DOSSIER D'ARCHITECTURE LOGICIELLE DE L'APPLICATION TEMOIN MORSE

URD14/PLD/MORSE

DOSSIER D'ARCHITECTURE LOGICIELLE DE L'APPLICATION TEMOIN MORSE

URD14/PLD/MORSE

APPROBATION

	NOM	FONCTION	DATE	VISA
Etabli par :	E. BOCANFUSO	Ingénieur Logiciel		
Approuvé par :	/	Ingénieur Sécurité	/	/
Approuvé par :	/	Ingénieur Achat	/	/
Approuvé par :	J.C. DEMAGNY	Ingénieur Qualité (IQ)		
Approuvé par :	G. LABIT	Responsable de Projet (RdP)		
Autorisé par :	J.C. DERRIEN	Chef de Projet (CP)		

EVOLUTION

INDICE	DATE	NATURE DE LA MODIFICATION
01	Juillet 2006	Première version de ce document

SOMMAIRE

1. OBJECTIFS

Cette architecture logicielle a pour objectif de définir :

- Les composants logiciels de l'application témoin du projet MORSE.
- Les aspects fonctionnels de ces composants.
- Les interactions et les échanges de données entre les composants.

L'application témoin du projet MORSE est basée sur :

- La simulation d'un système de drone.
- Les échanges de données entre les composants de ce système de drone.
- L'utilisation de ce système de drones dans un environnement opérationnel.

Documents de référence :

DR[1] Spécification fonctionnelle de l'application témoin du projet MORSE ; réf. SK-0000039265-01

DR[2] Spécification des communications de l'application témoin du projet MORSE ; réf. SK-0000041444-01

2. INTRODUCTION

L'objectif du projet MORSE est de montrer qu'il est possible de vérifier de façon formelle un composant logiciel de communication asynchrone. Pour cela, on propose d'intégrer ce composant logiciel au sein d'une application témoin simulant un système de drone.

La vérification formelle est appliquée sur ce composant pour s'assurer que la solution logicielle retenue respecte les exigences exprimées dans les spécifications du système de drones.

2.1 PRESENTATION DE L'APPLICATION TEMOIN

Afin de vérifier formellement le composant de communication, il est nécessaire de découpler strictement les aspects métiers des aspects communication de l'application témoin.

Les propriétés et l'implémentation de la partie « contrôle » de ce système doivent pouvoir être respectivement vérifiées et automatiquement générées. Les spécifications exprimées doivent donc décrire précisément la manière dont les composants « métier » de l'application communiquent et interagissent entre eux.

Par contre, le découplage implique également que la communication entre les composants métiers repose entièrement sur les composants de contrôle et communication.

Plus précisément, chaque composant « métier » est constitué d'un ensemble de composants de base interagissant pour remplir une ou plusieurs fonctions « métier » de l'application témoin. Ces interactions doivent respecter un ensemble de contraintes garantissant qu'elles n'influent pas sur la partie « contrôle » de l'application :

Un composant « métier » ne doit donc pas :

- Modifier directement l'état courant d'un composant de la partie « contrôle » ou une de ces données.
- Appeler une fonction d'un composant de la partie « contrôle ».
- Communiquer directement avec un autre composant manipulé par un autre composant « contrôle ».

Un composant « contrôle » ne doit pas :

- Altérer le contenu des messages circulant entre les composants « métier ».

2.2 PRESENTATION DU SYSTEME DE DRONES

L'application témoin du projet MORSE simule un système de drones possédant les propriétés précédemment citées : distributivité et asynchronisme.

Ce système de drones est constitué de plusieurs sous-systèmes indépendants inter-agissant entre eux pour mener à bien des missions d'observation et d'attaque.

Ce système exécute ses missions dans un contexte ou les communications entre ses différents composants manquent de fiabilité. Un aspect de l'application témoin est, par exemple, de simuler le caractère non fiable des communications par de la perte d'information. Malgré cela, le système doit toujours rester stable, sous contrôle et ne pas sortir du cadre de sa mission.

2.2.1 Composition du système de drones

L'application témoin simule un système de drones S constitué des sous-systèmes suivant :

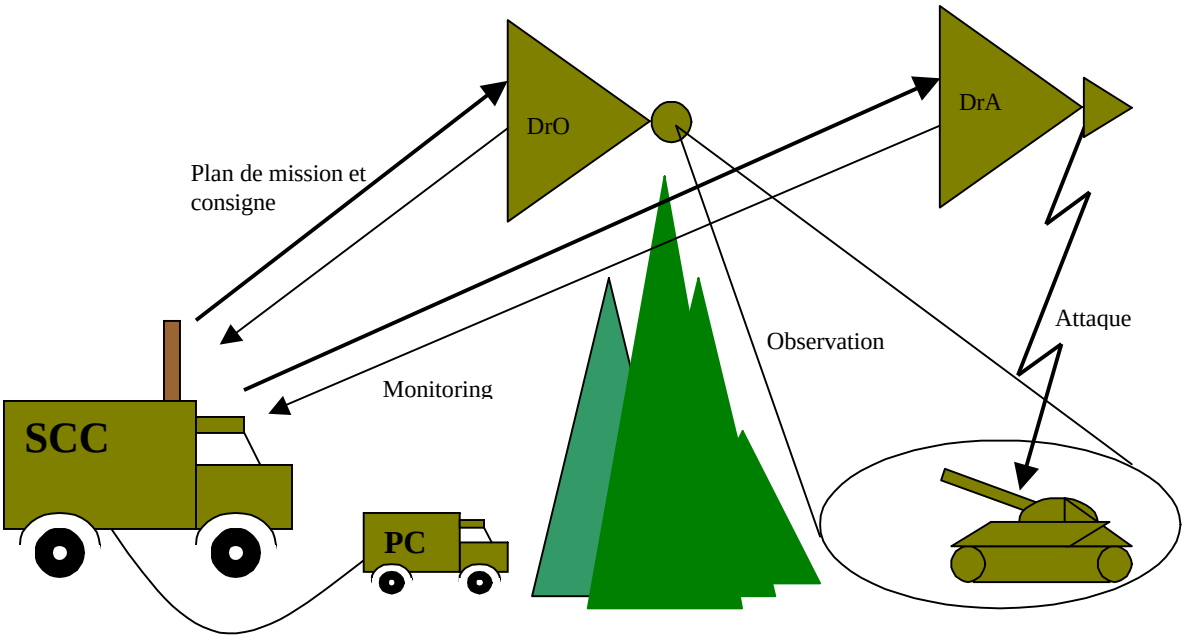
- Un poste de commandement PC.
- Deux drones Dr :
  - Un drone dédié à l'observation DrO.
  - Un drone dédié à l'attaque DrA.
- Une station sol SCC.

2.2.2 Mission du système de drones

La mission de système de drones S consiste à observer et traiter des objectifs sur un terrain (zone d'activité, champ de bataille) :

- L'observation d'un objectif est effectuée par le drone d'observation DrO.
- Le traitement (ou attaque) d'un objectif est effectué par le drone d'attaque DrA sur ordre de l'opérateur depuis la station SCC.

La mission du système de drones est programmée, c'est à dire qu'elle est définie à l'avance dans un plan de mission (PdM) établi par l'opérateur du SCC pendant une phase de préparation de mission (avant la mission proprement dite). Au cours de la mission, l'opérateur du SCC peut également envoyer une consigne, un ordre momentané et exécutable immédiatement, afin de répondre à un besoin non programmé dans le PdM.



3. DESCRIPTION DU SYSTEME DE DRONES

Un système de drones est constitué de 4 sous-systèmes communiquant entre eux de manière synchrone et asynchrone :

- Un poste de commandement (PC).
- Une station de contrôle et de communication (SCC).
- Un drone d'observation (DrO).
- Un drone d'attaque (DrA).

Le système drone d'observation (DrO) est lui-même constitué de 2 sous-systèmes :

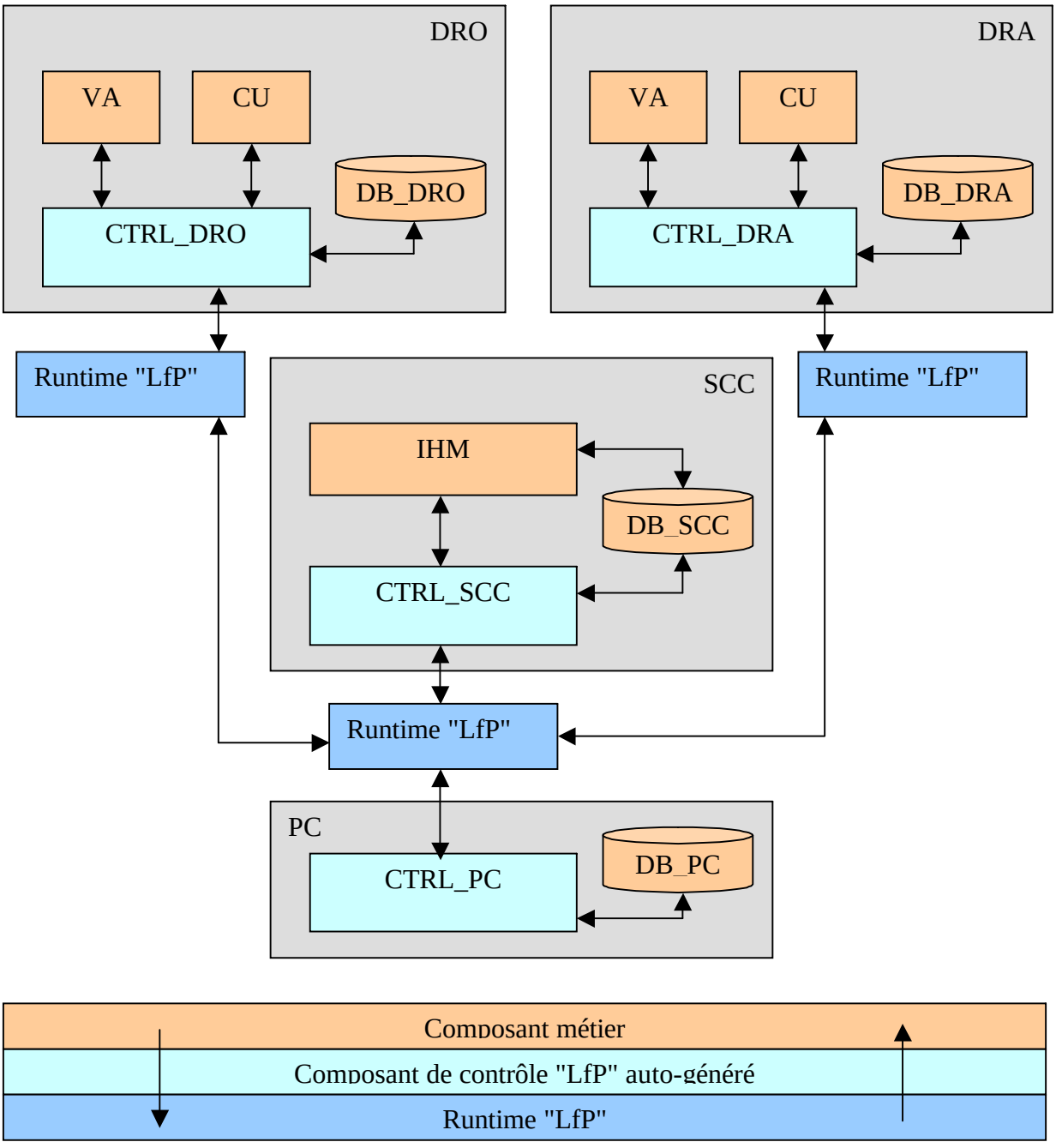
- Un véhicule aérien (VA).
- Une charge utile (CU) ou capteur.

Le système drone d'attaque (DrA) est lui-même constitué de 2 sous-systèmes :

- Un véhicule aérien (VA).
- Une charge utile (CU) ou arme.

4. ARCHITECTURE GLOBALE DU SYSTÈME DE DRONES

Les composants "métiers" de haut niveau, comme le poste de commandement PC, la station au sol SCC et les drone DRO et DRA s'exécute dans un environnement réparti. La communication entre ces composants repose sur la runtime dite "LfP". Il s'agit d'un composant logiciel développé par le partenaire AONIX. La structure d'un composant "métier" de haut niveau est formée de deux parties de composants logiciels : une partie proprement dite "métier" (VA, CU, etc.) et une partie dite "de contrôle" auto-générée (CTRL_XXX).

