



Cas 9 : Automobile



FERRARA Arnaud GI04
LOURD Rodolphe GI05

1. Objectif

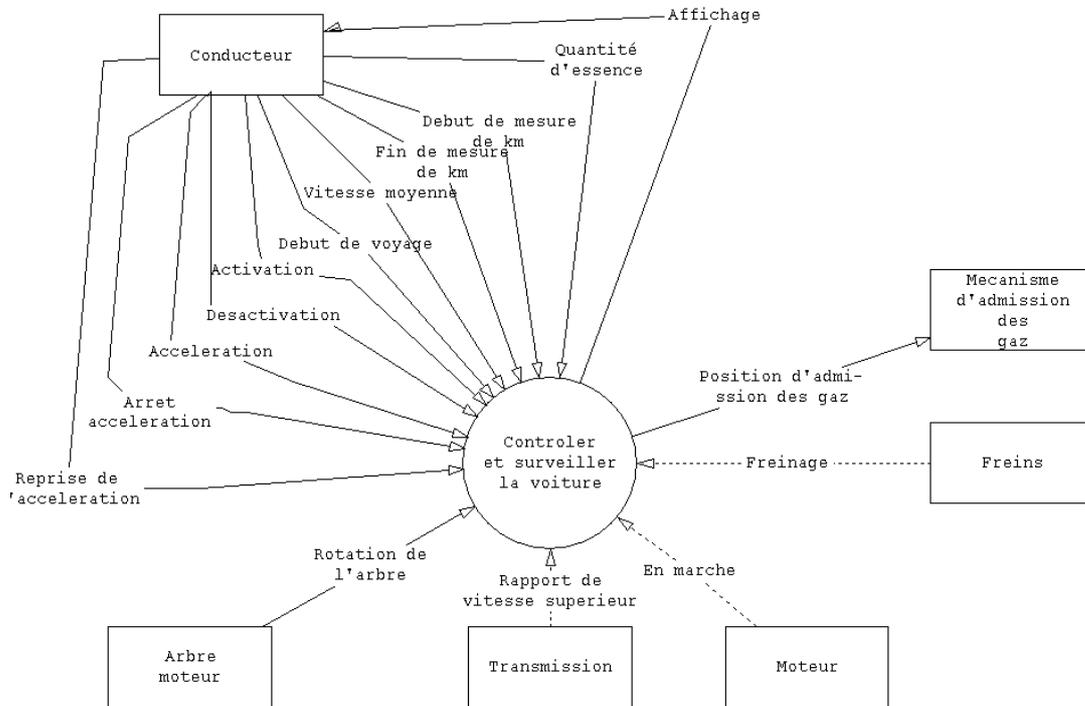
Il s'agit de développer un système, appelé « ordinateur de bord pour une automobile », comme étant une option à rajouter sur certains modèles de voiture. Le système doit prendre en compte plusieurs tâches routinières effectuées habituellement par le constructeur aussi bien en conduite qu'en entretien du véhicule :

- *contrôle de la vitesse de croisière*
- *calcul de la vitesse moyenne*
- *surveillance de la consommation d'essence*
- *aide au déclenchement de l'entretien*

Ces fonctions seront décrites plus précisément dans les différentes phases de modélisation par la suite (selon les normes OMT et UML).

2. Schéma de contexte (OMT)

Le diagramme suivant représente le système dans son ensemble, avec les parties opératives et informatique ainsi que les données qui transitent dans le cadre de notre application :

