



HAL
open science

Modèle d'interaction liant Modèles d'actions métier et interactions tangibles en environnement virtuel

Jean-Michel Fazzari

► **To cite this version:**

Jean-Michel Fazzari. Modèle d'interaction liant Modèles d'actions métier et interactions tangibles en environnement virtuel. Intelligence artificielle [cs.AI]. École Nationale d'Ingénieurs de Brest, 2022. Français. NNT : 2022ENIB0015 . tel-04041487

HAL Id: tel-04041487

<https://theses.hal.science/tel-04041487>

Submitted on 22 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE

ENIB
ÉCOLE NATIONALE
D'INGÉNIEUR DE BREST

Ecole Doctorale N°601
*Mathématique et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : (voir liste des spécialités)

Par

Jean-Michel FAZZARI

**Modèle de reconnaissance d'actions
à partir des interactions en Réalité Mixte et Tangibles**

Thèse présentée et soutenue à PLOUZANÉ , le 5 septembre 2022
Unité de recherche : Lab-STICC UMR 628
Thèse N° :

Rapporteurs avant soutenance :

Frédéric MERIENNE, Professeur à Arts et Metiers ParisTech
Jean-Pierre JESSEL, Professeur à Université Toulouse 1

Composition du jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition ne comprend que les membres présents

Président :

Examineurs : Sophie DUPUY-CHESSA, Professeur à IMAG / LIG
Cédric BACH, Docteur chez HDG

Dir. de thèse : Ronan QUERREC, Professeur des universités, École Nationale d'Ingénieur de Brest

Encadrant : Sébastien KUBICKI, Maître de conférence, École Nationale d'Ingénieur de Brest



ACKNOWLEDGEMENT

Je tiens à remercier en premier lieu l'institut de recherche b<>com, qui a financé mes stages de Master I et II, puis ma thèse. Je remercie plus particulièrement Guillaume Jégou, qui a cru en moi depuis le début de ma reprise d'étude et m'a accompagné durant ces six longues années. Merci également à Nico Pallamin, Mathieu Quentel, Sonia Em, Anaïs Raison pour leur aide, leur bonne humeur, leur soutien et leur amitié. Merci.

Un grand merci aussi à Pierre Mahieux, mon "jumeau de thèse", qui a supporté mes "dad jokes" depuis le master II, et durant ces trois années de doctorat où nous avons partagé le même bureau, le même encadrant et le même directeur. Sans toi, ces années auraient été bien moins supportables.

Merci également à Sébastien Kubicki, pour qui nous avons été les premiers doctorants avec Pierre, j'espère qu'on ne t'a pas dégoûté de l'encadrement. Tu as su m'apporter ta rigueur (et ce n'était pas gagné) et ton perfectionnisme tout au long de ces années, mais également ton amitié et ton soutien.

Merci à Ronan Querrec, mon directeur de thèse. Tu m'as ouvert des perspectives de développement extraordinaires avec Mascaret. Ton humour et ta bonne humeur n'ont d'égal que ton amour pour le chocolat (et pour ceux qui connaissent, ce n'est pas rien). Tu as guidé mes travaux, à moitié dans l'ombre, à moitié dans la lumière, tel un Batman venant prodiguer son savoir, et disparaissant dans un nuage de fumée.

Et bien sûr, merci à toi, ma chérie. Six ans à supporter mon stress et ma mauvaise humeur. Six ans à faire tous les sacrifices nécessaires à ma réussite, sans jamais te plaindre (ou alors juste un petit peu). Durant ces années, et bien plus de fois que je ne saurais l'admettre, tu as été la bouée qui m'a empêché de sombrer. Sans toi, je n'y serais pas arrivé. Mon amour, merci.

Je souhaite dédicacer ce doctorat à la mémoire de ma sœur Isabelle.
Tu avais raison. Finalement, tout est possible.

