الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUE ET D'INFORMATIQUE

MEMOIRE DE MASTER

En vue de l'obtention du Diplôme de
Master en Informatique

Option : Ingénierie des systèmes et technologie du web

THEME

Une démarche basée sur SysML pour l'ingénierie des jumeaux numériques -Application sur les systèmes cyber physiques.

Présenté par :

- **CHEBLAOUI Mohamed Lamine**
- **DHAIF Smail**

Dirigé par : Mr. RAHIM Messaoud

Devant le jury :

Mr. TOUBAL El Bahi	**Président**
Mr. RAFA Tahar	**Examinateur**

Année Universitaire : 2018/2019

Remerciements

Avant tout, merci DIEU de nous avoir donné le courage, la volonté et la patience de mener ce travail à terme.

Ensuite, il n'aurait pas pu être achevé sans le soutien, les conseils et l'aide de certaines personnes auxquelles on tient ici à exprimer nos sincères remerciements.

Nos sincères remerciements vont à Mr. RAHIM Messaoud pour son encadrement avec patience. Son encouragement et ses remarques pertinentes nous ont permis de mieux structurer ce travail et de mieux le décrire. Nous lui remercions aussi de nous avoir fait profiter de ses expériences, ses orientations et ses conseils nous ont énormément aidés. L'aboutissement de celui-ci doit beaucoup à sa contribution.

Nous remercions tout particulièrement les membres de jury d'avoir accepté d'examiner notre mémoire.

Nos sincères remerciements s'adressent à nos parents, nos frères, nos sœurs ainsi qu'à toute la famille pour leur soutien moral, leur encouragement inconditionnel, et surtout pour la confiance qu'ils nous accordent.

Enfin, Nous remercions tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail, en particulier tous nos ami(e)s pour leur soutien moral et leur présence à nos côtés.

DEDICACE

Une chance m'a été offerte aujourdhui pour vous parler des personnes qui me sont

Chers et a qui je leurs doit tout.

Avec ce dédicace, j'éspére pouvoir leurs exprimer toutes mes remerciement

Et ma profonde gratitudes .

Tout d'abords, je commence par mes très chers parents qui m'ont toujours tout donné

Que dieu les protéges et les gardes en bonne santé.

Je dédie aussi à :

Grand pére , grand mére

Ma chére frére :Zine el dinne.

Mes chéres sœurs : Lamia-Siham- Farah-Fadwa.

A tout ma famille sans exception du plus grand au plus petit (la famille cheblaoui et la famille khelifa)

A toute mes amies : Redha, Hichem, Mohamed, Youness, Tarek, Smail, Rachid, Ahmed, Aymane.

- ***A tout les professeurs de l'université de médéa.***

Et en darnier je dirais que j'ai beacoup de chance d'avoir des personnes qui m'aiment

Et qui sont toujours présentes pour moi.

Merci

<u>Mohamed lamine</u>

DEDICACE

Tout d'abord je dois remercier mon bon dieu qui m'a donné la force et le Courage pour réussir à

Réaliser ce modeste travail.

Je dédie ce modeste travail à :

Mes très chers parents :

Aucune dédicace, aucun mot ne saurait exprimer ma gratitude, et mon profond respect.

Votre affection.

Je prie Dieu, le tout puissante, de vous protéger

Et de vous procurer Santé, bonheur et longue vie.

A mes parents : Rabeh, Razika.

A mes très chers Frères : Brahim, Mohamed, Issa.

A tous mes familles.

A tous mes amies sont nommées.

. A tout les professeur de université de médéa.

Et en darnier je dirais que j'ai beacoup de chance d'avoir des personnes qui m'aiment

Et qui sont toujours présentes pour moi.

Smail

Merci

Résumé

Un jumeau numérique est une image virtuelle d'un système physique. Ce travail vise à améliorer le processus de développement des jumeaux numériques concernant des systèmes cyber-physiques. Au lieu d'attendre la fin de la construction du système physique pour générer une présentation du jumeau numérique, nous proposons une démarche employant l'ingénierie des systèmes basée sur les modèles pour développer le jumeau numérique du système en se basant sur la modélisation de ce dernier. Dans cette démarche, nous utilisons le langage SysML pour la modélisation et la plateforme (Eclipse Ditto) pour la création des jumeaux numériques. Pour aider les concepteurs, nous avons conçus et développé une application web permettant de supporter toutes les étapes de la démarche proposée.

Mots clés : Jumeau numérique, ingénierie, système cyber-physique, diagramme de blocs, SysML.

Abstract

A digital twin is a virtual image of a physical system. This work aims to improve the development process of digital twins regarding cyber-physical systems. Instead of waiting for the completion of the physical system construction to generate a presentation of the digital twin, we propose an approach using model-based systems engineering to develop the digital twin of the system based on the modeling of this system. latest. In this approach, we use the SysML language for modeling and the platform (eclipse ditto) for the creation of digital twins. To help designers, we designed and developed a web application to support all stages of the proposed approach.

Key words: Digital Twin, cyber physical system, block diagram, SysML, UML, eclipse ditto.

Résumé

الملخص

التوأم الرقمي هو صورة افتراضية للنظام المادي. يهدف هذا العمل إلى تحسين عملية تطوير التوائم الرقمية المتعلقة بالأنظمة الفيزيائية الإلكترونية. بدلاً من انتظار اكتمال بناء النظام المادي لإنشاء عرض تقديمي للتوأم الرقمي، نقترح نهجًا يستخدم هندسة النظم القائمة على النماذج لتطوير التوأم الرقمي للنظام استنادًا إلى نمذجة هذا النظام. الماضي. في هذا النهج، نستخدم لغة سيسيمال للنمذجة والنظام الأساسي (اكليبس ديتو) لإنشاء توائم رقمية. لمساعدة المصممين، قمنا بتصميم وتطوير تطبيق ويب لدعم جميع مراحل النهج المقترح.

الكلمات المفتاحية :التوائم الرقمية، النظام السيبراني، مخطط الكتلة، سيسيمال، ايامال، اكليبس ديتو.

Table des matières

Table des matières

Table des matières VI

Liste des figures X

Liste des tableaux XII

Liste des acronymes XIII

Introduction générale 0

CHAPITRE I : Système cyber-physique et jumeau numérique (digital twin) 2

1. Introduction 3

2. Industrie 4.0 3

2.1. Présentation de l'Industrie 4.0 4

2.2. Définition d'Industrie 4.0 4

3. Systèmes cybers physiques 4

4. Internet des objet IoT 6

5. Jumeaux numériques 7

5.1. Définition 7

5.2. Pourquoi le jumeau numérique (Digital Twin) 10

5.3.1. Visibilité : 12

5.3.2. Prédiction : 12

5.3.3. Analyse : 12

5.3.4. Outil de documentation et de communication : 12

5.3.5. Outil d'aide à la décision : 12

.6 **Conclusion** **12**

CHAPITRE II : État de l'art ..... Error! Bookmark not defined.

1. Introduction **14**

2. Langage de modélisation UML **14**

2.1. Structuration du langage UML **14**

2.2. Profil UML **15**

3. SysML **16**

3.1. Structuration du langage SysML **17**

3.2. Quatre piliers de SysML **18**

3.3. Diagramme de bloc **19**

3. Relation entre UML et SysML **20**

4. Ingénierie dirigée par les modèles (IDM) **21**

4.1. Méta modélisation **22**

4.3. Transformation de modèles **24**

4.3.1. Pourquoi la transformation de modèles **25**

4.3.2. Approche de transformation de modèles **25**

4.3.3. Transformation modèle vers texte M2T **26**

5. Conclusion **27**

CHAPITRE III : Méthodologie. **28**

1. Introduction **29**

2. Démarche Globale 30

3. Démarche de modélisation des systèmes cyber-physiques 31

.3.1**Diagramme de modélisation du système cyber physique 31**

3.2. Modélisation des trois types de composants d'un système cyber physique. 31

3.3. Identification des types des composants d'un système cyber physique 31

4. Méta-modèle du jumeau numérique 33

4.1. Objets (thing) 34

4.2. Liste de contrôle d'accès (ACL) 34

4.3. Fonctionnalité (Feature) 35

5. Démarche de transformation 36

5.1. Transformation modèle vers modèle M2M 36

5.1.1. Bloc physique 37

5.1.2. Blocs cyber physique 38

5.1.3. Acteur 39

5.1.4. Exemple 41

5.2. Transformation modèle vers texte M2T 41

6. Application de la création du jumeau numérique 42

7. Intégration avec le système physique 57

8. Conclusion 57

CHAPITRE IV : Implémentation et étude de cas 58

1. Introduction 59

2. Implémentation 59

2.1. Modélisation 59

2.2. Transformation 61

2.2.1. Transformation ATL 61

2.2.2. Transformation ACCELEO 62

2.3. Intégrations des transformations 63

2.4. Plateforme des jumeaux numériques utilisé 64

3. Etude de cas 65

3.1. Système utilisé 65

3.1.1. Contrôle et gestion de moteur 65

3.1.2. Gestion de verrouillage des portes et système d'alarme 65

3.1.3. Gestion d'ABS / ESP 66

3.1.4. Gestion de boîte de vitesse : 66

3.2. Simulation de développement du système cyber physique 68

3.3. Interfaces d'application web 71

4. Conclusion 80

Conclusion générale 81

Bibliographies. 82

Liste des figures

Figure I. 1:Capacités de base d'un CPS. 5
Figure I. 2 *: CPS au cœur des 3 types d'Internet [47].* 7
Figure I. 3 *: Idéal conceptuel pour le PLM [11].* 9
Figure I. 4 *: jumeau numérique appliqué à un système de production [12].* 10
Figure I. 5 : Processus d'évolution de l'atelier de production [13]. 11

Figure II. 1: *Taxonomie des diagrammes UML.* 15
Figure II. 2: *Taxonomie des diagrammes SysML.* 17
Figure II. 3*: Quatre piliers de SysML.* 19
Figure II. 4: *Exemple d'un bloc.* 20
Figure II. 5*: Relation entre UML et SysML.* 21
Figure II. 6: *Couches de Méta modélisation.* 22
Figure II. 7: *Architecture MDA [28].* 24
Figure II. 8: *Concepts de base de transformation de modèles.* 24

Figure III. 1 : concept système cyber physique. 29
Figure III. 2 : Démarche globale du travail. 30
Figure III. 3 : Exemple d'un système cyber physique. 33
Figure III. 4*: Méta modèle des jumeaux numériques.* 33
Figure III. 5*: Démarche de transformation.* 36
Figure III. 6 : exemple d'un modèle généré par la transformation M2M. 41
Figure III. 7 : Exemple de format de JSON du jumeau numérique. 42
Figure III. 8 : Architecture d'application. 44
Figure III. 9: Diagramme de cas d'utilisation. 45
Figure III. 10 *: Diagramme de séquence de gestion de la transformation.* 52
Figure III. 11 *: Diagramme de séquence de gestion d'objet.* 53
Figure III. 12 *: Diagramme de séquence de gestion de création des objets.* 54
Figure III. 13 : *Diagramme de séquence de gestion d'interface du jumeau numérique.* 55
Figure III. 14: Options utilisateur dans l'interface du jumeau numérique. 56
Figure III. 15: Intégration avec le système physique. 57

Figure IV. 1*: Annotation des blocs physiques.* 60
Figure IV. 2:Annotation des blocs cyber physiques. 60
Figure IV. 3:Annotation des acteurs. 60
Figure IV. 4: Règle ATL des objets (thing). 61
Figure IV. 5: Règle ATL des fonctionnalités (feature). 62
Figure IV. 6: Règle ATL de la liste de contrôle des accédés. 62
Figure IV. 7*: Règle ACCELEO pour transformer les objets.* 63
Figure IV. 8: Exporte de jar étape 1. 64
Figure IV. 9: Exporte de jar étape 2. 64
Figure IV. 10 *: Diagramme de bloc de la voiture.* 67

Figure IV. 11 : Simulation d'unité de gestion d'ABS / ESP. 68
Figure IV. 12 : Simulation d'unité de contrôle du moteur. 69
Figure IV. 13 : Simulation d'unité de gestion du port. 70
Figure IV. 14 : Simulation d'unité de gestion de la transmission. 71
Figure IV. 15: Interface de la page accueil. 72
Figure IV. 16: *Interface de sélection de modèle.* 73
Figure IV. 17: Interface transformation M2M. 73
Figure IV. 18: *Interface transformation M2M (2).* 74
Figure IV. 19: Interface transformation M2T. 74
Figure IV. 20: Interface transformation M2T. 75
Figure IV. 21: *Interface de gestion des objets 1.* 76
Figure IV. 22: Interface de gestion d'objet 2. 76
Figure IV. 23: Interface de gestion d'objet 3. 77
Figure IV. 24: *Interface création a échoué.* 77
Figure IV. 25: *Interface de création réussie* 78
Figure IV. 26 : Formulaire d'identification des actionneurs. 78
Figure IV. 27: Interface de jumeau numérique 1. 79
Figure IV. 28: *Interface du jumeau numérique après connexion avec le système cyber-physique via Ditto.* 79
Figure IV. 29 *:* Structure de fichiers du jumeau numérique généré. 80

Liste des tableaux

Tableau II. 1 : Approche de transformation de modèles. 25
Tableau III. 1 : Transformation des objets. 38
Tableau III. 2 : Transformation des fonctionnalités. 39
Tableau III. 3 : Transformation d'ACL. 40
Tableau III. 4: Description textuelle de sous cas d'utilisation « Gestion de l'interface de jumeau numérique ». 46
Tableau III. 5 : Description textuelle de sous cas d'utilisation « Générer jumeaux numériques modèle ». 47
Tableau III. 6 : Description textuelle de cas d'utilisation « Générer fichier JSON ». 48
Tableau III. 7 : Description textuelle de cas d'utilisation « Gestion de THINGS ». 48
Tableau III. 8 : Description textuelle du cas d'utilisation « gestion de l'interface de jumeau numérique ». 49
Tableau III. 9: Description textuelle de cas d'utilisation « Test de changement de valeur ». 50
Tableau III. 10: Description textuelle du cas d'utilisation « Exporter interface finale ». 51

Liste des acronymes

ACL : Access Control Lists
BDD : Block Definition Diagram
C : Cyber
CP : Cyber-Physique
CPS : Cyber Physical System
CSS : Cascading Style Sheets
DSML : Domain Specific Modeling Languages
html : Hypertext Markup Language
IDM : Ingénierie Dirigée par les Modèles
IDO : Internet Des Objets
IoO : Internet Des Objets
IoP : Internet Des Personnes
IoS : Internet De Service
LHS : Left Hand Side
M2M : Model to Model
M2T : Model to Texte
MDA : Model Driven Architecture
MDE : Model Driven Engineering
MOF : Meta-Object Facility
NSF: la National Science Fondation
OMT : Object Modeling Technique
P : Physique
PHP : Hypertext Preprocessor
PIM : Platform-Independent Models
PLM : Product Lifecycle Management
PSM : Platform-Specific Models
SysML : Systems Modeling Language
UML : Unified Modeling Language
XML : EXtensible Markup Language

Introduction générale

Les systèmes sont présents, aujourd'hui, dans tous les domaines de l'activité humaine (industrie, transports, communications, construction, etc.). Avec le temps, ces systèmes sont devenus de plus en plus complexes, et par conséquent, leur suivi, leur analyse et leur vérification représentent un enjeu capital.

Le terme « Système cyber-physique » est une traduction directe du terme anglais "cyber-physical system" (CPS) qui correspond à un système intégrant de l'électronique et du logiciel, des capteurs et des actionneurs et dotés de capacité de communication entre eux .Un système cyber-physique est autonome et embarqué de ce fait les éléments mentionnés précédemment. Un CPS interagit avec son environnement dans lequel il prend des données, les traite et au travers d'une boucle de rétroaction contrôle ou influence le processus auquel il est associé. Les CPS sont utilisés pour contrôler et piloter les processus physiques et augmentent ainsi ces processus de fonctionnalités nouvelles. Par ses capacités de communication, un CPS peut agir en collaboration avec d'autres systèmes et/ou échanger des données avec des systèmes distants. La communication peut être filaire via un bus industriel par exemple et/ou sans fil. Lorsqu'un CPS utilise les technologies de communication de l'internet, il devient une brique de base de l'internet des objets.

L'internet des objets, ou IdO (en anglais Internet of Things, ou IoT) est l'interconnexion entre Internet et des objets, des lieux et des environnements physiques. L'appellation désigne un nombre croissant d'objets connectés à internet permettant ainsi une communication entre nos biens dits physiques et leurs existences numériques. Ces formes de connexions permettent de rassembler de nouvelles masses de données sur le réseau et donc, de nouvelles connaissances et formes de savoirs.

Près de la moitié des organisations qui mettent en œuvre l'IoT indiquent qu'elles utilisent déjà ou prévoient d'utiliser des jumeaux numériques en dès que possible.

Un jumeau numérique ou digital twin en anglais, peut être vu comme la modélisation numérique d'un système réel. Cette modélisation peut se faire via une représentation 3D qui reflète en temps réel le comportement du système. Le jumeau numérique peut être particulièrement utile pour l'optimisation, la surveillance et la simulation de systèmes industriels.

Introduction générale

A mesure du développement de la numérisation des processus industriels, les jumeaux numériques sont de plus en plus répandus, ce concept existe depuis un certain temps, mais il n'a pris son essor que ces dernières années avec la croissance de l'internet des objets (IoT), du big data, et de l'industrie 4.0 [1].

Dans ce contexte, nous proposons une démarche pour le développement des jumeaux numériques concernant des systèmes cyber-physique. Celle-ci est basée sur une ingénierie système dirigée par les modèles. L'idée consiste à exploiter la modélisation du système cyber-physique à construire pour générer son jumeau numérique. Nous proposons donc, de passer d'une modélisation SysML du système cyber-physique vers une représentation du jumeau numérique. La plateforme éclipse ditto [2] sert de support pour la création des jumeaux numériques.

Afin de faciliter la tâche aux développeurs des jumeaux numériques, nous proposons une application web supportant la démarche proposée. Cette application automatise le processus de passage de la modélisation SysML à la représentation des jumeaux numériques et aide les développeurs à la création des jumeaux numériques dans la plateforme éclipse ditto.
La démarche et l'application ont été évaluées à travers une étude de cas pour montrer leur utilisation pour le développement des jumeaux numériques.

Le reste de ce document détaille l'état de l'art concernant ce sujet et explique la démarche proposée et l'implémentation de l'application de création du jumeau numérique. Il est organisé en quatre chapitres comme suit :

Chapitre 1 : Dans ce chapitre, nous présentons les concepts relatifs aux systèmes cyber physique, à l'industrie 4.0, à l'internet des objets et leurs relations avec les jumeaux numériques (digital twin).

Chapitre 2 : Dans ce chapitre, nous présentons les langages et les concepts de base utilisés, SysML, UML, Ingénierie Dirigée par les Modèles (IDM), et les différentes notions qui servent pour le projet.

Chapitre 3 : Dans ce chapitre, nous décrirons notre démarche pour la génération des jumeaux numériques, en spécifiant les différentes étapes pour arriver à la phase de déploiement.

Chapitre 4 : Ce dernier chapitre présente l'application web que nous avons développé, et illustre notre méthodologie pour la génération des jumeaux numériques par un cas d'étude pratique.

CHAPITRE I : Système cyber-physique et jumeau numérique (digital twin).

1. Introduction

Le numérique envahit notre quotidien. Il se développe très rapide car il ne trouve aucun détracteur qui freine son expansion. Aujourd'hui il n'existe aucun service, aucun domaine de la société qui ne soit épargné. Le numérique se retrouve partout : dans l'agriculture, l'industrie, l'enseignement, l'administration, la santé, la défense, les transports, l'écologie, l'habitat et bien d'autres domaines encore.

Ce formidable élan technologique est un moteur de croissance économique et sociétal qui modifie la façon de penser notre vie quotidienne. Non seulement, il augmente les possibilités de diffusion et d'analyse de l'information mais aussi, il ouvre la porte de l'innovation pour la création de nouveaux produits et services.

Dans ce chapitre, nous présenterons quelques concepts de base relatifs aux systèmes cyber-physique et à l'industrie 4.0 qui ont été développés dans de nombreux domaines et leurs relations avec jumeaux numériques.

2. Industrie 4.0

La première révolution industrielle s'est déroulée au milieu et à la fin du 18ème siècle. Le point de départ fut l'introduction d'installations de production mécanique utilisant l'eau et la vapeur comme le premier métier à vapeur en 1784.

Environ un siècle après la première révolution industrielle, l'introduction de la production de masse à forte intensité de main-d'œuvre à l'aide de l'énergie a marqué le début de la deuxième révolution industrielle à la fin du 19ème siècle.

Alors que la première et deuxième révolution parle d'une mécanisation croissante de la production, la troisième révolution industrielle, qui se situe au début des années soixante du 20ème siècle, représente une nouvelle dimension dans l'automatisation de la production. Elle se caractérise par l'utilisation de l'électronique ainsi que des technologies de l'information et de la communication (TIC), qui permettent l'automatisation et la production en série plus variée.

Les visions de l'Industrie 4.0 incluent la personnalisation des produits, la production flexible, l'optimisation de la chaîne de valeur en temps réel, les systèmes de gestion adaptative des effectifs, les systèmes d'apprentissage mobiles, l'interaction homme-machine multimodale et l'usine intelligente [3].

2.1.Présentation de l'Industrie 4.0

Le terme ‘Industrie 4.0’ regroupe un ensemble de technologies et de concepts liés à la réorganisation de la chaîne de valeur [4]. Ainsi, la vision relative à l’Industrie 4.0 prend appui sur la communication en temps réel pour surveiller et agir sur les systèmes physiques. Les systèmes communiquent et coopèrent entre eux, mais également avec les humains, pour décentraliser la prise de décisions. L’Industrie 4.0 met donc l'accent sur la connectivité, favorisant ainsi le développement de nouveaux processus, produits et services. Son déploiement requiert une intégration de différents savoir-faire propres aux technologies numériques.

2.2.Définition d’Industrie 4.0

Sur la base des conclusions de la revue de la littérature, nous définissons Industrie 4.0 comme suit : Industrie 4.0 est un terme collectif désignant les technologies et les concepts d’organisation de la chaîne de valeur. Au sein des usines intelligentes d’industrie 4.0 à structure modulaire, CPS surveille les processus physiques, crée une copie virtuelle du monde physique et prend des décisions décentralisées Ces systèmes reposent sur une communication directe entre eux de manière instantanée, ils dépendent donc de l'Internet des objets pour la communication. les CPS communiquent et coopèrent les uns avec les autres et avec les humains en temps réel. Via l’IOS (internet of services), les participants de la chaîne de valeur offrent et utilisent des services internes et inter organisationnels [5].

3. Système cyber physique

La première définition que l’on peut trouver des (CPS) date de 2006 [6] lors de travaux avec la National Science Fondation 1 (NSF) américaine : les CPS intègrent des processus physiques et computationnels. Des ordinateurs et réseaux embarqués surveillent et contrôlent les processus physiques, généralement avec des boucles de rétroaction où les processus physiques affectent les calculs et vice versa. En d’autres mots, les CPS utilisent des computations et de la communication profondément intégrée et interagissant avec les processus physiques afin de produire de nouvelles capacités du système. Un CPS peut être considéré aussi bien à une petite échelle (e.g. pace maker) qu’à de grandes échelles (un réseau national de distribution d’énergie) [6]. De cette définition, nous retiendrons que les CPS sont des systèmes résultants d’un

couplage entre des processus physiques et des capacités de calcul, de communication et de commande.

Dans les travaux de [7], l'auteur présente une synthèse de définitions et de classifications des CPS ainsi qu'une analyse quantitative montrant la prolifération des travaux autour de ce nouveau concept. Il explique que la terminologie « cyber », venant de la cybernétique[8], où la cybernétique est l'étude des mécanismes d'information des système complexes, prend tout son sens avec les CPS. Ainsi, des auteurs comme [6] ont donné une classification du niveau d'intelligence d'un CPS :

–Niveau C1 : capable de se connecter à un réseau.

–Niveau C2 : traiter de l'information et la retranscrire.

–Niveau C3 : connaître son environnement.

–Niveau C4 : avoir des capacités cognitives (diagnostiquer et agir) et le niveau supérieur.

–Niveau C5 : le CPS peut s'adapter en cas de perturbation pour retrouver l'équilibre.

Parmi les composants nécessaires dans un système cyber physique est :

–Capteur : Dispositif permettant de capter un phénomène physique et de le restituer sous forme de signal.

–Actionneur : Organe d'un système agissant sur une machine de manière à modifier son état ou son comportement.

Cette interconnexion de systèmes, provient du fait qu'un CPS comprend à la fois des moyens de contrôle, de calcul, mais également de communication, ce qui peut être synthétisé par la Figure I.1

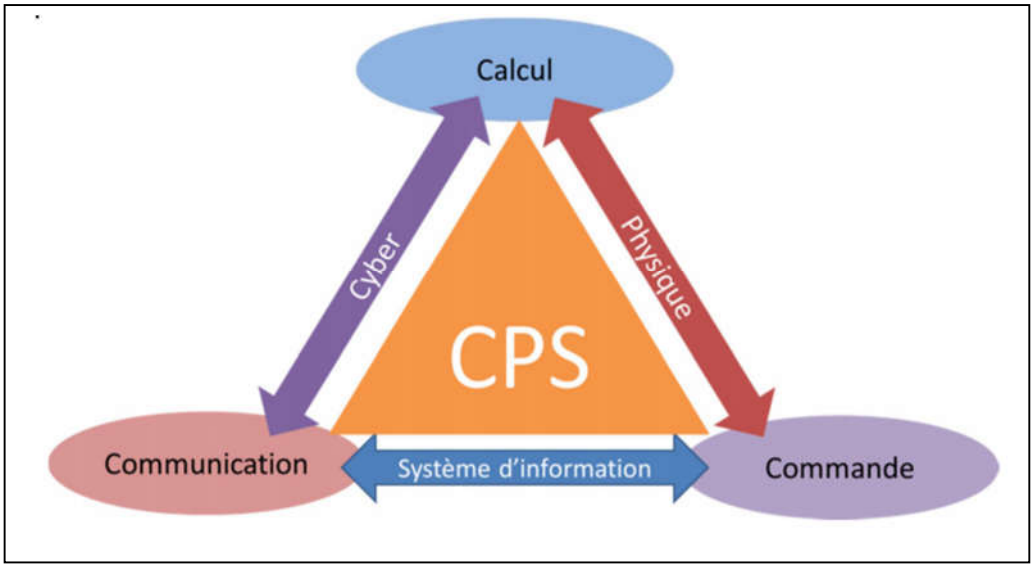


Figure I. 1*:Capacités de base d'un CPS.*

L'interconnexion des composants du système, nous amène à un nouveau concept appelé Internet des objets, technologie qui a permis au système cyber physique de progresser et de contrôler toutes les parties du système.

4. Internet des objet IoT

Le terme Internet des objets (IoT) a été introduit par Kevin Ashton pour décrire comment l'internet des objets peut être créé par l'association des technologies d'auto-identification et des capteurs aux objets de tous les jours. Au fil du temps, la definition a évolué « the IoT as a network of entities that are connected through any form of sensor, enabling these entities, which we term as Internet-connected constituents to be located, identified, refund even operated upon" [9].

La prolifération des objets connectés à Internet permet d'interagir entre les dispositifs et les objets, mais aussi entre les objets et les personnes, offrant ainsi de nouvelles opportunités d'applications. En effet, les personnes peuvent se connecter directement à des éléments (tels que des téléphones portables, des enregistrements de santé électroniques, etc.

La figure I.2 propose l'architecture de la mise en œuvre de l'IoT. Cette figure présente un cadre conceptuel permettant de classer l'IoT en trois catégories, à savoir internet des objets (IoO), Internet de service (IoS), Internet des personnes (IoP).

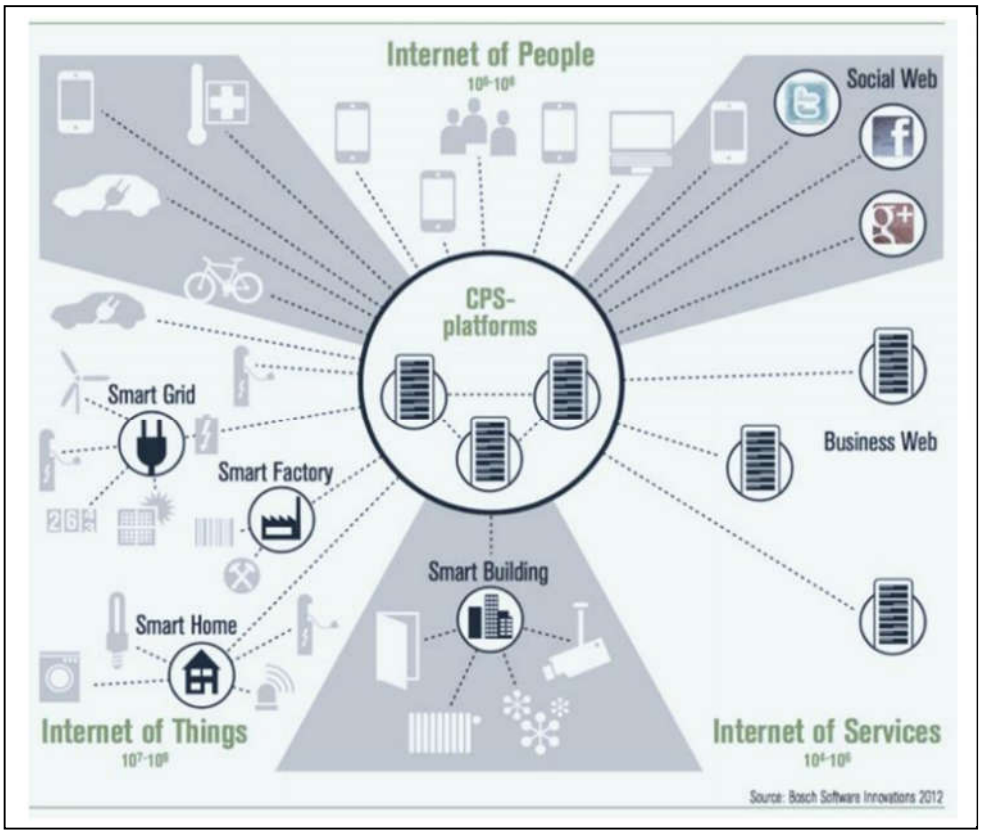


Figure I. 2 *: CPS au cœur des 3 types d'Internet [47].*

5. Jumeaux numériques

L'essor du jumeau numérique doit son succès à l'amélioration des capacités de simulation de systèmes en temps réel, d'analyse et de stockage des données, au développement de l'internet des objets connectés et de l'intelligence artificielle.

