

Dossier de conception de MARCO

Incrément 1

MARCO

AVERTISSEMENT :

Le présent document est à but pédagogique. Il a été réalisé par les étudiants Jordan LAUDE, Axel LORY, Hugo BURY, Lucas BODIN et Pierre MERLET de l'équipe 2 de l'option Informatique et Systèmes embarqués (ISE) de la branche Alternance de l'ESEO sous la direction de Frédéric JOUAULT et de Jérôme DELATOUR. Ce document est la propriété de ses auteurs. En dehors des activités pédagogiques de l'ESEO, ce document ne peut être diffusé ou recopié sans l'autorisation écrite de l'un de ses propriétaires.

Table des matières

Table des matières	2
1 Introduction	4
1.1 Objet	4
1.2 Portée	4
1.3 Définitions, acronymes et abréviations	5
1.4 Vue d'ensemble	6
1.5 Versions	7
2 Conception générale	8
2.1 Architecture candidate	8
2.2 Diagrammes de séquence	9
2.3 Types de données	11
2.4 Classes	13
2.4.1 Vue générale	13
2.4.2 La classe User	13
2.4.2.1 Attributs	14
2.4.2.2 Services offerts	14
2.4.3 La classe GUIPhone	14
2.4.3.1 Attributs	14
2.4.3.2 Services offerts	14
2.4.3.3 Description comportementale	15
2.4.4 La classe NetworkPhone	15
2.4.4.1 Attributs	15
2.4.4.2 Services offerts	15
2.4.5 La classe NetworkRobot	16
2.4.5.1 Attributs	16
2.4.5.2 Services offerts	16
2.4.6 La classe Archivist	16
2.4.6.1 Attributs	16
2.4.6.2 Services offerts	16
2.4.7 La classe UIRobot	16
2.4.7.1 Attributs	16
2.4.7.2 Services offerts	17
2.4.8 La classe AlgoManager	17
2.4.8.1 Attributs	17
2.4.8.2 Services offerts	17
2.4.9 La classe Driver	17
2.4.9.1 Attributs	17
2.4.9.2 Services offerts	17
2.4.9.3 Description comportementale	18
2.4.10 La classe Explorer	18
2.4.10.1 Attributs	18
2.4.10.2 Services offerts	18
2.4.11 La classe Radar	19
2.4.11.1 Attributs	19
2.4.11.2 Services offerts	19
2.4.12 La classe Mapper	19

2.4.12.1	Attributs	19
2.4.12.2	Services offerts	19
2.4.13	La classe Locator	19
2.4.13.1	Attributs	19
2.4.13.2	Services offerts	19
2.4.14	La classe Robot	20
2.4.14.1	Attributs	20
2.4.14.2	Services offerts	20
2.5	Diagrammes d'activité	20
3	Conception détaillée	20
3.1	Architecture physique	20
3.2	Description des classes	20
4	Dictionnaire du domaine	21
5	Table des figures	22

1 Introduction

1.1 Objet

Ce dossier de conception a pour objectif de définir la structuration et le comportement du projet MARCO. Il permettra de développer le projet ainsi qu'élaborer les tests.

Les éléments de conception présentés dans ce document ont été déterminés suite à l'étude du dossier de spécification.

Ce dossier de conception suit les recommandations de la norme IEEE 830 [IEEE-830_1998]. Il utilise des schémas et illustrations respectant la norme UML en version 2.5 [UML_2.5.1_2017].

1.2 Portée

Ce document décrit les éléments de conception du Système à l'étude (SaE). Il est destiné :

- à l'équipe de développement C et celle de développement Android afin de préciser l'implémentation des objets constituant le SaE.
- aux testeurs, afin qu'ils puissent élaborer les tests adéquats vérifiant la philosophie de conception adoptée par l'équipe.
- au client pour que le cadre du projet et la direction prise par l'équipe soient clairs et dans la continuité des spécifications.

1.3 Définitions, acronymes et abréviations

Les abréviations utilisées dans le présent document sont répertoriées et expliquées dans le tableau ci-dessous. Les termes utiles pour interpréter correctement ce document sont définis dans le dictionnaire du domaine section 3.3.

Terme	Description
Client	Jérôme DELATOUR
CU	Cas d'Utilisation
IEEE	(Institute of Electrical and Electronics Engineers) Association professionnelle internationale définissant entre autres des, normes dans le domaine informatique et électronique.
N.A	Non Applicable
OMG	(Object Management Group) Association professionnelle internationale définissant entre autres, des normes dans le domaine informatique.
pp.	Abréviation utilisée pour référencer une ou plusieurs page(s).
RDPM	Référentiel Documentaire Projet MARCO
SaE	(Système à l'étude) Il s'agit de l'ensemble des composants MARCO, MARCOContrôle et MARCOEmbarqué.
UML	(Unified Modeling Language) Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel.