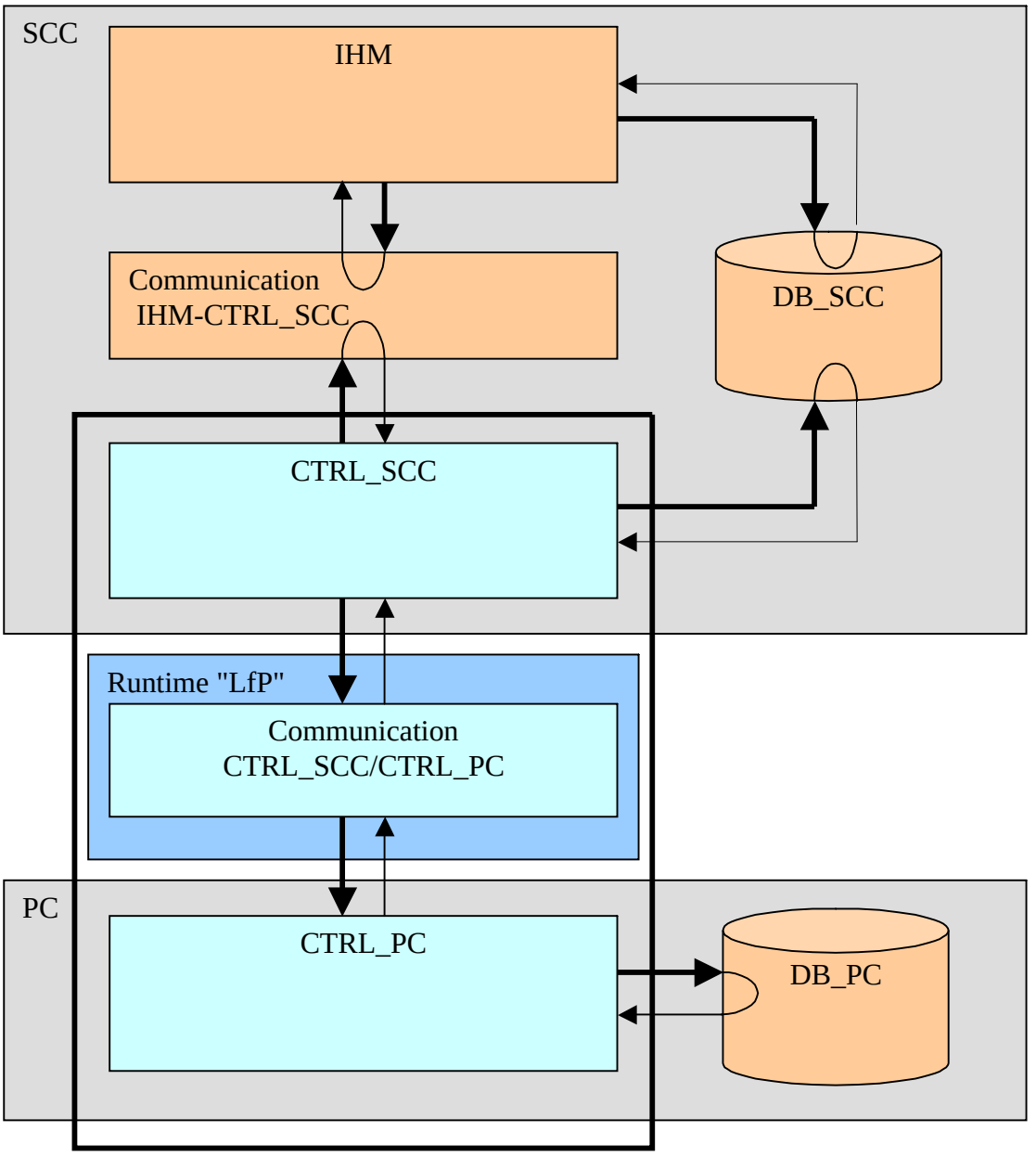


5. SOUS-SYSTÈMES SCC-PC

La simulation du système de drones repose sur deux aspects fonctionnels : la préparation et l'exécution de la mission. La préparation de la mission repose essentiellement sur le couple SCC-PC. L'exécution de la mission fait intervenir les acteurs SCC, DRO et DRA du système. Ce chapitre décrit seulement les aspects logiciels du couple SCC-PC.

5.1 DEFINITION DU PERIMETRE LFP ET DES INTERACTIONS IHM/SCC/PC

La figure ci-dessous montre deux aspects importants du fonctionnement du couple SCC-PC : les interactions IHM-CTRL_SCC et les interactions SCC-PC basées sur les concepts LfP.



Les interactions IHM-CTRL-SCC sont basées sur une communication par "boîte aux lettres" ou "file d'attente". Lorsqu'une commande IHM requière une interaction SCC-PC, le composant IHM dépose la commande dans la file d'attente et attend qu'elle soit consommée par le composant CTRL_SCC. La partie CTRL_SCC reste donc maître des actions IHM et en cela répond à l'exigence "LfP" suivante : *un composant métier ne doit pas appeler une fonction d'un composant de la partie contrôle*.

Les interactions SCC-PC repose sur la runtime "LfP" qui permet une communication à distance. Le protocole de communication qui s'appuie sur cette runtime est modélisé en UML sur la base d'un profile dit "LfP". Ce protocole agit dans un périmètre bien défini : ce périmètre englobe tous les acteurs du protocole, c'est à dire les composants de contrôle de la station et du PC, respectivement CTRL_SCC et CTRL_PC, le média "LfP" de communication CTRL_SCC/CTRL_PC. Le périmètre défini aussi la frontière qui sépare la partie "contrôle" des composants métiers. La visibilité de la partie "contrôle" sur ces composants est dite "opaque", vu de la modélisation "LfP", on parle alors de composants "opaques". En terme d'interface, il s'agit d'une vu essentiellement fonctionnelle. Le cadre préparation de mission fait intervenir les composants opaques comme les bases de données DB_SCC et DB_PC et l'interface de communication IHM_CTRL_SCC.

5.2 DIAGRAMMES DE SEQUENCE IHM/SCC/PC

Afin de bien comprendre le principe qui fonde les relations des composants SCC-PC, il a été choisi de présenter un scénario type de la préparation de mission. Ce scénario correspond à l'élaboration du dossier d'objectifs constituant une partie du dossier de mission.

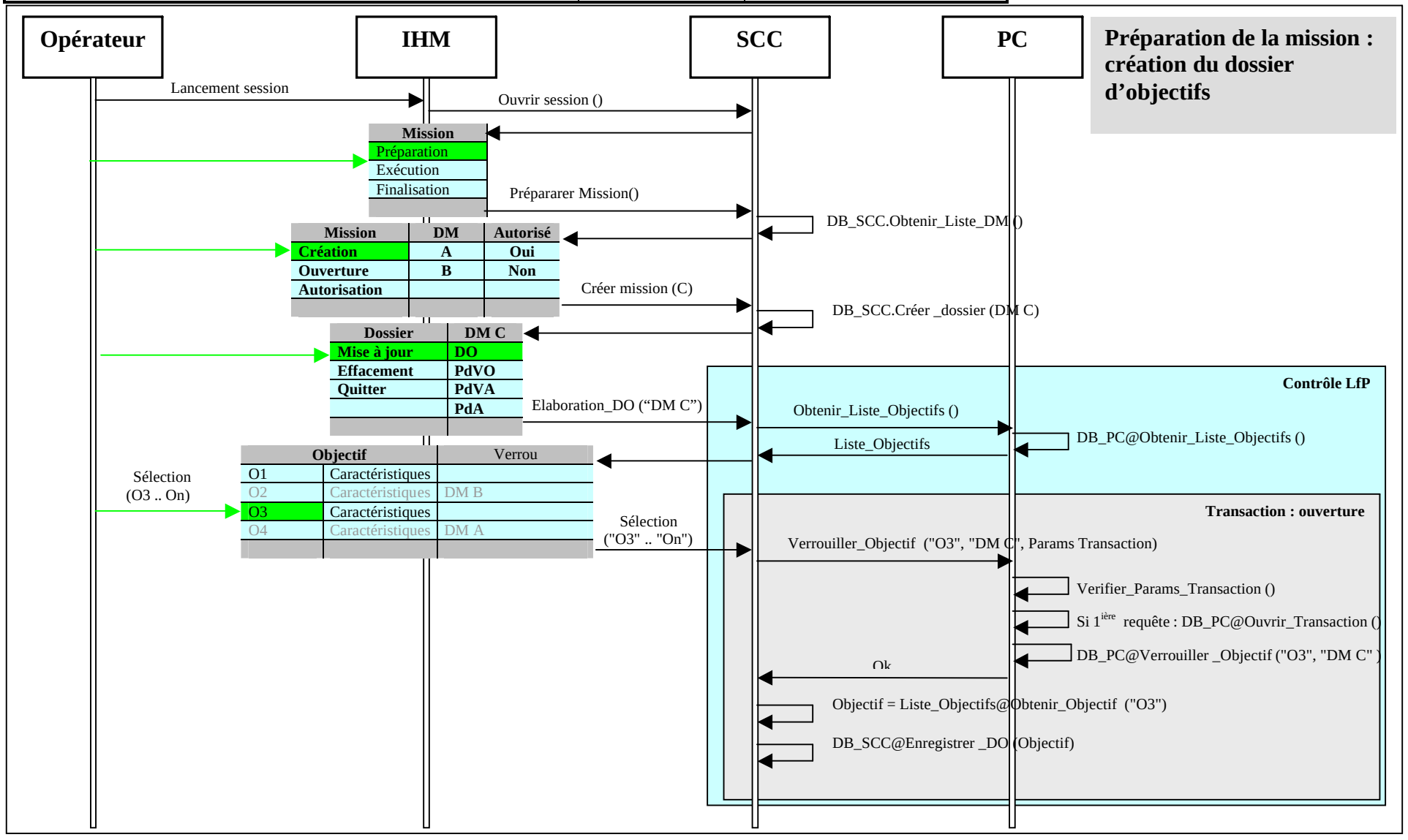
Dans ce contexte, il s'agit pour l'opérateur de pouvoir sélectionner un ensemble d'objectifs à partir d'une liste d'objectifs fournie par le PC, le PC étant la base de données de référence.