5.1. Définition

L'idée du jumeau numérique est une idée originale de Dr. Michael Grieves et John Vickers (utilisés à l'origine en 2003 lors d'un cours à l'Université du Michigan), experts en fabrication et en gestion du cycle de vie des produits (PLM). La notification de base est que, pour chaque produit physique, il existe une contrepartie virtuelle qui peut parfaitement imiter les attributs physiques et les performances dynamiques de son jumeau physique. Le jumeau virtuel existe dans un environnement simulé qui peut être contrôlé de manière très précise et qui ne peut pas être facilement dupliqué dans le monde réel, tel que l'accélération du temps afin que des années d'utilisation puissent être simulées en une fraction du temps. Ces modèles et simulations hyper-précis offrent aux ingénieurs et aux concepteurs de produits des informations inégalées tout au

long du cycle de développement du produit. Malgré tout, les jumeaux numériques ne représentent pas seulement une évolution des modèles numériques, même si leur objectif est similaire : obtenir des produits de meilleure qualité et un meilleur support produit à un coût moindre et avec moins d'effort [10].

L'une des diapositives de la présentation, comme le montre la Figure I.3, avait comme titre « Conceptuel Ideal for PLM » ce qui deviendra par la suite « Digital Twin ». Ce concept reflète en effet une représentation dynamique d'un système composé de deux systèmes, l'un réel et l'autre virtuel et que ces deux derniers sont liés tout au long du cycle de vie du système [11].

Comme le montre la Figure I.3, ce concept est composé de trois parties : Le produit physique dans l'environnement réel, le produit virtuel dans un environnement virtuel et la connexion des données et de l'information qui relie les deux environnements [11].

Le concept du jumeau numérique (Digital Twin) consiste à analyser continuellement le flux de données récolté de la partie matérielle. En effet, des capteurs placés tout au long du processus vont acquérir les données relatives à l'évolution du produits, ces données vont ensuite être transmises en temps réel à une plateforme numérique qui, à son tour, effectue une analyse pratiquement en temps réel afin d'optimiser ce processus de manière transparente, et de façon continue. Ceci va permettre entre autres d'analyser les données, de contrôler le système et même de prédire une éventuelle défaillance permettant ainsi d'agir et de réparer de façon proactive [12].

Le jumeau numérique (Digital Twin) va aussi servir à faire fonctionner des machines et à planifier les ressources ce qui va augmenter l'efficacité opérationnelle et améliorer le développement des produits.

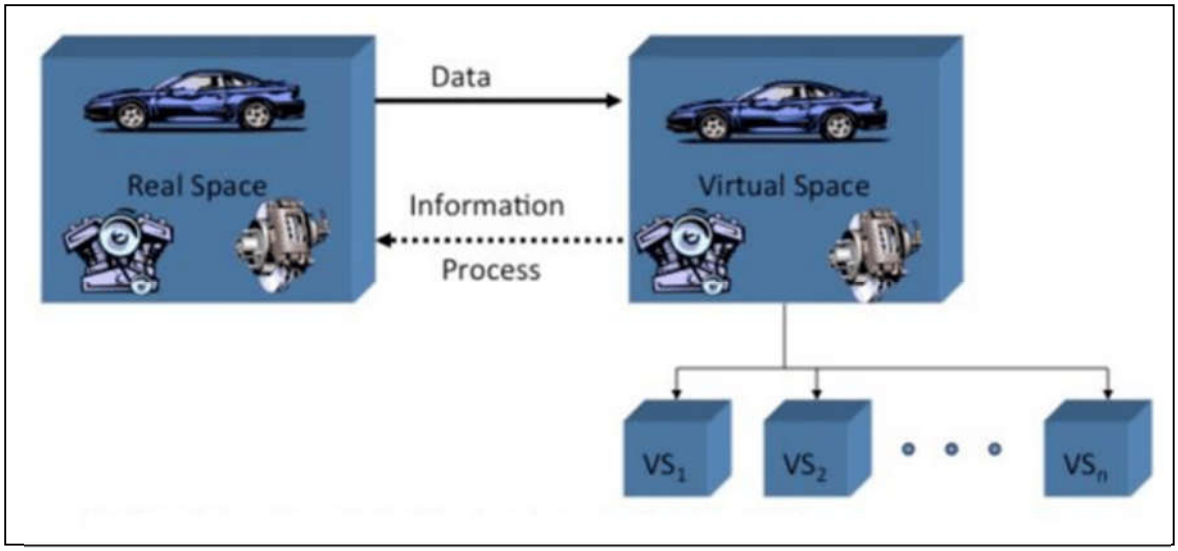


Figure I. 3 *: Idéal conceptuel pour le PLM [11].*

La Figure I.4 met en évidence la rotation du flux des données entre, d'un côté, la partie réelle (PHYSICAL) qui se compose de capteurs et d'actionneurs, et de l'autre côté, la partie numérique ou virtuelle (DIGITAL) qui est constamment mise à jour.

Les éléments cités ci-dessus sont expliqués dans ce qui suit :

Capteurs : Distribués tout au long de la chaîne, ils créent des signaux qui permettent au jumeau numérique de reproduire le comportement du réel dans les mêmes conditions.

Données : les informations récoltées par les capteurs, mélangées avec les informations de l'entreprise telles que les nomenclatures, les systèmes de d'entreprise et les spécifications de conception. Ces données peuvent aussi contenir d'autres types d'informations qui dépendent du contexte dans lequel on se trouve.

Intégration : les deux mondes (réel et virtuel) communiquent entre eux via des technologies d'intégration (tel que les ponts, les interfaces de communication et de sécurité).

Analytique : Ce sont des techniques utilisées pour analyser les informations en utilisant des algorithmes, des simulations et des outils de visualisation qui vont aboutir à d'éventuelles idées.

Actionneurs : Le jumeau numérique (Digital Twin) envoie ces idées sous forme d'informations numériques (ou messages) aux actionneurs (opérateurs), qui vont effectuer l'action physique.

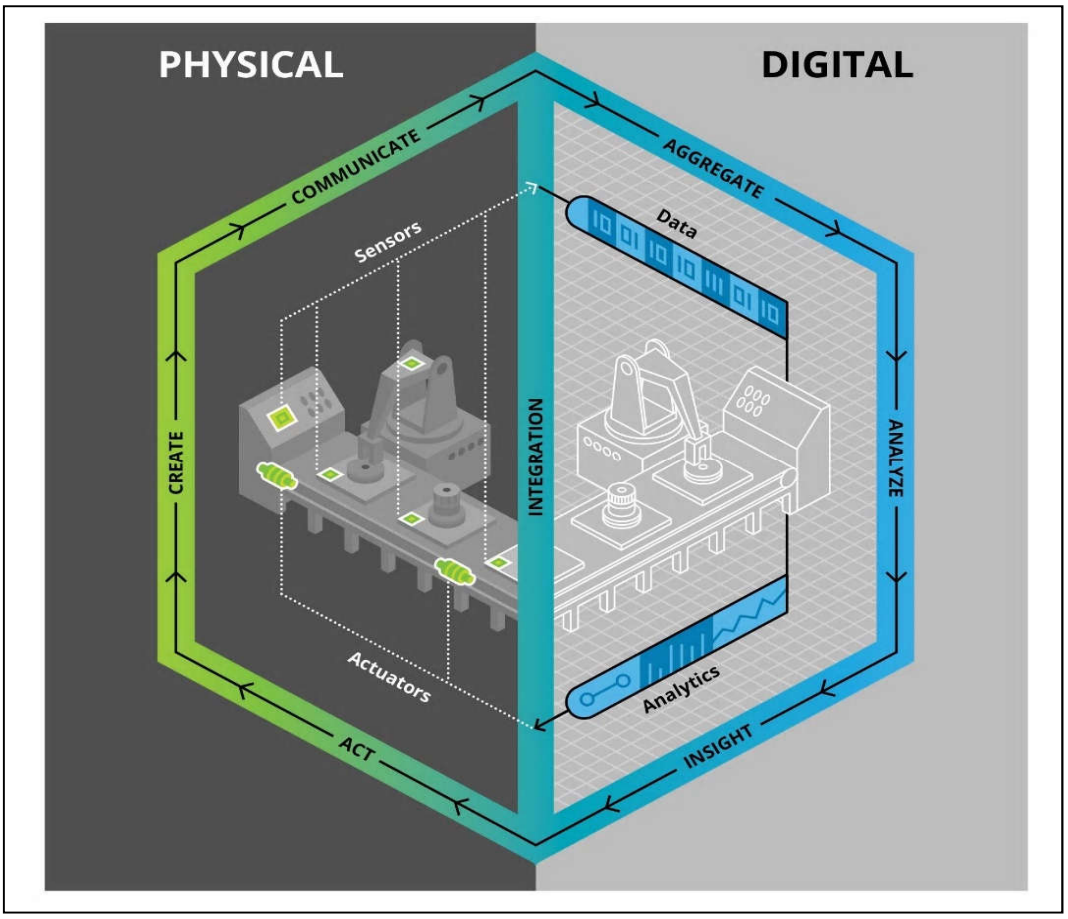


Figure I. 4 *: jumeau numérique appliqué à un système de production [12].*

5.2.Pourquoi le jumeau numérique (Digital Twin)

Rappelons que le but de l'industrie 4.0 et de toutes les technologies qui la compose (entre autres le Digital Twin), est de répondre à différents besoins. Ces derniers sont imposés par l'évolution technologique et un environnement externe de plus en plus rude (concurrence, clientèle exigeante, parties prenantes…). Par conséquent, une maîtrise parfaite des chaînes de production est nécessaire, chose qui n'est pas évidente car ces dernières sont de plus en plus complexes et les maîtrises est une chose humainement impossible. Avant de continuer, il faudra peut-être se poser la question sur le vrai sens du terme « maîtriser » : d'un point de vue technique « maîtriser un système » veut dire : connaître toutes les données relatives à ce dernier pour pouvoir agir sur lui sans pour autant affecter son comportement normal. Le processus d'évolution de l'atelier de production (Figure I.5) reflète clairement la volonté des ingénieurs et des chercheurs à optimiser et à mieux comprendre les systèmes de production, et ce en ayant recours à la simulation et à la modélisation (Virtual Space) qui sont des outils devenus indispensables à la compréhension de n'importe quel système, qui dans ce cas-là, est un atelier

de production noté « espace physique » (Physical Space). Sur la Figure I.5, quatre étapes sont illustrées pour montrer le processus d'évolution de l'atelier de production [13]. Au tout début, faute de moyens d'information efficaces, la production dans l'atelier dépend entièrement de l'espace physique, ce qui conduit à une efficacité, une précision et une transparence faible. Ensuite, avec le développement des technologies de l'information, les systèmes assistés par ordinateur commencent à être appliqués dans la production, mais comme les méthodes d'interaction sont faibles, l'espace virtuel est en décalage avec le physique. Au troisième stade, bénéficiant des technologies de la communication (des capteurs, de l'IoT, etc.), il existe une interaction entre les deux espaces. À l'avenir, avec les développements continus des nouvelles technologies de l'information, l'espace virtuel jouera progressivement un rôle tout aussi important avec la connexion physique et la connexion bidirectionnelle qui sera également améliorée, ce qui favorise la poursuite de la convergence [13].

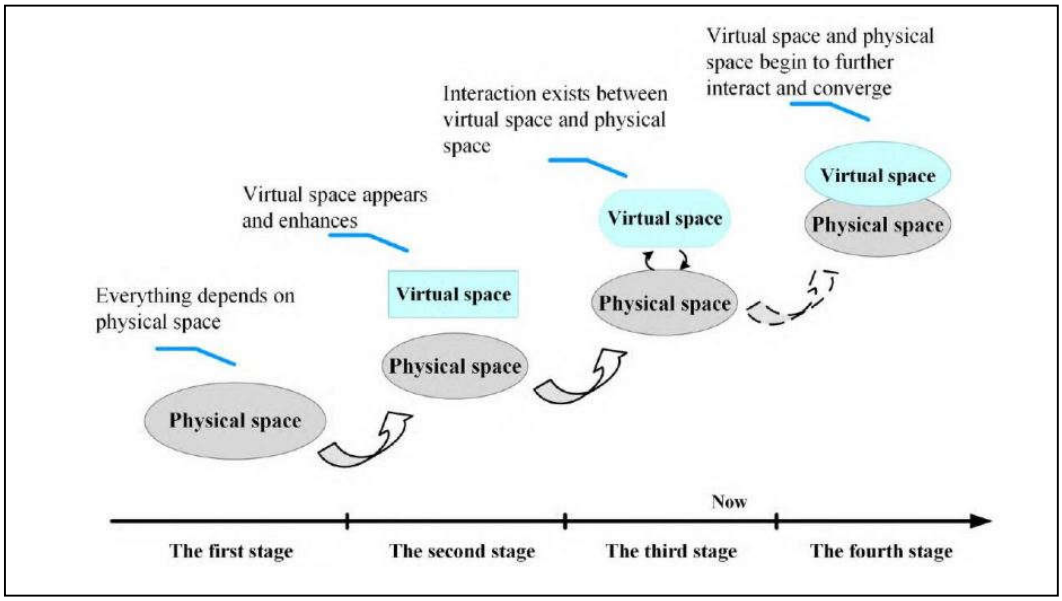


Figure I. 5 : *Processus d'évolution de l'atelier de production [13].*

5.3.Utilisations du jumeau numérique

Le jumeau numérique (Digital Twin) s'avère être un outil puissant qui va permettre d'accompagner et d'aider les organisations, et plus particulièrement les ingénieurs. Cet outil va leurs permettre d'apprendre plus, plus rapidement et briser les vieilles limites entourant l'innovation des produits, les cycles de vie complexes et la création de valeur [24], voici comment [25] :

5.3.1. Visibilité :

Le jumeau numérique (Digital Twin) permet une visibilité dans les opérations de machines ainsi que dans les grands systèmes interconnectés tel qu'une usine de fabrication ou un aéroport.

5.3.2. Prédiction :

En utilisant les différentes techniques de modélisation, le jumeau numérique peut être utilisé pour prédire l'état futur d'un système.

5.3.3. Analyse :

Grâce à des interfaces bien conçues, il est facile d'interagir avec le jumeau numérique (Digital Twin) et de poser les questions de simulation au modèle pour simuler diverses conditions qui ne sont pas pratiques à créer dans la vraie vie.

5.3.4. Outil de documentation et de communication :

Le modèle du jumeau numérique (Digital Twin) peut être utilisé comme un mécanisme de communication et de documentation qui peut être à son tour utilisé pour comprendre et expliquer les comportements d'une machine ou d'un ensemble de machines.

5.3.5. Outil d'aide à la décision :

Le jumeau numérique (Digital Twin) peut aussi interagir et communiquer avec les différentes autres applications présentes dans l'organisation, telles que les applications de gestion (ERP…) pour atteindre les résultats opérationnels dans le contexte des opérations de la chaîne d'approvisionnement : fabrication, approvisionnement, entreposage, transport et logistique, service, etc.

6. Conclusion

Après avoir identifié les jumeaux numériques et sa relation avec l'IDO, avec les CPS et son rôle dans le développement de l'industrie dans sa quatrième génération, l'utilité des jumeaux numériques et leurs avantages sont devenus clairs. Ceci nous pousse à penser à l'amélioration du processus de leur développement. C'est l'objectif de notre travail qui va être développé à travers la proposition d'une démarche d'ingénierie systèmes basée sur les modèles utilisant le langage de modélisation SysML.

CHAPITRE II : UML et SysML

1. Introduction

Après avoir présenté les notions des jumeaux numériques et leur utilité et la nécessité d'améliorer leur développement en se basant sur l'ingénierie système basée sur les modèles, nous présentons, dans ce chapitre, le langage de modélisation SysML, un profil du langage de modélisation UML et l'ingénierie dirigée par les modèles.

2. Langage de modélisation UML

Le langage de modélisation unifié (UML) propose une modélisation visuelle à usage général. UML est le résultat de la fusion de trois notations majeures : la méthodologie de Grady Booch [16] pour décrire un ensemble d'objets et leurs relations, la technique de modélisation des objets de James Rumbaugh (OMT [17]), et la méthodologie de cas d'utilisation d'Ivar Jacobson [48].

Bien qu'UML ait été conçu à l'origine pour spécifier, visualiser et documenter des systèmes logiciels, il peut également être appliqué à divers autres domaines tels que l'organisation des sociétés et les processus métiers. UML à de nombreux avantages, il est largement accepté par de nombreux leaders de l'industrie, est non exclusif et extensible. Il est également commercialement soutenu par de nombreux outils et manuels. La norme UML a été révisée plusieurs fois et de nombreuses versions ont été publiées jusqu'à la version 2.5.1 publiée en 2018[18].

2.1. Structuration du langage UML

Les diagrammes UML sont classés en deux catégories principales : structurelle et comportementale. Cette dernière catégorie comprend une catégorie de sous-ensembles appelée diagrammes d'interaction. La taxonomie des diagrammes est illustrée à la figure

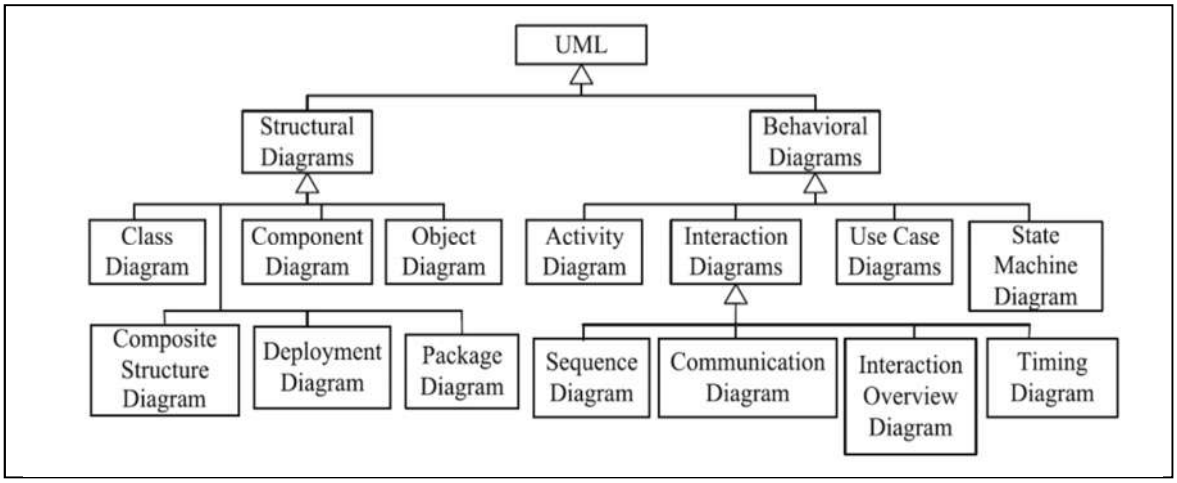


Figure II. 1: *Taxonomie des diagrammes UML.*

Les diagrammes structurels modélisent les aspects statiques et les caractéristiques structurelles du système, tandis que les diagrammes comportementaux décrivent le comportement dynamique du système. Les diagrammes structurels incluent le diagramme de classe, le diagramme de composants, le diagramme de structure composite, le diagramme de déploiement, le diagramme d'objet, et le diagramme de packages.

Les diagrammes comportementaux incluent le diagramme d'activité, le diagramme de cas d'utilisation et le diagramme d'état machine. Les diagrammes d'interaction font partie de de la catégorie des diagrammes comportementaux car ils soulignent l'interaction entre les composants modélisés. Ils comprennent le diagramme de communication, le diagramme global d'interactions, le diagramme de séquence et le diagramme de temps.

Les nouveaux diagrammes proposent dans UML 2.x [22] sont le diagramme de structure composite, le diagramme global d'interactions et le diagramme de temps. En outre d'autres diagrammes ont été étendus et modifiés depuis UML 1.x, à savoir, le diagramme d'activité, le diagramme d'état machine, le diagramme de séquence et le diagramme de communication [22].

2.2.Profil UML

UML définit un concept de profil spécifique qui fournit un mécanisme d'extension générique pour construire des modèles UML dans un domaine particulier. Le profilage UML permet un affinage de la sémantique des éléments UML standard d'une manière strictement additive et sans introduire de contradictions sémantiques [23].

Les principaux mécanismes de profilage UML sont les stéréotypes et les valeurs marquées. Ces mécanismes sont appliqués à des éléments de modèle spécifique tels que les classes, les attributs, les opérations et les activités afin de les adapter à différentes fins. Les mécanismes sont définis comme suit :

– Les stéréotypes UML sont utilisés pour étendre le vocabulaire UML en créant de nouveaux éléments de modelé dérives des existants.

– Les stéréotypes ajoutent des propriétés spécifiques adapté à un domaine de problème spécifique et sont utilisés pour la classification ou le marquage des blocs de construction UML.

Les annotations sont utilisées pour spécifier des valeurs de mot-clé. Elles permettent l'extension des propriétés d'un bloc de construction UML et la création de nouvelles informations dans la spécification d'un élément défini pour des éléments de modelé existants ou pour stéréotypes individuels. De plus, des valeurs balisées peuvent être utilisées pour spécifier des propriétés es qui sont pertinentes pour la génération de code ou la gestion de la configuration. Divers profils UML ont et émis par l'OMG afin de personnaliser UML pour des domaines particuliers (par exemple, l'aérospatiale, la santé, la finance, les transports, l'ingénierie des systèmes) ou des plates-formes (par exemple, J2EE, .NET) [23].

3. SysML

Le langage de modélisation des systèmes (SysML [17]) est un langage dédié aux applications d'ingénierie des systèmes. C'est un profil UML qui non seulement utilise un sous-ensemble d'UML 2.1.1 mais fournit également des extensions supplémentaires pour mieux s'adapter aux besoins spécifiques de l'ingénierie des systèmes.

Il est destiné à aider à spécifier et structurer des systèmes complexes et de leurs composants afin de permettre leur analyse, conception, vérification et validation. Ces systèmes peuvent être constitués de composants hétérogènes tels que du matériel, des logiciels, des informations, des processus, du personnel et des installations.

SysML englobe des capacités de modélisation qui permettent de représenter les systèmes et leurs composants en utilisant :

– La composition, classification et interconnexion des composants structuraux ;

– Les comportements, notamment les flux d'activité, les scénarios d'interaction et la transmission de messages ainsi que les comportements réactifs de pendants de l'état.

– L'allocation d'un élément d'un modèle a un autre, comme les fonctions aux composants, composants logiques aux composants physiques et logiciels au matériel.

– Les contraintes sur les valeurs de propriété du système telles que la performance, la fiabilité et les propriétés physiques.

– Les hiérarchies et les dérivations ainsi que leurs relations avec d'autres éléments du modèle [18].

3.1. Structuration du langage SysML

Les diagrammes SysML [17] définissent une syntaxe concrète qui décrit comment les concepts SysML sont visualise graphiquement ou textuellement. Dans la spécification SysML, cette notation est décrite dans des tableaux qui montrent la cartographie des concepts de langage en symboles graphique sur les diagrammes [23]. La taxonomie des diagrammes est illustrée à la figure II.2.

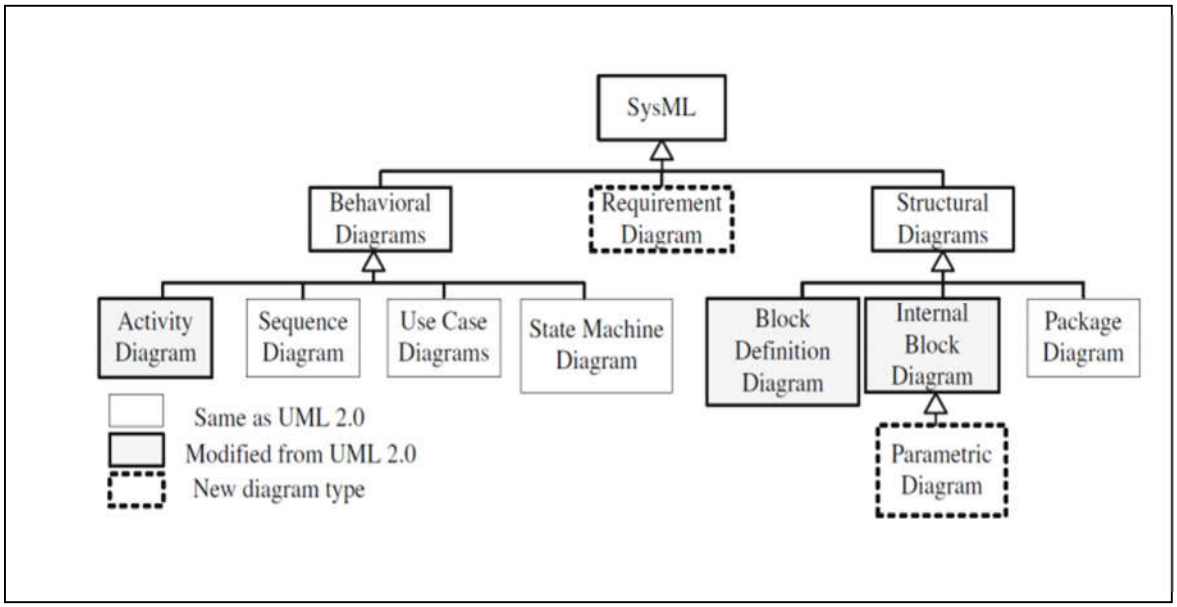


Figure II. 2: *Taxonomie des diagrammes SysML.*

SysML réutilise complètement certains diagrammes d'UML 2.1 [23] :

- Diagramme de cas d'utilisation.
- Diagramme de séquence.
- Diagramme d'état machine.
- Diagramme de package.

En outre, deux nouveaux diagrammes sont ajoutés :

- Diagramme d'exigences.
- Diagramme paramétrique.

De plus, d'autres diagrammes UML sont réutilisés sous forme étendue :

- Diagramme d'activité (étend le diagramme d'activité UML).
- Diagramme de définition de bloc (étend le diagramme de classes UML).
- Diagramme de bloc interne (extension du diagramme de structure composite UML).

3.2. Quatre piliers de SysML

Sur le plan des concepts, le langage SysML repose sur quatre ensembles fondamentaux, qui constituent les quatre piliers de SysML illustrés ci-dessous (Figure II.3) :

– La notion de structure. Pour représenter la structure, on utilise la notion de bloc, qui permet de modéliser tout élément concret : système, sous systèmes, composants logiciels ou matériels, mais aussi acteurs, systèmes externes, ou encore éléments de matière, d'énergie ou d'information échanges. Les diagrammes correspondants sont les diagrammes de définition de bloc et le diagramme de bloc interne.

– La notion de comportement, pour la modélisation dynamique ou comportement- tale du système ou d'un constituant. Les diagrammes correspondants sont les diagrammes d'activités et le diagramme d'états (et au besoin de séquence).

– La notion d'exigence. Le diagramme correspondant est le diagramme d'exigences.

– Les équations et contraintes régissant le système ou ses constituants. Le diagramme correspondant est le diagramme paramétrique [24].