1.4 Vue d'ensemble

Ce document de conception est structuré en plusieurs parties :

- une première partie, qui concerne la conception générale du prototype MARCOEmbarqué. Cette partie présente l'architecture candidate et donne les grands principes de fonctionnement du projet MARCOEmbarqué. Elle détaille ensuite chaque composante du système, en présentant pour chacune leur description structurelle. Une description comportementale est présente pour les composantes actives ayant une machine à états.
- une seconde partie présentera la conception détaillée. Cette partie présente les composantes du système en précisant cette fois-ci la gestion des entrées et des sorties, le multi-tâche ainsi que la gestion de la persistance.
- un dictionnaire de domaine constitue la dernière partie du document.

1.5 Versions

Le projet MARCO est réalisé en 2 incréments. Nous avons détaillé le système pour les versions 1 et 2, nous devons alors définir ce que fera la version 1 et ce que fera la version 2.

- Lors de la version 1, MARCOContrôle doit pouvoir interagir avec MARCOEmbarqué. L'ensemble des fonctionnalités présentes pour l'incrément 1 doivent être assurées.
- Lors de la version 2, toutes les fonctionnalités présentes lors de l'incrément 1 doivent être assurées. De plus, les fonctionnalités présentes pour l'incrément 2 doivent être assurées.

2 Conception générale

2.1 Architecture candidate

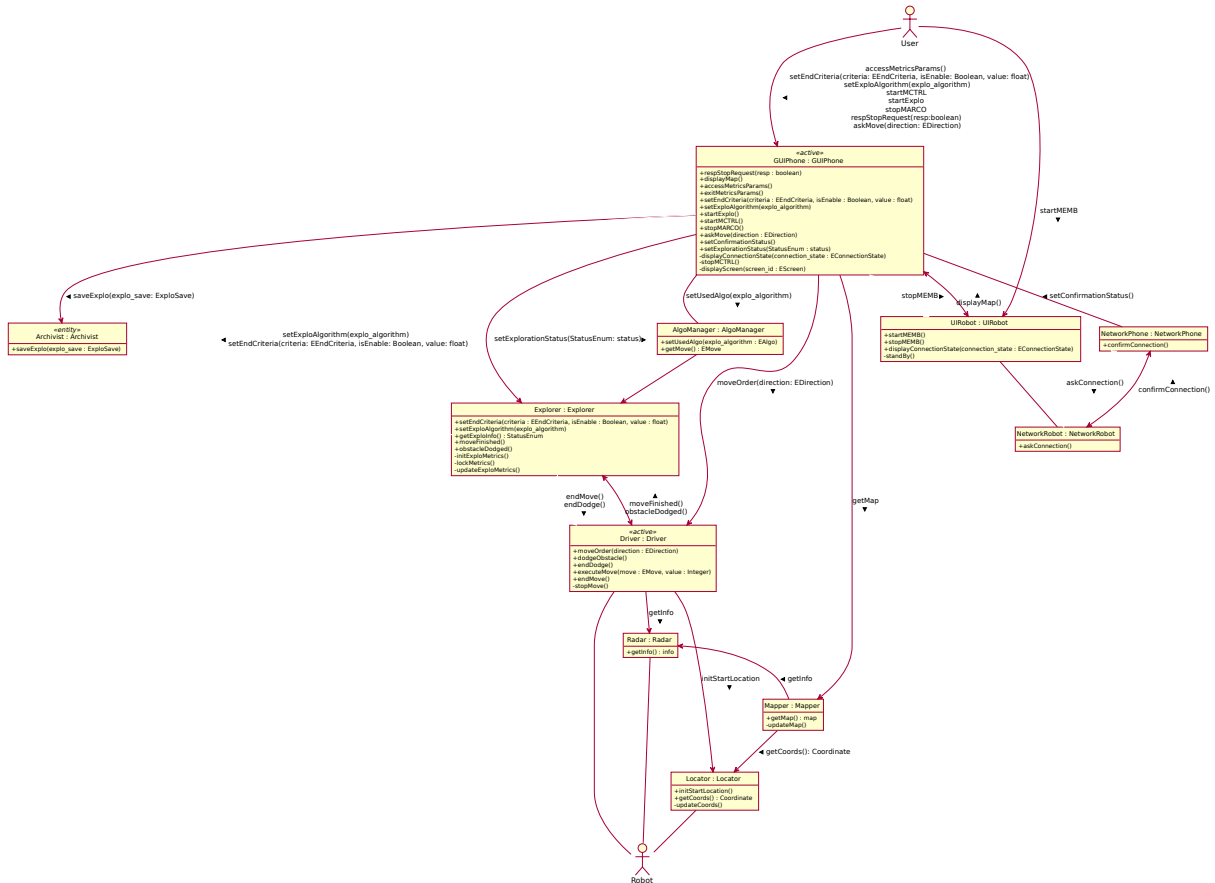


FIGURE 1 – Architecture candidate

Le diagramme de la figure 1 représente l'architecture candidate du système.

