

# RAPPORT D'AVANCEMENT

## BMONS

Beehive Monitoring System

Rédigé par :

Alice Danckaers

Benoît Raymond

Etienne Dalcol

Nicolas Van-Nhân Nguyen

Tao Zheng

Armand Sellier

Sous la direction de :

Olivier Reynet



**ENSTA**  
Bretagne

Option Systèmes Perception Information Décision

 2014 Alice Danckaers, Benoît Raymond, Etienne Dalcol, Nicolas Van-Nhân Nguyen, Tao Zheng and Armand Sellier.

Licensed under the Creative Commons Attribution-ShareAlike 4.0 International Public License.

*Première impression, décembre 2014*

# Sommaire

<b>Remerciements</b>	<b>i</b>
<b>I Introduction au projet</b>	<b>1</b>
<b>1 Formulation initiale du projet</b>	<b>3</b>
1.1 Contexte . . . . .	3
1.2 Expression initiale du besoin . . . . .	3
<b>2 État de l’art</b>	<b>5</b>
<b>II Dossier fonctionnel</b>	<b>9</b>
<b>3 Ingénierie des exigences</b>	<b>11</b>
3.1 Approche Top-Down . . . . .	11
3.2 Approche Bottom-Up . . . . .	13
3.3 Fonctions métiers du système . . . . .	15
<b>4 Spécification fonctionnelle 3 axes</b>	<b>17</b>
4.1 Raffinement FAST . . . . .	17
4.2 Spécification des données . . . . .	19
4.3 Modèle de Données . . . . .	20
4.4 Spécification des comportements . . . . .	21
<b>5 Architecture fonctionnelle</b>	<b>27</b>
<b>III Implémentation</b>	<b>29</b>
<b>6 Architecture physique</b>	<b>31</b>
6.1 Architecture physique matérielle . . . . .	31
6.2 Architecture physique logicielle . . . . .	33
<b>7 Tests unitaires</b>	<b>35</b>

<b>IV</b>	<b>Organisation</b>	<b>37</b>
<b>8</b>	<b>Méthodes de travail</b>	<b>39</b>
8.1	Communication . . . . .	39
8.2	Outils pour les échanges . . . . .	39
<b>9</b>	<b>Répartition des tâches</b>	<b>41</b>
9.1	Structure de découpage du projet . . . . .	41
9.2	Diagramme de Gantt . . . . .	41
<b>V</b>	<b>Journal du projet</b>	<b>43</b>
<b>10</b>	<b>Choix et justifications</b>	<b>45</b>
10.1	Choix des capteurs . . . . .	45
10.2	Choix de la carte Arduino . . . . .	47
10.3	Choix de la solution technique . . . . .	48
<b>11</b>	<b>Résultats et analyses</b>	<b>49</b>
<b>12</b>	<b>Conclusion</b>	<b>51</b>
<b>VI</b>	<b>Annexes</b>	<b>53</b>
<b>A</b>	<b>Première annexe</b>	<b>55</b>
<b>B</b>	<b>Deuxième annexe</b>	<b>61</b>
<b>C</b>	<b>Troisième annexe</b>	<b>63</b>
	<b>Bibliographie</b>	<b>65</b>
	<b>Index</b>	<b>67</b>
	<b>Glossaire</b>	<b>67</b>

# Remerciements

La gratitude est non seulement la plus grande des vertus, mais c'est également la mère de tous les autres.

---

Emil Cioran

L'équipe du projet BMONS (BeeHive Monitoring System) souhaiterait remercier M. REYNET, responsable de l'UV 3.4 et auteur de notre sujet ainsi que notre encadrant IS, M.CLEMENT pour leurs conseils, leur soutien et leur disponibilité. Ils ont su nous guider tout au long du projet notamment sur le plan technique et architectural de notre système.

Nous tenons également à remercier M.SINGHOFF, professeur à l'université de Bretagne Occidentale à Brest dans le domaine des systèmes embarqués et apiculteur qui a accepté de prendre part à notre projet en tant que "client" en partageant ainsi son expérience de l'apiculture. Il a aussi mis à notre disposition une de ses ruches afin que nous puissions nous rendre compte des dimensions et de l'espace disponible pour installer notre système.



# 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 disparaîtraient. Or les abeilles sauvages sont aujourd'hui rares et l'espèce ne survivra 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 telle 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'ait 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 sécurité 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 [9] 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 [8], 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 [4]. 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.

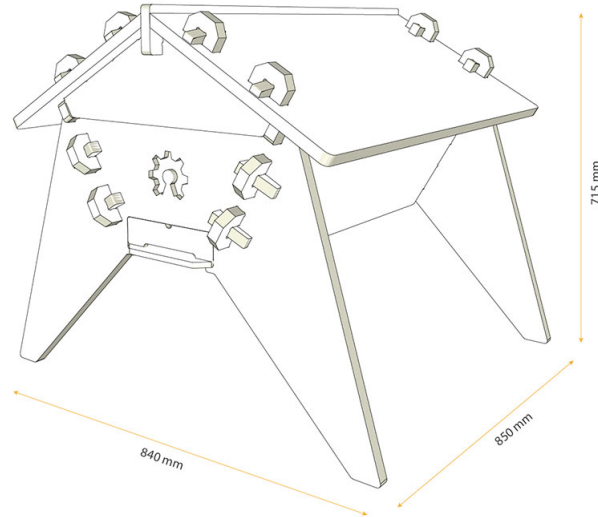


FIGURE 2.2 – Modèle de ruche Open Source Beehive

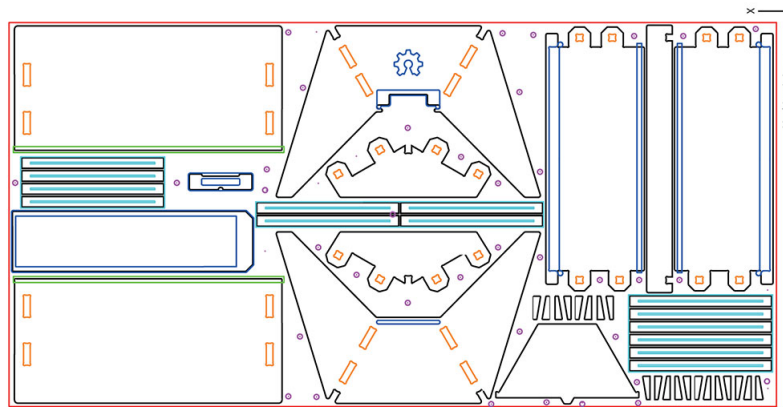


FIGURE 2.3 – Plan de la ruche Open Source Beehive

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.

A propos de la transmission de l'information à l'apiculteur, deux moyens sont très souvent utilisés : un site internet sécurisé et une application pour smart-phone. Concernant le site internet, celui de la société Arnia nous a semblé complet et clair. Pour l'application smart-phone, l'entreprise américaine B-Ware en a mis une au point mettant bien en avant le tableau de bord, l'affichage de l'historique des paramètres de la ruche et la personnalisation des alertes en laissant même l'apiculteur définir lui même les seuils de déclenchement des avertissements comme nous pouvons le voir sur la figure 2.4.

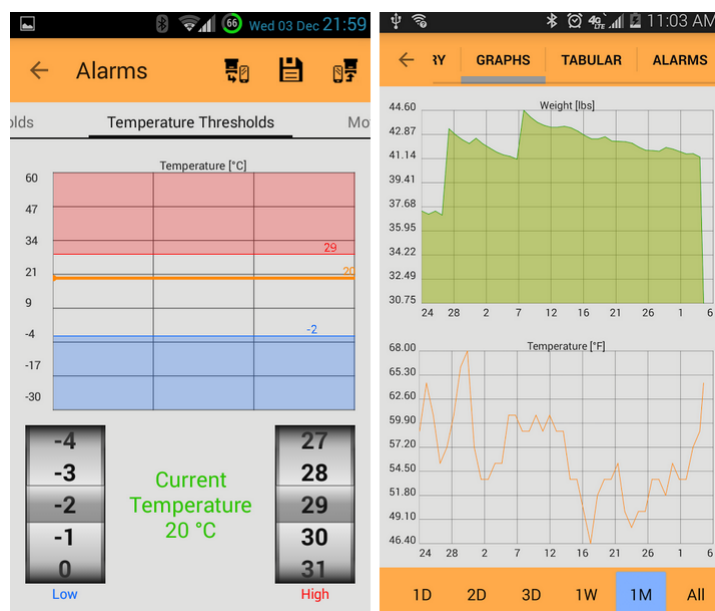


FIGURE 2.4 – Capture d'écran de l'application B-Ware

La société BeeWise d'origine française a développé un système d'alimentation écologique à l'aide de panneaux solaires connectés au boîtier regroupant les cartes électroniques pour le traitement des données et de transmission [3]. Il suffirait juste de la fixer sur le couvercle de la ruche en prenant soit d'étudier l'orientation adéquate au préalable comme sur la figure 2.5. Cette solution semble être la plus répandue car elle est également utilisée par les développeurs de l'ApiScan, un système de comptage de d'abeilles installé sur la planche d'envol pour minimiser la gêne occasionnée.