SOMMAIRE

| | |
|---|-----------|
| Introduction | 14 |
| 1 État de l'art | 23 |
| 1.1 Introduction | 23 |
| 1.2 Modèles génériques d'interaction | 24 |
| 1.2.1 Paradigmes d'interaction en Réalité Virtuelle | 24 |
| 1.2.2 Modèles abstraits d'interaction | 27 |
| 1.3 Modèles d'activité | 33 |
| 1.3.1 Modèles d'activité en ergonomie | 34 |
| 1.3.2 Modèles d'activité en Réalité Mixte | 37 |
| 1.4 Conclusion | 46 |
| 2 Prisme : Un modèle liant activité métier et interaction en environnement virtuel | 49 |
| 2.1 Proposition d'une typologie d'interacteurs de Réalité Mixte et Tangibles adaptée aux EV | 49 |
| 2.1.1 Les objets tangibles (TangibleInteractionController) | 51 |
| 2.1.2 Les périphériques d'entrées (InputPeripheralController) | 53 |
| 2.1.3 Les composants WIMP (WIMPController) | 55 |
| 2.1.4 Les gestes (GestureController) | 56 |
| 2.1.5 La reconnaissance vocale (SpeechRecognitionController) | 57 |
| 2.2 Modèle d'interaction | 58 |
| 2.3 Conclusion | 63 |
| 3 Utilisation de notre typologie et implémentation de PRISME sur un cas d'usage en aéronautique | 65 |
| 3.1 Vers une plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en Environnement Virtuel | 66 |

| | | |
|----------|---|------------|
| 3.2 | Description des interacteurs constitutifs d'une plateforme à l'aide de PRISME | 70 |
| 3.3 | Définition des interactions dans PRISME | 71 |
| 3.4 | Suivi et création d'une activité à l'aide de PRISME | 78 |
| 3.4.1 | Suivi d'une activité | 79 |
| 3.4.2 | Création d'activité | 82 |
| 3.5 | Conclusion | 83 |
| 4 | Évaluation de PRISME par des ergonomes pour de la conception de plateformes | 85 |
| 4.1 | Méthode / Protocole | 86 |
| 4.1.1 | Population | 86 |
| 4.1.2 | Matériel | 87 |
| 4.1.3 | protocole | 91 |
| 4.2 | Résultats | 94 |
| 4.2.1 | Facilité d'Utilisation Perçue (FUP) de PRISME | 95 |
| 4.2.2 | Utilité Perçue (UP) de PRISME | 99 |
| 4.2.3 | Alignement des résultats TAM sur l'échelle SUS | 103 |
| 4.3 | Discussion | 104 |
| 4.3.1 | Acceptation de PRISME | 104 |
| 4.3.2 | Retours qualitatifs | 105 |
| 4.4 | Conclusion | 106 |
| 5 | Conclusion | 109 |
| 5.1 | Principaux résultats obtenus | 112 |
| 5.2 | Limites de nos travaux et perspectives | 113 |
| A | Liste des interacteurs composant notre plateforme expérimentale | 117 |
| B | Document de présentation de l'expérimentation aux participants | 122 |
| C | Modèle de tâche décrivant l'activité de ravitaillement en vol | 133 |
| D | Questionnaire TAM adapté au modèle PRISME | 136 |

LISTE DES FIGURES

| | | |
|------|--|----|
| 1 | Cycle de vie et évolution des coûts engagés et des coûts d'extraction des défauts d'un système (adapté de Walden <i>et al.</i> [130]). | 14 |
| 2 | Focus sur le cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système). | 15 |
| 3 | Exemple de maquette (ici un cockpit d'avion) à taille réelle | 16 |
| 4 | Maquette de cockpit d'avion de modèle CESSNA | 16 |
| 5 | Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM | 17 |
| 6 | Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM en Environnement Virtuel | 17 |
| 7 | Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM en Environnement Virtuel et des Interfaces Tangibles | 18 |
| 8 | Exemple d'environnement virtuel sans TUI | 18 |
| 9 | Exemple d'environnement virtuel avec TUI | 18 |
| 1.1 | Modèle abstrait d'interaction en EV selon Lacoche [64] | 28 |
| 1.2 | Modèle abstrait de périphériques d'entrée selon Casarin <i>et al.</i> [22] | 29 |
| 1.3 | Modèle d'interaction des TUI : MCRit proposé par Ulmer <i>et al.</i> [118] et Ishii <i>et al.</i> [54] | 30 |
| 1.4 | Modélisation d'une TUI et des boucles de rétroaction associées proposée par Ishii [53] | 31 |
| 1.5 | Premier cycle de rétroaction d'une TUI proposé par Ishii [53] | 32 |
| 1.6 | Second cycle de rétroaction d'une TUI proposé par Ishii [53] | 32 |
| 1.7 | L'instrument, médiateur entre l'utilisateur et l'objet selon Beaudouin-Lafon [10] | 33 |
| 1.8 | Modèle de tâche issu de ConcurTaskTrees [95] | 34 |
| 1.9 | Modèle de tâche issu de K-MAD [8] | 35 |
| 1.10 | Modèle de tâche issu de HAMSTERS selon Barboni <i>et al.</i> [7] | 36 |
| 1.11 | Stéréotypes d'action suivant la notation HAMSTER selon Barboni <i>et al.</i> [7] | 36 |

