

Projet CSC4102 : *Smart* Traçabilité pour Suivi des Incidents

Fauchier Camille et Ammari Saad

Année 2021–2022 — 03 janvier 2022

Table des matières

1	Spécification	2
1.1	Diagrammes de cas d'utilisation	2
1.2	Priorités, et préconditions et postconditions des cas d'utilisation	3
2	Préparation des tests de validation	7
2.1	Tables de décision des tests de validation	7
3	Conception	11
3.1	Diagramme de classes	11
3.2	Diagrammes de séquence	12
4	Diagrammes de machine à états et invariants	13
4.1	Classes Envoi	13
5	Préparation des tests unitaires	14
5.1	Classe Envoi	14
5.2.	Classe Client	14
5.3.	Classe Transporteur	15
5.4.	Classe JalonAvancement	15
5.5.	Classe SourceIoT	15
5.6.	Classe Enregistrement	16
5.7.	Classe Condition	16

1 Spécification

1.1 Diagrammes de cas d'utilisation

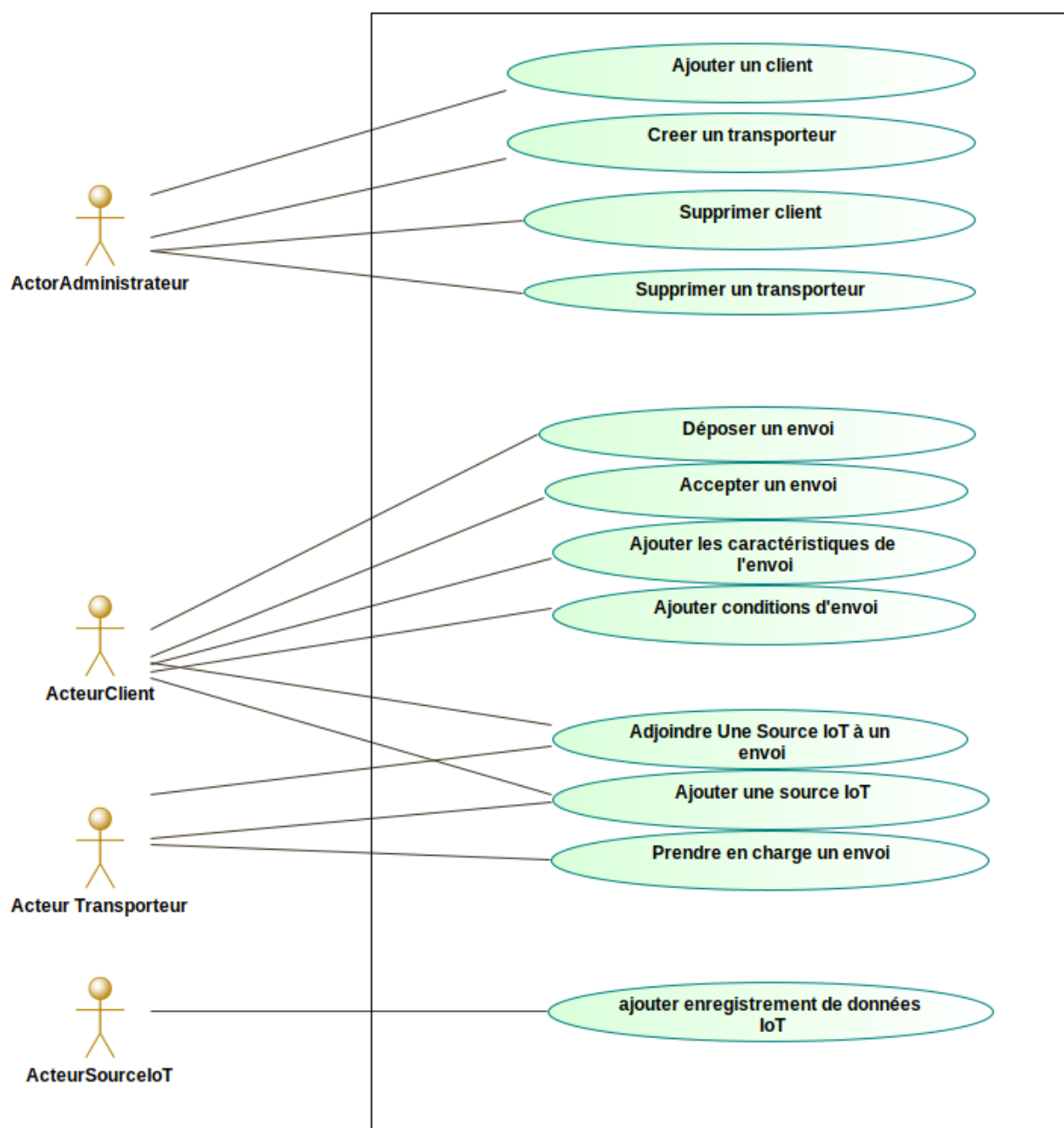


FIGURE 1 - Diagramme de cas d'utilisation

1.2 Priorités, et préconditions et postconditions des cas d'utilisation

Les priorités des cas d'utilisation pour le sprint 1 sont choisies avec les règles de bon sens suivantes :

- Pour retirer une entité du système, elle doit y être. La priorité de l'ajout est donc supérieure ou égale à la priorité du retrait ;
- Pour lister les entités d'un type donné, elles doivent y être. La priorité de l'ajout est donc supérieure ou égale à la priorité du listage ;
- Il est *a priori* possible, c.-à-d. sans raison contraire, de démontrer la mise en œuvre d'un sous-ensemble des fonctionnalités du système, et plus particulièrement la prise en compte des principales règles de gestion, sans les retraits ou les listages.
- La possibilité de lister aide au déverminage de l'application pendant les activités d'exécution des tests de validation ;
- Pour prendre en charge un envoi, ce dernier doit déjà exister. La priorité de déposer un envoi est donc supérieure ou égale à la priorité de prendre en charge l'envoi ;
- Pour déposer un envoi, le client doit avoir été ajouté par l'administrateur.

Par conséquent, les cas d'utilisation d'ajout sont *a priori* de priorité « haute », ceux de listage de priorité « moyenne », et ceux de retrait de priorité « basse ».

Dans la suite, nous donnons les préconditions et postconditions pour les cas d'utilisation de priorité « Haute ». Pour les autres, nous indiquons uniquement leur niveau de priorité.

Voici les préconditions et postconditions des cas d'utilisation du premier sprint :