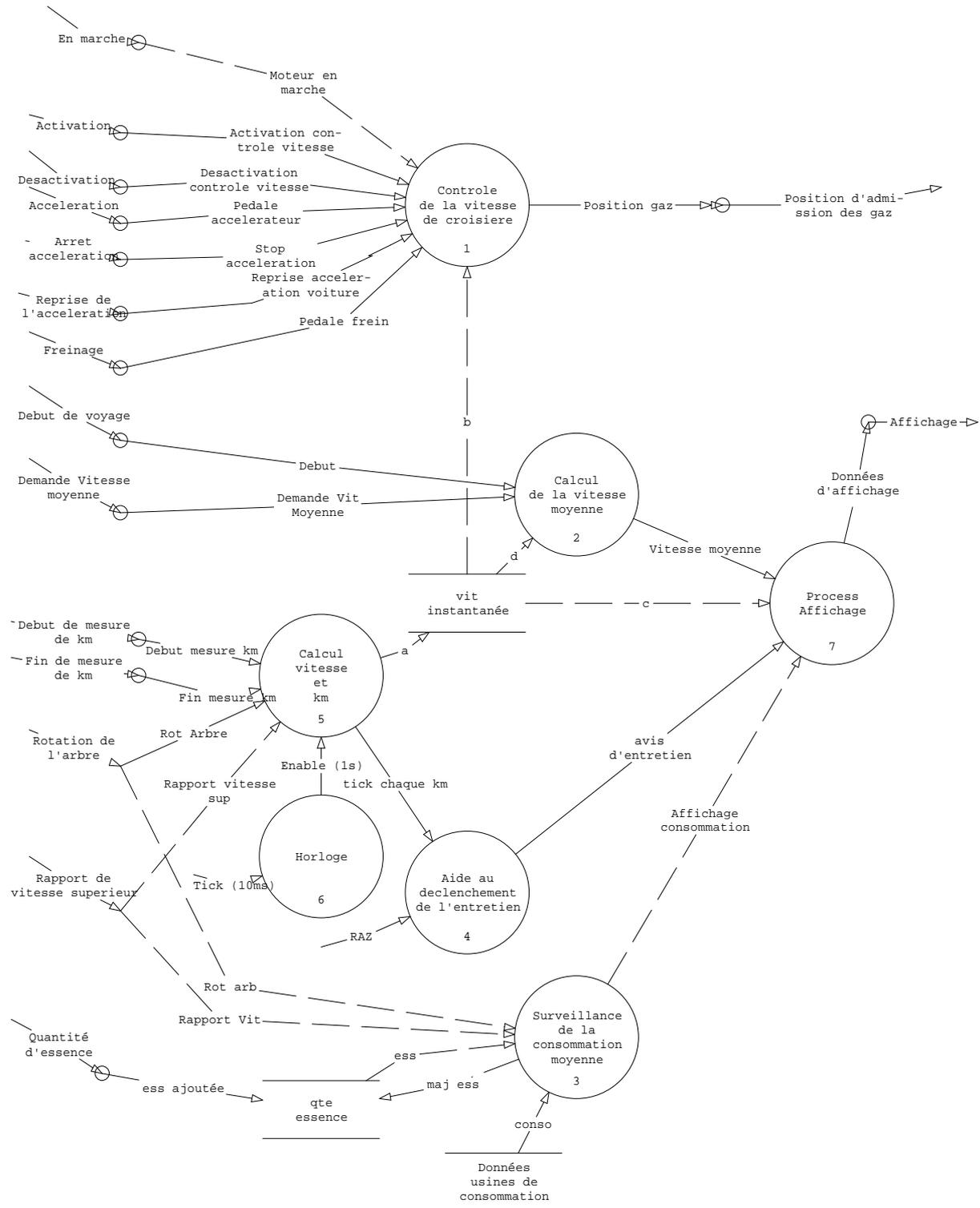


Dictionnaire de données

Type	Nom
Processus fonctionnel	Controler et surveiller la voiture
Partie opérative	Arbre moteur
Partie opérative	Conducteur
Partie opérative	Freins
Partie opérative	Mecanisme d'admission des gaz
Partie opérative	Moteur
Partie opérative	Transmission
Evènementiel (<i>flux discret</i>)	Acceleration
Evènementiel (<i>flux discret</i>)	Activation
Evènementiel (<i>flux discret</i>)	Affichage
Evènementiel (<i>flux discret</i>)	Arret acceleration
Evènementiel (<i>flux discret</i>)	Debut de mesure de km
Evènementiel (<i>flux discret</i>)	Debut de voyage
Evènementiel (<i>flux discret</i>)	Desactivation
Evènementiel (<i>flux discret</i>)	Fin de mesure de km
Evènementiel (<i>flux discret</i>)	Position d'admission des gaz
Evènementiel (<i>flux discret</i>)	Quantité d'essence
Evènementiel (<i>flux discret</i>)	Reprise de l'acceleration
Evènementiel (<i>flux discret</i>)	Rotation de l'arbre
Evènementiel (<i>flux discret</i>)	Vitesse moyenne
Flux de données (<i>flux continu</i>)	En marche
Flux de données (<i>flux continu</i>)	Freinage
Flux de données (<i>flux continu</i>)	Rapport de vitesse superieur