2.2 Diagrammes de séquence

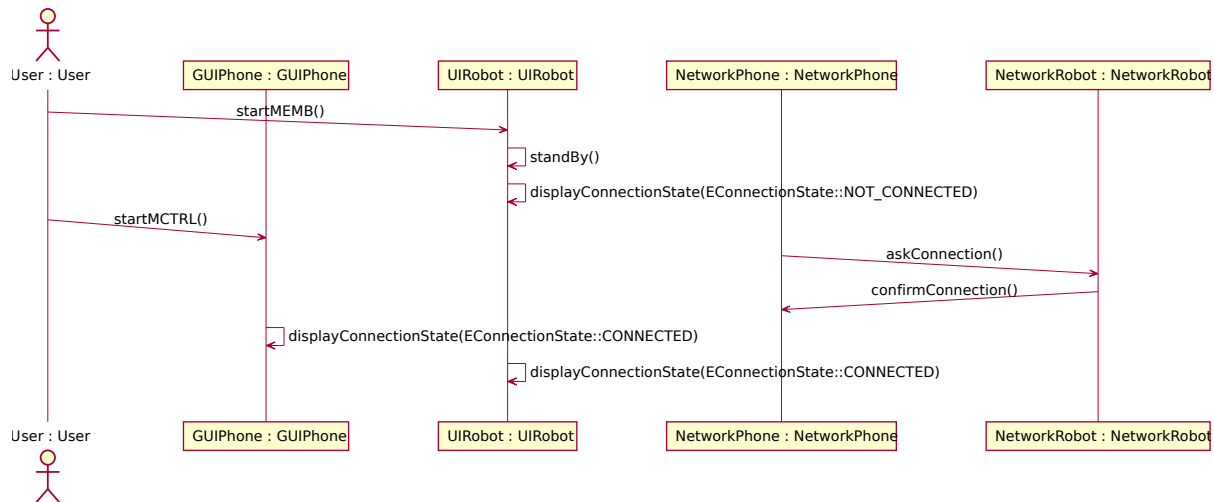


FIGURE 2 – Diagramme de séquence *CU1 INIT*

Le diagramme de la figure 2 représente le diagramme de séquence *CU1 INIT*.

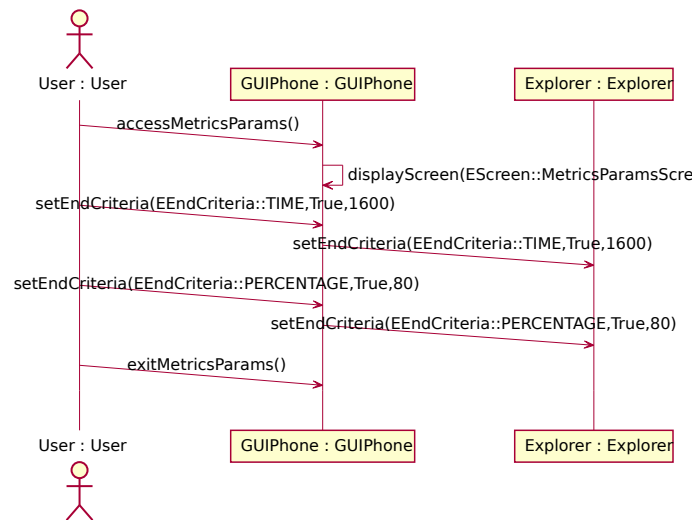
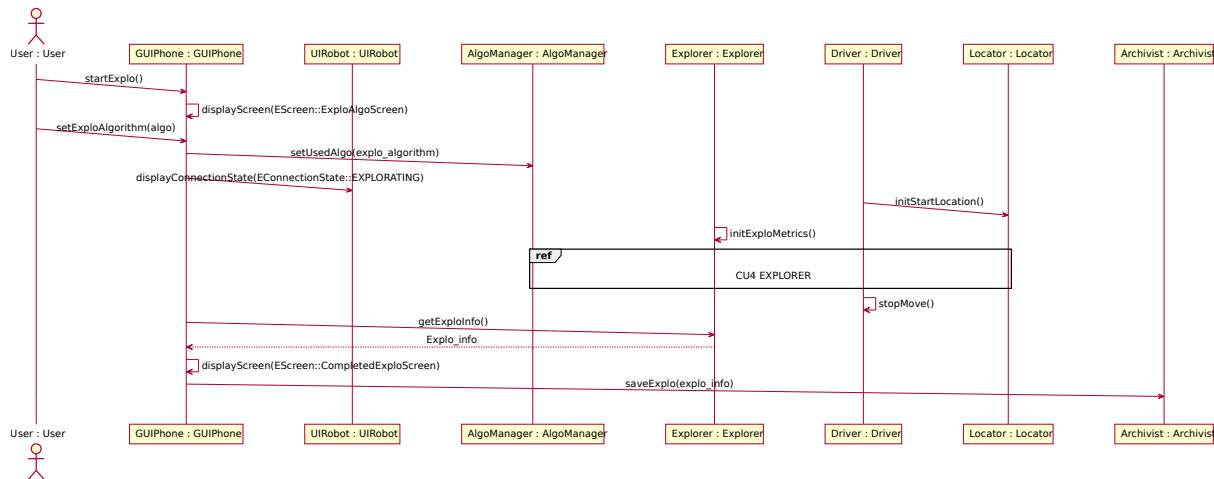
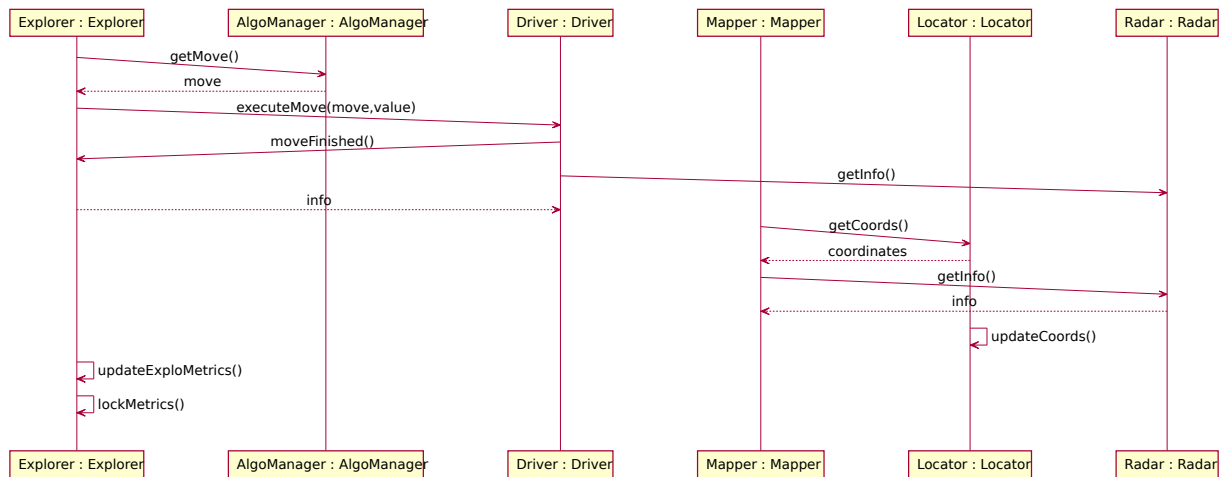