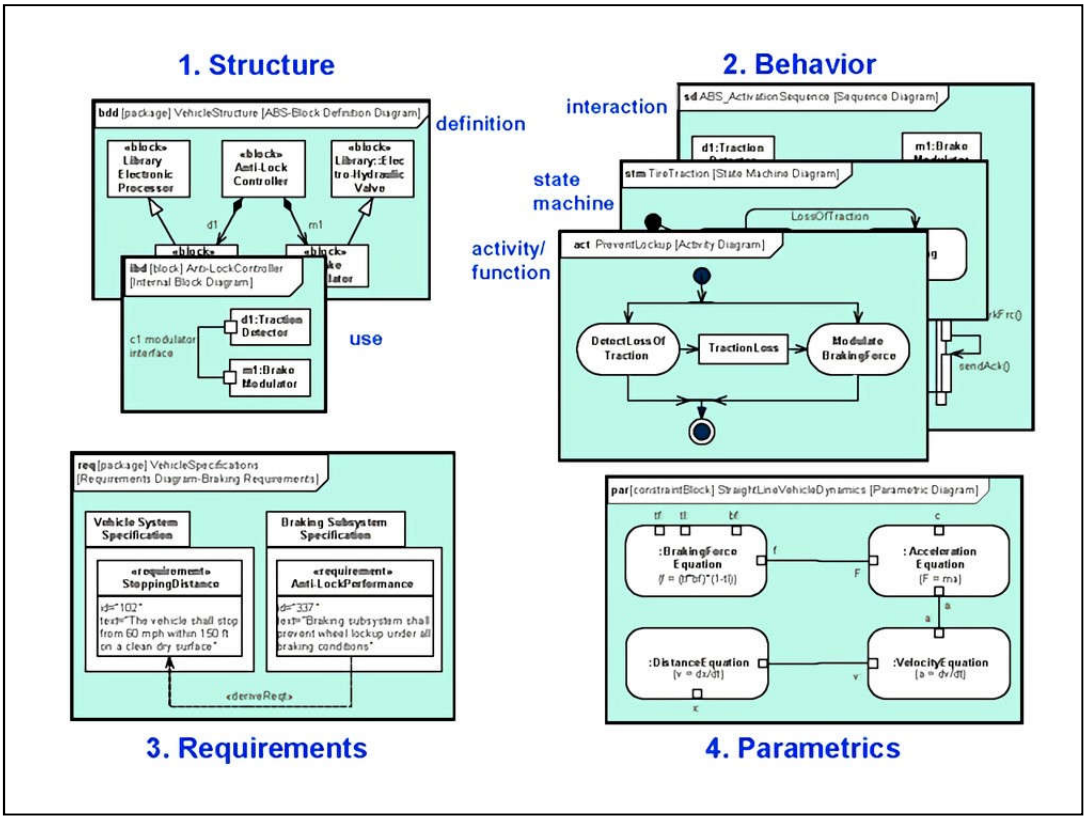


Figure II. 3*: Quatre piliers de SysML.*

3.3. Diagramme de bloc

Le diagramme de définition de bloc (BDD), est un diagramme copie à partir d'UML avec quelques modifications. Il repose sur le diagramme de classes UML, avec l'exclusion de certaines capacités, telles que certaines formes d'associations spécialisées. Sur d'autre part, il a modifié certains concepts tels que la notion de classe a été remplacée par la notion de bloc. Il a également ajouté de nouveaux concepts tels que les ports des blocs. Il peut se référer à une partie matérielle ainsi qu'à une partie logicielle, il peut aussi représenter une personne qui utilise ou interagit avec ce système.

Chaque bloc a un nom, un ensemble de valeurs, des propriétés, des blocs références, des parties, des opérations qui ont des préconditions et des postpositions qui sont liées à des exigences dans notre cas.

En SysML on distingue 3 types de propriétés :

- La propriété de type « Part » qui est une partie intégrante du bloc elle-même caractérisée par un bloc. Par exemple : « roueAvantDroite:Roue »
- La propriété de type « Reference » qui décrit un bloc qui ne fait pas partie intégrante du bloc. Toutes les propriétés qui ne sont pas des compositions.

- La propriété de type « Value » qui définit une valeur quantifiable avec son unité, sa dimension, ou sa distribution statistique comme par exemple :

pressionPneu:psi=30 ou « uniform » {min=28, max32}pressionPneu:psi

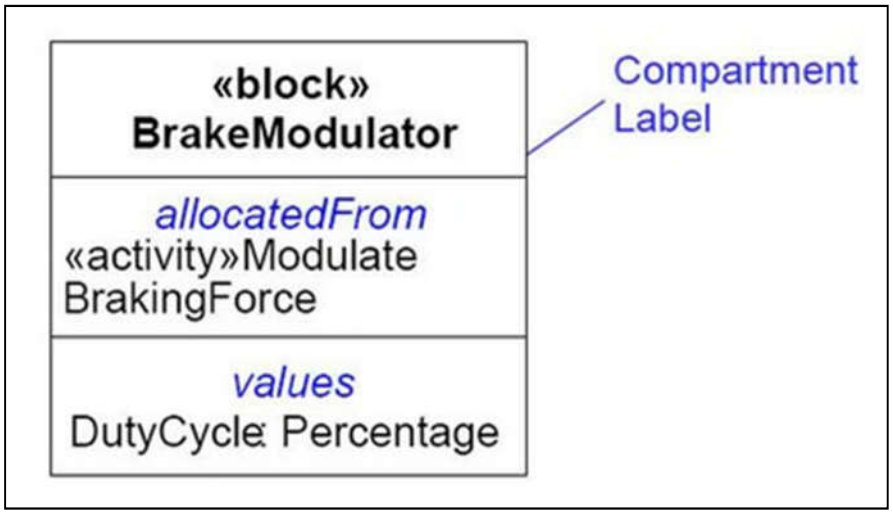


Figure II. 4: *Exemple d'un bloc.*

3. Relation entre UML et SysML

SysML réutilise un sous-ensemble de UML 2.1 [22], appelé "UML4SysML", qui représente environ la moitié du langage UML. Une partie importante de l'UML des concepts ont été écartés car ils n'étaient pas considérés comme pertinents pour les besoins du modelé de SE. Certains des diagrammes utilisés ont été inclus comme dans UML 2.1.1 [22].

Ceux-ci incluent la machine d'état, la séquence et les diagrammes de cas d'utilisation. D'autres diagrammes ont été étendus, tels que les diagrammes d'activité, afin de répondre aux besoins spécifiques des SE. De plus, SysML a omis certains diagrammes UML, à savoir d'objet, de composant, de déploiement, de communication, de temps et d'interactions.

Les diagrammes de structure dont diagramme de Classe et composition ont été fondamentalement modifiés et remplacés par des diagrammes de définition de blocs et diagrammes de blocs internes. Ces extensions sont basées sur le mécanisme de profilage UML standard, qui inclut les stéréotypes UML, extensions de diagramme UML et les bibliothèques de modelés. Le mécanisme de profilage a été choisi sur d'autres mécanismes d'extension afin de tirer parti des outils existants basés sur UML pour la modélisation de systèmes.

En outre, SysML ajoute deux nouveaux diagrammes, ceux-ci étant diagrammes d'exigences et diagrammes paramétriques, et intègre de nouvelles capacités de spécification liens tels que l'allocation. La relation entre UML et SysML est illustrée dans la figure II.5.

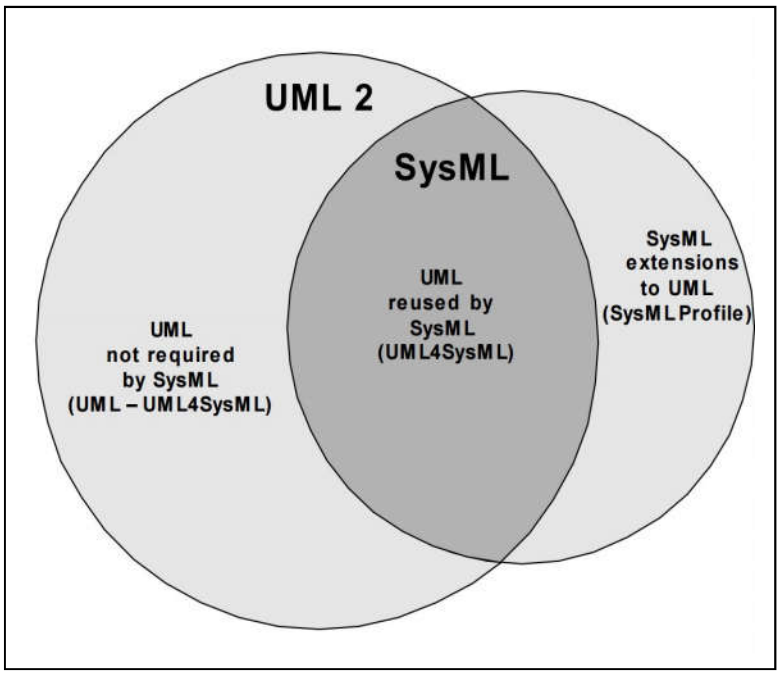


Figure II. 5*: Relation entre UML et SysML.*

4. Ingénierie dirigée par les modèles (IDM)

L'ingénierie dirigée par les modèles (IDM), ou Model Driven Engineering (MDE) en anglais, a permis plusieurs améliorations significatives dans le d'enveloppement de systèmes complexes en permettant de se concentrer sur une préoccupation plus abstraite que la programmation classique. Il s'agit d'une forme d'ingénierie générative dans laquelle tout ou partie d'une application est engendrée à partir de modèles.

Un modelé est une abstraction, une simplification d'un système qui est suffisante pour comprendre le système modélisé et répondre aux questions que l'on se pose sur lui. Un système peut être décrit par différents modèles liés les uns aux autres.

L'idée phare est d'utiliser autant de langages de modélisation différents (Domain Specific Modeling Languages – DSML) que les aspects chronologiques ou technologiques du développement du système le nécessitent. La définition de ces DSML, appelée Meta modélisation, est donc une problématique clé de cette nouvelle ingénierie.

Par ailleurs, afin de rendre opérationnels les modèles (pour la génération de code, de documentation et de test, la validation, la vérification, l'exécution, etc.), une autre problématique clé est celle de la transformation de modelé.

En résumé, IDM couvre le cycle de vie complet de l'application, elle est basée sur deux principaux concepts : la Méta modélisation et la transformation de modèles qui sont toujours des domaines de recherche [24].

4.1. Méta modélisation

La Méta modélisation est une activité qui consiste à définir le Méta modèle d'un langage de modélisation. Il s'agit non seulement de produire des Méta modèles mais aussi de définir la sémantique du langage, de mettre en œuvre des analyseurs, des compilateurs, des générateurs de code et plus généralement, à construire un ensemble d'outils exploitant les Méta modèles [25].

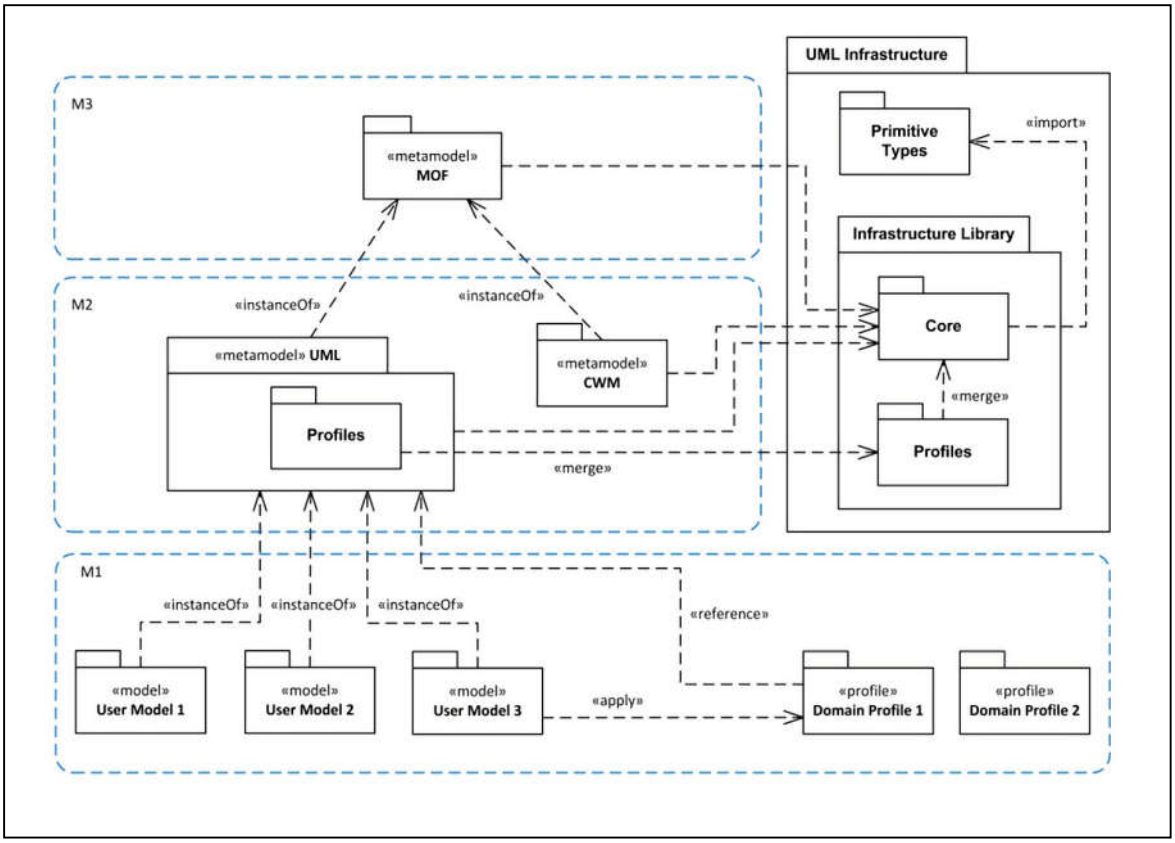


Figure II. 6: *Couches de Méta modélisation.*

4.2.Model Driven Architecture (MDA) :

Le MDA C'est une proposition à la fois d'une architecture et d'une démarche de développement. L'idée de base du MDA est la séparation des spécifications fonctionnelles d'un système des détails de son implémentation sur une plate-forme donnée. Pour cela, le MDA classe les modèles en modèles indépendants des plates-formes appelés PIM (Platform Independent Models), et en modèles spécifiques appelés PSM (Platform Specific Models) [26].

4.2.1. Architecture MDA

Le modèle MDA est décomposé en 4 niveaux [27] :

- Le niveau M0 qui est le niveau des données réelles, est composé des informations que nous souhaitons modéliser. Ce niveau est souvent considéré comme étant le monde réel.

- Le niveau M1 est composé de modèles d'information. Lorsque nous voulons décrire les informations de M0, le MDA considère que nous faisons un modèle appartenant au niveau M1. Tout modèle est exprimé dans un langage unique dont la définition est fournie explicitement au niveau M2.

- Le niveau M2 est donc composé de langages de définition des modèles d'information, appelés aussi méta-modèles. Un méta-modèle définit donc un langage de modélisation spécifique à un domaine et un profil définit un dialecte (une variante) de ce langage.

- Le niveau M3 qui est composé d'une unique entité qui est le langage unique de définition des métas modèles, aussi appelé le méta-méta-modèle ou le MOF (Meta-Object Facility). Le MOF définit la structure de tous les méta-modèles qui se trouvent au niveau M2. Le MOF est réflexif, c'est-à-dire que la description du MOF s'applique au MOF lui-même, ce qui permet de dire que le niveau M3 est le dernier niveau de l'architecture. Le niveau M3 correspond donc aux fonctionnalités universelles de modélisation logicielle, alors que le niveau M2 correspond aux aspects spécifiques des différents domaines, chaque aspect particulier étant pris en compte par un méta-modèle spécifique. Figure II.7 : L'architecture MDA [28].

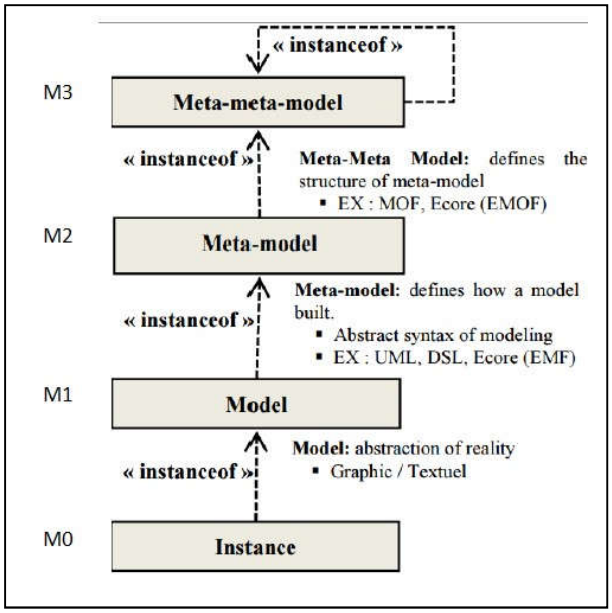


Figure II. 7: *Architecture MDA [28].*

4.3.Transformation de modèles

La transformation de modèles est au cœur de l'IDM et y joue un rôle important. Elle peut être définie comme une génération d'un modèle cible à partir d'un modèle source, suivant une définition de transformation qui est un ensemble de règles de transformation qui décrivent comment un modèle dans un langage source peut être transformé en un modèle dans un langage cible.

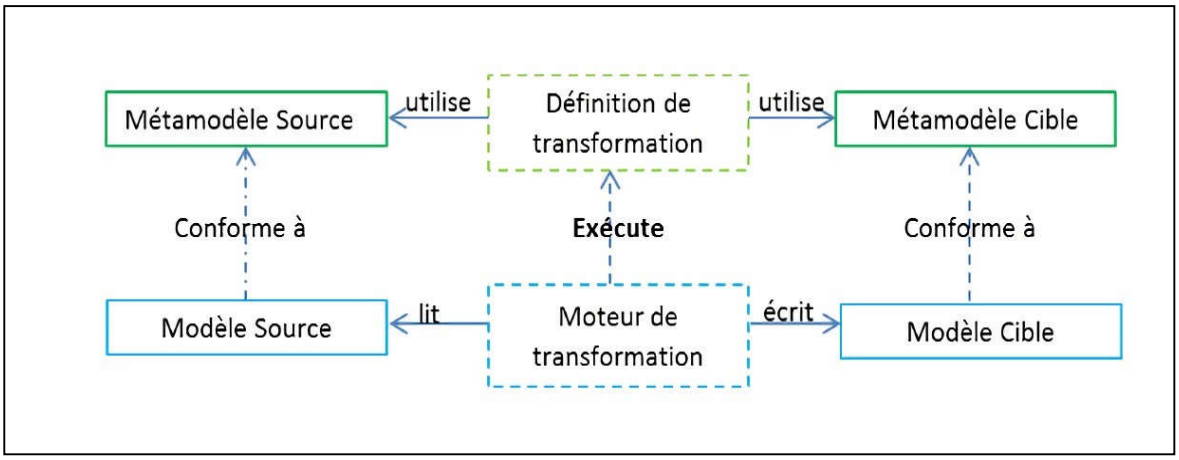


Figure II. 8: *Concepts de base de transformation de modèles.*

Une règle de transformation est une description de la manière d'une ou de plusieurs structures du langage source peuvent être transformées à une ou a plusieurs structures du langage cible.

Pour que la transformation soit appliquée à maintes reprises, elle doit être appliquée indépendamment du modèle source [25]. Les concepts de base de transformation sont présents dans la figure II.8.

4.3.1. Pourquoi la transformation de modèles

L'objectif principal et final de la transformation de modelés dans le contexte de l'IDM est la génération de code, mais la transformation de modèles couvre plusieurs activités parmi eux :

- **Vérification de modèle :** La vérification d'un modèle est le synonyme de modèle correct, c'est à dire le vérifier syntaxiquement et sémantiquement. L'analyse syntaxique vérifie si le modèle est conforme à un Méta modèle et s'il respecte les conditions "contraintes" imposées par le langage tandis que l'analyse sémantique vérifie les propriétés dynamiques d'un domaine.
- **Validation de modèle :** La validation est de vérifier si un modèle répond aux exigences, ou d'une autre façon de vérifier si un modèle est fiable. La simulation et le test de modèle sont utilisés pour vérifier si un modèle est valide.

4.3.2. Approche de transformation de modèles

Les approches de transformation de modèles sont classées en deux grandes familles : les approches de modèle au code (Model to Texte M2T) ou le code est considère comme un modèle spécifique et les approches de modèle a modèle (Model to Model M2M) [30].

Le tableau 2.1 ci-dessous, illustre les approches de transformation de modèles.

Approches de transformation de modèles	
M2M	*M2T*
Dirigée par la structure Manipulation directe	Parcours de modèles (Programmation)
Approche relationnelle Transformation des graphes Hybride	Template

Tableau II. 1 : *Approche de transformation de modèles.*

Cette catégorie porte sur la transformation de modèle à modèle. Les méthodes utilisées sont diverses et variées, parmi elles nous allons présenter les deux qui suivent :

Approche relationnelle Elle se base sur les transformations de type déclaratif reposant sur les relations mathématiques ou l'idée de base est d'indiquer les types d'éléments de la source et de la cible dans une relation et de les spécifier à l'aide des contraintes. Par définition, une telle transformation n'est pas exécutable telle quelle, mais son exécution est possible par l'adjonction d'une sémantique exécutable, par exemple au moyen de la programmation logique. Il n'y a pas d'effets de bord avec l'approche relationnelle, contrairement à l'approche par manipulation directe.

La plupart des transformations basées sur l'approche relationnelle permettent la multidirectionnalité des règles, et fournissent des liens de traçabilité e. Elles n'autorisent pas la transformation sur place (source et cible doivent être distinctes). Des exemples sur cette approche sont RFP [28], MOF 2.0 [30], QVT [31] [25].

Approche par manipulation directe : Cette approche offre une représentation interne d'un modèle plus une API pour le manipuler. Ils sont généralement implémentés dans un environnement orienté objet, qui peut également fournir une infrastructure minimale pour organiser les transformations. Mais les utilisateurs doivent implémenter les règles de transformations à partir de zéro en utilisant un langage de programmation comme Java. Des exemples de cette approche comprennent Jamda [31] et JMI (Java Metadata Interface est une API basée sur le MOF). L'approche par manipulation directe est de type impératif, elle est située à un bas niveau d'abstraction.

4.3.3. Transformation modèle vers texte M2T

Cette classe est caractérisée par la méthode de Génération de code source qui se base soit l'identification de canevas (Template), soit sur le parcours du modèle.

- **Génération de code par Template**

C'est l'une des approches la plus utilisée pour la génération de code. Cette approche consiste à définir des canevas de modèles cibles. Ces canevas sont des modèles cibles paramètre es ou modèles Template. L'exécution d'une transformation consiste à prendre un modèle Template et à remplacer ses paramètres par les valeurs d'un modèle source. D'une autre façon, elle consiste à utiliser comme RHS "Right Hand Side" (le côté droit de la règle) un texte fixe dans lequel certaines parties sont variables : elles sont renseignées en fonction des informations récupérées dans le modèle source LHS "Left Hand Side" (le côté gauche de la règle), et peuvent

notamment faire l'objet d'itérations. Parmi les langages les plus utilisées, on a Xpand [32], Acceleo [33].

– **Génération de code par parcours de modèle**

C'est l'approche de génération de code la plus basique qui consiste à parcourir la représentation du modèle en entrée et à écrire un code en sortie. L'exemple le plus connu avec cette approche est l'environnement orienté objet Jamda dont les modèles UML sont représentés par des classes ou une API manipule les modèles et un outil est utilisé pour la génération de code.

5. Conclusion

Le langage de modélisation SysML est profil du langage UML. Il est dédié à l'ingénierie systèmes basée sur les modèles. SysML permet de décrire différents types de systèmes, en particulier les systèmes cyber-physique. Le langage SysML à travers ces diagrammes qui peuvent être transformés en se basant sur l'ingénierie dirigée par les modèles constitue un choix très important pour servir au développement des jumeaux numériques via des transformations de modèles.

CHAPITRE III : Méthodologie proposée

1. Introduction

Les systèmes cyber physiques se composent de trois parties principales, la partie physique, la partie cyber et la partie cyber-physique. La partie physique regroupe tous les composants physiques du système. La partie cyber regroupe les softwares utilisés pour le contrôle et l'exploitation du système et la partie Cyber-physique regroupe les composants permettant de lier les composants des deux premières parties. Le développement de tels systèmes en suivant une démarche d'ingénierie systèmes basée sur les modèles passe par la création des modèles décrivant les composants des trois parties ainsi que leurs interactions.

Le développement d'un jumeau numérique d'un système cyber-physique peut passer par les mêmes phases de développement du système lui-même et peut ainsi être basé sur la modélisation de ce dernier. La même modélisation peut servir ainsi au développement de la partie physique du système par les ingénieurs spécialisés selon les composants à développer, au développement des parties cyber et cyber-physique et au développement du jumeau numérique du système. Cependant, pour réussir un tel développement, il est nécessaire de proposer une démarche d'ingénierie permettant le développement du jumeau numérique d'un système cyber-physique en se basant sur la modélisation de ce dernier.

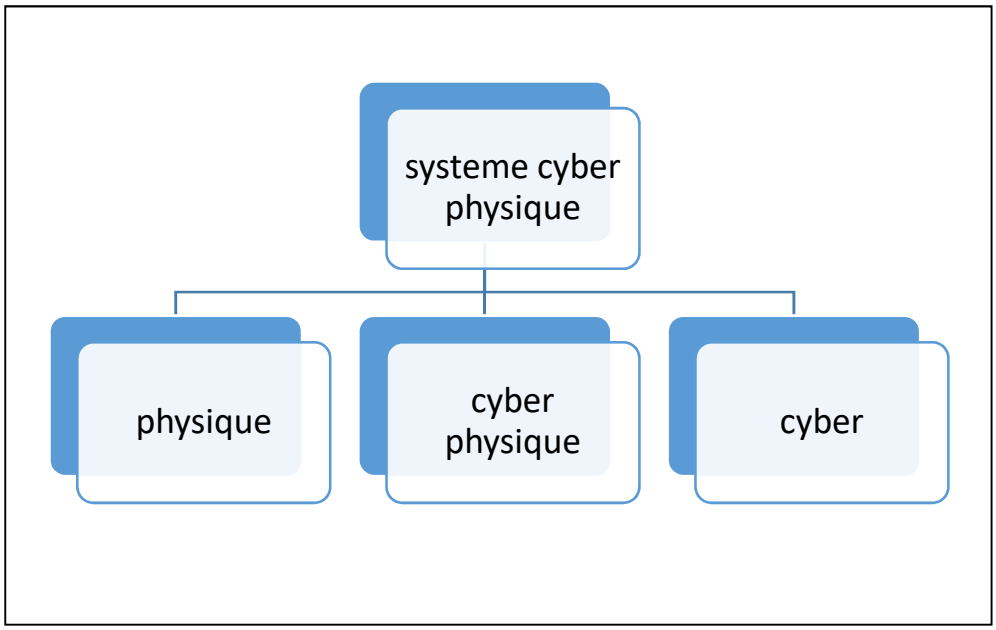


Figure III. 1 *: concept système cyber physique.*

2. Démarche Globale

La démarche que nous proposons pour le développement d'un jumeau numérique d'un système cyber-physique est orienté modèles. Elle est proposée pour être appliqué dès la phase de modélisation du système cyber-physique. En effet, il n'est pas nécessaire d'attendre l'implémentation du système réel pour développer son jumeau numérique. Ceci permet d'accélérer le temps de développement du système avec son jumeau numérique. Comme décrit par la figure III.2, la démarche d'ingénierie des jumeaux numériques proposée se base sur une méthodologie de modélisation des systèmes cyber-physiques permettant l'exploitation de cette modélisation pour la génération de l'implémentation des jumeaux numériques, ensuite, en se basant sur l'ingénierie dirigée par les modèles, le jumeau numérique est présenté sous forme d'un ensemble d'objets (things) avec leurs caractéristiques et leurs propriétés. Cette représentation est obtenue à partir d'une série de transformation. Enfin, le développeur du jumeau numérique est guidé par une application web pour créer les objets constituants le jumeau numérique. La création se fait au niveau d'une plateforme open source (eclipse ditto) implémentant les jumeaux numériques.

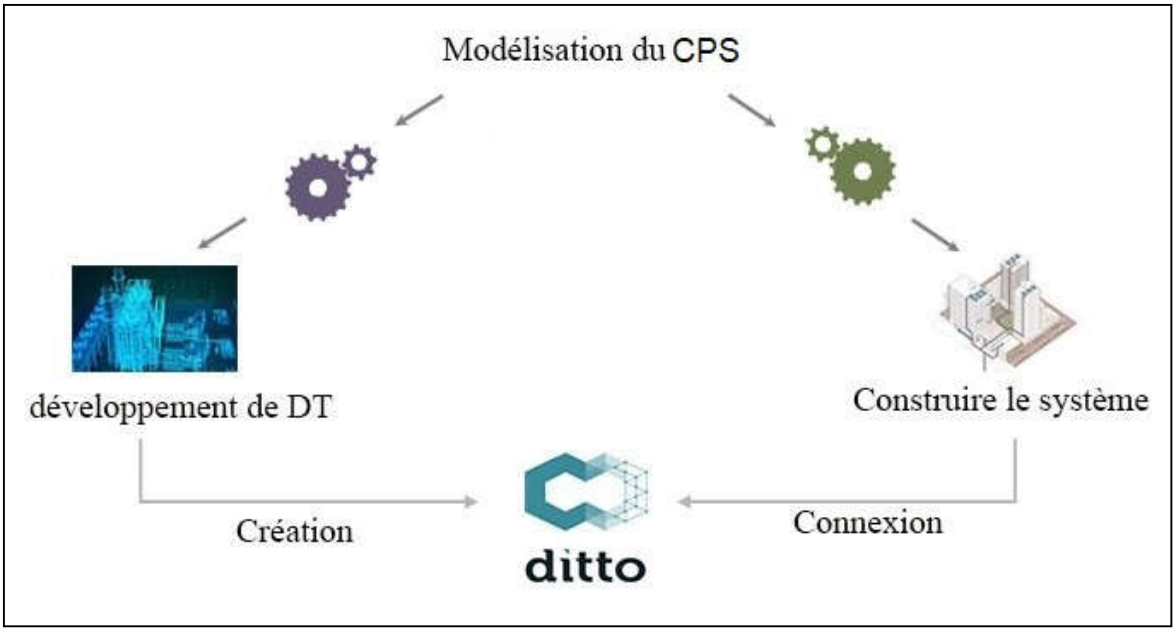


Figure III. 2 : *Démarche globale du travail.*

3. Démarche de modélisation des systèmes cyber-physiques

L'objectif de la démarche de modélisation est de permettre aux concepteurs d'acquérir une démarche rigoureuse pour modéliser les systèmes cyber physiques tout en évitant de faire des erreurs qui entravent le processus de transformation. Le travail de modélisation inclut trois questions, le diagramme à utiliser pour modéliser le système, comment l'ingénieur modélise chaque composant, et comment nous pouvons distinguer entre les types de ces composants. Chacune de ces questions est traitée et évaluée comme un problème en soi.

