|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activités | Semaine | Semaine | Semaine | Semaine | Semaine | Semaine | Semaine | Semaine | Semaine | Semaine |
| Attribution du sujet | TOUS |  |  |  |  |  |  |  |  |  |
| Brainstorming | TOUS |  |  |  |  |  |  |  |  |  |
| Bibliographie |  |  |  |  |  |  |  |  |  |  |
| Contact avec l’encadrant |  |  |  |  |  |  |  |  |  |  |
| Conduit de projet |  |  |  |  |  |  |  |  |  |  |
| Définition des exigences |  |  |  |  |  |  |  |  |  |  |
| Transposition des exigences |  |  |  |  |  |  |  |  |  |  |
| Définition des fonctions du système |  |  |  |  |  |  |  |  |  |  |
| Graphe des interactions |  |  |  |  |  |  |  |  |  |  |
| Précision des fonctions (FAST) |  |  |  |  |  |  |  |  |  |  |
| Détermination des flux du système |  |  |  |  |  |  |  |  |  |  |
| Ingénierie système |  |  |  |  |  |  |  |  |  |  |
| État de l’art |  |  |  |  |  |  |  |  |  |  |
| Rédaction du rapport |  |  |  |  |  |  |  |  |  |  |

## **Répartition des tâches :**

Figure 1: Diagramme de Gantt

[4. Ingénierie système](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940168)

4. Cycle de vie d’un système :

Figure : Cycle de vie d’un système

Un système est l’ensemble des éléments interconnectés permettant d’accomplir une mission définie.

Notre système est de type :

* Ouvert
* Artificiel
* Physique
* Connu

La phase de **conception** présente les comportements suivants :

* Expression de besoin
* Identification d’exigences
* Architecture fonctionnelle
* Architecture organique

La phase de **réalisation** présente les comportements suivants :

* Production
* Intégration
* Vérification
* Validation
* Qualification

La phase de **déploiement** présente les comportements suivants :

* Utilisation
* Maintenance
* Mise à jour

La phase de **fin de vie** présente les comportements suivants :

* Démantèlement
* Re-ciblage
* Recyclage

4. Cycle en V :

Phase descendante :

Figure 2: Les points de base de la phase de conception d’un système

* Les besoins représentent les capacités opérationnelles souhaitées par le client.
* Les exigences représentent structurellement et contractuellement les capacités opérationnelles souhaitées par le client et validées par le fournisseur.
* L’architecture fonctionnelle conduit à mettre en place les référentiels d’informations, à exprimer les services métiers dans l’architecture et à identifier les interfaces entre les sous-ensembles fonctionnels.
* L’architecture physique est liée au choix des matériaux dans notre système.

Phase montante :

Figure : Le système IVVQ

* L’intégration c’est le fait de tester les parties assemblées du système dans l’ensemble, suite à un test unitaire (test du bon fonctionnement de chaque composant à part). (Desmoulins, 2009)
* La vérification c’est le fait de savoir si le système est construit correctement et satisfait les conditions de départ.
* La validation c’est le fait de savoir si nous avons conçu le bon système et respecter les exigences pour atteindre la mission voulue.
* La quantification c’est le fait d’assurer formellement que le produit est conforme aux spécifications fonctionnelles. (Hassib, 2015)

4. Besoins de l’entreprise Novintec :

[4.1. Exigences](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940169)

Afin de distinguer les différents exigences de notre système, nous avons pris en compte une méthodologie qui consiste à identifier les exigences dans un premier temps, puis dans un deuxième temps à étudier les caractéristiques de chacun d’entre eux. Pour cela, notre étude est représentée sous forme d’un tableau qui implémente chaque exigence avec ses propres caractéristiques, ainsi que sa relation avec les autres exigences si besoin est. Cette méthode nous amène enfin à identifier la validation d’une telle exigence par rapport aux conditions imposées par notre projet (annexe 1 : exigences du système). Nous les avons ensuite triées par :

* Identifiant
* Exigences
* Type
* Priorité
* Vérification
* Raison d'être
* Relation avec d'autres exigences
* Risques

[4.2. Analyse fonctionnelle](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940170)

Une fois les exigences mises en place, il est le temps d’établir la stratégie de l’analyse fonctionnelle, ce qui donne naissance à une étude consacrée aux fonctions du système. Afin de surmonter notre problème, nous avons utilisée la méthode APTE (application aux techniques d’entreprise), universellement utilisée dans le domaine de la conduite de projet.

