit où le placer en fonction des données qu'il veut récupérer. On pourra aussi utiliser ce capteur pour détecter l'ouverture du couvercle à cause du vent.

Tilt sensor

Ce capteur permet de savoir si la ruche a reçu un choc ou si elle a été déplacé. Il renvoi une information binaire qui pourra être couplée avec les données de la carte GPS et ainsi avertir l'apiculteur en cas de détérioration, renversement ou vol de la ruche. Ce dernier évènement étant de plus en plus fréquent. Nous avons choisit le TikerKit Tilt Sensor.

Microphone

L'utilisation d'un microphone s'est imposée comme une nécessité dans la surveillance d'une ruche car il permet de recueillir des données pouvant avertir l'apiculteur sur plusieurs types d'évènements. En effet, grâce à ce dernier, on pourra détecter la présence d'abeilles dans la ruche notamment en hiver et éventuellement recueillir la source du bourdonnement et ainsi localiser la grappe approximativement. On pourra aussi détecter les prémisses d'un essaimage en percevant le champs d'une raine caractéristique de ce type d'évènement. Cette information pourra ensuite être couplée avec les données des capteurs de pression et confirmer l'essaimage en cours si le poids chute brutalement. Nous avons choisit le Capsule micro pour CI 2 V/DC sensibilité 44 dB (à 3 dB près) sur la plage 100 - 10000 Hz.

Chapitre 11

Résultats et analyses

analyse des tests et des performances analyse des échecs, des décalages et des retards Que reste-t-il à faire ? Comment ?

Chapitre 12

Conclusion

Sixième partie

Annexes

Annexe A

Première annexe

Annexe B

Deuxième annexe

Annexe C

Troisième annexe

Références bibliographiques

[1] Apidictor. http://www.beesource.com/build-it-yourself/apidictor/. Accessed : 2014-12-15.

[2] Arnia remote hive monitoring. http://www.arnia.co.uk/. Accessed : 2014-12-01.

[3] Open source beehive. http://www.opensourcebeehives.net/. Accessed : 2014-12-01.

[4] AFIS : Accueil - notre métier : L'ingénierie système. http://www.afis.fr/nm-is/default.aspx, 2010. Accédé le 30 août 2014.

[5] Denis Bitouzé et Jean-Côme Charpentier : *LATEX, l'essentiel*. Pearson Education France, 2010.

[6] Bernard Desgraupes : *LATEX : apprentissage, guide et référence*. Vuibert informatique, 2003.

[7] fablab Lannion : Bzzz : Suivi des ruches. http://fablab-lannion.org/wiki/index.php?title=Suivi_des_ruches. Accessed : 2014-12-01.

[8] Meyer Ken : Honeybee hive monitoring. http://hackaday.io/project/1741-honeybee-hive-monitoring. Accessed : 2014-12-01.

[9] Leslie Lamport : *LaTeX—A Document*, volume 410. pub-AW, 1985.

[10] Noël-Arnaud Maguis : *Rédigez des documents de qualité avec LaTeX*. Livre du Zéro, 2010.

[11] Noël-Arnaud Maguis : Rédigez des documents de qualité avec latex. http://fr.openclassrooms.com/informatique/cours/redigez-des-documents-de-qualite-avec-latex, 2013. Accédé le 30 août 2014.

[12] Fabrizio Sebastiani : Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.