3.1. Diagramme de modélisation du système cyber physique

Les systèmes cyber physiques peuvent être modélisés via plusieurs diagrammes de modélisation (séquence, diagramme de définition de bloc (BDD), Diagramme interne de boc (IBD...Est). Pour le développement du jumeau numérique, le diagramme BDD est de très grande utilité. En effet, ce diagramme de bloc est un diagramme structurel qui décrit physiquement le système et/ou ses sous-systèmes et leur hiérarchie. Aussi décompose le système a des blocs. Cette décomposition permet d'apparaître tous les composants du système cyber-physique (P, CP, C), ce qui facilite de reconnaître et l'identifier le type de chaque composant.

3.2. Modélisation des trois types de composants d'un système cyber physique.

Le système cyber physique se compose de trois types de composants mentionnés ci-dessus (P, CP, C). Chacun de ces composants est modélisé dans le diagramme de bloc, soit comme un bloc ou comme un acteur qui contrôle ces blocs. Les composants physiques et les composants cybers physiques sont décrits dans le diagramme de bloc comme des blocs du système, avec leurs fonctionnalités et les relations entre eux. Les composants cyber sont décrits par le diagramme de bloc comme des acteurs contrôlant les blocs physiques, avec leur droit d'accès à ces blocs. Les relations entre les composants physiques et cyber physiques et les acteurs.

3.3. Identification des types des composants d'un système cyber physique

Une fois le système cyber physique est bien modélisé via le diagramme de bloc, tous les blocs du système et leurs relations sont définies dans le diagramme de blocs, il n'existe aucun

mécanisme permettant de distinguer entre les différents types des blocs. Cependant, il est nécessaire de distinguer entre les blocs physiques, les blocs cyber physique et les blocs cybers. Pour ce faire, nous proposons d'ajouter des annotations définissant les types de bloc composant le système.

3.3.1. Annotation

L'annotation est rajoutée au niveau du bloc comme un commentaire. Le commentaire est utilisé comme un identificateur du type du bloc. Nous notons seulement les blocs qui représentent les parties du système cyber physique, Donc, il n'affecte pas le schéma général du diagramme de bloc. Pour un composant physique, le bloc est annoté par le commentaire « physique ». Pour un composant cyber-physique, le bloc est annoté par le commentaire « cyber-physique ». Les blocs qui représentent les propriétés complexes des composants cybers physiques sont annotés par le commentaire « propriété complexe ». Les acteurs qui contrôlent les composants physiques du système cyber-physique sont annotés par le commentaire « cyber » et par d'autres commentaires définissant les droits d'accès de cet acteur sur les blocs qui les contrôle. Les droits d'accès sont : (écriture (write), lecture (read) et administration (administrate)). Leurs commentaires ont la forme suivante :

- Si l'acteur a le droit de lire, nous écrivons « read », sinon, nous la laissons vide.
- Si l'acteur a le droit d'écrire, nous écrivons « write », sinon, nous la laissons vide.
- Si l'acteur a le droit d'administration, nous écrivons « administrate », sinon, nous la laissons vide.

"nom de bloc : autorisation" → Example d'autorisation : (bloc1 : read, write).

– **Exemple :**

La Figure III.3 décrit une représentation abstraite d'un diagramme de bloc de système cyber-physique.

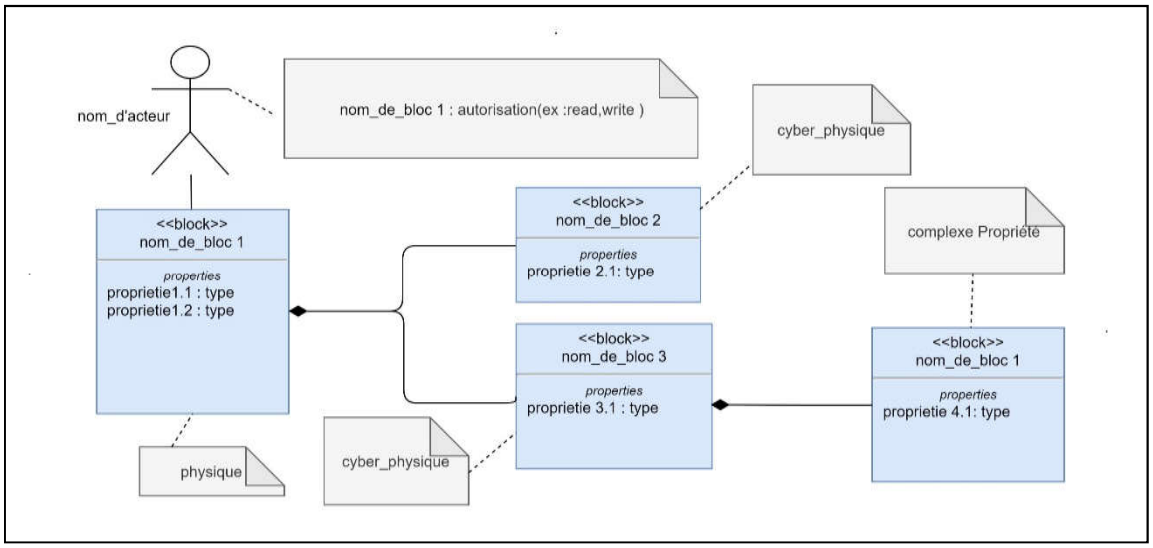


Figure III. 3 : *Exemple d'un système cyber physique.*

4. Méta-modèle du jumeau numérique

Pour pouvoir générer des jumeaux numériques via la transformation des modèles, nous proposons d'utiliser le méta-modèle présenté dans la figure III.4. Ce méta-modèle a été proposé dans le cadre de la plateforme Eclipse Ditto [2]. Comme montré par la figure III.4, la méta-classe Thing permet de représenter tous les objets. La méta-classe Acl et Acl-Enry permettent de représenter les listes de contrôle d'accès sur les objets. La méta-classe feature permet de représenter les caractéristiques des objets (Things), qui contient des propriétés de ces caractéristiques, les propriétés se divisent en deux types : complexe et simple.

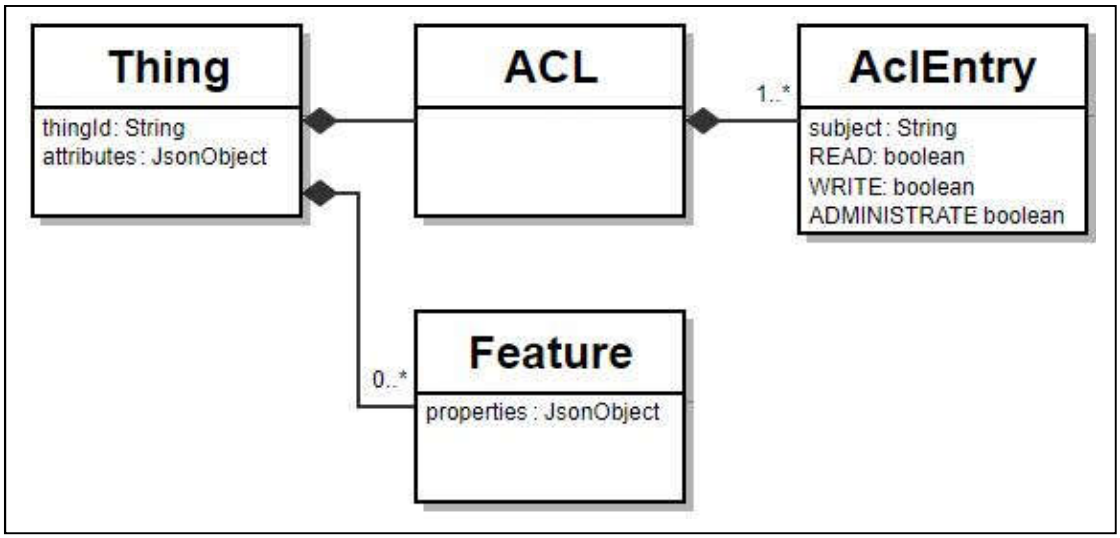


Figure III. 4*: Méta modèle des jumeaux numériques.*

4.1.Objets (thing)

Les objets sont des entités très génériques, et sont principalement utilisées en tant que "descripteur" de plusieurs fonctionnalités appartenant à cet objet.

Exemples :

- Dispositif physique : une tondeuse à gazon, un capteur, un véhicule, une lampe.
- Périphérique virtuel : une pièce dans une maison, une centrale électrique virtuelle couvrant plusieurs centrales, les informations météo d'un lieu spécifique collectées par divers capteurs.
- Entité transactionnelle : visite d'un véhicule (du début à la fin), série de mesures d'une machine.
- Entité de données maîtresse : fournisseur d'appareils ou fournisseur de services d'appareils, entité représentant une ville / une région.

– **Thingid** : Identifiant unique d'un objet.

– **Les attributs (Attributes)** : Les attributs décrivent l'objet (thing) plus en détail et peuvent être de n'importe quel type. Les attributs peuvent également être utilisés pour trouver des objets. Les attributs sont généralement utilisés pour modéliser des propriétés plutôt statiques au niveau d'objet. Statique signifie que les valeurs ne changent pas aussi souvent que les valeurs de propriété des entités.

4.2.Liste de contrôle d'accès (ACL)

La liste de contrôle d'accès (ACL), c'est une liste des configurations des permissions pour les sujets, les sujets sont identifiés par un identifiant arbitraire.

4.2.1. Sujet (Subject)

Un sujet peut avoir le type d'autorisation suivant : lecture, écriture, administration.

- Un sujet disposant de l'autorisation READ est autorisé à lire toutes les données d'un objet, c'est-à-dire ses attributs, caractéristiques et ACL.
- Une autorisation WRITE est requise pour pouvoir définir les données de l'objet ou envoyer des messages à un objet ou à une fonction.
- Pour pouvoir modifier la liste de contrôle d'accès elle-même, un sujet a besoin de l'autorisation ADMINISTRATE .

Lorsqu'un objet est initialement créé, au moins une entrée de la liste de contrôle d'accès avec les autorisations READ, WRITE et ADMINISTRATE est obligatoire et doit être fournie. Si aucune liste de contrôle d'accès n'est fournie lors de la création d'un nouvel objet, une liste de

contrôle d'accès par défaut sera créée. L'entrée par défaut de la liste de contrôle d'accès comprend un ID de sujet (par exemple, l'ID de l'utilisateur actuellement connecté) et les trois autorisations.

En outre, il doit toujours y avoir au moins une entrée avec toutes les autorisations. La dernière entrée ne peut pas être supprimée ou modifiée pour ne pas inclure toutes les autorisations.

– **READ**

Permet de lire un objet. Cela inclut la lecture de la liste de contrôle d'accès, des attributs, des fonctionnalités et de leurs propriétés.

Le transfert des messages aux abonnés est également sécurisé en fonction des entrées de la liste de contrôle d'accès figurant sur les objets elles-mêmes. Par exemple, seules les entités disposant de l'autorisation READ sur l'objet seront informées des modifications liées à cet objet et à ses fonctionnalités.

– **WRITE**

Permet d'écrire / modifier un objet. Par exemple, créer / mettre à jour / supprimer des objets, attributs, fonctionnalités et leurs propriétés, exécuter des opérations de fonctionnalité.

Afin d'envoyer des messages vers ou de recevoir des messages d'un objet, il vous faut l'autorisation WRITE sur l'objet spécifique.

– **ADMINISTRATE**

Permet de modifier la liste de contrôle d'accès d'un objet et de créer / modifier / supprimer ses entrées en fonction des restrictions mentionnées ci-dessus.

4.3.Fonctionnalité (Feature)

Une fonctionnalité est utilisée pour gérer toutes les données et fonctionnalités d'un objet pouvant être mises en cluster dans un contexte technique défini.

Pour différents contextes ou aspects d'un objet, vous pouvez utiliser différentes fonctionnalités qui appartiennent toutes au même objet et n'existent pas sans cet objet.

4.3.1. ID feature

Dans un objet, chaque entité est identifiée par une chaîne unique - appelée Identité. Un ID de fonctionnalité doit souvent être défini dans le chemin d'une requête HTTP. De ce fait, il est fortement recommandé d'utiliser un jeu de caractères restreint.

4.3.2. Propriété (property) :

Les données relatives aux fonctionnalités sont gérées sous la forme d'une liste des propriétés. Ces propriétés peuvent être classées, par exemple pour gérer l'état, la configuration ou toute information de panne. Les propriétés des fonctionnalités sont représentées sous la forme d'un seul objet JSON. Chaque propriété elle-même peut être une valeur simple / scalaire ou un objet complexe. Autorisé est une valeur JSON.

5. Démarche de transformation

Pour créer un jumeau numérique dans une plateforme, on doit représenter les concepts de ce dernier au format JSON, et pour obtenir le fichier JSON à partir de la modélisation initiale d'un système cyber physique, on doit faire une chaîne des transformations, qui nous permet de faire un passage régulier entre ses modèles. Les transformations passées à deux étapes principales, le premier, c'est la transformation du modèle de bloc vers le modèle du jumeau numérique (M2M), cette étape est basée sur l'annotation proposée dans la démarche de la modélisation, la deuxième étape, c'est la transformation du modèle de jumeau numérique obtenu dans l'étape précédente, vers un fichier JSON (M2T), cette transformation permet de décrire les concepts du modèle de jumeaux numérique au format JSON.

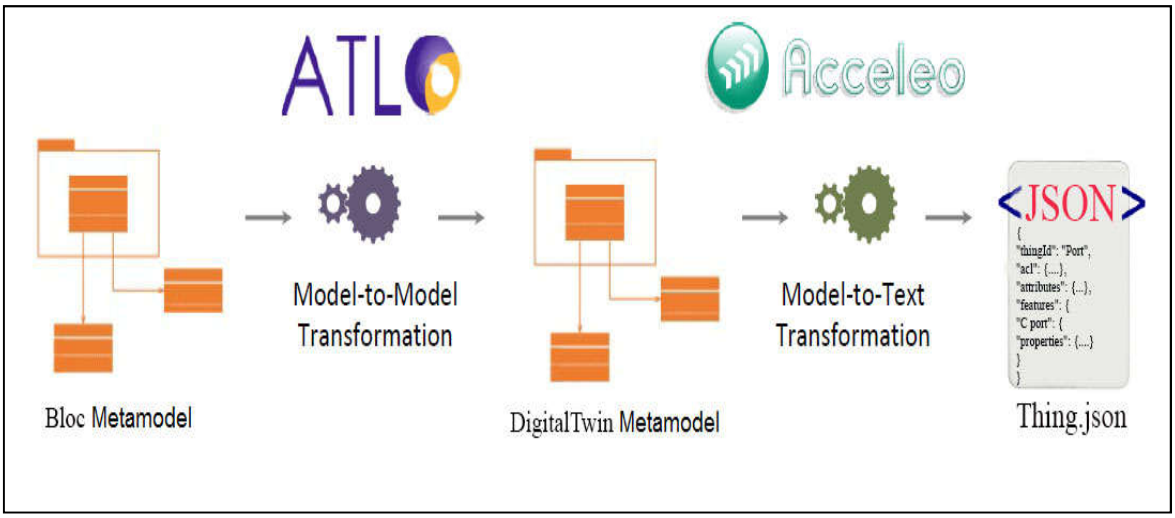


Figure III. 5*: Démarche de transformation.*

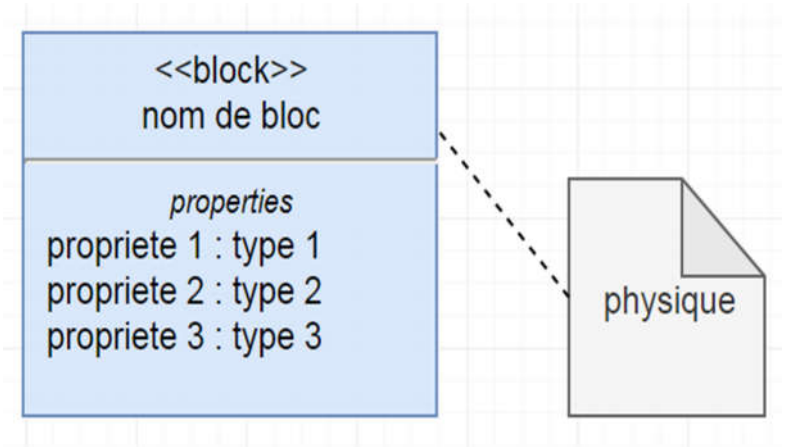
5.1. Transformation modèle vers modèle M2M

la transformation entre le diagramme du bloc et le jumeau numérique passe en trois étapes principales, la première étape, c'est la transformation des blocs physiques vers des objet (things)

et leurs attributs(Attributes), le deuxième étape, c'est la transformation des blocs cyber physique vers des fonctionnalités (Feature) et leur propriété(Property), le troisième étape, c'est la transformation des acteurs vers des listes de contrôle d'accès (ACLs et AclEntry), en suivant la méthode suivante :

5.1.1. Bloc physique

Le bloc qui représente le composant physique contient un nom qui identifie le bloc, et contient des attributs qui identifient les caractéristiques de ce bloc. Dans notre proposition nous transformons les blocs physiques vers des objets (thing). Dans ce contexte, nous transformons le nom de bloc vers le thingid d'objet, et les caractéristiques de bloc vers les attributs d'objet, dans la transformation des caractéristiques, nous transformons les noms de la caractéristique vers les noms des attributs et les types des caractéristiques vers les types des attributs, le tableau III.1 explique un exemple d'une transformation d'un bloc physique vers un objet (thing).

Modèle	Figure
Modèle de bloc.	

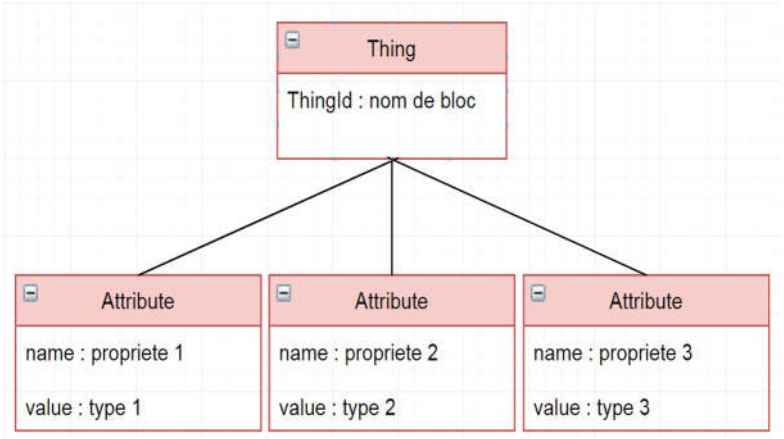
Modèle de jumeau numérique.	

Tableau III. 1 : *Transformation des objets.*

5.1.2. Blocs cyber physique

Le bloc qui représente le composant cyber physique contient un nom qui identifie le bloc, et contient des attributs qui identifient les caractéristiques de ce bloc. Dans notre proposition nous transformons les blocs cyber physique vers des fonctionnalités (Feature). Dans ce contexte, nous transformons le nom de bloc vers le nom de fonctionnalité (idfeature), et les caractéristiques de bloc vers les propriétés (property) de la fonctionnalité (Feature). Dans la transformation des caractéristiques, il y a deux types des propriétés (simple et complexe). Pour les propriétés simples, nous transformons les noms des caractéristiques de bloc vers les noms des propriétés d'objet et les types des caractéristiques vers les types des propriétés, et pour les propriété complexe, ils sont modélisés comme un bloc séparé, contient un nom et des attributs qui identifie le bloc, nous transformons les noms de bloc vers les noms de la propriété complexe, et les caractéristiques de bloc vers des propriétés de la propriété complexe.

Le tableau III.2 explique un exemple d'une transformation d'un bloc cyber physique vers un fonctionnalité (Feature).

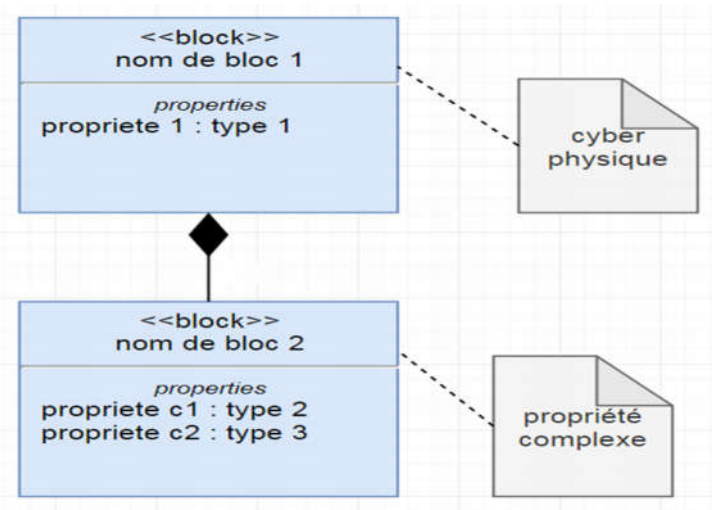
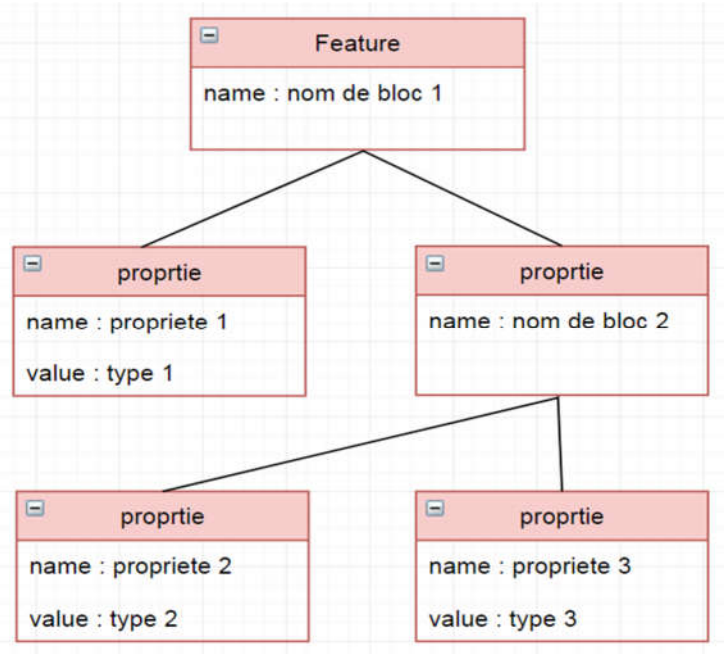
Modèle	Figure
Modèle de bloc.	
Modèle de jumeau numérique.	

Tableau III. 2 : *Transformation des fonctionnalités.*

5.1.3. Acteur

Les acteurs doivent contenir un nom et des commentaires définissant les droits d'accès de cet acteur pour les blocs qui il les contrôle. Nous expliquons le format de ces commentaires dans l'annotation, nous transformons le nom d'acteur vers le sujet (subject) d'ACL, et nous transformons les commentaires vers les droits d'accès (ACL) d'objet, comme suivant :

–Si le commentaire contient le mot « read » alors le champ « Read » de la classe « aclEntry » prend la valeur « true », sinon, il prend la valeur « false ».

–Si le commentaire contient le mot « write » alors le champ « Read » de la classe « aclEntry » prend la valeur « true », sinon, il prend la valeur « false ».

–Si le commentaire contient le mot « administrate » alors le champ « Read » de la classe « aclEntry » prend la valeur « true », sinon, il prend la valeur « false ».

Le tableau III.3 explique un exemple :

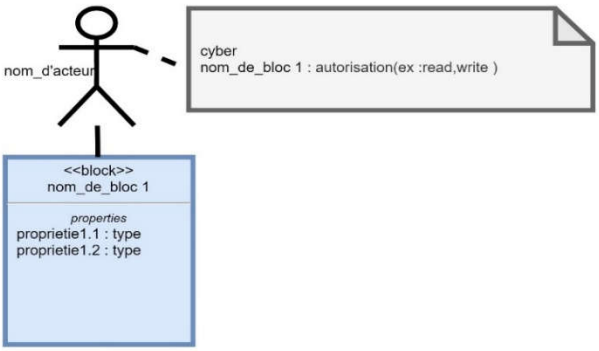
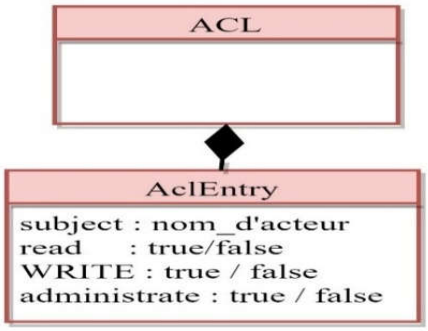
Modèle	Figure
Modèle de bloc.	
Modèle de jumeau numérique.	

Tableau III. 3 *: Transformation d'ACL.*

❖ **Remarque**

A la fin de la transformation, on obtient un modèle de type XML (.xml) qui est conforme au méta modèle des jumeaux numériques, qui contient la hiérarchie des objets (thing) du système cyber physique.

5.1.4. Exemple

La figure III.6 ci-dessous donne une représentation abstraite d'un modèle de jumeau numérique généré par la transformation M2M, d'une représentation abstraite d'un diagramme de bloc de système cyber physique.

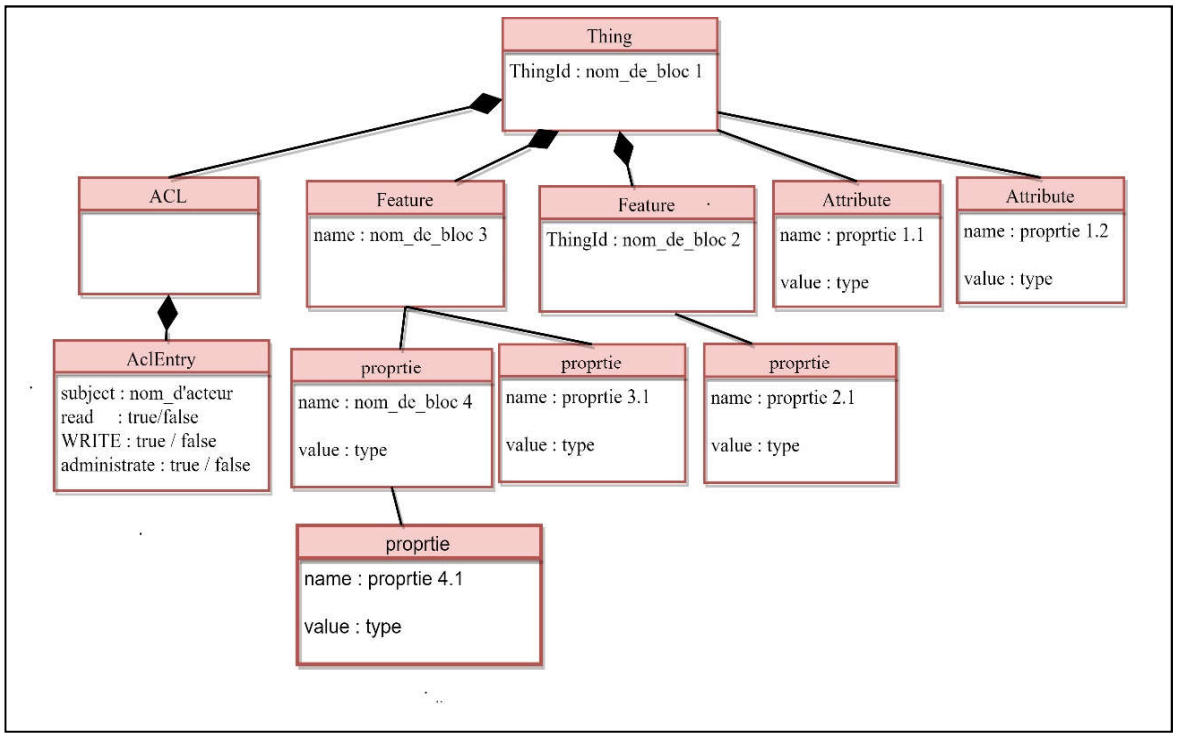


Figure III. 6 : *exemple d'un modèle généré par la transformation M2M.*

5.2. Transformation modèle vers texte M2T

La transformation du modèle jumeau numérique vers un fichier JSON (M2T) doit respecter la hiérarchie et la décomposition des concepts de l'objet, la figure III.7 définit comment la forme des objets devrait être au format JSON, comme illustré à la figure III.7 on transforme le modèle obtenu dans la première transformation M2M, et on transforme chaque composant dans le modèle vers leur opposite composant dans le format JSON.



Figure III. 7 : *Exemple de format de JSON du jumeau numérique.*

Toutes les opérations des transformations (M2M et M2T), qui sont faites appartient à une exécution manuelle du projet, pour cela nous avons besoin d'un outil pour lancer les processus de transformation d'une manière réutilisable et structuré, pour éviter beaucoup de problèmes potentiels dans le futur, nous développerons une application web pour créer les jumeaux numériques et contrôler la création.

