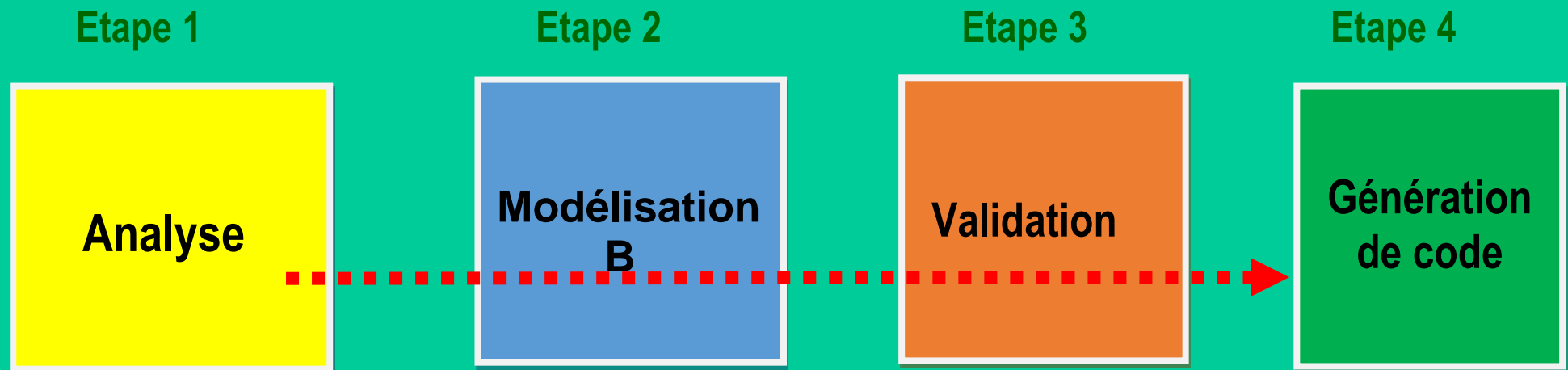


Projet d'appontage automatique



Cycle de développement en B

I- Analyse du système d'appontage

1-Vue globale du système

Le système est organisé autour des systèmes suivants :

- le système **Pilote** qui représente le système de contrôle automatique de l'appontage.
- le sous-système **Contexte** qui définit le contexte de fonctionnement global du système : paramètres du système, contraintes,....

-le sous-système **Appontage** système opérationnel chargé de coordonner:

- le maintien de l'avion dans le plan d'approche
- et la trajectoire de descente de l'avion dans ce plan.

-le sous-système **PlanApproche** qui garantit le maintien de l'avion dans le **plan d'approche**.

Au cours de la manœuvre, l'avion évolue dans plan vertical (plan d'approche) qui coupe le plan d'appontage (du porte avion) suivant la piste d'appontage.

-le sous-système **CapteurPlan** qui récupère les données réelles fournies par les capteurs du **roulis** et du **lacet** de l'avion.

-le-sous-système **CapteurRoulis** qui mesure le **roulis** de l'avion.

- le sous-système **CapteurLacet** qui mesure le **lacet** de l'avion.
- le sous-système **RegulateurPlan** qui corrige en temps réel les paramètres (lacet-roulis) de sorte à maintenir l'avion dans le **plan d'approche**
- le sous-système **RegulateurRoulis** corrige le paramètre roulis de sorte que l'avion soit maintenu dans le plan d'approche
- le sous-système **RegulateurLacet** corrige le paramètre lacet de sorte que l'avion soit maintenu dans le plan d'approche
- le sous-système **Trajectoire** qui contrôle la trajectoire de descente de l'avion maintenu dans le **plan d'approche**

- le sous-système **Aiguilleur** qui reçoit les données réelles sur les paramètres vitesse et tangage de l'avion et retourne les données pour corriger, en temps réel, la trajectoire d'appontage.
- le sous-système **BASIC_IO** gère les entrées/sorties du simulateur.
- le sous-système **CapteurTrajectoire** récupère les données réelles sur la vitesse et le tangage de l'avion.
- le sous-système **CapteurVitesse** qui mesure la vitesse réelle de l'avion.
- le sous-système **CapteurTangage** qui mesure l'angle de tangage réel de l'avion.