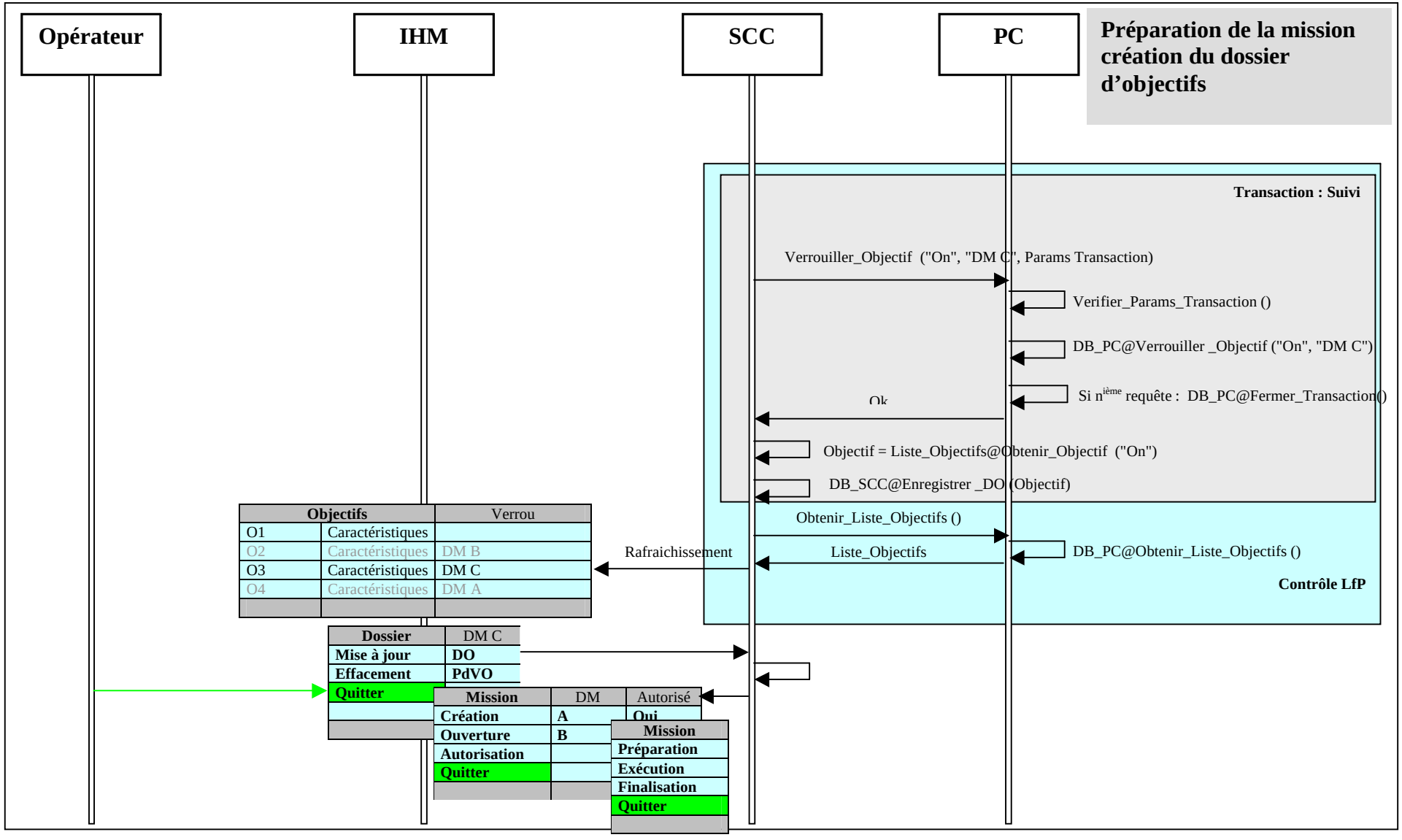
Les diagrammes de séquence mettent en relief les acteurs Opérateur, IHM, SCC et PC.

Il faut comprendre qu'ici, SCC représente l'entité logicielle qui englobe deux parties logicielles distinctes : l'implémentation des fonctionnalités IHM étrangère dans cette relation au contrôle LfP et l'implémentation des fonctionnalités sous contrôle LfP. Quand au PC, cette distinction n'est pas faite pour l'instant, dans la mesure ou le PC n'a qu'une fonction de serveur de données pré-établies.

A titre d'exemple, une action sur un composant, comme la base de donnée DB_SCC, se distingue par une syntaxe différente. En dehors du contrôle LfP, L'action demandée est invoquée par le nom de l'objet suivi la fonction et séparé par un point ex : DB_SCC.Obtenir_Liste_DM (). Sous contrôle LfP, une action similaire sera invoquée en remplaçant le point par @, il s'agit là de la vue opaque du composant DB_SCC.

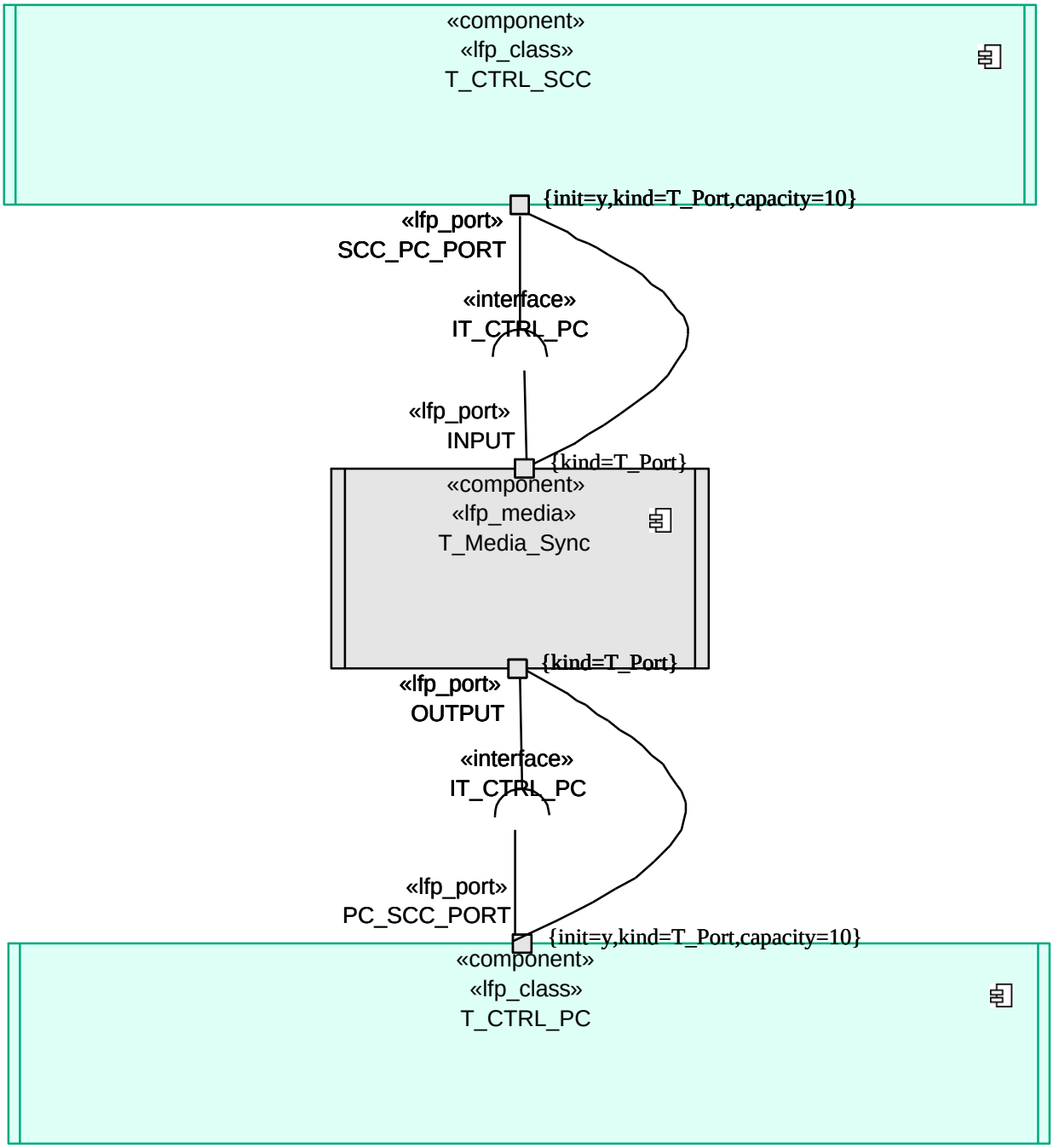
Du point de vue fonctionnel, la partie contrôle LfP supervise la sélection des objectifs et leur réservation dans la base DB_PC. Cette réservation est réalisée sous la forme d'une transaction. Une transaction est validée si et seulement si aucune erreur n'est intervenue pendant cette transaction. Les diagrammes de séquence montre le cas nominal sans erreur. Le fonctionnement de la transaction SCC-PC et PC-DB_PC est complètement décrit dans les diagrammes de comportement des classes "LfP" CTRL_SCC et CTRL_PC.





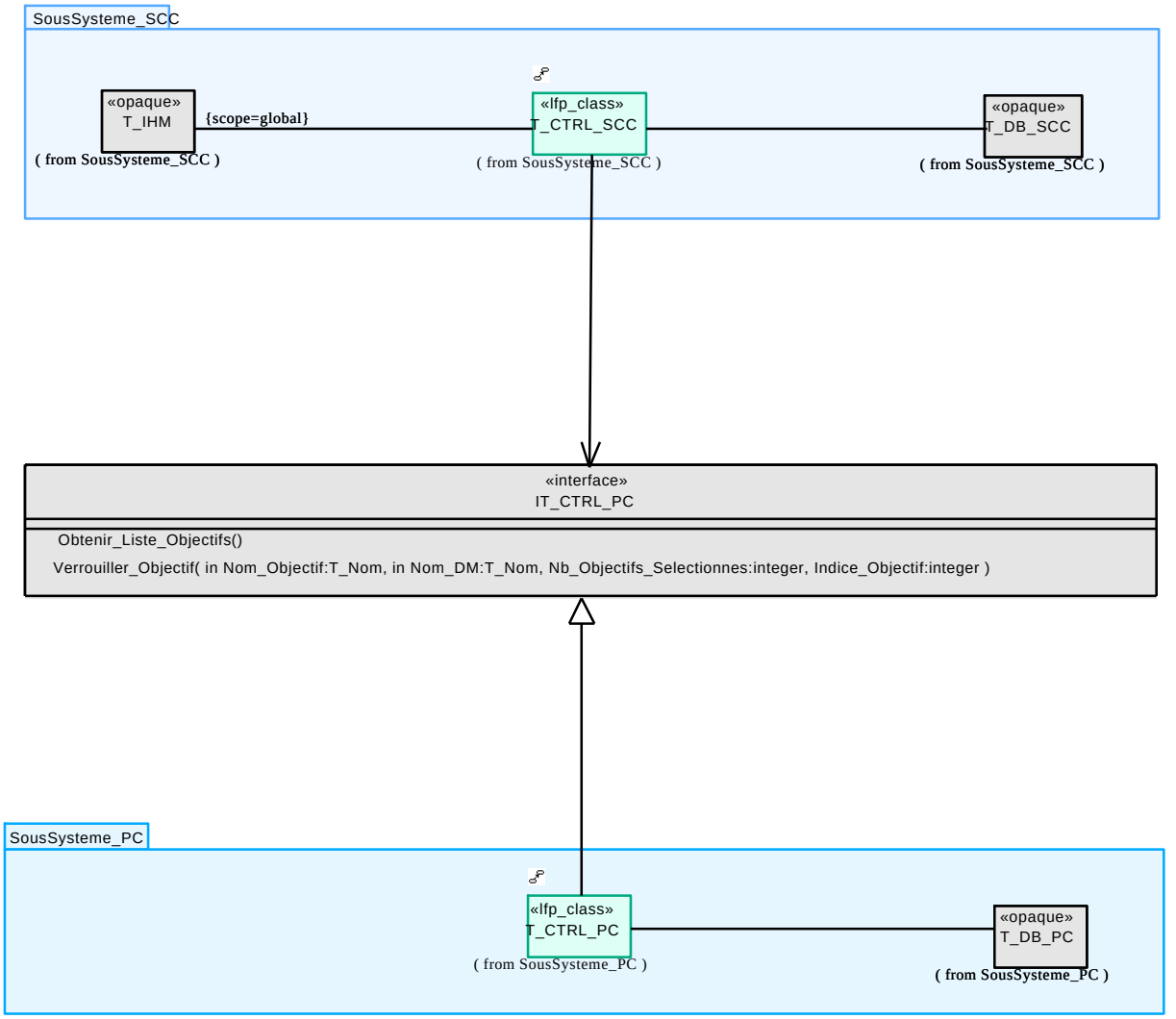
5.3 DIAGRAMME DE COMPOSANT LFP

Le diagramme suivant montre l'architecture de la partie contrôle en terme de composants "LfP". Elle est constituée des classes "LfP" T_CTRL_SCC et T_CTRL_PC et du média "LfP" sur lequel s'appuie le protocole de communication SCC-PC. Le média est du type synchrone, c'est à dire qu'un appel de méthode est bloquant pour l'appelant, en l'occurrence la classe T_CTRL_SCC.



