

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

Remerciements	i
Préambule	iii
0.1 Comment compiler ce document ?	iii
0.2 Références internes	iii
0.2.1 Tableaux et figures	iv
0.2.2 Codes	iv
0.2.3 Index et glossaire	v
0.3 Références bibliographiques	v
I Introduction au projet	1
1 Formulation initiale du projet	3
1.1 Contexte	3
1.2 Expression initiale du besoin	3
2 État de l’art	5
II Dossier fonctionnel	7
3 Ingénierie des exigences	9
3.1 Approche Top-Down	9
3.2 Approche Bottom-Up	10
3.3 Fonctions principales du système	10
4 Spécification fonctionnelle 3 axes	11
4.1 Raffinement FAST	11
4.2 Spécification des données	13
4.3 Modèle de Données	14
4.4 Spécification des comportements	15
5 Architecture fonctionnelle	21

III	Organisation	23
6	Méthodes de travail	25
7	Outils pour les échanges	27
8	Répartition des tâches dans le temps	29
IV	Journal du projet	31
9	Choix et justifications	33
10	Résultats et analyses	35
11	Conclusion	37
V	Annexes	39
A	Première annexe	41
B	Deuxième annexe	43
C	Troisième annexe	45
	Bibliographie	47
	Index	49
	Glossaire	49

Remerciements

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

Emil Cioran

Je tiens à remercier tous les contributeurs de L^AT_EX qui nous permettent aujourd'hui de produire des documents de qualité professionnelle sans avoir à se préoccuper de son apparence : des livres, des articles, des mémentos dans presque toutes les langues, mais aussi de la musique et des dessins. Ce logiciel ne connaît pas de limites.

Préambule

Le chemin est long du projet à la chose.

Molière

0.1 Comment compiler ce document ?

Un document \LaTeX peut se compiler au travers d'un IDE (TexStudio, TeXMaker par exemple). Le répertoire de ce document contient également un Makefile qui permet de compiler simplement en ligne de commande. La fabrication du glossaire et de l'index est prise en charge par ce Makefile. Pour l'utiliser, il suffit d'ouvrir un terminal, de se placer dans le répertoire du document puis d'invoquer la commande `make`.

Voici les différentes cibles disponibles pour ce Makefile :

<code>make</code>	- construit le document
<code>make all</code>	- construit le document
<code>make index</code>	- construit l'index
<code>make glossaire</code>	- construit le glossaire
<code>make bib</code>	- construit la bibliographie
<code>make pdf</code>	- construit le document PDF
<code>make clean</code>	- supprime les fichiers LaTeX intermédiaires
<code>make clean-all</code>	- supprime tous les fichiers générés par la compilation
<code>make help</code>	- cette information

0.2 Références internes

Les références internes sont des renvois vers des figures, des tableaux ou des sections du rapport. \LaTeX introduit un mécanisme simple pour établir ce genre de référence, via les commandes `\label` et `\ref`. La première sert à définir une ancre dans le document, la seconde à la citer. Voici par exemple une référence interne vers la section intitulée *Approche Top-Down* (cf. section ??). Ce renvoi est le résultat de la commande `\ref{sec:top-down}`. Si vous vous rendez dans le corps de cette section, vous y trouverez le label en question `\label{sec:top-down}`.

0.2.1 Tableaux et figures

Les figures et les tableaux sont référencés de la même la manière (cf. figure 1 et tableau 1).

Algorithmes	Performance (s)	Gain (dB)
Algorithme 1	0.0003262	0.562
Algorithme 2	0.0015681	0.910
Algorithme 3	0.0009271	0.296

TABLE 1 – Performances et gains des algorithmes envisagés.

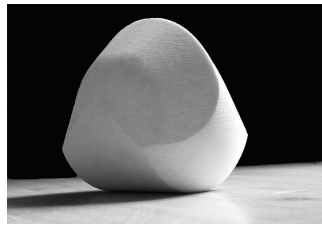


FIGURE 1 – Gömböc : un objet homogène tridimensionnel mono-monostatique. (source : Wikipedia)

0.2.2 Codes

Si vous souhaitez insérer du code dans votre rapport, invoquez les commandes :

```
\lstset{language=python}
\lstinputlisting[caption={Titre du listing}, label={lst:code}]{./code/code.py}
```

La première commande sélectionne le langage, pour que les mots clés de celui-ci soient correctement détectés et mis en valeur. La seconde commande permet d'insérer le code contenu dans le fichier code.py qui se trouve dans le sous-répertoire code. Pour faire référence au code, il suffit de sélectionner le label du listing 1, comme pour les figures et les tableaux.

```
1 import itertools

def iter_primes():
    # an iterator of all numbers between 2 and +infinity
    numbers = itertools.count(2)

6     # generate primes forever
    while True:
        # get the first number from the iterator (always a prime
    )

        prime = numbers.next()
11     yield prime

    # this code iteratively builds up a chain of
    # filters...slightly tricky, but ponder it a bit
    numbers = itertools.ifilter(prime.__rmod__, numbers)
```


16

```

for p in iter_primes():
    if p > 1000:
        break
    print p

```

Listing 1 – Titre du listing

0.2.3 Index et glossaire

Pour insérer des entrées dans l’index, il suffit de déclarer des mots via la commande `\index{Fabrication d’un index}` comme suit ¹.

Pour utiliser le glossaire, il faut définir les termes dans le fichier `glossaire.tex` en utilisant la commande `\newacronym{label}{abbréviation}{Signification}`. Puis, `\gls{label}` permet de les utiliser dans le document.

