



SWARMBOTS

Dossier de Spécifications

Incrément 2 - Version 2.00 - Révision 18

VERSION 2023



Responsable du document : Florentin LEPELTIER

État du document : Validé

AVERTISSEMENT :

Le présent document a été réalisé dans un cadre pédagogique relatif au projet ProSE A2 de l'option Systèmes Embarqués (SE) du cursus ingénieur de l'ESEO Angers. Il a été conçu par l'équipe A2 constituée des 7 membres (Mahery FONG, Louison LEGROS, Joshua MONTREUIL, Fatoumata TRAORÉ, Julien NICOT, Matéo RONDEAU et Florentin LEPELTIER). La réalisation a été accompagnée par Thales, entreprise partenaire de l'ESEO et considérée comme client dans le cadre du projet ProSE.

Le présent document est réalisé sous la direction de Jérôme Delatour, il a été modifié et adapté par l'équipe A2 travaillant pour l'entreprise Thales dans le cadre pédagogique du projet ProSE de l'ESEO, année 2023. En dehors des activités pédagogiques de l'ESEO, ce document ne peut être diffusé ou recopié sans l'autorisation écrite de son propriétaire. Ce document est à destination de l'entreprise Thales, à qui sont cédés les droits de lecture et de modification dès lors de la livraison de la suite logicielle SWARMBOTS.

Date	Modifications apportées	Auteur	Version	Révision
30/05/2023	Validation du document pour l'incrément 2.	Florentin LEPELTIER	2.00	18
27/05/2023	Revue des descriptions IHM + modifications des CU.	Florentin LEPELTIER	1.02	17
25/05/2023	Modifications des CU, du contexte logique et du dictionnaire de domaine après AN Specifications.	Florentin LEPELTIER	1.02	16
23/05/2023	Relecture et légères modifications du dossier.	Mahery FONG	1.01	15
21/05/2023	Complément pour l'incrément 2.	Florentin LEPELTIER	1.00	14
05/05/2023	Validation du document pour l'incrément 1.	Florentin LEPELTIER	1.00	13
26/04/2023	Modifications suite aux remarques client.	Florentin LEPELTIER	0.08	12
25/04/2023	Modifications suite aux remarques client. Modifications relatives à la conception.	Florentin LEPELTIER	0.08	11
21/04/2023	Modifications suite aux remarques client.	Mahery FONG	0.07	10
23/03/2023	Relecture globale et légères corrections du dossier.	Florentin LEPELTIER	0.06	9
23/03/2023	Finalisation des modifications globales suite à AC spécification.	Joshua Montreuil	0.05	8
20/03/2023	Premier apport de modification globales suite à AC spécification.	Joshua Montreuil	0.05	7
17/03/2023	Revue globale en groupe avant AC Spécifications.	Florentin LEPELTIER	0.04	6
16/03/2023	Revue des CUs et du contexte logique en équipe.	Florentin LEPELTIER	0.04	5
16/03/2023	Revue générale après consulting.	Joshua MONTREUIL	0.03	4
15/03/2023	Revue générale avec l'équipe avant consulting.	Florentin LEPELTIER	0.02	3
13/03/2023	Modification des CUs.	Mahery FONG	0.01	2
07/03/2023	Revue générale par l'équipe après consulting.	Florentin LEPELTIER	0.00	1
01/03/2023	Création du document (fusion version éclatée).	Florentin LEPELTIER	0.00	0

Tableau 1 : Table des évolutions et validations internes du document

Table des matières

1. Introduction	5
1.1. Objet.....	5
1.2. Portée	5
1.3. Définition, acronymes et abréviations	5
1.1. Références.....	6
1.2. Vue d'ensemble	7
2. Description générale	8
2.1. Caractéristiques des acteurs.....	8
2.1.1. Acteurs directs.....	8
2.1.2. Acteurs indirects.....	8
2.2. Environnement	9
2.2.1. Architecture matérielle et logicielle	9
2.2.2. Les interfaces du système.....	10
2.2.2.1. Les interfaces logiques	11
2.2.2.2. Les interfaces avec les acteurs	11
2.2.2.2.1 En provenance de l'Utilisateur	11
2.2.2.2.2 À destination de l'Utilisateur	12
2.2.2.2.3 En provenance de Robot	13
2.2.2.2.4 À destination de Robot.....	13
2.2.2.3. Les interfaces physiques	13
2.2.2.4. Les interfaces avec les logiciels	13
2.2.2.5. Les interfaces de communication	13
2.2.3. Les contraintes de mémoires.....	13
2.2.4. Les activités	14
2.2.4.1. Fréquences d'utilisation de SWARMBOTS.....	14
2.2.4.2. Activité de maintenance	14
2.2.5. Les exigences d'adaptation	14
2.3. Fonctions principales développées	14
2.3.1. Rappel sur les cas d'usage	14
2.3.2. Rappel sur les cas d'utilisation.....	14
2.3.2.1. Représentation graphique des CUs	15
2.3.2.2. Représentation textuelle des CUs	15
2.3.3. Résumé des cas d'usage considérés pour SWARMBOTS	16
2.3.4. CU_01 Piloter un essaim de robots	17
2.3.4.1. Description graphique.....	17
2.3.4.2. Description textuelle.....	17
2.3.5. CU_02 Initialiser le système.....	19
2.3.5.1. Description graphique.....	19
2.3.5.2. Description textuelle.....	19
2.3.6. CU_03 Initialiser un nouveau robot connecté	20
2.3.6.1. Description graphique.....	20
2.3.6.2. Description textuelle.....	20
2.3.7. CU_04 Commander un robot	21
2.3.7.1. Description graphique.....	21
2.3.7.2. Description textuelle.....	21
2.3.8. CU_05 Sélectionner un robot	22
2.3.8.1. Description graphique.....	22
2.3.8.2. Description textuelle.....	22
2.3.9. CU_06 Déplacer un robot	22
2.3.9.1. Description graphique.....	22

2.3.9.2.	Description textuelle	22
2.3.10.	CU_07 Vérifier la présence d'un obstacle	23
2.3.10.1.	Description graphique	23
2.3.10.2.	Description textuelle	23
2.3.11.	CU_08 Rafraichir Ecran_Commande	24
2.3.11.1.	Description graphique	24
2.3.11.2.	Description textuelle	24
2.3.12.	CU_09 Transmettre flux caméra	25
2.3.12.1.	Description graphique	25
2.3.12.2.	Description textuelle	25
2.3.13.	CU_10 Vérifier connexion	26
2.3.13.1.	Description graphique	26
2.3.13.2.	Description textuelle	26
2.3.14.	CU_11 Consulter les logs	27
2.3.14.1.	Description graphique	27
2.3.14.2.	Description textuelle	27
2.3.15.	CU_12 Flusher les logs	28
2.3.15.1.	Description graphique	28
2.3.15.2.	Description textuelle	28
2.3.16.	CU_13 Exporter les logs	28
2.3.16.1.	Description graphique	28
2.3.16.2.	Description textuelle	28
2.3.17.	CU_14 Journaliser un évènement	29
2.3.17.1.	Description graphique	29
2.3.17.2.	Description textuelle	29
2.3.18.	CU_15 Quitter SB_IHM	29
2.3.18.1.	Description graphique	29
2.3.18.2.	Description textuelle	29
2.4.	Contraintes	30
2.4.1.	Politiques réglementaires	30
2.4.2.	Contraintes matérielles	30
2.4.3.	Exigences de fiabilité	30
2.4.4.	Exigences de maintenabilité	30
2.4.5.	Exigences de disponibilité	30
2.5.	Hypothèses et dépendances	31
2.6.	Répartition des exigences	31
3.	Exigences spécifiques	31
3.1.	Interfaces Homme Machine	31
3.1.1.	Généralités	31
3.1.2.	Les actions utilisateur	31
3.1.3.	Les écrans	31
3.1.3.1.	Vue générale	31
3.1.3.2.	Ecran_Accueil	33
3.1.3.3.	Ecran_Commande	34
3.1.3.4.	Ecran_Logs	37
3.1.3.5.	Popup_Erreur	38
3.1.3.6.	Popup_Flush	39
3.2.	Description des fonctions	40
3.2.1.	Récupérer Mode_Fonctionnement	40
3.2.1.1.	Description graphique	40
3.2.1.2.	Description textuelle	40
3.3	Dictionnaire du domaine	41
3.3.1	Définitions	41

1. Introduction

1.1. Objet

Ce dossier de spécifications a pour objectif de définir les fonctionnalités et exigences attendues par la société Thales Services Numériques (désignée dans la suite du document comme le client) pour le développement logiciel d'un démonstrateur d'un essaim de robots : SWARMBOTS. Dans le contexte de développement d'un prototype, ce dossier de spécifications se focalise sur une simple étape du cycle de vie du produit, à savoir l'étape d'utilisation du démonstrateur par un utilisateur, en n'adressant que les fonctionnalités principales attendues pour ce prototype.

Ce document permettra à l'équipe de concevoir, développer et tester les parties informatiques du prototype.

Les fonctionnalités et exigences présentées dans ce document ont été déterminées suite à l'étude du Cahier des charges [CdC_Thales_ProSE_2023] et des rencontres et échanges avec un représentant de la société Thales Services Numériques (voir compte-rendus des réunions [[ProSE]_A2_CR_01_2024], [[ProSE]_A2_CR_09_2024])

Ce document constitue l'une des pièces du contrat signé avec le client et par extension la société Thales Services Numériques, dans le cadre du projet ProSE de l'ESEO.

Ce dossier de spécification s'inspire de la norme [IEEE/ISO/IEC 29148-2018]. Il utilise des schémas et illustrations respectant la norme UML en version 2.5.1 [UML 2.5.1_2017].

1.2. Portée

Ce document décrit les fonctionnalités et exigences du Système à l'Étude (SàE) constitué :

- Du logiciel **SB_IHM**, permettant à un utilisateur de piloter et de commander un ou plusieurs robots de l'essaim.
- Du logiciel **SB_C** embarqué sur les robots permettant le contrôle de leurs périphériques associés.

1.3. Définition, acronymes et abréviations

Les abréviations utilisées dans le présent document sont répertoriées et expliquées dans le tableau présenté ci-dessous. Les termes utiles pour interpréter correctement ce dossier de spécifications sont définis dans le dictionnaire de domaine présent dans ce dossier dans la partie 3.3.

Acronymes, abréviations	Définition
Client	Société Thales Services Numériques
CSI (Camera Serial Interface)	Connecteur utilisé et implémenté sur les raspberry pi pour s'interfacer avec des caméras utilisant la même connectique.
CU	Cas d'Utilisation
ENT (Environnement Numérique de Travail)	Environnement de travail utilisé par l'équipe dans le cadre du projet ProSE afin d'y centraliser les documents du projet (outil Redmine).
GPIO (General Purpose Input Output)	Terme désignant les broches d'entrées / sorties de la Raspberry Pi.
ID	Identifiant.
IEEE (Institute of Electrical and Electronics Engineers)	Association professionnelle internationale définissant entre autres des normes dans le domaine informatique et électronique.

IHM (Interface Homme Machine)	Moyens permettant aux utilisateurs de SWARMBOTS d'interagir avec SWARMBOTS.
IP (Internet Protocol)	Famille de protocoles de communication de réseaux informatiques conçus pour être utilisés sur internet.
IR (Infra Rouge)	Infrarouge, rayonnement électromagnétique.
LED (Light-Emitting Diode)	Diode électroluminescente.
MAC (Media Access Control)	Adresse/identifiant physique stocké dans une carte réseau ou une interface réseau.
N.A	Non Applicable
PWM (Pulse Width Modulation)	Modulation de largeur d'impulsions, génération de signaux pseudo analogique à partir d'un environnement numérique.
RGB (Red Green Blue)	Rouge Vert Bleu, espace de couleur défini par la Commission internationale de l'éclairage, permet de quantifier la couleur.
SàE (Système à l'étude)	Il s'agit de l'ensemble des logiciels SWARMBOTS_CONTROLEUR et SB_IHM .
SAV	Service Après-Vente
SB	SwarmBots
SB_C	SwarmBots_Controleur, contrôleur utilisé pour les robots du SàE.
SB_IHM	SwarmBots_IHM. Interface Homme-Machine utilisée pour le SàE.
UDP (User Datagram Protocol)	Protocole de communication utilisé sur internet pour les transmissions sensibles au temps (vidéo etc). Il fournit une communication rapide car il n'y a pas d'établissement formel de connexion avant le transfert de données.
UML (Unified Modeling Language)	Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel.
WIFI (Wireless Fidelity)	Technologie de connexion sans fil.

1.1. Références

Voici un tableau récapitulatif des documents utilisés pour le dossier de spécifications ainsi que les liens permettant d'accéder aux fichiers :

Références	Description
[[ProSE]_A2_CR_01_2024]	ROBOMATIC, « compte rendu de réunion avec M. RAJAOARISOA, M. ROCOURT, M.STAMBOULI en date du 09/02/2023. » ENT rubrique gestion de projet – Equipe ProSE A2, 2023.
[[ProSE]_A2_CR_09_2024]	ROBOMATIC, « compte rendu de réunion avec M. RAJAOARISOA, M. ROCOURT, M.STAMBOULI en date du 09/03/2023. » ENT rubrique gestion de projet – Equipe ProSE A2, 2023.
[[ProSE]_fonct_client]	Document de spécification des fonctionnalités développées au cours du projet. Disponible sur l'ENT rubrique gestion de projet - M.LEPELTIER, 2023.
[Alphabot2-user-manual-en_2023]	Documentation technique de l'alphabot 2 disponible sur le site http://wvashare.com (Alphabot2-user-manual-en.pdf) et sur l'ENT rubrique specification/doc - Wvashare, juillet 2019.
[GA12-N20_2023]	Documentation technique des moteurs du robot. Disponible sur l'ENT rubrique

	specification/doc - Handson Technology, 2023.
[IEE Std 802.11b-1999_2023]	Standard pour les communications WIFI. Disponible sur l'ENT rubrique specification/doc - IEE, 2023.
[raspberrypi-3-b-plus-product-brief_2023]	Brève documentation technique de la raspberry 3B+. Détaille les spécifications techniques générales de la carte. Disponible sur l'ENT rubrique specification/doc, raspberrypi.org, 2023.
[RPi_Camera_(B)_2023]	Caméra adaptée à la raspberry de waveshare. Disponible sur l'ENT rubrique specification/doc - raspberrypi.com, 2023.
[ST188-EN-2_2023]	Documentation technique du capteur IR. Disponible sur l'ENT rubrique specification/doc - npec.com, 2023.
[SG90_2023]	Documentation technique des servo motor. Disponible sur l'ENT rubrique specification/doc - luxoparts, 2023.
[SMA13_2023]	Documentation technique du buzzer. Disponible sur l'ENT rubrique specification/doc - cnmic.com, 2023.
[UML 2.5.1_2017].	OMG, "Unified Modeling Language", version 2.5.1, https://www.omg.org/spec/UML/2.5.1/ , 2017.
[WS2812B_2023]	Documentation technique des LEDs adressables RGB. Disponible sur l'ENT rubrique specification/doc - worldSemi, 2023.

