

# Dossier de conception de SmartHouseSim

## Incrément 1



**AVERTISSEMENT :**

Le présent document est à but pédagogique. Il a été réalisé par les étudiants Théo FAUCHER, Eliot COULON, Valentin HUBERT, Antoine LARGEAUD, Quentin SOUTELO et Léo LAMANAC de l'équipe B2\_2024 de l'option Systèmes Embarqués (SE) 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.

Le modèle de ce document a été mis à disposition par Edouard GAUTIER étudiant de l'ESEO selon le plan rédigé par Jérôme DELATOUR.

La société UKIO n'a pas d'existence légale, il s'agit d'une société fictive créée pour des besoins pédagogiques.

## Table des versions

Date	Description	Auteur	Version	Révision
18/03/2023	Création du document	Théo FAUCHER	0.0	0
18/03/2023	Ajout de l'introduction	Quentin SOUTELO	0.0	1
28/03/2023	Correction de l'introduction	Léo LAMANAC	0.0	2

# Table des matières

<b>Table des versions</b>	<b>2</b>
<b>Table des matières</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Objet . . . . .	5
1.2 Portée . . . . .	5
1.3 Définitions, acronymes et abréviations . . . . .	6
1.4 Références . . . . .	6
1.5 Vue d'ensemble . . . . .	8
1.6 Versions . . . . .	9
<b>2 Conception générale</b>	<b>10</b>
2.1 Architecture candidate . . . . .	10
2.2 Diagrammes de séquence . . . . .	11
2.3 Types de données . . . . .	11
2.4 Classes . . . . .	12
2.4.1 Vue générale . . . . .	12
2.4.2 La classe USER . . . . .	12
2.4.2.1 Attributs . . . . .	13
2.4.2.2 Services offerts . . . . .	13
2.4.3 La classe GUIPhone . . . . .	13
2.4.3.1 Attributs . . . . .	13
2.4.3.2 Services offerts . . . . .	13
2.4.4 La classe ARCHIVIST . . . . .	14
2.4.4.1 Attributs . . . . .	14
2.4.4.2 Services offerts . . . . .	14
2.4.5 La classe UIRobot . . . . .	14
2.4.5.1 Attributs . . . . .	14
2.4.5.2 Services offerts . . . . .	14
2.4.6 La classe DRIVER . . . . .	14
2.4.6.1 Attributs . . . . .	14
2.4.6.2 Services offerts . . . . .	15
2.4.7 La classe EXPLORER . . . . .	15
2.4.7.1 Attributs . . . . .	15
2.4.7.2 Services offerts . . . . .	15
2.4.8 La classe RADAR . . . . .	15
2.4.8.1 Attributs . . . . .	15
2.4.8.2 Services offerts . . . . .	15
2.4.9 La classe MAPPER . . . . .	16
2.4.9.1 Attributs . . . . .	16
2.4.9.2 Services offerts . . . . .	16
2.4.10 La classe LOCATOR . . . . .	16
2.4.10.1 Attributs . . . . .	16
2.4.10.2 Services offerts . . . . .	16
2.4.11 La classe ROBOT . . . . .	16
2.4.11.1 Attributs . . . . .	16
2.4.11.2 Services offerts . . . . .	17

2.5	Diagrammes d'activité . . . . .	17
<b>3</b>	<b>Conception détaillée</b>	<b>17</b>
3.1	Architecture physique . . . . .	17
3.2	Description des classes . . . . .	17
<b>4</b>	<b>Dictionnaire du domaine</b>	<b>18</b>
<b>5</b>	<b>Table des figures</b>	<b>19</b>

# 1 Introduction

## 1.1 Objet

Ce dossier de conception a pour objectif de définir la structuration et le comportement du projet SmartHouseSim. Il permettra à UKIO 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 [SPEC\_B2].

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]. Il respecte les exigences du Plan d'Assurance Qualité Logicielle (PAQL) défini par l'équipe B2 [PAQL\_B2].

## 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.
- aux auditeurs de la société FORMATO lors de leurs différents suivis du projet.
- 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 CU	Entreprise Davidson Consulting 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).
RDPS	Référentiel Documentaire Projet SmartHouseSim
SaE	(Système à l'étude) Il s'agit de l'ensemble des composants SmartHouseSoft, SmartHouseApp et SmartHouseModel.
UML	(Unified Modeling Language) Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel.