HAUTE n°1	<ul style="list-style-type: none">- Ajouter un client<ul style="list-style-type: none">- précondition :<ul style="list-style-type: none">\wedge identifiant sur le client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ nom sur le client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ prénom sur le client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ client avec cet identifiant inexistant.- postcondition : client avec un identifiant associé existant
Moyenne	<ul style="list-style-type: none">- Afficher les caractéristiques du client
Moyenne	<ul style="list-style-type: none">- Lister les clients
basse	<ul style="list-style-type: none">- Modifier les caractéristiques du client
basse	<ul style="list-style-type: none">- Supprimer un client<ul style="list-style-type: none">- précondition:<ul style="list-style-type: none">\wedge identifiant sur le client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ nom du client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ prénom du client bien formé ($\neq \text{null} \wedge \neq \text{vide}$)$\wedge$ client avec cet identifiant existant- postcondition: client et son identifiant inexistant

HAUTE n°2

- Déposer un envoi sans adjoindre de source IoT
 - précondition :
 - \wedge identifiant de l’envoi bien formé (non null et non vide)
 - \wedge origine bien formée (non null et non vide)
 - \wedge destination bien formée (non null et non vide)
 - \wedge envoi avec cet identifiant inexistant
 - \wedge identifiant du client expéditeur bien formé (non null et non vide)
 - \wedge **identifiant client expéditeur** prénom existant
 - postcondition :
 - \wedge envoi avec cet identifiant existant et disponible pour la prise en charge
 - \wedge instant de dépôt est aujourd’hui

Moyenne

- Accepter la livraison d’un envoi
 - précondition :
 - \wedge identifiant de l’envoi bien formé (non null et non vide)
 - \wedge envoi avec cet identifiant existe
 - \wedge envoi avec cet identifiant est “en cours de livraison”
 - \wedge identifiant du jalon bien formé
 - \wedge jalon avec cet identifiant existe
 - \wedge emplacement du jalon = destination de l’envoi
 - postcondition :
 - \wedge Envoi livré
 - \wedge date de livraison ajoutée

Moyenne

- Lister les envois

Basse

- Retirer un envoi

HAUTE n°3

- Prendre en charge un envoi
 - précondition :
 - \wedge identifiant du transporteur bien formés (\neq null \wedge \neq vide)
 - \wedge identifiant transporteur existant
 - \wedge identifiant sur l’envoi bien formé (\neq null \wedge \neq vide)
 - \wedge envoi avec cet identifiant existant et disponible pour la prise en charge
 - postcondition : envoi pris en charge