Par exemple, les UVs 3.4 et 4.4 sont une initiation à l’[Ingénierie Système \(IS\)](#). Un concept de gestion de projet souvent mal connu est le [Work Breakdown Structure \(WBS\)](#).

0.3 Références bibliographiques

Les références bibliographiques sont des documents numériques, des livres, des articles, des images ou des vidéos qui ne sont pas présents dans le rapport. \LaTeX propose un mécanisme simple de citation. Pour plus de détails, vous pouvez consulter les références suivantes [?, ?, ?] qui sont présentes à la médiathèque de l’ENSTA Bretagne, ou celle-ci directement sur le web [?].

Pour citer des documents, il suffit d’appeler la commande `\cite{key}` en choisissant la clé qui identifie le document, comme suit : [?]. Cette clé de citation est celle qui référence l’ouvrage dans le fichier de bibliographie intitulé `bibliographie.bib`. Ce fichier d’exemple contient tous les types de documents dont vous aurez besoin : livre, article de journal, références web, rapport... Une fois insérée et compilée, la citation devient un lien dans le fichier pdf, redirigeant ainsi directement vers le détail de l’ouvrage cité dans la bibliographie située à la fin du document.

1. Allez donc voir l’index [C](#) à la fin du document !

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 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'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 [?] 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 [?] 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 [?], 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 [?]. 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.

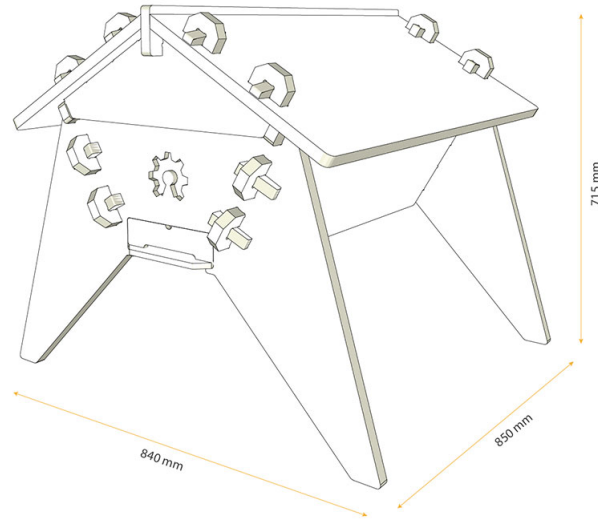


FIGURE 2.2 – Modèle de ruche Open Source Beehive

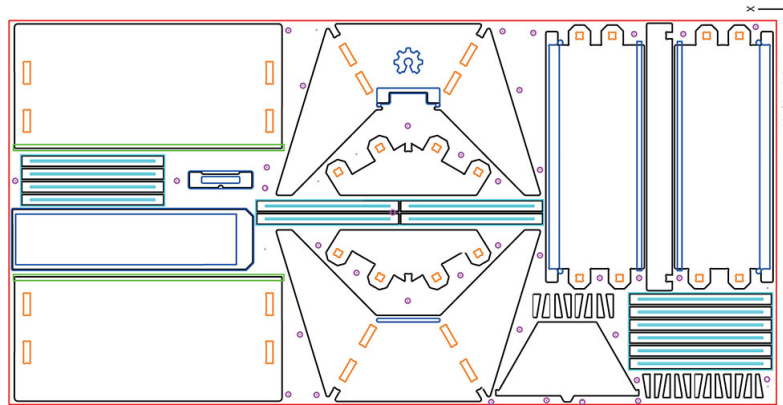


FIGURE 2.3 – Plan de la ruche Open Source Beehive

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.

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.

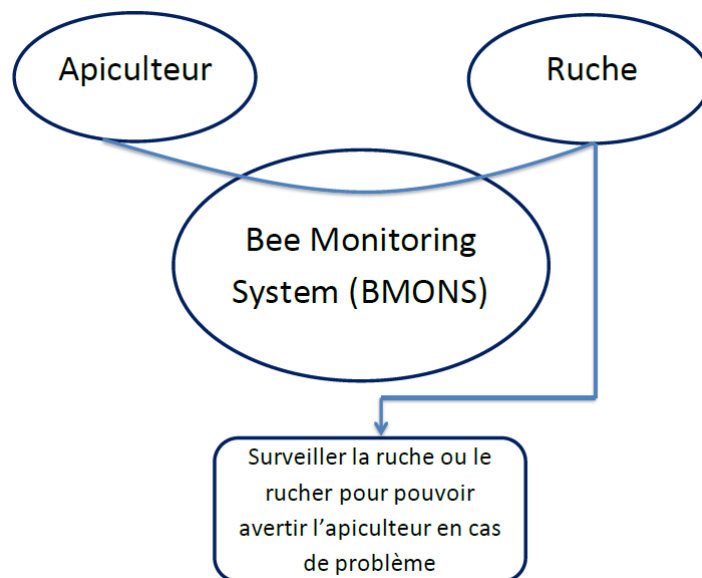


FIGURE 3.1 – Diagramme "bête à cornes" du système BMONS

3.2 Approche Bottom-Up

3.3 Fonctions principales du système

Les fonctions principale, services et contraintes du système sont regroupées dans ce diagramme pieuvre, figures 3.2 et 3.3 . Les buts et les contraintes imposées au système y sont également représentés.

