

LIVRABLE L2.4-3

Architecture générique d'une Plateforme de Simulation Outillée (APSO)

Version : 0.3

Date de version : 11/12/2021

Numéro Chrono : ISX-SAM-LIV-1345



Opération réalisée avec le concours
des Investissements d'avenir de
l'Etat confiés à l'ADEME

Informations du document

Périmètre de diffusion : Consortium / Administration

Type : Final

Date prévue de livraison : 15/12/2021

Statut : Pour révision Bureau / Validé

Auteurs :

Resp. du livrable	Organisation	Rôle dans le projet
Marc PIACENTINI	IRT SystemX	Coordinateur et contributeur de tâche 2.4
Loutfi BELAARIBI	IRT SystemX	Contributeur de tâche 2.4
Relecteurs	Organisation	
Vincent HONNET	IRT SystemX	Responsable du lot 2 du projet SAM
Mohamed TLIG	IRT SystemX	Responsable de tâche 2.4

Table de révision :

Version	Date	Contenu de la modification
0.1	09/11/2021	Construction Draft
0.2	26/11/2021	Description détaillée de l'architecture
0.3	07/12/2021	Navette autonome et référence

Table des matières

Table des matières	3
Table des illustrations et tableaux	4
Glossaire	5
Résumé exécutif	6
Introduction	7
1. Contexte	8
1.1. Outils de simulation	8
1.2. Plateforme générique	9
1.2.1. Scenario Manager	11
1.2.2. Système de simulation	12
1.2.3. Post-traitement	12
2. Architecture générique	13
2.1. Description Générale	13
2.2. Description des interfaces	15
2.2.1. Entrées de l'utilisateur	15
2.2.2. Outil de simulation	16
3. Plateforme Navette autonome (SystemX)	17
3.1. Outil de simulation numérique (OSN)	17
3.2. Modèles et interfaces	18
3.2.1. Entrées de l'OSN	18
3.2.2. Modèles de l'OSN	20
3.2.3. Chauffeur autonome (Navette SystemX)	21
3.2.4. Traitement des données	22
Conclusion	23
Références	24

Table des illustrations et tableaux

Figure 1 : Plateforme de simulation à base de scénarios	10
Figure 2 : Schéma du système de simulation générique.....	14
Figure 3 : Architecture de la plateforme Navette autonome (IRT SystemX)	19

Glossaire

AEB	Autonomous Emergency Braking (Freinage d'urgence autonome)
API	Application Programming Interface
ASAM	Association for Standardization of Automation and Measuring Systems
FMI	Functional Mock-up Interface (Interface de maquette fonctionnelle)
OSI	Open Simulation Interface
POC	Proof-Of-Concept (Preuve de concept)
STPA	Systèmes de Transports Publics Automatisés
VP	Véhicule privé

Résumé exécutif

Ce livrable du projet est attaché à la tâche 2.4 du projet SAM et a pour but de présenter une architecture générique de plateforme de simulation. La tâche 2.4 a pour objectif de concevoir une méthodologie de simulation à base de scénarios pour la validation du véhicule autonome.

Les étapes de cette méthodologie sont les suivantes :

1. Sélection des cas d'usage
2. Création des scénarios de simulation
3. Simulations numériques des scénarios
4. Comparaison des résultats avec les essais physiques équivalents
5. Validation du système grâce aux critères de sécurités

Les différentes étapes seront présentées dans le livrable 2.4-2 développant la méthodologie. Le document ici présent développe la troisième partie de la méthodologie en proposant une architecture de plateforme générique afin de permettre à un utilisateur de la méthodologie de construire sa propre plateforme à partir d'une base commune.

Le livrable 2.4-4 fourni en juin 2021 présentait des POCs d'expérimentations de cette méthodologie sur des systèmes de véhicule privé (VP) et de navette autonome (STPA). Des plateformes étaient donc exposées à travers le livrable, celles-ci ont servi de base pour le développement de la plateforme générique. Ce livrable présente le contexte général du projet pour définir ensuite la plateforme générique visant à intégrer la méthodologie pour enfin présenter une plateforme existante utilisée dans le cadre d'une expérimentation.

La suite de cette tâche sera en juin 2022 avec la présentation du livrable finale 2.4-2 exposant la méthodologie de simulation numérique, ses étapes et les principaux enseignements tirés des POCs réalisés.

Introduction

Les véhicules autonomes sont une technologie émergente du futur, néanmoins leur développement est complexe et la validation de ces derniers longue et fastidieuse. Pour faciliter l'intégration des futurs véhicules autonomes, il est nécessaire d'expérimenter massivement. Ceci pour permettre à la fois de développer les usages et la connaissance de ces systèmes par les citoyens et les acteurs des territoires, et de construire le futur cadre de régulation, notamment en termes de validation de la sécurité.

La sécurisation des véhicules de conduite automatisée est l'un des facteurs les plus importants et aucun système n'a aujourd'hui fait la preuve de sa capacité à tenir les exigences requises. Elle englobe plusieurs aspects qui nécessitent des approches spécifiques dépendantes du type du système, de son environnement et de son domaine d'opération, ainsi que des niveaux de sécurité à atteindre. En effet, il faut sécuriser le système par rapport aux défaillances potentielles ("Safety"), le protéger des événements extérieurs à risque ("Security"), éviter de mauvaises décisions du système ("Fonctionnel sûr") et enfin, écarter les risques d'une utilisation erronée par le conducteur.

Pour la démonstration de la sécurité d'un tel système, il est nécessaire de disposer de méthodologies et référentiels communs partagés entre toutes les parties prenantes (constructeurs, opérateurs, collectivités et autorités, etc.). Il est notamment nécessaire de disposer de référentiels d'objectifs de sécurité, d'un catalogue de scénarios critiques pertinents pour la démonstration de la sécurité, ainsi que de plusieurs méthodes complémentaires permettant de démontrer les niveaux d'atteinte des objectifs de sécurité (selon le type de système à tester dans leur environnement et contextes opérationnels) suivant une approche globale cohérente.

Le projet SAM vise à élaborer un « bien commun », défini par l'ensemble des connaissances dont la mutualisation et le partage avec les autorités publiques bénéficient à l'élaboration des politiques publiques et à la construction d'un état de l'art, notamment en matière de sécurité, d'impacts et d'acceptabilité. Ce bien commun sera construit autour d'une approche méthodologique commune et partagée, dans les trois domaines d'application ciblés : véhicule particulier autonome, système autonome de transport collectif et partagé, système autonome de transport de marchandises.