1.2. Vue d'ensemble

Ce document de spécification est structuré en 3 parties :

- La partie I présente les objectifs et la portée de ce document.
- La partie II, intitulée « description générale », a pour objectif de présenter l'environnement et le contexte de SWARMBOTS, ainsi que les fonctionnalités principales attendues du SàE.
- La partie III présente en détail les IHM attendues, les fonctionnalités détaillées de l'IHM ainsi que le dictionnaire du domaine.

2. Description générale

Par l'incapacité d'exposer simplement ses domaines d'activités à son public externe sans compromettre sa confidentialité, le département EE4 de TSN souhaite mettre en place un démonstrateur constitué d'un essaim de robots pilotable via une interface homme machine simple d'utilisation.

L'équipe ProSE A2 de l'ESEO est en charge du développement logiciel de SWARMBOTS, destiné à mettre en place le démonstrateur souhaité par le client. Cette solution logicielle doit permettre l'interaction entre les robots de l'essaim du démonstrateur et un utilisateur.

Les robots du démonstrateur en question seront pilotés par le logiciel SWARMBOTS constitué de deux entités :

- Une IHM permettant l'interaction avec un utilisateur (envoi et réception de commandes)
- Un contrôleur embarqué sur les robots permettant de contrôler leurs périphériques

Le présent document spécifie l'incrément 2 du prototype SWARMBOTS. Il ne traite que d'une utilisation simplifiée de SWARMBOTS (ciblant les fonctionnalités principales de son prototype). ***Dans le cadre de cette étude, l'ensemble du système est spécifié, néanmoins, la réalisation est divisée en deux incréments.*** Le fichier du cahier des charges client indique les fonctionnalités prévues pour l'incrément 1 et l'incrément 2[[ProSE]_fonct_client]. L'incrément 1 simplifie la réalisation à l'existence d'un seul robot. ***Dans cet incrément 2, la gestion de deux robots est réalisée.***

2.1. Caractéristiques des acteurs

Par le terme d'acteur, nous désignons tout rôle joué par une entité (morale ou physique) qui interagit directement ou non avec le SàE. Cette entité peut être une personne (généralement un utilisateur du système) ou un autre système.

Nous distinguons les acteurs, dits directs (qui interagissent directement avec le SàE) et les acteurs dits indirects (qui n'ont pas d'interaction directe avec le SàE) mais qui sont à l'origine d'exigences à respecter par le SàE.

2.1.1. Acteurs directs

Les acteurs directs sont :

- **Utilisateur** : Utilisateur principal de SWARMBOTS. Il est destiné à saisir des commandes via l'IHM de SWARMBOTS.
- **Robot** : Cible physique disposant de périphériques pilotables par SWARMBOTS.

2.1.2. Acteurs indirects

Les acteurs indirects sont :

N.A.

2.2. Environnement

2.2.1. Architecture matérielle et logicielle

Le diagramme de déploiement UML de la figure 1 représente l'architecture logicielle et matérielle du SàE. Les conventions graphiques utilisées sont explicitées en figure 2.

Ce diagramme de déploiement identifie les entités matérielles et/ou logicielles avec lesquelles le SàE (composé des entités **SB_IHM** et **SB_C**) doit interagir et permet ainsi de déterminer les principaux échanges qu'il entretient avec son environnement.

ProSE A2 - Architecture matérielle et logicielle de SWARMBOTS

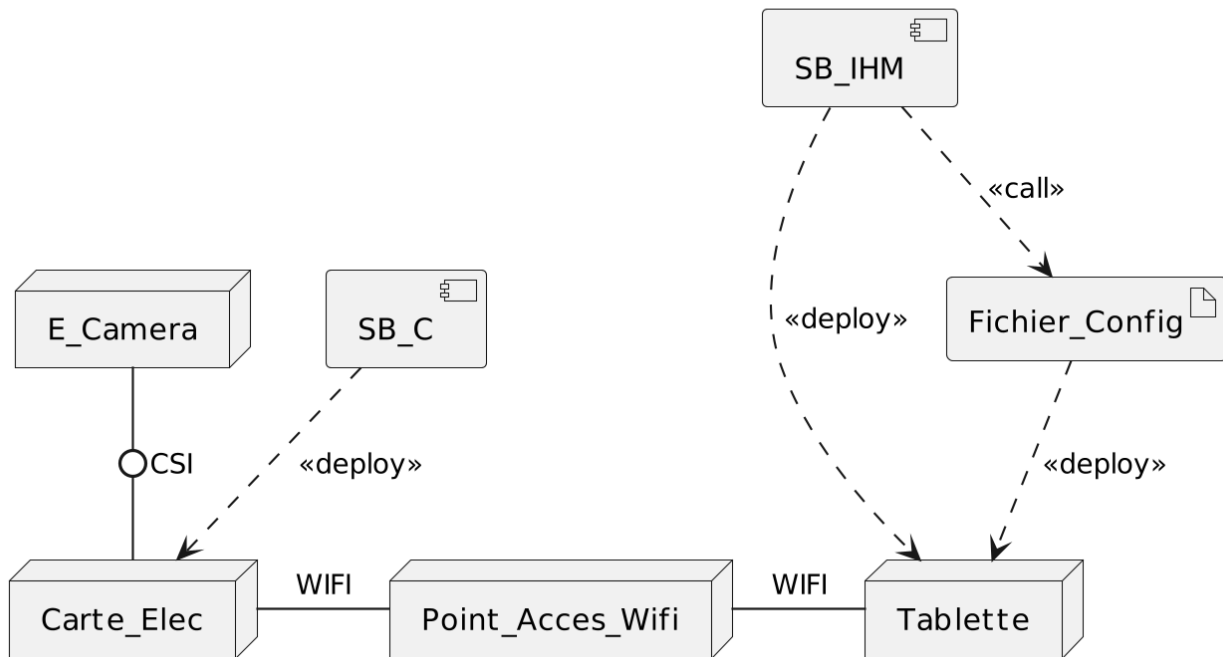


Illustration 1: Architecture matérielle et logicielle représentée par un diagramme de déploiement UML

Comme indiqué sur l'illustration 1, **SB_C** est en interaction avec différentes entités externes au SàE. Par convention, le nom de ces entités est préfixé par les lettres «E_» (E pour Externe), elles sont aussi désignées, dans ce document, par le terme de «périphérique SB».



Illustration 2 : Légende du diagramme de déploiement UML

Le SàE, qui comprend **SB_IHM** et **SB_C**, sera respectivement déployé sur les dispositifs matériels *Tablette* et *Carte_Elec*.

En fonction des ordres donnés par l'utilisateur, **SB_C** commandera *Carte_Elec* ainsi que ses périphériques respectifs, fournis par le client. Ces ordres provenant de l'utilisateur sont envoyés par l'intermédiaire de l'IHM.

SB_C sera développé en Langage C et exécuté sur la cible physique *Carte_Elec*.

SB_IHM sera développé en Java sous la forme d'une application Android exécutée sur la cible physique *Tablette*.

Fichier_Config est un fichier de configuration présent sur la cible physique *Tablette*, son format sera défini en conception, néanmoins, la description de son contenu est détaillée au chapitre 3, dans le dictionnaire de domaine.

Tablette dispose de la version Android 6.0 (Marshmallow) afin de pouvoir exécuter **SB_IHM**. Dans le cadre de l'exécution de **SB_IHM**, une résolution de 1920x1200 est recommandée. Le mode paysage sera utilisé dans le cadre de cette étude. Le **multi-touch** ne sera pas géré par le SàE sur *Tablette*, l'utilisateur ne pourra exercer une pression qu'à un seul endroit à la fois sur l'écran de *Tablette*.

Carte_Elec est une Raspberry Pi 3B+ [[raspberry-pi-3-b-plus-product-brief 2023](#)] disposant de l'OS Raspbian GNU/Linux, de version 10 (buster). Ce dispositif matériel intègre plusieurs périphériques fournis et imposés par le client. *E_Camera* [[RPi_Camera \(B\) 2023](#)] est le seul périphérique à figurer sur l'architecture matérielle car il est le seul à être directement connecté à *Carte_Elec*. Les autres périphériques sont connectés à la *Carte_Elec* au travers d'une carte électronique intermédiaire, correspondant à l'Alphabot2 dont la documentation est fournie [[Alphabot2-user-manual-en 2023](#)]. L'Alphabot2 n'est qu'une interface entre *Carte_Elec* et ses périphériques permettant de les disposer de manière intelligente afin de former un robot.

Ces derniers sont listés ci-dessous :

- Moteur_Gauche : Il s'agit d'un moteur à courant continu pilotable en PWM. [[GA12-N20 2023](#)]
- Moteur_Droit : Il s'agit d'un moteur à courant continu pilotable en PWM. [[GA12-N20 2023](#)]
- Power_Btn : Il s'agit d'un bouton poussoir permettant de mettre *Robot* sous tension.
- Buzzer : Il s'agit d'un buzzer [[SMA13 2023](#)] permettant d'émettre un son aigu.
- Servo : Il s'agit d'un servomoteur pilotable en PWM [[SG90 2023](#)] permettant de positionner *E_Camera*.
- IR : Il s'agit d'un capteur infrarouge [[ST188-EN-2 2023](#)] permettant de détecter des obstacles.
- LEDs_RGB : Il s'agit d'un ruban de LEDs RGB [[WS2812B 2023](#)] permettant d'indiquer des informations relatives au SàE.

La communication entre *Tablette* et *Carte_Elec* se fait par wifi. Cette connexion wifi respecte la norme wifi IEE 802.11b [[IEEE Std 802.11b-1999 2023](#)]. *Tablette* et *Carte_Elec* seront respectivement connectés au réseau wifi existant généré par **Point_Acces_Wifi**. L'adressage IP des cibles physiques respectives sera statique sur ce réseau wifi de façon à ce que ces adresses IP statiques soient renseignées dans **Fichier_Config** présent sur *Tablette*. La configuration du **Point_Acces_Wifi** sera prise en charge par l'équipe ProSE A2 lors des livraisons des incréments.

Bien que plusieurs Carte_Elec puissent se connecter à SB_IHM, la connexion à une seule Carte_Elec sera gérée afin de simplifier la réalisation dans l'incrément 1. Dans l'incrément 2, la connexion à deux Carte_Elec sera gérée par SB_IHM.

2.2.2. Les interfaces du système

Ce chapitre décrit les entrées et sorties dites «logiques» et «physiques» du SàE. En effet, nous différencions dans cette étude deux grands types d'entrées/sorties :

- Celles dites de haut niveau (dites aussi logiques) qui décrivent les événements et données échangées entre l'utilisateur et le SàE. Ces entrées et sorties portent sur les intentions des acteurs interagissant avec le SàE.
- Celles dites de bas-niveau (dites aussi physiques) qui sont les entrées/sorties réellement échangées entre le SàE et les périphériques du SàE. Les entrées/sorties physiques (ou bas niveau).

2.2.2.1. Les interfaces logiques

L'illustration 3 (page 11) présente le contexte de SWARMBOTS en faisant figurer les entrées/sorties dites de haut niveau (ou logiques). Elles sont regroupées en grande famille. Pour représenter ce contexte logique, un diagramme de communication UML a été élaboré. Nous retrouvons les périphériques du SàE déjà présentés en illustration 1 (page 9).

ProSE A2 - Contexte logique de SWARMBOTS

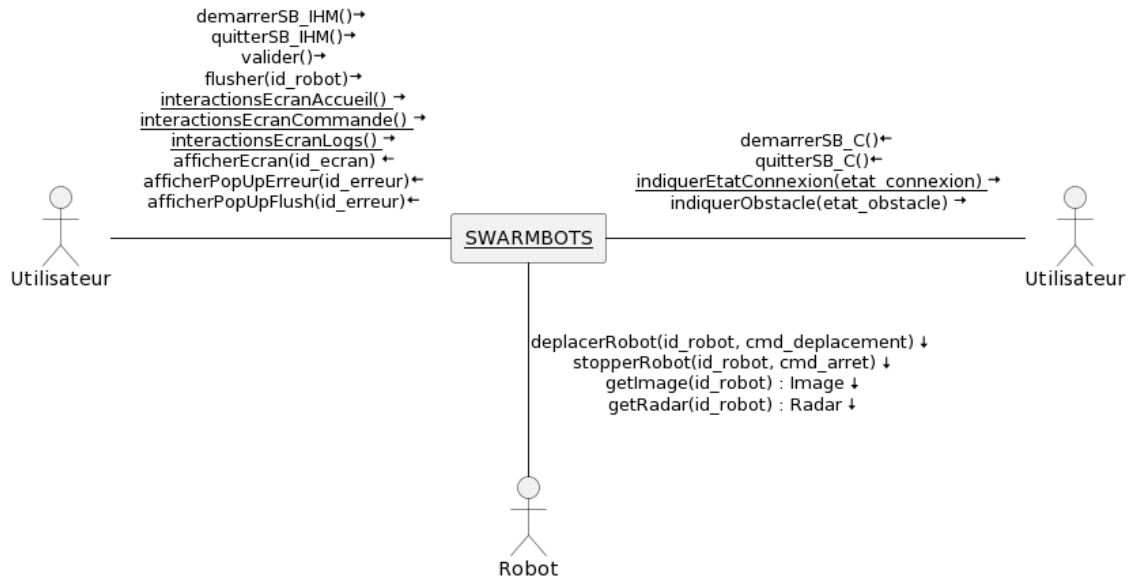


Illustration 3: Contexte logique de SWARMBOTS, représenté par un diagramme de communication UML

Dans ce diagramme de communication, seules les entrées/sorties logiques entre les acteurs et le SàE sont présentées.

2.2.2.2. Les interfaces avec les acteurs

Nous allons maintenant détailler les interactions entre l'Utilisateur, le Robot et SWARMBOTS sous forme d'entrées et sorties logiques.

2.2.2.2.1 En provenance de l'Utilisateur

2.2.2.2.1.1 Vers SWARMBOTS

Nous décomposerons cette description en deux parties, l'une décrivant les interactions entre l'utilisateur et SWARMBOTS, axée **SB_IHM**, et l'autre décrivant les interactions entre l'utilisateur et SWARMBOTS, axée **SB_C**.

Voici les événements logiques de l'Utilisateur vers SWARMBOTS (axés **SB_IHM**) :

- **demarrerSB_IHM()** : Utilisateur exécute le logiciel **SB_IHM** sur *Tablette*
- **quitterSB_IHM()** : Utilisateur quitte le logiciel **SB_IHM** sur *Tablette*
- **interactionsEcranAccueil()** :
 - **commanderRobot()** : Utilisateur demande à afficher **Ecran_Commande**.
- **interactionsEcranCommande()** :
 - **selectionnerRobot(id_robot)** : Utilisateur sélectionne un robot pour le commander.
 - **deplacerRobot(id_robot, cmd_deplacement)** : Utilisateur déplace le robot sélectionné.
 - **stopperRobot(id_robot, cmd_arret)** : Utilisateur stoppe le robot sélectionné.
 - **modifierModeFonctionnement(id_robot, Mode_Fonctionnement)** : Utilisateur modifie le mode de fonctionnement du robot sélectionné.
 - **retournerEcranAccueil()** : Utilisateur demande à afficher **Ecran_Accueil** de **SB**.
 - **consulterLogs()** : Utilisateur demande à afficher **Ecran_Logs** de **SB**.