Moyenne

- Lister les envois pris en charge

HAUTE n°4

- Ajouter un transporteur

- précondition :

\wedge nom sur le transporteur bien formé

\wedge périmètre de livraison du transporteur bien formé

\wedge transporteur avec ce nom inexistant

- postcondition :

\wedge transporteur et son identifiant existant

Moyenne – Lister les transporteur

HAUTE N°5

– Ajouter une condition à un envoi

– précondition :

\wedge identifiant de la condition bien formé (non null et non vide)

\wedge condition avec cet identifiant n'existe pas

\wedge identifiant de l'envoi bien formé (non null et non vide)

\wedge envoi avec cet identifiant existe

\wedge nom de l'attribut bien formé (non null et non vide)

\wedge valeur haute et valeur basse ne sont pas **toutes les deux** null

– postcondition :

\wedge condition avec cet identifiant ajouté

Moyenne - Lister les conditions d'envoi

HAUTE N°6

– Ajouter une source IoT

– précondition :

\wedge identifiant de la source bien formé (non null et non vide)

\wedge source avec cette identifiant non existant.

\wedge description de la source bien formée (non null et non vide)

– postcondition :

\wedge source IoT créée et disponible (ie source.idenvoi = null)

Moyenne – Adjoindre une source IoT à un envoi

– précondition :

\wedge identifiant de la source bien formé (non null et non vide)

\wedge identifiant de l'envoi bien formé (non null et non vide)

\wedge envoi avec cet identifiant existe.

\wedge source avec cet identifiant existe

\wedge source IoT avec cet identifiant n'est pas adjoint à un autre envoi

– postcondition :

\wedge source IoT adjointe à cet envoi

2 Préparation des tests de validation

2.1 Tables de décision des tests de validation

La fiche programme du module CSC4102 ne permettant pas de développer des tests de validation couvrant l'ensemble des cas d'utilisation de l'application, les cas d'utilisation choisis sont de priorité HAUTE.

Table de décisions Dans cette table de décisions nous noterons :

T: lorsque la condition exprimée sur la ligne est vraie,

F : lorsque la valeur de la condition est fausse,

: lorsque la valeur de la condition n'influence pas le résultat

Numéro de test	1	2	3	4	5
Identifiant du client bien formé (non null et non vide)	F			T	T
Nom du client bien formé (non null et non vide)		F			T
Prénom du client bien formé (non null et non vide)			F		T
Client avec cet identifiant inexistant				F	T
Client ajouté	F	F	F	F	T
Nombre de jeu de tests	2	2	2	1	1

TAB. 1: Cas d'utilisation « Ajouter un client »

Numéro de test	1	2	3	4	5	6	7
Envoi	Identifiant de l'envoi bien formé (non null non vide)	F		T			T
	Origine bien formé (non null non vide)		F				T
	Destination bien formée (non null non vide)			F			T
	Envoi avec cet identifiant inexistant			F			T
Client expéditeur	identifiant du client expéditeur bien formé (non null et non vide)				F	T	T
	identifiant (du client expéditeur) existant					F	T
Création acceptée		F	F	F	F	F	T
Instant de dépôt est aujourd'hui		F	F	F	F	F	T
Nombre de jeu de tests		2	2	2	1	2	1

TAB. 2: Cas d'utilisation « déposer un envoi sans adjoindre une source de données IoT »

Numéro de test		1	2	3	4	5
Transporteur	nom du transporteur bien formés ($\neq \text{null} \wedge \neq \text{vide}$)	F	T			T
	identifiant transporteur associé à ce nom existant		F			T
Envoi	code sur l'envoi bien formé ($\neq \text{null} \wedge \neq \text{vide}$)			F	T	T
	envoi avec ce code existant et chez le client				F	T
Envoi pris en charge		F	F	F	F	T
Nombre de jeu de tests		2	1	2	2	1

TAB. 3: Cas d'utilisation « Prendre en charge un envoi »

Numéro de test	1	2	3
nom sur le transporteur bien formé (non null non vide)	F	T	T
Transporteur avec ce nom inexistant		F	T
Transporteur ajouté	F	F	T
Nombre de jeu de tests	2	1	1

Tab. 4 : Cas d'utilisation « ajouter un transporteur »

Remarque : Nous avons eu besoin de la méthode “AjouterUnTransporteur” pour coder les tests de validations d’autres fonctions, mais nous n’avons pas eu le temps de coder ses tests.