La validation des véhicules autonomes est un véritable défi au vu de la complexité et la variation infinie de l'environnement de roulage que peut rencontrer sur la route le véhicule autonome. Afin d'accélérer le processus de validation et de réduire les coûts de développement, la validation par simulation devient une solution indispensable dans le processus de développement. La plateforme décrite dans ce livrable représente une solution générique pouvant intégrer le processus de validation.

1. Contexte

1.1. Outils de simulation

La validation du véhicule autonome est un processus long et complexe. Afin de l'accélérer, la simulation est devenue l'outil préférentiel des acteurs de l'automobile. Le rendement offert par la simulation permet d'outrepasser les lacunes des essais physiques :

- Coûts élevés
- Difficultés de réalisation de cas d'usage spécifiques (accidents/environnement difficiles)
- Durée de validation (nombre de kilomètres nécessaires extrêmement élevé)

La nécessité d'obtenir des résultats de test rapides et reproductibles impose d'utiliser divers outils de simulation "in-the-loop" lors de la conception et la validation des systèmes autonomes (Chauffeur autonome/ADAS). Ces outils sont exposés dans les parties suivantes.

MODEL-IN-THE-LOOP (MIL)

La conception et les spécifications initiales du chauffeur autonome s'appuient sur des simulations MIL dans lesquelles la logique du contrôleur est simulée en boucle fermée, et ce avec des modèles de dynamique du véhicule, de capteurs, d'actionneurs et de trafic environnant. La configuration du système de contrôle et les paramètres de la loi de contrôle sont adaptés pour répondre aux exigences de stabilité et de performances souhaitées.

Les outils de simulation actuels standards ne permettent pas de tester de manière fiable un ADAS complet avec une intégration complète des conditions de fonctionnement, des caractéristiques des capteurs, de la dynamique du véhicule et des scénarios de trafic. Il est nécessaire de définir des domaines d'emploi des outils de simulation. La méthodologie proposée à travers le projet SAM contient cette phase de spécification des domaines d'application. Des outils de simulation de haute représentativité tels que SCANeR™studio, CarMaker ou PreScan permettent une simulation MIL fiable des ADAS dans une simulation microscopique de trafic, en utilisant des modèles de capteurs pour la vision radar, lidar et caméra dans un environnement virtuel.

SOFTWARE-IN-THE-LOOP (SIL)

Dès lors que les résultats obtenus à partir des simulations MIL sont jugés suffisants, le code logiciel peut être généré à partir du modèle du système de contrôle en utilisant la génération automatique de code. La simulation SIL grâce aux composants matériels restants (environnement, dynamique véhicule) simulés permet de vérifier que le code réel est bien fonctionnel. Les techniques d'analyse de logiciel sont appliquées sur le code généré pour vérifier sa cohérence par rapport aux spécifications et aux exigences attendues.

HARDWARE-IN-THE-LOOP (HIL)

De façon analogue à l'analyse du logiciel dans une simulation SIL, le matériel réel peut être testé dans une simulation Hardware-in-the-loop en temps réel. Les simulations HIL consistent en une combinaison de

composants simulés et physiques. En variante, un composant réel peut être simulé, ou alors remplacé par un composant artificiel ayant les mêmes caractéristiques d'entrée et de sortie. Idéalement, chaque composant ne perçoit pas de différences entre les composants réels ou simulés auxquels il est connecté dans une configuration en boucle fermée. Par conséquent, les simulations HIL offrent la flexibilité d'une simulation tout en incorporant la représentativité du matériel réel. Le principal avantage d'une simulation HIL est qu'elle fournit un environnement de laboratoire reproductible pour une validation du contrôleur sûre, flexible et fiable. L'ensemble des éléments peuvent être systématiquement testés sans perturbation provenant d'autres systèmes indépendants, cela permet d'augmenter les performances et la stabilité du contrôleur. La simulation HIL présente aussi un avantage conséquent dans la validation de matériel physique réel : pouvoir valider en phase de développement en se passant des autres éléments du véhicule (prototype non nécessaire), ceux-ci étant simulés. Pour ces raisons, les simulations HIL sont plus efficaces et moins chères que les essais sur piste, et sont largement utilisées pour le développement de systèmes de contrôle des véhicules.

VEHICLE IN THE LOOP (VIL)

Un essai VIL consiste à modifier un véhicule à tester équipé d'un système d'aide à la conduite en reliant les sorties des capteurs à un ordinateur embarqué qui modélise l'environnement. Une scène incorporant des éléments virtuels (obstacle, véhicule, ...) est simulée en temps réel pendant toute la durée de l'essai. Le véhicule est opéré sur une piste d'essais qui peut être soit une voie soit une aire d'évolution, mais également sur un banc dynamométrique pour certains systèmes d'aide à la conduite. Ce type d'essai permet de valider le comportement dynamique du véhicule lorsqu'il est contrôlé par le système d'aide à la conduite, y compris dans des situations critiques. Par contre, les performances des capteurs ne sont pas évaluées puisque les signaux sont injectés en aval.

1.2. Plateforme générique

La méthodologie de simulation qui sera présentée dans le livrable 2.4-2 intègre dans son processus la simulation de scénarios pour éprouver le modèle de chauffeur autonome ou des différents ADAS. La *Figure 1* : Plateforme de simulation à base de scénarios présente le fonctionnement global d'une plateforme de simulation pouvant être utilisée dans des modes de simulation présentés plus tôt (MIL/SIL/...).

Pour étendre l'utilisation de la méthodologie au plus grand nombre, l'objectif est de créer une plateforme aussi générique que possible. En effet la principale difficulté rencontrée pour l'application de la méthodologie de simulation est la diversité des environnements de simulation présents chez les acteurs de l'automobile. Pour pallier à ce problème, l'idée est de permettre à la plateforme de s'intégrer dans le processus de simulation facilement. Cependant avant d'établir des formats standards permettant de connecter des modèles entre eux ou directement avec des outils de simulation il faut établir l'architecture globale d'une plateforme.