Les recherches sur l'exploitation des ruchers nous a mené à comprendre que l'un des principaux risques pour les apiculteurs est le vol de ruche. Cette pratique s'est très largement répandue ces dernières années. C'est pour cela que la société Apimiel met à disposition un système GPS pour permettre à l'apiculteur

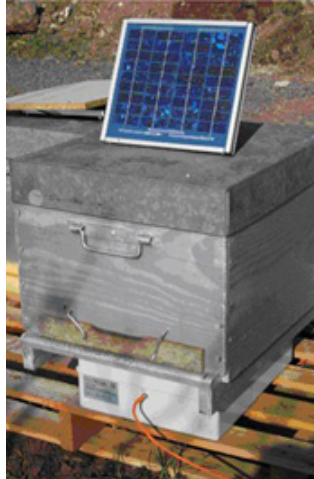


FIGURE 2.5 – Système de panneau solaire installé sur une ruche

de connaître en temps réel la position de ses ruches [?]. Au moindre déplacement, l'apiculteur recevra une alerte SMS. Compte tenu de sa taille, le traceur GPS devra se trouver à l'extérieur de la ruche mais il existe aussi des versions plus petites que l'on peut directement placer dans la ruche comme sur la figure 2.6. C'est ce que Michel BOCQUET, ingénieur agronome, propose sur son blog. Néanmoins, son objectif final est d'étudier le comportement des abeilles pour mieux comprendre celui de notre environnement et non commercialiser son projet aux apiculteurs. Le cout du matériel n'est donc pas pris en compte.



FIGURE 2.6 – Système GPS placé dans la ruche

# 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.

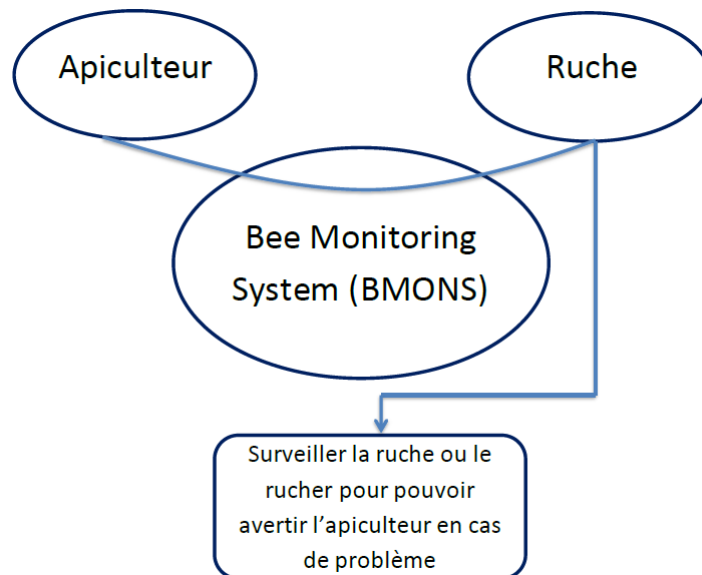


FIGURE 3.1 – Diagramme "bête à cornes" du système BMONS

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.

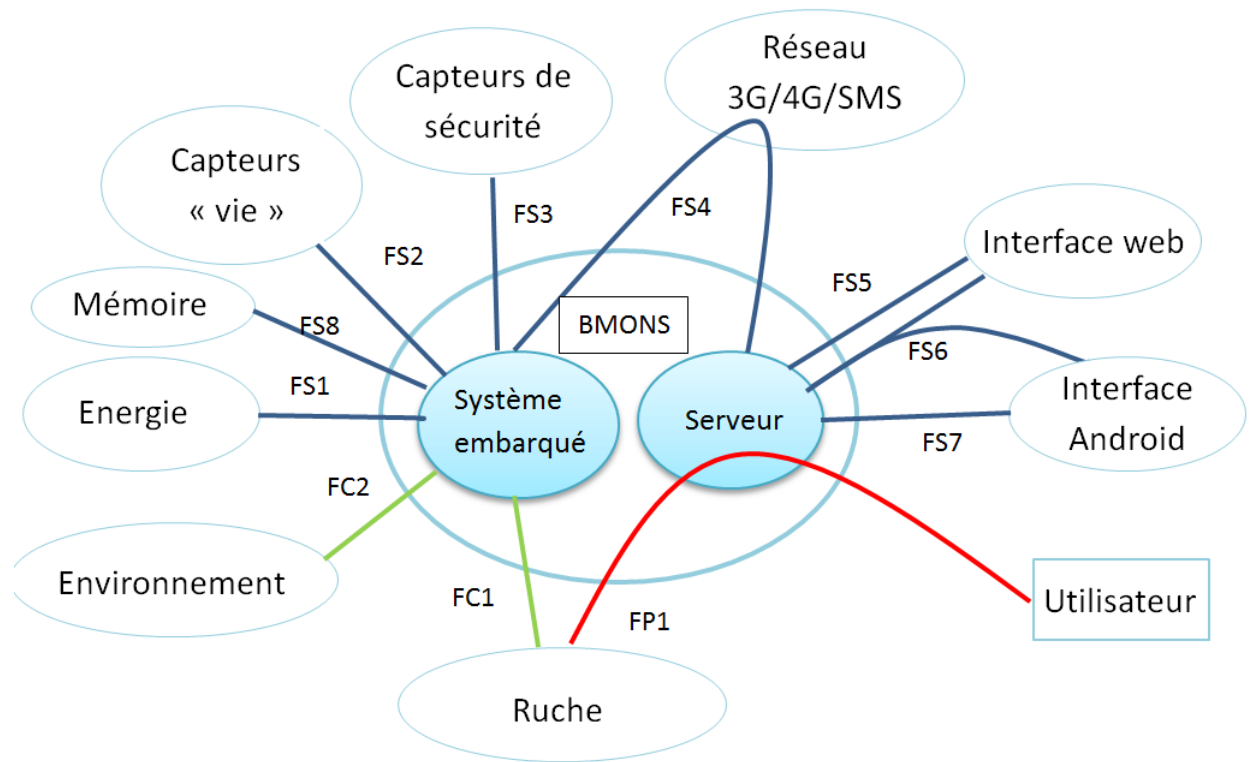


FIGURE 3.2 – Diagramme pieuvre du système BMONS

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 [A.1](#) et [A.2](#) 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 réglementation alimentaire				Adapter à la législation
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 données reçues
exi_42	service	Reconnaitre le chant des futures 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

Après avoir réalisé une approche du point de vue "concepteur" du système, nous allons maintenant nous intéresser à la formulation des fonctions qu'un apiculteur souhaiterait avoir pour pouvoir suivre l'évolution de son rucher. Ces fonctions métiers ont été discutées avec notre "client", M. SINGHOFF. Elles sont présentées sur la figure A.1.

	expression de la fonction métier	exigences	Performance
exi met 1	suivre la miellée	Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 2/jour
		Avertir l'apiculteur en cas de diminution significative du poids	
		Transmettre les données par ondes radio	3G ou 4G
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 2	détecter la présence d'abeille dans la ruche	Relever le bruit dans la ruche	
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
		Transmettre les données par ondes radio	3G ou 4G
exi met 3	suivre le trafic	Mesurer le poids de la ruche	
		Relever le bruit dans la ruche	
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
		Transmettre les données par ondes radio	3G ou 4G

FIGURE 3.6 – Fonctions métiers du système (1/4)



# 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](#)

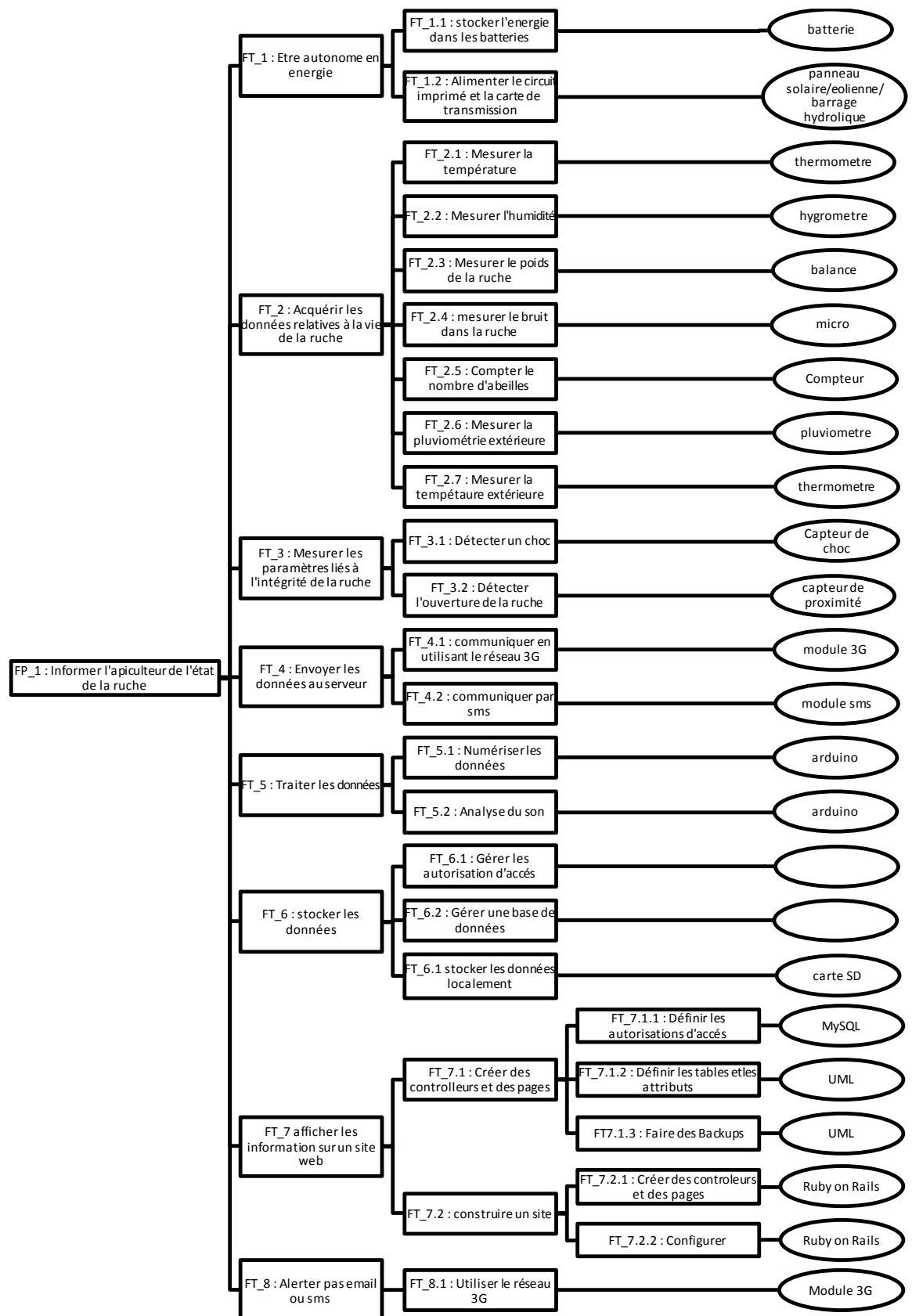


FIGURE 4.1 – Diagramme FAST du système BMONS



## 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 préconiseront 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 événements et des alarmes est fourni en annexe. Le diagramme des données est décrit dans la figure 4.2