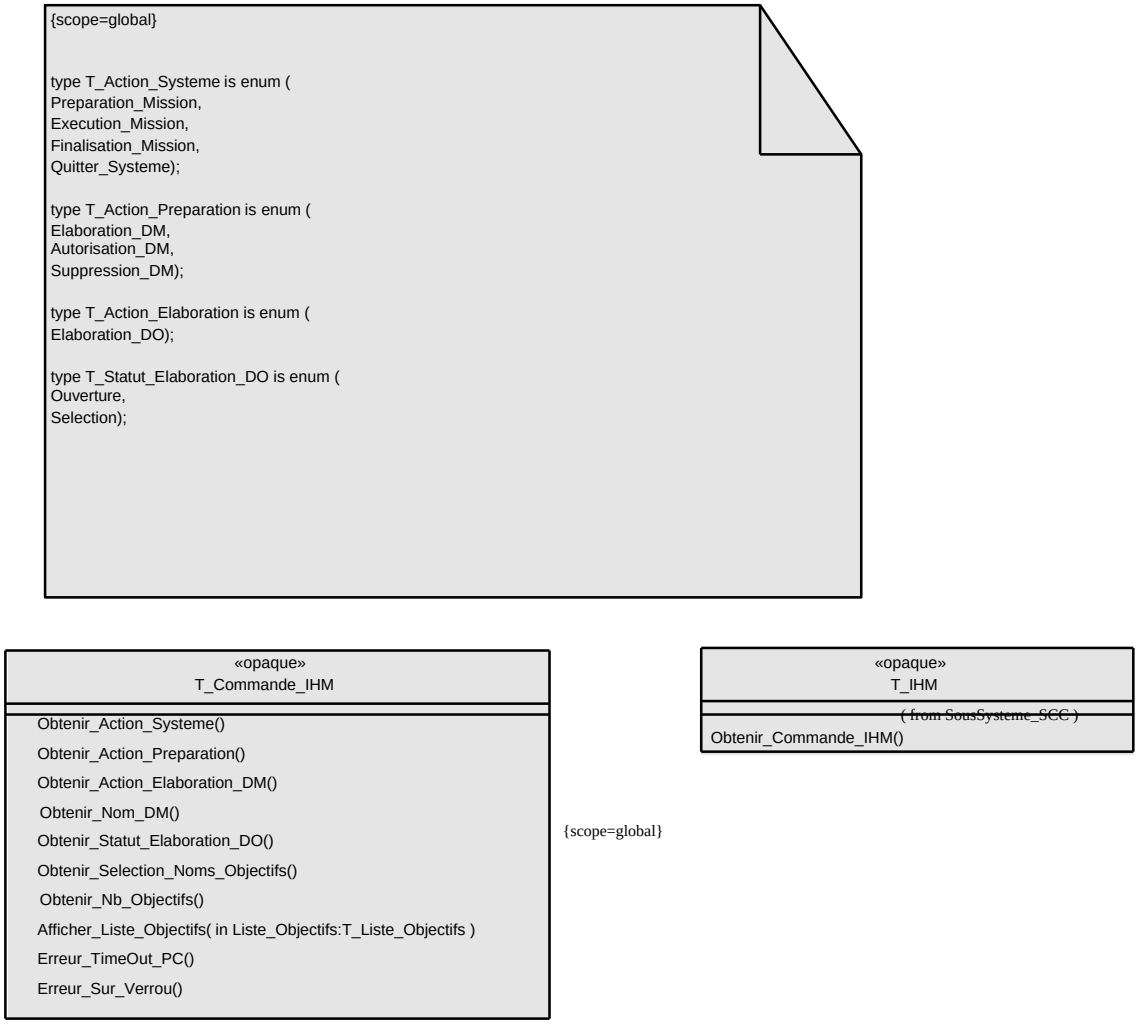
5.4 DIAGRAMME DE CLASSE UML

Cette vue de l'architecture "LfP" montre plus précisément l'interface fonctionnelle IT_CTRL_PC entre la station SCC et le PC dans le cadre de l'élaboration du dossier d'objectif. Le composant CTRL_SCC peut invoquer les deux méthodes distantes Obtenir_Liste_Objectifs et Verrouiller_Objectif du composant CTRL_PC. L'interface IT_CTRL_PC est réalisée par la classe T_CTRL_PC et utilisée par la classe T_CTRL_SCC :



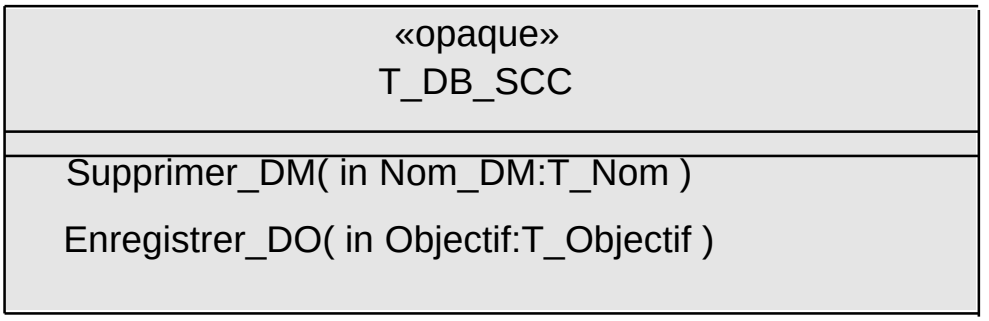
5.5 SPECIFICATION DES INTERFACES OPAQUES IHM

La figure qui suit montre les interfaces IHM :



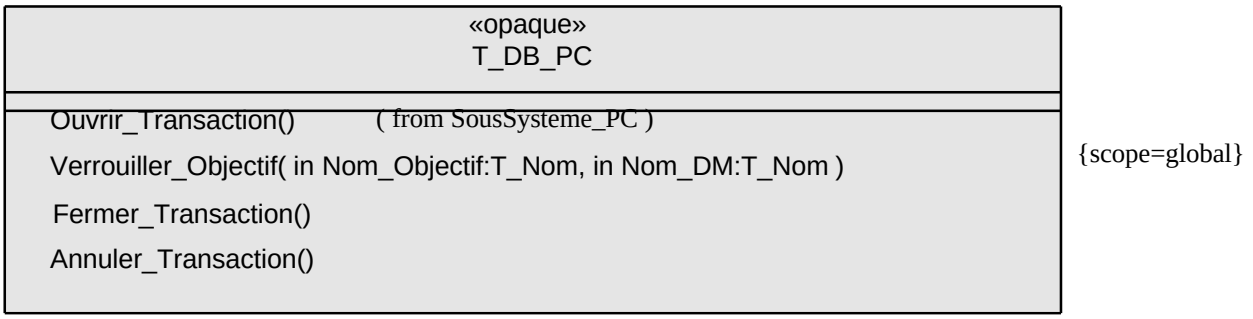
5.6 SPECIFICATION DES INTERFACES OPAQUES DB_SCC

La figure qui suit montre l'interface avec la base de données SCC :



5.7 SPECIFICATION DES INTERFACES OPAQUES DB_PC

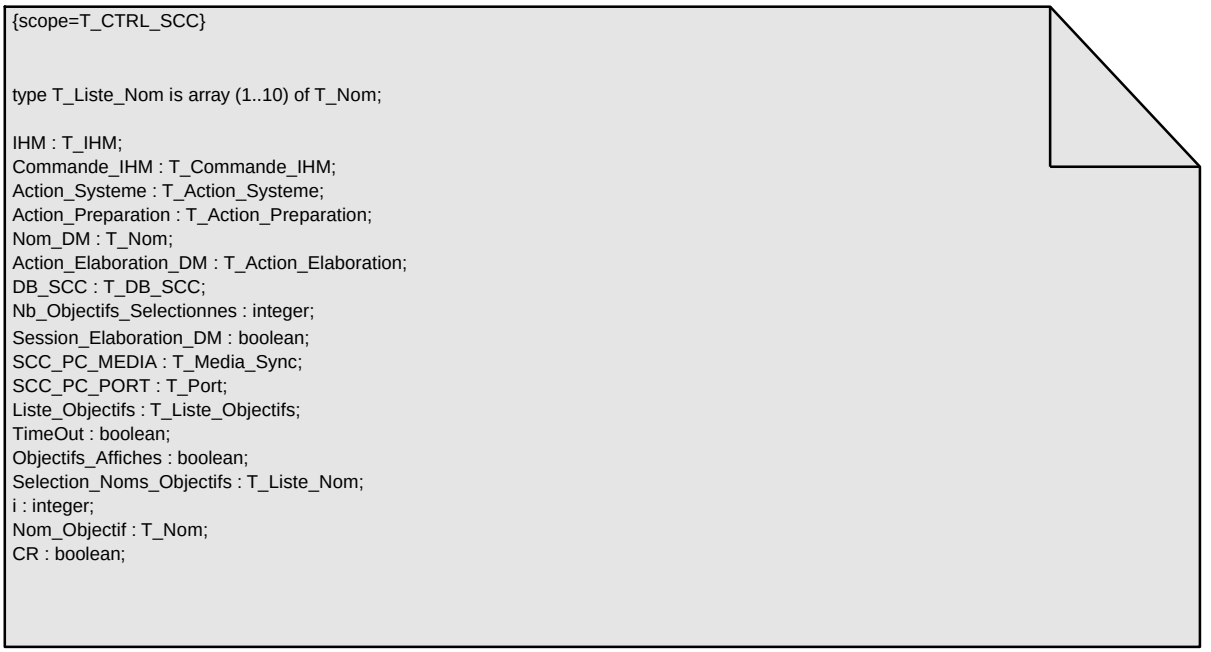
La figure qui suit montre l'interface avec la base de données DB_PC :