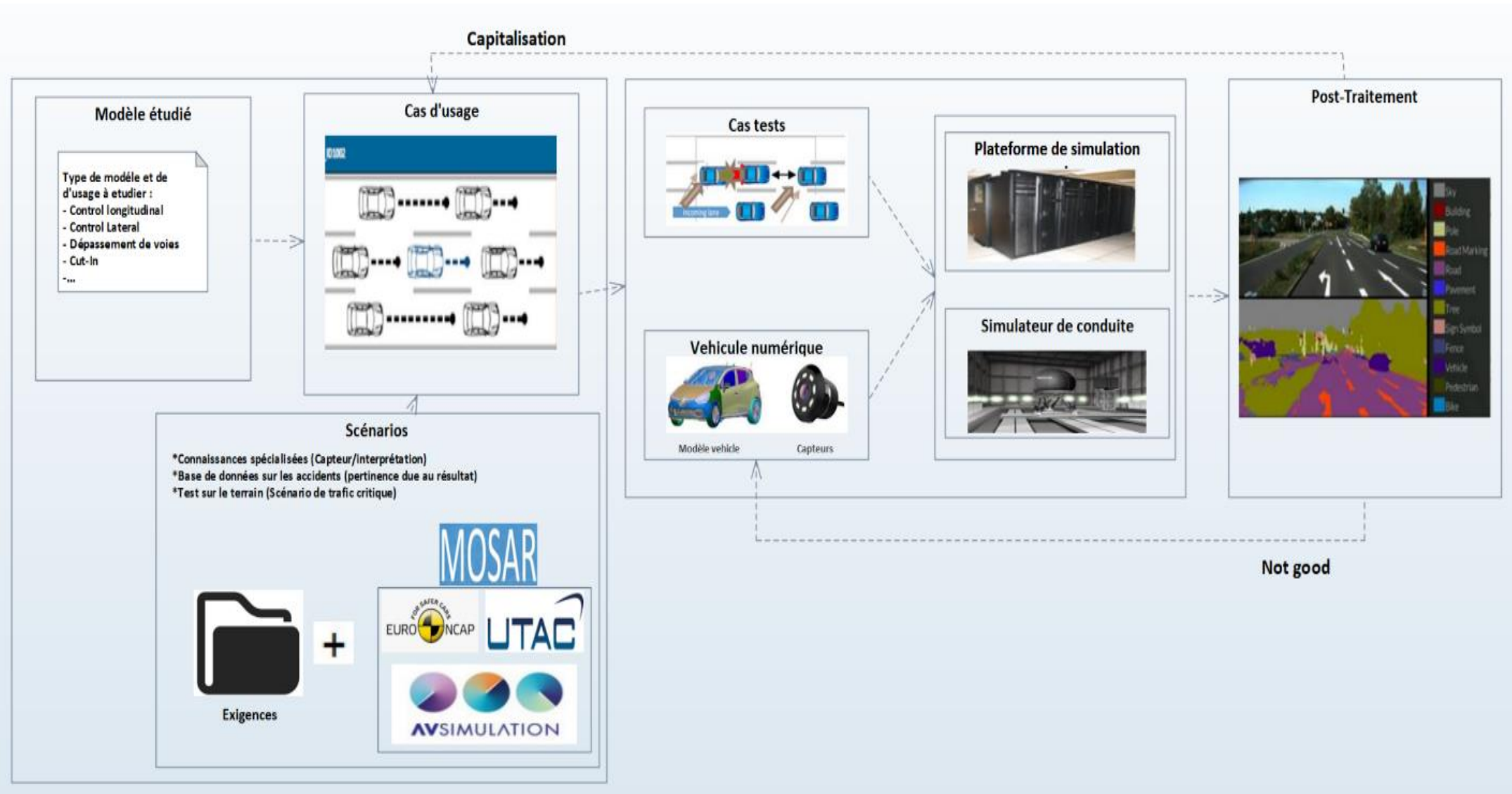


Figure 1 : Plateforme de simulation à base de scénarios

L'architecture d'une telle plateforme s'articule autour de trois blocs distincts présentés dans la *Figure 1* :

- Bibliothèque de scénarios
- Outils de simulation
- Post-traitement

Chaque bloc représente une étape dans le processus de simulation que nous allons définir dans la suite de ce document.

1.2.1. Scenario Manager

Tout d'abord le premier bloc représente les entrées à fournir à la plateforme de simulation, ce sont les futurs scénarios de simulation. Dans ce premier bloc on distingue plusieurs tâches qui permettront de définir des cas tests qui seront les descriptions des futurs scénarios de simulation. On nommera ce bloc *Scenario Manager*.

Dans le cas du projet SAM le module Scenario Manager est tiré de la plateforme MOSAR (Méthodes et Outils pour l'évaluation de la Sûreté de fonctionnement et l'Analyse de la Robustesse des véhicules autonomes), fruit de la collaboration des acteurs automobiles français de premier rang. Cette plateforme propose une méthodologie et une suite outillée pour concevoir et valider la sécurité du véhicule autonome avec notamment l'utilisation d'une base de scénarios.

La plateforme MOSAR définit un modèle de données et un format de fichier dérivé pour la description de scénarios qui pourraient être extraits pour être utilisés dans les simulateurs de conduite et de trafic, ainsi que dans le développement, les tests et la validation virtuels automobiles. La description d'un scénario peut être basée sur des actions du conducteur (par exemple, effectuer un changement de voie) ou sur des trajectoires (par exemple, dérivées d'une manœuvre de conduite enregistrée). Nous fournissons, dans ce qui suit une définition des éléments hiérarchiques, à partir desquels les scénarios sont construits, et leurs attributs et relations correspondants :

- Le storyboard, composé d'histoires, d'actes et de séquences ;
- Événements, déclenchés par des conditions, qui provoquent l'exécution d'actions ;
- Références aux descriptions logiques du réseau routier ;
- Les entités, telles que les véhicules, les piétons, agissant sur et hors de la route ;
- Paramétrage ;
- Catalogage.

D'autres contenus, tels que la description du véhicule ego, des piétons, des conditions de circulation et de l'environnement, sont également inclus.

Les descriptions de scénarios sont une partie essentielle pour tester, valider et certifier la sécurité des systèmes d'aide à la conduite et des voitures autonomes. L'industrie, les agences de certification et les autorités gouvernementales travaillent conjointement sur la définition de bibliothèques de scénarios, qui peuvent être utilisées pour tester et valider le fonctionnement sûr de tels systèmes.

Prenons l'exemple d'un système ADAS bien connu, l'*Autonomous Emergency Braking* (AEB). Nous avons ici un modèle qui contrôle la partie longitudinale du mouvement du véhicule, on vient ensuite dans la bibliothèque sélectionner les scénarios fonctionnels associés à ce type de manœuvre du véhicule, mais

aussi au type d'environnement que l'on souhaite (véhicule seul, trafic environnant, piétons). On obtient ainsi un lot de scénarios logiques associés à un modèle que l'on souhaite tester c'est ce qu'on nomme des cas d'usage. Les cas d'usage qui représentent les sorties du *Scenario Manager* sont ensuite détaillés en un panel de scénarios concrets appelés cas tests qui seront exécutés via le système de simulation. Les cas tests sont obtenus à partir des scénarios logiques en fixant les paramètres relatifs aux environnements, aux autres véhicules. Par exemple les vitesses, les distances entre véhicules. Les combinaisons possibles entre l'ensemble des paramètres peuvent conduire à un nombre très élevé de scénario à simuler.