- interactionsEcranLogs() :
 - selectionnerSourceLogs(*id_file*) : Utilisateur demande à afficher les logs d'un robot.
 - exporterLogs(*id_file*) : Utilisateur demande à exporter les **Source_Logs**.
 - retournerEcranCommande() : Utilisateur demande à afficher l'écran de commande de **SB**.
- valider() : Utilisateur valide un message d'avertissement, d'erreur ou d'information, entraînant la fermeture de la **popUp_Erreur**.
- flusher(*id_robot*) : Utilisateur demande la récupération des logs d'un certain robot.

Voici les événements logiques de Utilisateur vers SWARMBOTS (axés **SB_C**) :

- demarrerSB_C () : Utilisateur démarre le logiciel **SB_C**. Le démarrage s'effectue en ligne de commande via une connexion SSH à Carte_Elec.
- quitterSB_C () : Utilisateur quitte le logiciel **SB_C**. L'arrêt s'effectue en ligne de commande via une connexion SSH à Carte_Elec.

La description de l'ensemble des paramètres utilisés dans ces fonctions logiques se trouve dans le dictionnaire de domaine à la fin de ce document (chapitre 3.3, page 32).

2.2.2.2.2 À destination de l'Utilisateur

2.2.2.2.2.1 Depuis SWARMBOTS

Voici les événements logiques de SWARMBOTS vers Utilisateur (axés **SB_IHM**) :

- afficherEcran(*id_ecran*) :
 - Lorsqu'*id_ecran* vaut ECRAN_ACCUEIL :
 - afficherEcranAccueil() : **SB** affiche **Ecran_Accueil**.
 - majEcranAccueil() : **SB** met à jour **Ecran_Accueil**. Cette fonction permet d'afficher la liste des robots et leurs états de connexion respectifs. Elle est appelée lorsque SB a terminé son initialisation. Elle fait le lien entre les deux états possibles de l'Ecran_Accueil. Les illustrations 12 et 13 représentent ces deux états possibles : avant et après initialisation. Ils sont décrits en section 3.1.3.2 (IHM).
 - Lorsqu'*id_ecran* vaut ECRAN_COMMANDE :
 - afficherEcranCommande() : **SB** affiche **Ecran_Commande**.
 - majEcranCommande() : **SB** met à jour **Ecran_Commande**. Cette fonction permet de mettre à jour les différents éléments présents sur l'écran **Ecran_Commande**, notamment les flux vidéo et radar, l'état des périphériques et l'état de connexion des robots (voir l'IHM).
 - indiquerRobotDeconnecte(*id_robot*) : **SB** indique la déconnexion d'un robot à Utilisateur.
 - indiquerRobotDeselectionne(*id_robot*) : **SB** indique la désélection d'un robot à Utilisateur.
 - indiquerRobotSelectionne(*id_robot*) : **SB** indique la sélection d'un robot à Utilisateur.
 - indiquerCommande(*cmd*) : **SB** indique la commande de déplacement demandé par Utilisateur.
 - indiquerDesactivationFlux(*id_flux*) : indique à l'utilisateur que le flux concerné est désactivé.
 - indiquerObstacle() : **SB** informe qu'un obstacle a été détecté à Utilisateur.
 - majFluxRadar() : **SB** met à jour le flux radar. Cette action est déclenchée tous les **Temps_Rafraichissement** si **Mode_Fonctionnement** radar est activé.
 - majFluxCamera() : **SB** met à jour le flux caméra. Cette action est déclenchée tous les **Temps_Rafraichissement** si **Mode_Fonctionnement** caméra est activé.
 - autoriserCommandeRobot(*id_robot*) : **SB** autorise Utilisateur à déplacer le robot. Cette action est déclenchée à chaque sélection d'un robot. La commande de direction est affichée si aucun robot n'était précédemment sélectionné.
 - Lorsqu'*id_ecran* vaut ECRAN_LOGS :
 - afficherEcranLogs() : **SB** affiche **Ecran_Logs**. Cette action est déclenchée au premier affichage de l'écran.

- majEcranLogs() : **SB** met à jour Ecran_Logs. Cette action prend en compte les modifications des éléments de l'écran pour les afficher.
- afficherLogsSourceSelectionne(**id_file**) : **SB** affiche les **Source_Logs** par Utilisateur.
- afficherPopUpErreur(**id_erreur**) : **SB** affiche une **PopUp** en superposition de l'écran affiché mentionnant un message expliquant un avertissement ou une erreur à l'Utilisateur (voir le dictionnaire de domaine pour la liste détaillée des erreurs et avertissements).
- afficherPopUpFlush(**id_erreur**) : **SB** affiche une **PopUp** en superposition de l'écran affiché.

Voici les événements logiques de SWARMBOTS vers Utilisateur (axés **SB_C**) :

- indiquerEtatConnexion(**etat_connexion**) : **SB_C** informe Utilisateur de l'état de connexion de Robot.
- indiquerObstacle(**etat_obstacle**) : **SB_C** informe Utilisateur qu'un obstacle a été détecté.

2.2.2.2.3 En provenance de Robot

2.2.2.2.3.1 Vers SWARMBOTS

N.A.

2.2.2.2.4 À destination de Robot

2.2.2.2.4.1 Depuis SWARMBOTS

Voici les événements logiques de SWARMBOTS vers Robot :

- deplacerRobot(**id_robot**, **cmd_deplacement**) : **SB** déplace Robot.
- stopperRobot(**id_robot**, **cmd_arret**) : **SB** stoppe Robot.
- getImage(**id_robot**) : Image : **SB** récupère le flux vidéo de Robot.
- getRadar(**id_robot**) : Radar : **SB** récupère le flux radar de Robot.

2.2.2.3. Les interfaces physiques

N.A.

2.2.2.4. Les interfaces avec les logiciels

Il n'est pas prévu que le SàE communique avec d'autres logiciels autres que ceux prévus pour les activités de développement de **SB_IHM** et **SB_C**.

2.2.2.5. Les interfaces de communication

Dans ce paragraphe sont définies les interfaces de communication (et les protocoles de communication associés). Dans cette étude, la communication effectuée entre le **SB_C** et **SB_IHM** (comme illustré au chapitre 2.2.1 - Illustration 1) s'effectue par wifi. Le signal wifi exploite les bandes de fréquence proches des 2,4GHz. Aucun autre protocole n'intervient dans le cadre de cette étude.

2.2.3. Les contraintes de mémoires

Dans le cadre de leur exécution, **SB_C** et **SB_IHM** généreront des logs qui seront stockés sur leur dispositifs matériel respectifs (*Carte_Elec* et *Tablette*). Une contrainte mémoire peut apparaître sur *Carte_Elec* s'il n'y a plus assez d'espace mémoire pour stocker les logs (**Mémoire_Min**), dans ce cas, une **popUp_Flush** sera affichée sur l'IHM afin d'avertir l'utilisateur et de lui permettre de flush. Cette popUp est décrite dans la section 3.1.3.1, vue générale.

*Seule la génération des logs en console lors de l'exécution du programme sera gérée à l'incrément 1. Dans l'incrément 2, le stockage des logs respectifs et le flush des logs de **SB_C** vers **SB_IHM** sera traité.*

2.2.4. Les activités

2.2.4.1. Fréquences d'utilisation de SWARMBOTS

N.A.

2.2.4.2. Activité de maintenance

N.A.

2.2.5. Les exigences d'adaptation

N.A.

2.3. Fonctions principales développées

Ce chapitre présente les fonctionnalités principales développées dans l'incrément 1 en utilisant une démarche par cas d'usage et par Cas d'Utilisation (CU).

2.3.1. Rappel sur les cas d'usage

Les cas d'usage recensent les étapes essentielles du cycle de vie d'un produit depuis la fabrication du produit jusqu'à l'élimination ou recyclage de ce produit.

À chaque étape du cycle de vie correspond un cas d'usage (si cette étape induit des fonctionnalités à définir pour le produit considéré). Pour chaque cas d'usage, plusieurs cas d'utilisations distincts peuvent être définis.

Parmi les cas d'usage, on retrouve généralement ceux de fabrication du produit (comprenant ou non les activités de test du produit fabriqué), de conditionnement (paramétrage éventuel, expédition et transport), de commercialisation (paramétrage éventuel, mode de démonstration, installation sur site, etc.), d'utilisation (par le ou les utilisateurs), de maintenance (SAV ou diagnostique), de désinstallation et de recyclage (élimination ou valorisation).

2.3.2. Rappel sur les cas d'utilisation

Un Cas d'Utilisation (CU) représente un ensemble d'interactions entre un ou des acteurs et le système à spécifier. Ce cas d'utilisation étant souvent lié à un ou parfois plusieurs cas d'usage. Un CU est principalement décrit par un scénario d'utilisation (nommé scénario nominal), scénario d'une utilisation représentative du système. Ces CU sont ensuite détaillés jusqu'à un niveau de décomposition suffisant pour décrire les fonctions attendues du système.

2.3.2.1. Représentation graphique des CUs

Les CU peuvent être représentés sous forme graphique, voir l'illustration 4 ci-dessous pour une illustration. Les acteurs directs sont représentés sous forme de petits personnages. Dans les bulles sont représentés les cas d'utilisation. Un trait entre un acteur et un CU indique que l'acteur participe à ce CU. Les liens hachurés entre CU, préfixés par le mot <<use>> (ou <<include>>), indique que ce CU fait appel à l'autre CU – on parle alors de sous- cas d'utilisation. Les liens hachurés entre CU, préfixés par le mot <<extend>>, indique qu'il s'agit d'une extension d'un CU : un CU qui ne se déclenche que sous certaines conditions.

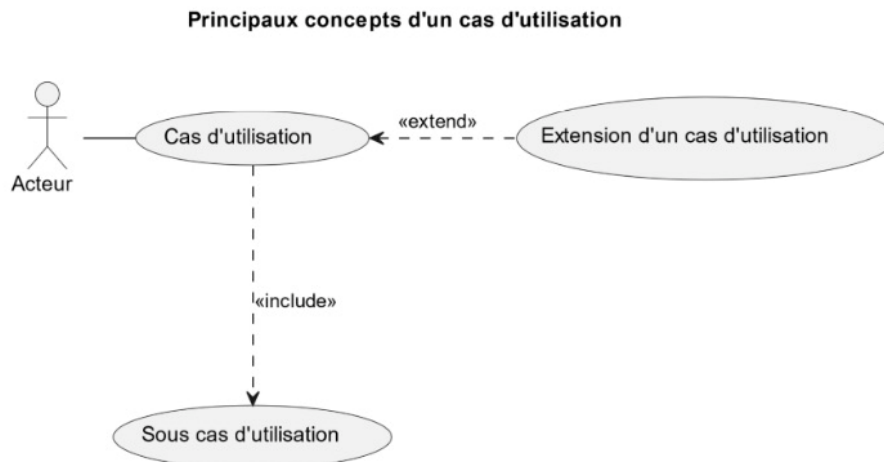


Illustration 4 : Légende explicative d'un diagramme de cas d'utilisation UML

2.3.2.2. Représentation textuelle des CUs

La description textuelle des cas d'utilisation est souvent présentée sous forme d'un tableau, constitué des champs suivants :

Titre	Rappelle en quelques mots l'objectif principal du cas d'utilisation.
Résumé	Décrit brièvement le comportement du cas d'utilisation.
Portée	Définis la portée de conception du CU (étendue spatiale).
Niveau	Niveau de granularité du cas d'utilisation (Stratégique, utilisateur ou sous-fonction).
Acteurs directs	Acteur qui participent au CU.
Acteurs indirects	Acteurs qui ne participent au CU, mais qui ont un intérêt dans sa réalisation
Préconditions	Ensemble des conditions qui doivent être vérifiées avant le déroulement du CU. Les préconditions, sans mention contraire explicite, des CU parents au CU doivent toujours être vérifiées.
Garanties minimales	Définis ce qui est garanti par le système à l'étude même en cas d'échec du cas d'utilisation.
Garanties en cas de succès	Définis les garanties en cas de succès (par le scénario nominal ou par ses variantes).
Scénario nominal	C'est un scénario représentatif de l'utilisation du système où tout se passe bien. Il se termine par la réussite des objectifs. Il peut être constitué d'une condition déclenchant le scénario, d'un ensemble d'étape, d'une condition de fin, et éventuellement d'extensions ou de variantes. Une étape peut être une interaction entre acteurs, une étape de validation, ou un changement interne.
Variantes	Lorsqu'il y a plusieurs façons de procéder à une même étape sans remise en cause du scénario nominal. Le caractère * peut être utilisé dans cette section pour exprimer un parallélisme. Cela interrompt l'exécution du CU en cours afin d'exécuter la variante. une fois la variante exécutée, le fil d'exécution reprend où il a précédemment été interrompu.
Extensions	Définissent les autres scénarios que le scénario nominal (par exemple ceux qui se terminent par un échec). Ils se déclenchent sur des conditions spécifiques détectées par le SàE.
Informations complémentaires	Informations diverses nécessaires à la compréhension du CU

2.3.3. Résumé des cas d'usage considérés pour SWARMBOTS

Dans le cadre de ce document de spécification, le seul cas d'usage considéré pour le SàE est le l'utilisation nominale par l'Utilisateur.

L'incrément décrit dans ce dossier de spécification concerne le développement d'un prototype, par conséquent, les autres cas d'usage comme celui de la fabrication, du recyclage ou de la maintenance de second niveau ne sont pas pris en compte. Concernant le cas d'usage de la maintenance du SàE, les activités classiques de développement et de débogage sont par conséquent ignorées.