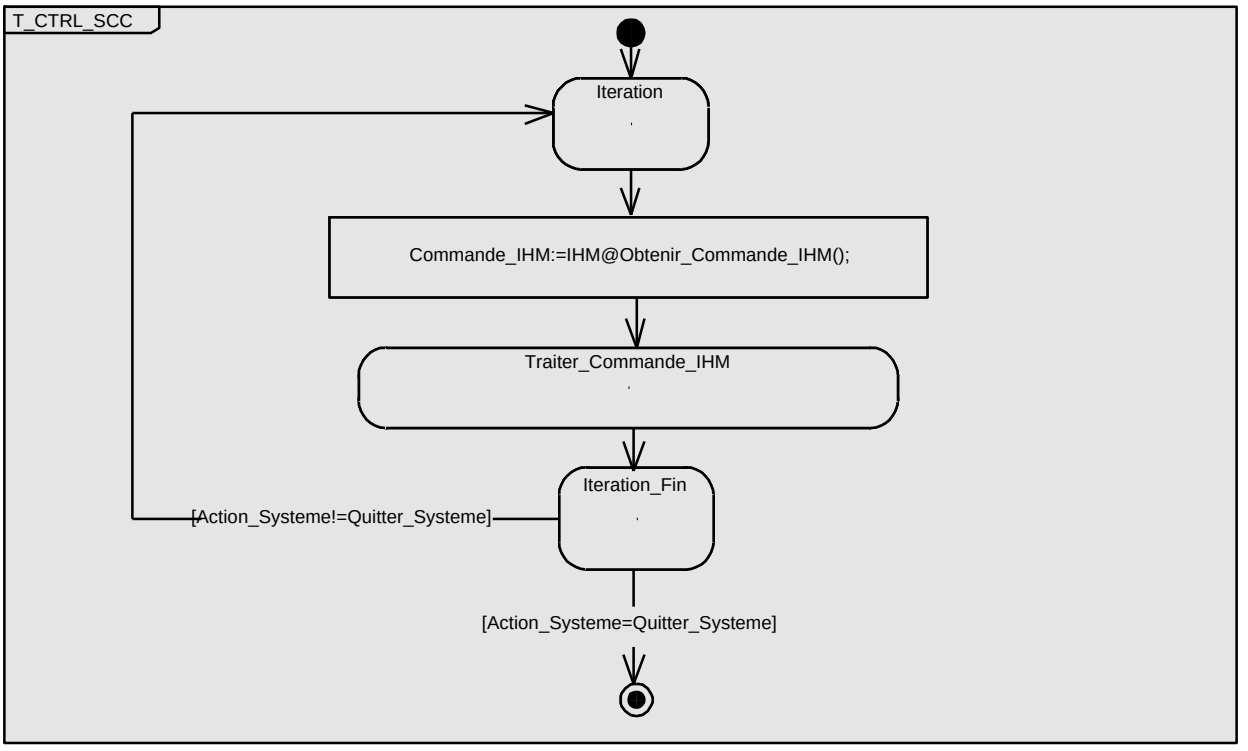


5.8 DIAGRAMME DE COMPORTEMENT LFP DU COMPOSANT CTRL_SCC

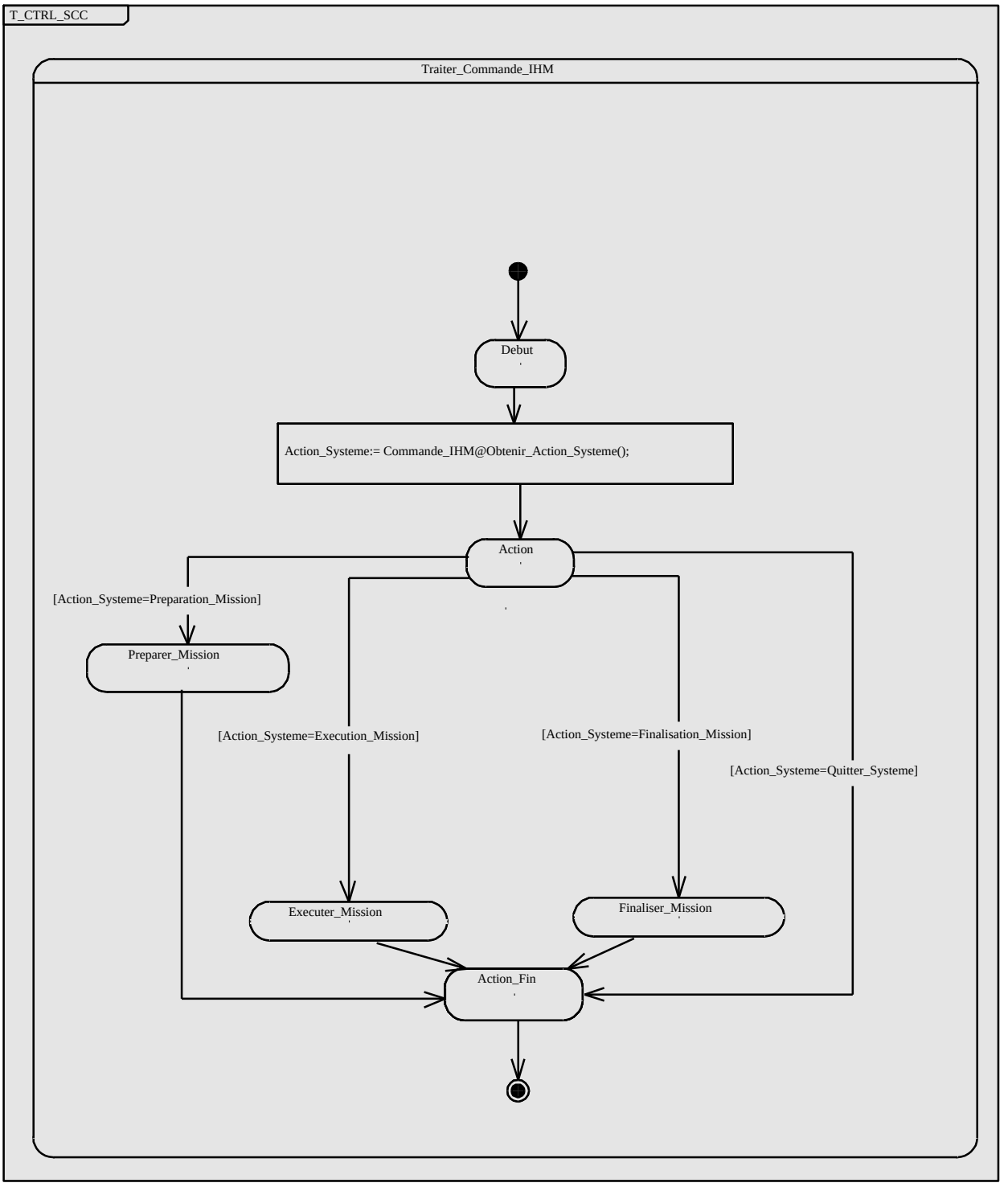
L'ensemble des diagrammes que contient ce chapitre défini le comportement détaillé de la partie contrôle SCC-PC.