### 1.4 Références

Voici un tableau récapitulatif des documents utilisés pour la réalisation de ce dossier de spécification, ainsi que des liens pour accéder aux éventuelles sources.

Nom du document	Référence
[CDC_Davidson_prose2023_v1.2]	Société Davidson Consulting, « Plateforme de simulation d'effacement énergétique d'un foyer », version 1.2, disponible sur le RDPS.
[CR_09_02]	Société UKIO, "Compte-rendu de la réunion du 09/02/2023", disponible sur le RDPS.
[SPEC_B2]	Société UKIO, "Dossier de spécification du projet SmartHouseSoft", Projet SmartHouseSoft, disponible sur le RDPS.
[PAQL_B2]	Société UKIO, "Plan d'Assurance Qualité Logicielle", Projet SmartHouseSoft, disponible sur le RDPS.

Suite à la page suivante...

... suite de la page précédente.

Nom du document	Référence
[ISO/IEC/IEEE 29148 :2018]	ISO/IEC/IEEE, "International Standard, Systems and software engineering - Life cycle processes - Requirements engineering", 2018, <a href="https://standards.ieee.org/standard/29148-2018.html">https://standards.ieee.org/standard/29148-2018.html</a> .
[UML_2.5.1_2017]	OMG, "Unified Modeling Language", version 2.5.1, 2017.



## 1.5 Vue d'ensemble

Ce document de conception est structuré en plusieurs parties :

- une première partie, qui concerne la conception générale du prototype SmartHouseSoft. Cette partie présente l'architecture candidate et donne les grands principes de fonctionnement du projet SmartHouseSoft. 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.

## 1.6 Versions

Le projet SmartHouseSim 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, SmartHouseApp et SmartHouseModel doivent pouvoir interagir avec SmartHouseSoft. 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