3. Diagramme de flux de données (OMT)

Ce schéma détaille les 4 fonctions principales de notre système. Celles-ci seront soit décomposées pour les rendre plus élémentaires, soit directement implémentées en pseudo code :



Dictionnaire de données

Type	Nom
Partie opérative	Freins
Partie opérative	Mecanisme d'admission des gaz
Partie opérative	Moteur
Partie opérative	Transmission
Partie opérative	Conducteur
Partie opérative	Arbre moteur
Processus fonctionnel	Aide au declanchement de l'entretien
Processus fonctionnel	Surveillance de la consommation moyenne
Processus fonctionnel	Processus fonctionnel Affichage
Processus fonctionnel	Horloge
Processus fonctionnel	Controle de la vitesse de croisiere
Processus fonctionnel	Calcul de la vitesse moyenne
Processus fonctionnel	Calcul vitesse et km
Flux de données (<i>flux continu</i>)	Moteur en marche
Flux de données (<i>flux continu</i>)	Position gaz
Flux de données (<i>flux continu</i>)	Rapport de vitesse superieur
Flux de données (<i>flux continu</i>)	Freinage
Flux de données (<i>flux continu</i>)	Données d'affichage
Flux de données (<i>flux continu</i>)	En marche
Flux de données (<i>flux continu</i>)	Affichage consommation
Evènementiel (<i>flux discret</i>)	Acceleration
Evènementiel (<i>flux discret</i>)	Activation
Evènementiel (<i>flux discret</i>)	Activation controle vitesse
Evènementiel (<i>flux discret</i>)	Affichage
Evènementiel (<i>flux discret</i>)	Arret acceleration
Evènementiel (<i>flux discret</i>)	avis d'entretien
Evènementiel (<i>flux discret</i>)	Debut
Evènementiel (<i>flux discret</i>)	Debut de mesure de km
Evènementiel (<i>flux discret</i>)	Debut de voyage
Evènementiel (<i>flux discret</i>)	Debut mesure km
Evènementiel (<i>flux discret</i>)	Demande Vit Moyenne
Evènementiel (<i>flux discret</i>)	Demande Vitesse moyenne
Evènementiel (<i>flux discret</i>)	Desactivation
Evènementiel (<i>flux discret</i>)	Desactivation controle vitesse
Evènementiel (<i>flux discret</i>)	Enable (1s)
Evènementiel (<i>flux discret</i>)	Fin de mesure de km
Evènementiel (<i>flux discret</i>)	Fin mesure km
Evènementiel (<i>flux discret</i>)	Pedale accelerateur
Evènementiel (<i>flux discret</i>)	Pedale frein
Evènementiel (<i>flux discret</i>)	Position d'admission des gaz
Evènementiel (<i>flux discret</i>)	Qte essence ajoutée
Evènementiel (<i>flux discret</i>)	Quantité d'essence
Evènementiel (<i>flux discret</i>)	Rapport vitesse sup
Evènementiel (<i>flux discret</i>)	RAZ
Evènementiel (<i>flux discret</i>)	Reprise acceleration voiture
Evènementiel (<i>flux discret</i>)	Reprise de l'acceleration
Evènementiel (<i>flux discret</i>)	Rot Arbre