| | |
|---|----|
| 1.12 Moteur de relations permettant de raisonner sur les actions possibles (#FIVE), selon Bouville <i>et al.</i> [14] | 38 |
| 1.13 #SEVEN, selon Claude [25] | 39 |
| 1.14 Exemple de réseaux de Petri utilisé dans #FIVE et #SEVEN | 39 |
| 1.15 Organisation générale de HUMANS selon Lourdeaux <i>et al.</i> [83] | 41 |
| 1.16 Modèle ACTIVITY-DL selon Lourdeaux <i>et al.</i> [83] | 41 |
| 1.17 Exemple de modèle décrit par WORLD-DL de Lourdeaux <i>et al.</i> [83] | 42 |
| 1.18 Exemple représenté en CAUSALITY-DL selon Lourdeaux <i>et al.</i> [83] | 42 |
| 1.19 Fonctions de création de l'architecture INTERVALS selon Richard <i>et al.</i> [104] | 43 |
| 1.20 Utilisation du framework INTERVALS selon Richard <i>et al.</i> [104] | 44 |
| 1.21 Workflow de conception d'un environnement interactif avec MASCARET | 45 |
| 2.1 Modèle de classes comportant nos cinq catégories issues de notre proposition de typologie d'interacteurs | 50 |
| 2.2 Modélisation des interacteurs de type <code>TangibleInteractionController</code> | 51 |
| 2.3 Vue simplifiée du méta-modèle MASCARET permettant la définition du système. | 52 |
| 2.4 Exemples d'interacteurs de type <code>TangibleInteractionController</code> : A droite, SABLIER, un interacteur tangible pour naviguer à travers l'espace et le temps en EV proposé par Mahieux <i>et al.</i> [86] et à gauche, un exemple d'objet tangible sonore proposé par Antle <i>et al.</i> [4] | 53 |
| 2.5 Modélisation des interacteurs de type <code>InputPeripheralController</code> | 53 |
| 2.6 Exemples d'interacteurs de type <code>InputPeripheralController</code> : a) un dispositif de contrôle radio et b) un joystick X52 | 54 |
| 2.7 Modélisation des interacteurs de type <code>WIMPController</code> | 55 |
| 2.8 Exemples d'interacteurs (composants d'interface) de type <code>WIMPController</code> | 56 |
| 2.9 Modélisation des interacteurs de type <code>GestureController</code> | 56 |
| 2.10 Exemples d'interacteurs de type <code>GestureController</code> dont le périphérique de capture est basé sur un casque de Réalité Augmentée (à droite) et sur un dispositif de capture de mouvements (à gauche) | 57 |
| 2.11 Modélisation des interacteurs de type <code>SpeechRecognitionController</code> | 57 |
| 2.12 Exemple d'interface pouvant utiliser un interacteur de type <code>SpeechRecognitionController</code> dont le périphérique de capture pourrait être ici basé sur le micro d'un casque de Réalité Mixte | 58 |