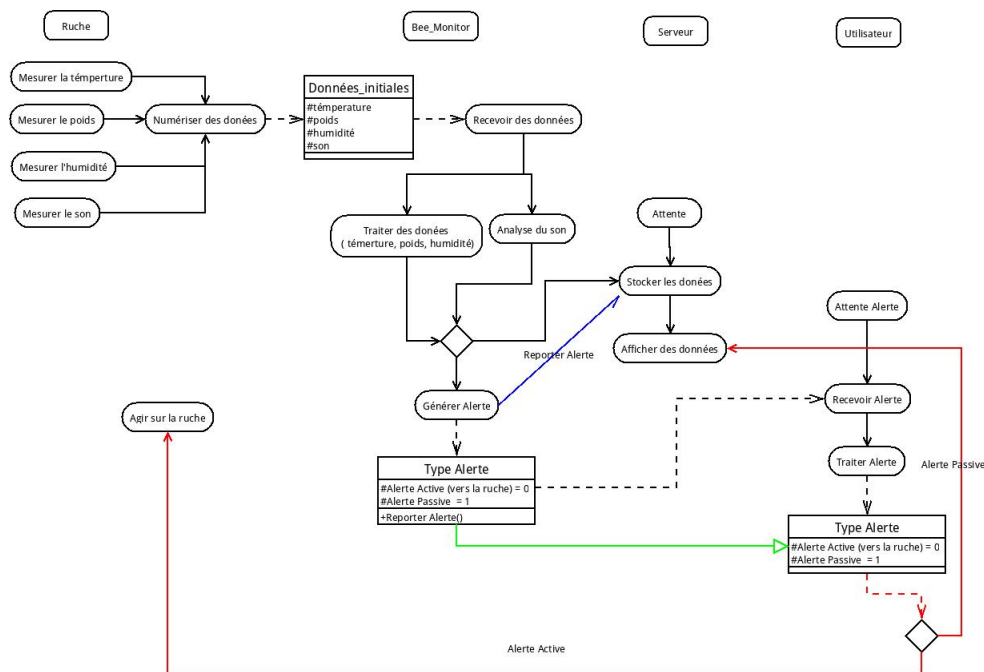


FIGURE 4.2 – Diagramme d'activité du flux de données

### 4.3 Modèle de Données

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.

4.3

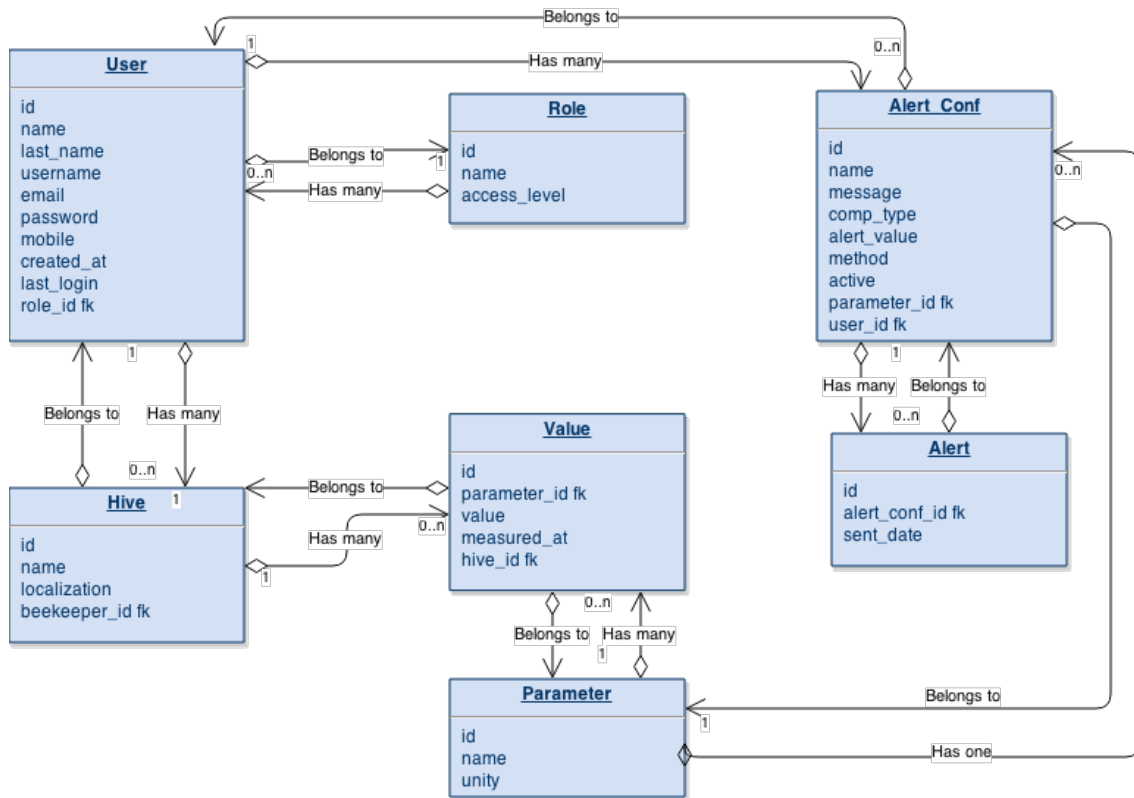


FIGURE 4.3 – Modèle de données pour le système BMONS

## 4.4 Spécification des comportements

Nous allons ici décrire le fonctionnement de notre système. Il est résumé dans la figure 4.4.

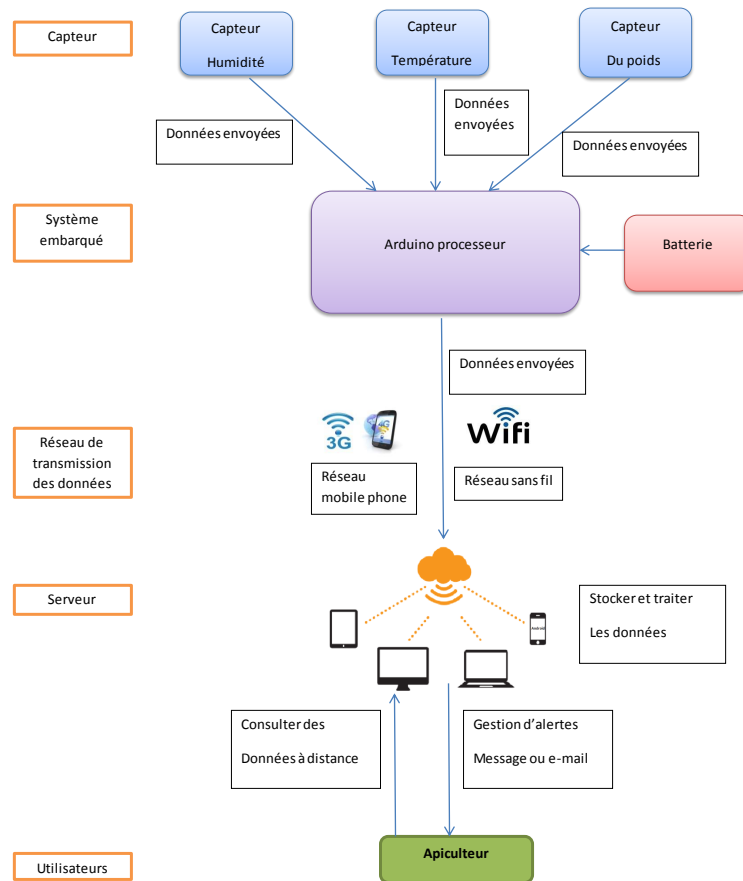


FIGURE 4.4 – Diagramme de spécification des comportements

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 envoie 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 certains 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 se fait également grâce aux diagrammes de séquence. Ces diagrammes expliquent précisément comment les différentes parties du système interagissent dans le but d'exécuter une action. Ils détaillent l'ordre des actions ainsi que leur nature et leur durée d'exécution. Les 5 diagrammes de séquence ont été réalisés concernant l'action de récupération de la mémoire flash voir la figure 4.5, l'analyse des informations par le serveur voir la figure 4.6, le service permettant à l'apiculteur d'avertir le serveur d'une intervention sur une ruche voir la figure 4.7, la fonction permettant la configuration des alertes voir la figure 4.8 et l'action de connexion d'un apiculteur à son compte voir la figure 4.9.