1.2.2. Système de simulation

Le second bloc est composé principalement des éléments alloués à l'exécution des scénarios de simulation tels que l'outil de simulation numérique, les modèles composant le véhicule (dynamique/ADAS/capteurs) et les cas tests issus du premier bloc. L'outil de simulation numérique peut prendre plusieurs formes, une multitude de logiciels peuvent prétendre à cette fonction. Comme présenté dans la *Figure 1*, en fonction des résultats que l'utilisateur souhaite mettre en avant lors des simulations, un être humain peut faire partie de la chaîne de simulation. Il est donc possible d'intégrer un simulateur physique en plus de l'outil de simulation numérique. La pluralité des simulateurs existants et donc des composants mécaniques vient appuyer la nécessité de sélectionner un outil de simulation ayant la capacité de s'interfacer avec eux.

En effet chaque outil étant différent, les formats et interfaces associés changent. Il n'existe pas de nos jours de formats standards suffisamment développés ou alors généralisés à l'ensemble des acteurs de l'automobile pour que la plateforme puisse être pleinement générique.

1.2.3. Post-traitement

Enfin pour compléter la plateforme, il faut associer aux deux premiers blocs, une étape de post-traitement et de validation. L'objectif est d'extraire de la simulation les données pertinentes pour les confronter à des tests d'évaluation, ou des critères de performances définis dans la méthodologie. Cette étape permet de caractériser les simulations avec deux objectifs à la clé :

- L'amélioration continue de la plateforme
- La validation du système sous test

Le premier objectif comprend la partie corrélation avec les essais physiques associés aux cas tests. A l'aide de méthode de corrélation et d'indicateurs spécifiques la simulation est caractérisée pour définir si la plateforme de simulation est bien valide pour tester le modèle sous test (chauffeur, ADAS, modèle dynamique). Comme dans un système en boucle fermée, dans le cas où le système testé ne présente pas de résultats cohérents lors de la simulation, il est donc nécessaire de modifier le deuxième bloc de la plateforme et donc d'améliorer le modèle mis en cause. Dès lors que les simulations sont considérées comme satisfaisantes, celles-ci sont comparées aux critères de validités et de performances du système de manière similaire à des essais réels. C'est cette dernière étape qui sera utilisée pour conclure sur la validité d'un système vis-à-vis des normes et de la réglementation en vigueur sur les systèmes du véhicule autonome.

2. Architecture générique

Maintenant que le contexte général a été présenté, l'architecture générique de la plateforme de simulation sera développée dans les parties suivantes. Cette architecture est une proposition construite autour de formats « standards » ou avec des capacités d'interfaçage très large, ainsi qu'avec des logiciels populaires dans le domaine du transport (particulier, public, marchandise) autonome.

2.1. Description Générale

L'architecture global de la plateforme de simulation est présentée en *Figure 1*, nous nous intéresserons ici au système de simulation à savoir le deuxième bloc de la plateforme, voir la *Figure 2*.

Ici encore le système global est découpé en trois parties que nous détaillerons par la suite. Le bloc central représente le système de simulation. C'est le cœur de la plateforme, il regroupe l'outil de simulation numérique qui sera en charge de centraliser l'ensemble des modèles du système de simulation pour calculer le pas de temps suivant, ainsi que les modèles structurant le véhicule autonome testé, tel que le modèle de chauffeur autonome, les modèles d'ADAS (décision et contrôle du véhicule), mais aussi les modèles de détection (perception et fusion). Ces modèles contrairement aux paramètres décrits précédemment fonctionnent en parallèle de l'Outil de Simulation Numérique (OSN), voir grâce à l'OSN dans certains cas, sur des logiciels de modélisation système multi-physique tel que SIMULINK, RT Maps, etc. Il n'y a pas actuellement d'outil générique qui soit un standard dans la simulation numérique. Plusieurs acteurs majeurs sont positionnés sur ce segment (CarMaker, SCANeR™studio, etc.) avec des solutions offrant diverses possibilités mais aucune ne peut être considérée comme générique car elles sont soumises à des licences logicielles. Cependant, des outils de simulation Open Source tel que *CARLA* permettent de rendre la simulation accessible à un plus grand nombre d'utilisateurs.

La partie gauche représente les entrées de l'outil définies par l'utilisateur, le scénario, le terrain, mais aussi les configurations de capteurs. Tous ces éléments qui caractérisent la simulation qui sera jouée à l'aide de l'OSN et qui ne nécessitent pas de modèle dynamique propre. Ce sont les paramètres d'entrées, ils sont toujours nécessaires à l'exécution de la simulation mais doivent correspondre aux interfaces définies par l'OSN, ou/et par les standards existants actuellement.

Enfin, la partie droite de la *Figure 2* symbolise l'étape de traitement des données après l'exécution des simulations. L'export des données y est traité pour les rendre utilisables via les logiciels dédiés. Il s'agit ensuite de faire l'évaluation du système testé sur la plateforme à partir de critères définis associés à l'étude ou en comparaison à des essais similaires. L'objectif ici est d'automatiser ou de standardiser la phase de traitement post-simulation.