FIGURE 3 – Diagramme de séquence *CU2 MAJ EXPL*

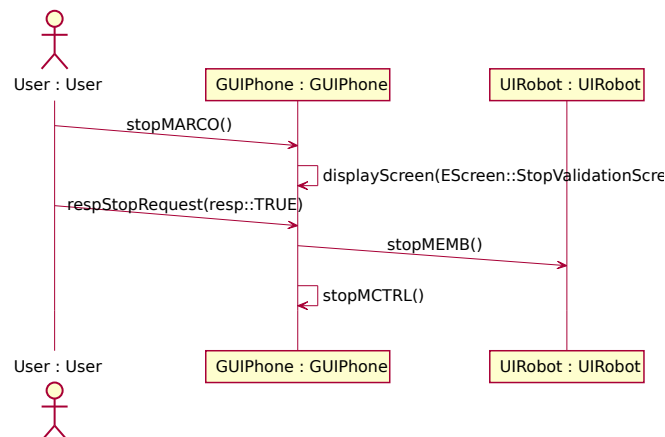
Le diagramme de la figure 3 représente le diagramme de séquence *CU2 MAJ EXPL*.

FIGURE 4 – Diagramme de séquence *CU3 LANCER EXPLO*

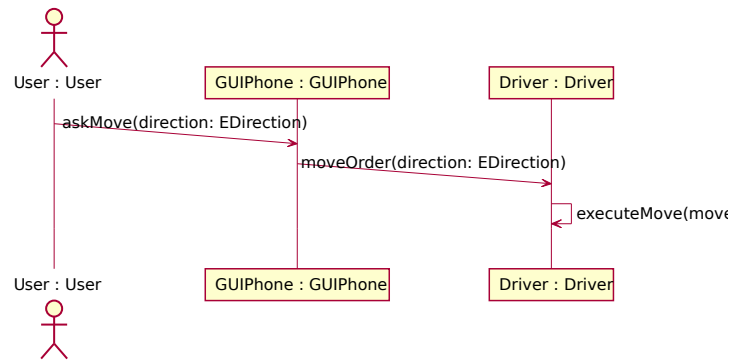
Le diagramme de la figure 4 représente le diagramme de séquence *CU3 LANCER EXPLO*.

FIGURE 5 – Diagramme de séquence *CU4 EXPLORER*

Le diagramme de la figure 5 représente le diagramme de séquence *CU4 EXPLORER*.

FIGURE 6 – Diagramme de séquence *CU5 ETEINDRE MARCO*

Le diagramme de la figure 6 représente le diagramme de séquence *CU5 ETEINDRE MARCO*.

FIGURE 7 – Diagramme de séquence *CU6 DEPLACEMENT MANUEL*

Le diagramme de la figure 7 représente le diagramme de séquence *CU6 DEPLACEMENT MANUEL*.

