



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique

Cahier de spécification système & plan de développement

Projet :	Maquette détaillée du cahier de spécif. pour les PFE		
Emetteur :	Lin SHEN	Coordonnées : EPU-DI	
Date d'émission :	7 novembre 2014		
Validation			
Nom	Date	Valide (O/N)	Commentaires

N Ragot : 20/10/2009 ; O ; Ok pour 2009-2010

N Ragot : 14/10/2010 ; O ; Ok pour 2010-2011

Historique des modifications

Version	Date	Description de la modification
---------	------	--------------------------------

00 : 11/2008 ; Version initiale : synthèse de différents documents

01 : 10/10/2009 ; Refonte basée sur les docs de Génie Log (VTK)

02 : 21/10/2009 ; Correction suite aux relectures des EC-DI

02b : 14/10/2010 ; Correction des headers

Table des matières

Cahier de spécification système	5
1.1 Introduction	5
1.2 Contexte de la réalisation	5
1.2.1 Contexte	5
1.2.2 Objectifs	5
1.2.3 Hypothèses (selon les projets)	5
1.2.4 Bases méthodologiques (selon les projets)	5
1.3 Description générale	5
1.3.1 Environnement du projet	5
1.3.2 Caractéristiques des utilisateurs	5
1.3.3 Fonctionnalités et structure générale du système	5
1.3.4 Contraintes de développement, d'exploitation et de maintenance	6
1.4 Description des interfaces externes du logiciel	7
1.4.1 Interfaces matériel/logiciel	7
1.4.2 Interfaces homme/machine	7
1.4.3 Interfaces logiciel/logiciel	10
1.5 Architecture générale du système	10
1.6 Description des fonctionnalités	10
1.6.1 Définition de la fonction i	11
1.7 Conditions de fonctionnement	11
1.7.1 Performances	11
1.7.2 Capacités	11
1.7.3 Modes de fonctionnement (optionnel)	11
1.7.4 Contrôlabilité	12
1.7.5 Sécurité	12
1.7.6 Intégrité (optionnel)	12
1.7.7 Conformité aux standards (optionnel)	12
1.7.8 Facteurs de Qualité (optionnel)	12
 Plan de développement	 14
2.1 Découpage du projet en tâches	14
2.1.1 Tâche i	14
2.2 Planning	14
 A Glossaire	 15
 B Références	 16
 C Index	 17

Cahier de spécification système

1.1 Introduction

Nature et objectifs du document, les acteurs mis en jeu (client, MOA, MOE), les auteurs, relecteurs,...

1.2 Contexte de la réalisation

1.2.1 Contexte

Définir les enjeux et le contexte général dans lequel s'inscrit le projet (domaine d'application, marché, cible,...)

1.2.2 Objectifs

Définir le sujet en quelques lignes : s'il s'agit de faire ou d'améliorer un système...et surtout la nature du système en question (est-ce un système d'information, un système dynamique i.e. à contraintes temps réel, un logiciel d'aide à la décision ou d'optimisation,...). Expliquer également de quoi doit être constitué le système au niveau matériel et logiciel.

1.2.3 Hypothèses (selon les projets)

Les hypothèses décrivent tous les facteurs susceptibles de remettre en cause tout ou une partie de la réalisation des spécifications ainsi que d'éventuelles solutions de repli. Par exemple, « Si au cours du projet il se passe X, on fera Y, sinon on fera Z » ou encore « Si on n'arrive pas à faire X, ou si on ne trouve pas la librairie Y, on fera Z ». Il faut bien détailler toutes les alternatives qui se posent en début de projet.

1.2.4 Bases méthodologiques (selon les projets)

Elles précisent quelles procédures, méthodes, outils, normes, langages, règles de programmation devront être utilisés pour mener à bien le projet.

1.3 Description générale

1.3.1 Environnement du projet

Ce projet est basé sur un projet existant. On garde l'algorithme principal et optimise l'interface pour faciliter les utilisations.

1.3.2 Caractéristiques des utilisateurs

Identifier les différents types d'utilisateurs du système. Pour chacun on devra préciser les caractéristiques qui affecteront l'interface utilisateur (menus, commandes textuelles,...) :

- connaissance ou non de l'informatique ;
- expérience de l'application ;
- utilisateurs réguliers et/ou occasionnels ;
- droits d'accès utilisateurs.