6. Application de création du jumeaux numériques

Cette application a été créée afin d'aider l'utilisateur à créer les jumeaux numériques pour les systèmes cyber physiques d'une manière simple à travers un API web facile, qui permet à tous l'utiliser et assure toutes les étapes du développement des jumeaux numériques, de la création jusqu'à l'affichage d'une image numérique du système.

6.1. Problématique

Après avoir effectué une série des transformations et représenté le système cyber physique sous format JSON, nous avons rencontré un certain nombre des problèmes pour créer le jumeau numérique de ce système. Parmi eux :

- Comment modifier cette représentation et choisir les objets, leurs fonctionnalités, leurs propriétés.
- Comment le créer dans la plateforme de jumeau numérique.
- La difficulté des créer les interfaces de jumeaux numérique pour ce système de manière dynamique et les connecter avec la plateforme de jumeau numérique pour obtenir des informations instantanées sur les objets.

6.2. Objectif

Cette application permet à l'utilisateur d'effectuer les opérations de transformation mentionnées ci-dessus et d'afficher les résultats obtenus, facilement. Ainsi que l'identification des objets (things) et des fonctionnalités (features) et des propriétés (property) qui on veut les trouver dans le jumeau numérique obtenu, puis les créer dans la plateforme de jumeau numérique, après l'étape de l'identification et la création des objets (things), l'utilisateur peut créer l'interface de jumeau numérique et le connecter à la plateforme Eclipse Ditto, pour obtenir des informations en temps réel. Cette application permet de réduire le temps et les efforts nécessaires pour effectuer cette opération dans des cas normaux, et permet également de réduire le nombre d'erreurs pouvant être obtenues, que ce soit des erreurs de fonction de système, ou des erreurs dans la conception.

La figure III.8, présente les étapes de fonctionnalité de l'API web, l'API web prend le diagramme de bloc du système ou format d'UML et le transforme à un modèle de jumeau numérique. Puis à un fichier JSON, après l'application crée les objets et leurs interfaces d'une manière dynamique, ces interfaces permet de contrôler les fonctionnalités de chaque objet (thing), finalement l'API web permet à l'utilisateur de créer les objets en Eclipse Ditto et créer une interface de jumeau numérique permet d'afficher l'état de système en temps réel, avec une possibilité de contrôler le système à distance.

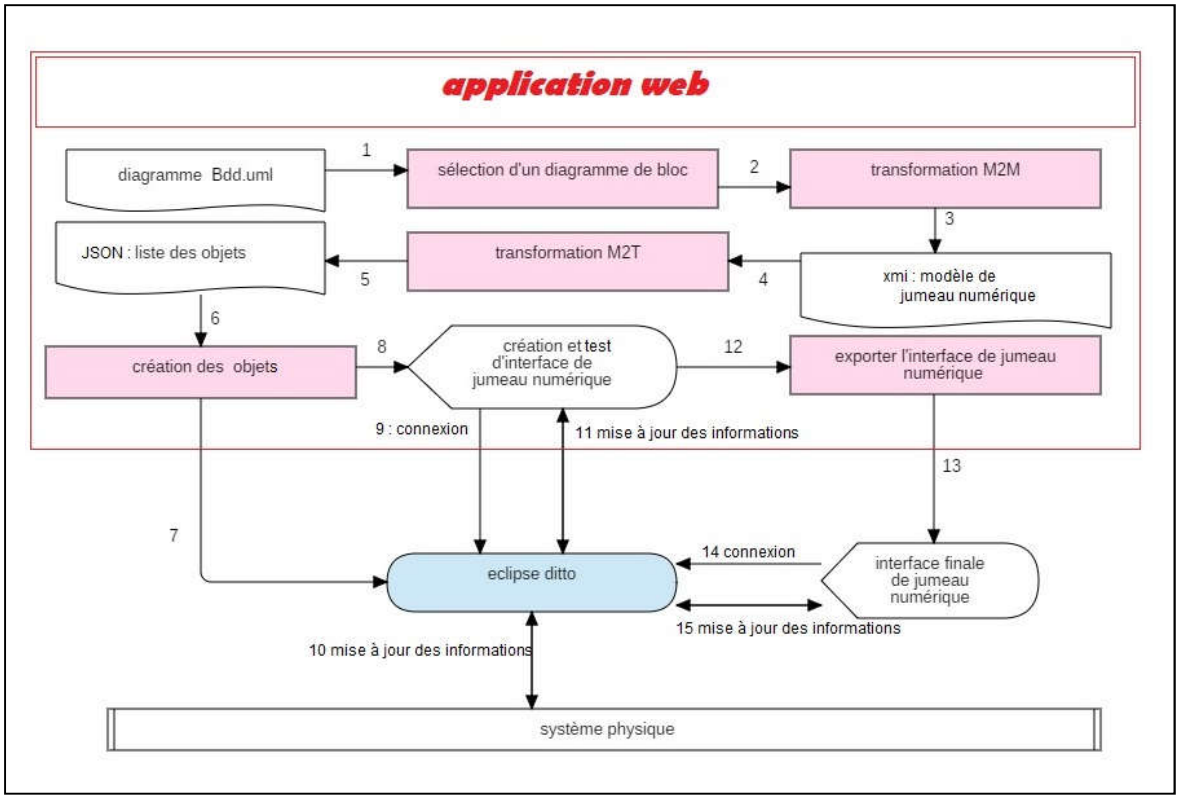


Figure III. 8 : Architecture d'application.

6.3.Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle, car c'est une forme souple qui convient dans bien des situations.

Ci-dessous, nous présentons le diagramme de cas d'utilisation d'un utilisateur qui utilise les cas d'utilisations suivants : générer jumeaux numériques modèle, générer fichier JSON, gestion d'objet (THINGS), créer jumeaux numériques interface (voir Figure III.9).

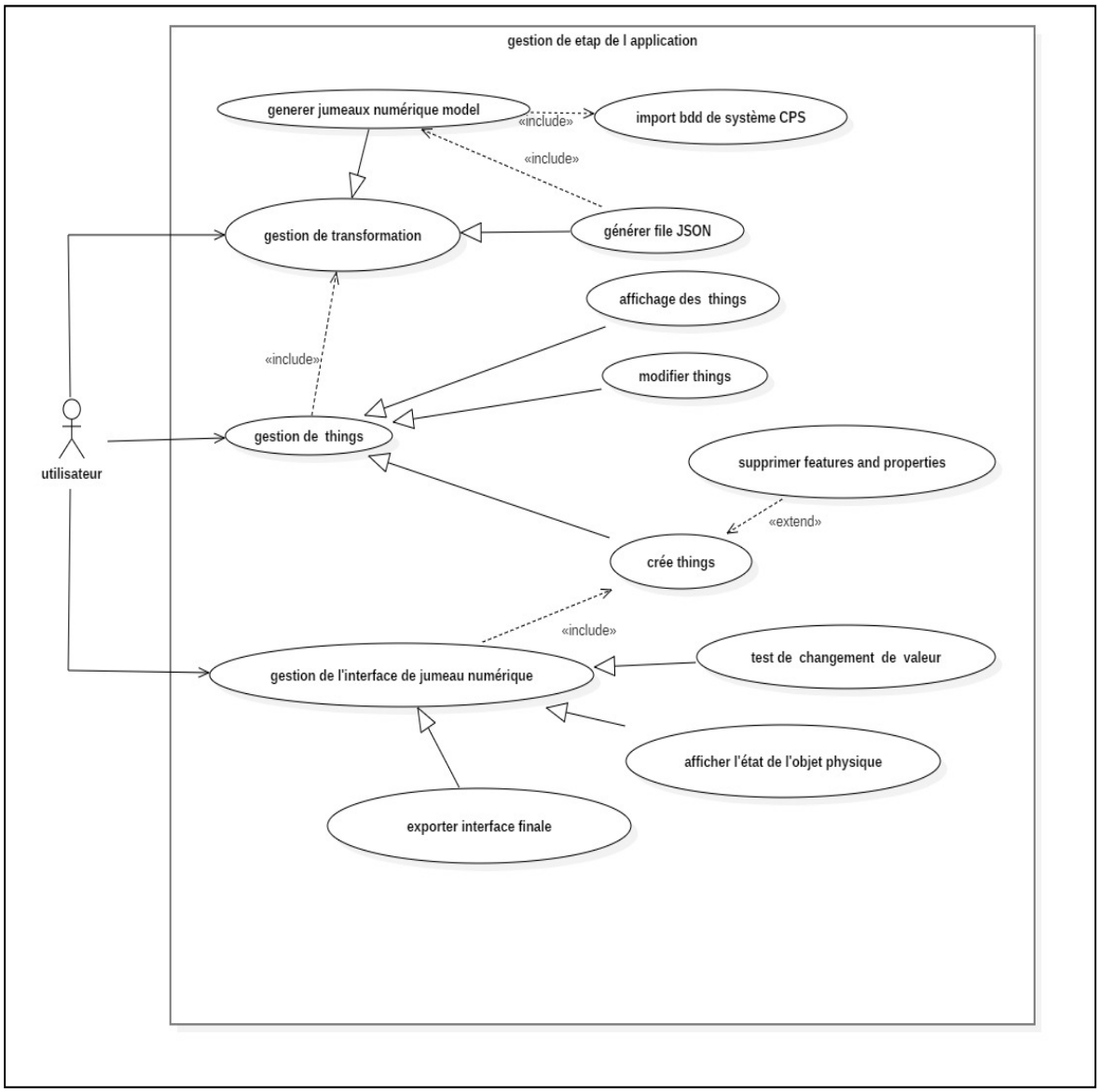


Figure III. 9*: Diagramme de cas d'utilisation.*

Le tableau ci-dessous représente la description textuelle de sous cas d'utilisation « Générer jumeaux numériques model » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre :	Gestion de l'interface de jumeau numérique
Description :	Ce cas permet d'exécuter une série de transformation et générer une représentation au format JSON à une modélisation de système cyber physique donné avec diagramme de bloc.

Acteur :	Utilisateur
Sous cas :	Générer jumeaux numériques model, générer fichier JSON.
Description des enchaînements	
Préconditions :	• Existence de diagramme de bloc
Post conditions :	• Existence de fichier JSON. • Gestion des objets (things).
Scénario nominal	
• Le cas d'utilisation passe par les trois étapes suivantes : o Sélection de diagramme de bloc. o Générer jumeaux numériques modèle. o Générer fichier JSON.	
Enchaînement alternatif	
• L'utilisateur décide de quitter l'application.	

Tableau III. 4: *Description textuelle de sous cas d'utilisation « Gestion de l'interface de jumeau numérique ».*

Le tableau ci-dessous représente la description textuelle de sous cas d'utilisation « Générer jumeaux numériques modèle » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre :	Générer jumeaux numériques modèle
Description :	Ce cas permet d'exécuter la transformation M2M et générer un modèle jumeau numérique conforme au méta-modèle de plateforme utilisé.
Acteur :	Utilisateur
Sous cas :	/
Description des enchaînements	
Préconditions :	• Existence de diagramme de bloc
Post conditions :	• Existence de jumeau numérique modèle. • Générer fichier JSON.
Scénario nominal	
• L'application demande de sélectionner un diagramme de bloc (ce diagramme est un fichier d'extension. uml).	

• L'utilisateur sélectionne un diagramme de bloc pour faire la transformation. • L'application affiche une fenêtre de traitement de transformation. • Après la fin de transformation il affiche que la transformation est finie et le modèle jumeau numérique est bien créé.
Enchaînement alternatif
• L'utilisateur décide de quitter l'application avant la transformation se complète.

Tableau III. 5 : *Description textuelle de sous cas d'utilisation « Générer jumeaux numériques modèle ».*

Le tableau ci-dessous représente la description textuelle de sous cas d'utilisation « Générer fichier JSON » ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre	Générer fichier JSON
Description	Ce cas permet d'exécuter la transformation M2T pour obtenir la représentation en format JSON du modèle jumeau numérique.
Acteur	Utilisateur de l'application
Sous cas	/
Description des enchaînements	
Préconditions :	Existence de jumeau numérique model.
Post conditions :	• Existence de fichier JSON contient tous les éléments (things) du modèle jumeau numérique. • Gestion de things. .
Scénario nominal	
• L'application affiche que la transformation M2M est finie, et le modèle jumeau numérique est bien créé, et demande l'autorisation pour exécuter la transformation M2T et générer le fichier JSON. • L'utilisateur donne l'autorisation pour démarrer la transformation.	

• L'application affiche une fenêtre de traitement de transformation. • Après la fin de transformation, il affiche que la transformation est finie et le fichier JSON est bien créé.
Enchaînement alternatif
L'utilisateur décide de quitter l'application avant la transformation complète.

Tableau III. 6 *: Description textuelle de cas d'utilisation « Générer fichier JSON ».*

Le tableau ci-dessous représente la description textuelle du cas d'utilisation « gestion de THINGS » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre	Gestion de THINGS
Description	Ce cas permet afficher l'objet (THING) pour les créer ou les modifier.
Acteur	Utilisateur
Sous cas	Créer objet(thing), modifier objet (thing).
Description des enchaînements	
Préconditions :	Transformation M2M et M2T est finie.
Post conditions :	Afficher l'interface d'objet (THINGS).
Scénario nominal	
• Le système affiche une liste d'objet(thing), avec un formulaire qui contient les fonctionnalités (features) et les attributs d'objet (thing) sélectionné (premier par défauts) • L'utilisateur sélectionne les propriétés et les fonctionnalités (features) qu'il souhaite dans cet objet (thing).	
Enchaînement alternatif	
• L'utilisateur sélectionne un autre objet (thing). • L'utilisateur clique sur le bouton de création d'interface de jumeau numérique. • L'utilisateur quitte l'application**.**	

Tableau III. 7 *: Description textuelle de cas d'utilisation « Gestion de THINGS ».*

Le tableau ci-dessous représente la description textuelle de cas d'utilisation « gestion de l'interface de jumeau numérique » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre :	Gestion de l'interface de jumeau numérique
Description :	Ce cas permet à l'utilisateur de voir l'interface de jumeau numérique et l'exporter.
Acteur :	Utilisateur
Sous cas :	Test de changement de valeur, afficher l'état de l'objet physique, exporter interface finale.
Description des enchaînements	
Préconditions :	Il y a au moins un objet (thing) créé.
Post conditions :	Afficher le jumeau numérique des objets (things) créé
Scénario nominal	
• L'utilisateur clique sur créer un jumeau numérique modèle. • L'application affiche l'interface de jumeau numérique avec des valeurs de propriétés par défaut. • Le système connecte à la plateforme de jumeau numérique et affiche l'information instantanée du système cyber physique.	
Enchaînement alternatif	
• L'utilisateur retourne à l'étape de gestion des objets (things). • L'utilisateur quitte l'application**.**	

Tableau III. 8 *: Description textuelle du cas d'utilisation « gestion de l'interface de jumeau numérique ».*

Le tableau ci-dessous représente la description textuelle de sous cas d'utilisation « test de changement de valeur » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre :	Test de changement de valeur
Description :	Ce cas permet de modifier l'état de l'actionneur en modifiant les valeurs dans l'interface de jumeau numérique.
Acteur :	Utilisateur

Sous cas :	/
Description des enchaînements	
Préconditions :	• Il y a au moins un objet(thing) créé. • L'utilisateur a l'autorisation de modifier (écrire).
Post conditions :	Afficher le jumeau numérique des objets (things) créé.
Scénario nominal	
• L'utilisateur clique sur le bouton modifier. • L'application affiche un formulaire et demande d'entrer la nouvelle valeur. • L'application de changement dans la plateforme de jumeau numérique. • L'application affiche changement réussi.	
Enchaînement alternatif	
• Le changement n'est pas réussi, donc l'application affiche changement échoué. • L'utilisateur quitte l'application**.**	

Tableau III. 9: Description textuelle de cas d'utilisation « *Test de changement de valeur* ».

Le tableau ci-dessous représente la description textuelle de cas d'utilisation « exporter interface finale » et ce dernier montre en détail le scénario et l'enchaînement de ce cas.

Sommaire	
Titre :	Exporter l'interface finale.
Description :	Ce cas permet d'exporter l'interface de jumeau numérique pour l'usage quotidien.
Acteur :	Utilisateur
Sous cas :	/
Description des enchaînements	
Préconditions :	Afficher l'interface du jumeau numérique.
Post conditions :	Créer un nouveau dossier nommé « JN » dans le même répertoire qui contient l'application, ce dossier contient le nouvelle interface de jumeau numérique.
Scénario nominal	
• L'utilisateur clique sur le bouton export interface.	

• L'application crée le nouveau dossier et exporte l'interface. • L'application affiche que l'interface est créée.

Tableau III. 10: *Description textuelle du cas d'utilisation « Exporter interface finale ».*

6.4.Diagramme de séquence

Le diagramme de séquence permet de décrire comment les éléments du système interagissent entre eux et avec les acteurs pour accomplir une tâche du système. Ce diagramme montre les interactions selon un point de vue temporel.

Un diagramme de séquence représente l'interaction entre les objets en insistant sur la chronologie des envois des messages [29].

La Figure III.10, suivante va vous montrer les séquences et les étapes à suivre dans l'exécution de la transformation.

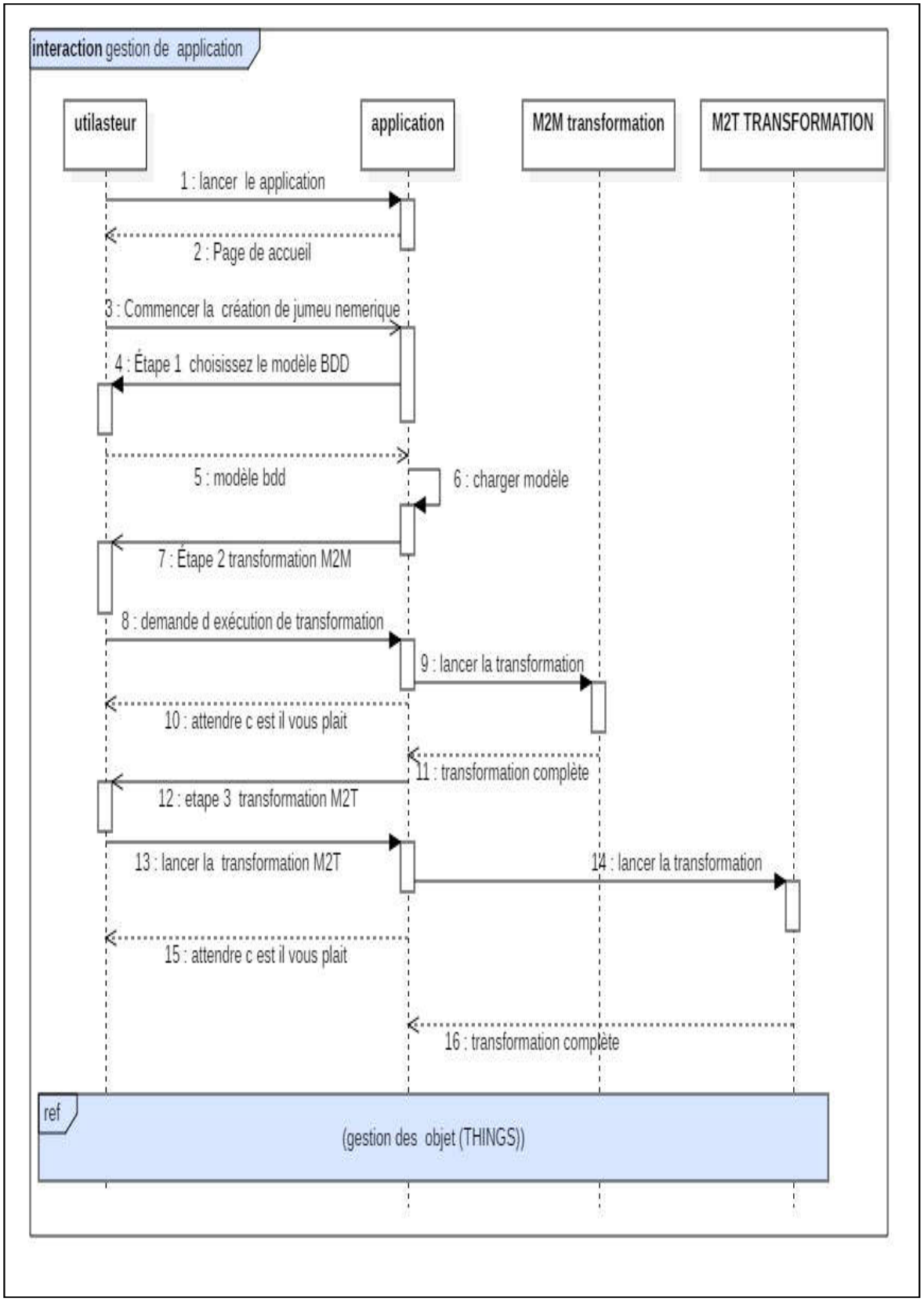


Figure III. 10 : *Diagramme de séquence de gestion de la transformation.*

La Figure III.11, suivante va vous montrer les séquences à effectuer pour entamer la phase de sélectionner et naviguer entre les objets (things).

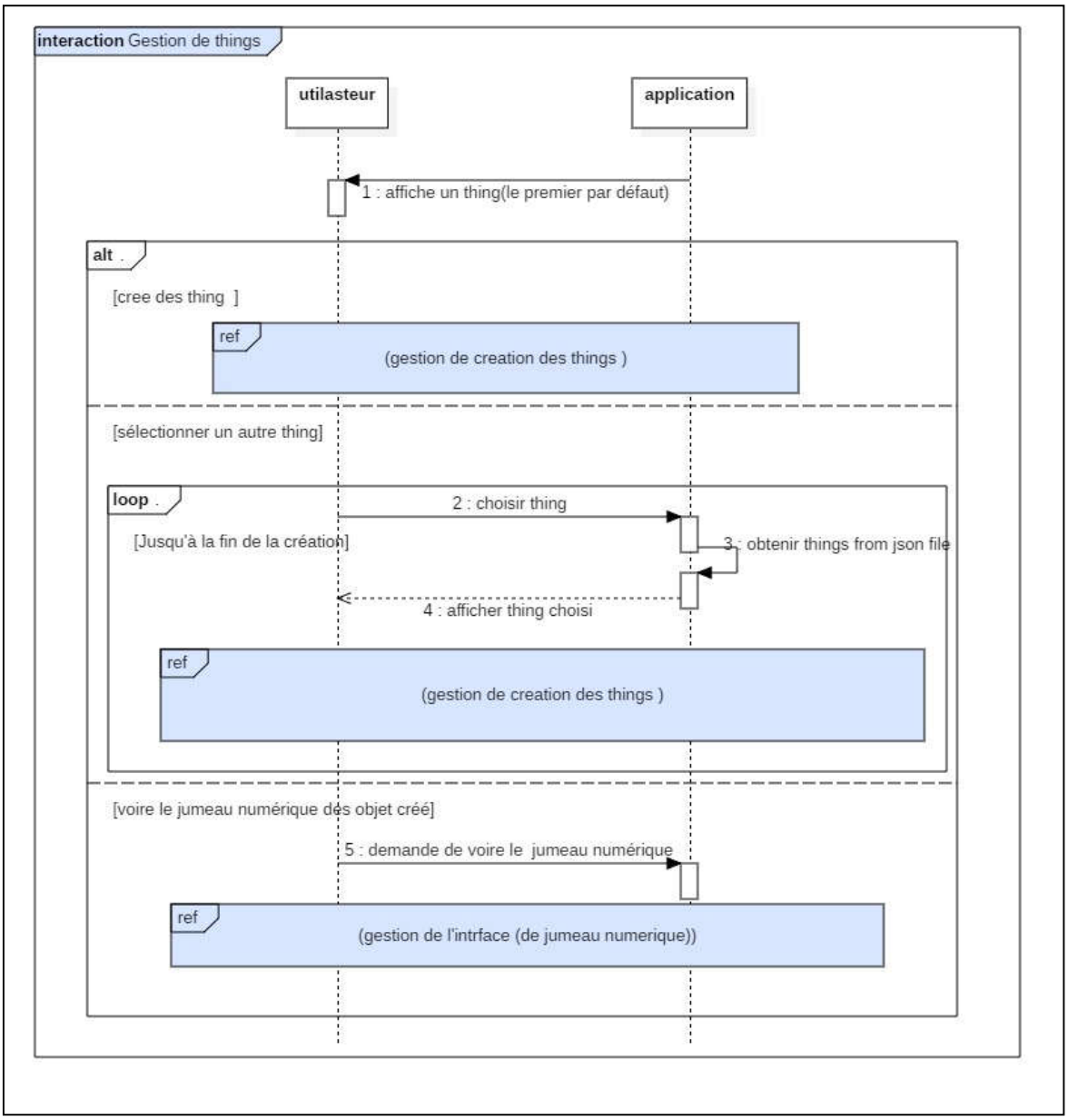


Figure III. 11 : *Diagramme de séquence de gestion d'objet.*

La Figure III.12, suivante va vous montrer les séquences à effectuer pour entamer la phase de création des objets (things) dans la plateforme de jumeau numérique.

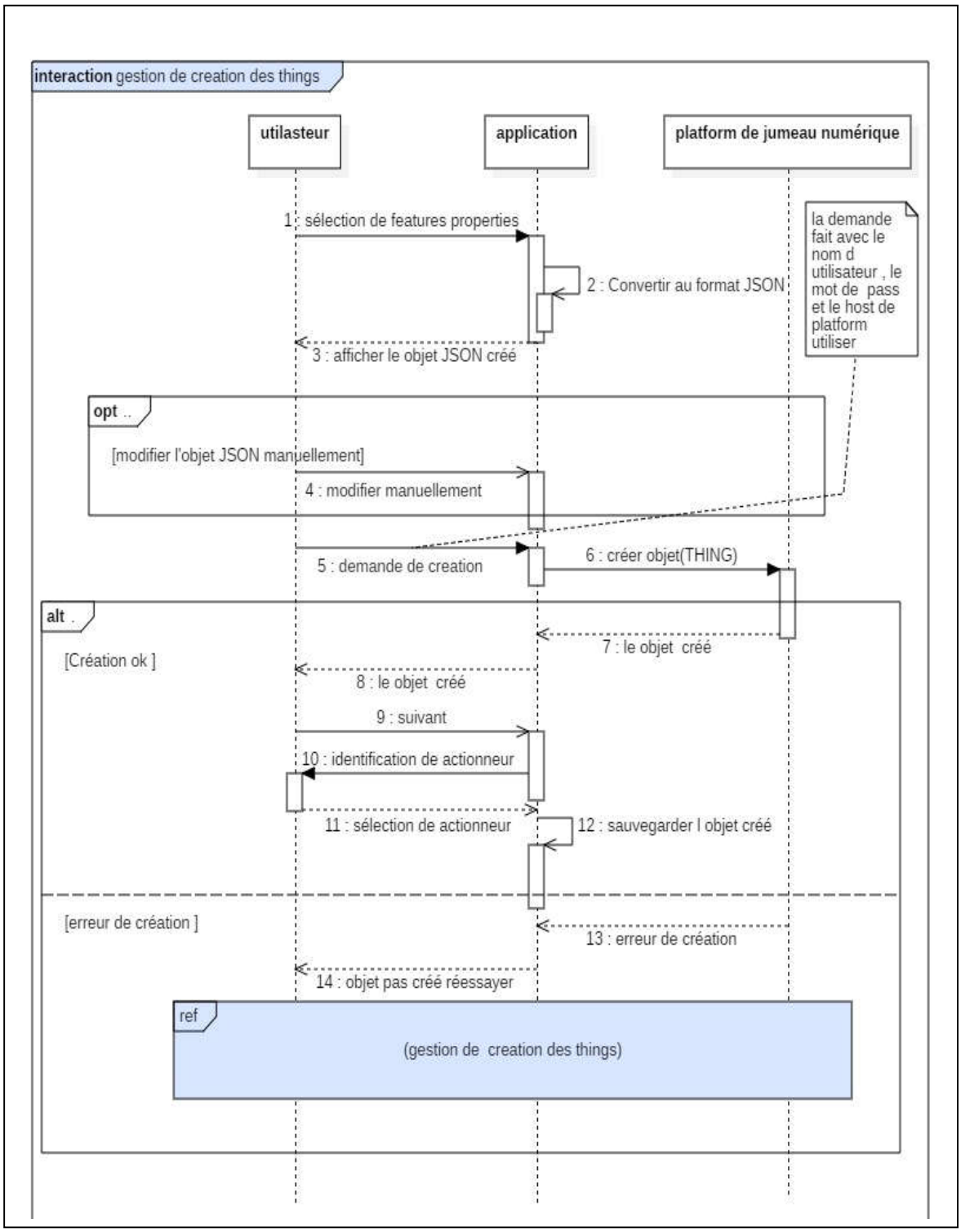


Figure III. 12 *: Diagramme de séquence de gestion de création des objets.*

La Figure III.13, va vous montrer les séquences à effectuer pour entamer la phase de création de jumeau numérique interface :

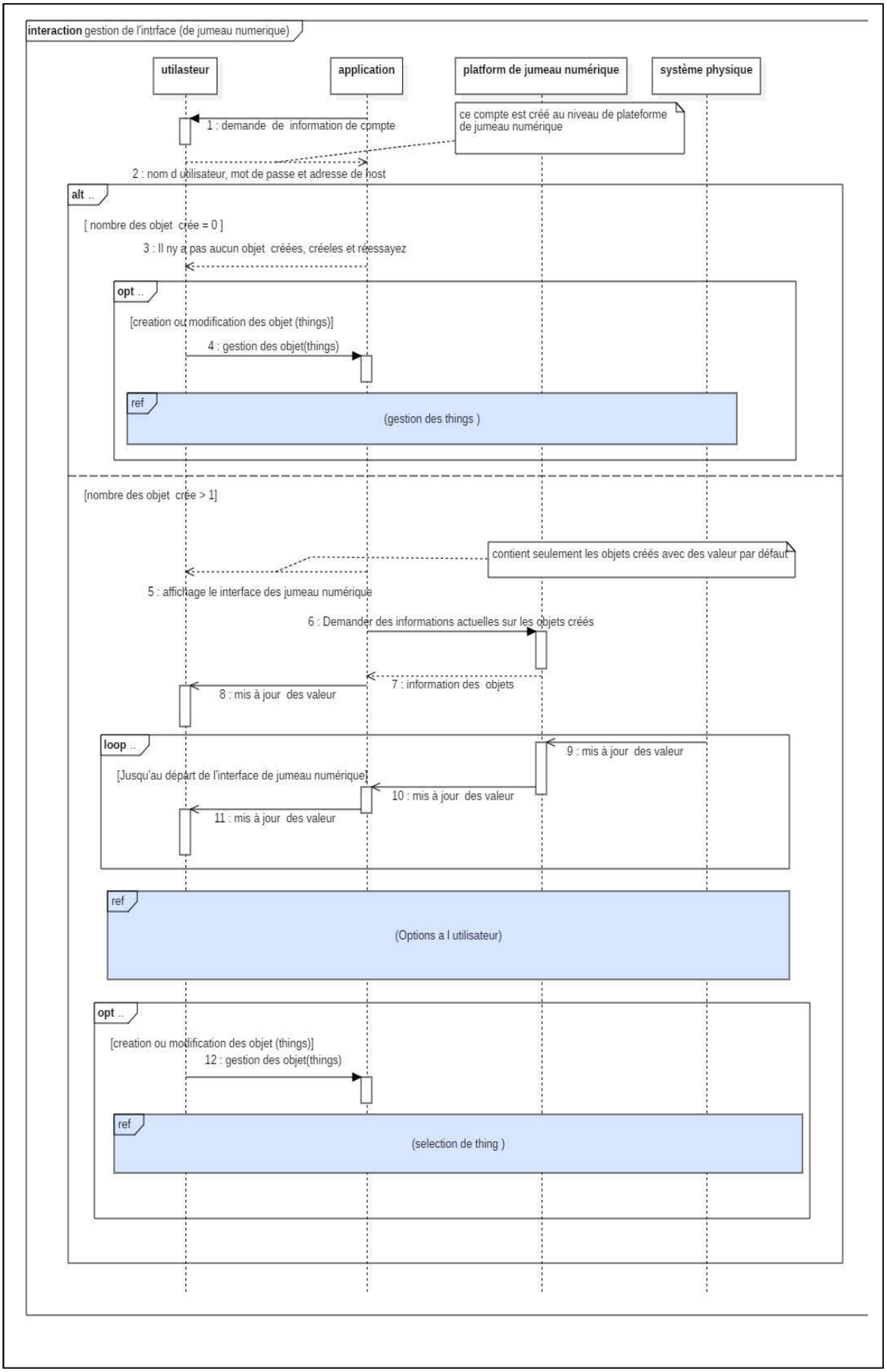


Figure III. 13 : *Diagramme de séquence de gestion d'interface du jumeau numérique.*

La Figure III.14, va vous montrer les opérations que l'utilisateur peut effectuer pendant l'affichage de la fenêtre du jumeau numérique.