3.1.1. Comptage Arbre

L'algorithme suivant permet de calculer le nombre de tours de l'arbre de transmission de l'automobile ; sa variable de sortie g est nécessaire au calcul de la vitesse :

```
@IN = Tick Arbre
@OUT = g

@PSPEC Comptage Arbre
-- g
-- Tick Arbre

@ g := g+1
```

3.1.2. Etalonnage

Le pseudo code suivant a pour but l'étalonnage du système, afin d'initialiser correctement les variables du programme :

```
@IN = Debut km
@IN = Fin km
@IN = l
@OUT = h
@OUT = i
@OUT = k

@PSPEC Etalonnage
-- Debut km
-- Fin km
-- h
-- i

@i:=0
@k:=0
@Attendre "Fin km"
@h:=l
@i:=0
```

3.1.3. Calcul vitesse

L'algorithme suivant va permettre de calculer la vitesse instantanée du véhicule, donnée primordiale car nécessaire au contrôle de la vitesse de croisière notamment :

```
@IN = 1s
@IN = f
@IN = g
@IN = j
@IN = vitesse enclenchée
@OUT = e
@OUT = tick km
@OUT = Vit inst
```

```

@PSPEC Calcul vitesse
-- 1s
-- e
-- f
-- g
-- j
-- tick km
-- Vit inst
-- vitesse enclenchée

@Si f == 1 alors
@Vit inst := f*vitesse enclenchée*3600/g
@e:= 0
@FSI

```

3.2. Horloge

Fonction classique dans tout système temps réel, la tâche horloge permet d'avoir une base de temps fiable et nécessaire dans notre système :

```

@IN = Tick (10ms)
@OUT = Enable (1s)

@PSPEC Horloge
-- Enable (1s)
-- Tick (10ms)

@cpteur := cpteur + 1
@Si cpteur == 10 Alors
@cpteur:=0
@Emettre "Enable (1s)"
@FSI

```

3.3. Aide au déclenchement de l'entretien

Cette tâche permet d'assister le conducteur dans l'entretien de son véhicule à partir d'un kilométrage précis ; en l'occurrence le système avertit l'utilisateur qu'un entretien est nécessaire tous les 10000 km (vidange par exemple) :

```

@IN = RAZ
@IN = tick chaque km
@OUT = avis d'entretien

@PSPEC Aide au déclenchement de l'entretien
-- avis d'entretien
-- RAZ
-- tick chaque km

@nb_km:= nb_km +1
@
@SI "RAZ" == 1 Alors nb_km =0
@FSI

```

```
@
@SI nb_km == 10000 Alors
@ Emettre "avis d'entretien"
@FSI
```

3.4. Surveillance de la consommation moyenne

L'algorithme suivant permet au conducteur de connaître à tout moment la consommation moyenne en carburant de son véhicule, à l'aide des données *Rapport Vit*, *Rot arb*, *conso* et *ess* :

```
@IN = Rapport Vit
@IN = Rot arb
@IN = conso
@IN = ess
@OUT = Affichage consommation
@OUT = maj ess

@PSPEC Surveillance de la consommation moyenne
-- Affichage consommation
-- ess
-- maj ess
-- Rapport Vit
-- Rot arb
-- conso

@ Selon valeur de "Rapport Vit"
@ Rapport Vit = -1 : Lire "consoR" dans "Données usines de consommation"
@ consommation := Rot arb * consoR
@ Rapport Vit = 0 : Lire "consoN" dans "Données usines de consommation"
@ consommation := Rot arb * consoN
@ Rapport Vit = 1 : Lire "conso1" dans "Données usines de consommation"
@ consommation := Rot arb * conso1
@ Rapport Vit = 2 : Lire "conso2" dans "Données usines de consommation"
@ consommation := Rot arb * conso2
@ Rapport Vit = 3 : Lire "conso3" dans "Données usines de consommation"
@ consommation := Rot arb * conso3
@ Rapport Vit = 4 : Lire "conso4" dans "Données usines de consommation"
@ consommation := Rot arb * conso4
@ Rapport Vit = 5 : Lire "conso5" dans "Données usines de consommation"
@ consommation := Rot arb * conso5
@
@maj ess := ess - consommation
@
@Emettre "Affichage consommation"
```