Numéro de test	1	2	3	4	5
identifiant de l'envoi bien formé	F	T	T	T	T
envoi existant et “en cours de livraison”		F	T	T	T
identifiant du jalon bien formé			F	T	T
jalón avec cet identifiant existe				F	T
emplacement du jalon = destination de l'envoi					T
Envoi livré	F	F	F		T

Nombre de jeu de tests	2	1	2		1
-------------------------------	----------	----------	----------	--	----------

Tab. 5 : Cas d'utilisation « Accepter la livraison d'un envoi »

Numéro de test		1	2	3	4	5	6
Envoi	Identifiant de l'envoi bien formé (non null non vide)	F	T				T
	Envoi avec cet identifiant existe		F				T
source IoT	identifiant de la source IoT bien formé (non null non vide)			F	T	T	T
	identifiant de la source IoT existant				F	T	T
	source IoT avec cet identifiant non déjà adjointe à un envoi					F	T
Source IoT adjointe à cet envoi		F	F	F	F	F	T
Nombre de jeu de tests		2	1	2	1	1	1

Tab. 6 : “Cas d'utilisation « Adjoindre une source IoT à un envoi »

Numéro de test		1	2	3	4	5	6	7
Condition	Identifiant de la condition bien formé (non null et non vide)	F	T					T
	Condition avec cet identifiant n'existe pas		F					T
	nom de l'attribu bien formé (non null non vide)			F				T
	valeur haute et valeur basse non toutes les deux null				F			T
Envoi	Identifiant de l'envoi bien formé (non null non vide)					F	T	T
	Envoi avec cet identifiant existe						F	T
Condition créée		F	F	F	F	F	F	T
Nombre de jeu de tests		2	1	2	1	2	1	1

Tab. 6 : “Cas d'utilisation « Ajouter une condition à un envoi »

Numéro de test	1	2	3	4
identifiant source bien formé (non null non vide)	F	T		T
source avec cet identifiant inexistant		F		T
description bien formée (non null non vide)			T	T
Source ajoutée	F	F	F	T
Nombre de jeu de tests	2	1	1	1

Tab. 7 : Cas d'utilisation « ajouter une Source IoT »