5.8.1 Diagramme principal

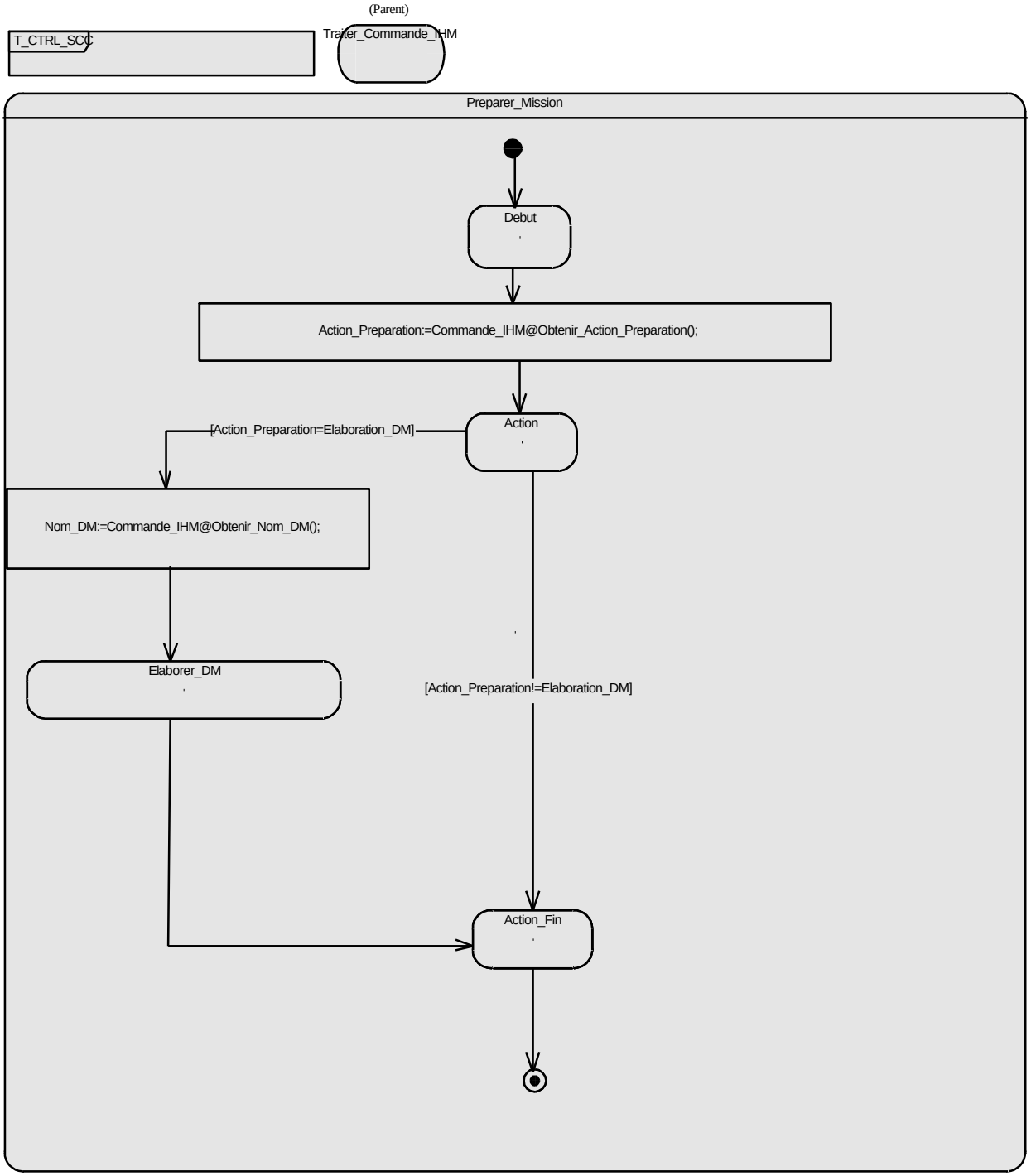




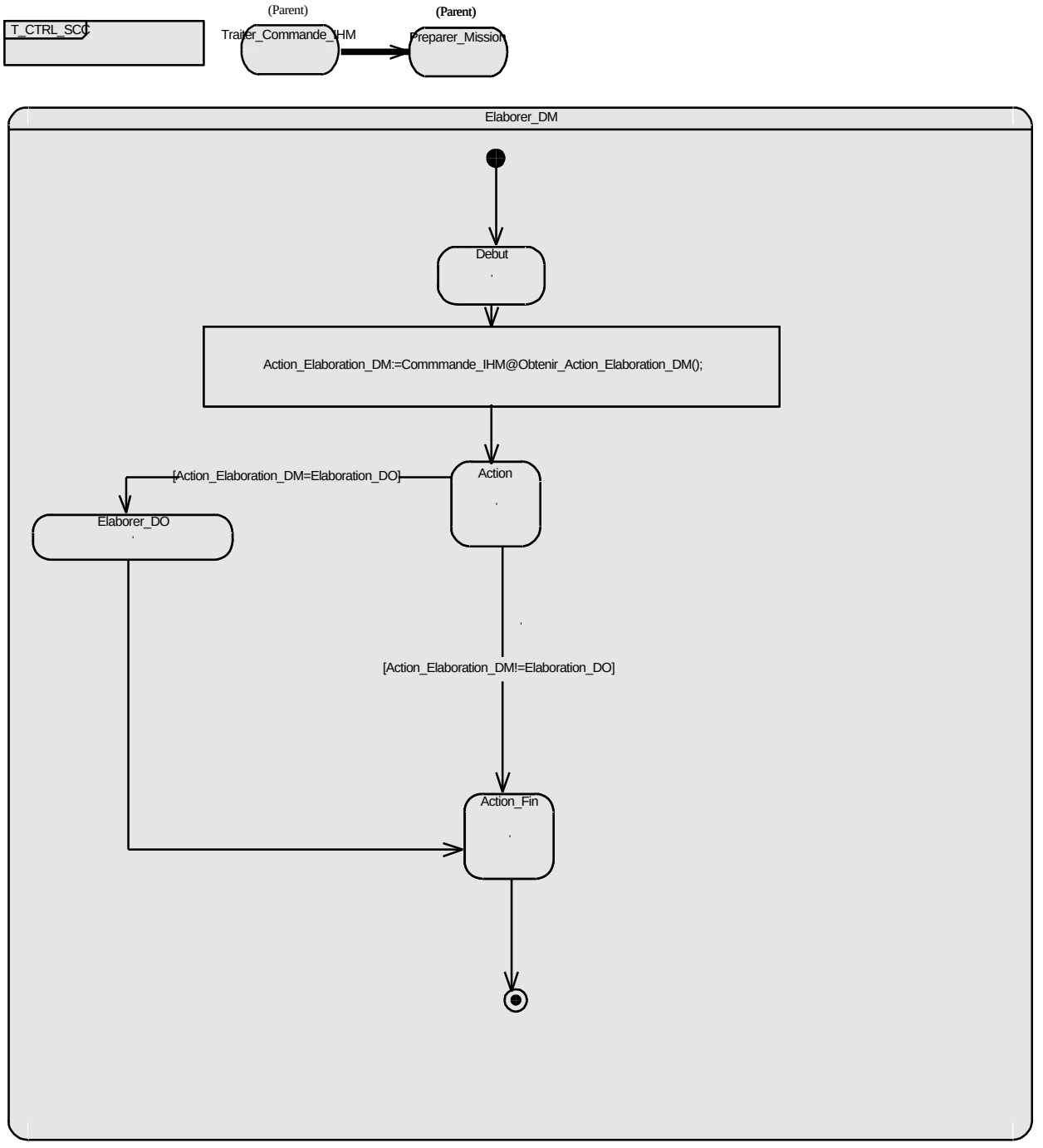
5.8.2 Décomposition : système



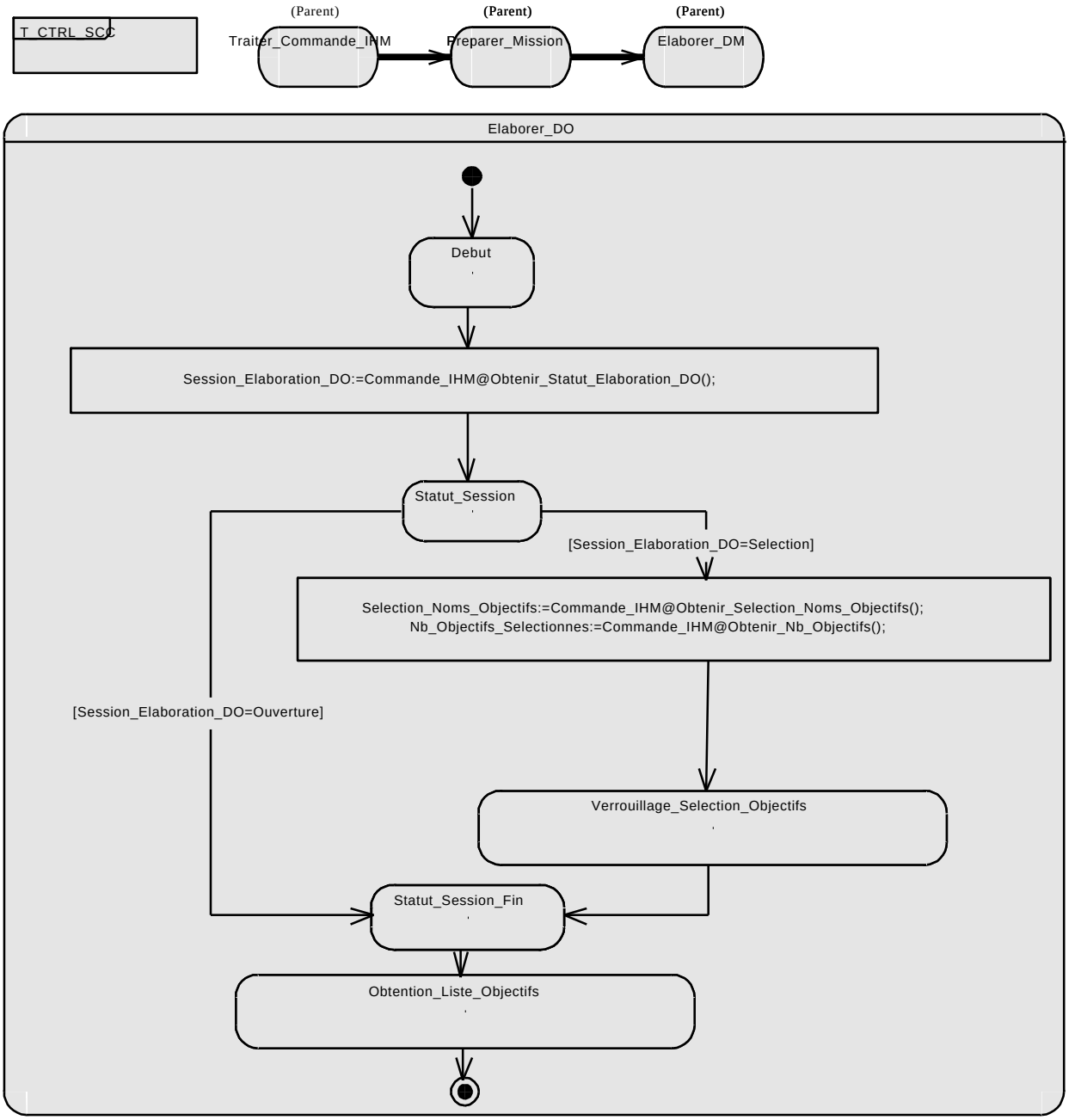
5.8.3 Décomposition : Préparation de mission



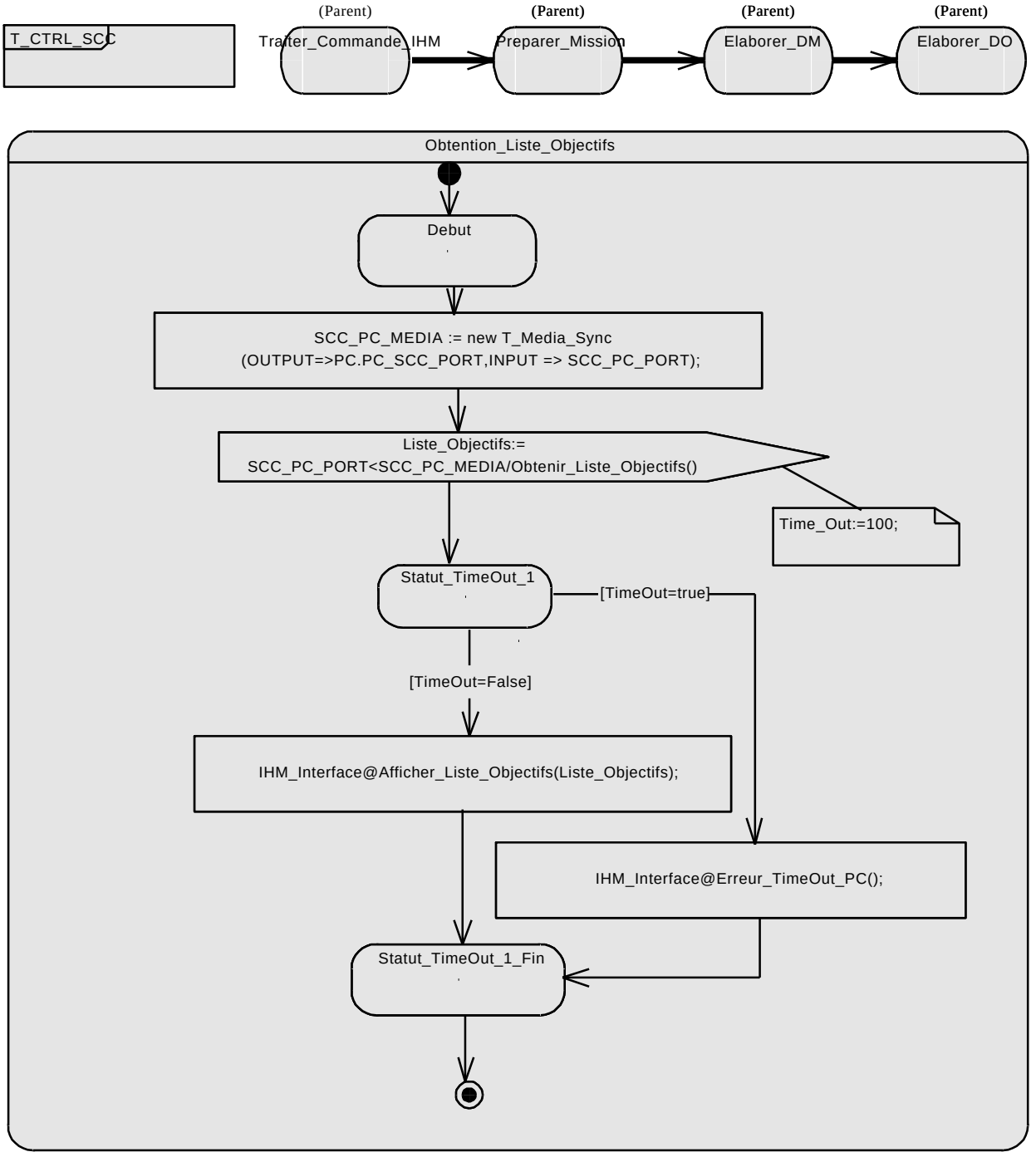
5.8.4 Décomposition : Elaboration Dossier de Mission



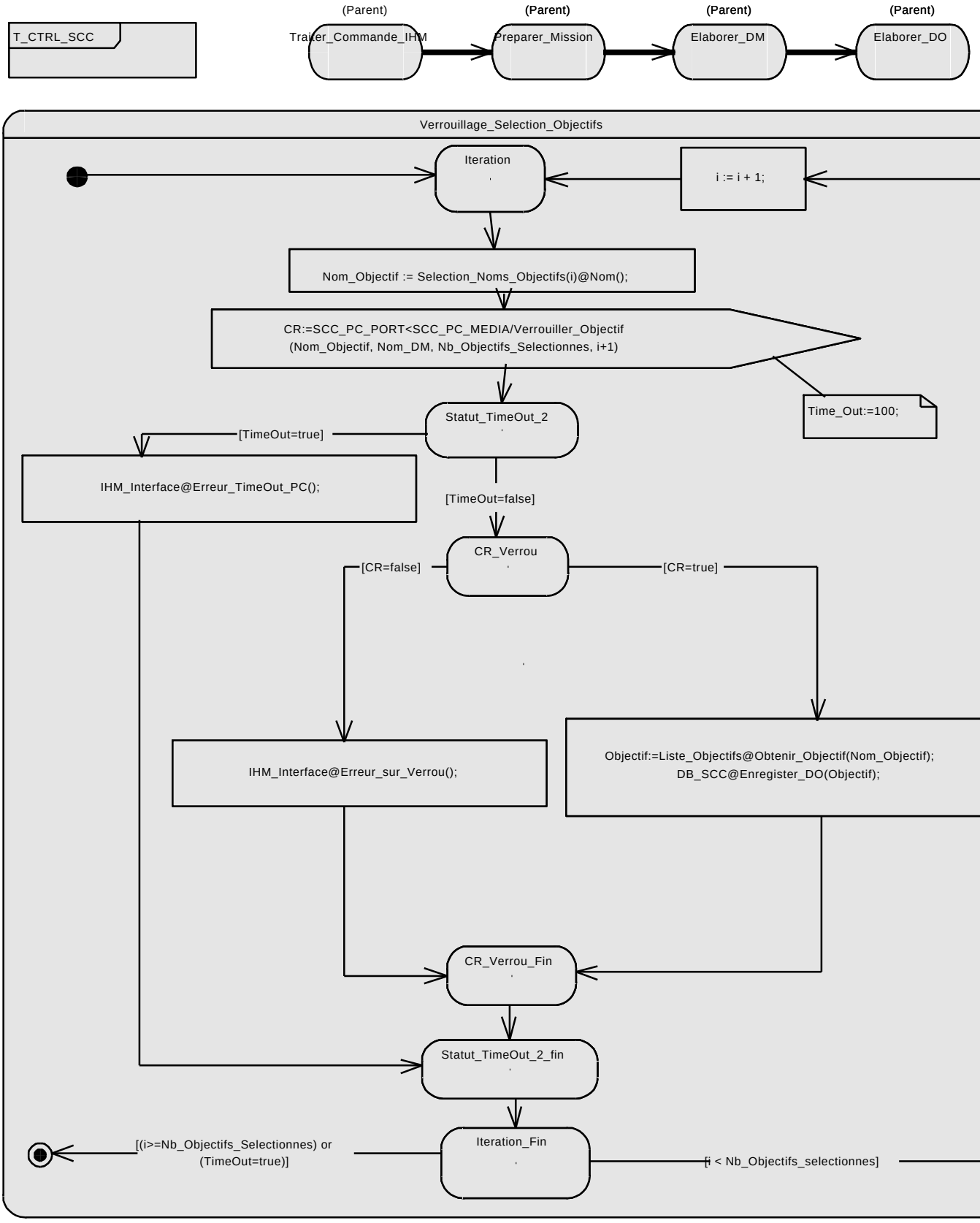
5.8.5 Décomposition : Elaboration Dossier d'Objectifs