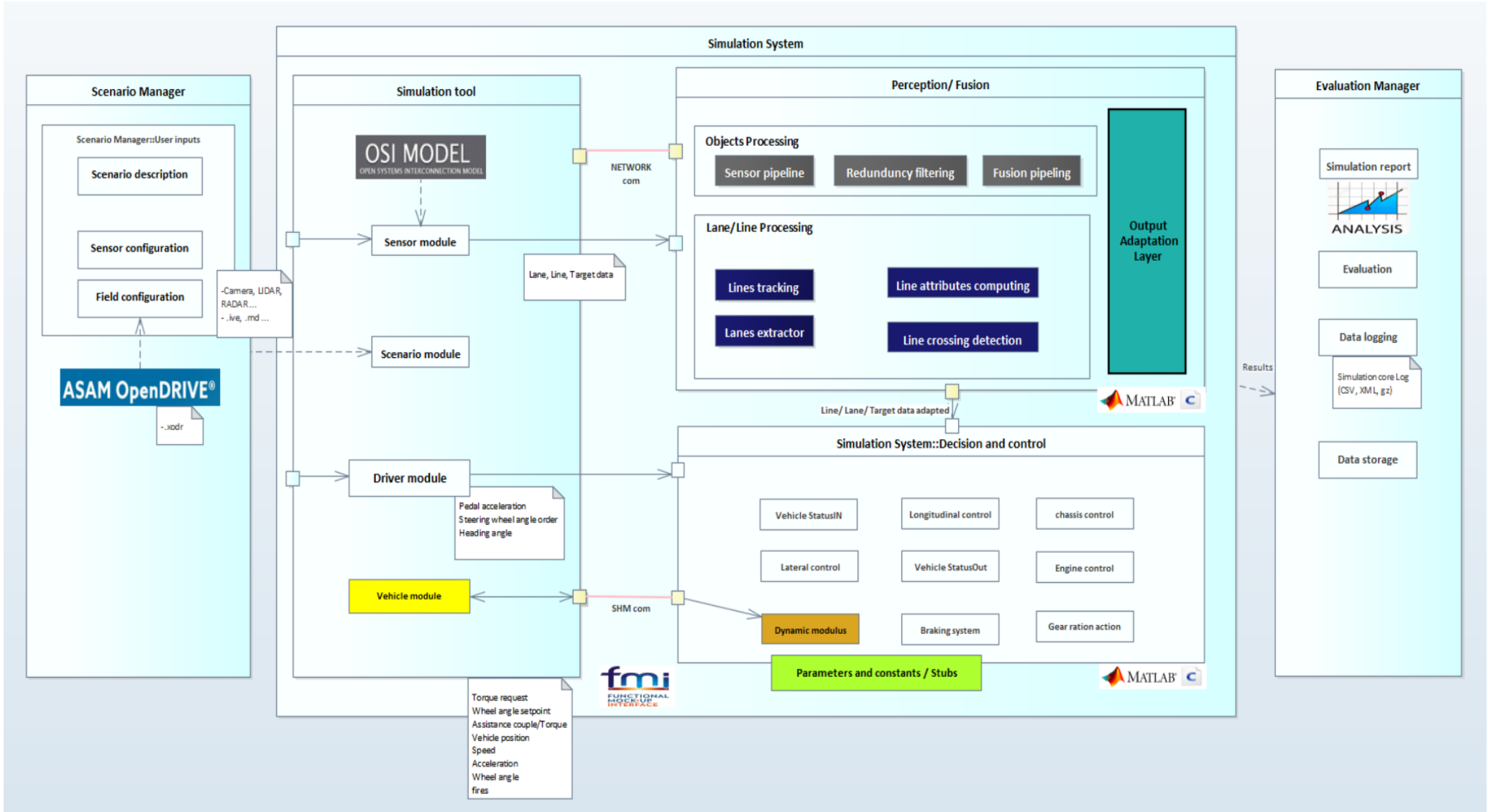


Figure 2: Schéma du système de simulation générique

2.2. Description des interfaces

2.2.1. Entrées de l'utilisateur

Dans le cadre de la simulation de véhicule autonome (ou robot), l'OSN nécessite un certain nombre de paramètres qui sont à définir par l'utilisateur. Ceux-ci sont schématisés dans la partie gauche de la *Figure 2* et sont les entrées du système de simulation. En se basant sur des formats standards le *Scenario Manager* décrit dans la *Figure 1* doit fournir les données au système sous la forme de fichier scénario.

Les scénarios de simulation, bien que présentant des structures parfois similaires entre les logiciels, ne possèdent pas de format standard. Néanmoins, des travaux de l'ASAM (Association for Standardization of Automation and Measuring Systems) sont en cours pour proposer un format appelé *OpenSCENARIO* qui vise une normalisation de ce format dans le domaine de la simulation. ASAM *OpenSCENARIO* définit un format de fichier pour la description du contenu dynamique des simulateurs de conduite et de trafic. Le cas d'utilisation principal d'*OpenSCENARIO* est de décrire des manœuvres complexes et synchronisées qui impliquent plusieurs entités telles que des véhicules, des piétons et d'autres participants à la circulation. La description d'une manœuvre peut être basée sur des actions du conducteur (par exemple, effectuer un changement de voie) ou sur des trajectoires (par exemple, dérivées d'une manœuvre de conduite enregistrée). D'autres contenus, tels que la description du véhicule ego, l'apparence du conducteur, les piétons, la circulation et les conditions environnementales, sont également inclus dans la norme. C'est pourquoi, lorsque ce format sera suffisamment mature, il deviendra le format privilégié pour la création de scénarios.

Les scénarios de simulation contiennent la description des événements et regroupent le contenu de la simulation, c'est-à-dire les différents éléments qui serviront à l'exécution : le terrain, les acteurs (véhicules, piétons) ainsi que les capteurs nécessaires au fonctionnement du véhicule autonome. Ces différents éléments sont aussi à fournir par l'utilisateur. Dans le cas du terrain, celui-ci sera défini par la description fonctionnel du scénario (dans la bibliothèque MOSAR). Pour suivre le même processus entre le *Scenario Manager* et le système de simulation, la description du terrain se doit d'être dans un format standard. Un format standard actuel et mature de description de terrains existe également via l'ASAM : *OpenDRIVE*. Ce format permet de définir une base commune pour la description de réseaux routiers (géométries des routes, voies, objets ainsi que des marquages et autres signaux). De plus ce standard décrit aussi bien un terrain virtuel qu'un terrain issu de données réelles.

Les éléments restants à renseigner par l'utilisateur en tant qu'entrées du système de simulation sont les informations des capteurs, ou plutôt la configuration attendue. Actuellement, de simples fichiers XML permettent aux outils de simulation de construire la configuration dans la simulation.

2.2.2. Outil de simulation

La partie précédente décrit les interfaces que l’outil de simulation numérique devra gérer pour récupérer les entrées utilisateurs, il est nécessaire ensuite de définir le reste des interfaces qui permettront à l’outil d’exécuter la simulation avec des modèles extérieurs. Pour cela d’autres standards sont mis en place afin de faciliter les échanges de données avec les modèles dynamiques par exemple ou bien de perception et fusion.

Dans un premier temps, pour les modèles dynamiques, le format Functional Mock_up Interface (FMI) semble convenir le plus de par sa capacité d’interopérabilité. Le FMI est une norme gratuite définissant un conteneur et une interface d’échange des modèles dynamiques à l’aide d’une combinaison de fichiers XML, de binaires et de code C compressés dans un seul fichier. Il est pris en charge par plus de 150 outils. Ce format est donc privilégié afin de faciliter les connexions avec l’outil de simulation sans avoir à prendre en compte son API (Application Programming Interface) propre.

Ensuite, tel que précisé précédemment, bien que l’OSN s’occupe de l’exécution générale de la simulation, une partie des algorithmes nécessaires à la perception et fusion des données de capteurs n’est réalisée qu’à l’aide de modèles extérieurs (dans le cadre de simulation MIL/HIL). Dans ce cas aussi, il est donc nécessaire de s’appuyer sur des formats de données standards. Une nouvelle fois l’ASAM s’est penchée sur le sujet et propose un type de messagerie appelée *Open Simulation Interface* (OSI). OSI contient une description d’environnement basée sur des objets utilisant le format de message de la bibliothèque de tampons de protocole développée et maintenue par Google [1]. OSI se compose de deux messages individuels de niveau supérieur définissant l’interface GroundTruth et l’interface SensorData.