-le sous-système **RegulateurTrajectoire** qui commande les régulateurs de vitesse et de tangage afin de corriger la trajectoire de l'avion.

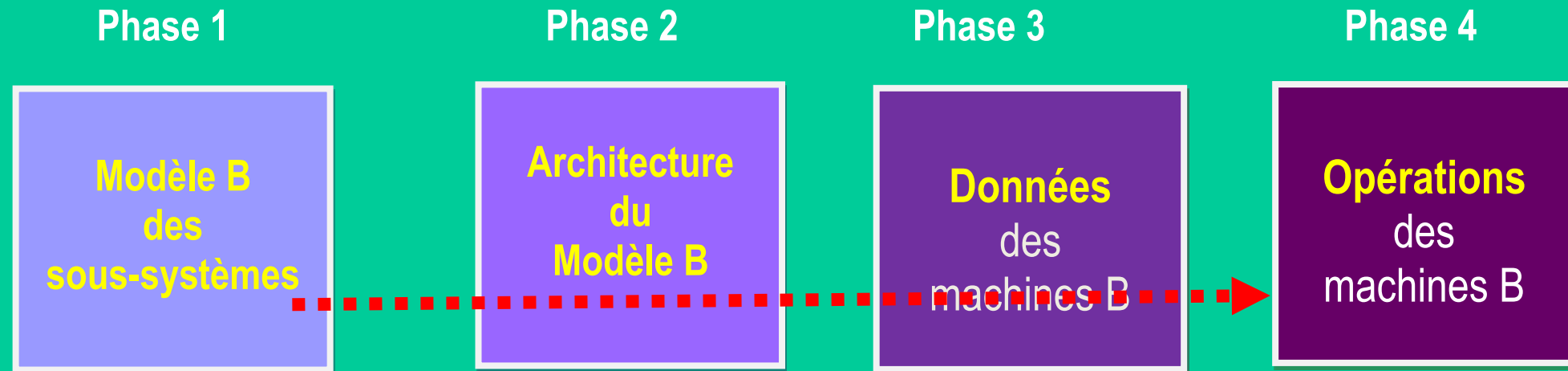
-le sous-système **RegulateurVitesse** qui corrige la vitesse de l'avion en appliquant la consigne de l'aiguilleur automatique.

-le sous-système **RegulateurTangage** qui corrige le tangage de l'avion en appliquant la consigne de l'aiguilleur automatique

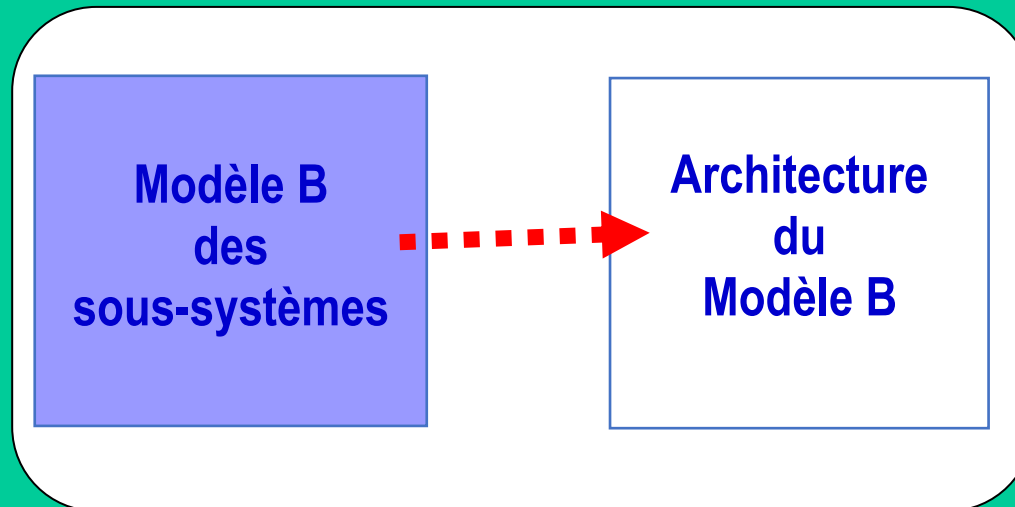
1-Vue détaillée du système

A proposer par les étudiants dans le cadre du projet.