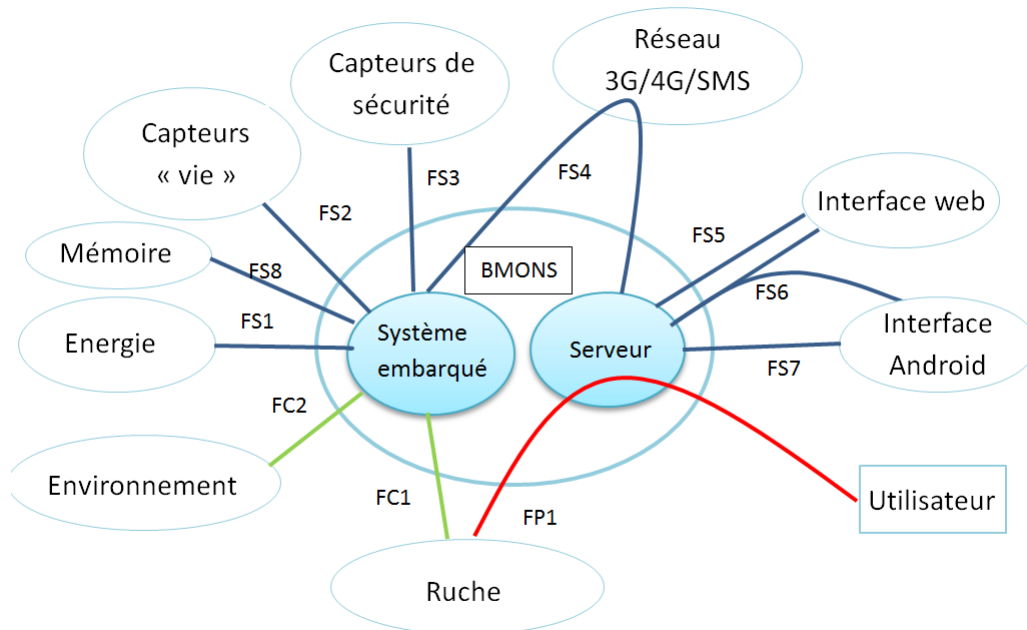


FIGURE 3.2 – Diagramme pieuvre du système BMONS

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

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](#)