3.5. Calcul de la vitesse moyenne

La tâche suivante permet de connaître en temps réel la vitesse moyenne du véhicule en procédant par comparaison avec la vitesse instantanée donnée en entrée :

```
@IN = Vit instantanee
@IN = Debut
@IN = Demande Vit Moyenne
@OUT = Vitesse moyenne

@PSPEC Calcul de la vitesse moyenne
-- Vit instantanee
-- Debut
-- Demande Vit Moyenne
-- Vitesse moyenne

@Vitesse moyenne := Vit instantanee
@
@SI "Debut" == 1 Alors Vitesse moyenne := (Vitesse moyenne + Vit instantanee)/2
@FSI
@
@Emettre "Vitesse moyenne"
```

3.6. Contrôle de la vitesse de croisière

Le code suivant assure une des fonctions principales du système, le « cruise control » ; son rôle va consister à maintenir le véhicule à une vitesse constante définie par le conducteur :

```
@IN = b
@IN = moteur en marche
@IN = activation controle vitesse
@IN = desactivation controle vitesse
@IN = pedale accelerateur
@IN = stop acceleration
@IN = reprise acceleration voiture
@IN = pedale frein
@IN = rapport vit
@OUT = position gaz

@PSPEC Controle de la vitesse de croisiere
-- b
-- moteur en marche
-- activation controle vitesse
-- desactivation controle vitesse
-- pedale accelerateur
-- stop acceleration
-- reprise acceleration voiture
-- pedale frein
-- rapport vit
-- position gaz
-- vit crois

@SI activation controle vitesse == 1 ET b !=0 ET rapport vit == 5
```

```

@ALORS SI b >= 50
@     ALORS SI b >= 50
@         ALORS vit crois := b
@         FSI
@         SI b > vit crois
@         ALORS "stop acceleration"
@         SINON SI b < vit crois
@             ALORS "reprise acceleration voiture"
@             FSI
@         FSI
@     FSI
@FSI
@
@SI "desactivation controle vitesse"
@ALORS "stop acceleration"
@FSI
@
@SI "pedale frein" == 1
@ALORS "stop acceleration"
@FSI
@
@SI "stop acceleration" == 1
@ALORS position gaz := arret acceleration
@FSI
@SI "reprise acceleration voiture" == 1
@ALORS position gaz := debut acceleration
@FSI

```

3.7. Affichage

Cette dernière tâche sert simplement à tenir informé le conducteur des différents états du système :

```

@IN = vitesse moyenne
@IN = c
@IN = affichage conso
@IN = avis d'entretien
@OUT = donnees d'affichage

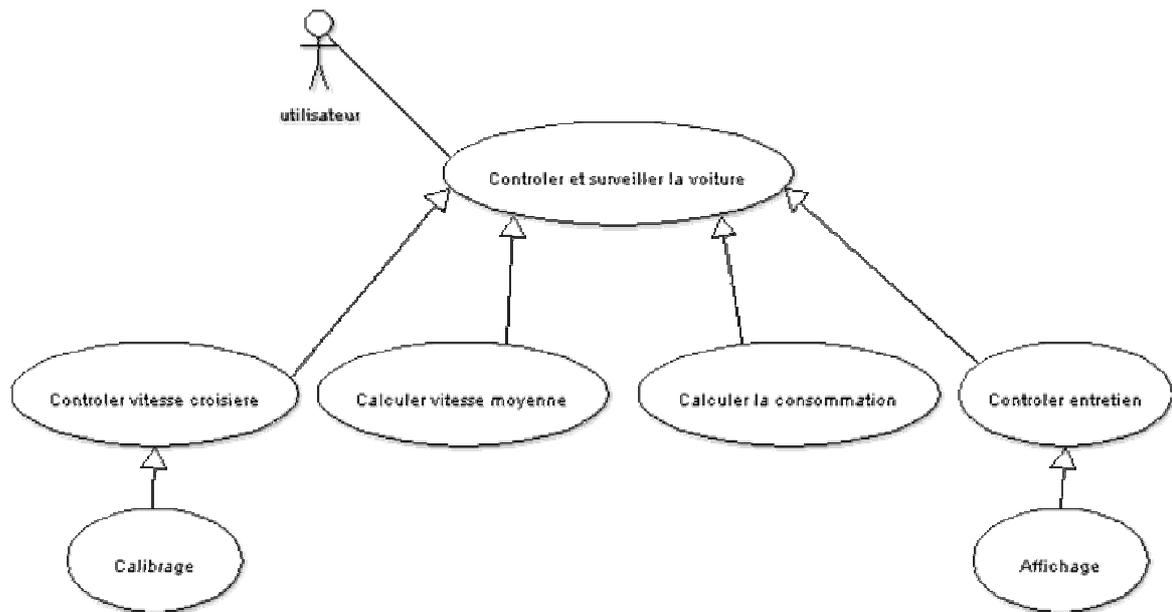
@PSPEC Process Affichage
-- vitesse moyenne
-- c
-- affichage conso
-- avis d'entretien
-- donnees d'affichage

@SI "vitesse moyenne" == 1 ALORS donnees d'affichage := vitesse moyenne
@FSI
@SI "c" == 1 ALORS donnees d'affichage := c
@FSI
@SI "affichage conso" == 1 ALORS donnees d'affichage := affichage conso
@FSI
@SI "avis d'entretien" == 1 ALORS donnees d'affichage := avis d'entretien
@FSI

