

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

Table des versions	2
Table des matières	3
1 Introduction	4
1.1 Objet	4
1.2 Portée	4
1.3 Définitions, acronymes et abréviations	5
1.4 Références	5
1.5 Vue d'ensemble	6
1.6 Versions	7
2 Conception générale	8
2.1 Architecture candidate	8
2.2 Diagrammes de séquence	9
2.3 Types de données	9
2.4 Classes	10
2.4.1 Vue générale	10
2.4.2 La classe USER	10
2.4.2.1 Attributs	10
2.4.2.2 Services offerts	10
2.4.3 La classe GUI _P HONE	11
2.4.3.1 Attributs	11
2.4.3.2 Services offerts	11
2.4.4 La classe ARCHIVIST	11
2.4.4.1 Attributs	11
2.4.4.2 Services offerts	11
2.4.5 La classe UI _R OBOT	11
2.4.5.1 Attributs	11
2.4.5.2 Services offerts	11
2.4.6 La classe DRIVER	11
2.4.6.1 Attributs	12
2.4.6.2 Services offerts	12
2.4.7 La classe RADAR	12
2.4.7.1 Attributs	12
2.4.7.2 Services offerts	12
2.4.8 La classe MAPPER	12
2.4.8.1 Attributs	12
2.4.8.2 Services offerts	12
2.4.9 La classe LOCATOR	13
2.4.9.1 Attributs	13
2.4.9.2 Services offerts	13
2.4.10 La classe ROBOT	13
2.4.10.1 Attributs	13
2.4.10.2 Services offerts	13
2.5 Diagrammes d'activité	13

3	Conception détaillée	13
3.1	Architecture physique	13
3.2	Description des classes	13
4	Dictionnaire du domaine	14
5	Table des figures	15

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, https://standards.ieee.org/standard/29148-2018.html .
[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

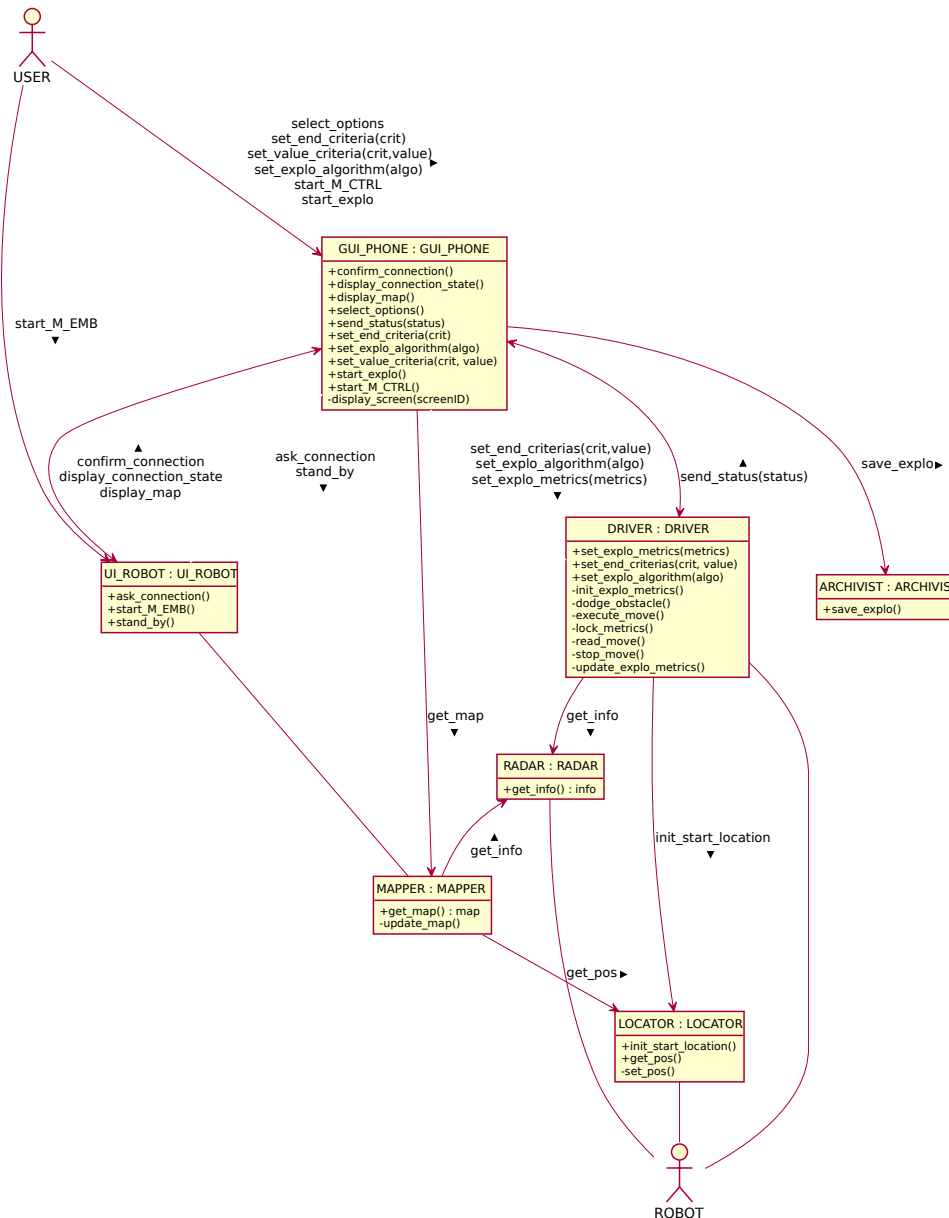


FIGURE 1 – Architecture candidate

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

2.2 Diagrammes de séquence

2.3 Types de données

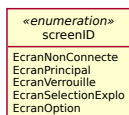


FIGURE 2 – Diagramme des types de données

Le diagramme de la figure 2 représente les types de données utilisés. L'énumération screenID possède les littéraux suivants :