1.3.3 Fonctionnalités et structure générale du système

Décrire les fonctions utilisateurs du système et les principaux objets qui constituent le système. Ceci peut être fait au travers d'un ou plusieurs diagrammes de cas d'utilisation généraux par type d'utilisateur, ainsi que par un diagramme statique (« objet » au sens large) qui est le pendant de la partie 1.3.1 (externe/interne). RQ : ces points seront détaillés par la suite (cf. parties 1.5 et 1.6)



1.3.4 Contraintes de développement, d'exploitation et de maintenance

Contraintes de développement

Préciser les contraintes liées aux :

- matériels : quelles sont les particularités du matériel qui vont contraindre le développement logiciel ;
- langages de programmation imposés ou adoptés ;
- logiciels de base à utiliser pour le développement ;
- environnements nécessaires : simulateurs, outils logiciels ;
- algorithmes imposés : ne pas hésiter à mettre en annexe les documents présentant ces algorithmes ;
- bibliothèques de programmes imposées ;
- protocoles de communication imposés : si nécessaire mettre en annexe une présentation de ces protocoles de communication ;
- délais de réalisation ;
- etc.

Contraintes d'exploitation (optionnel)

Préciser les contraintes liées aux :

- règles de gestion du système ;
- affectations des responsabilités des utilisateurs ;
- planning d'exploitation ;
- problèmes de sauvegardes et de reprises (architecture minimale nécessaire à un fonctionnement dégradé) ;
- problèmes de sécurité et d'intégrité ;
- intervention d'une équipe système ;
- etc.

Maintenance et évolution du système (optionnel)

Préciser les contraintes liées aux procédures de maintenance :

- curative ou corrective ;
- adaptative ;
- évolutive du système ;
- perfective.

1.4 Description des interfaces externes du logiciel

1.4.1 Interfaces matériel/logiciel

La figure ci-dessous décrit l'interface matériel et logiciel. On a une caméra pour noter les comportements d'agneau. On peut récupérer les vidéos et les traiter dans notre logiciel.



FIGURE 1.1 – Interface matériel/logiciel

1.4.2 Interfaces homme/machine

Voici la figure ci-dessous, c'est l'interface homme/machine.