4.2.1 Analyse fonctionnelle externe

[4.2.1.1 Le diagramme des prestations](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940171)

Nous avons réalisé un diagramme des prestations dit diagramme « bête à cornes » afin de définir l’expression du besoin :

Projet CCM

Figure 3: diagramme "bête à cornes"

[4.2.1.2 Le graphe des interactions](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940172)

Le diagramme  « pieuvre » ci-dessous est un graphique qui montre les interactions entre les fonctions du système ; elle renferme essentiellement les fonctions principales et les fonctions contraintes :

FP 1 : Le système doit informer l’utilisateur de l'état des vérins

FP 2 : Le système doit différencier les utilisateurs selon plusieurs niveaux

FP 3 : Le système doit avoir une application

FC 1 : Le système doit être géré facilement par l'utilisateur

FC 2 : L'utilisateur doit être capable de surveiller l'état du système

FC 3 : Le système doit respecter la réglementation relative à l’aviation et doit s’adapter à l’environnement

FC 4 : Le système doit respecter la confidentialité des informations

FC 5 : Le système doit avoir une application qui doit fonctionner sur une tablette Android

FC 6 : Le système doit être capable de lire les différentes données depuis les capteurs

FC 7 : Le système doit respecter les contraintes budgétaires

FC 8 : Le système doit s'adapter à une source d'énergie électrique

Toutes ces fonctions sont déclinées en sous-fonctions en annexe (Annexe 2 : Description des fonctions).

**FP3**

**FP1**

**FC8**

**FC1**

**FC2**

**FC3**

**FC7**

**FC4**

**FP2**

**FC6**

**FC5**

Figure 4 :diagramme "pieuvre"

[4.2.2 Analyse](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940173) fonctionnelle interne

Après avoir déterminé toutes les fonctions de service, nous avons cherché à les traduire rigoureusement en fonctions techniques pour ensuite les matérialiser en solutions constructives. Cela veut dire que dans cette partie, on va décrire l’analyse fonctionnelle interne du système. On se concentre maintenant seulement sur le fonctionnement interne du système et non plus l’analyse des fonctions externes au projet. En ce qui concerne ce point, il est nécessaire de se focaliser sur deux méthodes qui résultent de cette analyse : le raffinement de fonctions avec la méthode FAST (Function Analysis System Technique) et l’analyse fonctionnelle descendante avec la méthode SADT (Structured Analysis and Design Technique).

4.2.2.1 La méthode FAST

Le but de cette méthode est de mettre en place un flux entre les fonctions internes du système pour savoir comment les fonctions interagissent entre elles afin de réaliser la mission du système. Au contraire de l’analyse fonctionnelle externe qui consiste à étudier les flux entre les fonctions et l’environnement du système, la méthode FAST identifie les flux internes entre ces fonctions. Le digramme FAST ci-dessous schématise les sous-fonctions des fonctions principales déjà identifiées dans notre étude.

Projet CCM :

FP 1 : Le système doit informer l’utilisateur

de l'état des vérins :

FP 3 : Le système doit avoir une application

FP 2 : Le système doit différencier les utilisateurs selon plusieurs niveaux :

4.2.2.2 La méthode SADT

La méthode SADT est une méthode dédiée à la réalisation d’une analyse fonctionnelle descendante (top-down), c’est-à-dire du général vers le détaillé (de l’analyse vers la réalisation). C'est une démarche systématique de modélisation des systèmes complexes. Elle économise du temps et elle fournit une structure hiérarchisée par niveau permettant une clarification et une décomposition analytique de la complexité d'un système.

Un diagramme SADT est représenté par une boite qui contient un verbe à l’infinitif qui définit l’action avec des activités numérotés. Sur cette boite, on met des flèches représentants les entrées et les sorties. La figure suivante représente une simple boite SADT.