| | |
|---|----|
| 2.13 Représentation de PRISME avec un focus sur la liaison entre Interaction et Interacteur | 59 |
| 2.14 Représentation de PRISME avec un focus sur les types d'interaction selon Bowman <i>et al.</i> [16] | 59 |
| 2.15 Description d'une activité ravitaillement d'un avion par un diagramme d'activités | 60 |
| 2.16 Vue simplifiée du méta-modèle MASCARET permettant la description d'une activité. | 61 |
| 2.17 Lien entre une Manipulation de notre modèle et une Action de MASCARET | 61 |
| 2.18 Liaison entre une Action et une Activity | 62 |
| 2.19 Exemple de types d'action présents dans la norme UML (en jaune) et deux exemples de type d'action ajouté par MASCARET (en bleu) | 62 |
| 2.20 Intégration des types d'actions dans notre proposition | 63 |
| 2.21 Modèle présentant la liaison entre les interacteurs (<i>Interactor</i> manipulés par l'utilisateur et les actions (<i>Action</i>) issues d'une activité (<i>Activity</i>) | 63 |
| | |
| 3.1 Principe global de notre plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en EV. | 67 |
| 3.2 Notre plateforme physique constituée d'interacteurs tangibles (en haut) et l'EV (ici projeté dans un casque de RV) représentatif d'un cockpit d'avion (en bas) | 68 |
| 3.3 Notre plateforme physique et l'EV (ici projeté dans un casque de RA). Les écrans sont placés en fonction de la position du tracker et le pilote peut contrôler l'avion à l'aide des interacteurs tangibles et de la voix. . . | 69 |
| 3.4 Interface dédiée à la spécification des interactions | 72 |
| 3.5 Interface dédiée à la gestion des profils et à la sélection d'un type d'action | 72 |
| 3.6 Interface dédiée à la gestion des interactions | 73 |
| 3.7 Interface dédiée aux actions de type <i>CallOperationAction</i> | 73 |
| 3.8 Exemple des opérations qu'il est possible de sélectionner lors du choix d'une opération | 74 |
| 3.9 Exemple d'actionneurs compatibles avec l'exécution d'une opération . . | 74 |
| 3.10 Interface dédiée aux actions de type <i>VariableAction</i> | 75 |
| 3.11 Exemple de sélection d'un type de ressource, ressource et attribut pour la modification de la valeur d'un attribut | 75 |

LISTE DES FIGURES

| | |
|---|----|
| 3.12 Exemple de sélection d'une transformation associée à un attribut en cours de modification | 76 |
| 3.13 Exemple de sélection d'une option de transformation | 76 |
| 3.14 Exemple de sélection d'une option de transformation | 77 |
| 3.15 Exemple de description d'interaction permettant de modifier l'attribut <code>A1-titude</code> de la ressource <code>avion</code> grâce à l'actionneur <code>Pitch</code> de l'interacteur <code>JoystickX52</code> | 78 |
| 3.16 Exemple de suivi d'activité | 80 |
| 3.17 Exemples d'interactions enregistrées dans PRISME | 80 |
| 3.18 Exemple de suivi d'activité. À gauche, une activité nouvellement débütée. À droite, la première action a été réalisée. | 82 |
| 3.19 Exemple d'action avec visualisation d'indicateurs clés | 83 |
| 4.1 Plate-forme opérateur composée des quatre interacteurs | 88 |
| 4.2 Interface permettant de lier les interacteurs aux actions issues du modèle d'activité | 88 |
| 4.3 Interface permettant d'enregistrer et de charger un ensemble d'interactions | 89 |
| 4.4 Interface permettant de définir une interaction réalisant une action de type <code>CallOperationAction</code> | 89 |
| 4.5 Interface permettant de définir une interaction réalisant une action de type <code>VariableAction</code> | 89 |
| 4.6 Interface listant les interactions enregistrées dans un profil particulier . . | 90 |
| 4.7 Interface permettant de visualiser le suivi de l'activité et des KPI en temps réel. Ici, l'action <code>ModifierTypeFrequence</code> a été réalisée, l'action <code>ModifierFrequence</code> est en cours de réalisation et l'action <code>ValiderFrequence</code> n'a pas encore débuté. | 91 |
| 4.8 Réponse moyenne et écart-type pour la question n°1 <i>Je pense qu'utiliser PRISME serait facile pour moi</i> | 95 |
| 4.9 Réponse moyenne et écart-type pour la question n°2 <i>Je pense que mon interaction avec PRISME serait claire et compréhensible</i> | 96 |
| 4.10 Réponse moyenne et écart-type pour la question n°3 <i>Je pense qu'il serait facile de devenir habile à utiliser PRISME</i> | 96 |
| 4.11 Réponse moyenne et écart-type pour la question n°4 <i>Je trouverais PRISME flexible pour interagir avec</i> | 96 |