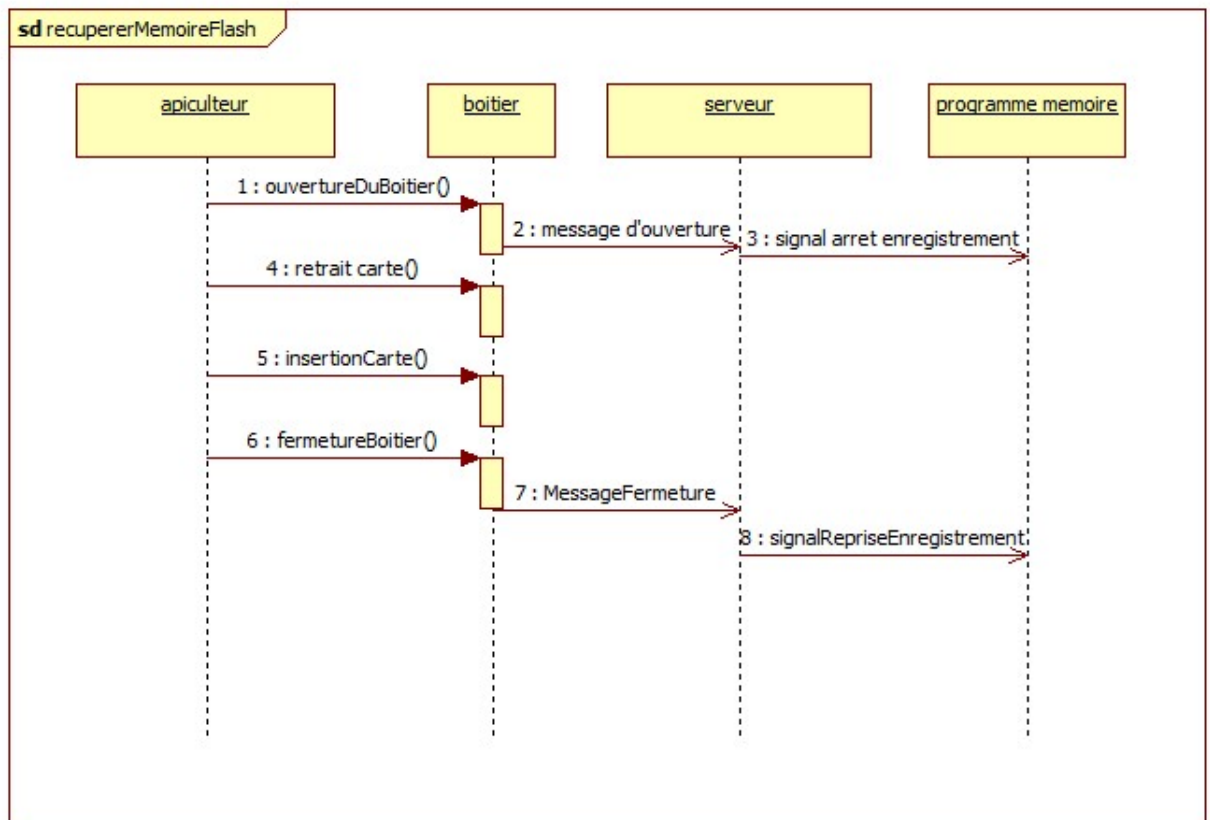


FIGURE 4.5 – Diagramme de récupération de la mémoire flash

La figure 4.5 illustre de la façon suivante les quatre blocs en haut qui représentent les acteurs de l'action `recupererMemoireFlash`. Ces acteurs interagissent par l'intermédiaire de requêtes ou d'appels de fonction représentés par les flèches. Le tout se déroule selon un axe temporel symbolisé par les lignes verticales sous les acteurs.