3 Conception

3.1 Diagramme de classes

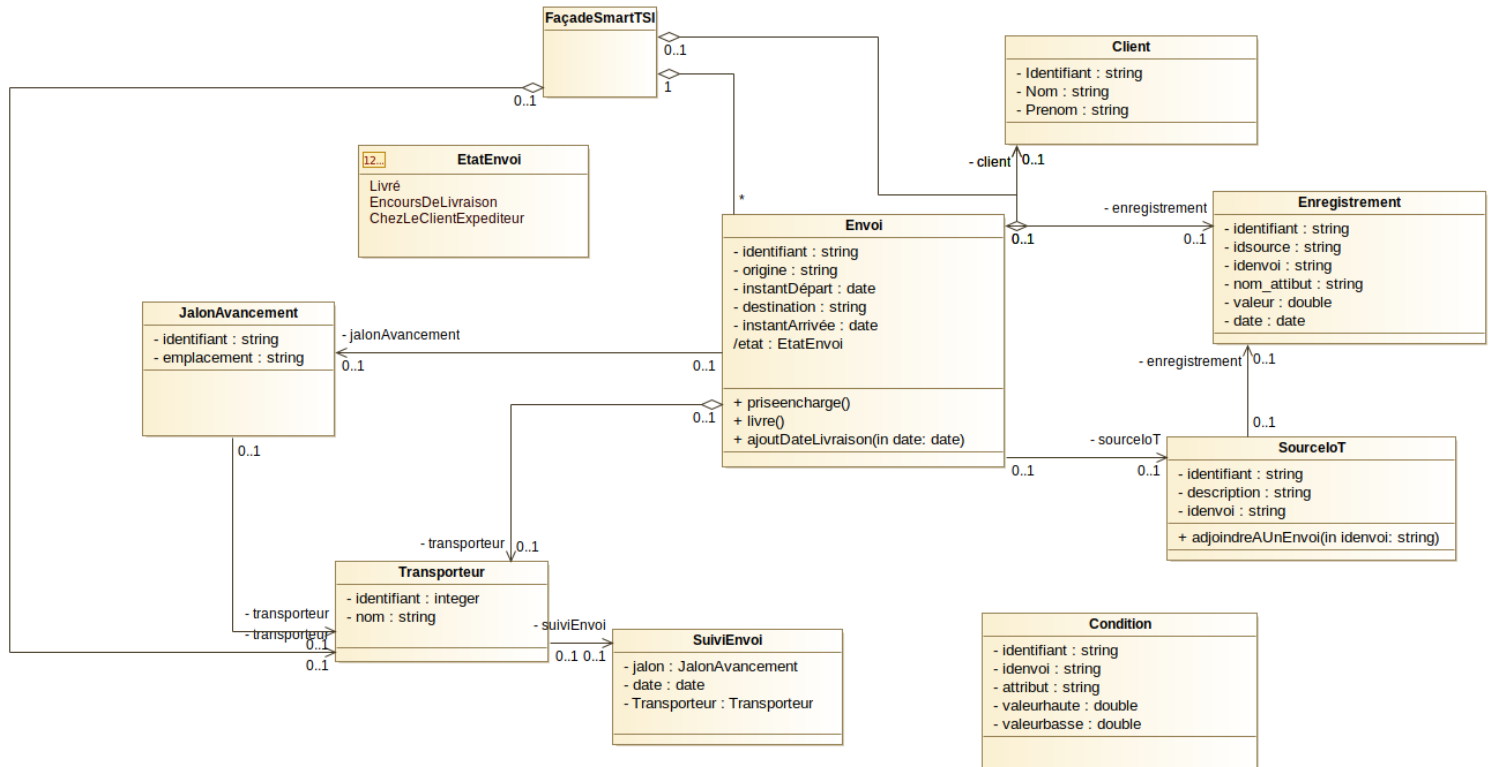


FIG. 2: Diagramme de classes.

3.2 Diagrammes de séquence

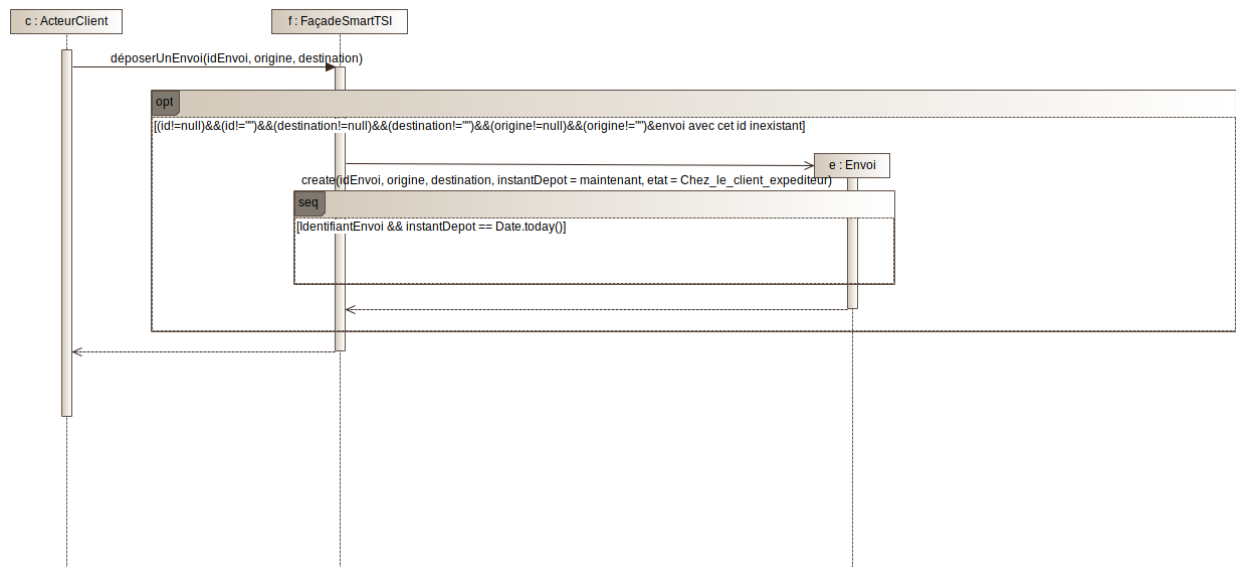


FIG. 3: Diagramme de séquence du cas d'utilisation « déposer un envoi » .