```

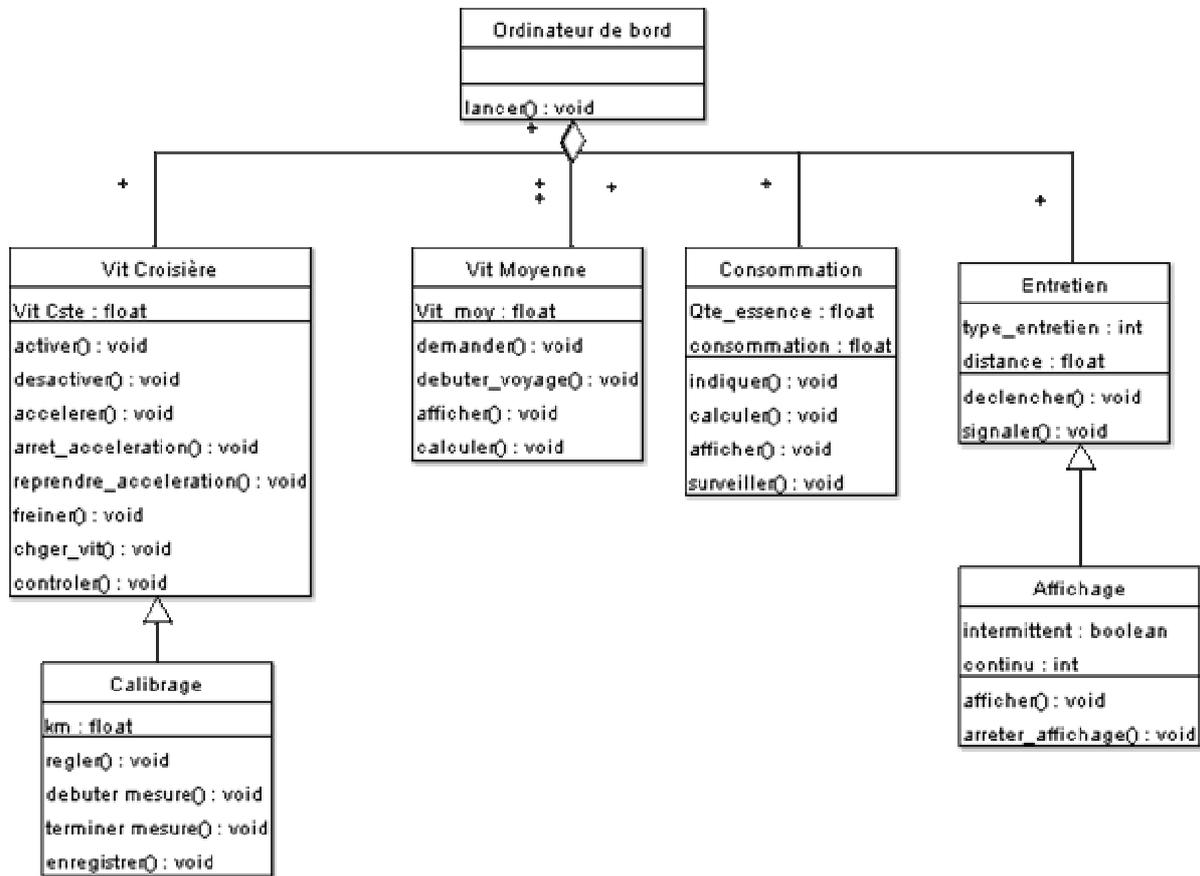
4. Diagramme de cas d'utilisation (UML)