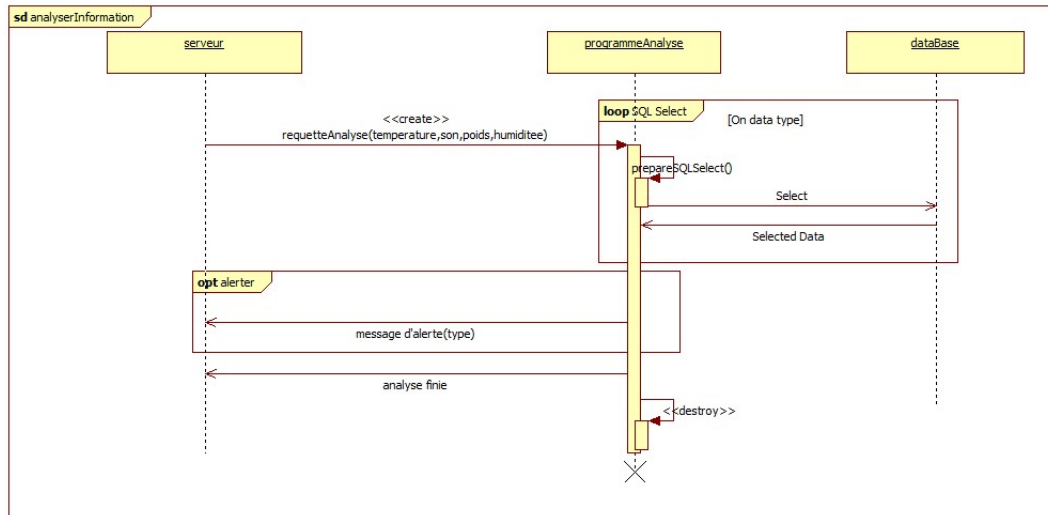


FIGURE 4.6 – Diagramme d'analyse de l'information

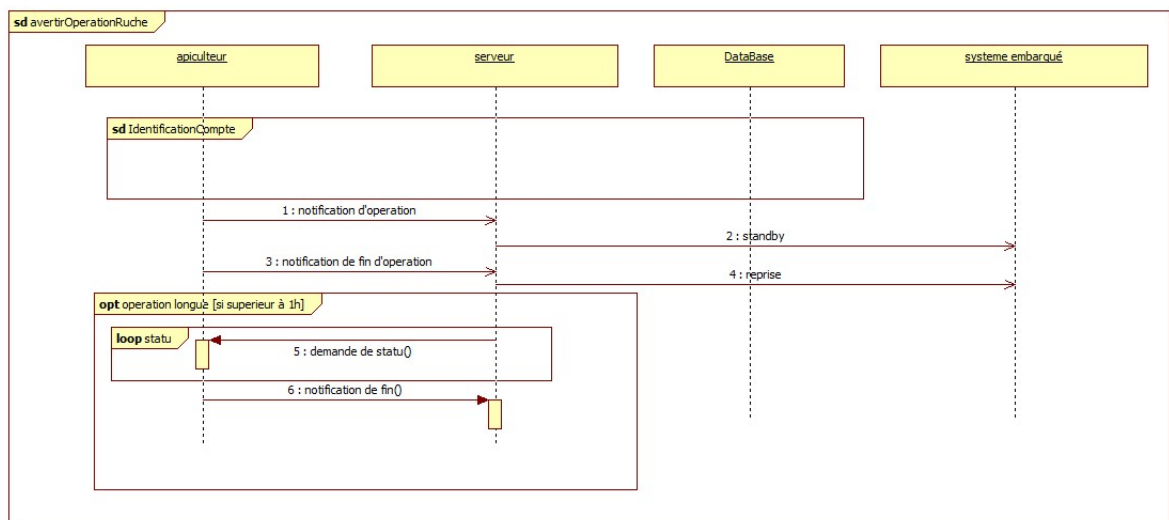


FIGURE 4.7 – Diagramme d'avertissement d'opération sur la ruche

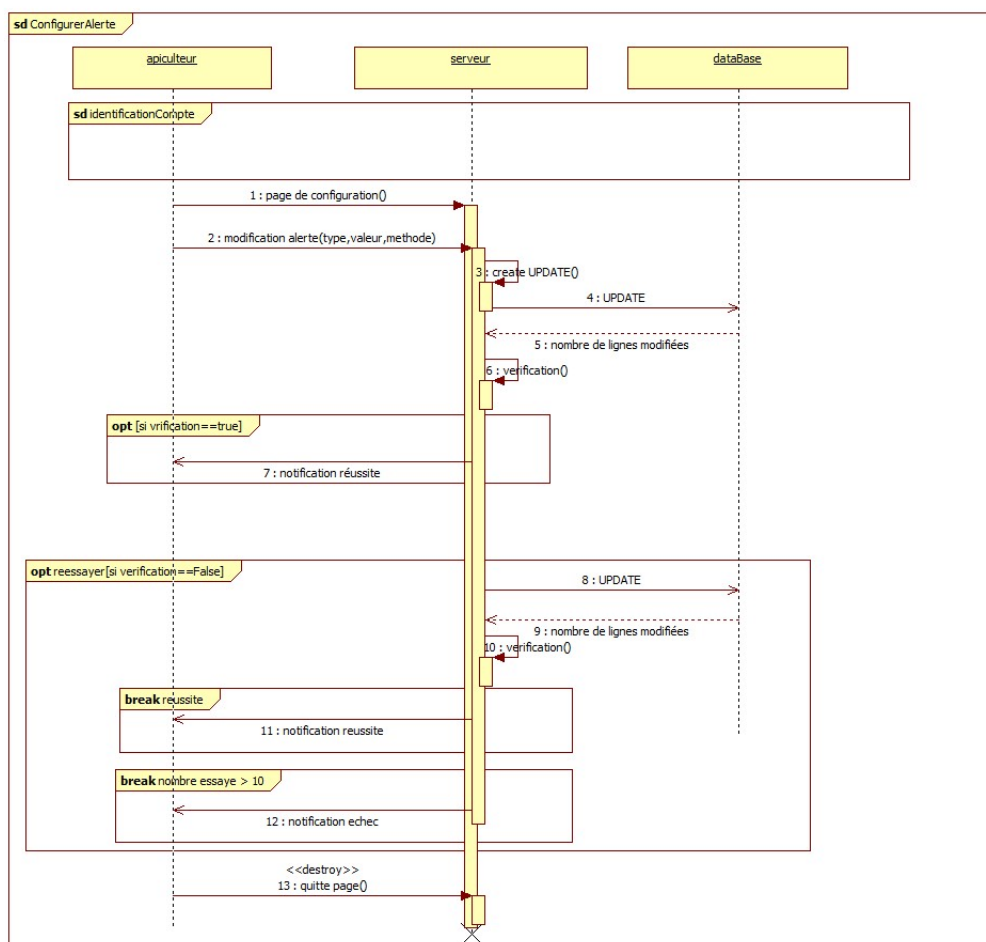


FIGURE 4.8 – Diagramme de configuration des alertes

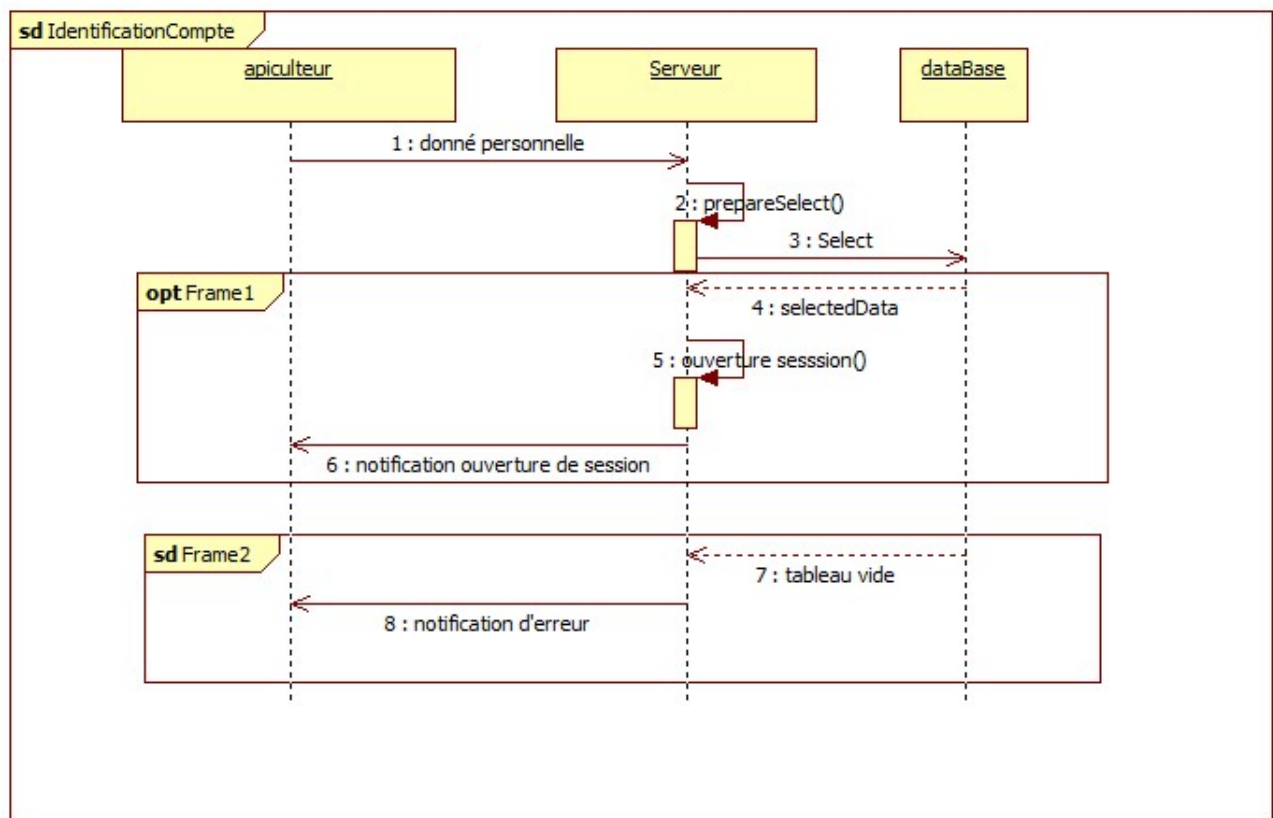


FIGURE 4.9 – Diagramme d'identification compte





# 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.

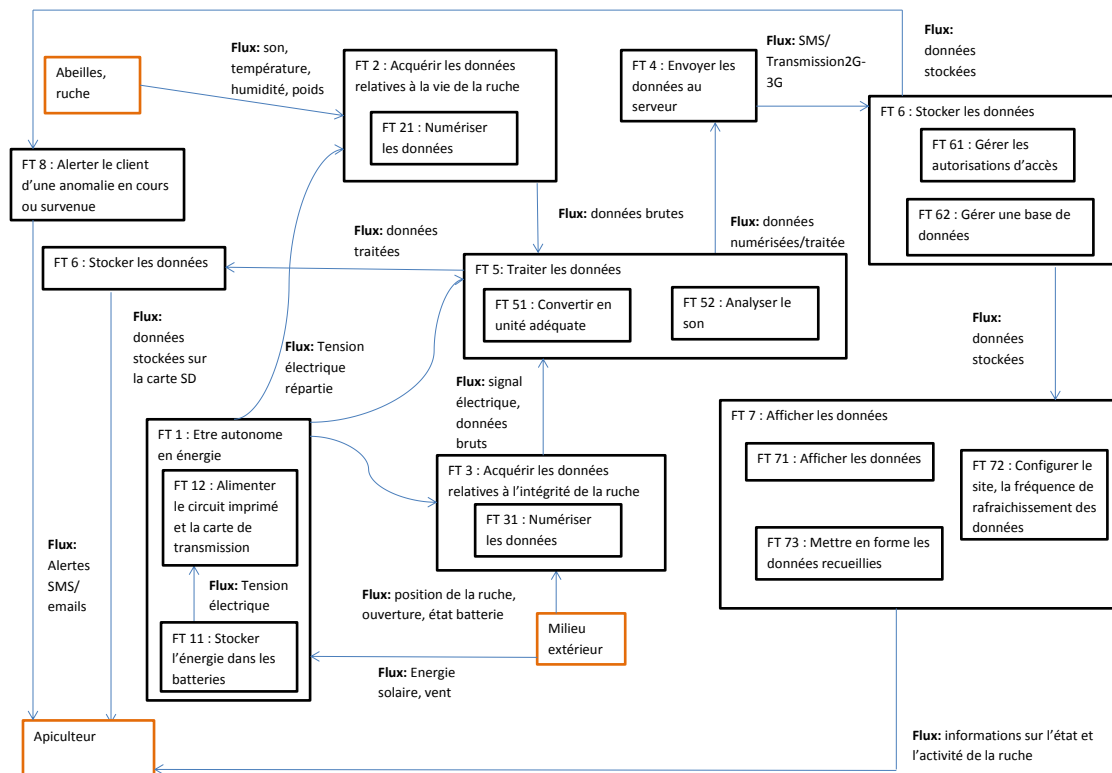


FIGURE 5.1 – Architecture fonctionnelle



# Troisième partie

## Implémentation



# Chapitre 6

## Architecture physique

### 6.1 Architecture physique matérielle

La solution technique que nous avons choisie se présente sous la forme d'un cadre en bois léger (type contreplaqué) dans lequel les capteurs sont incrustés. On peut voir sur la vue du dessus du cadre en coupe, figure 6.3, la position des différents capteurs. Sur deux cotés opposés il y aura 4 capteurs de pression et 3 capteurs de température, voir figure 6.1. Puis sur un autre coté le capteur d'humidité, le microphone et la sortie des fils, voir figure 6.2. Enfin sur le dernier côté il y aura uniquement un microphone. Les capteurs seront placés entre deux épaisseurs de bois préalablement travaillées et dépasseront si nécessaire du cadre. Les fils de connexion seront rassemblés et sortiront à un seul endroit du cadre. Ils seront placés dans une gaine protectrice et se connecteront au boîtier extérieur.