L’interface GroundTruth donne une vue exacte sur les objets simulés dans un système de coordonnées global. Ce message est renseigné à l’aide des données disponibles en interne, puis publié aux abonnés externes par un plugin s’exécutant dans le cadre de simulation de conduite.

L’interface SensorData décrit les objets dans le référentiel d’un capteur de perception environnementale. Il est généré à partir d’un message GroundTruth et peut être utilisé pour se connecter directement à une fonction de conduite automatisée à l’aide de données simulées idéales, ou peut servir d’entrée pour un modèle de capteur simulant une perception limitée en tant que réplique du comportement du capteur du monde réel.

C’est à l’aide de ces différents formats et messages que le système de simulation global pourra fonctionner durant toute la durée de la simulation. Il faut préciser que la plateforme est décrite tel que l’utilisateur puisse intégrer ses modèles simplement en les interfaçant via ces formats avec l’OSN. Néanmoins, la plateforme de simulation doit permettre de valider une multitude de fonctionnalités du véhicule autonome (dynamique véhicule, ADAS, capteurs) c’est pourquoi l’outil de simulation peut varier afin de permettre à l’utilisateur de ne pas posséder l’un des éléments et de proposer un modèle « générique ».

3. Plateforme Navette Autonome (SystemX)

Il est présenté dans cette dernière partie une architecture de plateforme créée dans le cadre des expérimentations du lot 2.4 de SAM. Celle-ci a été construite et testée par les équipes de *l'Institut de Recherche et Technologique SystemX* et présente une partie des éléments de la plateforme de simulation générique avec des adaptations en fonction des outils mis à disposition.

L'expérimentation menée sur un modèle de navette autonome a permis de développer la plateforme de simulation telle que décrite dans la *Figure 3*. On retrouve l'architecture générale de la plateforme de simulation générique adaptée aux besoins des expérimentations ainsi qu'aux outils disponibles chez les partenaires et au sein de l'IRT SystemX.

3.1. Outil de simulation numérique (OSN)

La construction de la plateforme de simulation numérique se base dans un premier temps autour de l'OSN. En effet, comme développé précédemment, l'ensemble des modèles et paramètres vont devoir s'interfacer avec cet outil, c'est pourquoi son choix est crucial. Les expérimentations menées avec nos partenaires sur le lot 2.4 du projet SAM, RENAULT pour le cas du VP et TRANSDEV dans le cas STPA, ont définies l'usage d'un OSN en particulier. Il s'avère que L'IRT SystemX, RENAULT et Transdev sont tous utilisateurs de SCANeR™studio, ainsi le choix s'est rapidement porté sur ce logiciel.

Développé pour des experts de l'automobile, SCANeR™studio est conçu pour répondre aux besoins spécifiques des professionnels de la simulation dynamique. L'utilisation du logiciel a été pensée, initialement, autour du processus d'utilisation des simulateurs de conduite et a été structuré autour de cinq modes dédiés accessibles depuis l'interface graphique :

- TERRAIN : Édition de réseau routier RoadXML sous format manuel ou automatique via l'import de différents types de base de données routières (GPX, XML, OSM, SHP, ...);
- VÉHICULE : Outil de mise au point et d'étude de modèles dynamiques dispose d'un modèle dynamique non linéaire et peut importer des modèles externes;
- SCÉNARIO : Editeur de scénario de simulation de conduite qui dispose de ses propres fonctions qui permettent de définir des scénarios complexes de simulation et il permet également l'import de scénarii développés sous python.
- SIMULATION : Outil de supervision de la simulation et du simulateur, couplé à Matlab/Simulink pour une co-simulation entre les deux environnements;
- ANALYSE : Outil graphique d'analyse fine des données de simulation. Toutes les données du scénario de simulation sont enregistrées d'une manière synchrone et peuvent être affichées dans des graphiques ou exportées sous d'autres formats.

SCANeR™studio dispose de plusieurs modèles génériques de capteurs (Radar, Caméra, GPS, Ultrason, etc.). La modularité et la richesse de l'environnement SCANeR lui permettent d'être utilisé pour concevoir les systèmes avancés d'aide à la conduite (ADAS), les analyser, les optimiser et les valider :

- Conception des ADAS :
 - Navigation system;
 - Adaptive Cruise Control (ACC);

- Lane Departure Warning (LDW) ;
- Distance Warning (DW) ;
- Analyse et optimisation « Environnement-Véhicule-conducteur »
 - Environnement
 - Modéliser une infrastructure routière et la parcourir avec différents acteurs.
 - Gestion des conditions météorologiques et d'éclairage.
 - Modéliser le comportement dynamique d'un ou de plusieurs acteurs.
 - Véhicule
 - Gérer les capteurs (Radar, Caméra, GPS, etc.).
 - Mettre en situation et évaluer les ADAS.
 - Conducteur
 - Modéliser et analyser le comportement du conducteur.
- Validation
 - MIL (Model in the Loop)
 - SIL (Software in the Loop)
 - HIL (Hardware in the Loop)

3.2. Modèles et interfaces

Le choix de cet OSN induit des interfaces et types de format spécifiques, comme il est noté dans la partie 0.

3.2.1. Entrées de l'OSN

Les entrées définies par l'utilisateur, cf. partie 2.2.1, sont indexées sur les formats que l'OSN est capable de gérer. Comme le montre la *Figure 3*, trois différents paramètres sont fournis à SCANeR™studio :

- Scénario
- Terrain
- Configuration capteurs

Pour le premier, les scénarios logiques sont tirés de la bibliothèque MOSAR. La description ainsi faite, un travail manuel est fourni pour créer le scénario sous SCANeR™studio. A l'aide de l'interface, il faut disposer les acteurs du scénarios (véhicules, piétons, etc...) et créer la scène telle qu'elle est décrite dans la bibliothèque. A l'aide de script intégrés aux scénarios, il est possible de définir des variables en paramètres du scénario qui représenteront les domaines de validité des scénarios logiques. Lorsque les domaines sont définis, il est aisé de choisir un scénario concret par simple choix d'un set de valeurs de ses paramètres. Lorsque le scénario concret est définit, il est alors décrit sous la forme d'un fichier « .SCE » et peut être simulé sur le système de simulation.