2.3.4. CU_01 Piloter un essaim de robots

2.3.4.1. Description graphique

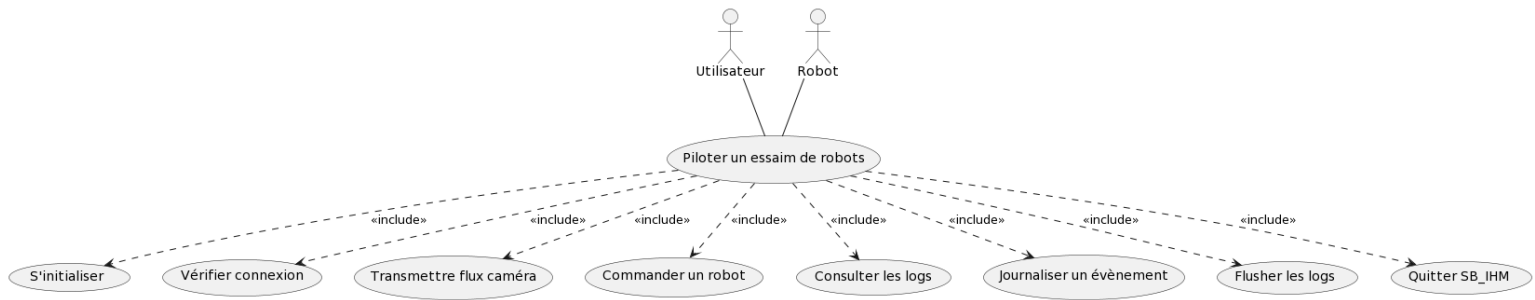


Illustration 5 : CU "Piloter un essaim de robots"

2.3.4.2. Description textuelle

Titre	Piloter un essaim de robots
Résumé	Utilisateur pilote un essaim de robots
Portée	SB
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun
Niveau	Stratégique
Préconditions	<p>SB_IHM est présent sur <i>Tablette</i> et est en état de fonctionner.</p> <p>SB_C est présent sur <i>Carte_Elec</i> et est en état de fonctionner.</p> <p>Fichier_Config est valide et présent sur <i>Tablette</i>.</p> <p><i>Carte_Elec</i> est connectée à Point_Acces_Wifi.</p> <p><i>Tablette</i> est connectée à Point_Acces_Wifi.</p> <p>Utilisateur est connecté à <i>Carte_Elec</i> en SSH.</p>
Garanties minimales	Aucune.
Garanties en cas de succès	Confère sous-CU.
Scénario nominal	<ol style="list-style-type: none"> Utilisateur démarre SB_C. SB_C indique qu'il est en attente de connexion à Utilisateur. Utilisateur démarre SB_IHM. SB_IHM affiche Ecran_Accueil. SB s'initialise. Utilisateur demande à commander robot. SB_IHM récupère Mode_Fonctionnement. SB_IHM affiche Ecran_Commande. Utilisateur <u>commande un robot</u>. Utilisateur <u>quitte SB_IHM</u>.
Variantes	<p>3-10 :</p> <p>3-10.a.1. Utilisateur démarre un nouveau SB_C.</p> <p>3-10.a.2. SB_C indique qu'il est en attente de connexion à Utilisateur.</p> <p>3-10.a.3. Retour en *.</p> <p>4-9 : [Utilisateur demande à quitter SB_IHM]</p> <p>4-9.a.1. Va en 10.</p> <p>6-9 : [Tous les Temps_Rafraichissement]</p> <p>6.9.a.1. SB_C <u>transmet son flux caméra</u> à SB_IHM.</p> <p>6.9.a.2. Retour en *.</p> <p>6-9 : [Tous les Temps_Envoi_Ping]</p> <p>6-9.b.1. SB <u>vérifie la connexion</u>.</p>

	<p>6-9.b.2. Retour en *.</p> <p>9 : [Utilisateur demande à consulter les logs] 9.a.1. Utilisateur <u>consulte les logs</u>. 9.a.2. Va en 8.</p> <p>9 : [Utilisateur demande à retourner à Ecran_Accueil] 9.b.1. Va en 4.</p> <p>* : [Un evenement_loggable_SB_IHM intervient sur SB_IHM] *.a.1. SB_IHM enregistre et date le nouveau log dans Fichier_Logs_SB_IHM. *.a.2. Retour en *.</p> <p>* : [Un evenement_loggable_SB_C intervient sur un des SB_C connectés] *.b.1. SB_C concerné <u>journalise l'évènement</u>. *.b.2. Retour en *.</p>
Extension	<p>5 : [ERREUR_AUCUN_ROBOT_RENSEIGNE] 5.a.1. SB_IHM met à jour le message d'accueil. 5.a.2. SB_IHM met à jour Ecran_Accueil. 5.a.3. Retour en *.</p> <p>6-9 : [La var_connexion d'un SB_C atteint Limite_Connexions_Manquees] 6-9.c.1. SB_C indique qu'il est en attente de connexion à Utilisateur. 6-9.c.2. SB_C execute le déplacement cmd_arret. 6-9.c.3. Retour en *.</p> <p>6-9 : [La var_connexion d'un robot de Liste_Robots de SB_IHM atteint Limite_Connexions_Manquees] 6-9.d.1. SB_IHM lève ERREUR_CONNEXION_ROBOT. 6-9.d.2. Retour en *.</p> <p>6-9 : [ERREUR_ESPACE_MEMOIRE] 6-9.e.2. SB_IHM affiche PopUp_Flush. 6-9.e.3. Utilisateur <u>flush les logs</u>. 6-9.e.3. Retour en *.</p> <p>6-9 : [ERREUR_CONNEXION_ROBOT] 6-9.f.1. SB_IHM met à jour l'état de connexion du robot dans Liste_Robots. 6-9.f.2. SB_IHM signale la déconnexion du robot à Utilisateur. 6-9.f.3. SB_IHM affiche PopUp_Erreur. 6-9.f.4. Utilisateur valide. 6-9.f.5. Retour en *.</p>

2.3.5. CU_02 Initialiser le système

2.3.5.1. Description graphique

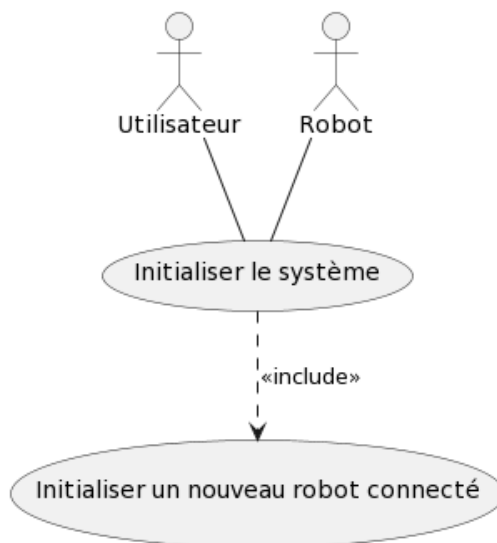


Illustration 6 : CU « Initialiser le système »

2.3.5.2. Description textuelle

Titre	Initialiser le système
Résumé	SB_IHM se connecte aux robots de l'essaim
Portée	SB
Niveau	Utilisateur
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun.
Préconditions	SB_IHM affiche <i>Ecran_Accueil</i> .
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM consulte <i>Fichier_Config</i>. 2. SB_IHM choisit le premier robot défini dans <i>Fichier_Config</i>. 3. SB_IHM se connecte au SB_C du robot choisi. 4. SB_C du robot choisi réinitialise sa <i>var_connexion</i> à 0. 5. SB_C du robot choisi indique qu'il est connecté à Utilisateur. 6. SB_IHM initialise le nouveau robot connecté. 7. SB_IHM met à jour l'état de connexion du robot dans <i>Liste_Robots</i>. 8. SB_IHM initialise sa <i>Liste_Fichiers_Logs</i> à l'aide de <i>Liste_Robots</i>. 9. SB_IHM met à jour <i>Ecran_Accueil</i>.
Variantes	<p>2 : [Aucun robot n'est renseigné dans <i>Fichier_Config</i>]</p> <p> 2.a.1. SB_IHM lève <i>ERREUR_AUCUN_ROBOT_RENSEIGNE</i>.</p> <p> 2.a.2. Va en 10.</p> <p>3 : [SB_C ne répond pas après épuisement du délai <i>TIME_OUT</i>]</p> <p> 3.a.1. SB_IHM abandonne la demande de connexion.</p> <p> 3.a.2. Va en 8.</p>

	<p>9 : [D'autres robots sont disponibles dans <i>Fichier_Config</i>]</p> <p>8.a.1. <i>SB_IHM</i> sélectionne le robot suivant dans le fichier de configuration de l'essaim.</p> <p>8.a.2. Va en 3.</p>
--	---

2.3.6. CU_03 Initialiser un nouveau robot connecté

2.3.6.1. Description graphique

N.A.

2.3.6.2. Description textuelle

Titre	Initialiser un nouveau robot connecté
Résumé	<i>SB_IHM</i> créer les <i>Fichier_Logs_flushed</i> et les <i>var_connexion</i> du nouveau robot connecté.
Niveau	Utilisateur
Préconditions	<i>SB_IHM</i> affiche <i>Ecran_Accueil</i> .
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> <i>SB_IHM</i> vérifie la présence de <i>Fichier_Logs_flushed</i> pour le robot choisi. <i>SB_IHM</i> créer le <i>Fichier_Logs_flushed</i> du robot choisi. <i>SB_IHM</i> vérifie si une <i>var_connexion</i> existe pour le robot choisi. <i>SB_IHM</i> créer une <i>var_connexion</i> pour le robot choisi dans <i>Liste_Robots</i>. <i>SB_IHM</i> initialise la <i>var_connexion</i> à 0. <i>SB_IHM</i> transmet l'heure de son système à <i>SB_C</i>. <i>SB_C</i> reçoit et mémorise l'heure du système de <i>SB_IHM</i>. <i>SB_C</i> synchronise l'heure de son système avec celle de <i>SB_IHM</i>.
Variantes	<p>2 : [Un <i>Fichier_Logs_flushed</i> est déjà présent sur <i>SB_IHM</i> pour ce robot]</p> <p>2.a.1. Va en 3.</p> <p>4 : [Une <i>var_connexion</i> est déjà présente dans <i>Liste_Robots</i> pour ce robot]</p> <p>4.a.1. Va en 5.</p>

2.3.7. CU_04 Commander un robot

2.3.7.1. Description graphique

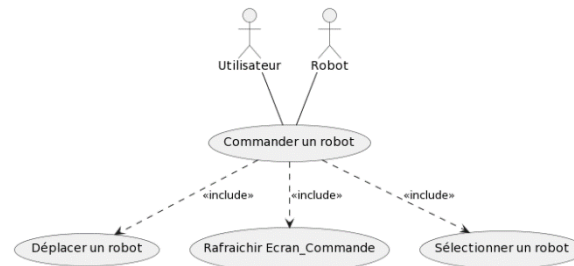


Illustration 8 : CU « Commander un robot »

2.3.7.2. Description textuelle

Titre	Commander un robot
Résumé	Utilisateur sélectionne un robot et commande ses périphériques et ses déplacements.
Portée	SB
Niveau	Utilisateur
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun.
Préconditions	SB_IHM affiche Ecran_Commande . SB_C est connecté à SB_IHM .
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> Utilisateur demande à sélectionner un robot à SB_IHM. Utilisateur <u>sélectionne un robot</u>. Utilisateur <u>déplace un robot</u>. Va en 3.
Variantes	<p>3 : [Utilisateur modifie Mode_Fonctionnement]</p> <ol style="list-style-type: none"> 3.a.1. Utilisateur demande la modification de Mode_Fonctionnement à SB_IHM. 3.a.2. SB_IHM mémorise Mode_Fonctionnement. 3.a.3. SB_IHM demande à mettre à jour Mode_Fonctionnement de SB_C. 3.a.4. SB_C mémorise Mode_Fonctionnement. 3.a.5. SB_C applique la modification de Mode_Fonctionnement. 3.a.6. SB_IHM met à jour Ecran_Commande. 3.a.7. Va en 3. <p>3 : [Utilisateur souhaite sélectionner un autre robot]</p> <ol style="list-style-type: none"> 3.b.1. Utilisateur demande à sélectionner un autre robot à SB_IHM. 3.b.2. SB_IHM indique à SB_C qu'il est désélectionné. 3.b.3. SB_C indique que sa connexion est établie. 3.b.4. SB_IHM indique que le robot est désélectionné à Utilisateur. 3.b.5. SB_IHM efface Robot_Selectionne de sa mémoire. 3.b.6. Va en 2. <p>* : [Tous les Temps_Rafraichissement]</p> <ol style="list-style-type: none"> *.a.1. SB_IHM rafraichit <u>Ecran_Commande</u>. *.a.2. Retour en *.

2.3.8. CU_05 Sélectionner un robot

2.3.8.1. Description graphique

N.A.

2.3.8.2. Description textuelle

Titre	Sélectionner un robot
Résumé	Utilisateur sélectionne un robot en cliquant sur le flux vidéo correspondant.
Portée	SB
Niveau	Fonction
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun
Préconditions	SB_IHM affiche Ecran_Commande . SB_IHM a récupéré les états des périphériques.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM mémorise le Robot_Selectionne. 2. SB_IHM met à jour l'affichage de Mode_Fonctionnement de Robot_Selectionne. 3. SB_IHM indique à SB_C qu'il est sélectionné. 4. SB_C indique qu'il est sélectionné à Utilisateur. 5. SB_IHM indique le Robot_Selectionne à Utilisateur. 6. SB_IHM autorise la commande robot à Utilisateur.
Variantes	N.A.

2.3.9. CU_06 Déplacer un robot

2.3.9.1. Description graphique

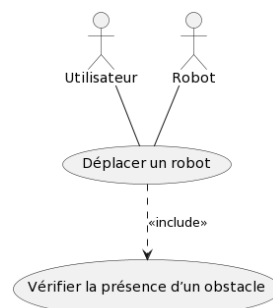


Illustration 9 : CU « Déplacer un robot »

2.3.9.2. Description textuelle

Titre	Déplacer un robot
Résumé	Utilisateur indique la direction dans laquelle le robot se déplace.
Portée	SB
Niveau	Utilisateur.
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun

Préconditions	SB_IHM affiche Ecran_Commande . Utilisateur a sélectionné un robot.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> Utilisateur demande à déplacer Robot_Selectionne. SB_IHM indique la cmd_deplacement à Utilisateur. SB_IHM transmet la cmd_deplacement à SB_C. SB_C acquitte la commande à SB_IHM. SB_C execute le déplacement cmd_deplacement. Utilisateur demande à stopper Robot_Selectionne. SB_IHM indique la cmd_arret à Utilisateur. SB_IHM transmet la cmd_arret à SB_C. SB_C acquitte la commande à SB_IHM. SB_C execute le déplacement cmd_arret.
Variantes	<p>3 : [Déplacement vers l'avant et un obstacle est détecté pour le Robot_Selectionne] 3.a.1. SB_IHM indique la cmd_arret à Utilisateur. 3.a.2. Fin du CU.</p> <p>4 : [SB_C n'acquitte pas après épuisement du délai TIME_OUT] 4.a.1. SB_IHM lève ERREUR_CONNEXION_ROBOT. 4.a.2. Fin du CU.</p> <p>* : [Tous les Délai_Detection_Obstacle] *.a.1. SB_C vérifie la présence d'un obstacle. *.a.2. Retour en *.</p>
Extensions	<p>* : [OBSTACLE_DETECTE] *.a.1. SB_IHM indique à l'Utilisateur qu'un obstacle a été détecté *.a.2. Utilisateur demande à stopper Robot_Selectionne. *.a.3. SB_IHM indique la cmd_arret à Utilisateur. *.a.4. Fin du CU.</p>

2.3.10. CU_07 Vérifier la présence d'un obstacle

2.3.10.1. Description graphique

N.A.

2.3.10.2. Description textuelle

Titre	Vérifier la présence d'un obstacle
Résumé	SB_C vérifie la présence d'un obstacle à l'aide de son radar et alerte SB_IHM le cas échéant
Portée	SB_IHM, SB_C
Niveau	Utilisateur.
Acteurs directs	Utilisateur.
Acteurs indirects	Aucun.
Préconditions	Utilisateur déplace un robot
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	

	<ol style="list-style-type: none"> 1. SB_C récupère son Mode_Fonctionnement radar. 2. SB_C vérifie la présence d'un obstacle. 3. SB_C transmet l'état de son flux radar a SB_IHM. 4. SB_IHM enregistre l'état du flux radar du SB_C concerné.
Variantes	<p>2 : [Mode_Fonctionnement radar de SB_C est désactivé]</p> <p>2.a.1. Fin du CU.</p> <p>3 : [Un obstacle est présent et déplacement vers l'avant]</p> <p>3.a.1. SB_C exécute le déplacement cmd_arret.</p> <p>3.a.2. SB_C indique à l'Utilisateur qu'un obstacle a été détecté.</p> <p>3.a.3. SB_C signale à SB_IHM qu'un obstacle a été détecté.</p> <p>3.a.4. SB_IHM lève OBSTACLE_DETECTE.</p> <p>3.a.5. Retour en *.</p>
Extensions	<p>4 : [ERREUR_RECEPTION_FLUX_RADAR]</p> <p>4.a.1. SB_IHM indique que flux radar de robot choisi est désactivé à Utilisateur.</p> <p>4.a.2. Retour en *.</p>
Détails	<p>Dans ce CU, SB_C transmet son flux radar à chaque appel. Afin de simplifier la communication en conditions réelles, SB_C ne transmettra l'état de son flux radar à SB_IHM uniquement lors du premier appel et lors des changements d'état du radar.</p>

2.3.11. CU_08 Rafraichir *Ecran_Commande*

2.3.11.1. Description graphique

N.A.