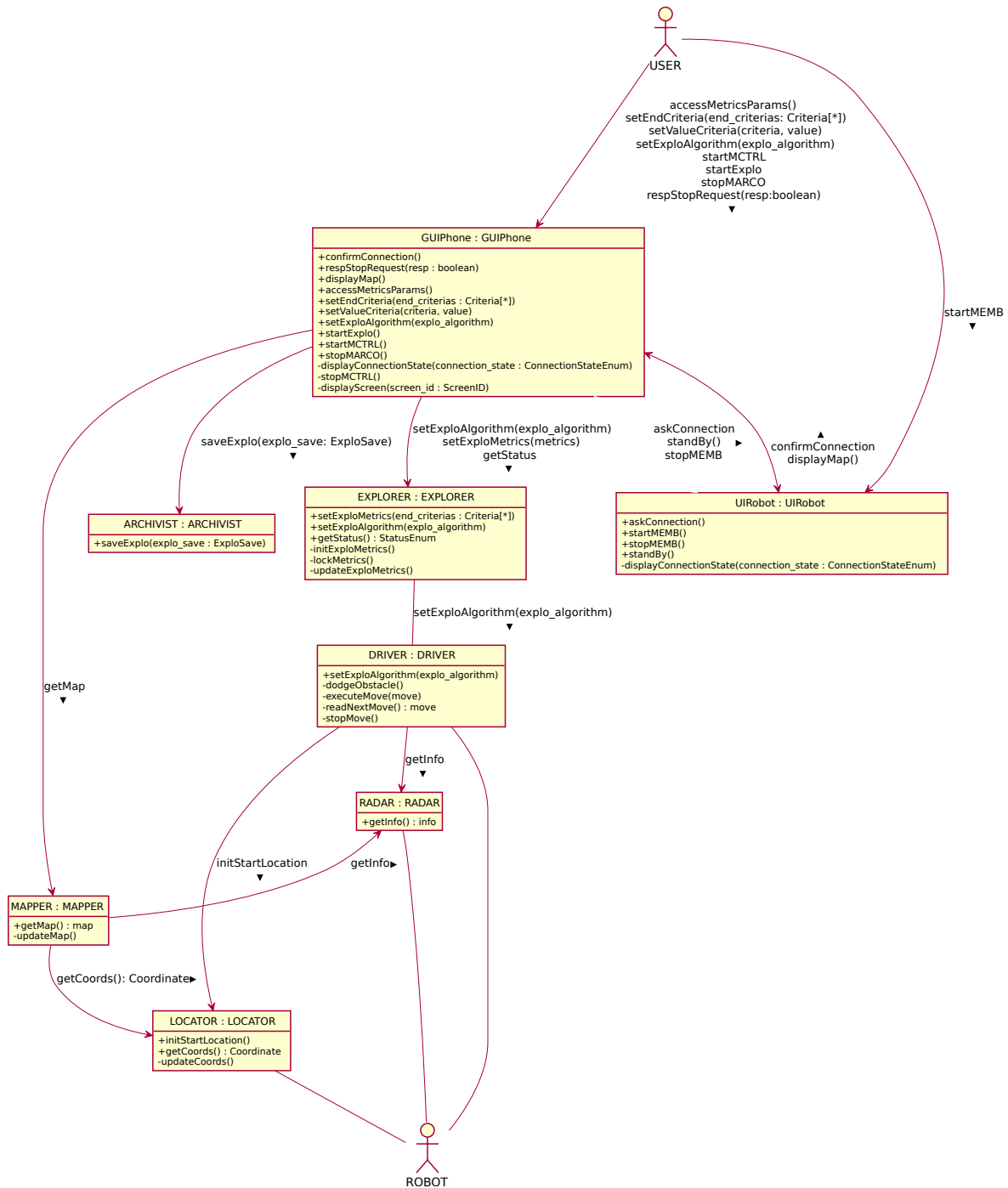


FIGURE 1 – Architecture candidate

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



- DisconnectedScreen
- MainScreen
- LockedScreen
- ExploSelectionScreen
- MetricsParamsScreen