Le scénario décrit précédemment, nécessite un terrain sur lequel exécuter les événements décrits. Ce terrain peut, grâce à SCANeR™studio, soit être créé à partir de zéro dans l'interface logiciel, ou bien être importé depuis un format de fichier supporté. En effet, il est précisé dans la partie 2.2.1 que, pour ces entrées, le format visé est OpenDRIVE, celui-ci représentant un standard du secteur de la simulation.

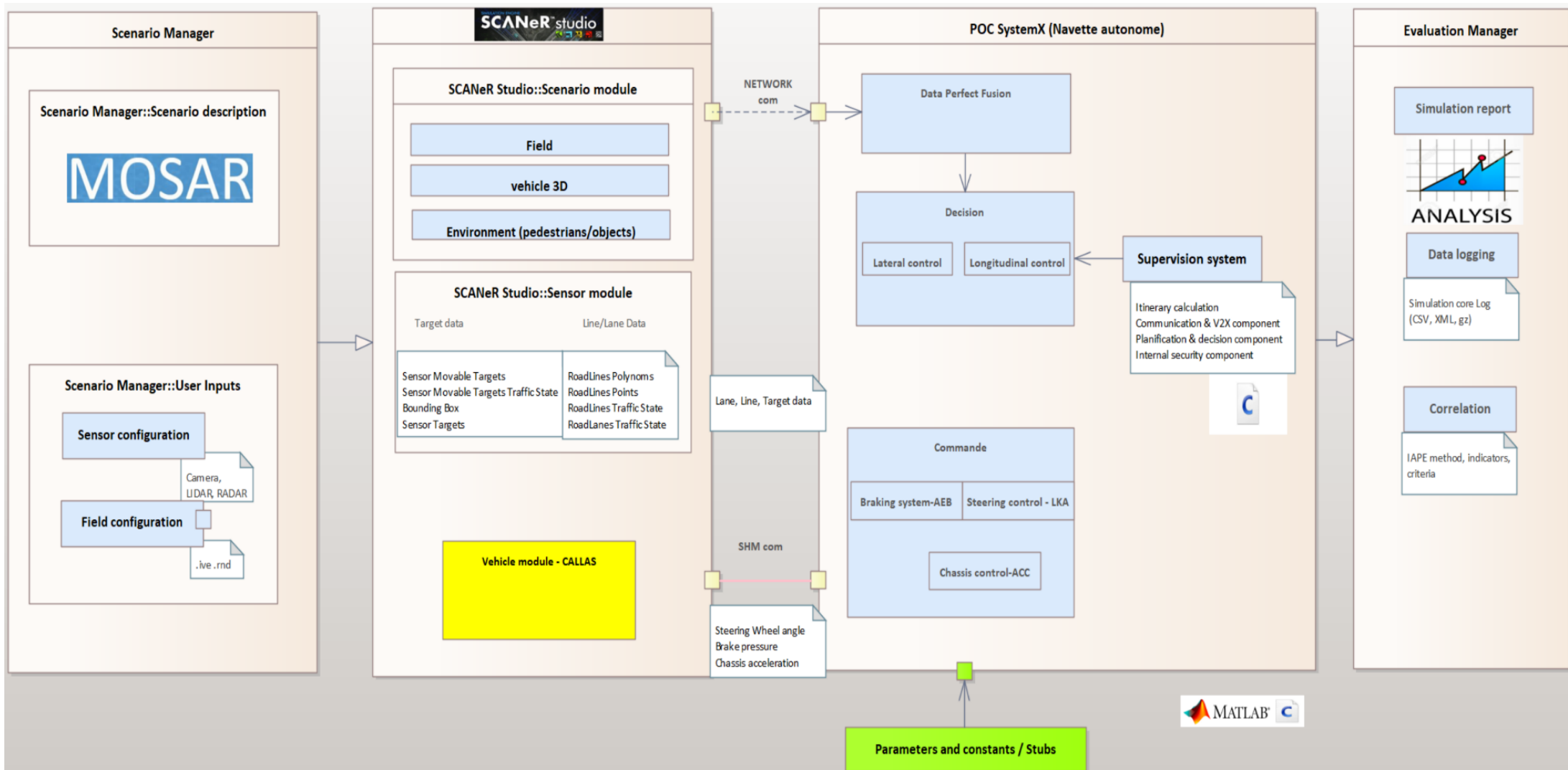


Figure 3 : Architecture de la plateforme Navette autonome (IRT SystemX)

L'outil SCANeR™studio est compatible avec ce format, ainsi qu'avec d'autres tels que OpenStreetMap, CSV, Mapple, etc. Le terrain ainsi construit dans SCANeR™studio se retrouve sauvegarder sous 2 formats liés :

- « .RND » pour l'information logique du réseau routier (marquages, nombre de voies, signalisation)
- « .IVE » contenant les informations graphiques (objets 3D) utilisées pour le rendu visuel

Ici, l'OSN est donc compatible avec les formats usuels, il se charge de la conversion dans les formats qu'il gère dans la simulation.

Enfin, les configurations de capteurs et/ou de modèles de capteurs sont décrites dans l'OSN. Le travail que l'utilisateur doit fournir lors de la création du scénario comprend la création de la configuration capteurs. Néanmoins, cette configuration est décrite dans un fichier « .XML » qui peut tout aussi bien être édité manuellement si l'on souhaite le modifier directement. Il faut pour cela savoir exactement ce que l'on souhaite modifier sous peine de rendre la configuration inutilisable.

3.2.2. Modèles de l'OSN

Les expérimentations menées sur la simulation de la navette autonome (Transdev-SystemX) ont nécessité un système de simulation où certains modèles sont intégrés directement dans l'OSN. Ici, l'OSN n'est pas seulement l'outil d'exécution de la simulation mais permet de faire fonctionner le modèle dynamique de la navette via le modèle existant dans SCANeR™studio : CALLAS. CALLAS est l'acronyme de : Couplé A La Limite d'Adhérence au Sol, ce modèle est conçu pour fournir un large éventail d'applications : camions, bus, voitures, sports automobile, machines, tracteurs et véhicules militaires (tels que les véhicules à chenilles), etc. C'est un mélange de modélisation fonctionnelle et multi-corps. L'objectif de ce concept de modélisation fonctionnelle est de fournir un moyen flexible de configuration de n'importe quelle solution technologique. Via ce modèle, Transdev et AV Simulation (fournisseur de SCANeR™studio et CALLAS) ont travaillé conjointement à la modélisation de la navette autonome au plus proche du modèle réel pour obtenir un modèle dynamique représentatif en simulation.