4 Diagrammes de machine à états et invariants

4.1 Classes Envoi

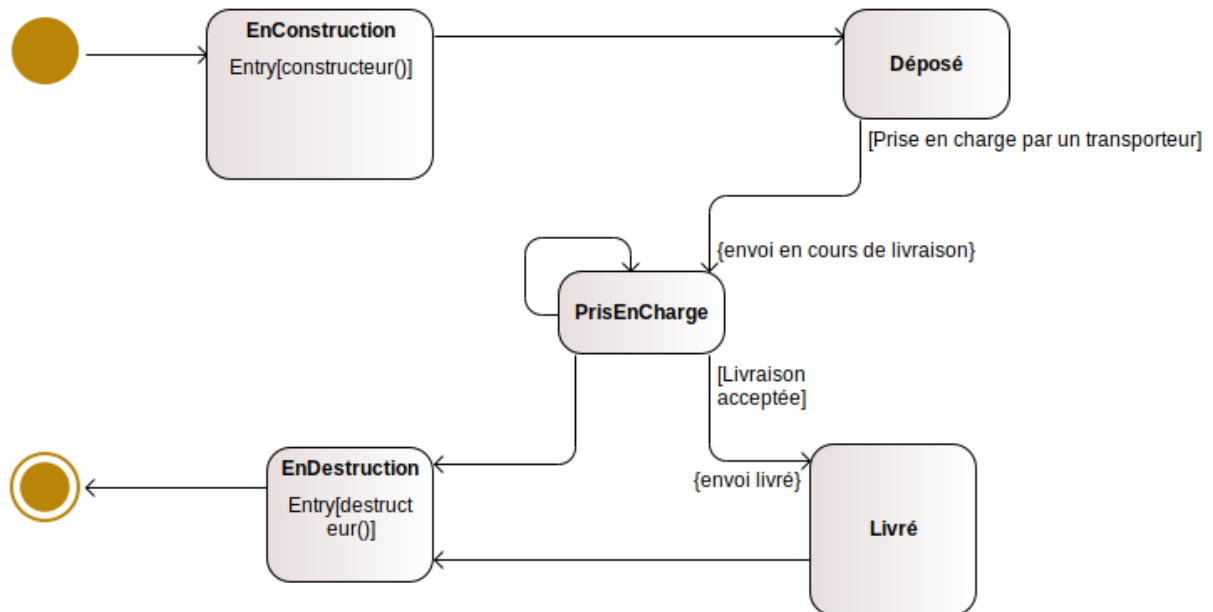


FIG. 4: Diagramme de machine à états de la classe Envoi..

L'invariant de la classe Envoi est le suivant :

$\wedge \text{identifiant} \neq \text{null} \wedge \text{identifiant} \neq \text{vide}$

$\wedge \text{origine} \neq \text{null} \wedge \text{origine} \neq \text{vide}$

$\wedge \text{instantDepot} \neq \text{null} \neq \text{DansleFutur}$

$\wedge \text{destination} \neq \text{null} \wedge \text{destination} \neq \text{vide}$

$\wedge \text{etat} \neq \text{null}$

5 Préparation des tests unitaires

5.1 Classe Envoi

Numéro de test	1	2	3	4	5	7
$\text{identifiant} \neq \text{null} \wedge \neq \text{vide}$	F	T	T	T	T	T
$\text{origine} \neq \text{null} \wedge \neq \text{vide}$		F	T	T	T	T
$\text{instantDépôt} \neq \text{null}$			F	T	T	T
instantDépôt n'est pas dans le futur				F	T	T
$\text{destination} \neq \text{null} \wedge \neq \text{vide}$					F	T
$\text{identifiant}' = \text{identifiant}$						T
$\text{origine}' = \text{origine}$						T
$\text{instantDepot}' = \text{instantDepot}$						T
$\text{destination}' = \text{destination}$						T
$\text{instantLivraison}' = \text{instantLivraison}$						T
<i>invariant</i>						T
Levée d'une exception	OUI	OUI	OUI	OUI	OUI	NON
Objet créé	F	F	F	F	F	T
Nombre de jeux de test	2	2	1	1	2	1