2.3.11.2. Description textuelle

Titre	Rafraichir écran
Résumé	SB_IHM met à jour les flux vidéo et radar des robots connectés
Portée	SB_IHM
Niveau	Fonction.
Acteurs directs	Utilisateur.
Acteurs indirects	Aucun.
Préconditions	Ecran_Commande est affiché par SB_IHM . Au moins un SB_C est connecté à SB_IHM .
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM consulte Liste_Robots. 2. SB_IHM choisit le premier robot connecté dans Liste_Robots. 3. SB_IHM consulte Mode_Fonctionnement radar de robot choisi. 4. SB_IHM récupère le dernier état du radar du robot choisi. 5. SB_IHM met à jour le voyant radar de robot choisi sur Ecran_Commande. 6. SB_IHM consulte Mode_Fonctionnement camera de robot choisi. 7. SB_IHM récupère la dernière image du flux camera de robot choisi. 8. SB_IHM met à jour le flux camera de robot choisi sur Ecran_Commande. 9. Fin du CU.

Variantes	<p>4 : [Mode_Fonctionnement radar de robot choisi est désactivé] 4.a.1. SB_IHM indique que le radar du robot choisi est désactivé à Utilisateur. 4.a.2. Va en 6.</p> <p>7 : [Mode_Fonctionnement camera de robot choisi est désactivé] 7.a.1. SB_IHM indique que la caméra du robot choisi est <i>désactivée</i> à Utilisateur. 7.a.2. Va en 9.</p> <p>9 : [D'autres robots connectés sont disponibles dans Liste_Robots] 9.a.1. SB_IHM choisit le robot suivant dans Liste_Robots. 9.a.2. Va en 3.</p>
-----------	---

2.3.12. CU_09 Transmettre flux caméra

2.3.12.1. Description graphique

N .A.

2.3.12.2. Description textuelle

Titre	Transmettre flux caméra
Résumé	SB_C transmet son flux caméra à SB_IHM .
Portée	SB
Niveau	Utilisateur.
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun
Préconditions	<i>La connexion entre SB_C et SB_IHM est établie.</i>
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> SB_C récupère son Mode_Fonctionnement caméra. SB_C envoie son flux caméra à SB_IHM. SB_IHM enregistre le flux caméra.
Variantes	<p>2 : [Mode_Fonctionnement caméra de SB_C est désactivé] 2.a.1. Fin du CU.</p>
Extensions	<p>3 : [ERREUR_RECEPTION_FLUX_CAMERA] 3.a.1. SB_IHM indique que flux camera de robot choisi est désactivé à Utilisateur 3.a.2. Retour en *.</p>

2.3.13. CU_10 Vérifier connexion

2.3.13.1. Description graphique

N.A.

2.3.13.2. Description textuelle

Titre	Vérifier connexion
Résumé	SB_IHM vérifie l'état de sa connexion avec chaque SB_C connecté
Portée	SB
Niveau	Fonction
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun.
Préconditions	SB_IHM et au moins un SB_C sont allumés et une connexion est active entre les deux.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM consulte Liste_Robots. 2. SB_IHM choisit le premier robot connecté dans Liste_Robots. 3. SB_IHM envoie I_AM_HERE au SB_C du robot choisi. 4. SB_C du robot choisi reçoit I_AM_HERE et met à jour sa var_connexion à 0. 5. SB_C du robot choisi acquitte la reception à SB_IHM. 6. SB_IHM reçoit l'acquittement et met à jour la var_connexion correspondante au robot choisi à 0. 7. Fin du CU.
Variantes	<p>4 : [SB_C ne reçoit pas de I_AM_HERE de SB_IHM après Time_Out_Ping]</p> <p>4.a.1. SB_C incrémente sa var_connexion de 1.</p> <p>4.a.2. Va en 7.</p> <p>5 : [SB_IHM ne reçoit pas d'acquittement du SB_C du robot choisi après Time_Out_Ping]</p> <p>5.a.1. SB_IHM incrémente la var_connexion du robot choisi de 1.</p> <p>5.a.2. Va en 7.</p> <p>7 : [D'autres robots connectés sont disponibles dans Liste_Robots]</p> <p>7.a.1. SB_IHM choisit le robot connecté suivant dans Liste_Robots.</p> <p>7.a.2. Va en 3.</p>
Détails	SB_IHM envoie régulièrement, en parallèle de toute exécution, un message aux SB_C connectés pour vérifier qu'ils sont toujours connectés. Les SB_C accusent réception du message pour signifier qu'ils sont toujours connectés. Les var_connexion de chaque SB_C s'incrémentent dès qu'un message n'est pas reçu.

2.3.14. CU_11 Consulter les logs

2.3.14.1. Description graphique

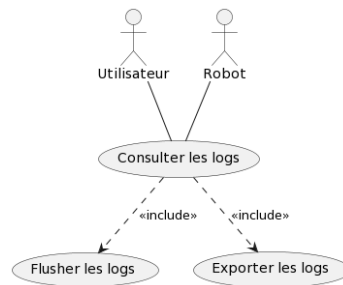


Illustration 7 : CU « Consulter les logs »

2.3.14.2. Description textuelle

Titre	Consulter les logs.
Résumé	L'utilisateur consulte les logs du SàE.
Portée	SB_IHM et SB_C .
Niveau	Utilisateur.
Acteurs directs	Utilisateur.
Acteurs indirects	Aucun.
Préconditions	SB_IHM affiche Ecran_Commande .
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM affiche Ecran_Logs. 2. Utilisateur sélectionne la Source_Logs des logs qu'il souhaite consulter. 3. SB_IHM flush les logs du SB_C lié au Source_Logs. 4. SB_IHM met à jour Ecran_Logs.
Variantes	<p>2 : [Utilisateur demande à retourner à Ecran_Commande] 2.a.1. Fin du CU.</p> <p>2 : [Une Source_Logs est sélectionnée et Utilisateur demande à exporter les logs] 2.b.1. Utilisateur <u>exporte les logs</u>. 2.b.1. Va en 4.</p> <p>2 : [Aucune Source_Logs n'est sélectionnée et Utilisateur demande à exporter les logs] 2.c.1. SB_IHM lève ERREUR_SELECTION_LOGS 2.c.1. Retour en 2.</p> <p>4 : 4.a.1. Retour en 2.</p>
Extensions	<p>* : [ERREUR_SELECTION_LOGS] *.a.1. SB_IHM affiche PopUp_Erreur. *.a.2. Utilisateur valide. *.a.3. Retour en *.</p>

2.3.15. CU_12 Flusher les logs

2.3.15.1. Description graphique

N.A.

2.3.15.2. Description textuelle

Titre	Flusher les logs
Résumé	SB_C envoie ses logs à SB_IHM puis les efface de sa mémoire
Portée	SB
Niveau	Utilisateur.
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun
Préconditions	La connexion entre SB_C et SB_IHM est établie.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM demande à récupérer les logs du SB_C concerné. 2. SB_C envoie ses logs à SB_IHM. 3. SB_IHM enregistre les nouveaux logs de SB_C dans son <i>Fichier_Logs_flushed</i> correspondant au SB_C concerné. 4. SB_IHM acquitte l'enregistrement des logs à SB_C. 5. SB_C efface les logs envoyés de sa mémoire.
Variantes	<p>2 : [SB_C ne répond pas après <i>Time_Out_Ping</i>] 2.a.1. SB_IHM lève <i>ERREUR_CONNEXION_ROBOT</i>. 2.a.2. Fin du CU.</p> <p>4 : [SB_C ne reçoit pas l'acquittement de SB_IHM après <i>Time_Out_Envoi_Logs</i>] 4.a.1. Fin du CU.</p>
Détails	Si SB_C ne reçoit pas l'acquittement de SB_IHM , alors il ne supprime pas les logs de sa mémoire. Au retour dans le CU stratégique, SB_C sauvegarde tout de même les nouveaux logs dans son fichier. S'il n'y parvient pas, les logs sont ignorés.

2.3.16. CU_13 Exporter les logs

2.3.16.1. Description graphique

N.A.

2.3.16.2. Description textuelle

Titre	Exporter les logs.
Résumé	L'utilisateur exporte les logs sélectionnés.
Portée	SB_IHM .
Niveau	Utilisateur.
Acteurs directs	Utilisateur.
Acteurs indirects	Aucun.
Préconditions	SB_IHM affiche <i>Ecran_Logs</i> .

	Source_Logs n'est pas nulle.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> SB_IHM affiche Explorateur_OS_Android. Utilisateur choisit un emplacement mémoire pour l'export. Explorateur_OS_Android effectue l'export. SB_IHM ferme Explorateur_OS_Android.
Variantes	3 : [ERREUR_EXPORT_LOGS] 3.a.1. SB_IHM affiche PopUp_Erreur . 3.a.2. Utilisateur valide. 3.a.3. Retour en *.
Détails	La fonction d'Explorateur_OS n'est pas développée par la société Robomatic. Cette fonctionnalité est dédiée à l'OS Android.

2.3.17. CU_14 Journaliser un évènement

2.3.17.1. Description graphique

N.A.

2.3.17.2. Description textuelle

Titre	Journaliser un évènement
Résumé	SB_C vérifie qu'assez d'espace mémoire est disponible pour stocker les prochains logs
Portée	SB
Niveau	Utilisateur.
Acteurs directs	Utilisateur, Robot
Acteurs indirects	Aucun
Préconditions	<i>La connexion entre SB_C et SB_IHM est établie.</i> <i>Un evenement_loggable_SB_C est levé par SB_IHM.</i>
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> SB_C vérifie que Mémoire_Min est suffisante pour stocker les prochains logs SB_C enregistre et date le nouveau log dans Fichier_Logs_SB_C.
Variantes	2 : [Mémoire_Min est insuffisante sur SB_C] 2.a.1. SB_C signale à SB_IHM que sa mémoire est insuffisante. 2.a.2. SB_IHM reçoit l'avertissement et lève ERREUR_ESPACE_MEMOIRE . 2.a.3. Fin du CU.

2.3.18. CU_15 Quitter SB_IHM

2.3.18.1. Description graphique

N.A.

2.3.18.2. Description textuelle

Titre	Quitter SB_IHM
Résumé	Utilisateur quitte l'application SB_IHM .
Portée	SB
Niveau	Fonction
Acteurs directs	Utilisateur.
Acteurs indirects	Aucun.
Préconditions	SB_IHM est démarré et est en état de fonctionner.
Garanties minimales	Aucune.
Garanties en cas de succès	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. Utilisateur demande à quitter SB_IHM. 2. SB_IHM consulte Liste_Robots. 3. SB_IHM choisit le premier robot connecté de Liste_Robots. 4. SB_IHM demande la fin de communication au SB_C du robot choisi. 5. SB_C du robot choisi confirme la fin de communication à SB_IHM. 6. SB_C du robot choisi exécute le déplacement cmd_arret. 7. SB_C du robot choisi indique qu'il est en attente de connexion. 8. SB_IHM met à jour l'état de connexion du robot choisi dans Liste_Robots. 9. SB_IHM s'arrête.
Variantes	<p>3 : [Il n'y a pas de robot connecté] 3.a.1. Va en 9.</p> <p>9 : [Il y a encore un robot connecté dans Liste_Robots] 9.a.1. SB_IHM choisit le robot connecté suivant dans Liste_Robots 9.a.2. Va en 4.</p>

2.4. Contraintes

2.4.1. Politiques réglementaires

Les robots **Alphabot 2** de WaveShare fournis par Thales étant commercialisés en vente libre, aucune confidentialité ne s'applique sur ces éléments.

2.4.2. Contraintes matérielles

N.A.

2.4.3. Exigences de fiabilité

N.A.

2.4.4. Exigences de maintenabilité

N.A.

2.4.5. Exigences de disponibilité

L'incrément 1 de ce projet sera livré le 16 mai 2023.

L'incrément 2 de ce projet sera livré le 14 juin 2023.

2.5. Hypothèses et dépendances

2.6. Répartition des exigences

3. Exigences spécifiques

3.1. Interfaces Homme Machine

3.1.1. Généralités

L'utilisateur peut interagir avec SWARMBOTS via la *Tablette* sur laquelle s'exécute **SB_IHM**. **SB_IHM** peut envoyer des informations à l'utilisateur par l'intermédiaire de la *Tablette*.

Ce dossier de spécifications prend en compte la possibilité de connecter plusieurs robots au SàE. Néanmoins, pour les incréments 1 et 2, l'IHM ne représente que le cas d'affichage ou deux robots sont connectés simultanément.

3.1.2. Les actions utilisateur

Les actions possibles de l'utilisateur principal de SWARMBOTS sont données en chapitre 2.2.2.2.1, p.11 (contexte logique).

3.1.3. Les écrans

3.1.3.1. Vue générale

L'illustration 10 (page 30) représente les navigations possibles entre les différents écrans proposés par l'IHM. Ces enchaînements sont représentés par un diagramme d'état transition UML. L'illustration 11 (page 30) rappelle les principaux concepts du diagramme d'état transition UML et leur représentation graphique.

Chaque écran est représenté par un état (rectangle arrondi sur l'illustration). Les transitions entre les écrans représentent une navigation d'un écran à l'autre en précisant l'événement logique du contexte (cf. chap. 2.2.2.1 page 11) qui active la transition. Cela correspond généralement à des actions faites par l'utilisateur pour générer l'événement correspondant.

La lettre H (pour historique) indique que le sous état actif de l'état composite est sauvegardé lorsqu'on quitte ce dernier. Ainsi, lorsqu'un retour à l'état composite est fait, si un précédent sous état a été sauvegardé, l'état actif (courant) devient ce sous état précédemment sauvegardé au sein de l'état composite. En bref, cela permet de sortir et de revenir dans un même sous état au sein de l'état composite sans devoir reprendre l'état initial de l'état composite.

Certaines transitions ne sont pas franchies sur des événements logiques, ce sont :

- Les événements de temporisation (le mot clef « after » est alors noté sur la transition) : la transition doit être sensibilisée pendant une certaine durée pour être franchie.

- Des conditions qui deviennent vraies (la condition est alors exprimée entre crochet).