5.8.5.1 Obtention_Liste_Objectifs

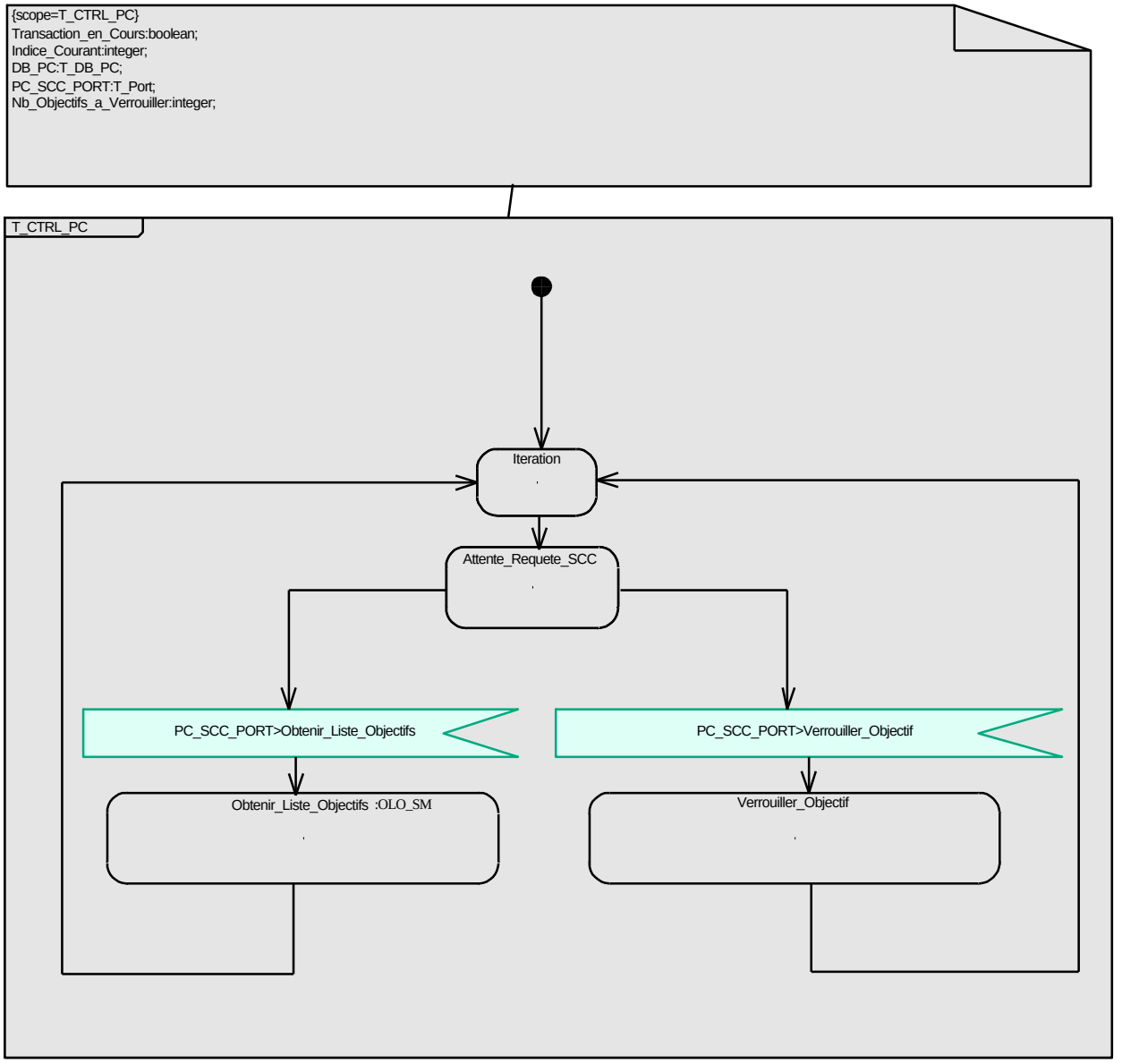


5.8.5.2 Verrouillage_Selection_Objectifs

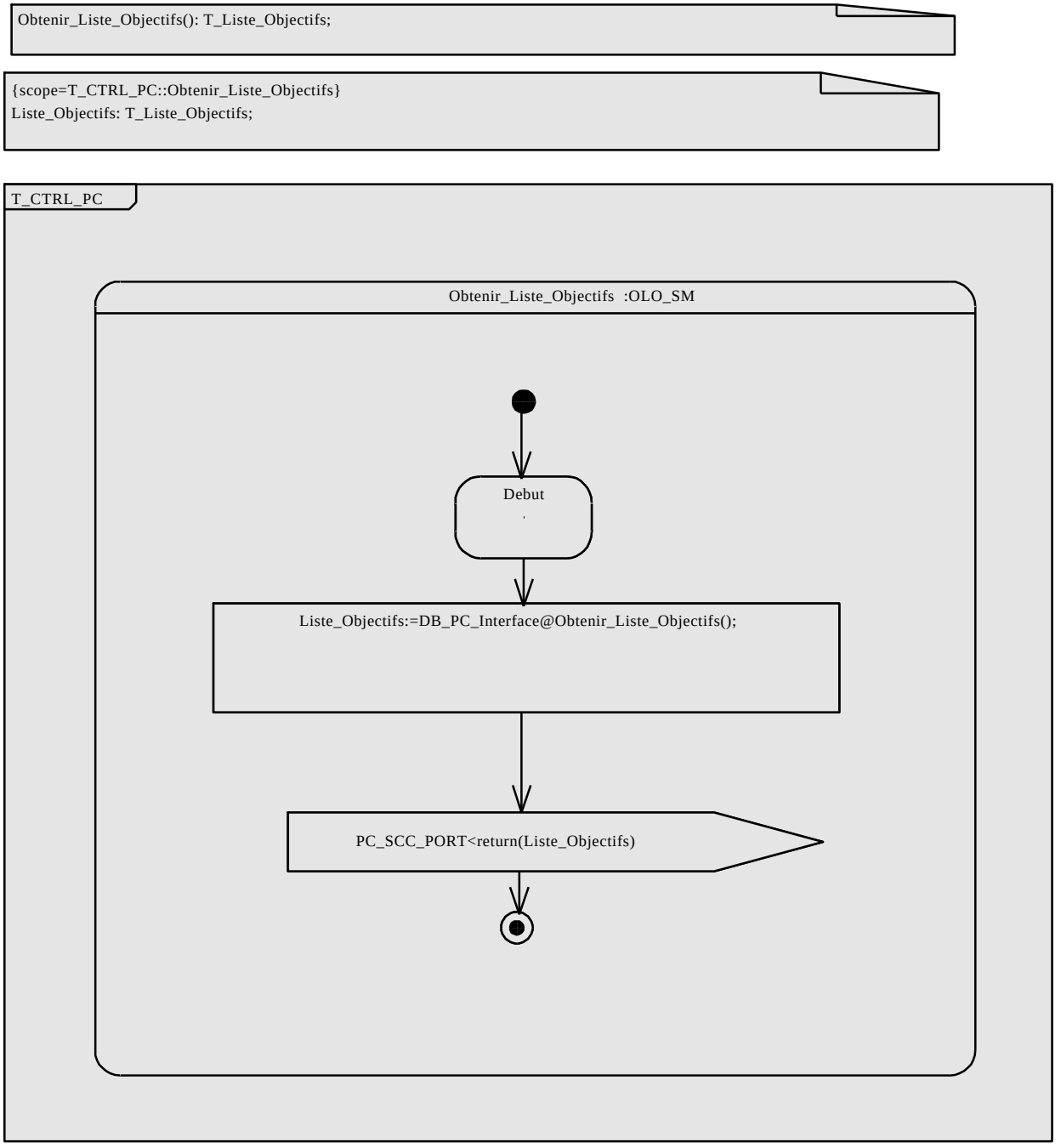


5.9 DIAGRAMME DE COMPORTEMENT LFP DU COMPOSANT CTRL_PC

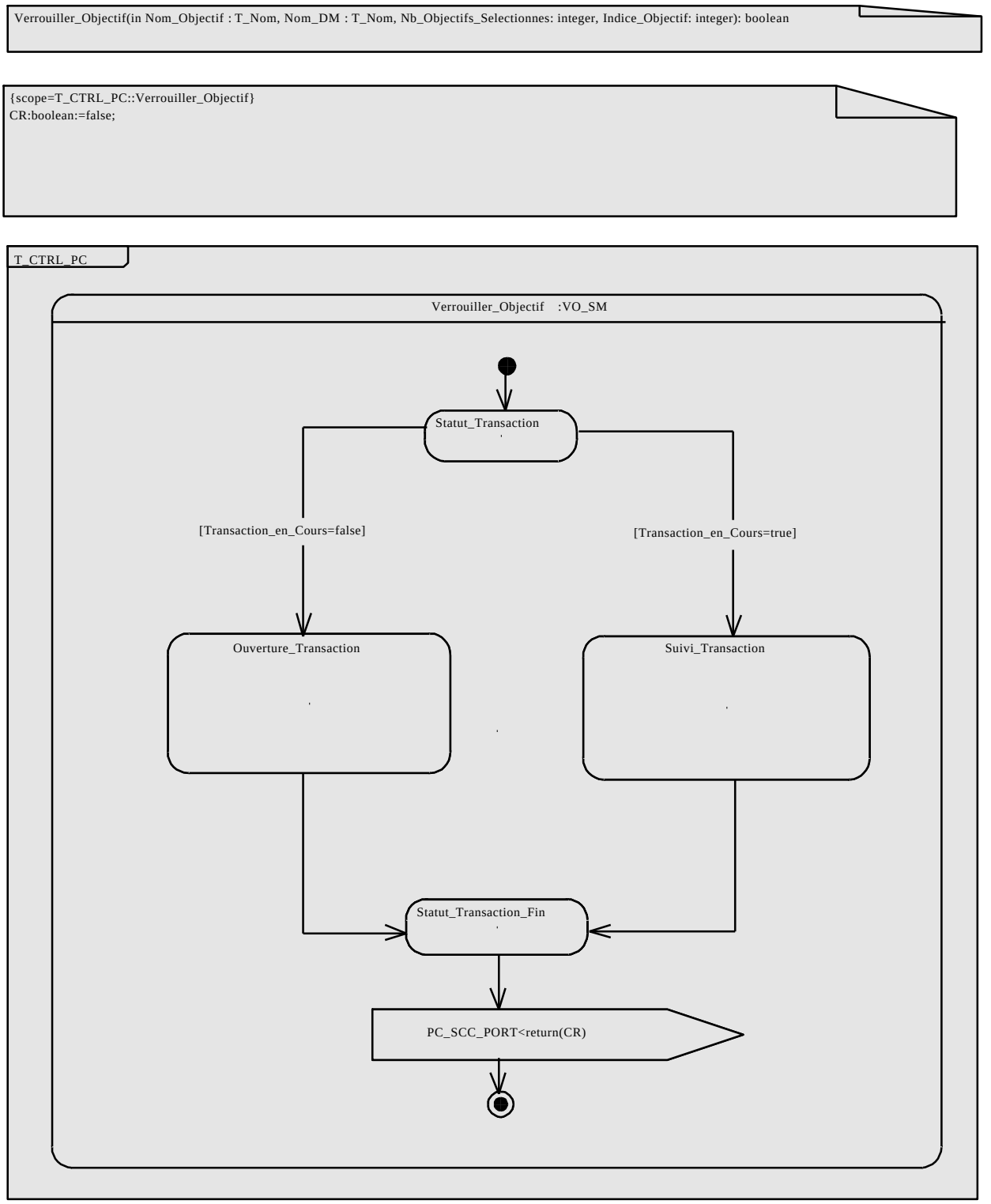
5.9.1 Diagramme principal



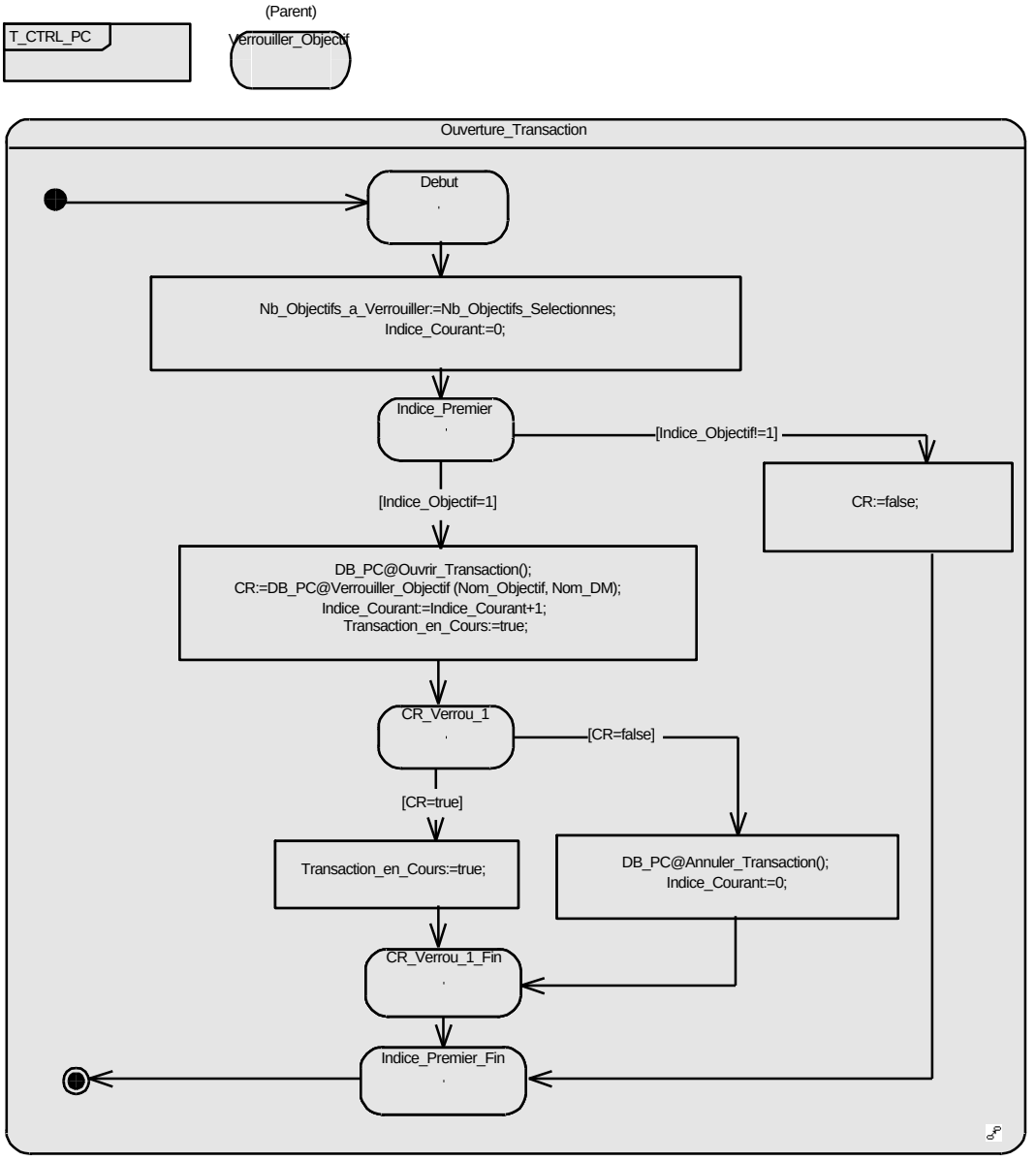
5.9.2 Diagramme de la méthode : Obtenir_Liste_Objectifs