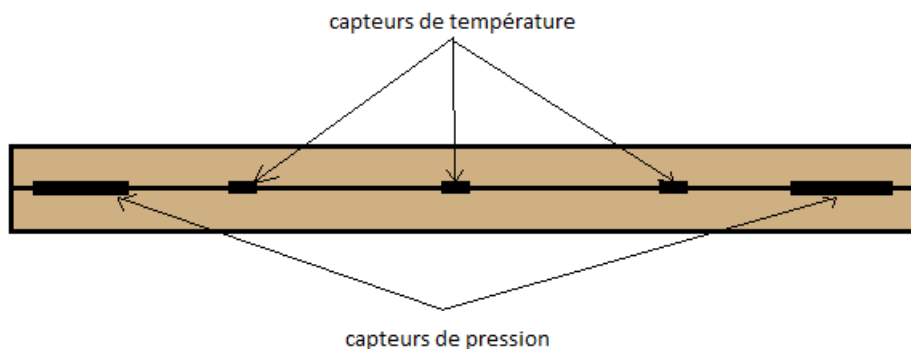


FIGURE 6.1 – Schéma du cadre de mesure. Coupe vue de côté

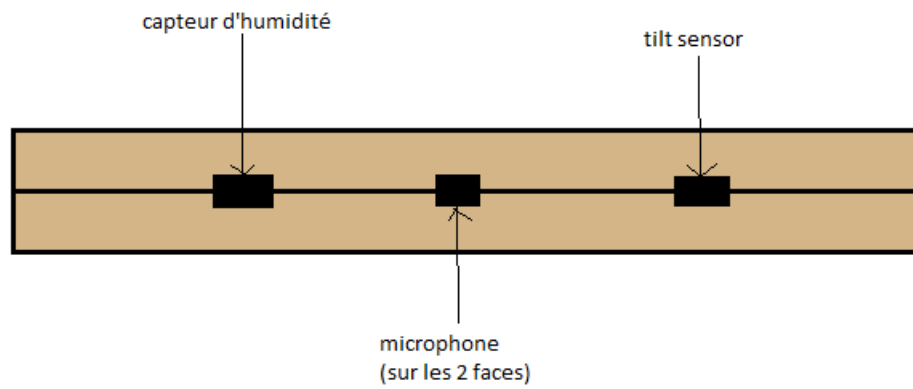


FIGURE 6.2 – Schéma du cadre de mesure. Coupe vue de côté

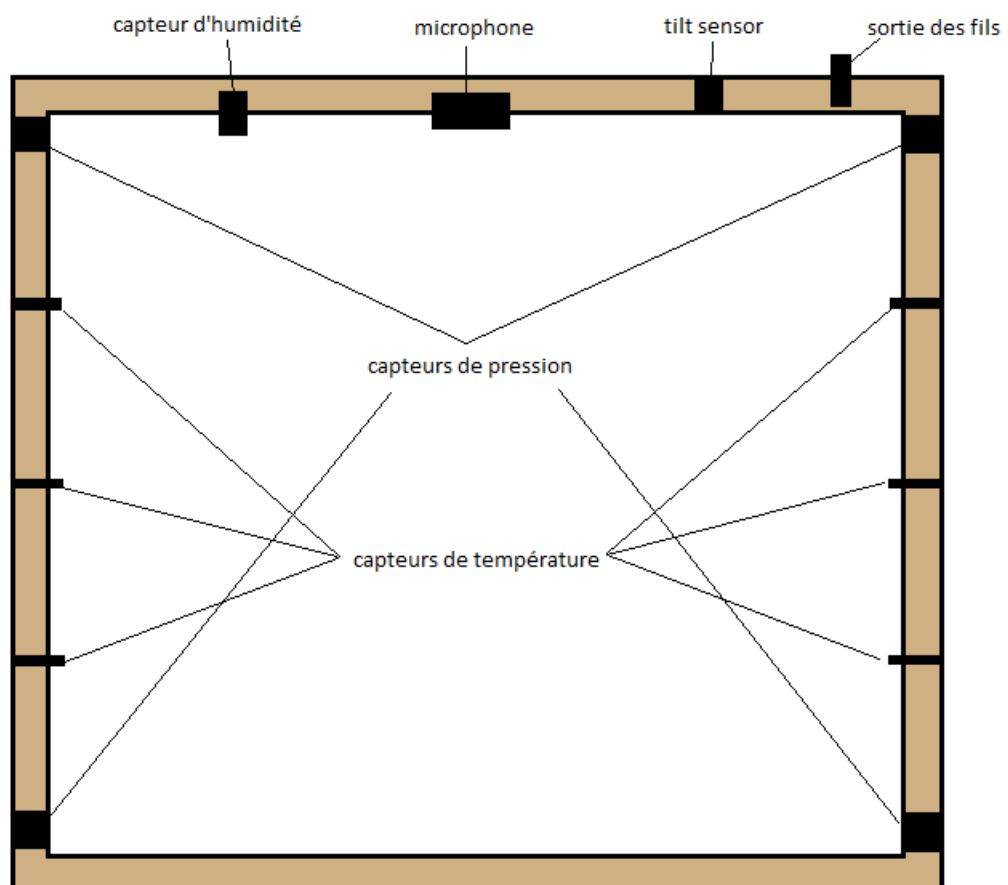


FIGURE 6.3 – Schéma du cadre de mesure. Coupe vue du dessus

## 6.2 Architecture physique logicielle

Dans cette partie nous allons analyser l'architecture logicielle côté logiciel. Elle est divisé 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. Ceci est résumé dans le diagramme ??.

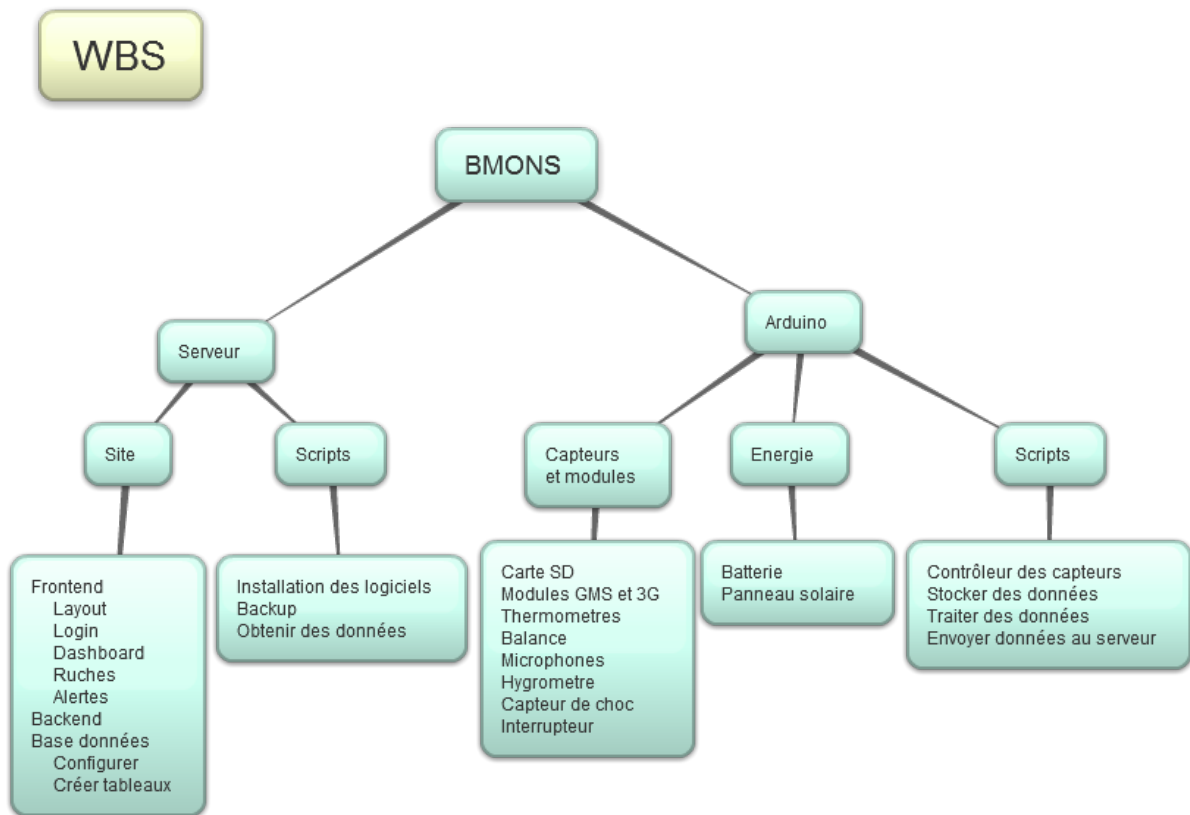


FIGURE 6.4 – Architecture physique logicielle du système BMONS





# Chapitre 7

## Tests unitaires

Test de résistance à la cire d'un capteur de température

Matériel : 3 capteurs de température Cire d'abeilles 1 support

Préparation : Fixer les capteurs de température sur le support espacés de manière régulière sur le capteur de gauche appliquer 0.5 cm de cire sur le capteur de droite appliquer 1 cm de cire le capteur central sera le témoin

Test 1 : Mettre le support au frigo et surveiller l'évolution de la température Relever le temps nécessaire pour arriver à la même température que le témoin pour les 2 capteurs tests

Test 2 : Directement après le test 1 enlever le support du frigo et surveiller l'évolution de la température Relever le temps nécessaire pour arriver à la même température que le témoin pour les 2 capteurs tests

Test 3 : Mettre le support dans un "four" à 30C et surveiller l'évolution de la température Relever le temps nécessaire pour arriver à la même température que le témoin pour les 2 capteurs tests

Test 4 : Directement après le test 3 enlever le support du "four" et surveiller l'évolution de la température Relever le temps nécessaire pour arriver à la même température que le témoin pour les 2 capteurs tests

Test Capteur d'humidité

Matériel : 1 capteur d'humidité des sachets déshydratants 2 boîtes/sachets hermétiques cotons et eau

Préparation : Commencer et finir le cycle avec un capteur propre

Etape 0 :

Relever le niveaux d'humidité d'origine

Etape 1 :

Mettre le support dans réceptacle à 100% Surveiller l'évolution Relever les courbes

Etape 2 :

Mettre le support dans un réceptacle à 0

Etape 3 :

Répéter les étapes 1 et 2 cinq fois avec différentes épaisseurs



# Quatrième partie

## Organisation



# Chapitre 8

## Méthodes de travail

### 8.1 Communication

Tout au long du projet notre méthode de travail a changée. Au fur et à mesure que les séances s'enchaînaient, et en prenant en compte les conseils qui nous ont été donnés (notamment à propos du carnet de bord et des objectifs à court terme) notre méthode de travail a tendu vers la suivante :

De manière générale, toute l'équipe du projet BMONS travaille dans la même salle pour faciliter la communication entre les membres du groupe. Une séance de travail commence par l'ouverture personnelle des mails de chacun, puis le groupe se réunit pour définir les objectifs de la matinée, ensuite chacun choisit la partie il va avancer. Le travail se fait en général seul ou en binôme et des points d'avancement sont faits à l'oral tout au long de la séance. Parfois des tâches comme la prise en main d'un logiciel ou la compréhension d'un diagramme sont faites en dehors des séances, mais la majeure partie du travail s'effectue lors du temps alloué au projet.