ProSE A2 - Machine à états de l'IHM de SWARMBOTS

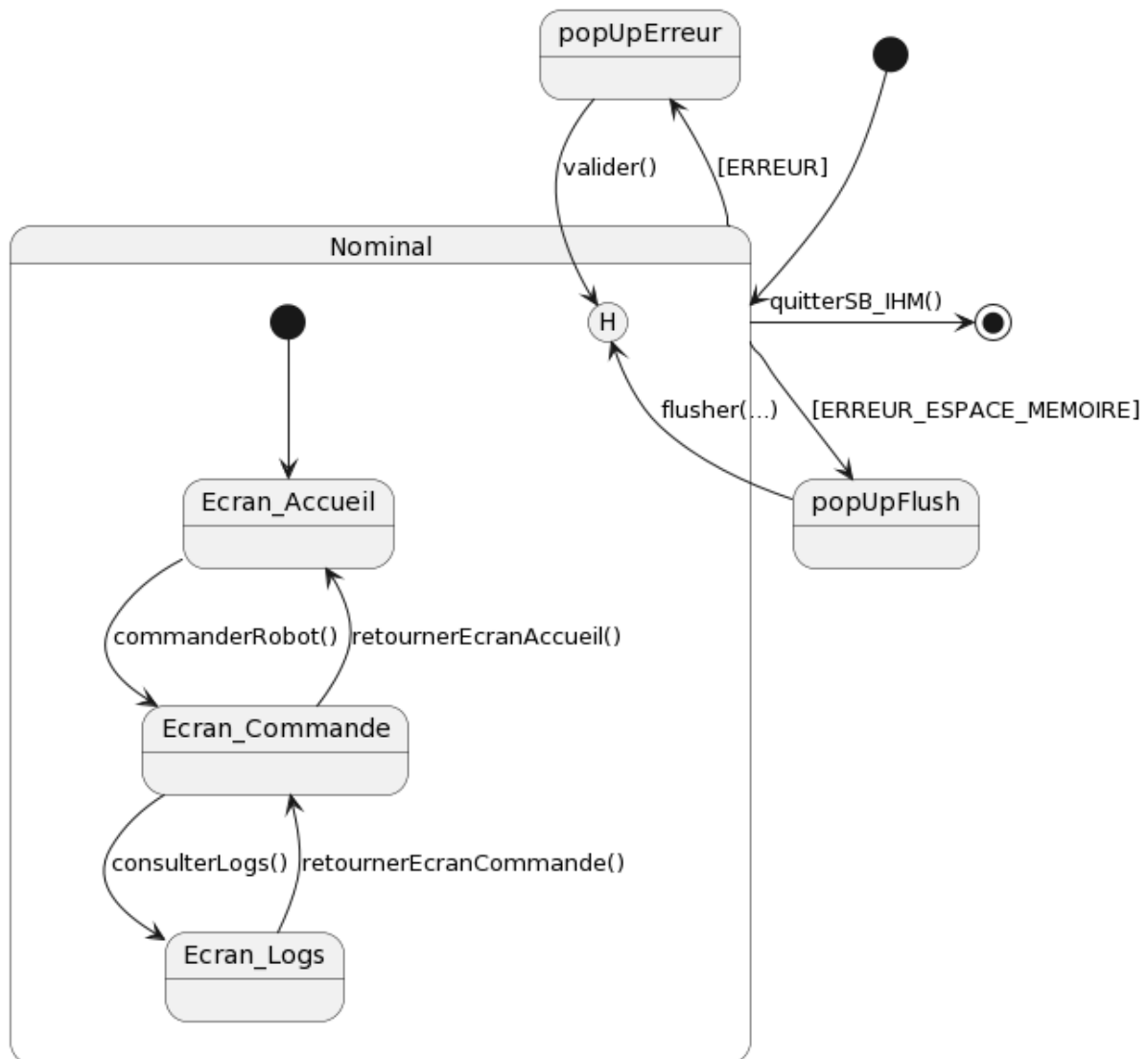


Illustration 10 : Enchaînement entre les écrans de l'IHM représenté par un diagramme état-transition UML

La sphère noire sur l'illustration représente l'état initial de création de l'IHM. Au démarrage du système, **Ecran_Accueil** est affiché. L'action `commanderRobot()` permet de passer à l'écran **Ecran_Commande**. L'action `consulterLogs()` permet d'aller vers **Ecran_Logs** depuis **Ecran_Commande**. Lors de l'exécution de **SB_IHM**, des erreurs peuvent être levées par l'application. Ces erreurs seront signalées à l'utilisateur à l'aide d'une popUp comme PopUpErreur ou PopUpFlush. Les popUp peuvent être fermées pour retourner à l'utilisation de **SB_IHM**. L'utilisateur peut effectuer différentes interactions au sein de chaque écran. Ces interactions sont détaillées dans le contexte logique, au chapitre 2.2.2.1, page 11.

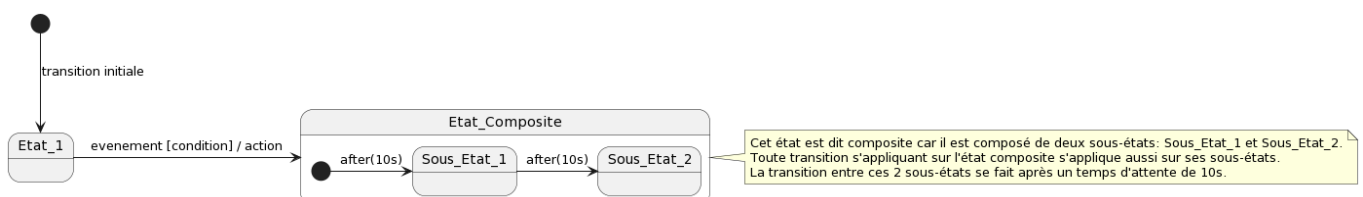


Illustration 11 : Explications sur le diagramme état-transition UML

La suite du document détaille les différents écrans présents sur l'IHM.

3.1.3.2. Ecran Accueil

L'Ecran **Ecran_Accueil** est le premier visible, il est affiché au démarrage de **SB_IHM** (demarrerSB_IHM()). Lors de l'initialisation, **SB_C** essaie de se connecter aux robots définis dans **Fichier_Config**. **Liste_Robots** est mise à jour à la suite de l'initialisation. Le message "Connexion en cours..." indique que **SB_C** est en cours d'initialisation et qu'il essaie de se connecter aux robots. Si une erreur survient lors de la lecture du Fichier_Config ou qu'aucun robot n'est renseigné, un message d'erreur est affiché à l'écran dans le champ <message_accueil>. C'est la levée de l'erreur **ERREUR_AUCUN_ROBOT_RENSEIGNE** qui permet de mettre à jour l'affichage.



Illustration 12 : Ecran Accueil (avant initialisation)

Une fois **SB_C** initialisé, la liste des robots **Liste_Robots** est affichée et renseigne de l'état de connexion de chaque robot.

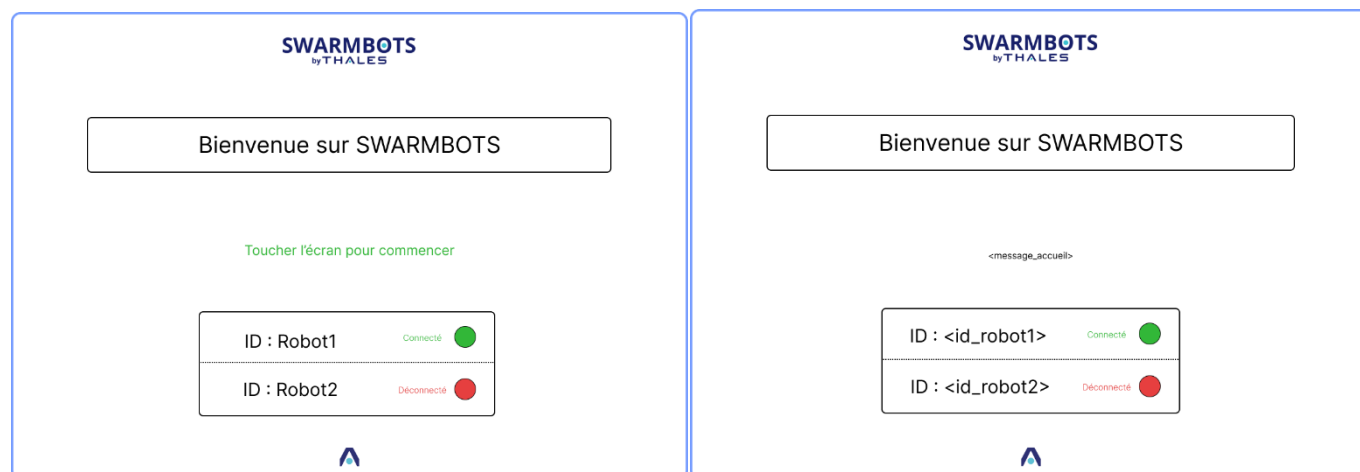






Illustration 13 : Ecran Accueil (après initialisation)

Depuis l'**Ecran_Accueil**, l'Utilisateur peut toucher l'écran pour commander un robot et passer à l'**Ecran_Commande** (commanderRobot()).

Description des éléments de **Ecran_Accueil** de **SB_IHM** :

Elément graphique	Fonction/Signification
Bienvenue sur SWARMBOTS	Message de bienvenue non interactif.
Toucher l'écran pour commencer	Apparaît lorsque SB est initialisé.
ID : RobotX	Désigne un robot dont l'ID correspondant à « X » est défini dans Fichier_Config .
Connecté 	Indique l'état d'un robot connecté.
Déconnecté 	Indique l'état d'un robot déconnecté.
<div> <div>ID : Robot1 </div> <div>ID : Robot2 </div> </div>	Liste des robots définis dans Fichier_Config contenant leur ID et l'état de connexion correspondant.

3.1.3.3. **Ecran_Commande**

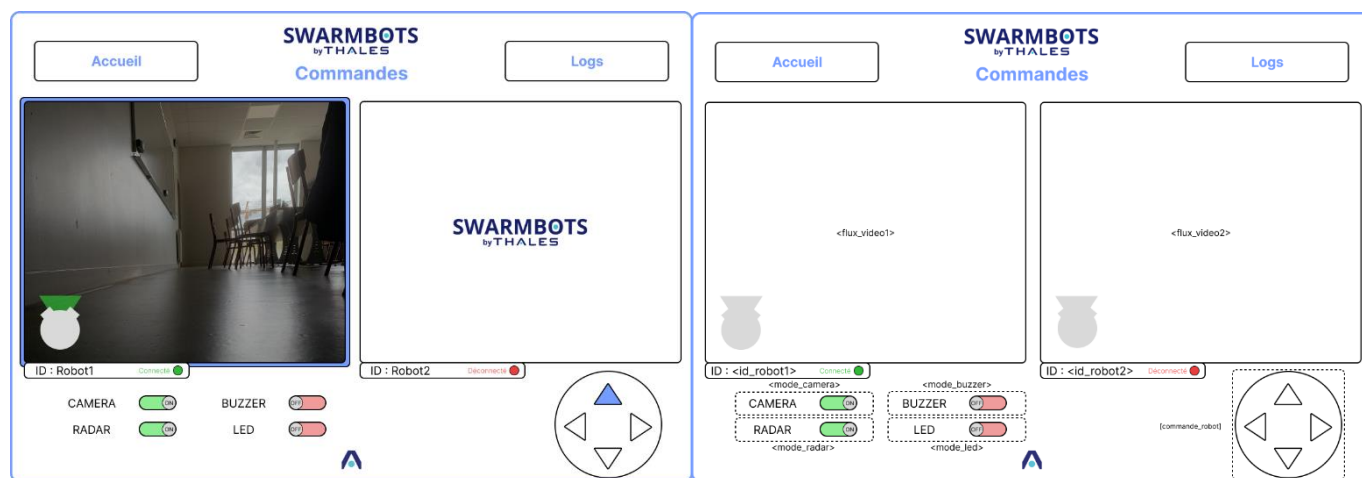


Illustration 14 : Ecran_Commande

L'écran **Ecran_Commande** est l'écran principal de SWARMBOTS, il permet :

- De sélectionner le robot à commander.
- De commander les robots connectés (avancer, reculer, tourner à gauche, tourner à droite et stopper).
- De modifier **Mode_Fonctionnement** de chacun des robots
- De visualiser le flux vidéo de chacun des robots connectés en **Temps_Reel**
- De visualiser le voyant radar de chacun des robots connectés en **Temps_Reel**

Boutons de navigation :

Le bouton <Accueil> situé en haut à gauche permet de retourner à l'écran **Ecran_Accueil**. Il déclenche l'action retournerEcranAccueil() du contexte logique.

Le bouton <Logs> situé en haut à droite permet de consulter les logs via l'écran **Ecran_Logs**. Il déclenche l'action `consulterLogs()` du contexte logique.

Flux vidéo :

Les deux rectangles situés au centre correspondent à deux affichages vidéo du retour caméra des robots définis par l'utilisateur dans **Fichier_Config**. Si le robot est déconnecté, le flux vidéo est remplacé par un écran grisé comme représenté sur l'illustration 14. L'ID de chaque robot est indiqué sous chaque flux vidéo (rectangle). Lorsque qu'un robot est connecté, il est possible de le sélectionner en cliquant sur son flux vidéo. Cela a pour effet de mettre à jour l'affichage de son **Mode_Fonctionnement** ainsi que d'autoriser la commande de direction du robot. Les flux radar et video sont mis à jour tous les **Temps_Rafraichissement** par les fonctions `majFluxRadar()` et `majFluxCamera()`, renseignées dans le contexte logique.

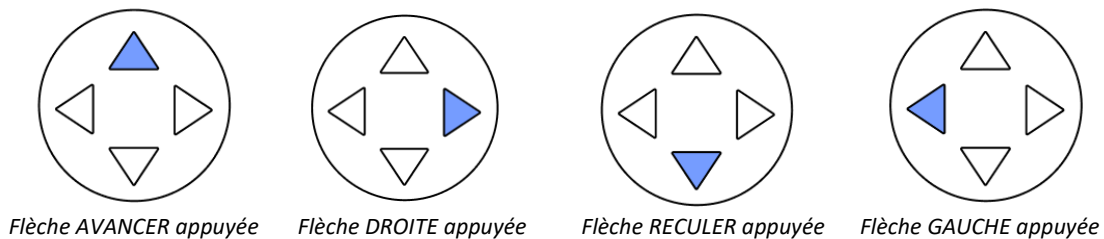
Commandes :

Les commandes correspondent aux actions avancer, reculer, tourner à gauche, tourner à droite et stopper. Elles sont représentées par des flèches directionnelles :

- Flèche directionnelle vers le haut : correspond à AVANCER de **cmd_deplacement**.
- Flèche directionnelle vers le bas : correspond à RECULER de **cmd_deplacement**.
- Flèche directionnelle vers la droite : correspond à DROITE de **cmd_deplacement**.
- Flèche directionnelle vers la gauche : correspond à GAUCHE de **cmd_deplacement**.
- Arrêt : correspond au moment où l'utilisateur relâche une des flèches directionnelles sur laquelle il a précédemment appuyé. Cette action correspond à STOPPER de **cmd_arret**.

Ces commandes correspondent aux fonctions `deplacerRobot(...)` et `stopperRobot(...)` du contexte logique.

Lorsque l'utilisateur appuie sur une des flèches directionnelles, SB l'indique à l'utilisateur en changeant la couleur du bouton en bleu. Le relâchement de la flèche par l'utilisateur engendre l'exécution de la fonction `stopperRobot(...)` et change à nouveau la couleur en blanc.