II- Modélisation en B



Phase 1 : Modèles B des sous-systèmes



Le système **Pilote** est représenté par un **module B** composé de deux composants B :

Pilote.mch

Pilote_i.imp

Le sous-système **Contexte** est représenté par un module B composé de deux composants B :

Contexte.mch

Contexte_i.imp

Le sous-système **Appontage** est représenté par un module B composé de deux composants B :

Appontage.mch

Appontage_i.imp

Le sous-système **Trajectoire** est représenté par un module B composé de deux composants B :

Trajectoire.mch

Trajectoire_i.imp

Le sous-système **PlanApproche** est représenté par un module B composé de deux composants B :

PlanApproche.mch

PlanApproche_i.imp

Le sous-système **RégulateurTrajectoire** est représenté par un module B composé de deux composants B :

RegulateurTrajectoire.mch

RegulateurTrajectoire_i.imp

Le sous-système **Aiguilleur** est représenté par un module B composé de deux composants B :

Aiguilleur.mch

Aiguilleur_i.imp

Le sous-système **CapteurTrajectoire** est représenté par un module B composé de deux composants B :

CapteurTrajectoire.mch

CapteurTrajectoire_i.imp

Le sous-système **CapteurTangage** est représenté par un module B composé de deux composants B :

CapteurTangage.mch

CapteurTangage_i.imp

Le sous-système **CapteurVitesse** est représenté par un module B composé de deux composants B :

CapteurVitesse.mch

CapteurVitesse_i.imp

Le sous-système **RegulateurTangage** est représenté par un module B composé de deux composants B :

RegulateurTangage.mch

RegulateurTangage_i.imp

Le sous-système **RegulateurVitesse** est représenté par un module B composé de deux composants B :

RegulateurVitesse.mch

RegulateurVitesse_i.imp

Le sous-système **CapteurPlan** est représenté par un module B composé de deux composants B :

CapteurPlan.mch

CapteurPlan_i.imp

Le sous-système **RegulateurPlan** est représenté par un module B composé de deux composants B :

RegulateurPlan.mch

RegulateurPlan_i.imp

Le sous-système **CapteurLacet** est représenté par un module B composé de deux composants B :

CapteurLacet.mch

CapteurLacet_i.imp

Le sous-système **CapteurRoulis** est représenté par un module B composé de deux composants B :

CapteurRoulis.mch

CapteurRoulis_i.imp

Le sous-système **RegulateurRoulis** est représenté par un module B composé de deux composants B :

RegulateurRoulis.mch
RegulateurRoulis_i.imp

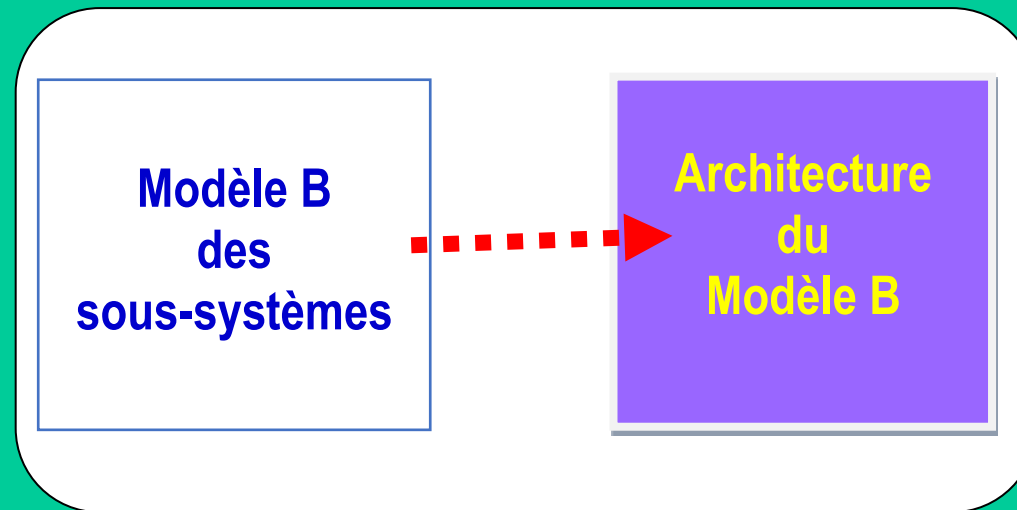
Le sous-système **RegulateurLacet** est représenté par un module B composé de deux composants B :

RegulateurLacet.mch
RegulateurLacet_i.imp

Le sous-système de base **BASIC_IO** est représenté par un module B abstrait constitué de la machine de base réutilisable :

BASIC_IO

Phase 2 : Architecture du modèle B



L'ensemble des modules B sont liés par les liens **d'importation et de visibilité**.