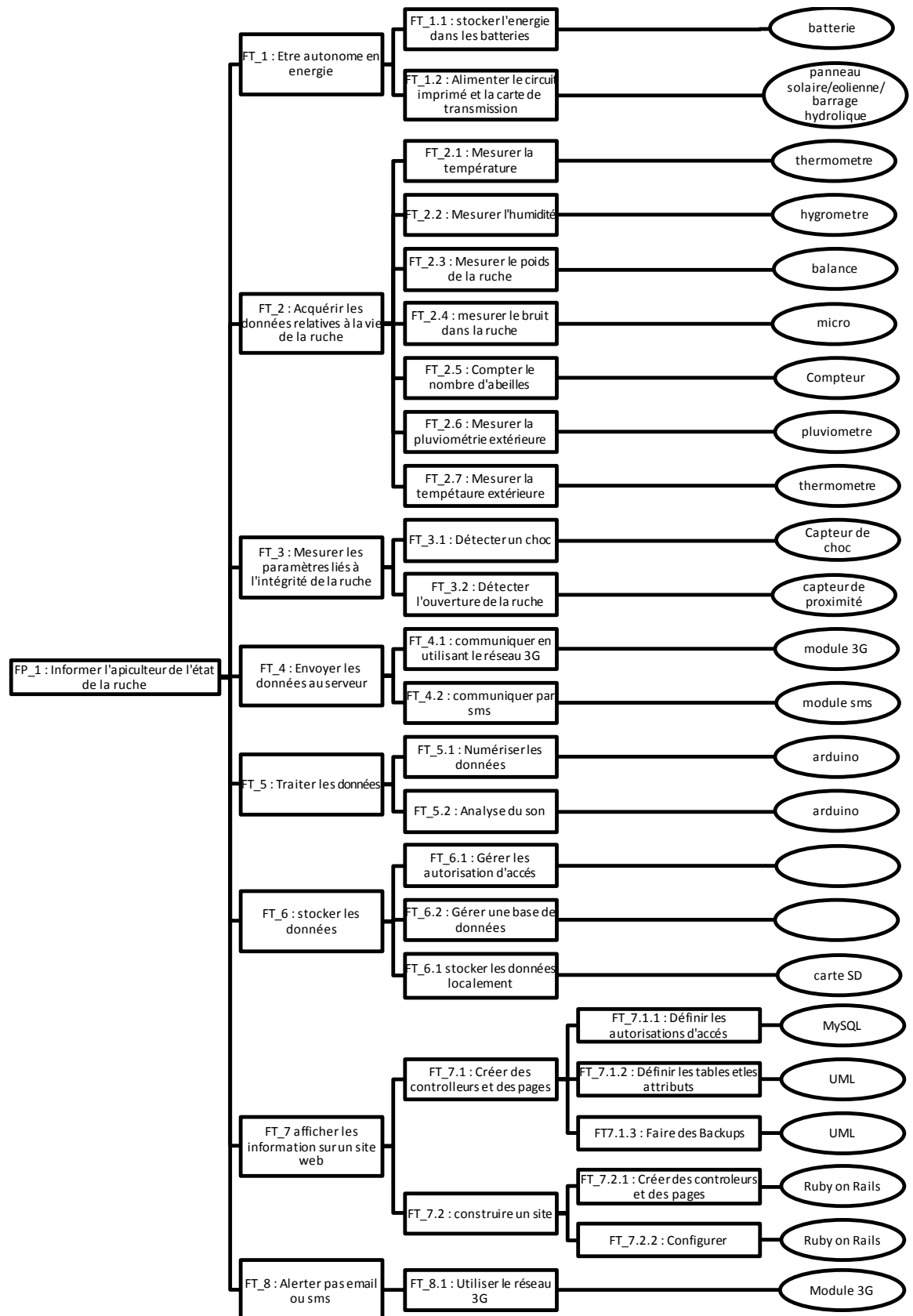


FIGURE 4.1 – Diagramme FAST du système BMONS

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

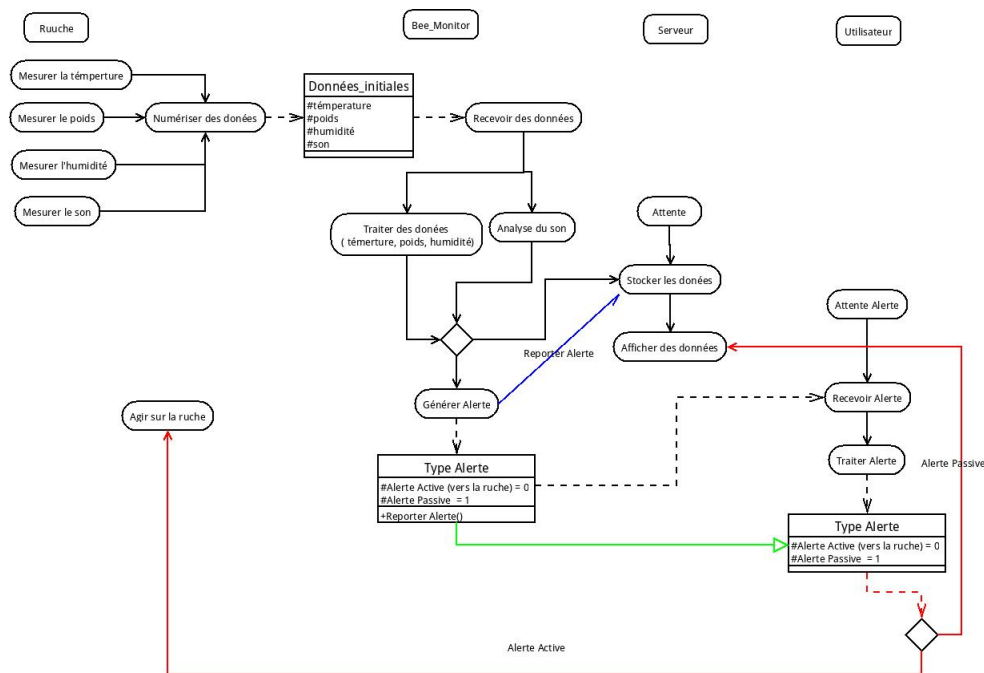


FIGURE 4.2 – Diagramme d'activité du flux de données

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 d'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