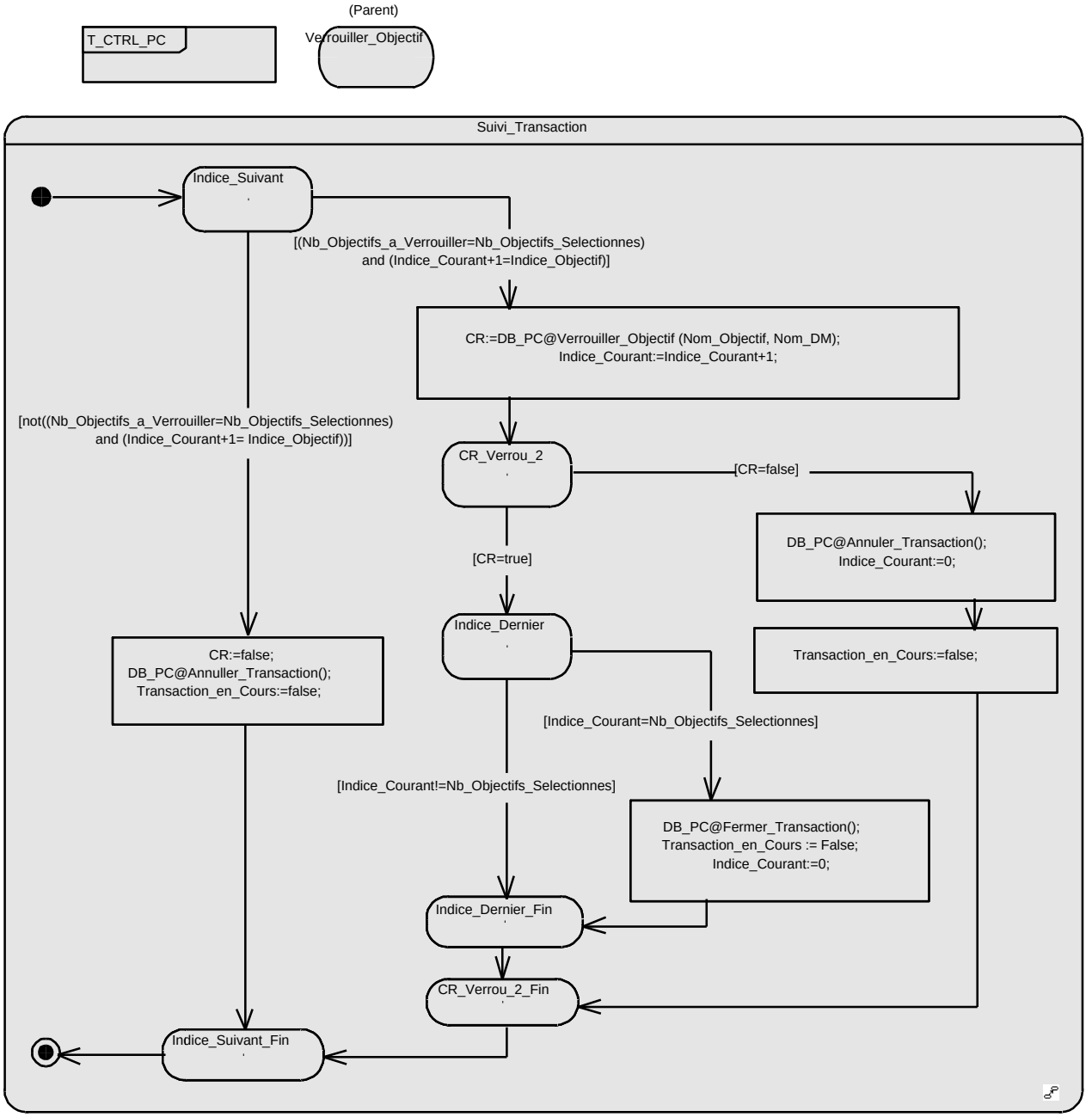
5.9.3 Diagramme de la méthode : Verrouiller_Objectif



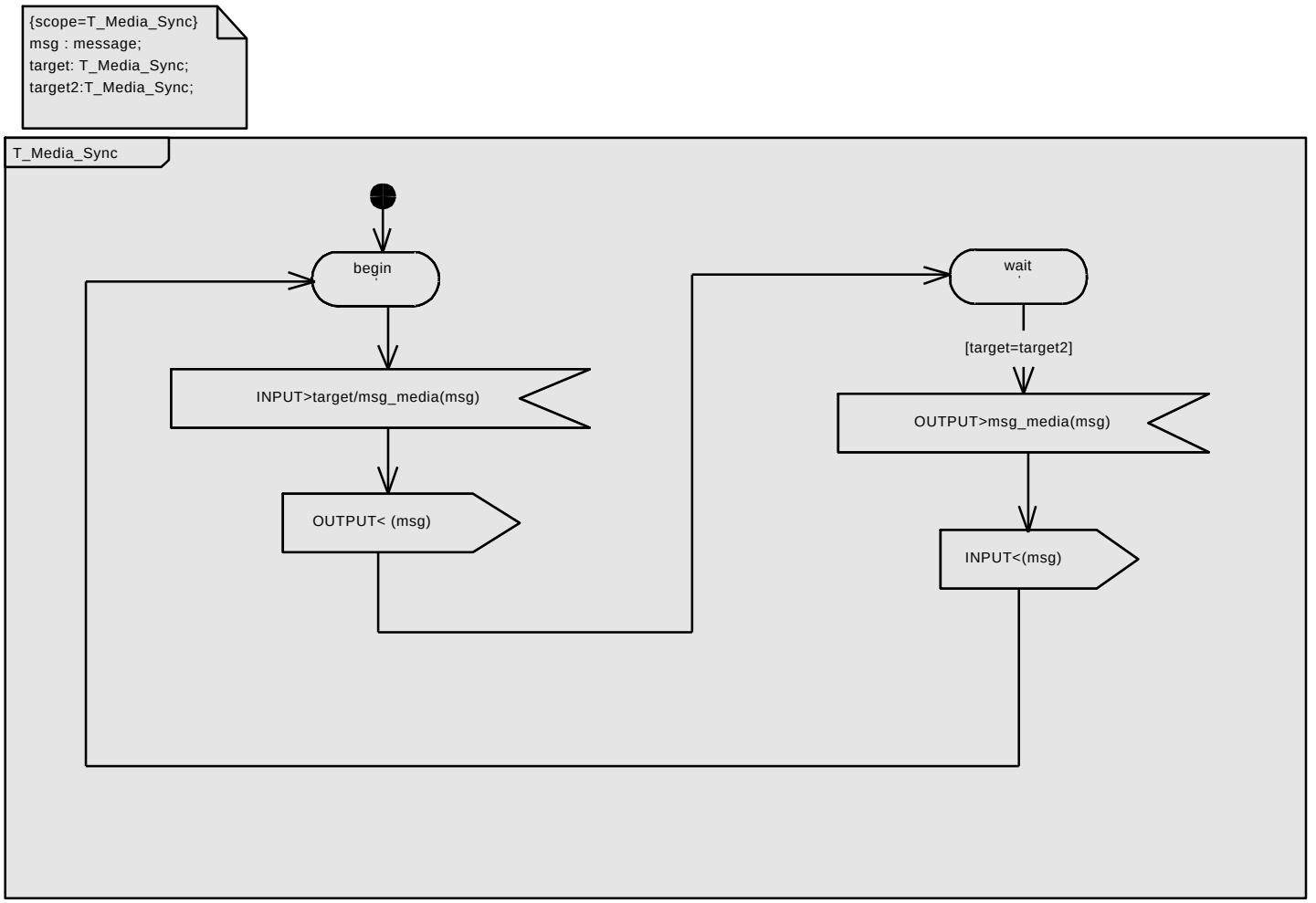
5.9.3.1 Ouverture d'une transaction



5.9.3.2 Suivi transaction



6. DIAGRAMME DE COMPORTEMENT DU MEDIA LFP SYNCHRONE



7. LEXIQUE

Système de drones		
S	:	Système de drones
PC	:	Poste de Commandement
SCC (ou GCS)	:	Station sol de Contrôle et de Communication
Dr	:	Drone
VA	:	Véhicule Aérien
CU	:	Charge Utile
DrO	:	Drone d'Observation
VA_DrO	:	Véhicule Aérien du DrO
CU_DrO	:	Charge Utile du DrO
DrA	:	Drone d'Attaque
VA_DrA	:	Véhicule Aérien du DrA
CU_DrA	:	Charge Utile du DrA
OP	:	Opérateur

Bases de données		
DB_PC	:	Base de Données du PC
DB_SCC	:	Base de Données du SCC
DB_DrO	:	Base de Données du DrO
T_DB_DrO	:	Date (ou numéro) de dernière modification de DB_DrO
DB_DrA	:	Base de Données du DrA
T_DB_DrA	:	Date (ou numéro) de dernière modification de DB_DrA

Données de mission		
PdM	:	Plan de Mission
DO	:	Dossier d'Objectifs
DM	:	Dossier de Mission
DH	:	Dossier Historique
DH_SCC	:	Dossier Historique du SCC
DH_DrO	:	Dossier Historique du DrO
DH_DrA	:	Dossier Historique du DrA
PdV	:	Plan de Vol
PdVO	:	Plan de Vol du DrO
PdVA	:	Plan de Vol du DrA
PdO	:	Plan d'Observation

Données de contrôle

CPdV	:	Consigne de Plan de Vol
CPdVO	:	Consigne de Plan de Vol d'Observation
CPdVA	:	Consigne de Plan de Vol d'Attaque
CPdO	:	Consigne de Plan d'Observation
CdV	:	Consigne de Vol
CdO	:	Consigne d'Observation
CdA	:	Consigne d'Attaque

Données de monitoring

Ve	:	Vecteur d'état
Ve_VA	:	Vecteur d'état du VA
Ve_CU	:	Vecteur d'état de la CU
VeO	:	Vecteur d'état consolidé du DrO (composé du VeO_VA et du VeO_CU)
VeO_VA	:	Vecteur d'état du VA du DrO
VeO_CU	:	Vecteur d'état de la CU du DrO
VeA	:	Vecteur d'état consolidé du DrA (composé du VeA_VA et du VeA_CU)
VeA_VA	:	Vecteur d'état du VA du DrA
VeA_CU	:	Vecteur d'état de la CU du DrA

Autres

LfP	:	Langage pour la preuve formelle