2.3 Types de données

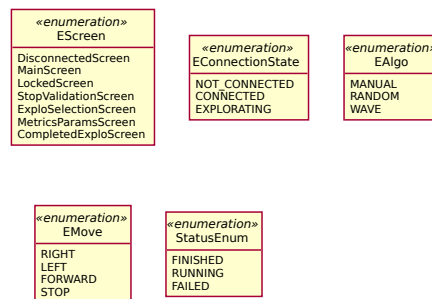


FIGURE 8 – Diagramme des types de données

Le diagramme de la figure 8 représente les types de données utilisés. L'énumération EScreen possède les littéraux suivants :

- **DisconnectedScreen**
- **MainScreen**
- **LockedScreen**
- **StopValidationScreen**
- **ExploSelectionScreen**
- **MetricsParamsScreen**
- **CompletedExploScreen**

L'énumération EConnectionState possède les littéraux suivants :

- **NOT_CONNECTED**
- **CONNECTED**
- **EXPLORATING**

L'énumération EAlgo possède les littéraux suivants :

- **MANUAL**
- **RANDOM**
- **WAVE**

L'énumération EMove possède les littéraux suivants :

- **RIGHT**
- **LEFT**
- **FORWARD**
- **STOP**

L'énumération StatusEnum possède les littéraux suivants :

- **FINISHED**
- **RUNNING**
- **FAILED**

2.4 Classes

2.4.1 Vue générale

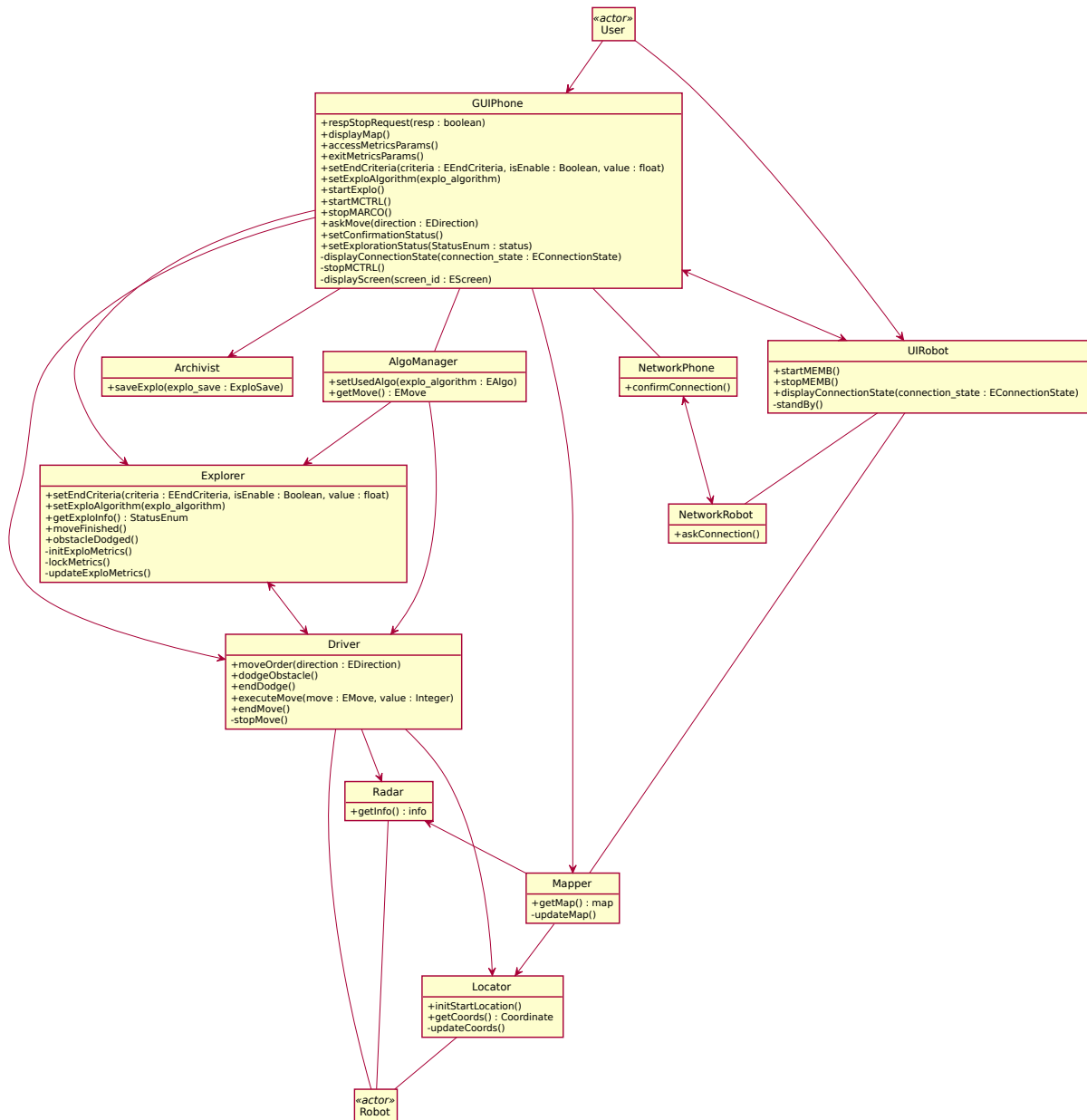


FIGURE 9 – Diagramme de classes

Le diagramme de la figure 9 représente les classes du système.