## 2.4 Classes

### 2.4.1 Vue générale

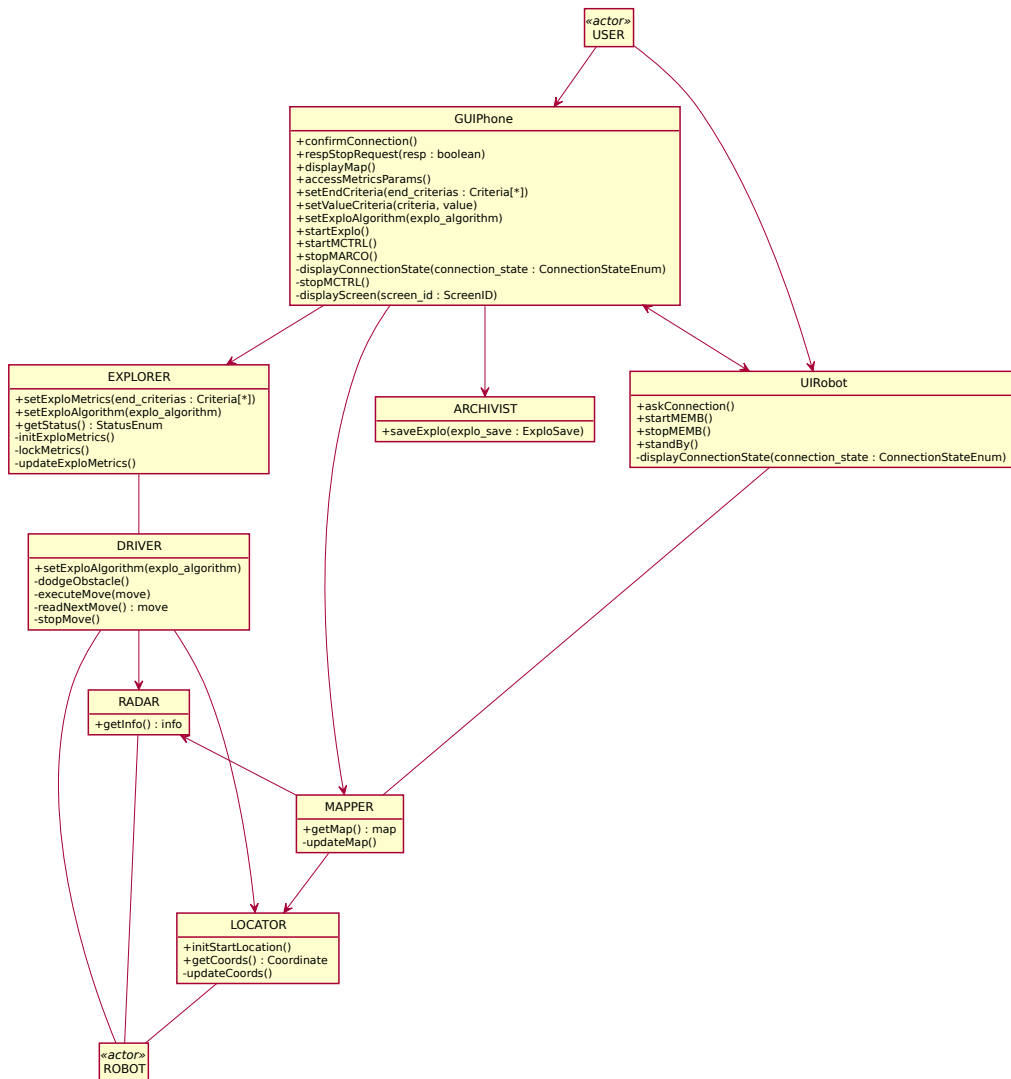


FIGURE 6 – Diagramme de classes

Le diagramme de la figure 6 représente les classes du système.

### 2.4.2 La classe USER

Le diagramme de la figure 7 représente la classe USER.



FIGURE 7 – 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 8 représente la classe GUIPhone.

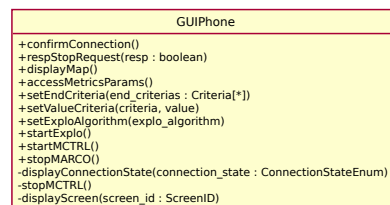


FIGURE 8 – Diagramme de la classe GUIPhone

#### 2.4.3.1 Attributs

N.A.

#### 2.4.3.2 Services offerts

- **confirmConnection()**
- **respStopRequest(resp : boolean)**
- **displayMap()**
- **accessMetricsParams()**
- **setEndCriteria(end\_criterias : Criteria[\*])**
- **setValueCriteria(criteria, value)**
- **setExploAlgorithm(explo\_algorithm)**
- **startExplo()**
- **startMCTRL()**
- **stopMARCO()**
- **displayConnectionState(connection\_state : ConnectionStateEnum)**
- **stopMCTRL()**
- **displayScreen(screen\_id : ScreenID)**

#### 2.4.4 La classe ARCHIVIST

Le diagramme de la figure 9 représente la classe ARCHIVIST.



FIGURE 9 – Diagramme de la classe ARCHIVIST

##### 2.4.4.1 Attributs

N.A.

##### 2.4.4.2 Services offerts

- **saveExplo(explo\_save : ExploSave)**

#### 2.4.5 La classe UIRobot

Le diagramme de la figure 10 représente la classe UIRobot.

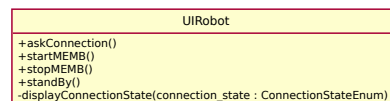


FIGURE 10 – Diagramme de la classe UIRobot

##### 2.4.5.1 Attributs

N.A.

##### 2.4.5.2 Services offerts

- **askConnection()**
- **startMEMB()**
- **stopMEMB()**
- **standBy()**
- **displayConnectionState(connection\_state : ConnectionStateEnum)**

#### 2.4.6 La classe DRIVER

Le diagramme de la figure 11 représente la classe DRIVER.

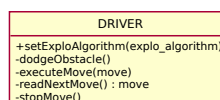


FIGURE 11 – Diagramme de la classe DRIVER

##### 2.4.6.1 Attributs

N.A.

#### 2.4.6.2 Services offerts

- **setExploAlgorithm(explo\_algorithm)**
- **dodgeObstacle()**
- **executeMove(move)**
- **readNextMove() : move**
- **stopMove()**

#### 2.4.7 La classe EXPLORER

Le diagramme de la figure 12 représente la classe EXPLORER.