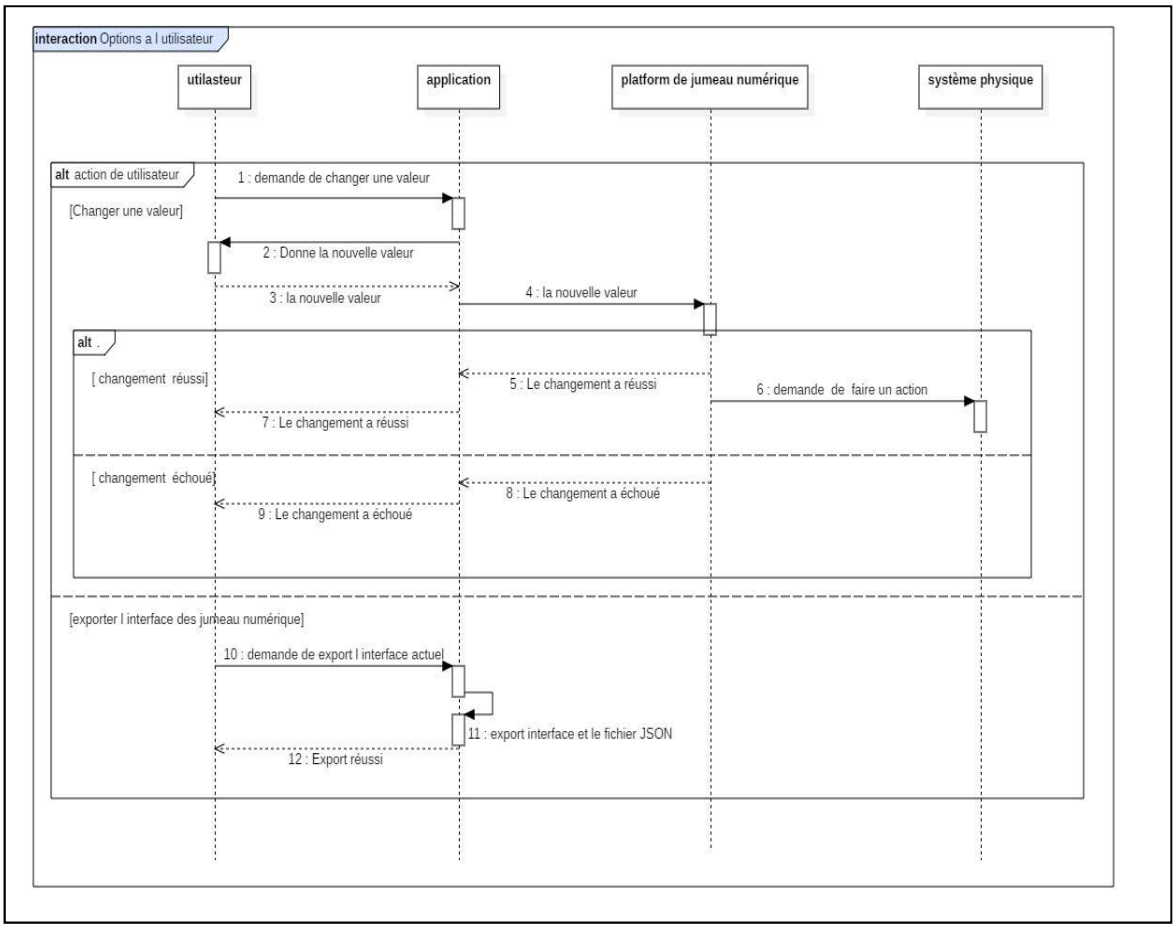


Figure III. 14: *Options utilisateur dans l'interface du jumeau numérique.*

7. Intégration avec le système physique

À la suite de création de l'interface des jumeaux numériques pour un système cyber physique, ce système est lié avec une plateforme des jumeaux numériques (Eclipse Ditto, Eclipse Hono ...etc.). D'une manière permet à cette plateforme de connaître toutes les modifications dans les composant cyber physique, au moment où ils se produisent, d'autre part cette interface crée une connexion avec la plateforme Eclipse Ditto, pour être capable de suivre les changements au niveau de cette plateforme et ainsi connaître les changements qui se produisent au niveau du système Cyber Physical.

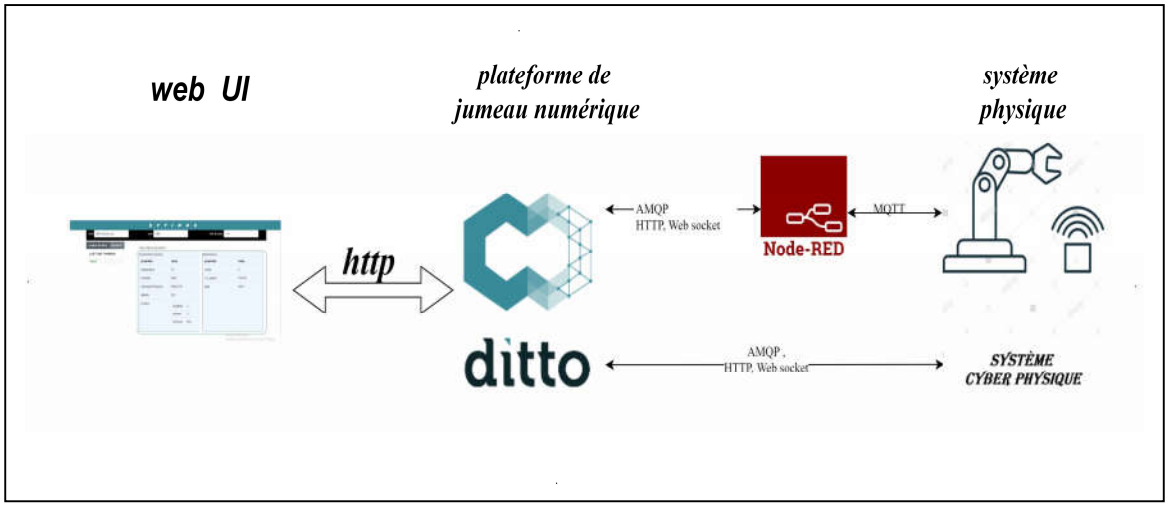


Figure III. 15: *Intégration avec le système physique.*

8. Conclusion

Dans ce chapitre, nous avons présenté l'approche que nous avons proposée pour générer des jumeaux numériques à partir de la modélisation du système cyber physique via un diagramme de bloc. Nous avons expliqué les différentes étapes de notre démarche, en commençant par la proposition d'une annotation pour la modélisation. Après, nous avons présenté nos transformations (M2M, M2T) pour construire la représentation du jumeau numérique à partir de la modélisation SysML du système cyber-physique. Enfin, nous avons expliqué l'objectif de l'application web permettant la création du jumeau numérique et ses fonctionnalités ainsi que l'architecture de son intégration avec le système physique.

Le chapitre suivant sera consacré à la présentation de l'implémentation de notre approche et son évaluation par un cas d'étude utilisant l'application conçue et développée pour supporter notre démarche de conception des jumeaux numériques.

CHAPITRE IV : Implémentation et étude de cas.

1. Introduction

Le but de ce chapitre est d'implémenter la démarche proposée dans le chapitre précédent et de l'évaluer via un cas d'étude. Il sert aussi à présenter l'application web développée pour supporter la démarche proposée. Les outils et les technologies employés sont au fur et à mesure expliqués. L'implémentation a passé par trois étapes principales, la transformation des diagrammes de bloc vers le modèle du jumeau numérique, la transformation du modèle du jumeau numérique vers un fichier JSON contient les propriétés de ce modèle et finalement l'application web proposée pour la création du jumeau numérique et le contrôle du système cyber-physique.

2. Implémentation

Dans la partie implémentation nous avons expliqué la modélisation du système cyber physique, les outils utilisés, la façon d'annoter les blocs et on a aussi donné un exemple illustratif, puis nous expliquons les règles de transformation M2M et M2T, en donnant des exemples, finalement nous expliquons le format d'exportation du projet et comment nous intégrons cette exportation avec l'application web.

2.1. Modélisation

Afin de modéliser le système cyber physique, on a utilisé l'outil papyrus [37] qui est un outil intégré dans éclipse PolarSys [38], Papyrus fournit des éditeurs de diagrammes pour les langages de modélisation basés sur les champs EMF, notamment UML 2 et SysML, nous utilisons pour modéliser le système cyber physique via le diagramme de bloc. Aussi pour ajouter les commentaires sur les blocs, les Figure IV.1, Figure IV.2, Figure IV.3 illustrent les commentaires qui doivent être placés dans les blocs physique, cyber physique et les acteurs.

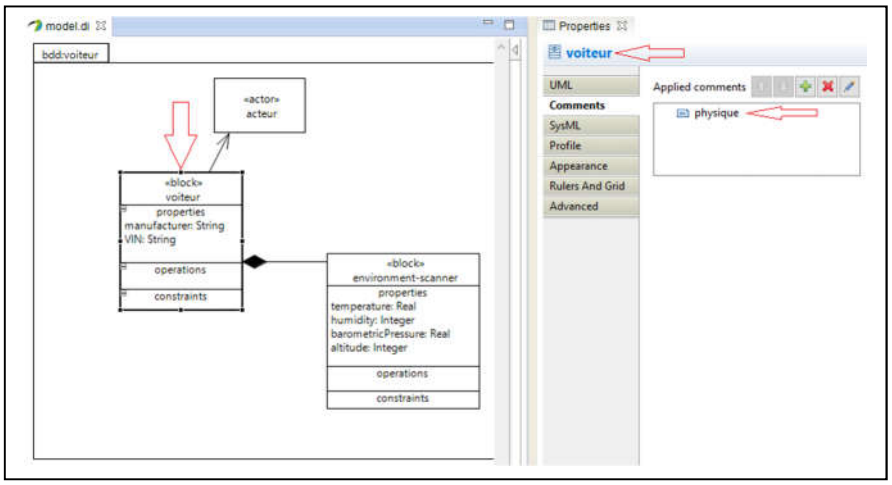


Figure IV. 1*: Annotation des blocs physiques.*

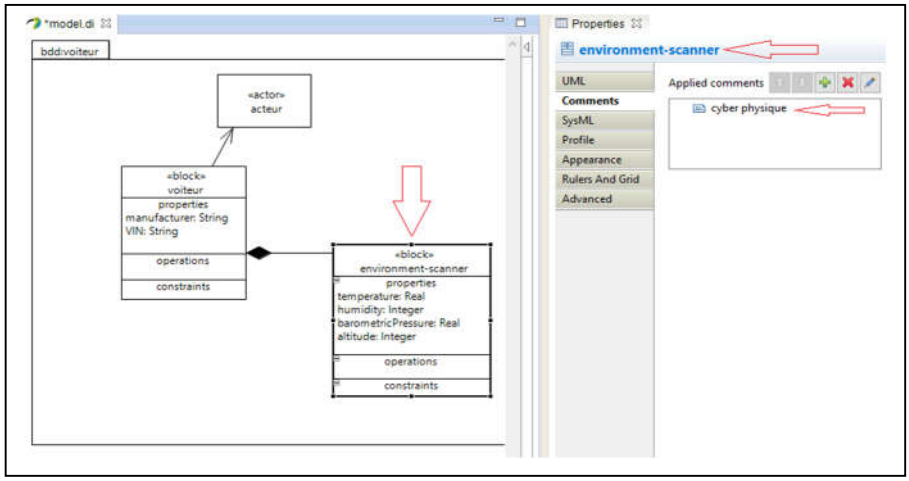


Figure IV. 2:*Annotation des blocs cyber physiques.*

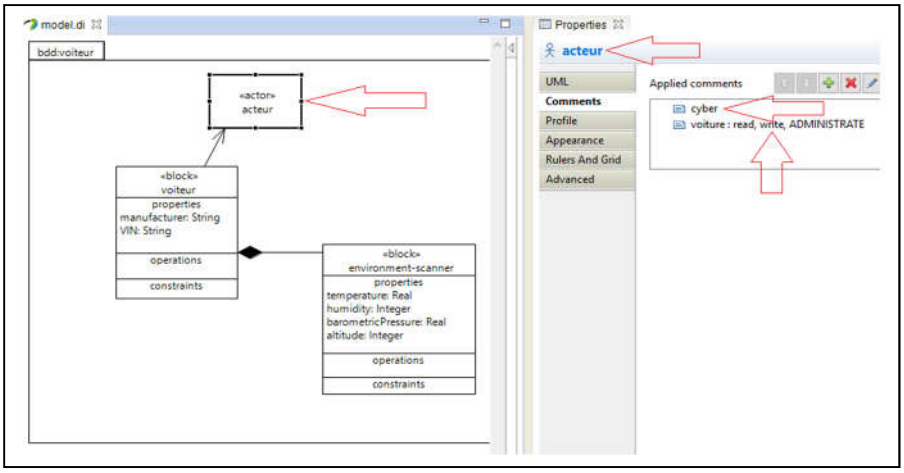


Figure IV. 3*:Annotation des acteurs.*

2.2. Transformation

Pour définir une transformation des modèles, on peut utiliser un langage non formel, un langage d'action pour représenter l'algorithme de la transformation, ou encore un langage bien défini de mapping des modèles.

2.2.1. Transformation ATL

Pour transformer les concepts du diagramme de bloc (BDD) vers les concepts du modèle du jumeau numérique, nous avons utilisé le langage de transformation atlas ATL (atlas transformation language), qui permet de transformer chaque bloc du modèle via des règles.

– **Transformation des concepts physique**

On utilise la règle ATL getthing pour transformer les blocs (classe) physiques vers les objets (thing) du jumeau numérique.

```
57 rule getthing {
58     from
59         c :MM1!Class(c.ownedComment-> select(it | it.body = 'physique').notEmpty
60     to
61         DT : MM2!Thing  (
62             thingId <- c.name.debug(),
63             attributes <- (c.primitive() ->collect(t | thisModule.getatribute(t)
64             Features <- c.helpergetFeature()->collect(e| thisModule.getfeature(e
65             ACL <- c.getactor() -> collect(i | thisModule.getACL(c.name,i))
66             )
67 }
```

Figure IV. 4: *Règle ATL des objets (thing).*

– **Transformation des concepts cyber physique**

On utilise la règle ATL getfeature pour transformer les blocs (classe) cybers physiques vers les fonctionnalités (feature) du jumeau numérique.

```
71 lazy rule getfeature {
72     from
73         f :MM1!Class
74     to
75         DTf : MM2!Feature  (
76             name <- f.name,
77             properties <- (f.primitive() -> collect(p | thisModule.getproperty(p
78              )
79 }
```

Figure IV. 5*: Règle ATL des fonctionnalités (feature).*

– **Transformation des concepts cyber**

On utilise la règle getAclentry pour transformer les droits d'accès de chaque acteur vers les concepts du jumeau numérique liste de contrôle d'accès aclentry (read, write, administrate, subject).

```
134 lazy rule getAclentry {
135     from
136         s : MM1!StringExpression,
137         a : MM1!Actor
138     to
139         dt : MM2!AclEntry (
140             READ <- a.getREAD(s),
141             WRITE <-a.getWRITE(s),
142             ADMINISTRATE <- a.getADMINISTRATE(s),
143             subject <- a.name
144         )
145 }
```

Figure IV. 6*: Règle ATL de la liste de contrôle des accédés.*

2.2.2. Transformation ACCELEO

Afin de faire la transformation du diagramme de bloc vers le modèle de jumeau numérique, on a utilisé l'outil ACCELEO, qui est un générateur de code source pour générer le fichier JSON.

– **Transformation des objets (thing)**

On utilise une règle de transformation generateElement(), qui prend le modèle de jumeau numérique sous format xml(.xmi) généré appartient à la première transformation et donne une présentation de la forme JSON du modèle du jumeau numérique, un exemple sur la règle illustré dans la figure IV.7.

```
[template public generateElement(aThing : Thing)]
[comment @main/]
[file ('Things.json', true, 'UTF-8')]
      [ '[' /]
            {
            "thingId": "www.eclipse.org:[aThing.thingId/]",
            "acl": {
                [for (it : AclEntry | aThing.ACL.AclEntry)separator
('',\n')]
                  "[it.subject/]": {
                        "READ": [it.READ/],
                        "WRITE": [it.WRITE/],
                        "ADMINISTRATE": [it.ADMINISTRATE/]
                  }[/for]
            },
            "attributes": {……},
            "features": {
                  [for (f : Feature | aThing.Features) separator (',\n')]
                  "[f.name/]": {
                        "properties": {……}
                  }[/for]

            }
      },
[/file]
[/template]
```

Figure IV. 7*: Règle ACCELEO pour transformer les objets.*

2.3.Intégrations des transformations

Pour donner à l'application web la possibilité d'accéder à des processus de transformation préalablement programmés et de les exécuter, nous avons exporté les projets (M2M, M2T) sous la forme d'un fichier jar, comme la figure IV.8 et la figure IV.9 illustrent, puis on les appelle par l'application à chaque fois on a besoin de ces transformations par un commande cmd « java -jar M2M.jar » et « java -jar M2T.jar ».

Pour exporter les fichiers jar des projets m2m et m2T.e On a passé par les étapes suivantes : export : Export -> Runnable JAR file.

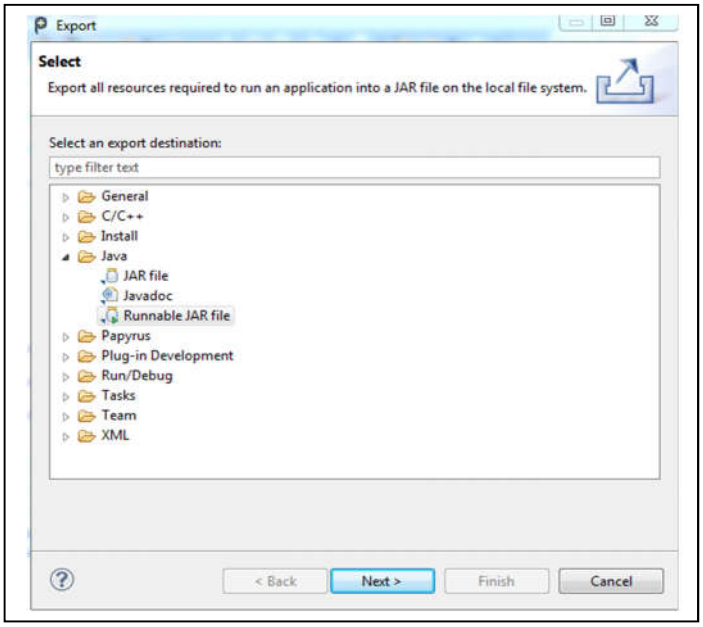


Figure IV. 8: *Exporte de jar étape 1.*

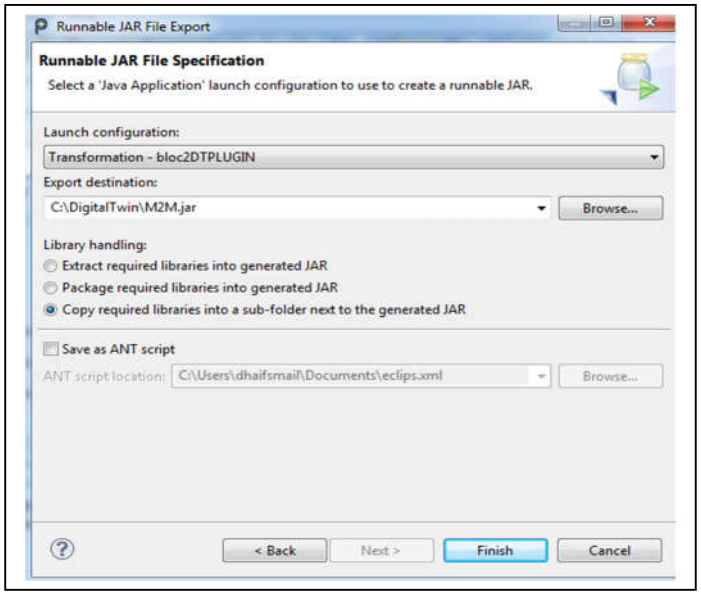


Figure IV. 9: *Exporte de jar étape 2.*

2.4.Plateforme des jumeaux numériques utilisé

Dans ce travail, on a utilisé la plateforme des jumeaux numériques Eclipse Ditto, Eclipse Ditto est un open-source projet d'Eclipse IoT qui fournit une fonctionnalité prête à l'emploi pour gérer l'état des jumeaux numériques, Il leur donne l'accès et sert comme un intermédiaire entre le monde physique et cette représentation numérique [45].

3. Etude de cas

Dans l'étude de cas on évoque trois points. En premier lieu, on explique le système utilisé et le diagramme de bloc. Ensuite la simulation du système, on utilise le node red [46] pour créer une simulation du système. A la fin, nous expliquons les interfaces de l'application web.

3.1. Système utilisé

Nous avons pris un exemple d'un système cyber physique d'une voiture pour l'étudier et l'appliquer une expérience pratique afin de consolider notre proposition d'établissement des jumeaux numériques, le système informatique de la voiture contient de nombreuses fonctions, nous avons choisi dans cette étude de cas, les fonctions suivantes :

1. Contrôle et gestion de moteur.
2. Gestion de verrouillage des portes et système d'alarme.
3. Gestion de ABS / ESP.
4. Gestion de boîte de vitesse.

Cette opération est effectuée par des unités de contrôle qui collectent diverses informations de base décrivant l'état des systèmes de la voiture à l'aide de leurs capteurs associés, afin de donner des instructions permettant d'améliorer les performances de la voiture et d'augmenter sa durée de vie.

3.1.1. Contrôle et gestion de moteur

L'unité responsable de l'exécution de ce processus est l'unité de contrôle du moteur, qui est gérée en contrôlant l'injection de carburant et le moment de l'allumage du mélange (air essence), de sorte que la quantité de carburant à pomper soit calculée en mesurant la quantité d'air entrant dans le moteur, sa température et sa densité, et La pression atmosphérique à l'intérieur et à l'extérieur du moteur, la quantité d'oxygène restant dans la fumée d'échappement, Quant aux facteurs affectant le temps d'allumage du mélange à l'intérieur du moteur, ils sont: la position de vilebrequin et la position de l'arbre à cames, et la volume du déclic à l'intérieur du moteur.

3.1.2. Gestion de verrouillage des portes et système d'alarme

L'unité responsable de ce processus est l'unité de commande de porte, qui ferme et ouvre les portes après avoir reçu une fréquence contenant son propre code.

La centrale de porte activerait le système d'alarme si les portes sont verrouillées, Il émet également un signal d'avertissement en activant la sirène d'alarme et en allumant les variateurs de lumière, en cas de grève sur les portes ou les fenêtres et lors de la réception de l'ordre de fermer ou d'ouvrir les portes.

3.1.3. Gestion d'ABS / ESP

L'unité responsable de cette opération est le boîtier électronique ABS / ESP, dont le rôle est de maintenir la stabilité de la voiture et d'empêcher le blocage des roues en cas de pression soudaine sur les freins. Son état est ajusté en ajustant la pression d'huile de frein de chaque roue du véhicule en fonction de sa position instantanée, il détermine également la direction de la voiture sur les trois axes longitudinal, transversal et vertical' et la proportion de son inclinaison pour pouvoir identifier les situations de glissade dangereuses et les éviter en réglant la pression d'huile de frein pour chaque roue, ainsi qu'en augmentant ou en diminuant le nombre de tours moteur

En cas de danger, ce système informe le conducteur, et l'avertir par les lampes sur le tableau de bord.

3.1.4. Gestion de boîte de vitesse :

L'unité responsable de ce processus est l'unité de contrôle de la transmission, il change la vitesse de la voiture (vers le haut ou vers le bas), Il change le mode de conduite en fonction de l'état de la route et du style de conduite du conducteur , Il verrouille également le levier de vitesses sur certaines positions dans certaines situations déterminées en mesurant la vitesse de voiture, la pression et la température de l'huile à l'intérieur de la boîte de vitesses, afin d'assurer la sécurité de la voiture et d'éviter certains scénarios graves pendant la conduite.

La figure IV.10, illustre la modélisation du système utilisé vais le diagramme de bloc.