Le schéma suivant présente les différents cas d'utilisation du système afin de mieux en définir l'ergonomie et le fonctionnement :



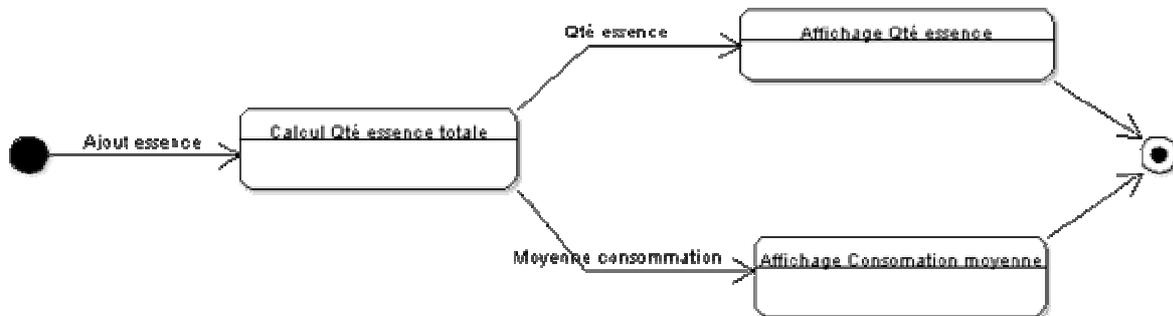
5. Diagramme de classe (UML)

Le schéma suivant nous a permis d'identifier les différents objets intervenant dans le système, et ce afin d'en optimiser le fonctionnement :



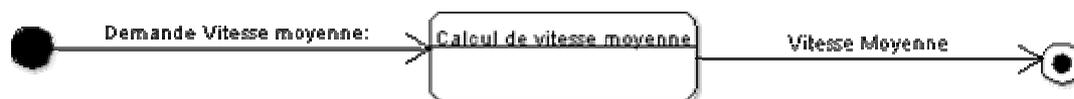
6. Diagramme d'état (UML)

6.1. Calculer la consommation



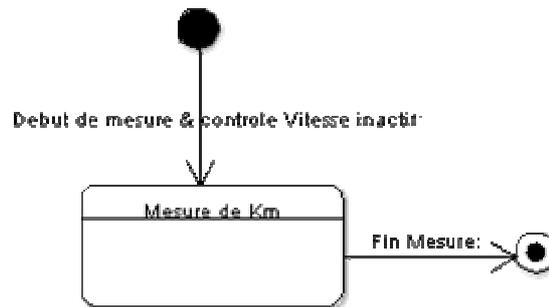
L'utilisateur renseigne la quantité d'essence ajoutée, l'ordinateur calcule ensuite la quantité d'essence totale qu'il y a dans le réservoir du véhicule. Une fois cette quantité définie le système transmet les informations sur le niveau d'essence et la consommation moyenne au module d'affichage.

6.2. Calculer la vitesse moyenne



Lorsque le module reçoit une demande de vitesse moyenne, il s'active et calcule cette moyenne à partir des informations dont il dispose puis renvoie cette information au module d'affichage.