Les réunions avec les encadrants et les intervenants extérieurs se déroulent dans des salles de l'ENSTA Bretagne équipées d'un vidéo projecteur et en présence de la totalité de l'équipe BMONS. Ces séances sont organisées à l'avance, les points sur lesquels des précisions sont nécessaires sont mis en avant avant la séance et des questions précises sont préparées. Cela permet de guider la réunion et de ne pas perdre de temps sur des points déjà vus. Chacun a son rôle lors de ces réunions, la prise de notes, le dialogue avec l'intervenant et la rédaction du compte rendu sont ainsi facilités.

### 8.2 Outils pour les échanges

Les outils qui nous ont permis de travailler ensemble et de partager nos fichiers ont également changés au cours du temps. Avant les premiers ateliers techniques, particulièrement celui sur github, nous partagions nos résultats sur le oneDrive d'office 365. Cet outil est efficace pour partager des documents, mais n'est pas optimisé pour le travail simultané de plusieurs membre du groupe sur un même document.

Nous avons donc utilisé github (figure 8.1), malgré une prise en main un peu longue pour ceux d'entre nous qui n'ont pas assisté à l'atelier, github s'est révélé être un outil optimisé pour le travail que nous faisons. Le journal de bord, et les objectifs à cours termes en particulier, sont regroupé dans un wiki (figure 8.2) que chacun peu consulter, les liens utiles au projet y sont également répertoriés. Notre projet étant open source, la totalité de ces informations sont accessible par tous ceux qui souhaitent s'informer sur l'avancement du projet, ainsi nous avons reçus un mail de la part d'une de ces personnes nous invitant à suivre un projet similaire.

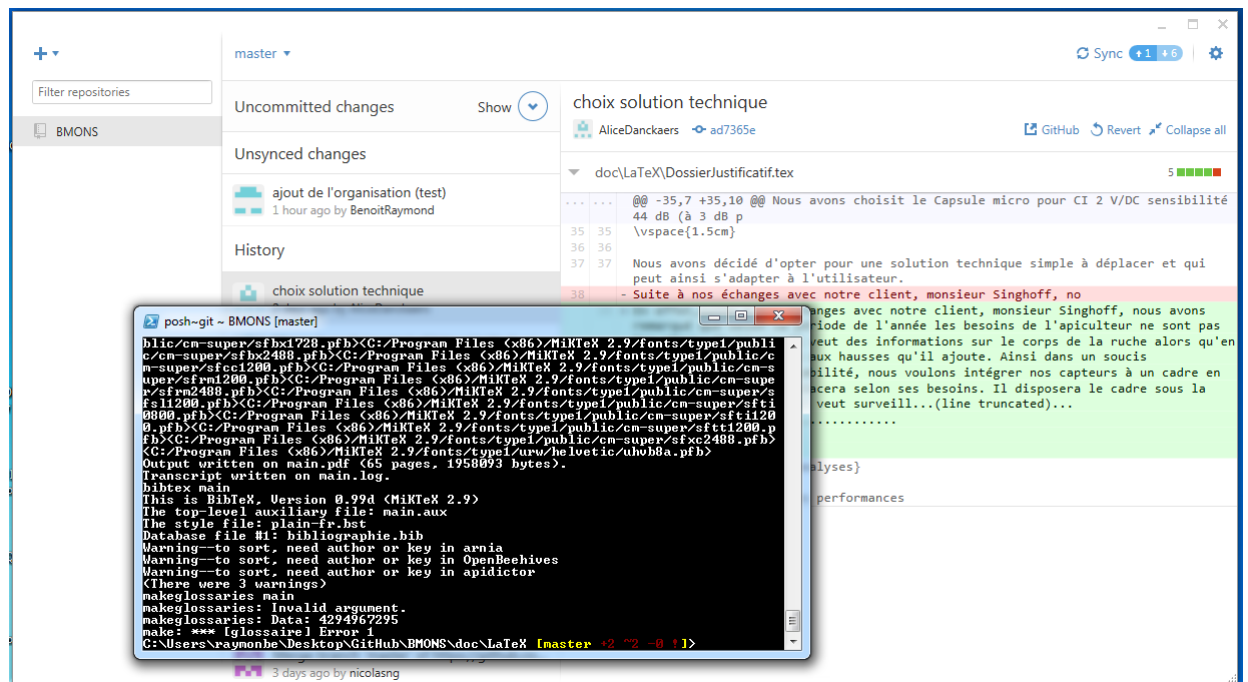


FIGURE 8.1 – interface de travail gitHub

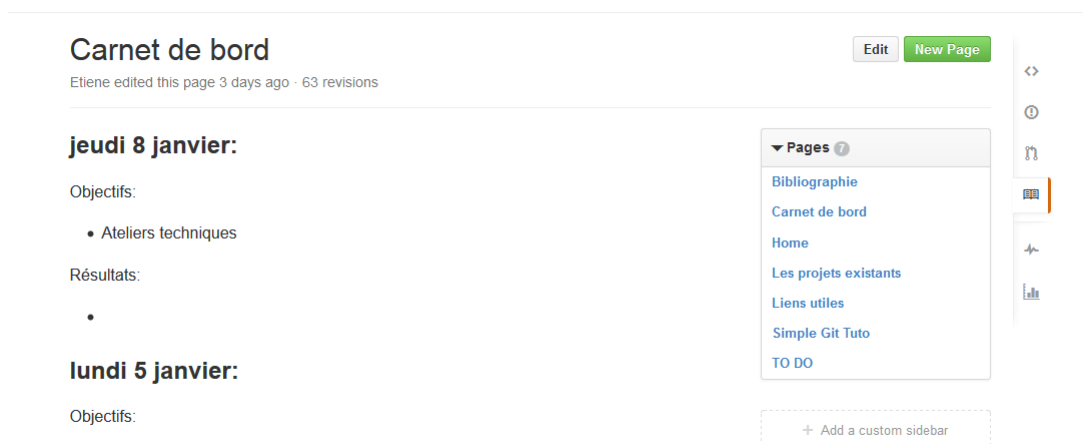


FIGURE 8.2 – Exemple d'utilisation du wiki de gitHub

# Chapitre 9

## Répartition des tâches

### 9.1 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 ce qui est lié au développement du site ainsi que 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.

### 9.2 Diagramme de Gantt

Le diagramme de Gantt est un outil utilisé en gestion de projet permettant de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation qui permet de représenter graphiquement l'avancement du projet par les séances. Les tâches du projet BMONS sont divisés en deux catégories, Serveur et Arduino, comme dans le structure de découpage du projet, en permettant que l'équipe se sépare en deux et partage les tâches en travaillant de façon concomitante.

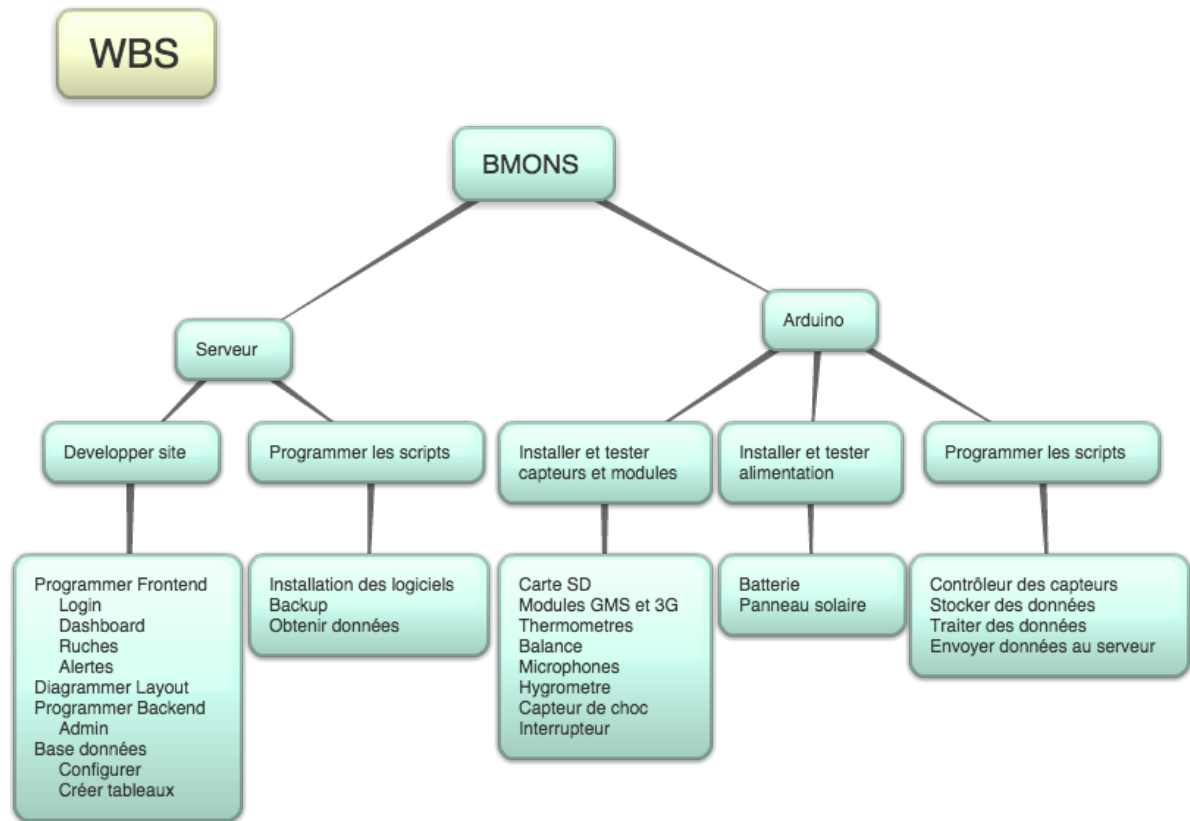


FIGURE 9.1 – Structure de découpage du projet du système BMONS

GANTT2.png

FIGURE 9.2 – Diagramme de Gantt du système BMONS



# 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 boîtier 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 varia-

tions 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 mêmes dimensions de la ruche dans lequel se trouveront 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 renvoie 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 événement étant de plus en plus fréquent. Nous avons choisi le TikerKit Tilt Sensor [10.1](#).