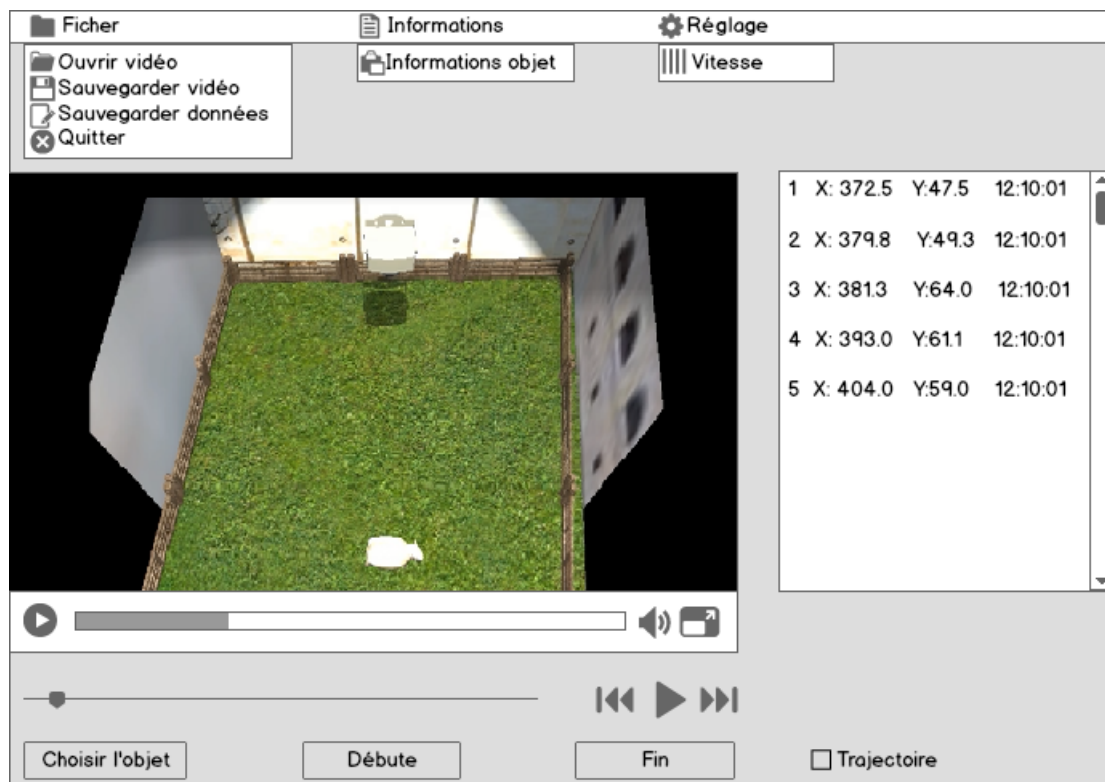


FIGURE 1.2 – Interface homme/machine

Dans la barre de menu, il y a trois parties : fichier, informations et réglage.

– **Fichier :**

- * *Ouvrir vidéo* : C'est pour ouvrir un vidéo d'animal. On peut enregistrer le chemin du dossier d'image. C'est plus pratique pour les utilisateurs.

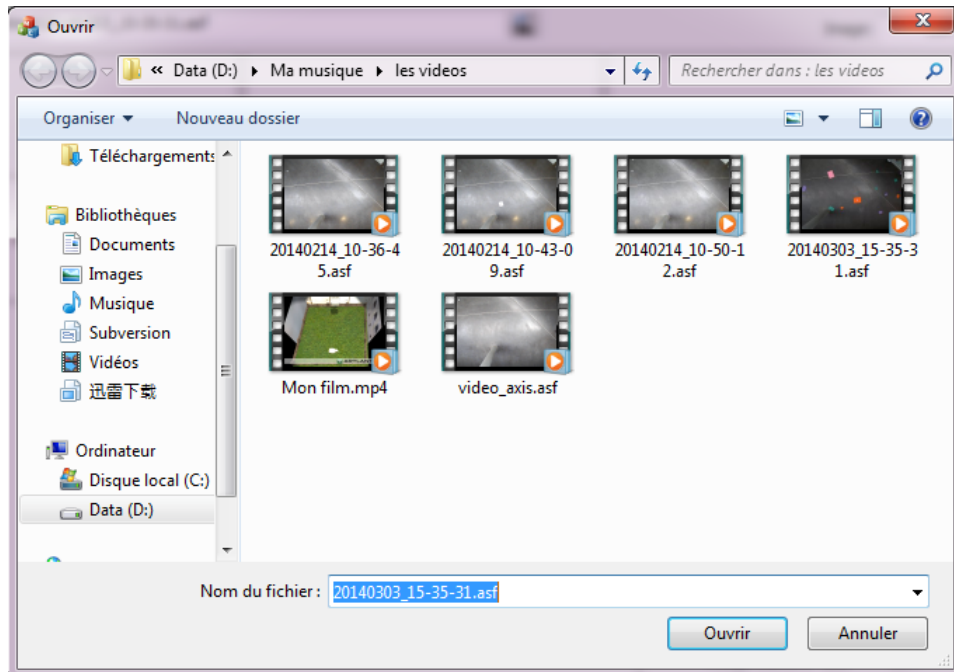


FIGURE 1.3 – Ouvrir le vidéo

- * *Sauvegarder vidéo* : On choisit une chemin pour sauvegarder le vidéo avec la trajectoire.

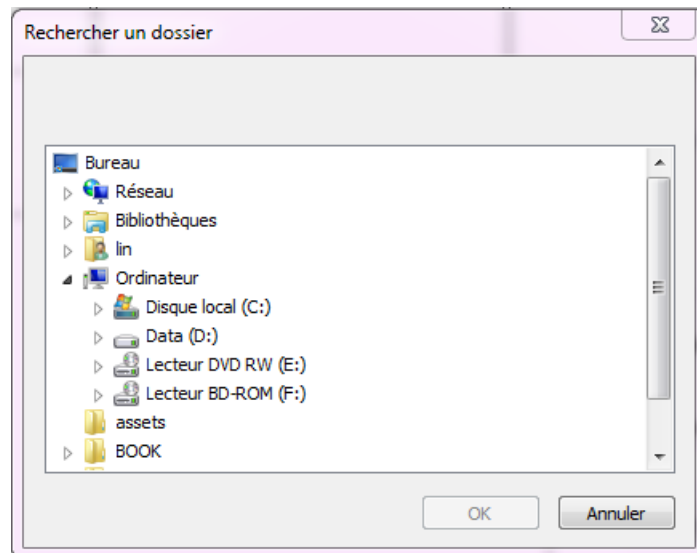


FIGURE 1.4 – Sauvegarder le vidéo

- * *Sauvegarder données* : On choisit une chemin pour sauvegarder les données des coordonnées.
- * *Quitter* : On quitte la fenêtre.