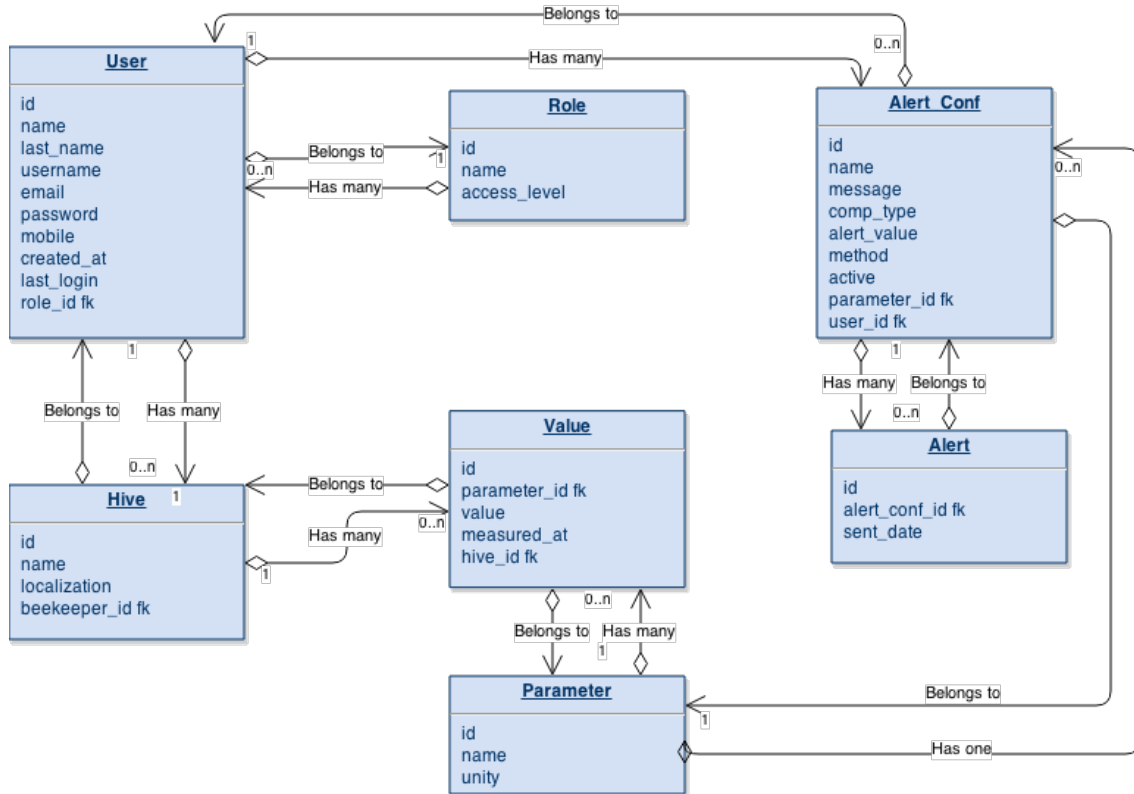


FIGURE 4.3 – Modèle de données pour le système BMONS

4.4 Spécification des comportements

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

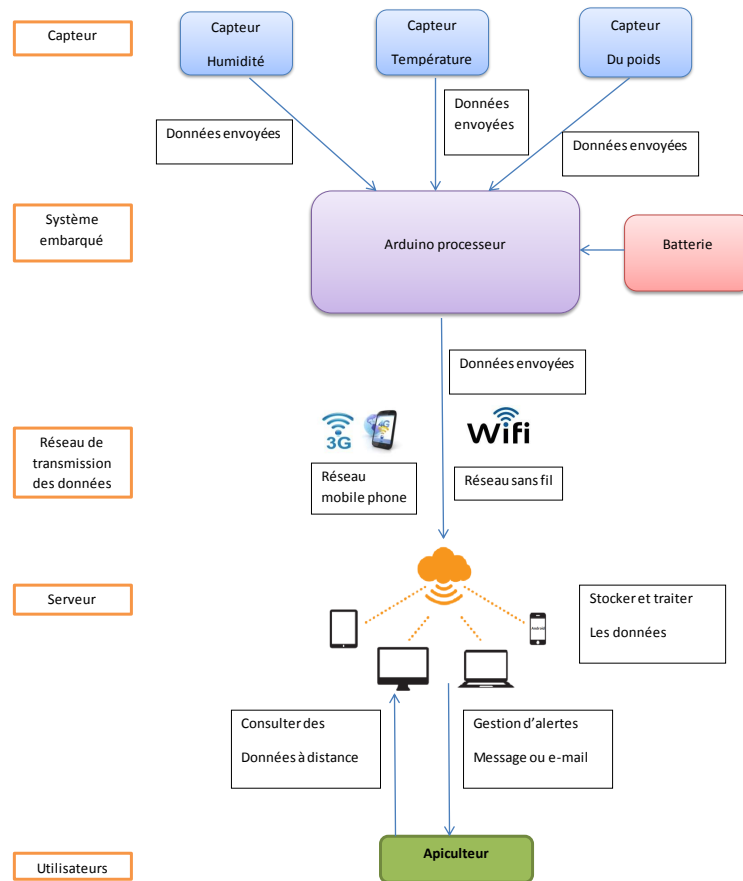


FIGURE 4.4 – Diagramme de spécification des comportements

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 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 ce fait également grace aux diagrammes de séquence. Ces diagrammes expliquent précisément comment les different parties du systeme interagissent dans le but d'exécuter une action. Ils détail l'ordre des actions ainsi que leur nature et leur durée d'exécution. A ce jour 5 diagrammes de séquence ont été réalisé ils concernent l'action de récupération de la mémoire flash, l'analyse des information par le serveur, le service permettant à l'apiculteur d'avertir le serveur d'une intervention sur une ruche, la fonction permettant la configuration des alertes et l'action de connexion d'un apiculteur à son compte.