2.4.2 La classe User

Le diagramme de la figure 10 représente la classe User.



FIGURE 10 – Diagramme de la classe User

2.4.2.1 Attributs

N.A.

2.4.2.2 Services offerts

N.A.

2.4.3 La classe GUIPhone

Le diagramme de la figure 11 représente la classe GUIPhone.

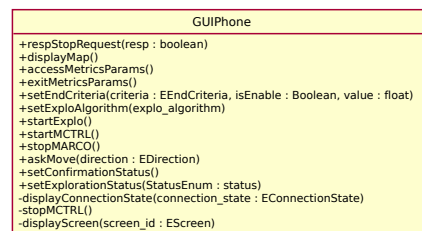


FIGURE 11 – Diagramme de la classe GUIPhone

2.4.3.1 Attributs

N.A.

2.4.3.2 Services offerts

- **respStopRequest(resp : boolean)** — Informe l'utilisateur que Robot va s'éteindre
- **displayMap()** — Affichage de la carte sur l'écran de PHONE
- **accessMetricsParams()** — Action de l'utilisateur visant à accéder au menu des métriques
- **exitMetricsParams()** — Sort du menu des paramètres des métriques
- **setEndCriteria(criteria : EEndCriteria, isEnabled : Boolean, value : float)** — Action de l'utilisateur visant à sélectionner les métriques à utiliser pour la prochaine exploration et d'entrer sa valeur
- **setExploAlgorithm(explo_algorithm)** — Action de l'utilisateur visant à sélectionner l'algorithme d'exploration
- **startExplo()** — Action visant à lancer une exploration
- **startMCTRL()** — Ordre de démarrage de M_CTRL
- **stopMARCO()** — Volonté d'extinction global du SAE
- **askMove(direction : EDirection)** — Ordre de mouvement dans une direction "direction" et de type EDirection cette méthode est intimement liée au joystick
- **setConfirmationStatus()** — Informe de l'état de la connexion
- **setExplorationStatus(statusEnum : status)** — Met à jour l'état de l'exploration sur PHONE en fonction du paramètre status
- **displayConnectionState(connection_state : EConnectionState)** — Affichage de l'état de connexion à Robot sur Phone via un logo sur l'écran l'état du logo est définie par la variable connection_state qui est du type EConnectionState

- **stopMCTRL()** — Ordre d'arrêt du logiciel embarquer M_CTRL
- **displayScreen(screen_id : EScreen)** — Méthode permettant l'affichage des divers écrans de PHONE le paramètre screen_id est une énumération de type EScreen