– **Informations :**

- * *Information objet* : C'est pour afficher la couleur d'objet suivi.

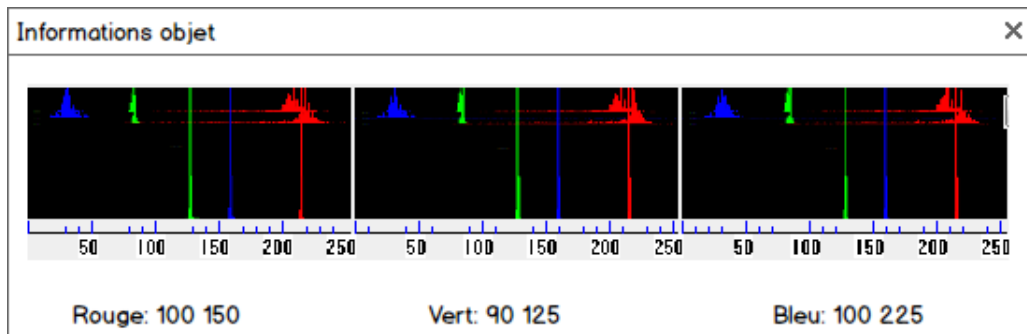


FIGURE 1.5 – La couleur d'objet

– Réglage :

* *Vitesse* : On définit la distance maximum d'objet entre 2 frames. C'est pour bien suivre l'objet.



FIGURE 1.6 – La vitesse

Au centre du fenêtre, on a une fenêtre de vidéo à gauche. C'est pour afficher notre vidéo.

A droite, on a une fenêtre d'affichage les informations de déplacement. Il y a 4 colonnes : les frames, les coordonnées d'objet(X,Y) et l'heure du frame correspondant .

En bas de la fenêtre, il y a un curseur pour déplacer le vidéo. Il y a un bouton précédent et un bouton suivant pour déplacer le vidéo frame par frame. Entre le bouton précédent et le bouton suivant on a un bouton de démarrage pour démarrer et terminer le vidéo.

Après on choisit un image avec le curseur, on appuie sur le bouton "Choisir l'objet", une fenêtre d'image va apparaître. Ici on peut choisir un objet qu'on veut suivre.

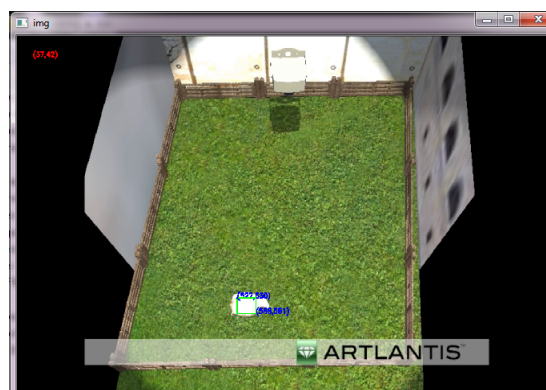


FIGURE 1.7 – Choisir l'objet

Après avoir choisi l'objet et défini le début et la fin du vidéo, on croche "trajectoire". Et on appuie le bouton de démarrage, on peut regarder le vidéo avec la trajectoire.