6.3. Calibrage



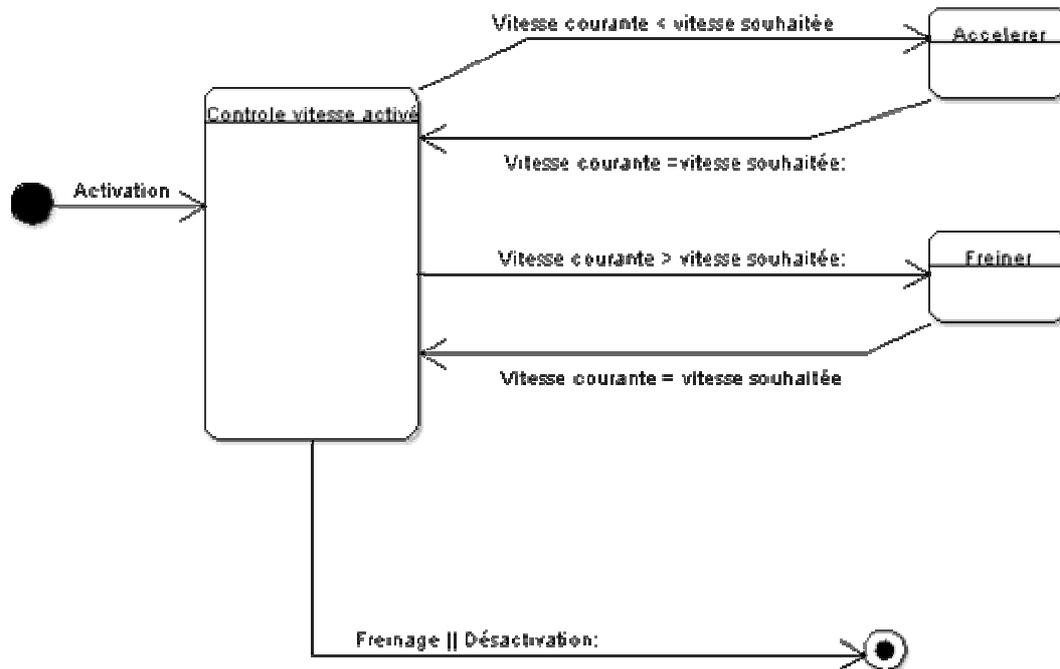
Ce module permet le calibrage de la vitesse de croisière. Le calibrage s'active sur demande de l'utilisateur, la fin de calibrage est également réalisée sur commande de l'utilisateur. Le résultat de ce module est stocké dans une variable du système pour être ensuite utilisée dans le calcul de vitesse.

6.4. Contrôler entretien



Le module est activé par le système à chaque palier de km stocké en mémoire. Ces paliers correspondent chacun à un entretien à faire par l'utilisateur. Une fois activé le module déclenche une alerte visuelle qui sera arrêtée lorsque l'utilisateur indiquera avoir réalisé l'entretien.

6.5. Contrôler la vitesse



Ce module est le module permettant le contrôle de la vitesse du véhicule. Il est activé par l'utilisateur pour ensuite fonctionner en mode de régulation autonome :

- *lorsque le véhicule va à une vitesse supérieure à la vitesse de croisière spécifiée, le module enclenche une phase de freinage/décélération de manière à retrouver sa vitesse de croisière.*
- *lorsque le véhicule va à une vitesse inférieure à la vitesse de croisière spécifiée, le module enclenche une phase d'accélération de manière à retrouver sa vitesse de croisière.*

Le module peut être désactivé de 2 manières :

- *soit l'utilisateur désengage le module manuellement*
- *soit le système repère un freinage brutal de l'utilisateur et désengage automatiquement le module.*

7. Conclusion

Ce projet a été très intéressant car il nous a permis de voir l'application des méthodes du temps réel à travers un exemple concret. En effet il nous a permis de mettre en application les méthodes SART et UML sur un système temps réel de plus en plus courant actuellement et d'ainsi percevoir la complexité d'un tel système.