2.4.3.3 Description comportementale

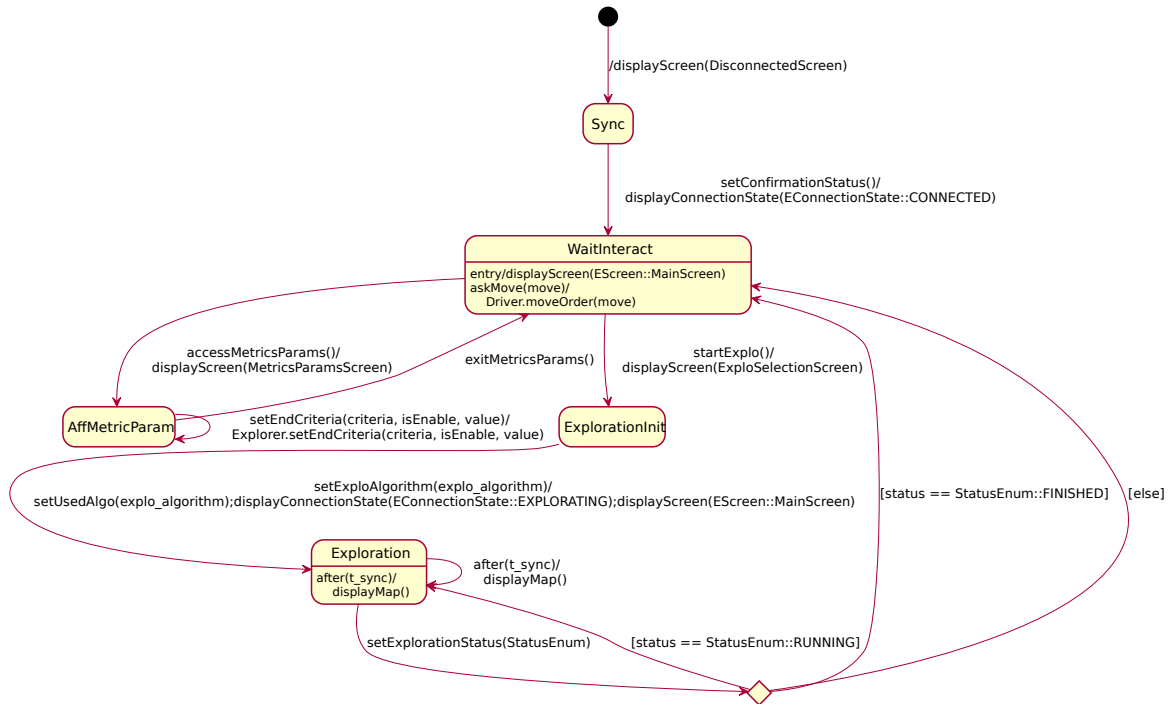


FIGURE 12 – Machine à états de *GUIPhone*

Le diagramme de la figure 12 représente la machine à états de *GUIPhone*.

2.4.4 La classe NetworkPhone

Le diagramme de la figure 13 représente la classe *NetworkPhone*.

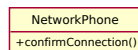


FIGURE 13 – Diagramme de la classe *NetworkPhone*

2.4.4.1 Attributs

N.A.

2.4.4.2 Services offerts

- **confirmConnection()** — Méthode de confirmation de connection a Robot

2.4.5 La classe NetworkRobot

Le diagramme de la figure 14 représente la classe NetworkRobot.

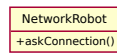


FIGURE 14 – Diagramme de la classe NetworkRobot

2.4.5.1 Attributs

N.A.

2.4.5.2 Services offerts

- **askConnection()** — Méthode pour demander une connexion à PHONE

2.4.6 La classe Archivist

Le diagramme de la figure 15 représente la classe Archivist.

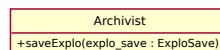


FIGURE 15 – Diagramme de la classe Archivist

2.4.6.1 Attributs

N.A.

2.4.6.2 Services offerts

- **saveExplo(explo_save : ExploSave)** — Sauvegarde les données de l'exploration courante

2.4.7 La classe UIRobot

Le diagramme de la figure 16 représente la classe UIRobot.

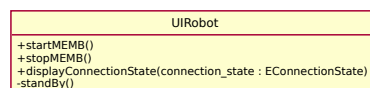


FIGURE 16 – Diagramme de la classe UIRobot

2.4.7.1 Attributs

N.A.

2.4.7.2 Services offerts

- **startMEMB()** — Démarre le logiciel MEMB
- **stopMEMB()** — Arrête le logiciel MEMB
- **displayConnectionState(connection_state : EConnectionState)** — Affiche l'état de la connexion à PHONE sur le Robot
- **standBy()** — Met le Robot en mode veille

2.4.8 La classe AlgoManager

Le diagramme de la figure 17 représente la classe AlgoManager.

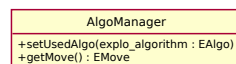


FIGURE 17 – Diagramme de la classe AlgoManager

2.4.8.1 Attributs

N.A.

2.4.8.2 Services offerts

- **setUsedAlgo(exploration_algorithm : EAlgo)** — Définit l'algorithme d'exploration utilisé
- **getMove() : EMove** — Retourne le prochain mouvement à effectuer selon l'algorithme

2.4.9 La classe Driver

Le diagramme de la figure 18 représente la classe Driver.

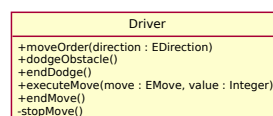


FIGURE 18 – Diagramme de la classe Driver

2.4.9.1 Attributs

N.A.

2.4.9.2 Services offerts

- **moveOrder(direction : EDirection)** — Donne un ordre de mouvement dans la direction indiquée
- **dodgeObstacle()** — Effectue une action pour éviter un obstacle
- **endDodge()** — Termine l'évitement d'obstacle
- **executeMove(move : EMove, value : Integer)** — Exécute le mouvement spécifié avec la valeur indiquée
- **endMove()** — Termine le mouvement en cours
- **stopMove()** — Arrête le mouvement en cours

2.4.9.3 Description comportementale

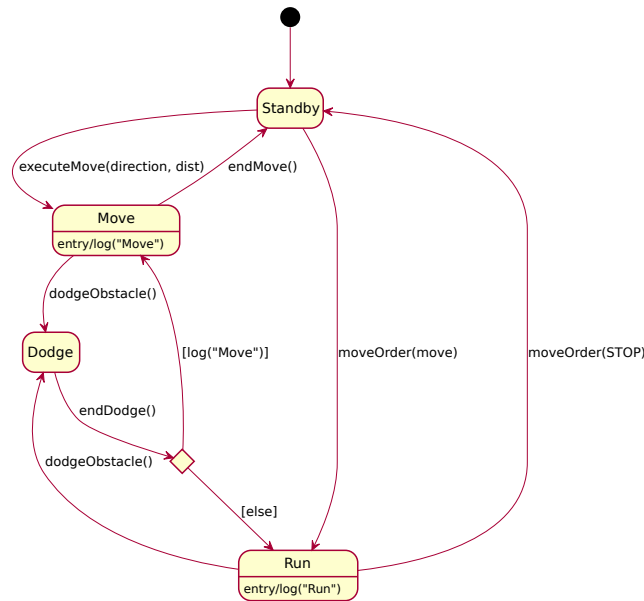


FIGURE 19 – Machine à états de *Driver*

Le diagramme de la figure 19 représente la machine à états de *Driver*.