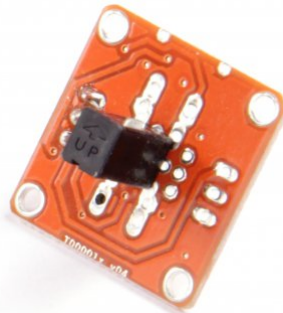


FIGURE 10.1 – TikerKit Tilt Sensor

### 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'événements. 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 prémices d'un essaimage en percevant le champs d'une raine caractéristique de ce type d'événement. 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 choisi le Capsule micro pour CI 2 V/DC sensibilité 44 dB (à 3 dB près) sur la plage 100 - 10000 Hz.

## 10.2 Choix de la carte Arduino

Compte tenu du fait que nous avons besoin d'un nombre d'entrées assez élevé (au moins 20) ainsi que d'un temps ADC supérieur ou égal à 5kHz. Nous avons alors sélectionné quatre cartes, que nous avons ensuite comparées. Ces résultats sont regroupés dans la Figure 10.2.

carte	ADC	nb pin	GSM	processeur	prix	Avantages	Inconvénients
APF28	2MHz	2x80	Apw	454MHz	55 €+ 164€(apw)	ADC rapide	Soudures a effectuer
arduino Mega	10 kHz	55(gpio) 16(analog)	shield (57€)	16MHz	58€ +57€(shield)	librairie fft disponible	pas de GPS avec ce shield
arduino Due	1MHz	54 gpio 12 analog	shield(57€)	64MHz	49€+57€(shield)	ADC rapide	pas de librairie fft
netduino	30MHz	6 analog 22 gpio	intégré	120MHz	30 €	TCP/IP déjà intégré ADC rapide	pas commun, 0°C mini, timing?, besoin de beaucoup de memoire

FIGURE 10.2 – Tableau comparatif des cartes Arduino et équivalents

Nous avons opté pour une carte Arduino Mega 2560, Figure 10.3, compte tenu de l'existence de nombreuses bibliothèques, notamment pour réaliser une fft, et du prix raisonnable. La carte netduino paraissait également très intéressante, car elle ne nécessite pas de shield supplémentaire pour accéder à internet et était trois fois moins chère que les autres. Cependant, comme ce système est peu utilisé, il sera difficile de trouver de la documentation sur internet. Enfin ce système utilise beaucoup de mémoire pour son fonctionnement, limitant ainsi la capacité de stockage. Nous ne pouvions donc pas envisager cette solution et nous avons donc choisi la carte classique Arduino Mega 2560.

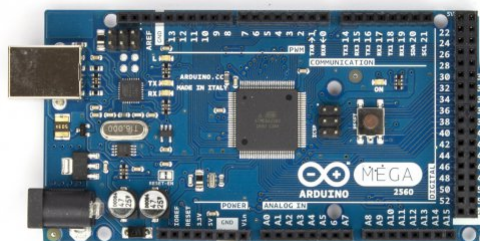


FIGURE 10.3 – Carte Arduino Mega 2560

### 10.3 Choix de la solution technique

Nous avons décidé d'opter pour une solution technique simple à déplacer et qui peut ainsi s'adapter à l'utilisateur. En effet, suite à nos échanges avec notre client, monsieur Singhoff, nous avons remarqué que selon la période de l'année les besoins de l'apiculteur ne sont pas les mêmes. En hiver il veut des informations sur le corps de la ruche alors qu'en été il s'intéresse plus aux hausses qu'il ajoute. Ainsi dans un soucis d'efficacité et de maniabilité, nous voulons intégrer nos capteurs à un cadre en bois que l'apiculteur placera selon ses besoins. Il disposera le cadre sous la partie de la ruche qu'il veut surveiller. Ainsi en hiver, le cadre de mesure sera normalement placé sous la ruche, Figure 10.4, pour vérifier l'activité et les réserves disponibles pour les abeilles. En été, le cadre de mesure sera placé sous une des hausses, Figure ??, logiquement la dernière pour suivre la miellée.



Luberon Apica  
L'univers du Miel - www.apic

FIGURE 10.4 – Position standard du cadre de mesure en hiver



Luberon Apiculture  
L'univers du Miel - www.apiculture.net

FIGURE 10.5 – Position standard du cadre de mesure en été

# 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

	expresion de la fonction metier	exigences	Performance
exi met 1	suivre la miellée	Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 2/jour
		Avertir l'apiculteur en cas de diminution significative du poids	
		Transmettre les données par ondes radio	3G ou 4G
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 2	détecter la présence d'abeille dans la ruche	Relever le bruit dans la ruche	
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
		Transmettre les données par ondes radio	3G ou 4G
exi met 3	suivre le trafic	Mesurer le poids de la ruche	
		Relever le bruit dans la ruche	
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
		Transmettre les données par ondes radio	3G ou 4G

FIGURE A.1 – Fonctions métiers du système (1/4)

exi met 4	disposer d'un tableau de bord	Mesurer la température dans la ruche	[-5 ; 50] résolution 1°C fréquence 1/heure
		Mesurer l'humidité dans la ruche	[0 ; 100] résolution 1% fréquence 1/heure
		Relever la température extérieure et la pluviométrie	[-15 ; 50] résolution 1°C fréquence 1/heure
		donner la position de la ruche	
		Afficher les données	
		Informier l'apiculteur sur l'alimentation électrique et l'état des capteurs	
		Pouvoir accéder rapidement à l'information	
		Donner la possibilité aux utilisateurs de configurer le niveau de confidentialité	
		Donner la possibilité aux utilisateurs de configurer les informations à afficher	
		détecter la position de la ruche	
		Transmettre les données par ondes radio	3G ou 4G
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
		Avertir l'apiculteur en cas de détection d'essaimage	
		Reconnaitre le chant des futurs reines	

FIGURE A.2 – Fonctions métiers du système (2/4)

exi met 5	detecter les premise de l'essaimage	Transmettre les données par ondes radio	3G ou 4G
		Pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 6	detecter l'essaimage	Avertir l'apiculteur en cas de détection d'essaimage	
		Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 1/heure
		Transmettre les données par ondes radio	3G ou 4G
		pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 7	veiller à la sécurité de la ruche	detecter les renversements	
		detecter l'ouverture de la ruche	
		donner la position de la ruche	
		Avertir l'apiculteur en cas de choc	
		Détecter la position de la ruche	
		Transmettre les données par ondes radio	3G ou 4G
		Pouvoir stocker les données localement	

FIGURE A.3 – Fonctions métiers du système (3/4)



		Déclancher des mesures à intervalles réguliers	
exi met 8	detecter la position de la grappe	Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 2/jour
		Relever le bruit dans la ruche	
		Transmettre les données par ondes radio	3G ou 4G
		pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 9	surveiller la consommation des réserves en hiver	Mesurer le poids de la ruche	[0 ; 150] résolution 100g fréquence 2/jour
		Transmettre les données par ondes radio	3G ou 4G
		pouvoir stocker les données localement	
		Déclancher des mesures à intervalles réguliers	
exi met 10	être adaptable	Avoir deux modes de fonctionnement (été/hiver)	
		Donner la possibilité aux utilisateurs de configurer les alertes	

FIGURE A.4 – Fonctions métiers du système (4/4)



## 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] Beewise. <http://www.beewise.eu/ruche/surveillance-ruches-sms-equipement-apiculture.php>. Accessed : 2014-12-01.
- [4] Open source beehive. <http://www.opensourcebeehives.net/>. Accessed : 2014-12-01.
- [5] AFIS : Accueil - notre métier : L'ingénierie système. <http://www.afis.fr/nm-is/default.aspx>, 2010. Accédé le 30 août 2014.
- [6] Denis BITOUZÉ et Jean-Côme CHARPENTIER : *LATEX, l'essentiel*. Pearson Education France, 2010.
- [7] Bernard DESGRAUPES : *LATEX : apprentissage, guide et référence*. Vuibert informatique, 2003.
- [8] fablab LANNION : Bzzz : Suivi des ruches. [http://fablab-lannion.org/wiki/index.php?title=Suivi\\_des\\_ruches](http://fablab-lannion.org/wiki/index.php?title=Suivi_des_ruches). Accessed : 2014-12-01.
- [9] Meyer KEN : Honeybee hive monitoring. <http://hackaday.io/project/1741-honeybee-hive-monitoring>. Accessed : 2014-12-01.
- [10] Leslie LAMPORT : *LT<sub>E</sub>X—A Document*, volume 410. pub-AW, 1985.
- [11] Noël-Arnaud MAGUIS : *Rédigez des documents de qualité avec LaTeX*. Livre du Zéro, 2010.
- [12] 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.
- [13] Fabrizio SEBASTIANI : Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.