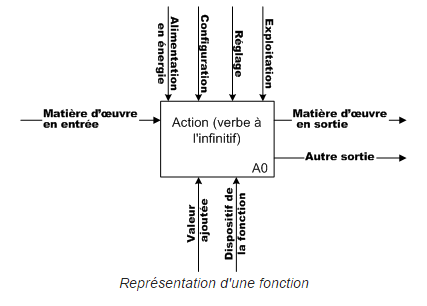


Figure : Représentation générale d’une fonction

Le diagramme SADT de notre système CCM :

Configuration

Consigne d’utilisateur

Projet CCM

Réglage

Energie électrique

Données non envoyées

Données reçus

Application Android

Capteurs

Figure : Diagramme SADT

4.2.3 Architecture fonctionnelle:

**Application**



**Boîte noire**

**(Carte arduino)**

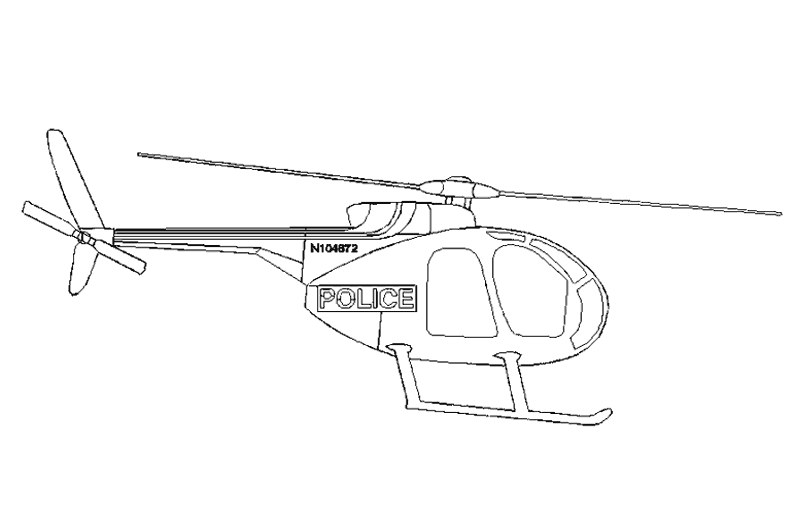


Figure :L’architecture fonctionnelle

4. Spécification fonctionnelle

Les spécifications fonctionnelles sont des méthodes dédiées à la description détaillée et complète des produits via un cahier des charges dans le but de décrire les différents cas d’utilisation du produit et ses différentes fonctionnalités. Généralement, elles sont faites par le chef du projet tout en respectant le souhait du client.

**Recevoir les informations**

**Avoir une nouvelle mesure**

**Chiffrer les données**

**Avoir une communication sans fil**

**Avoir le même protocole de communication**

**Afficher les données**

**Lire les données**

**Stocker les données**

**Récupérer les données**

**Envoyer les informations**

Figure :La spécification fonctionnelle

[4.3. Architecture](file:///C:\Users\Nancy\Desktop\SAUVMER_RA_20151116.docx#_Toc439940174) physique

L’architecture physique est incluse dans la phase de synthèse du système, elle sert à modéliser une solution sous la forme d’un assemblage de constituants existants ou réalisables. De plus elle sert à mettre en place l'allocation de chaque constituant à des fonctions et contraintes de performances de l’architecture fonctionnelle. En outre, elle traite l’état des organes physiques des interfaces et spécifications de chaque constituant.

Données binaire

Programmation language machine

Signal

Figure :L’architecture physique

Annexes:

Annexe 1 : Exigences du système:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifiant | Exigences | Type | Priorité | Vérification | Raison d'être | Relation avec d'autres exigences | Risques |
| EX001 | Le système doit informer l’utilisateur de l'état des vérins | Fonction | 1 | Faire le test sur un fichier contenant des données capteur | Réaliser la mission | Ex009, ex005 | Perte de surveillance du système |
| EX002 | Le système doit différencier les utilisateurs selon plusieurs niveaux | Fonction | 1 | Test sur les conditions d’accès aux informations  (Identifiant et mot de passe) | Augmenter la performance | Ex007, ex003 | Pirater la confidentialité des utilisateurs |
| EX003 | Le système doit avoir une application qui communique par bluetooth avec la carte | Fonction | 2 | Analyser des fichiers .trace et les afficher dans une interface graphique | Faciliter la maintenance du système |  | Avoir des problèmes lors de la communication |
| EX004 | Le système doit être géré facilement par l'utilisateur | Contrainte | 3 | Tester l’application | Faciliter l’utilisation | Ex003 | Difficultés à comprendre les données |
| EX005 | L'utilisateur doit être capable de surveiller l'état du système | Contrainte | 1 | Tester les mises à jour des données des capteurs | Réaliser la mission | Ex003 | Mancertaines informations |
| EX006 | Le système doit respecter la réglementation relative à l’aviation et doit s’adapter à l’environnement | Contrainte | 3 | Tester l’adaptation du système avec les différents conditions extérieurs | Respecter les règles et les conditions |  | Tomber en panne à cause d’une mal résistivité |
| EX007 | Le système doit respecter la confidentialité des informations | Contrainte | 2 | Tester la performance du cryptage | Sécuriser le système | Ex009 | Pirater la confidentialité des données |
| EX008 | Le système doit avoir une application qui se fonctionne sur une tablette Android | Contrainte | 2 | Vérifier le bon fonctionnement de l’application sur la tablette | Faciliter la maintenance du système | Ex003 | Avoir une difficulté de transformer le code java vers Android studio |
| EX009 | Le système doit être capable de lire les différentes données depuis les capteurs | Contrainte | 1 | Tester l’accès aux données des capteurs | Réaliser la mission |  | Avoir une mal connexion entre les capteurs et la carte |
| EX0010 | Le système doit respecter des contraintes budgétaires | Contrainte | 4 | Estimer le prix total des composants du système | Evident |  | Avoir un mal choix des composants |
| EX0011 | Le système doit s'adapter à une source d'énergie électrique | Contrainte | 4 | Alimenter le système en respectant la plaque signalétique de chaque composant électrique | Augmenter la performance | Ex006 | Avoir un court circuit |

Annexe 2 : Description des fonctions :

FP 1  Le système doit informer l’utilisateur de l'état des vérins

FP 11  Le système doit stocker les données depuis les capteurs

FP 12  Le système doit récupérer les données depuis la boîte noire

FP 13  Le système doit signaler à la boîte qu'elle doit envoyer les informations

FP 14  Le système doit recevoir les informations depuis l’application

FP 2 Le système doit différencier les utilisateurs selon plusieurs niveaux

FP 21 Le système doit faire une nouvelle lecture à une fréquence différente pour chaque n n n n capteur

FP 22 Le système doit chiffrer les données de manières différentes selon les niveaux

FP 3  Le système doit avoir une application qui se communique par bluetooth avec la carte

FP 31  Le système doit lire les données stockées

FP 32  Le système doit afficher les données envoyées

FC 33  Le système doit avoir une communication ou liaison sans fil

FC 331  Le système doit utiliser la liaison Bluetooth

FC 34  Les différentes parties du système ont le même protocole de communication

FC 1  Le système doit être géré facilement par l'utilisateur

FC 11  L'utilisateur doit être capable d'assigner un objectif de manière intuitive et précise

FC 2  L'utilisateur doit être capable de surveiller l'état du système

FC 21  Le système transmet l'état des capteurs vers l’application

FC 3  Le système doit respecter la réglementation relative à l’aviation et doit s’adapter à nnnnil’environnement

FC 31  Le système doit résister à la corrosion pour un temps raisonnable

FC 32  Le système doit résister aux chocs qui pourraient lui être soumis

FC 33  Le système doit résister aux contraintes météorologiques

FC 34  Le système doit s’installer sur un milieu extérieur

FC 35  Le système doit pouvoir fonctionner sous une pluie légère

FC 4  Le système doit respecter la confidentialité des informations

FC 41  Le système doit déterminer différents niveaux de sécurité

FC42 Le système doit crypter les informations à envoyer

FC43 Le système doit décrypter les informations reçus

FC 5  Le système doit avoir une application qui se fonctionner sur une tablette Android

FC 6  Le système doit être capable de lire les différentes données depuis les capteurs

FC 61  Les différentes parties du système émettent et réceptionnent

FC 7  Le système doit respecter des contraintes budgétaires

FC 8  Le système doit s'adapter à une source d'énergie électrique

FC 81  Le système doit avoir une faible consommation d’énergie

# Bibliographie

Desmoulins, N. (2009, mars 26). *Wikipedia.org*. Récupéré sur Wikipedia.org: https://fr.wikipedia.org/wiki/Cycle\_en\_V

Hassib, S. (2015, octobre 23). *International University*. Récupéré sur supinfo: www.supinfo.com/articles/single/795-cycle-v