2.4.10 La classe Explorer

Le diagramme de la figure 20 représente la classe Explorer.

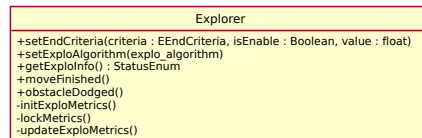


FIGURE 20 – Diagramme de la classe Explorer

2.4.10.1 Attributs

N.A.

2.4.10.2 Services offerts

- **setEndCriteria(criteria : EEndCriteria, isEnabled : Boolean, value : float)** — Définit les critères de fin d'exploration
- **setExploAlgorithm(explo_algorithm)** — Définit l'algorithme d'exploration à utiliser
- **getExploInfo() : StatusEnum** — Retourne l'état courant de l'exploration
- **moveFinished()** — Indique que le mouvement en cours est terminé
- **obstacleDodged()** — Indique que l'obstacle a été évité
- **initExploMetrics()** — Initialise les métriques d'exploration
- **lockMetrics()** — Verrouille les métriques d'exploration
- **updateExploMetrics()** — Met à jour les métriques d'exploration

2.4.11 La classe Radar

Le diagramme de la figure 21 représente la classe Radar.



FIGURE 21 – Diagramme de la classe Radar

2.4.11.1 Attributs

N.A.

2.4.11.2 Services offerts

- **getInfo() : info** — Retourne les informations recueillies par le radar

2.4.12 La classe Mapper

Le diagramme de la figure 22 représente la classe Mapper.

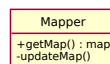


FIGURE 22 – Diagramme de la classe Mapper

2.4.12.1 Attributs

N.A.

2.4.12.2 Services offerts

- **getMap() : map** — Retourne la carte actuelle
- **updateMap()** — Met à jour la carte avec les nouvelles informations

2.4.13 La classe Locator

Le diagramme de la figure 23 représente la classe Locator.

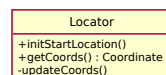


FIGURE 23 – Diagramme de la classe Locator

2.4.13.1 Attributs

N.A.

2.4.13.2 Services offerts

- **initStartLocation()** — Initialise la position de départ
- **getCoords() : Coordinate** — Retourne les coordonnées actuelles du robot
- **updateCoords()** — Met à jour les coordonnées du robot

2.4.14 La classe Robot

Le diagramme de la figure 24 représente la classe Robot.



FIGURE 24 – Diagramme de la classe Robot

2.4.14.1 Attributs

N.A.

2.4.14.2 Services offerts

N.A.

2.5 Diagrammes d'activité

3 Conception détaillée

3.1 Architecture physique

3.2 Description des classes

4 Dictionnaire du domaine

Les abréviations utilisées dans le présent document sont répertoriées et expliquées dans le tableau ci-dessous. Les termes utiles pour interpréter correctement ce document sont définis dans le dictionnaire du domaine section 3.3.

C

Client

Personne représentée par M. Jérôme DELATOUR Entité requérant les services du projet, susceptible d'avoir des consignes spécifiques quant à l'apparence globale du SaE et la charte graphique de ses IHMs.

M

MARCO

Nom du projet.

MARCOContrôle

Application Android pour contrôler MARCO.

MARCOEmbarqué

Logiciel embarqué dans MARCO.

5 Table des figures

1	Architecture candidate	8
2	Diagramme de séquence <i>CU1 INIT</i>	9
3	Diagramme de séquence <i>CU2 MAJ EXPLO</i>	9
4	Diagramme de séquence <i>CU3 LANCER EXPLO</i>	10
5	Diagramme de séquence <i>CU4 EXPLORER</i>	10
6	Diagramme de séquence <i>CU5 ETEINDRE MARCO</i>	10
7	Diagramme de séquence <i>CU6 DEPLACEMENT MANUEL</i>	11
8	Diagramme des types de données	11
9	Diagramme de classes	13
10	Diagramme de la classe User	13
11	Diagramme de la classe GUIPhone	14
12	Machine à états de <i>GUIPhone</i>	15
13	Diagramme de la classe NetworkPhone	15
14	Diagramme de la classe NetworkRobot	16
15	Diagramme de la classe Archivist	16
16	Diagramme de la classe UIRobot	16
17	Diagramme de la classe AlgoManager	17
18	Diagramme de la classe Driver	17
19	Machine à états de <i>Driver</i>	18
20	Diagramme de la classe Explorer	18
21	Diagramme de la classe Radar	19
22	Diagramme de la classe Mapper	19
23	Diagramme de la classe Locator	19
24	Diagramme de la classe Robot	20