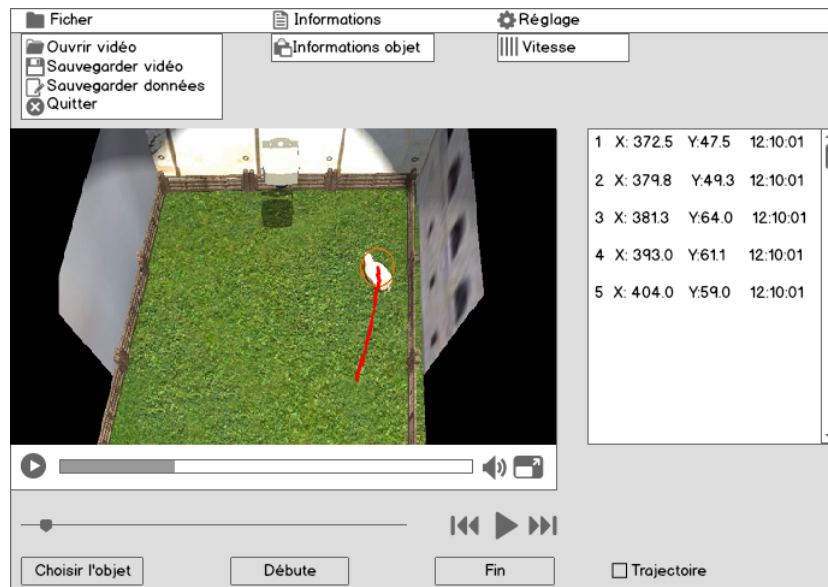


FIGURE 1.8 – Le vidéo avec la trajectoire

Le bouton "Début" et "Fin" est pour définir le début et la fin du vidéo.

1.4.3 Interfaces logiciel/logiciel

Il faut spécifier les points suivant :

- moyens d'accès à des systèmes de gestion de base de données, à des bibliothèques logicielles, description de la fréquence des accès, autorisations, etc. ;
- procédures de transferts d'information à distance (échanges d'informations par téléinformatique) ;
- procédures d'échange de messages entre application ;
- etc.

1.5 Architecture générale du système

Identifier les principaux composants/éléments du système ainsi que leurs relations. Sans être une analyse à part entière, cette partie doit montrer que vous avez une première réflexion sur la structure interne de votre projet qui complètera la vision sommaire donnée en 1.3.3. Un diagramme objet au sens large présentant les principales structures de données ainsi que les principaux composants du système pourra être fourni avec un ensemble de commentaires explicatifs.

1.6 Description des fonctionnalités

Il s'agit de l'expression des besoins fonctionnels. Cette partie a donc comme objectif de décrire l'ensemble des fonctionnalités du système en précisant avec quels composants de la partie 1.5 elles interagissent. Des diagrammes de cas d'utilisation plus détaillés que ceux présentés en 1.3.3, ainsi que l'arbre hiérarchique des fonctionnalités pourront être fournis ici pour donner une vision plus globale. En outre, chaque fonctionnalité sera décrite précisément (cf.ci-dessous). Là encore, il s'agit d'une pré-analyse indispensable à l'évaluation de la complexité de votre projet et à la planification de sa réalisation.

1.6.1 Définition de la fonction i

Identification de la fonction i

Présenter la fonction :

- nom de la fonction ;
- rôle, présentation générale ;
- priorité associée à la réalisation de la fonction (primordiale, secondaire, facultative).

Description de la fonction i

Décrire précisément :

- les entrées et les sorties ainsi que les préconditions et postconditions déjà connues, uniquement sous forme textuelle et en langage naturel (pas dans un pseudo langage algorithmique). Si ces E/S sont connectées à d'autres fonction ou interfaces, le préciser également ;
- les composants avec lesquels cette fonction interagit (données/composants utilisés/modifiés, etc. ;
- le traitement associé à la fonction et à ses interfaces. Il peut s'agir d'une explication ou d'un pseudo-algorithme général précisant les différentes étapes du traitement. Lors de l'analyse, ce dernier pourra être précisément représenté par un diagramme d'activité ;
- si une gestion des erreurs spécifique (hors format des E/S) est prévue et comment celle-ci doit être mise en place si cela est déjà connu (notamment pour les fonctions sensibles).

1.7 Conditions de fonctionnement

Il faut dans ce paragraphe décrire les dispositions qu'il est nécessaire de prendre en compte pour les différentes conditions de fonctionnement du système.

1.7.1 Performances

Préciser en termes mesurables, les spécifications temps réel liées à l'utilisation du système :

- du point de vue de l'utilisateur : temps de réponse souhaité, fréquence d'utilisation, temps d'indisponibilité acceptable, etc. ;
- du point de vue de l'environnement : fréquence moyenne d'acquisition d'états ou de mesures, fréquence maximale d'E/S, etc.

1.7.2 Capacités

Décrire les limites des problèmes traitables par le système et les limites des éventuelles extensions comme par exemple :

- nombre max de terminaux ;
- nombre max de points d'acquisition ;
- nombre max de transactions simultanées de tel type, etc. ;
- capacité max de stockage ;
- taille max des données traitées ;
- etc.