État des périphériques :

Quatre périphériques sont dits activable ou désactivable par l'utilisateur (voir dictionnaire de domaine), ils sont représentés par les 4 boutons ON/OFF, au centre, en dessous des flux vidéo. La liste des périphériques dont l'état est modifiable (activé/désactivé) est la suivante :

- Caméra : correspond à l'état de la caméra présent en tant que périphérique sur **Carte_Elec**.
- Radar : correspond à l'état du radar présent en tant que périphérique sur Robot.
- Buzzer : correspond à l'état du buzzer présent en tant que périphérique sur Robot.
- LED : correspond à l'état des LED présentes en tant que périphériques sur Robot.

L'activation ou la désactivation d'un périphérique par l'utilisateur entraîne l'exécution de la fonction `modifierModeFonctionnement(...)` du contexte logique. Les périphériques sont décrits plus précisément au chapitre 2.2.1, p.10 au niveau de l'architecture matérielle et logicielle.

Voyant radar :

En bas à gauche de chaque flux vidéo figure un logo permettant d'indiquer si un obstacle est détecté par le radar du robot. Le logo est constitué d'une partie ronde qui représente le robot et d'un voyant qui peut prendre les 3 états différents suivants :

- Lorsque le radar est désactivé : le voyant radar disparaît de l'écran.
- Lorsque le radar est activé, deux sous états sont possibles :
 - o Le voyant radar est vert lorsque le radar ne détecte pas d'obstacle.
 - o Le voyant radar est rouge lorsqu'un obstacle est détecté.



Voyant radar activé : pas d'obstacle



Voyant radar activé : obstacle détecté

Le changement d'état du logo est effectué par la fonction indiquerObstacle() du contexte logique.

Il est bon de rappeler qu'un affichage dynamique de cet écran n'est pas prévu sur cette phase de prototypage (incrément 1 et incrément 2). L'**Ecran_Commande** sera donc figé comme il est représenté ci-dessus dans le cadre des deux incréments. Le logiciel sera développé de la manière la plus modulaire possible afin de laisser place à une évolution future qui pourra rendre cet affichage dynamique en fonction du nombre de robots.

3.1.3.4. Ecran_Logs

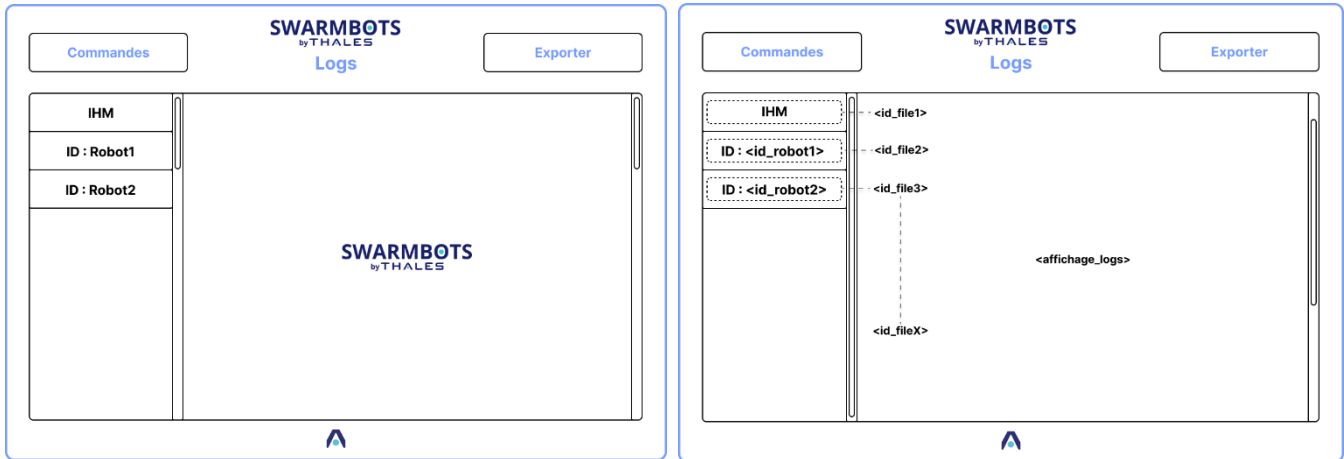


Illustration 15 : Ecran_Logs (Aucun Source_Logs)

Ecran_Logs est l'écran permettant à l'utilisateur de consulter les logs de SWARMBOTS, il permet :

- De consulter les logs de l'IHM.
- De consulter les logs de chaque robot séparément.
- D'**exporter** les logs dans un fichier externe.

Boutons de la page :

- Le bouton <Commande> situé en haut à gauche permet de retourner à l'écran **Ecran_Commande**. Il correspond à l'action retournerEcranCommande() du contexte logique.
- Le bouton <Export> situé en haut à droite permet d'**exporter** les logs sauvegardés sur **SB_IHM**. Ils sont copiés et exportés dans un nouveau fichier externe à **SB_IHM**. L'action correspondante au contexte logique est exporterLogs(...).

Menu déroulant :

- Partie gauche :
 - Cette liste déroulante contenant des boutons à sélectionner permet à l'utilisateur de sélectionner les logs qu'il souhaite consulter. selectionnerSourceLogs(...) du contexte logique est l'action associée.
 - L'utilisateur peut consulter les logs de <IHM>.
 - Il peut également consulter les logs de chaque robot séparément. Cette séparation est indiquée par l'identifiant des robots. <ID : <id_robot>>.

- Partie droite :
 - Cette partie d'écran affiche les logs de **Source_Logs**. Il est possible de faire défiler la liste des logs. Cela correspond à la fonction `afficherLogsSourceSelectionne(...)` du contexte logique.

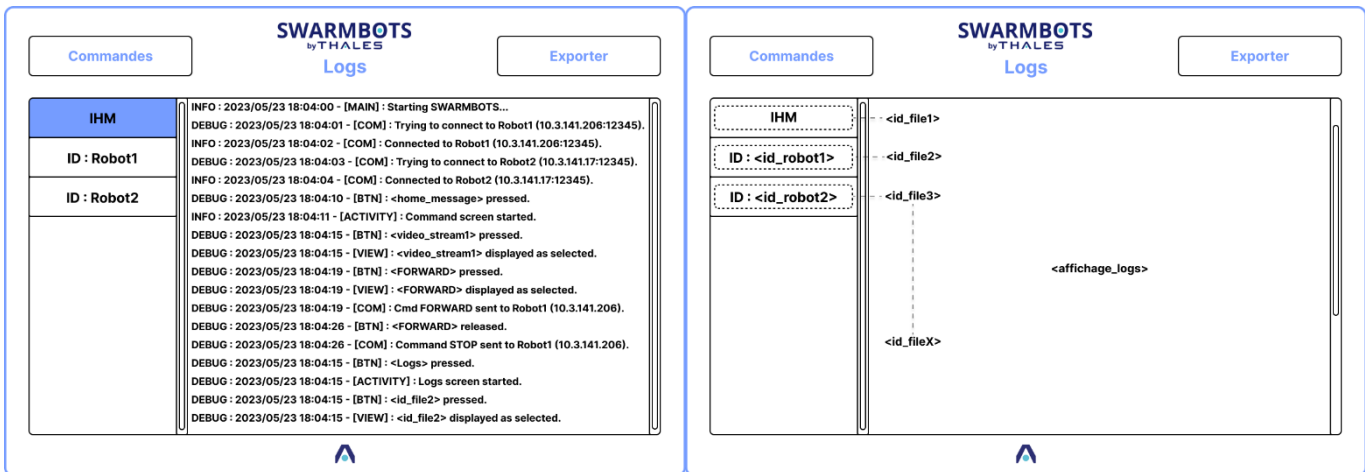


Illustration 16 : Ecran Logs (avec Source_Logs)

3.1.3.5. PopUp_Erreur

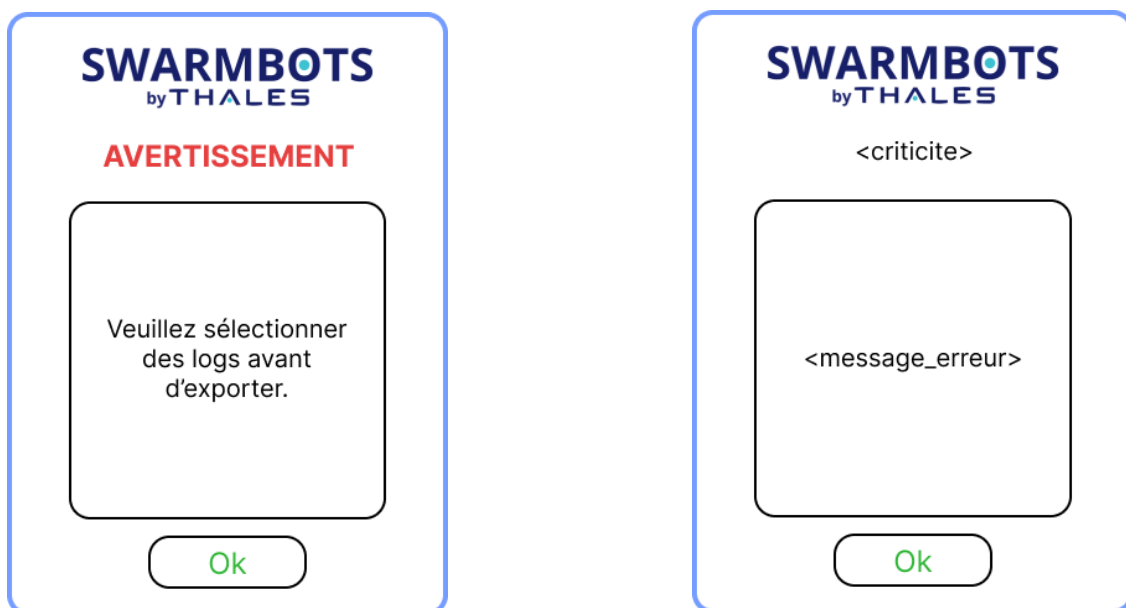


Illustration 17 : PopUp_Erreur

La PopUp d'erreur permet d'avertir l'utilisateur d'une erreur qui se produit durant le cours d'exécution de l'application. La liste des messages d'erreur affichés est décrite dans le dictionnaire de domaine, au chapitre 3.3.

Champs de la PopUp :

- <criticite> : champ indiquant le niveau de criticité de l'erreur. Se référer au dictionnaire de domaine : Criticité des logs.
- <message_erreur> : champ contenant le message relatif à l'erreur.

Boutons de la PopUp :

- [Ok] : La sélection de ce bouton entraîne l'action `valider()` décrite dans le contexte logique ainsi que la fermeture de la **PopUp_Erreur**.

3.1.3.6. PopUp Flush

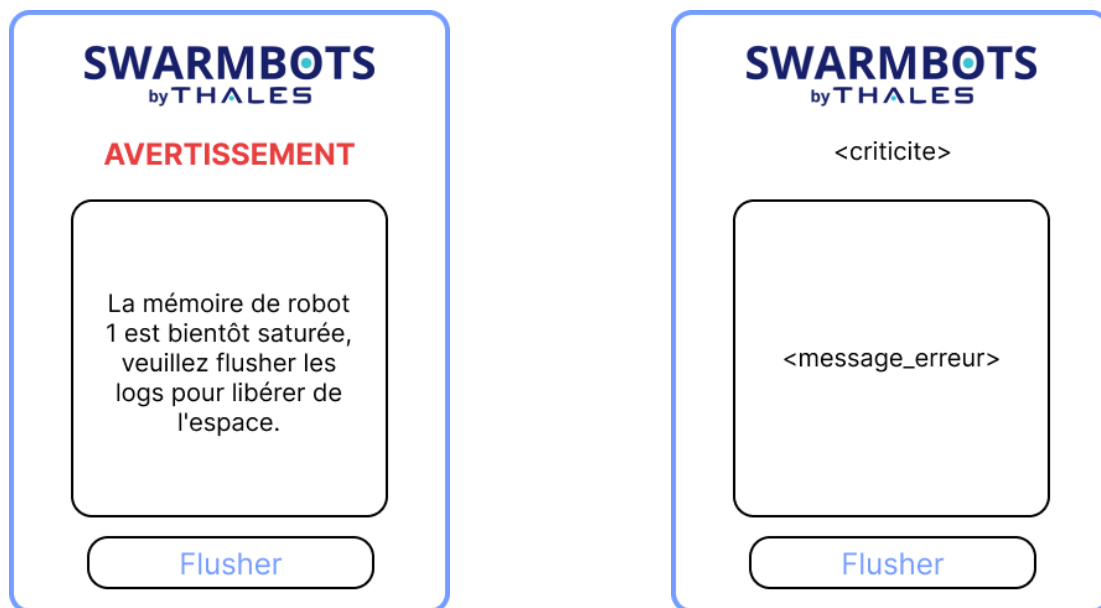


Illustration 18 : PopUp Flush

La **PopUp_Flush** permet d'avertir l'utilisateur qu'un des robots n'a plus la mémoire suffisante **Mémoire_Min** pour stocker de futurs logs.

Champs de la PopUpFlush :

- « criticite » : champ indiquant le niveau de criticité de l'erreur. Se référer au dictionnaire de domaine : Criticité des logs.
- <message_erreur> : champ permettant d'indiquer à l'utilisateur le robot concerné par le message.

Boutons de la PopUp :

- [Flusher] : La sélection de ce bouton entraîne la récupération des logs stockés dans le robot ayant levé l'alerte vers le stockage de logs de SB_IHM et ferme la PopUp. Cela correspond plus précisément à l'opération flusher(...) décrite dans le contexte logique.

3.2. Description des fonctions

3.2.1. Récupérer Mode_Fonctionnement

3.2.1.1. Description graphique

N.A.