L'échange de données par FMI n'est donc pas nécessaire dans cet exemple car le modèle dynamique est pleinement intégré dans l'OSN. Nous verrons dans la partie suivante les éléments hors de l'OSN nécessitant des interfaces spécifiques pour interagir avec.

Les capteurs présents sur la navette sont définis dans la simulation. Ce sont des capteurs « parfaits » ayant les mêmes caractéristiques (range, champ de vision, position, etc...), mais les modèles ne subissent pas directement les dégradations dues à l'environnement. Celles-ci devront être simulées à partir de seuils définis en amont. Lors de la simulation les capteurs envoient sur le protocole de communication de l'OSN des messages contenant les données de détection (cf *Figure 3*, Target data & Line/lane data). Ces messages serviront au modèle de chauffeur autonome qui permet de son côté la fusion des données capteurs pour la prise de décision afin de contrôler la navette.

La *Figure 3* présente les modèles décrits précédemment comme des modules, ceci car SCANeR™studio fonctionne de façon modulaire. En effet, chaque « type » de composant intégré est émulé par un module propre. Ces modules sont des exécutables propres capables d'interagir avec la SCANeR API. Dans le cas de la navette autonome il y a donc 3 modules utilisés :

- SENSORS
- SCENARIO
- MODELHANDLER

Le module SENSORS se charge d'envoyer sur le réseau de SCANeR™studio (appelé NETWORK) les messages associés aux capteurs. Ces messages fournissent les informations sur les objets détectés (signaux, véhicules) ainsi que les voies et marquages, ils sont décrits par l'interface suivante : ISensor. Les interfaces catégorisent les messages qui transitent sur la messagerie NETWORK.

Le module SCENARIO est en charge de l'exécution du scénario de simulation ainsi que du script associé. De la même façon, le déroulement du scénario donne lieu à l'envoi de messages sur le NETWORK. Il est ainsi possible d'utiliser ces informations avec les différents modèles du système de simulation.

Le module MODELHANDLER se charge de l'interaction entre le modèle dynamique et la surface de la route. Le modèle dynamique est décrit dans le fichier véhicule, un véhicule CALLAS dans le cas de l'expérimentation avec la navette Transdev. Ce module n'est en fait pas seul pour le contrôle du véhicule dans SCANeR™studio, il nécessite un module qui lui transmet les commandes du conducteur (autonome ou manuel). Cet autre module, dans notre exemple, est nommé DRIVERHANDLER, il sert de passerelle entre le chauffeur autonome et MODELHANDLER.

3.2.3. Chauffeur autonome

Les modèles testés dans l'expérimentation menée avec Transdev sont ceux de la dynamique véhicule (cf partie 3.2.2) et du chauffeur autonome. Ce chauffeur a été développé pour correspondre aux spécifications fonctionnelles du modèle de chauffeur des navettes Transdev. Il est en charge du contrôle de la navette à partir des données des capteurs reçues depuis SCANeR™studio. L'ensemble du modèle est réalisé sous SIMULINK et contient plusieurs blocs en charge des tâches suivantes :

- Fusion des données de détection
- Système de supervision
- Décision : contrôle latéral et contrôle longitudinale
- Commande : conversion de la décision en commande

L'ensemble de ces blocs permet à la navette d'être contrôlée simplement à partir des données des capteurs envoyées depuis SCANeR™studio. Le bloc *Décision* récupère les informations de fusion des données des capteurs et du système de supervision pour décider du comportement de la navette. Dans le cas de la navette, celle-ci suit plusieurs critères :

- Réglementation
- Confort
- Dynamique du véhicule

La décision ainsi prise, le calcul des grandeurs de contrôle est réalisé. L'objectif ici est de fournir à SCANeR™studio les valeurs suffisantes pour contrôler la navette. Ces grandeurs se trouvent être l'angle volant, le pourcentage d'appuie sur la pédale de frein et sur la pédale d'accélérateur.

L'interaction avec SCANeR™studio via SIMULINK est possible grâce à l'API de SCANeR™studio disponible sur ce logiciel tiers. Les deux protocoles de communication NETWORK et SHM (Shared Memory) voient leurs messages se retrouver sous forme de blocs dans SIMULINK.

3.2.4. Traitement des données

Le dernier bloc du système de simulation est dédié à l'analyse des données. Pour cela, des scripts de post-traitement sont développés pour extraire les grandeurs pertinentes à analyser des fichiers de simulation. Il est alors possible de caractériser la simulation vis-à-vis des essais physiques, en utilisant des méthodes de corrélation (IAPE et d'autres indicateurs définis par SystemX). Cette partie de corrélation est utile pour définir le domaine de validité de la plateforme de simulation. Ce domaine est nécessaire pour confirmer la cohérence des résultats.

Dès lors que le domaine est défini, les modèles testés peuvent être évalués en utilisant des critères de performances et de sécurités définis à travers les lots du projet SAM.

Conclusion

La complexité croissante des systèmes d'aide à la conduite et de leurs environnements se doit de satisfaire des exigences de plus en plus strictes. La simulation numérique est donc un outil de validation prisé car moins onéreux. Cependant les plateformes de simulation permettant d'effectuer de telles tâches sont trop hétérogènes. La pluralité des logiciels et systèmes physiques rend la construction d'une plateforme compliquée et rarement égale à une autre. Par soucis de compréhension entre les différents acteurs du secteur, la définition d'une plateforme de simulation générique est un objectif commun qui facilitera à l'avenir la mise en place de tels outils.

La plateforme générique décrite dans ce document utilise de nombreux formats, malheureusement pas encore tout à fait matures pour certains. Cependant, ces formats sont amenés à devenir des standards pour la simulation. C'est pourquoi l'architecture proposée les prend en compte pour servir de plateforme usuelle à l'avenir.

Dans l'expérimentation SAM, la plateforme s'adapte aux ressources de l'utilisateur. Suivant le format des modèles disponibles et leur nombre, il est nécessaire d'avoir un outil de simulation numérique capable de s'interfacer avec le plus grand nombre. De plus, dans cet exemple, l'OSN prend en charge le calcul de certains modèles (dynamique, capteurs) ce qui permet à l'utilisateur de profiter de l'usage d'une plateforme sans avoir à mettre en place l'ensemble des outils nécessaires initialement.

Références

Jochen Köhler, Heinz Sachsenweger, “Simulation-based Development and Testing of Automated Driving”, 2021, p5-7

Adel Djoudi, “Simulation pour la Sécurité du Véhicule Autonome”, p14-17

Danny Behnecke, Christopher Sontag, “A Platform for Automated Simulation of Scenario Sequences”

<https://www.asam.net/standards/detail/opendrive/>

<https://www.asam.net/standards/detail/osi/>

<https://www.asam.net/standards/detail/openscenario/>