- **EcranNonConnecte**
- **EcranPrincipal**
- **EcranVerrouille**
- **EcranSelectionExplo**
- **EcranOption**

2.4 Classes

2.4.1 Vue générale

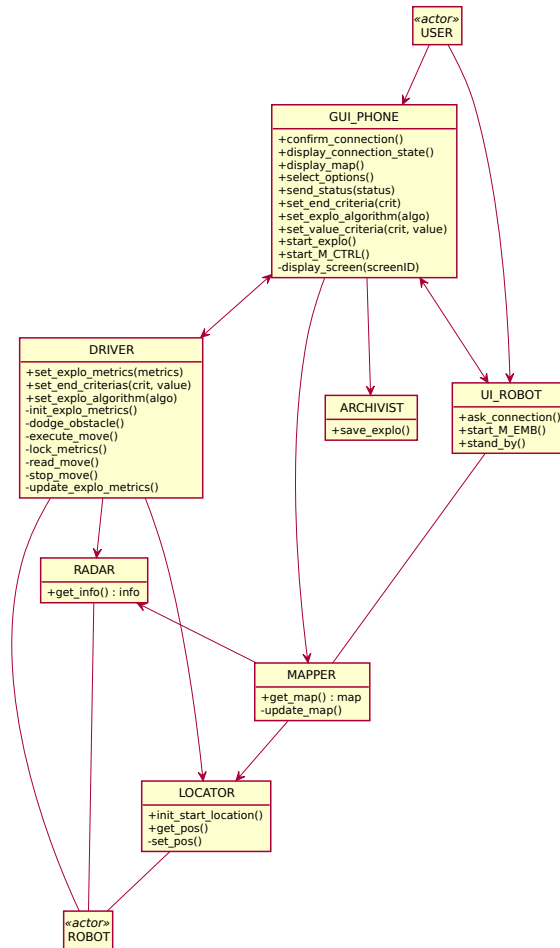


FIGURE 3 – Diagramme de classes

Le diagramme de la figure 3 représente les classes du système.

2.4.2 La classe USER

Le diagramme de la figure 4 représente la classe USER.



FIGURE 4 – 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 GUI_PHONE

Le diagramme de la ?? représente la classe GUI_PHONE .

2.4.3.1 Attributs

PHONEProperties

2.4.3.2 Services offerts

PHONEOperations

2.4.4 La classe ARCHIVIST

Le diagramme de la figure 6 représente la classe ARCHIVIST.



FIGURE 6 – Diagramme de la classe ARCHIVIST

2.4.4.1 Attributs

N.A.

2.4.4.2 Services offerts

- `save_explo()`

2.4.5 La classe UI_{ROBOT}

Le diagramme de la ?? représente la classe UI_{ROBOT} .

2.4.5.1 Attributs

ROBOTProperties

2.4.5.2 Services offerts

ROBOTOperations

2.4.6 La classe DRIVER

Le diagramme de la figure 8 représente la classe DRIVER.

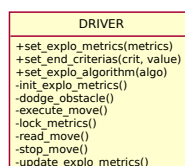


FIGURE 8 – Diagramme de la classe DRIVER

2.4.6.1 Attributs

N.A.

2.4.6.2 Services offerts

- `set_explo_metrics(metrics)`
- `set_end_criterias(crit, value)`
- `set_explo_algorithm(algo)`
- `init_explo_metrics()`
- `dodge_obstacle()`
- `execute_move()`
- `lock_metrics()`
- `read_move()`
- `stop_move()`
- `update_explo_metrics()`

2.4.7 La classe RADAR

Le diagramme de la figure 9 représente la classe RADAR.

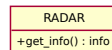


FIGURE 9 – Diagramme de la classe RADAR

2.4.7.1 Attributs

N.A.

2.4.7.2 Services offerts

- `get_info() : info`

2.4.8 La classe MAPPER

Le diagramme de la figure 10 représente la classe MAPPER.

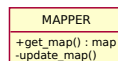


FIGURE 10 – Diagramme de la classe MAPPER

2.4.8.1 Attributs

N.A.

2.4.8.2 Services offerts

- `get_map() : map`
- `update_map()`

2.4.9 La classe LOCATOR

Le diagramme de la figure 11 représente la classe LOCATOR.

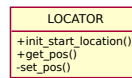


FIGURE 11 – Diagramme de la classe LOCATOR

2.4.9.1 Attributs

N.A.

2.4.9.2 Services offerts

- `init_start_location()`
- `get_pos()`
- `set_pos()`

2.4.10 La classe ROBOT

Le diagramme de la figure 12 représente la classe ROBOT.



FIGURE 12 – Diagramme de la classe ROBOT

2.4.10.1 Attributs

N.A.

2.4.10.2 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.

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	8
2	Diagramme des types de données	9
3	Diagramme de classes	10
4	Diagramme de la classe USER	10
6	Diagramme de la classe ARCHIVIST	11
8	Diagramme de la classe DRIVER	11
9	Diagramme de la classe RADAR	12
10	Diagramme de la classe MAPPER	12
11	Diagramme de la classe LOCATOR	13
12	Diagramme de la classe ROBOT	13