1.7.3 Modes de fonctionnement (optionnel)

Décrire les modes d'exploitation du système tels que :

- la mise sous tension ;
- l'arrêt ;
- la reprise de secours ;

- les modes dégradés ;
- etc.

1.7.4 Contrôlabilité

Il faut décrire, si elles existent, les spécifications particulières permettant de suivre l'exécution d'un traitement (fichier de log, niveaux d'affichages en mode debug, etc.)

1.7.5 Sécurité

Indiquer le niveau de confidentialité du système (contrôle d'accès des utilisateurs, mots clefs, mots de passe, etc.). Ceci est directement lié aux différents types d'utilisateurs (cf. 3.2).

1.7.6 Intégrité (optionnel)

Préciser les protections contre la déconnexion imprévue, les pertes d'information, etc. et quelles sont les procédures à suivre pour restaurer les données du système. Y-a-t-il des situations non protégées ?

1.7.7 Conformité aux standards (optionnel)

Les références aux standards ou normes (CCITT, ISO, AFNOR, ...) devront être notées.

1.7.8 Facteurs de Qualité (optionnel)

Il faut ici faire référence à un plan qualité du logiciel si nécessaire.

Plan de développement

2.1 Découpage du projet en tâches

Vous devez dans cette partie réfléchir à la structuration de votre projet en tâches. Une tâche correspond ici à un ensemble de réalisations ayant une cohésion d'ensemble. Il peut s'agir de composants ou de fonctionnalités du système tels qu'identifiés dans les sections 1.5 et 1.6 mais le plus souvent, les tâches sont transversales aux fonctions et/ou aux composants. Ainsi, selon le point de vue adopté, un composant pourra effectuer plusieurs fonctions, une fonction pourra travailler sur plusieurs composants, etc. De même, la réalisation des interfaces homme/machine peut faire l'objet de tâches à part entière (améliore leur indépendance et les rend plus facilement évolutive) ou encore peut être intégrée à la réalisation de chaque composant auquel elles se rapportent.

Ces tâches ne sont pas nécessairement indépendantes les unes des autres même si cela facilite souvent leur identification et permet de répartir plus facilement les charges. On indiquera également ici les tâches relatives à la gestion de projet (prise en mains de l'existant, bibliographie, rédaction du cahier de spécification, du rapport, de manuels techniques ou utilisateurs, mise en production et recette globale, etc. Chaque tâche doit être décrite précisément :

2.1.1 Tâche i

Description de la tâche

Dire en quoi consiste la tâche, quels sont les composants, parties ou fonctions du système qu'elle intègre, le degré de finalisation souhaité (version alpha, beta, etc.), etc.

Cycle de vie

S'il s'agit d'une tâche de développement, donner le cycle de vie associé pour préciser notamment si des phases de tests (unitaires, fonctionnels, d'intégration) et de recette sont prévus.

Livrables

Si la tâche requiert un livrable (rapport plus ou moins court, livraison d'un composant logiciel, etc.), préciser ici sa nature et ce qu'il doit contenir ou faire apparaître.

Estimation de charge

Estimation du temps en jour/homme nécessaire à la réalisation de la tâche.

Contraintes temporelles

Indique si des contraintes de temps fortes existent pour la réalisation de la tâche.

2.2 Planning

Le planning (sous forme de diagramme de Gantt si possible) synthétise l'ordonnancement de chacune des tâches en faisant apparaître leur éventuelle parallélisation. Il indique également les dates clés de la réalisation du projet (soutenance, etc.) ainsi que les dates de remise des livrables.

Glossaire

Dans cette partie on doit trouver, classés par ordre alphabétique, les définitions des termes courants utilisés, des termes techniques, abréviation, sigles et symboles employés dans l'ensemble du document.

Références

Cette dernière partie recense les références techniques sur le projet sur :

- les documents relatifs à l'existant et à l'environnement ;
- les documents sur les méthodes et algorithmes cités ;
- les documents bibliographiques (internes et externes) ;
- les sources d'obtention des documents.

Utiliser les commandes \LaTeX adéquates.

Index

Cette partie indique les pages où sont traités et mentionnés les sujets et les termes les plus importants du document.

Utiliser les commandes \LaTeX adéquates.