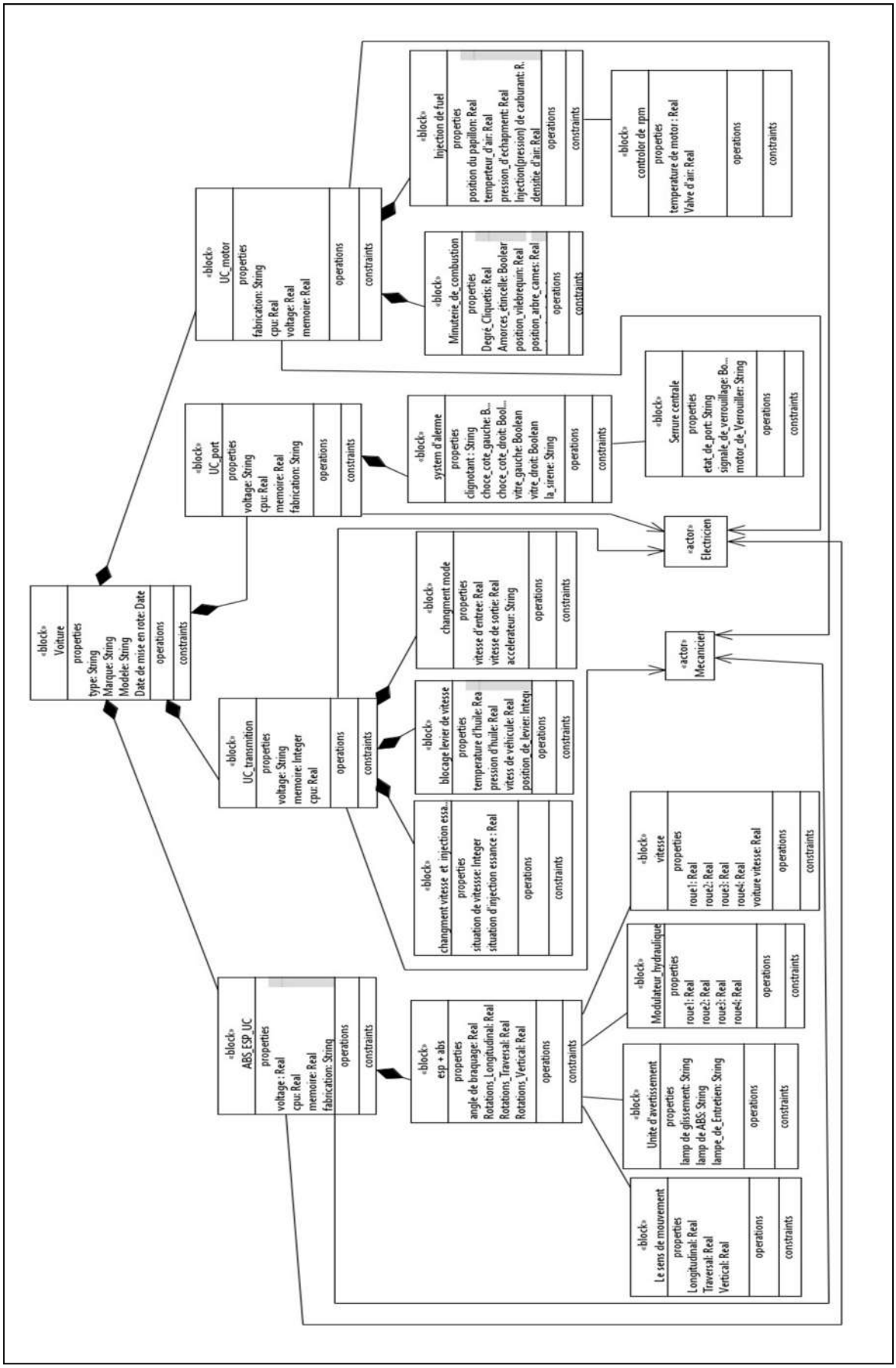


Figure IV. 10 *: Diagramme de bloc de la voiture.*

3.2.Simulation du système cyber-physique

Afin de simuler le système cyber physique, nous avons utilisé l'outil de simulation node red [46], pour modéliser le système et ses fonctionnalités, et programmer leurs différentes fonctions.

La figure IV.11, illustre la simulation des fonctionnalités d'unité de gestion d'ABS / ESP.

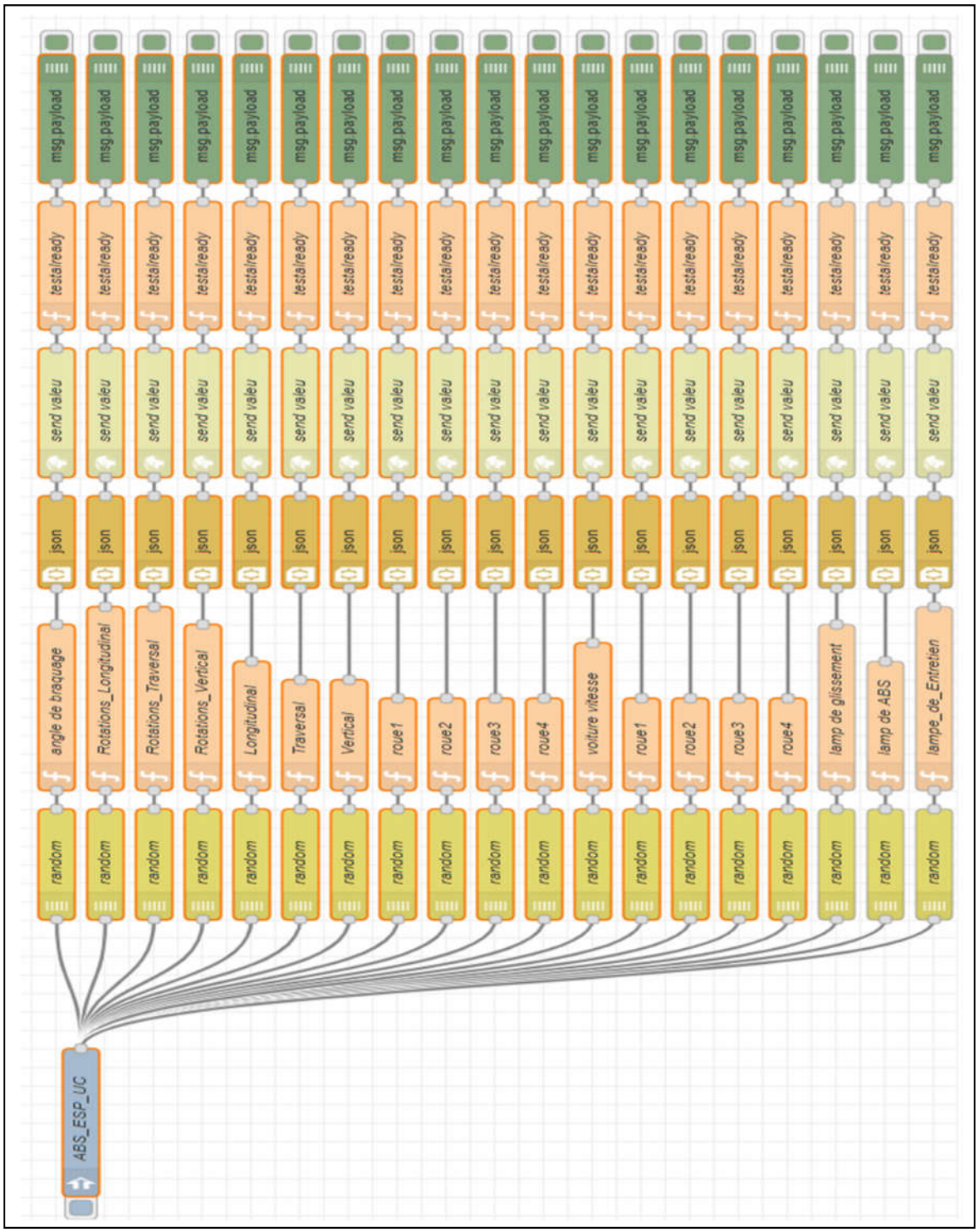


Figure IV. 11 : *Simulation d'unité de gestion d'ABS / ESP.*

La figure IV.12, illustre la simulation de la fonctionnalité d'unité de contrôle du moteur.

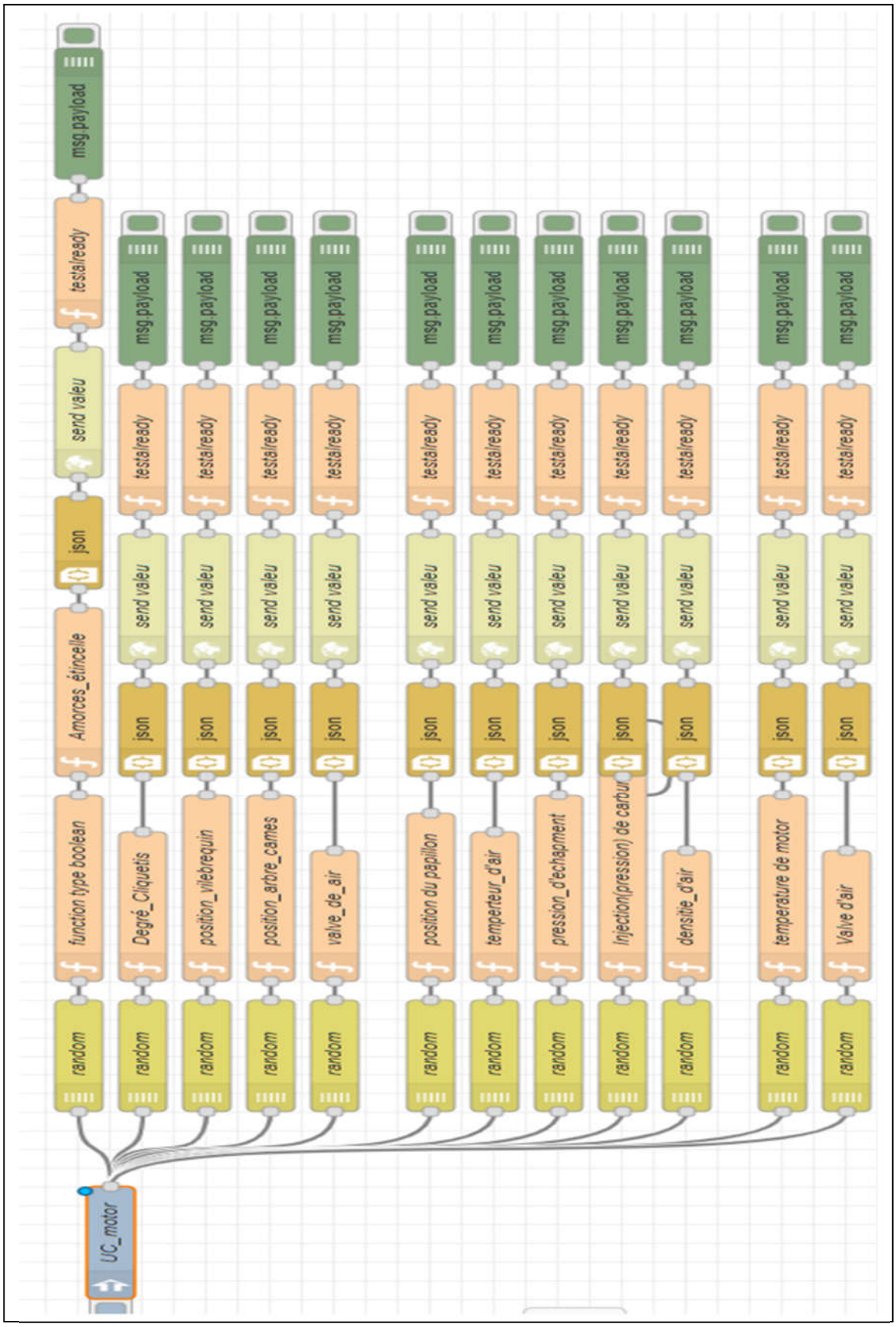


Figure IV. 12 : *Simulation d'unité de contrôle du moteur.*

La figure IV.13, illustre la simulation de la fonctionnalité d'unité de gestion du port.

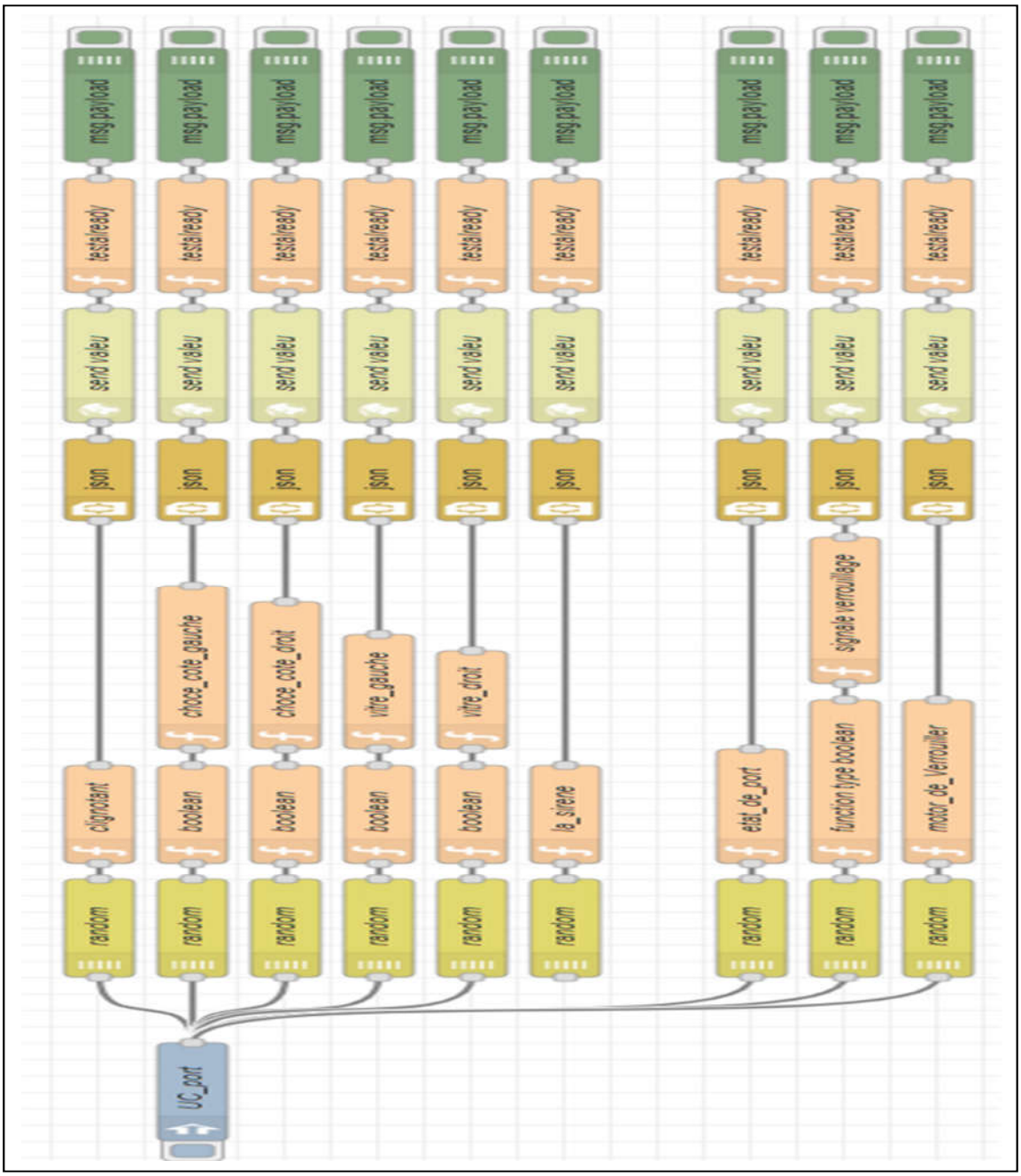


Figure IV. 13 : *Simulation d'unité de gestion du port.*

La figure IV.14, illustre la simulation de la fonctionnalité d'unité de gestion de la transmission.

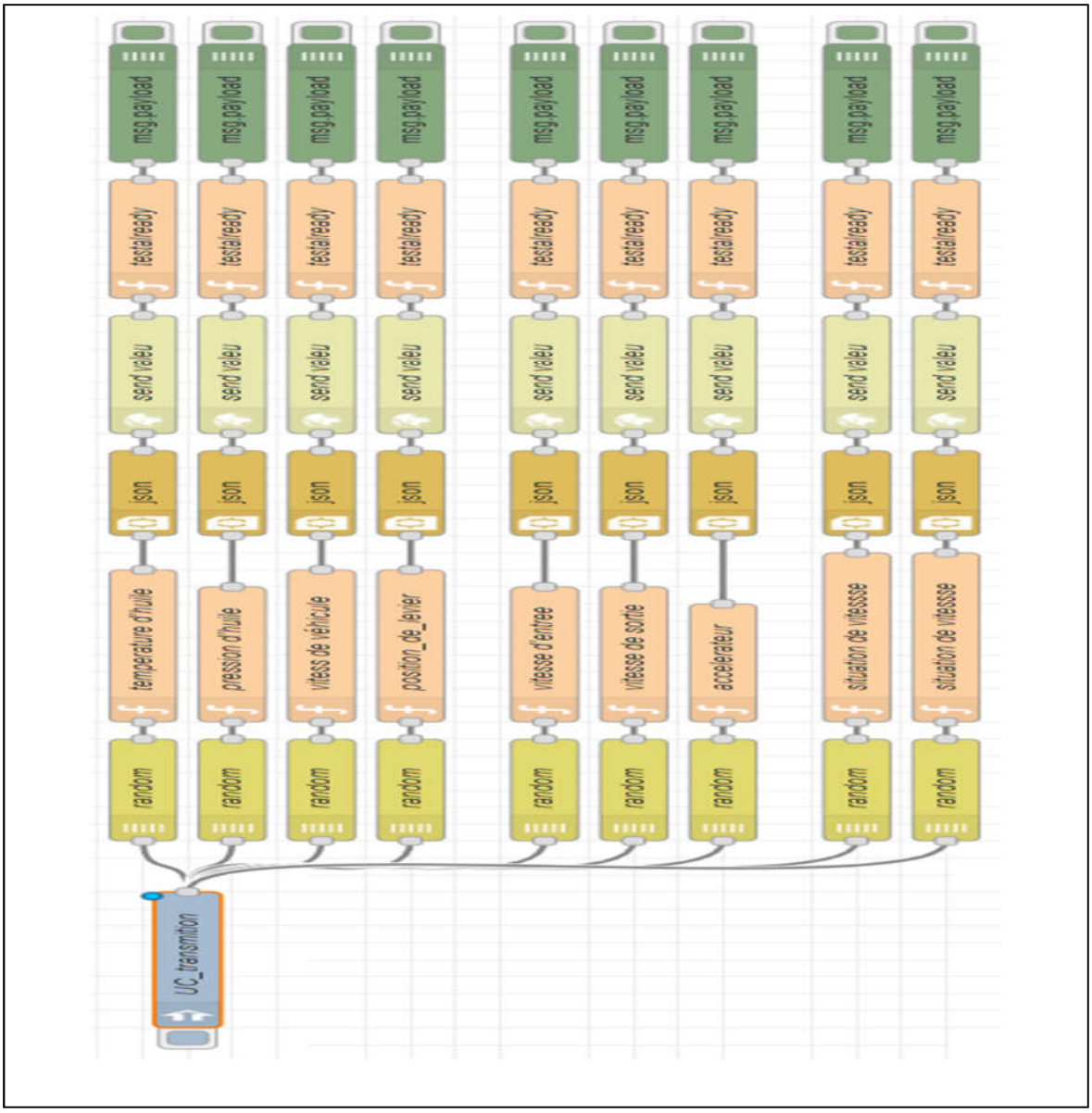


Figure IV. 14 : *Simulation d'unité de gestion de la transmission.*

3.3.Interfaces de l'application web

Pour réaliser notre application, nous avons utilisé le langage html [48], le langage de programmation JAVASCRIPT[39] et la langage PHP [40] dédié à la création des applications web dynamique, on a utilisé aussi le langage CSS[42] et la technologie BOOTSRAP[41] pour le style, celui-ci nous l'avons manipulé dans un environnement de développement intitulé WAMPSERVER[43], qui nous permet d'héberger notre application.

3.3.1. Interface d'accueil

La Figure IV.15 illustre l'interface d'accueil de l'application web, cette interface contient des descriptions des étapes qui permettent à l'utilisateur de lancer un nouveau processus de la création d'un jumeau numérique :



Figure IV. 15: *Interface de la page accueil.*

3.3.2. Interface de transformation

La Figure IV. 16 illustre l'interface montrant la première étape de la création des jumeaux numériques, elle permet à l'utilisateur de sélectionner le diagramme de bloc du système cyber-physique.

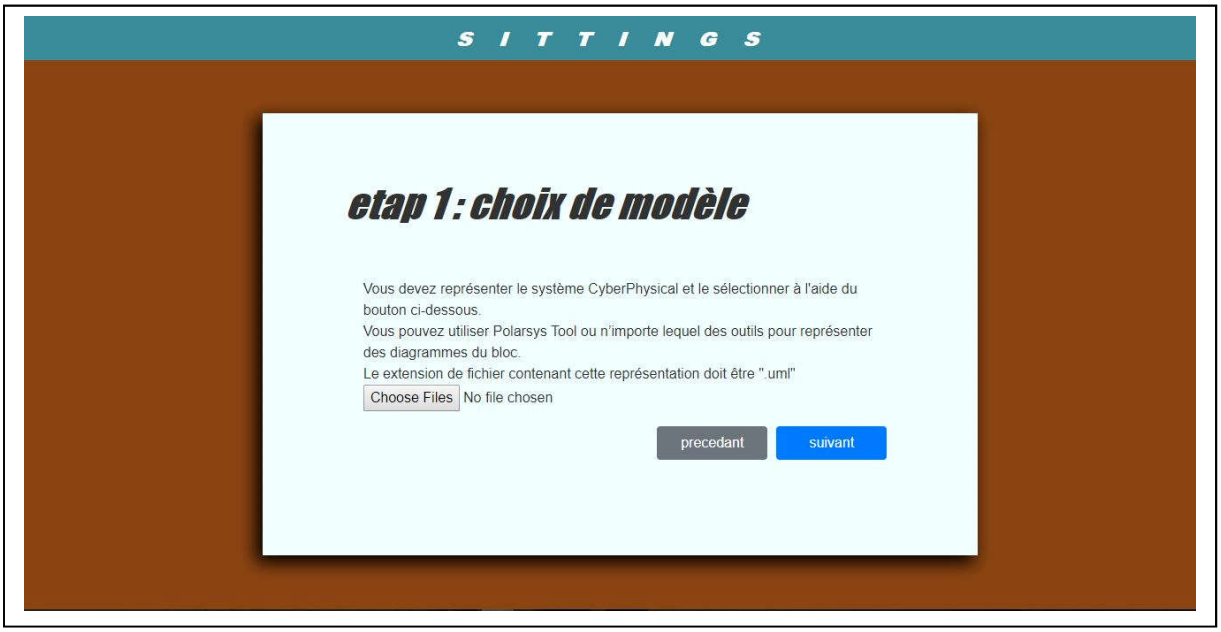


Figure IV. 16: *Interface de sélection de modèle.*

La Figure IV. 17 illustre l'interface de transformation M2M qui permet à l'utilisateur de lancer le processus de transformation du diagramme de bloc du système cyber-physique vers le modèle du jumeau numérique.

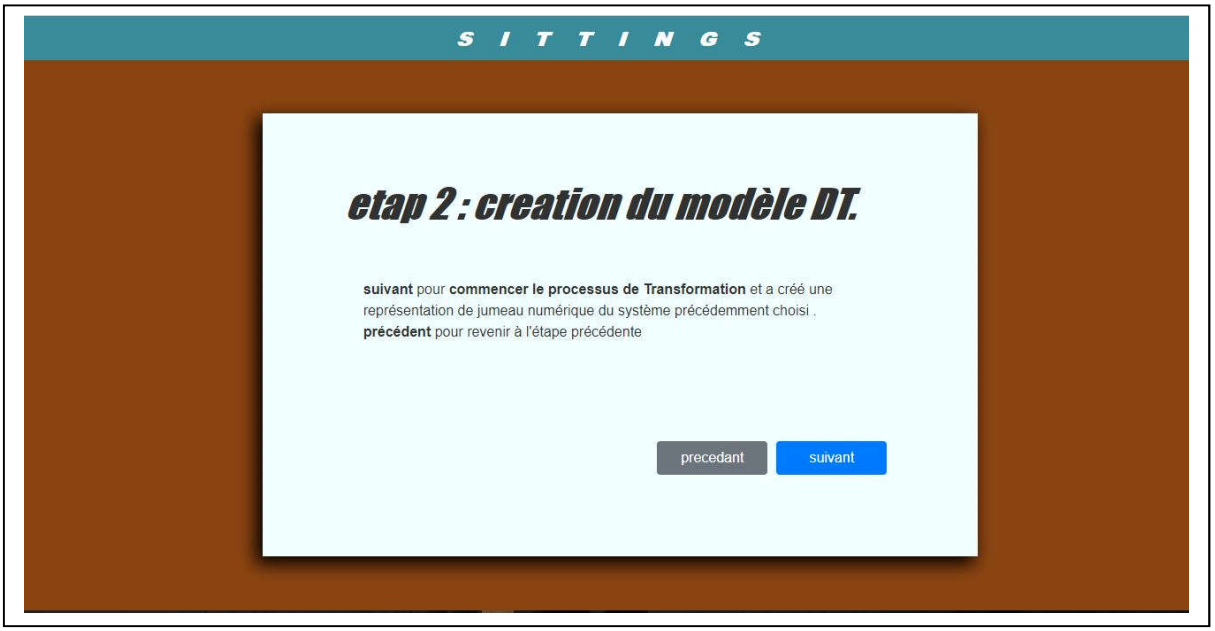


Figure IV. 17: *Interface transformation M2M.*

La Figure IV. 18 illustre l'interface qui s'affiche lors de la transformation du diagramme de bloc du système Cyber-physique vers un modèle de jumeau numérique.

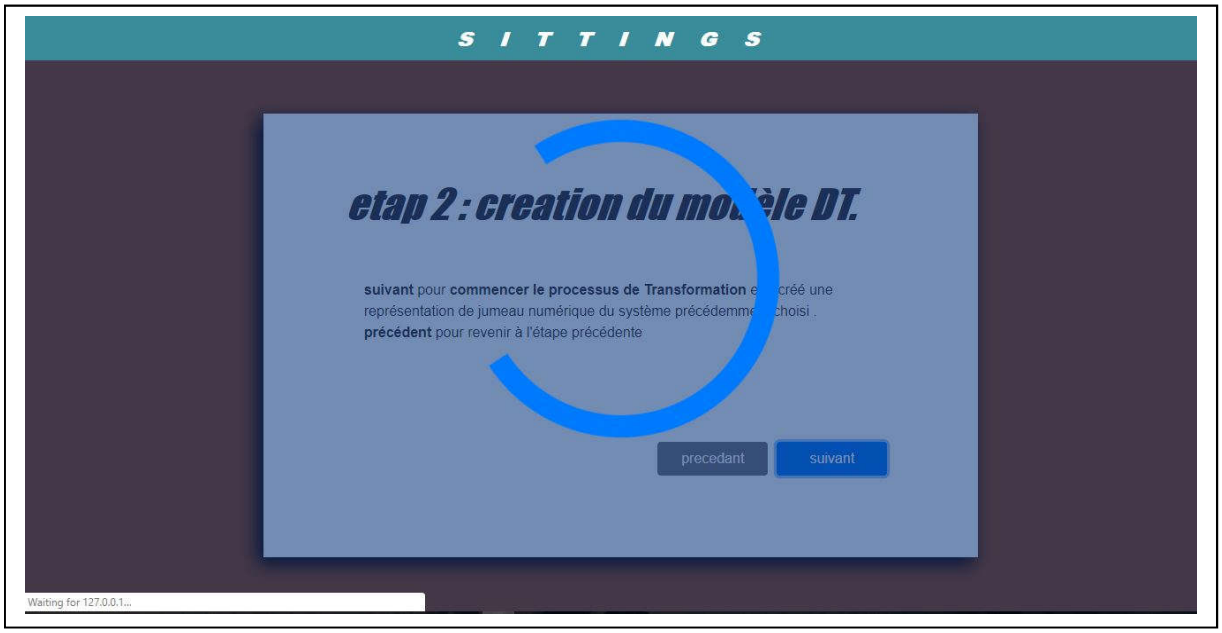


Figure IV. 18: *Interface transformation M2M (2).*

La Figure IV. 19 illustre l'interface de transformation M2T qui montre la troisième étape de création de jumeaux numérique, il permet à l'utilisateur de lancer le processus de transformation de diagramme du modèle du jumeau numérique vers le format JSON.

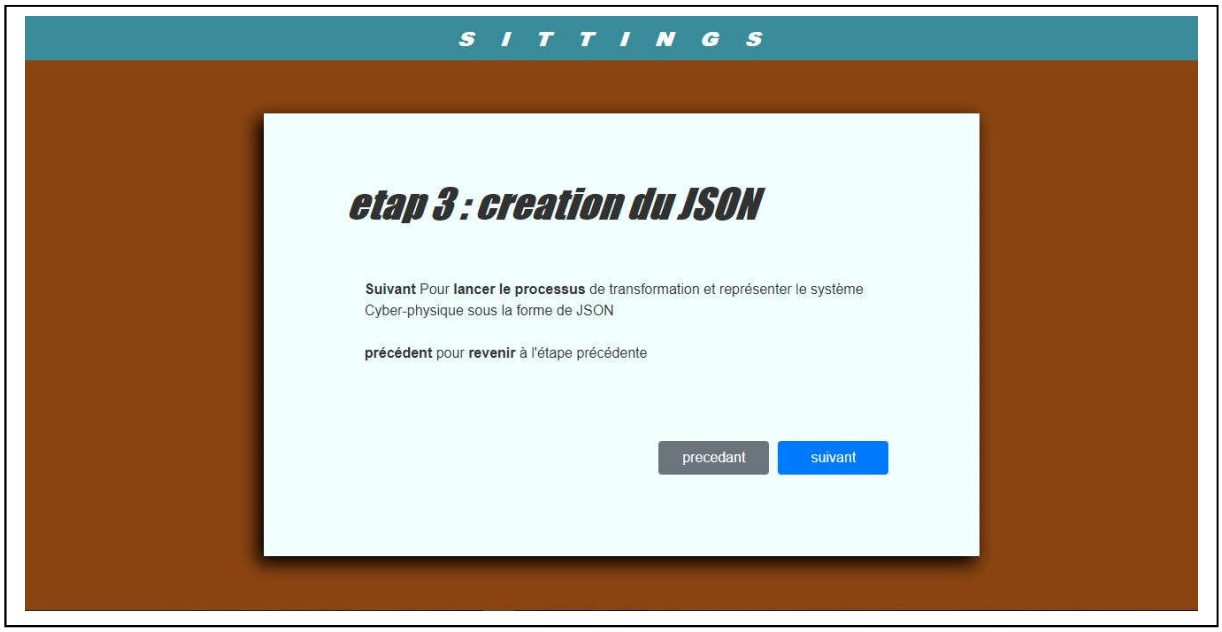


Figure IV. 19: *Interface transformation M2T.*

La Figure IV.20 montre l'interface qui s'affiche lors de la transformation du diagramme de bloc de système Cyber Physical vers un modèle de jumeau numérique.