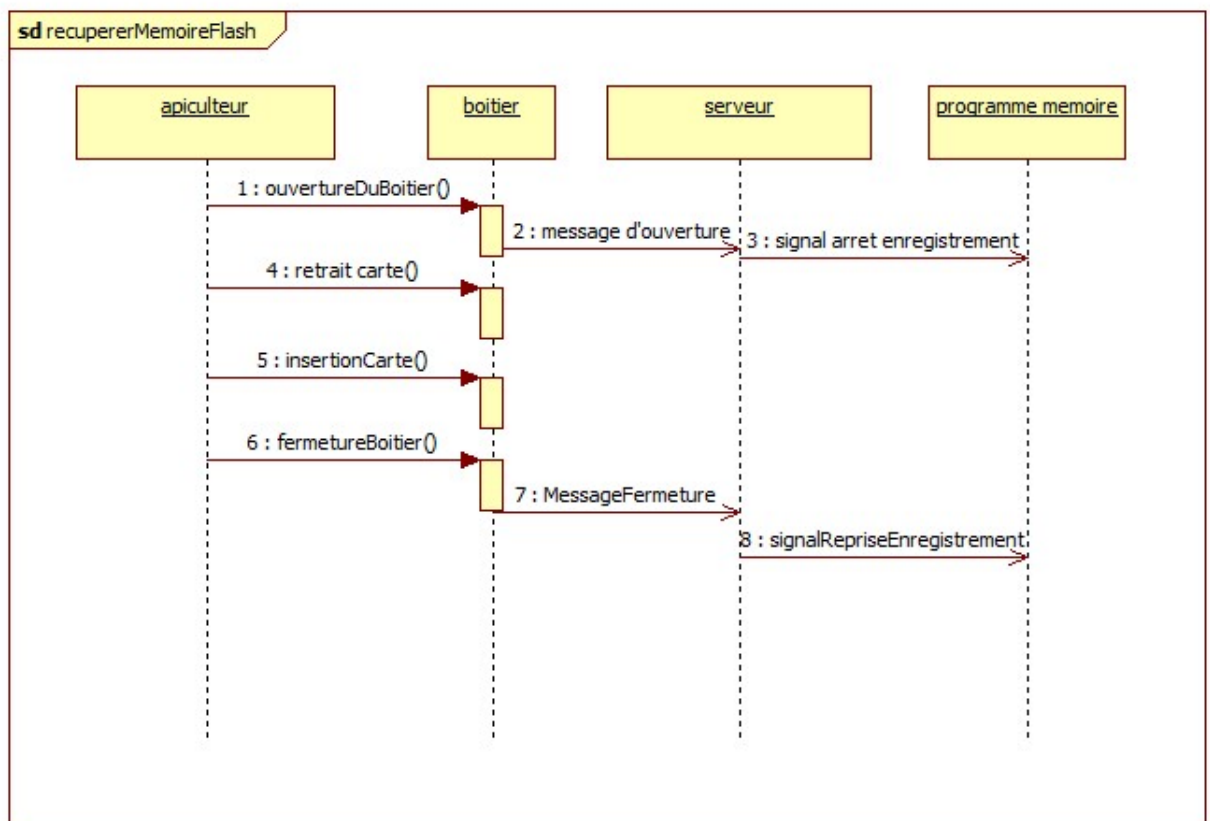


FIGURE 4.5 – Diagramme de spécification des comportements

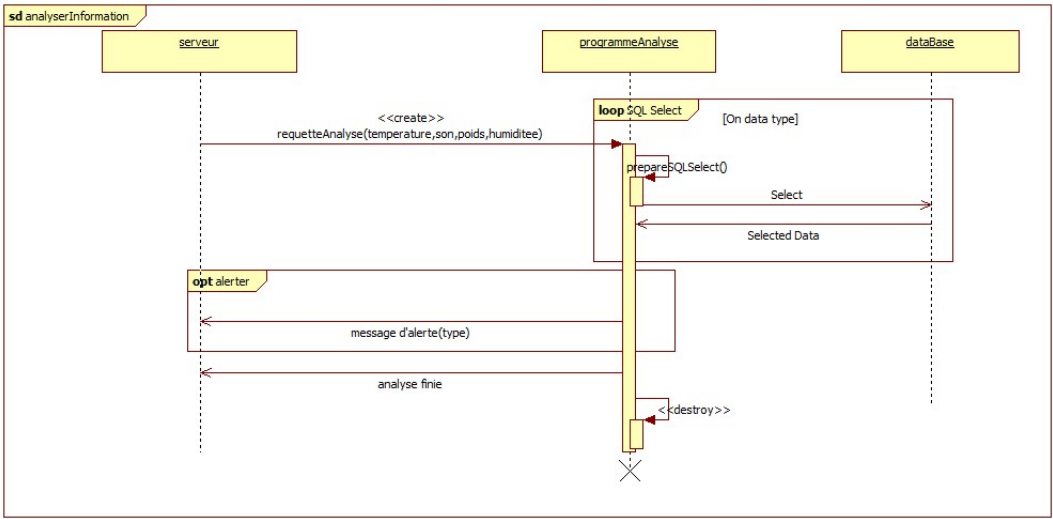


FIGURE 4.6 – Diagramme de spécification des comportements

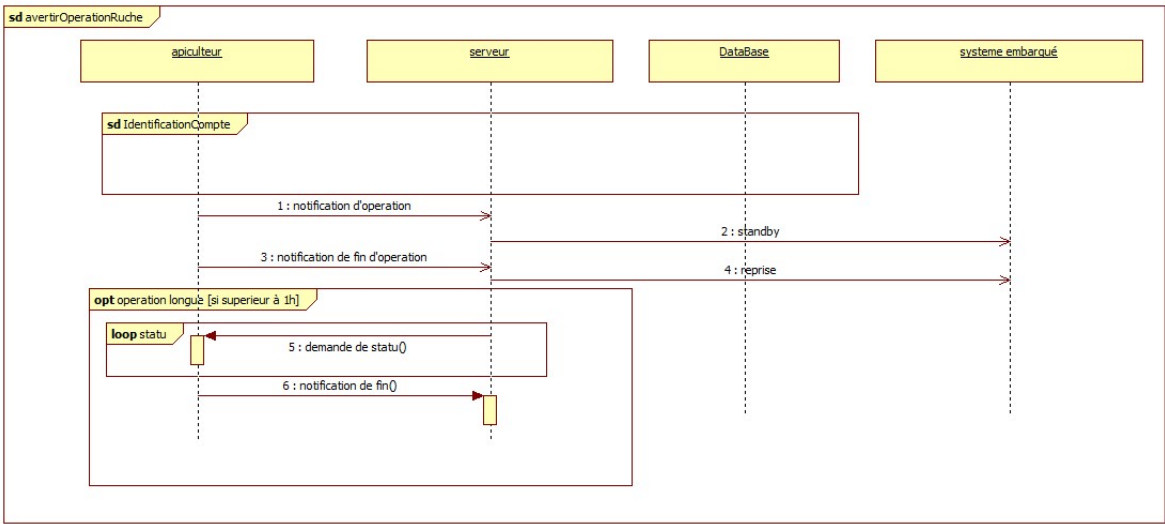


FIGURE 4.7 – Diagramme de spécification des comportements

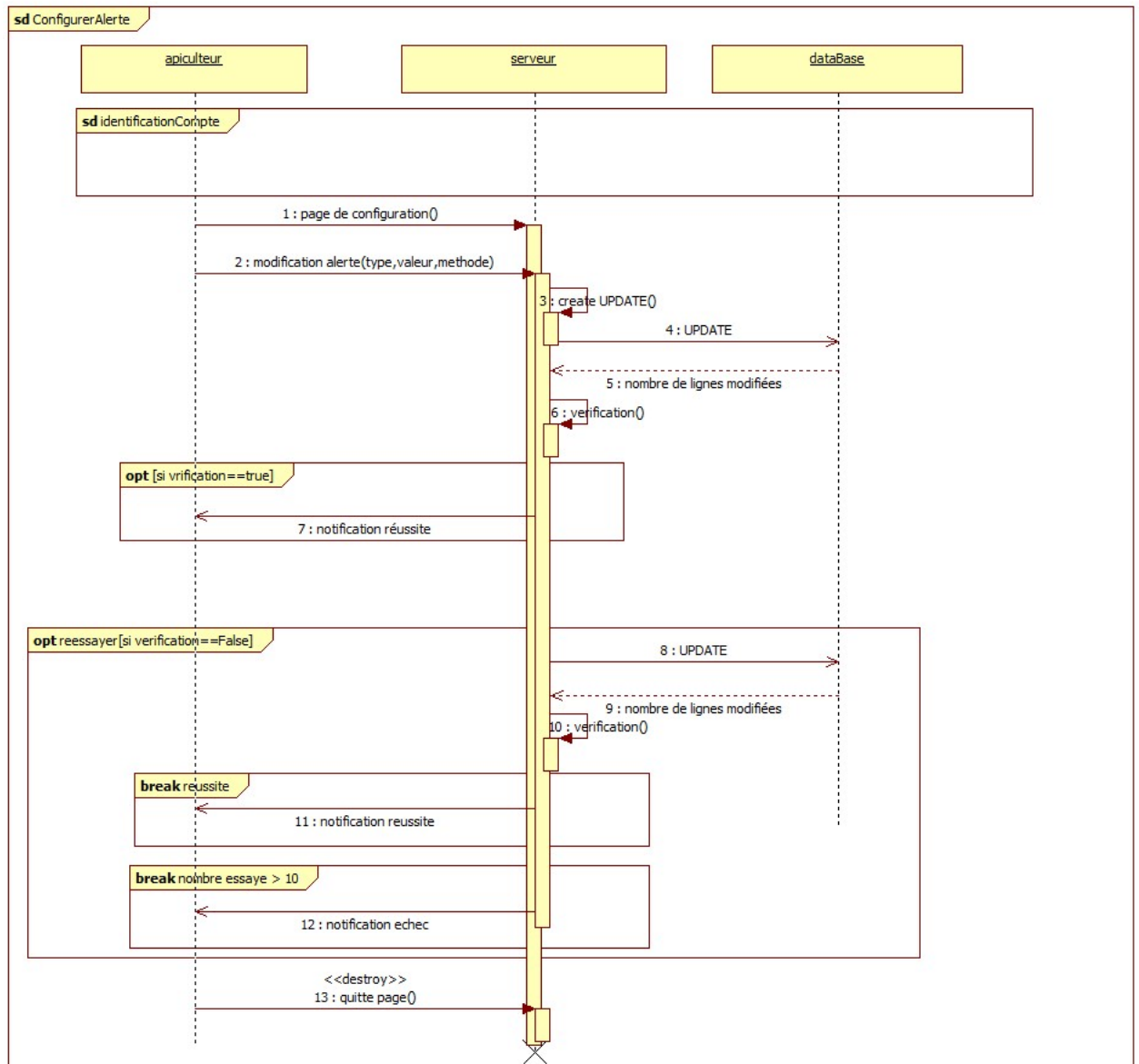


FIGURE 4.8 – Diagramme de spécification des comportements

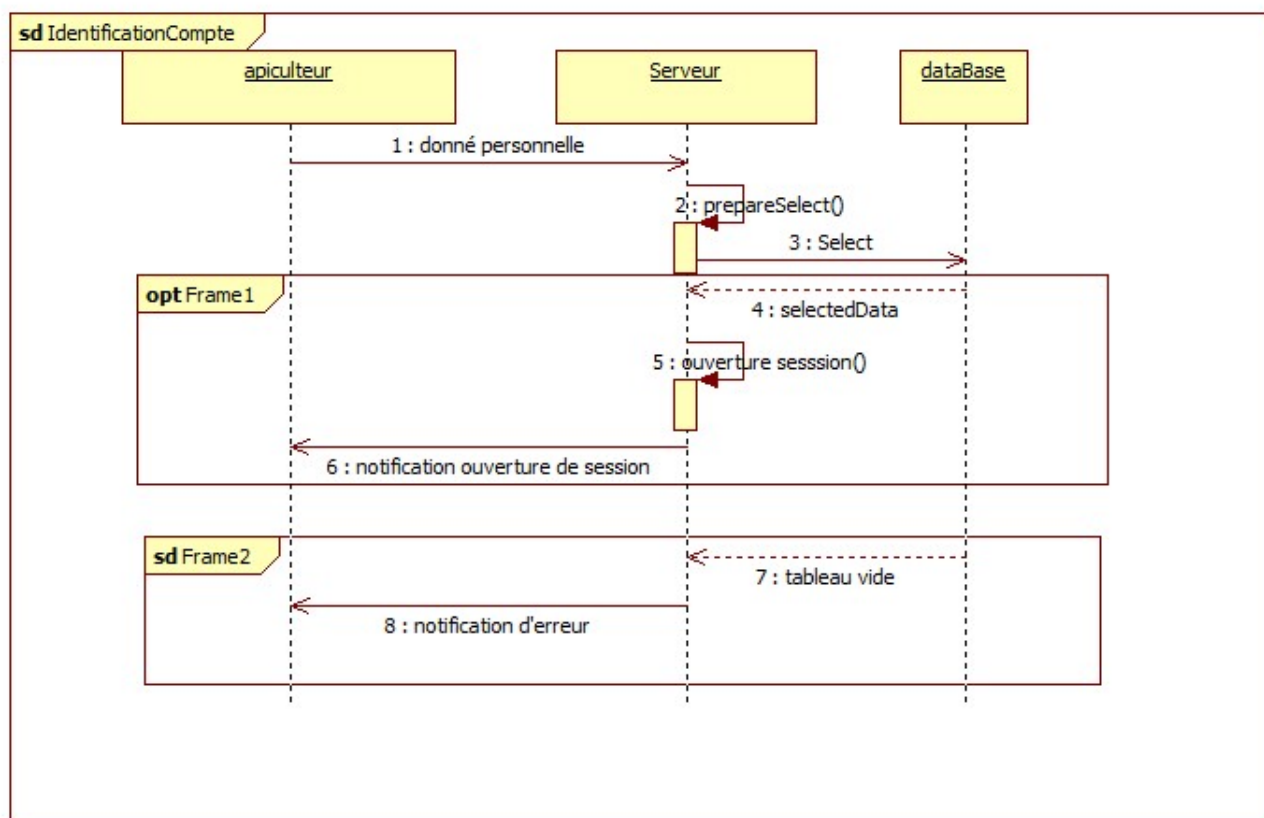