TAB. 1: Méthode constructeurEnvoi de la classe Envoi

5.2 Classe Client

Numéro de test	1	2	3	4
$\text{identifiant} \neq \text{null} \wedge \neq \text{vide}$	F	T	T	T
$\text{nom} \neq \text{null} \wedge \neq \text{vide}$		F	T	T
$\text{prenom} \neq \text{null} \wedge \neq \text{vide}$			F	T
$\text{identifiant}' = \text{identifiant}$				T
$\text{nom}' = \text{nom}$				T
$\text{prenom}' = \text{prenom}$				T
<i>invariant</i>				T
Levée d'une exception	OUI	OUI	OUI	NON
Objet créé	F	F	F	T
Nombre de jeux de test	2	2	2	1

TAB. 2: Méthode constructeurde la classe Client

5.3 Classe Transporteur

Numéro de test	1	2	3
$\text{identifiant} \neq \text{null} \wedge \neq \text{vide}$	F	T	T
$\text{nom} \neq \text{null} \wedge \neq \text{vide}$		F	T
$\text{nom}' = \text{nom}$			T
$\text{identifiant}' = \text{identifiant}$			T
<i>invariant</i>			T
Levée d'une exception	OUI	OUI	NON
Objet créé	F	F	T
Nombre de jeux de test	2	2	1

TAB. 3: Méthode constructeur de la classe Transporteur

5.4. Classe JalonAvancement

Numéro de test	1	2	3
$\text{identifiant} \neq \text{null} \wedge \neq \text{vide}$	F	T	T
$\text{emplacement} \neq \text{null} \wedge \neq \text{vide}$		F	T
$\text{nom}' = \text{nom}$			T
$\text{emplacement}' = \text{emplacement}$			T
<i>invariant</i>			T
Levée d'une exception	OUI	OUI	NON
Objet créé	F	F	T
Nombre de jeux de test	2	2	1

TAB. 4: Méthode constructeur de la classe JalonAvancement

5.5. Classe SourceIoT

Numéro de test	1	2	3
$\text{identifiant} \neq \text{null} \wedge \neq \text{vide}$	F		T
$\text{description} \neq \text{null} \wedge \neq \text{vide}$		F	T
$\text{identifiant}' = \text{identifiant}$			T
$\text{description}' = \text{description}$			T
$\text{idenvoi}' = \text{idenvoi}$			T
<i>invariant</i>			T
Levée d'une exception	OUI	OUI	NON
Objet créé	F	F	T
Nombre de jeux de test	2	2	1

TAB. 5: Méthode constructeur de la classe source IoT

5.6. Classe Enregistrement

Numéro de test	1	2	3	4	5	6	7	8
$identifiant \neq null \wedge \neq vide$	F							T
$idsource \neq null \wedge \neq vide$		F						T
$idenvoi \neq null \wedge \neq vide$			F					T
$nom_attribut \neq null \wedge \neq vide$				F				T
$valeur \neq null$					F			T
$date \neq null$						F		T
$date \text{ pas dans le futur}$							F	T
$identifiant' = identifiant$								T
$idsource' = idsource$								T
$idenvoi' = idenvoi$								T
$nom_attribut' = nom_attribut$								T
$valeur' = valeur$								T
$date' = date$								T
<i>invariant</i>								T
Levée d'une exception	OUI	OUI	OUI	OUI	OUI	OUI	OUI	NON
Objet créé	F	F	F	F	F	F	F	T
Nombre de jeux de test	2	2	2	2	1	1	1	1

TAB. 6: Méthode constructeur de la classe Enregistrement

5.7. Classe Condition

Numéro de test	1	2	3	4	5
$identifiant \neq null \wedge \neq vide$	F				T
$identifiant \text{ de l'envoi } \neq null \wedge \neq vide$		F			T
$nom \text{ de l'attribut } \neq null \wedge \neq vide$			F		T
$\neq (valeurhaute \text{ null } \wedge \text{ valeurbasse null})$				F	T
$identifiant' = identifiant$					T
$idenvoi' = idenvoi$					T
$attribut' = attribut$					T
$valeurhaute' = valeurhaute$					T
$valeurbasse' = valeurbasse$					T
<i>invariant</i>					T
Levée d'une exception	OUI	OUI	OUI	OUI	NON
Objet créé	F	F	F	F	T
Nombre de jeux de test	2	2	2	1	1

TAB. 7: Méthode constructeur de la classe Condition

Remarque : L'attribut valeurhaute (resp. valeurbasse) peut être null seulement si l'attribut valeurbasse (resp. valeurhaute) ne l'est pas. Dans le cas contraire, la condition n'a pas lieu d'exister.