3.2.1.2. Description textuelle

Titre	Récupérer Mode_Fonctionnement
Résumé	SB_IHM récupère les Mode_Fonctionnement stockés par les SB_C connectés.
Niveau	Fonction
Préconditions	SB_IHM affiche Ecran_Commande .
Garanties minimales	Aucune.
Scénario nominal	<ol style="list-style-type: none"> 1. SB_IHM consulte Liste_Robots. 2. SB_IHM choisit le premier robot connecté dans Liste_Robots. 3. SB_IHM demande à récupérer Mode_Fonctionnement au SB_C du robot choisi. 4. SB_C du robot choisi envoie Mode_Fonctionnement. 5. SB_IHM mémorise Mode_Fonctionnement du robot choisi. 6. Fin de la fonction.
Variantes	<p>6 : [D'autres robots connectés sont disponibles dans Liste_Robots]</p> <p>6.a.1. SB_IHM sélectionne le robot connecté suivant dans la Liste_Robots.</p> <p>6.a.2. Va en 3.</p>

3.3 Dictionnaire du domaine

3.3.1 Définitions

Terme	Définition
Alphabot 2	Support contenant <i>Carte_Elec</i> .
Carte_Elec	Support d'exécution de SB_C .
cmd	Ce terme désigne le type de commande qui peut être transmise à un robot. Il peut prendre deux valeurs : <ul style="list-style-type: none"> • cmd_arret ; • cmd_deplacement.
cmd_arret	Ce terme désigne une commande qui peut être exécutée par un Robot ou envoyée par SB_IHM à SB_C . Il en existe 1 : <ul style="list-style-type: none"> • STOPPER : Le Robot s'arrête.
cmd_deplacement	Ce terme désigne une commande qui peut être exécuté par un Robot ou envoyée par SB_IHM à SB_C . Il en existe 4 : <ul style="list-style-type: none"> • AVANCER : Le Robot avance. • RECULER : Le Robot recule. • GAUCHE : Le Robot tourne sur lui-même dans le sens antihoraire. • DROITE : Le Robot tourne sur lui-même dans le sens horaire.
Criticité des logs	Ce terme indique le niveau d'alerte des logs : <ul style="list-style-type: none"> • <u>DEBUG</u> : log de niveau très faible, remonte tout type d'action effectuée par le SàE. Souvent utilisé pour le debogage. • <u>INFO</u> : log de niveau faible, remonte des informations sur les actions du SàE à titre informatif. • <u>WARNING</u> : log de niveau élevé, remonte un point de vigilance auquel il est bon de prêter attention. Conduit rarement à un arrêt non désiré du SàE. • <u>ERROR</u> : log de niveau très élevé, remonte une erreur lors de l'exécution d'une portion du programme. Conduit souvent à un arrêt non désiré du SàE.
Delai_Detection_Obstacle	Constante désignant l'intervalle de temps de vérification de la présence d'un obstacle. Constante exprimée en millisecondes.
Ecran_Accueil	Terme désignant l'écran de SB_IHM utilisé pour l'accueil de l'application.
Ecran_Commande	Terme désignant l'écran de SB_IHM permettant de commander le robot.
Ecran_Logs	Terme désignant l'écran de SB_IHM utilisé pour consulter les logs.
En état de fonctionner	Cette expression signifie selon SB_IHM ou SB_C que : <ul style="list-style-type: none"> • SB_IHM a été correctement chargé sur <i>Tablette</i>. • SB_C a été correctement configuré sur <i>Carte_Elec</i>.
Log	Terme désignant la façon dont est construite l'en-tête d'un log. De la forme : [Criticité] : [Jour_semaine] [Mois] [Jour] [Timestamp] [Année] – [Message]. Avec : <ul style="list-style-type: none"> - [Jour_semaine] contient les 3 premières lettres du jour ; - [Mois] pour le mois ; - [Timestamp] de la forme : HH:MM:SS : <ul style="list-style-type: none"> - HH pour l'heure ; - MM pour les minutes ; - SS pour les secondes. - [Année] pour l'année ; - [Message] le message du log. Exemple : WARNING : Tue May 23 21:43:07 2023 – Message.

etat_connexion	<p>Renseigne sur l'état de connexion entre SB_IHM et SB_C. Prends trois valeurs énumérées :</p> <ul style="list-style-type: none"> ETABLIE : les deux systèmes sont connectés, les LEDs de SB_C sont en vert. EN_ATTENTE : SB_C est en attente de connexion de SB_IHM, les LEDs de SB_C clignotent en vert. EN_COURS : La connexion est en train de s'établir.
etat_obstacle	<p>Renseigne sur la présence d'un obstacle. Ce paramètre peut prendre 2 valeurs énumérées :</p> <ul style="list-style-type: none"> PRESENT ABSENT
evenement_loggable_SB_C	<p>Ce terme désigne tout évènement à enregistrer sous forme de log dans Fichier_Logs_SB_C.</p> <p>Les évènements loggables avec leurs niveaux de criticités sont :</p> <ul style="list-style-type: none"> Le démarrage et l'arrêt de SB_C au niveau INFO. Les changements d'etat_connexion de SB_C au niveau INFO. Les mouvements du robot et changement des modes de fonctionnement des périphériques au niveau INFO. Une déconnexion inattendue entre SB_C et SB_IHM au niveau ERROR. Tout échange entre SB_C et SB_IHM au niveau DEBUG.
evenement_loggable_SB_IHM	<p>Ce terme désigne tout évènement à enregistrer sous forme de log dans Fichier_Logs_SB_IHM.</p> <p>Les évènements loggables avec leurs niveaux de criticités sont :</p> <ul style="list-style-type: none"> Le démarrage et l'arrêt de SB_IHM au niveau INFO. Les changements de connexion entre SB_IHM et SB_C au niveau INFO. Les interactions de l'utilisateur avec l'écran au niveau INFO. Les erreurs levées par SB_IHM au niveau ERROR. Tout échange entre SB_C et SB_IHM au niveau DEBUG.
Explorateur_OS_Android	<p>Ce terme désigne le mécanisme du système d'exploitation Android permettant d'accéder aux fichiers stockés sur le périphérique. On peut y choisir d'enregistrer ou de charger des fichiers.</p>
Export	<p>Ce terme, dans le contexte du SàE, désigne le fait d'enregistrer la totalité des logs de SB_IHM dans un fichier externe sur <i>Tablette</i>. Cette fonctionnalité est dédiée à l'OS.</p>
Fichier_Config	<p>Fichier de configuration présent sur <i>Tablette</i> qui contient les paramètres de connexion pour chaque robot, listés ci-dessous :</p> <ul style="list-style-type: none"> Identifiants (ID) du robot ; Adresse IP du robot ; Port de connexion du robot ; Adresse MAC du robot.
Fichier_Logs_SB_IHM	<p>Ce fichier présent sur SB_IHM contient les logs générés par SB_IHM. Ces logs stockés sont de la forme Log.</p>
Fichier_Logs_SB_C	<p>Ce fichier présent sur SB_C contient les logs non flushés générés par SB_C. Ces logs sont de la forme Log.</p>
Fichier_Logs_flushed	<p>Ce fichier présent sur SB_IHM contient les logs flushés d'un SB_C. Chaque SB_C dispose de son propre fichier sur SB_IHM. Ces logs sont de la forme Log.</p>
I_AM_HERE	<p>Message envoyé par SB_IHM afin de vérifier que la connexion est toujours établie entre SB_IHM et les SB_C.</p>
id_ecran	<p>Ce terme désigne un des écrans du SàE. Il peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> Ecran_Accueil Ecran_Commande Ecran_Erreur

	<ul style="list-style-type: none"> • Ecran_Logs
id_erreur	<p>Des erreurs peuvent être levées afin d'avertir l'utilisateur d'un dysfonctionnement au sein du SàE. Les erreurs sont numérotées de 1 à 1 :</p> <ul style="list-style-type: none"> • ERREUR_ESPACE_MEMOIRE : Lorsque la Mémoire_Min disponible n'est pas suffisante sur Carte_Elec pour stocker de nouveaux logs. <i>Message associé : La mémoire de robot {id_robot} est bientôt saturée, veuillez flusher les logs pour libérer de l'espace.</i> • ERREUR_AUCUN_ROBOT_RENSEIGNE : Lorsque aucun robot n'est renseigné dans Fichier_Config. <i>Message associé : Veuillez renseigner au moins 1 robot dans le fichier de configuration.</i> • ERREUR_CONNEXION_ROBOT : Indique qu'un robot s'est déconnecté pendant l'exécution de l'application. <i>Message associé : Deconnexion inattendue du robot {id_robot}.</i> • ERREUR_RECEPTION_FLUX_CAMERA : Indique que SB_IHM n'a pas réussi à accéder au flux camera de SB_C. <i>Message associé : Impossible de récupérer le flux caméra de robot {id_robot}.</i> • ERREUR_RECEPTION_FLUX_RADAR : Indique que SB_IHM n'a pas réussi à accéder au flux radar de SB_C. <i>Message associé : Impossible de récupérer le flux radar de robot {id_robot}.</i> • ERREUR_SELECTION_LOGS : Lorsque l'utilisateur souhaite exporter les logs mais qu'aucun logs n'a été sélectionné. <i>Message associé : Veuillez sélectionner des logs avant d'exporter.</i> • ERREUR_EXPORT_LOGS : Lorsque Explorateur_OS_Android ne parvient pas à exporter les logs. <i>Message associé : Une erreur est survenue pendant l'export des logs.</i> <p>Le message associé est utilisé lorsqu'une PopUp_Erreur est affichée par SB_IHM.</p>
id_flux	<p>Ce terme désigne le type de flux. Deux valeurs sont possibles :</p> <ul style="list-style-type: none"> • Flux radar : correspond au flux radar du radar présent sur Robot • Flux caméra : correspond au flux caméra de la caméra présente sur Robot
id_file	<p>Identifiant unique permettant de désigner les différents fichiers de logs présents sur SB_IHM. Les fichiers de logs correspondent à ceux de l'IHM et des différents robots. Lorsque les logs d'un robot sont récupérés par SB_IHM, il les stocke dans le fichier de logs correspondant. Les valeurs que peut prendre id_file sont détaillées ci-dessous :</p> <ul style="list-style-type: none"> - GUILogsFile. - RobotXLogsF <p>Avec x correspondant à un des id_robot définis dans Fichier_Config</p>
id_robot	<p>Identifiant unique du robot, il s'agit d'un entier compris entre 1 et 2.</p>
Limite_Connexions_Manquees	<p>Constante comparée à var_connexion définissant le nombre maximum d'échecs de communication entre SB_IHM et SB_C. La valeur prise par cette constante est de 3 dans le cadre de cette étude.</p>
Liste_Fichiers_Logs	<p>Cette liste contient les id_File de chaque fichier log présent sur SB_IHM. Un fichier est présent par défaut pour SB_IHM, on retrouve également un fichier log par robot défini dans Liste_Robots (créé lors de l'initialisation de SB).</p>
Liste_Robots	<p>Ce terme désigne une liste contenant toutes les informations relatives aux robots définis dans Fichier_Config. Pour chaque robot contenu dans la liste, on retrouve les paramètres suivants :</p> <ul style="list-style-type: none"> • Les informations du robot contenues dans Fichier_Config (ID , IP, port, adresse MAC)

	<ul style="list-style-type: none"> • L'etat_connexion de chaque robot • Une variable var_connexion stockant les potentielles erreurs liées à la connexion entre SB_IHM et le SB_C du robot concerné. • Le Mode_Fonctionnement de chaque robot <p>Cette liste est manipulée par SB_IHM afin d'ajuster l'affichage.</p>
Source_Logs	<p>Terme désignant les logs choisis par l'utilisateur dans le cadre d'une sélection sur Ecran_Logs. Les logs consultables et sélectionnables par l'utilisateur sont :</p> <ul style="list-style-type: none"> • Les logs de SB_IHM, enregistrés dans Fichier_Logs_SB_IHM sur SB_IHM. • Les logs du SB_C ou du robot sélectionné, enregistrés dans Fichier_Logs_flushed sur SB_IHM une fois flushés.
Mémoire_Min	<p>Taille de la mémoire minimale qui doit être disponible sur Carte_Elec afin que de nouveaux logs soient stockés sur la cible. <i>Taille dimensionnée en phase de conception.</i></p>
Mode_Fonctionnement	<p>Regroupe les différents états des périphériques présents sur Carte_Elec :</p> <ul style="list-style-type: none"> • Etat_Caméra • Etat_Radar • Etat_Buzzer • Etat_LEDs <p>Ces états peuvent prendre une des deux valeurs énumérées suivantes :</p> <ul style="list-style-type: none"> • ACTIVE • DESACTIVE
Multi-Touch	<p>En informatique, le multi-touch est une technologie qui permet à une surface (un pavé tactile ou un écran tactile) de reconnaître la présence de plus d'un point de contact avec la surface en même temps.</p>
Obstacle	<p>Objet se trouvant sur la trajectoire du robot.</p>
OBSTACLE_DETECTE	<p>Ce terme correspond à la détection d'un obstacle. Lorsque SB_C détecte un obstacle, il prévient SB_IHM qui lève cet événement afin d'avertir l'utilisateur.</p>
Périphérique	<p>Dispositif connecté à Carte_Elec ayant une fonction spécifique. Il peut s'agir d'un moteur, capteur, actionneur ou de tout autre appareil physique.</p>
Périphérique activé {ACTIVE}	<p>Ce terme indique que le périphérique est actif.</p>
Périphérique désactivé {DESACTIVE}	<p>Ce terme indique que le périphérique est inactif.</p>
Point_Acces_Wifi	<p>Point d'accès wifi du réseau wifi utilisé par le SàE</p>
PopUp	<p>Ce terme désigne une fenêtre qui s'ouvre devant l'écran principal sans avoir été sollicitée par l'utilisateur.</p>
PopUp_Erreur	<p>Terme désignant l'apparition d'une PopUp en superposition de l'écran affiché sur SB_IHM. Cette PopUp apparaît afin d'afficher un message d'erreur à l'écran.</p>
PopUp_Flush	<p>Terme désignant l'apparition d'une PopUp en superposition de l'écran affiché sur SB_IHM. Cette PopUp apparaît afin d'avertir l'utilisateur d'un problème mémoire sur SB_C.</p>
Robot_Selectionne	<p>Cette expression désigne le robot en cours d'utilisation par l'utilisateur. Un robot est « sélectionné » lorsque l'utilisateur appuie sur SB_IHM sur le flux vidéo du robot en question.</p> <p>Lorsqu'un robot est sélectionné, sur Ecran_Commande, son flux vidéo est encadré en bleu et l'état des périphériques indique l'état actuel des périphériques du robot.</p>
SB_C	<p>Logiciel exécuté sur Robot et, communiquant avec SB_IHM.</p>
SB_IHM	<p>Logiciel exécuté sur <i>Tablette</i>, communiquant avec SB_C.</p>
Tablette	<p>Support d'exécution de SB_IHM.</p>
Temps_Envoi_Ping	<p>Correspond à la fréquence d'envoi du I_AM_HERE. Cette fréquence est de l'ordre de 0,1 Hz.</p>
Temps_Rafraichissement	<p>Correspond au nombre de fois par seconde durant lesquelles Ecran_Commande</p>

	rafraîchit les flux radar et caméra des robots. Constante exprimée en Hertz dont la valeur sera déterminée en phase de conception.
Temps_Reel	Exécution logicielle ou physique dans un intervalle de temps minime par rapport à l'évolution du système avec lequel il est en relation.
TIME_OUT	Constante correspondant au délai de réponse de SB_C après une demande de connexion de SB_IHM . Cette constante a pour valeur 4 secondes.
Time_Out_Ping	Constante correspondant au délai de réponse maximal accepté lors des vérifications de connexion entre SB_C et SB_IHM . Cette constante a pour valeur 2 secondes.
Time_Out_Envoi_Logs	Constante correspondant au délai de réponse maximal accepté lorsque SB_C a terminé la transmission de ses logs à SB_IHM . Cette constante a pour valeur 5 secondes.
Valeur énumérée	Correspond à un ensemble de valeurs constantes.
var_connexion	<p>Variable présente sur chaque SB_C et sur SB_IHM pour chaque robot dans Liste_Robots. Elle permet de stocker les erreurs de connexion entre SB_IHM et SB_C afin de détecter de potentielles deconnexions. Cette variable est incrémentée lorsque les erreurs de connexion suivantes surviennent :</p> <ul style="list-style-type: none"> • Dès qu'un « I_AM_HERE » de SB_IHM n'a pas été reçu par SB_C • Dès qu'un accusé de réception envoyé par SB_C n'a pas été reçu par SB_IHM.