Les liens d'importation («imports») et de visibilité («sees») déterminent l'**architecture du modèle**.

Les liens d'importation déterminent l'**arbre d'importation** dont les nœuds sont des modules B reliés par des liens «imports».

Le **graphe de dépendances** est construit à partir de l'arbre d'importation en y ajoutant les **liens de visibilité** («sees»).

Le module **Pilote** importe les modules :
Contexte et **Appontage**.

Le module **Appontage** importe les modules :
Trajectoire et **PlanApproche**
et voit le module **Contexte**

Le module **Trajectoire** importe les modules :
RegulateursTrajectoire, **Aiguilleur** et **CapteurTrajectoire**
et voit le module **Contexte**

Le module **PlanApproche** importe les modules :
CapteurPlan et **RegulateurPlan**
et voit le module **Contexte**

Le module **RegulateurTrajectoire** importe les modules :
RegulateurTangage et **RegulateurVitesse**
et voit le module **Contexte**

Le module **Aiguilleur** importe le module **BASIC_IO** et voit le module **Contexte**

Le module **CapteurTrajectoire** importe les modules :
CapteurTangage et **CapteurVitesse** et voit le module **Contexte**

Le module **CapteurPlan** importe les module **CapteurLacet** et **CapteurRoulis** et voit le module **Contexte**

Le module **RegulateurPlan** importe les modules :
RegulateurRoulis et **RegulateurLacet**
et voit le module **Contexte**

Le module **RegulateurTangage** voit le module **Contexte**

Le module **RegulateurVitesse** voit le module **Contexte**

Le module **CapteurTangage** importe le module **BASIC_IO** et voit le module **Contexte**

Le module **CapteurVitesse** importe le module **BASIC_IO** et voit le module **Contexte**

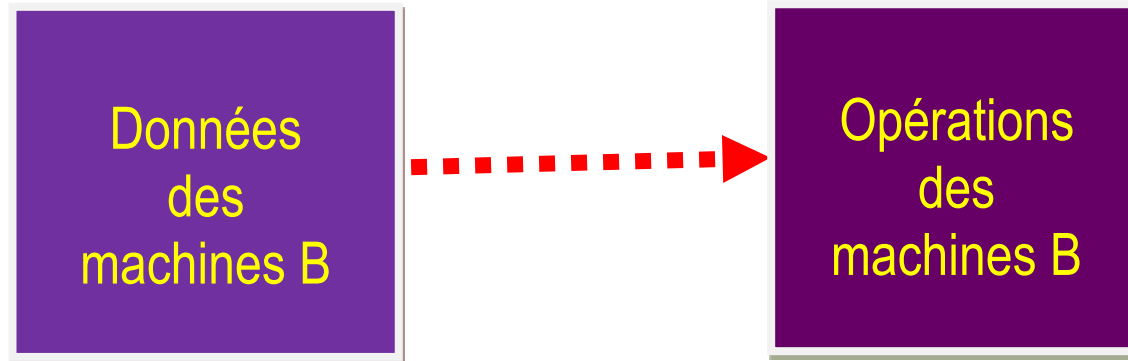
Le module **CapteurLacet** importe le module **BASIC_IO** et voit le module **Contexte**

Le module **CapteurRoulis** importe le module **BASIC_IO** et voit le module **Contexte**

Le module **RegulateurRoulis** voit le module **Contexte**

Le module **RegulateurLacet** voit le module **Contexte**

Phase 3 & 4 : Description des machines B



Décrire les machines B du modèle

Le principe est le suivant:

1-Le modèle d'un sous-système est un **Module B**

2-Un **module B** est constitué de deux **Machines B** :

- une **Machine de spécification**,
- une **Machine d'implémentation**

3-une **Machine B** est constituée de trois parties :

1-une partie **interface**,

2-une partie **statique**: **les données et variables d'état**

3-une partie **dynamique**: les **opérations** pour manipuler les variables d'état