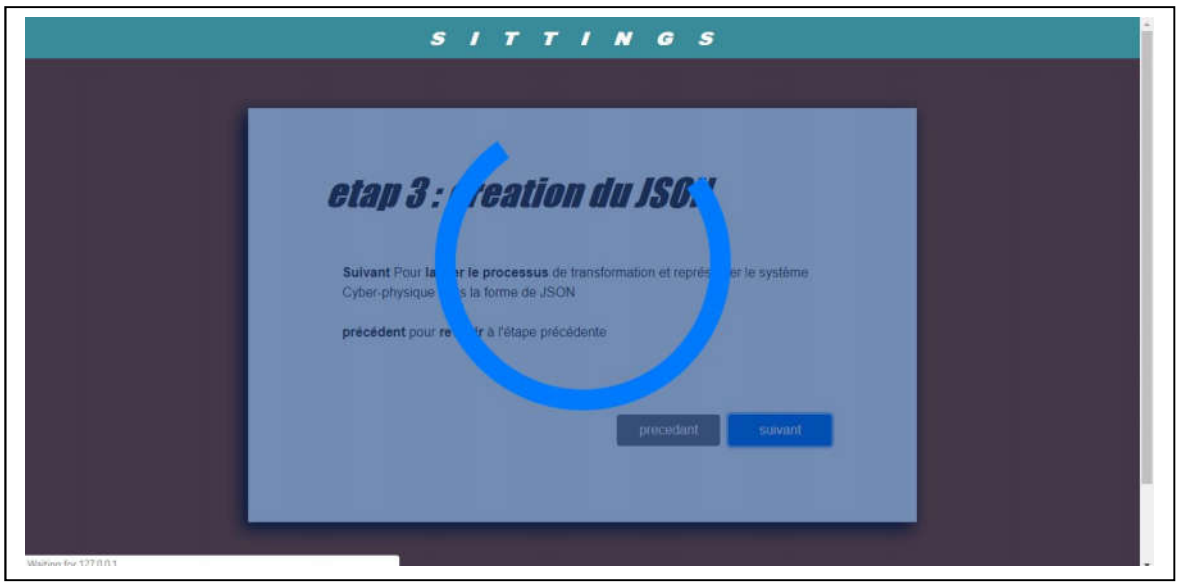


Figure IV. 20*: Interface transformation M2T.*

3.3.3. Interface de création des objets (thing)

Cette interface s'affiche une fois la transformation terminée, elle permet à l'utilisateur de créer et d'éditer les objets dans la plateforme des jumeaux numériques Eclipse Ditto.

Ces interfaces peuvent être divisés en trois éléments de base, le premier est un formulaire qui permet à l'utilisateur de saisir l'adresse de plateforme du jumeau numérique et les informations de compte utilisées pour créer les objets. Le second est un menu latéral qui inclut une liste des objets non créés, qui permet à l'utilisateur de choisir ce qu'il veut créer, une liste des objets créés permettant à l'utilisateur de choisir celui s'il désir de le modifier. Elle contient également un bouton pour afficher les objets créés. Le troisième est l'endroit où l'objet sélectionné est affiché, la figure IV.21 illustre l'interface de la création des objets.

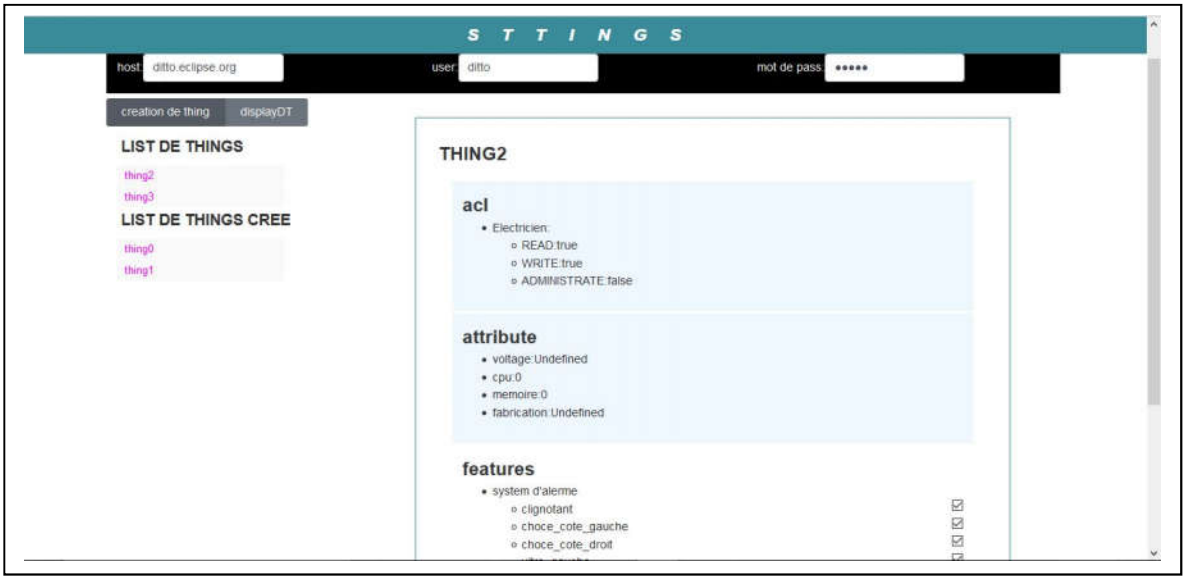


Figure IV. 21: *Interface de gestion des objets 1.*

Après la sélection d'un objet, on arrive à la partie de sélectionner les propriétés qui ont besoin des jumeaux numériques, comme la figure IV.22 illustre :



Figure IV. 22: *Interface de gestion d'objet 2.*

Lorsque vous sélectionnez les propriétés et vous cliquez sur le bouton créer, une fenêtre contenant le résultat de la sélection apparaît, il nous permet de le modifier manuellement pour le créer en cliquant sur le bouton créer, comme illustré par la figure IV.23 :

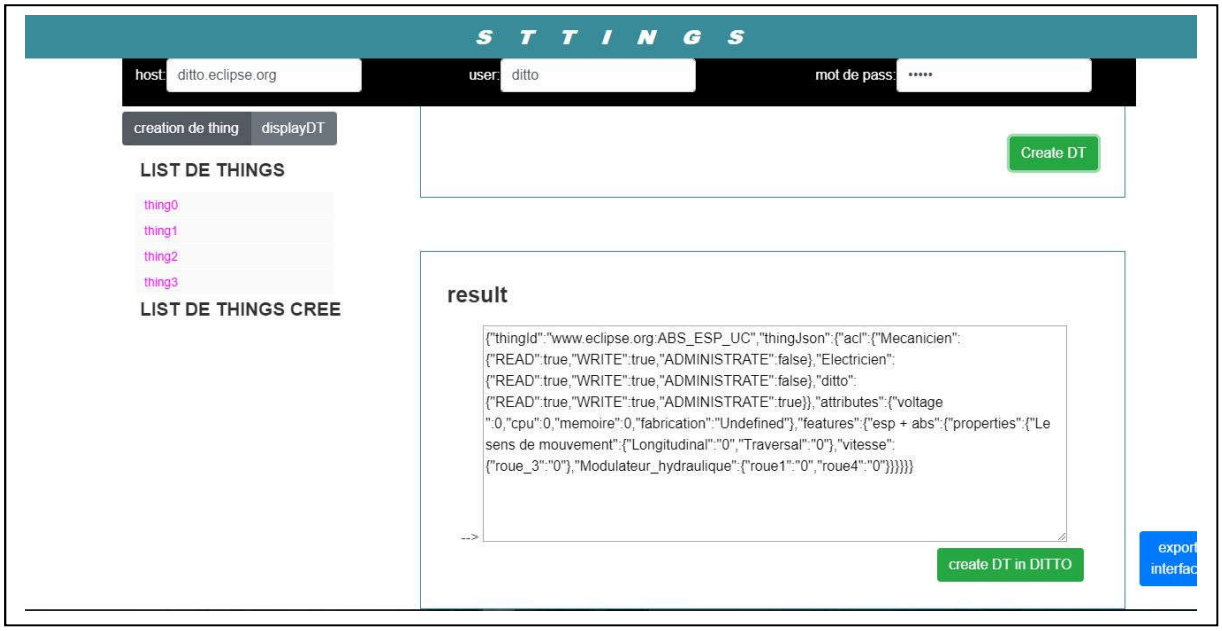


Figure IV. 23: *Interface de gestion d'objet 3.*

La Figure IV.24 ci-dessous illustre l'interface qui apparaît lorsque la création des objets est échouée.

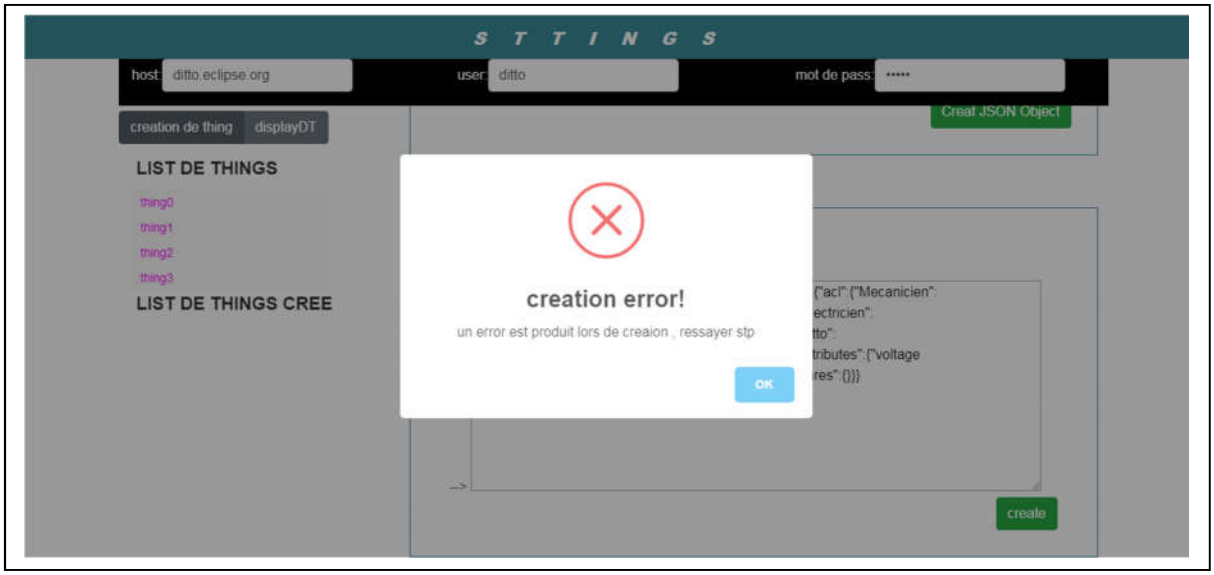


Figure IV. 24: *Interface création a échoué.*

La Figure IV. 25 ci-dessous montre l'interface qui apparaît lorsque la création des objets est réussie.

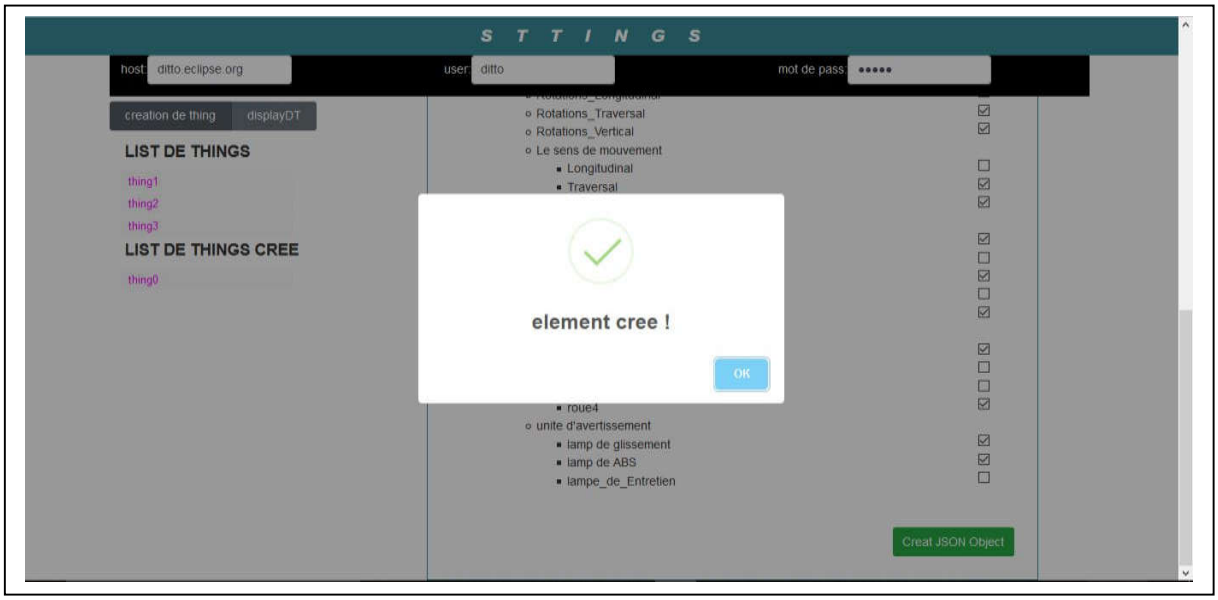


Figure IV. 25: *Interface de création réussie .*

La figure IV.26 ci-dessus montre l'interface qui permet à l'utilisateur d'identifier les actionneurs.

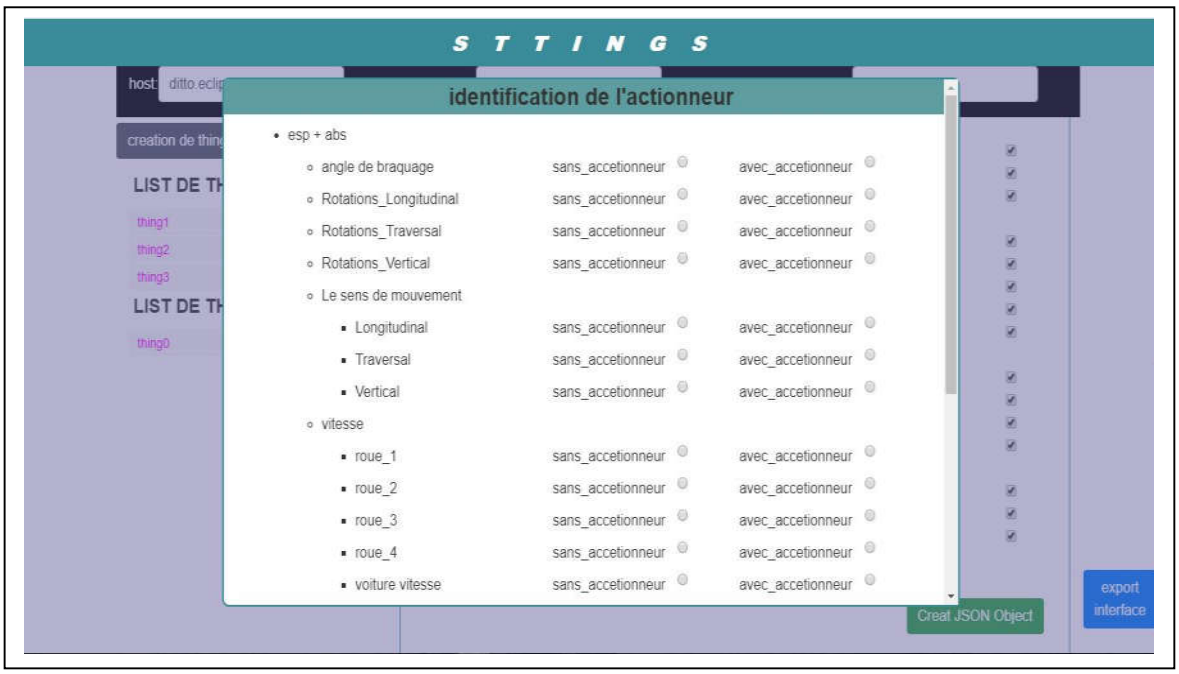


Figure IV. 26 : *Formulaire d'identification des actionneurs.*

La figure IV.27 ci-dessus montre l'interface du jumeau numérique créé. Les valeurs de propriétés sont remplies par des valeurs par défaut avant la connexion avec la plateforme Eclipse Ditto.

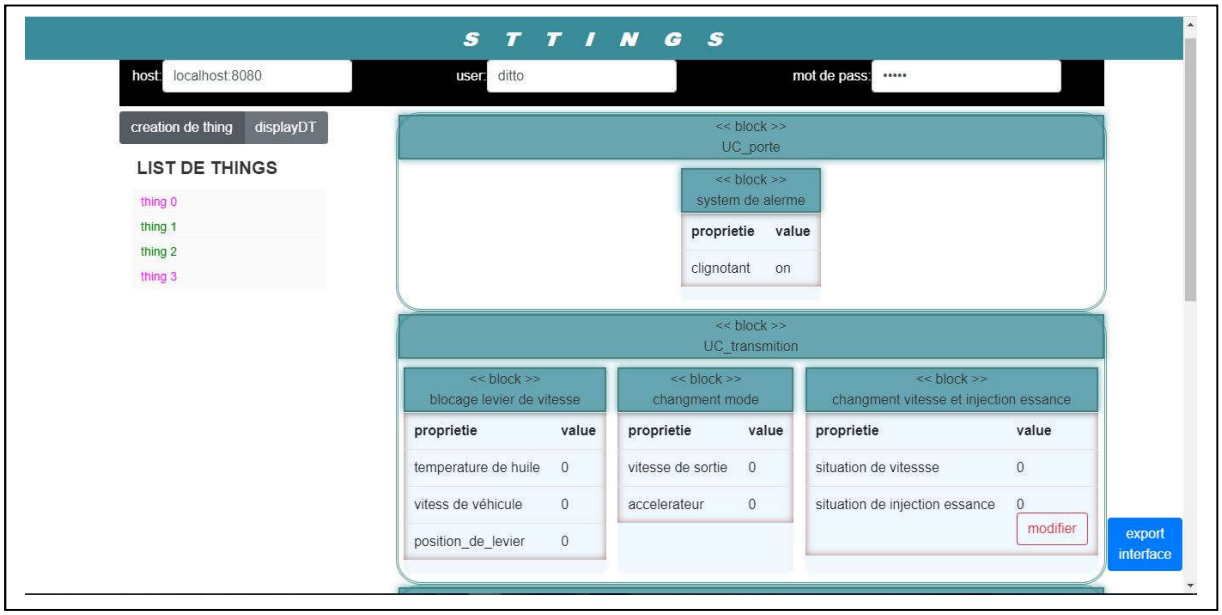


Figure IV. 27*: Interface de jumeau numérique 1.*

La figure IV.28, montre l'interface du jumeau numérique après la connexion avec le système cyber-physique via la plateforme Eclipse Ditto.



Figure IV. 28: *Interface du jumeau numérique après connexion avec le système cyber-physique via Ditto.*

La figure IV.29, illustre la structure des fichiers du jumeau numérique généré.

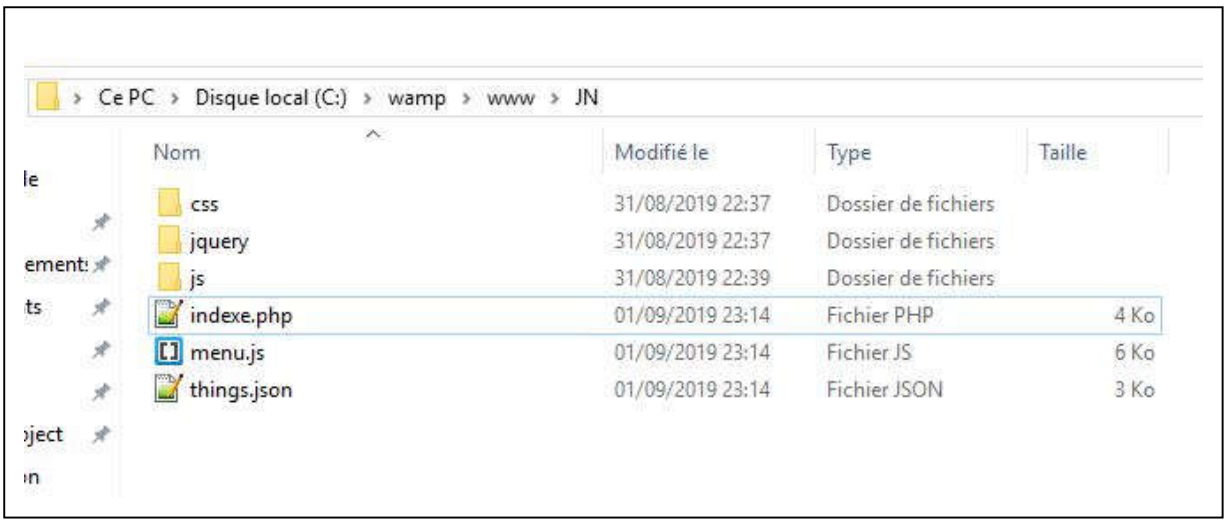


Figure IV. 29 *:* Structure de fichiers du jumeau numérique généré**.**

4. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre démarche pour la création des jumeaux numériques. Nous avons présenté l'environnement de développement utilisés ainsi que les différents outils employés et nous avons donné une description détaillée des étapes suivies pour l'implémentation de l'application web permettant la génération des jumeaux numériques selon la démarche proposée. La démarche a été évaluée via un cas d'étude et avec des captures des interfaces de l'application web conçues.

Conclusion générale

Un jumeau numérique est une réplique numérique d'un objet, d'un processus ou d'un système pouvant être utilisée à diverses fins. Il présente de nombreux avantages, il permet de contrôler facilement les différents composants d'un système, ce qui permet d'éviter des nombreux problèmes avant qu'ils ne se produisent, mais malgré ses multiples avantages, le problème de sa construction limite encore sa propagation, puisqu'il n'existe aucun moyen général de créer un jumeau numérique.

Dans ce travail, on s'est intéressé à la proposition d'une approche pour faciliter le développement des jumeaux numériques d'un système cyber-physique en se basant sur l'ingénierie systèmes basée sur les modèle et en utilisant le langage SysML. Comme première étape, nous avons proposé de modéliser le système cyber-physique via le diagramme de bloc. Comme SysML propose le mécanisme d'annotation (possibilité de rajouter des commentaires), nous avons défini une annotation pour distinguer les différents composants du système modélisé. Par la suite, on a transformé le diagramme de bloc vers le modèle du jumeau numérique tout en suivant une démarche d'ingénierie basée sur les modèles. Ainsi, le langage ATL et le langage ACCELEO ont été utilisé pour implémenter la transformation.

Dans le but de faciliter la création, nous avons groupé toutes les étapes précédentes dans une application web qui lance le processus de transformation et qui permet de gérer le processus de de développement des jumeaux numériques et générer une application web permettant de l'exploiter. La démarche proposée a été évaluée et expérimentée en considérant le développement d'un jumeau numérique pour un système concret que nous avons simulé via l'outil Nodered.

Ce travail nous a permis de connaître les concepts relatifs aux systèmes cyber-physiques et aux jumeaux numériques et d'apprendre les différents aspects concernant l'ingénierie des systèmes basée sur les modèles, notamment, la modélisation et le développement des systèmes via la transformation de modèles.

Comme perspective, il est important de développer d'autres fonctionnalités relatives aux jumeaux numériques en ajoutant la possibilité de collecter, d'optimiser et de traiter les données du système à l'aide d'algorithmes de Deep Learning.

Bibliographies.

[1] TAO, Fei, CHENG, Jiangfeng, QI, Qinglin, *et al.* Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 2018, vol. 94, no 9-12, p. 3563-3576.

[2] https://www.eclipse.org/ditto/, consulté le 31/08/2019.

[3] "Industry 4.0 | Festo Corporate." [Online]. Available: https://www.festo.com/group/en/cms/10966.htm. [Accessed: 14-Apr-2018].

[4] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016, vol. 2016–March, pp. 3928–3937.

[5] J. Smit, S. Kreutzer, C. Moeller, and M. Carlberg, "Industry 4.0," 2016.

[6] Lee, E.A., 2006. Cyber-Physical Systems - Are Computing Foundations Adequate? in: Position Paper for NSF Workshop on Cyber-Physical Systems. Austin, Texas

[7] CARDIN, Olivier. *Contribution à la conception, l'évaluation et l'implémentation de systèmes de production cyber-physiques*. 2016. PhD Thesis. Université de nantes.

[8] FAUCHEUX, Michel. *Norbert Wiener, le Golem et la cybernétique, éléments de fantastique technologique*. Editions du Sandre, 2017.

[9] NG, Irene CL; WAKENSHAW, Susan YL. The Internet-of-Things: Review and research directions. *International Journal of Research in Marketing*, 2017, 34.1: 3-21.

[10] Michael Parks, Maîtrise en ingénierie des systèmes de l'Université Johns Hopkins, https://eu.mouser.com/applications/digital-twins-offer-insight/.

[11] M. Grieves and J. Vickers, "Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, 2016, pp. 85–113.

[12] A. Parrott and W. Lane, "Industry 4.0 and the Digital Twin," Deloitte University Press, May-2017. [Online]. Available: https://dupress.deloitte.com/dup-us-en/focus/industry 4-0/digital-twin-technology-smart-factory.html. [Accessed: 15-Mar-2018].

[13] F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," *IEEE Access*, vol. 5, pp. 20418–20427, 2017.

[14] Omg. meta object facility (mof) 2.0 query/view/transformation specification version 1.2. http ://www.omg.org/spec/qvt/1.0/pdf/, april 2008.

[15] Omg. meta object facility (mof) core specification version 2.0, 2006.

[16] G. booch. object-oriented analysis and design with applications, second edition. addison-wesley, reading, ma, 1997.

[17] Object management group. what is omg-uml and why is it important? http ://www.omg.org/ news/pr97/umlprimer.html. last visited april 2006.

[18] Systems Engineering. Verification and Validation in Systems Engineering.

[19] C. Schönfelder, Muße – Garant für unternehmerischen Erfolg. 2018.

[20] Object management group. Unified modeling language: Infrastructure version 2.1.1, february 2007.

[21] Systems Engineering. *Verification and Validation in Systems Engineering*.

[22] Object management group. Unified modeling language: Infrastructure version 2.1.1, february 2007.

[23] Systems Engineering. *Verification and Validation in Systems Engineering*

[24] H A L Id. durable des systemes socio-environnementaux : To cite this version : l ' `1 optimisation durable des systemes socio - Ing ` enierie des syst ´ emes de d ` ecision en ´ univers. 2017.

[25] Memoire D E Magister and Mecheri Nacera. Remerciements. 2015.

[26] Diaw Samba, Lbath Rédouane, and Coulette Bernard. Etat de l'art sur le développement logiciel basé sur les transformations de modèles. 2010.

[27] M. Lachgar and A. Abdali. Şmodeling and generating the user interface of mobile devices and web development with dsl. 2015.

[28] H A L Id. Ingénierie dirigée par les Modelés (IDM) – Etat de l'art Benoıt Combemale To cite this version : HAL Id : hal-00371565 Ingénierie Dirige par les modèles.

[29] Bézivin Jean and Blanc Xavier. Mda : Vers un important changement de paradigme en genie logiciel. 2002.

[30] Object management group. request for proposal: Mof 2.0 query/views/transformations rep. http ://www.omg.org/cgi-bin/doc? ad/2002-4-10.

[31] http ://jamda.sourceforge.net.

[32] http ://wiki.eclipse.org//xpand.

[33] https ://www.eclipse.org/acceleo/documentation/.

[34] https://www.eclipse.org/ditto/basic-thing.html, consulter le 28/08/2019.

[35] https://www.eclipse.org/ditto/basic-acl.html, consulter le 28/08/2019.

[36] https://www.eclipse.org/ditto/basic-feature.html, consulter le 28/08/2019.

[37] https://www.polarsys.org/eclipse/papyrus consulter le 27/08/2019.

[38] https://www.polarsys.org/about-us consulter le 28/08/2019.

[39] Javascript. Date de consultation 06 2013. http://www.futura-sciences.com/fr/definition/t/internet-2/d/javascript_509/.

[40] http://creer-un-site.fr/sous-categorie-5-le-langage-php.php consulté le 16/02/2015

[41] https://www.w3.org/Style/CSS/Overview.fr.htmlconsulter le 16/09/2019.

[42] https://getbootstrap.com/ consulter le 16/09/2019.

[43] http://fr.wikipedia.org/wiki/WampServer ; consulter le 26/08/2019.

[45] https://projects.eclipse.org/projects/iot.ditto . Consulter le 26/08/2019.

[46] https ://nodered.org/, page consultée le 24/08/2019.

[47] Internet der Dinge/Industrie 4.0: Technologien - Anwendungen – Perspektiven, VDE-Trendreport, https://books.google.dz/books?id=RiKQtAEACAAJ, VDE, 2016.

[48] https://www.w3.org/html/. Consulter le 16/09/2019.