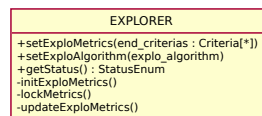


FIGURE 12 – Diagramme de la classe EXPLORER

#### 2.4.7.1 Attributs

N.A.

#### 2.4.7.2 Services offerts

- **setExploMetrics(end\_criterias : Criteria[\*])**
- **setExploAlgorithm(explo\_algorithm)**
- **getStatus() : StatusEnum**
- **initExploMetrics()**
- **lockMetrics()**
- **updateExploMetrics()**

#### 2.4.8 La classe RADAR

Le diagramme de la figure 13 représente la classe RADAR.



FIGURE 13 – Diagramme de la classe RADAR

#### 2.4.8.1 Attributs

N.A.

#### 2.4.8.2 Services offerts

- **getInfo() : info**



### 2.4.9 La classe MAPPER

Le diagramme de la figure 14 représente la classe MAPPER.

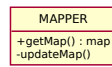


FIGURE 14 – Diagramme de la classe MAPPER

#### 2.4.9.1 Attributs

N.A.

#### 2.4.9.2 Services offerts

- **getMap() : map**
- **updateMap()**

### 2.4.10 La classe LOCATOR

Le diagramme de la figure 15 représente la classe LOCATOR.

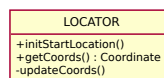


FIGURE 15 – Diagramme de la classe LOCATOR

#### 2.4.10.1 Attributs

N.A.

#### 2.4.10.2 Services offerts

- **initStartLocation()**
- **getCoords() : Coordinate**
- **updateCoords()**

### 2.4.11 La classe ROBOT

Le diagramme de la figure 16 représente la classe ROBOT.



FIGURE 16 – Diagramme de la classe ROBOT

#### 2.4.11.1 Attributs

N.A.

#### 2.4.11.2 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.

### A

#### **Algorithme**

Code qui régule les pics de consommation du foyer.

#### **Appareil**

Ensemble des appareils pouvant être pilotés suite à une action de Testeur modifiant ainsi son état et sa représentation physique et virtuelle sur SmartHouseModel et/ou SmartHouseApp.

### E

#### **Entreprise cliente**

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.

### O

#### **OpenSTLinux**

OpenSTLinux est une distribution Linux inspirée de OpenEmbedded qui est un environnement de compilation utilisé pour créer des distributions Linux pour les appareils embarqués.

### S

#### **Simulation**

Processus principal du SaE, représentation de la consommation et production électrique possible d'une maison sur une durée équivalente à  $DureeSimulation * NBJour$ .

#### **SmartHouseApp**

Application Android sur la tablette.

#### **SmartHouseControl**

Carte électronique de développement STM32MP157C-DK2 avec la distribution Linux OpenSTLinux 5.15.

#### **SmartHouseModel**

Maquette du foyer.

#### **SmartHouseSim**

Nom du projet.

#### **SmartHouseSoft**

Logiciel embarqué dans SmartHouseControl.

#### **SmartHouseTablet**

Tablette Asus Iconia Tab 10 A3-A40 avec Android 6.0.

### T

#### **Testeur**

Utilisateur de SmartHouseSim. Celui-ci a la possibilité d'interagir et de paramétrer SmartHouseSim.

## 5 Table des figures

1	Architecture candidate . . . . .	10
2	Diagramme de séquence <i>CU1 INIT</i> . . . . .	11
3	Diagramme de séquence <i>CU2 MAJ EXPLO</i> . . . . .	11
4	Diagramme de séquence <i>CU3 LAUNCH EXPLO</i> . . . . .	11
5	Diagramme des types de données . . . . .	11
6	Diagramme de classes . . . . .	12
7	Diagramme de la classe USER . . . . .	13
8	Diagramme de la classe GUIPhone . . . . .	13
9	Diagramme de la classe ARCHIVIST . . . . .	14
10	Diagramme de la classe UIRobot . . . . .	14
11	Diagramme de la classe DRIVER . . . . .	14
12	Diagramme de la classe EXPLORER . . . . .	15
13	Diagramme de la classe RADAR . . . . .	15
14	Diagramme de la classe MAPPER . . . . .	16
15	Diagramme de la classe LOCATOR . . . . .	16
16	Diagramme de la classe ROBOT . . . . .	16