FIGURE 4.9 – Diagramme de spécification des comportements

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.

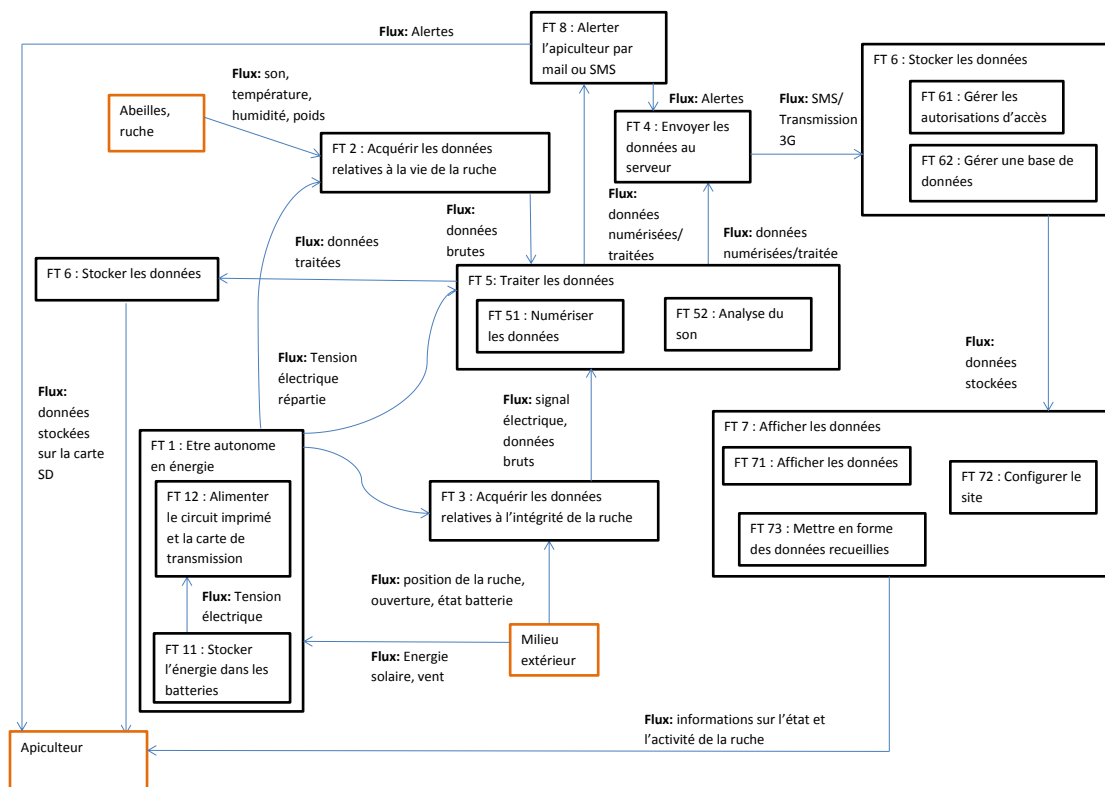


FIGURE 5.1 – Architecture fonctionnelle

Troisième partie

Organisation

Chapitre 6

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 7

Outils pour les échanges

Quels sont les outils qui nous permettent de travailler ensemble ?

Chapitre 8

Répartition des tâches dans le temps

WBS et diagramme de Gantt

Quatrième partie

Journal du projet

Chapitre 9

Choix et justifications

détails techniques et justification du choix des architectures cheminement du projet, évolution

Chapitre 10

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 11

Conclusion

Cinquième partie

Annexes

Annexe A

Première annexe

Annexe B

Deuxième annexe

Annexe C

Troisième annexe

Références bibliographiques