| | |
|--|-----|
| 4.12 Réponse moyenne et écart-type pour la question n°5 <i>Apprendre à utiliser PRISME serait facile pour moi</i> | 97 |
| 4.13 Réponse moyenne et écart-type pour la question n°6 <i>Ce serait facile pour moi d'amener PRISME à faire ce que je veux faire</i> | 97 |
| 4.14 Réponse moyenne et écart-type pour la question n°7 <i>Je pense que ma capacité à déterminer la facilité d'utilisation de PRISME est limitée par mon manque d'expérience</i> | 97 |
| 4.15 Réponse moyenne et écart-type pour la question n°8 <i>Utiliser PRISME dans mon travail me permettrait d'accomplir des tâches plus rapidement</i> | 98 |
| 4.16 Réponse moyenne et écart-type pour la question n°9 <i>L'utilisation de PRISME améliorerait mes performances professionnelles</i> | 98 |
| 4.17 Réponse moyenne et écart-type pour la question n°10 <i>Utiliser PRISME dans mon travail augmenterait ma productivité</i> | 99 |
| 4.18 Réponse moyenne et écart-type pour la question n°11 <i>L'utilisation de PRISME améliorerait mon efficacité au travail</i> | 99 |
| 4.19 Réponse moyenne et écart-type pour la question n°12 <i>Utiliser PRISME faciliterait mon travail</i> | 100 |
| 4.20 Réponse moyenne et écart-type pour la question n°13 <i>Je trouverai PRISME utile dans mon travail</i> | 100 |
| 4.21 Réponse moyenne et écart-type pour la question n°14 <i>Je pense que c'est une bonne idée d'utiliser PRISME</i> | 100 |
| 4.22 Réponse moyenne et écart-type pour la question n°15 <i>J'aime l'idée d'utiliser PRISME</i> | 101 |
| 4.23 Réponse moyenne et écart-type pour la question n°16 <i>Utiliser PRISME est une bonne idée</i> | 101 |
| 4.24 Réponse moyenne et écart-type pour la question n°17 <i>Je prévois d'utiliser PRISME à l'avenir</i> | 101 |
| 4.25 Réponse moyenne et écart-type pour la question n°18 <i>En supposant que j'aie accès à PRISME, j'ai l'intention de l'utiliser</i> | 102 |
| 4.26 Réponse moyenne et écart-type pour la question n°19 <i>Dans mon travail, l'utilisation de PRISME est importante</i> | 102 |
| 4.27 Réponse moyenne et écart-type pour la question n°20 <i>Dans mon métier, l'usage de PRISME est pertinent</i> | 102 |
| 4.28 Formule de calcul de la variable $PRISME_{FUP}$ | 103 |

LISTE DES FIGURES

| | | |
|------|--|-----|
| 4.29 | Formule de calcul de la variable $PRISME_{UP}$ | 103 |
| 4.30 | Formule de calcul de la variable $PRISME_{TAM}$ | 103 |
| 4.31 | Positionnement de PRISME sur l'échelle SUS selon les formules d'alignement entre TAM et SUS proposées par Lewis[81] | 104 |
| 5.1 | Cycle de vie et évolution des coûts engagés et des coûts d'extraction des défauts d'un système (adapté de Walden <i>et al.</i> [130]). | 109 |
| 5.2 | Notre proposition de cycle d'étude et d'évaluation d'un système complexe découpé en 4 étapes | 110 |
| 5.3 | Illustration synthétique de notre typologie d'interacteurs | 111 |
| 5.4 | Positionnement de PRISME sur la représentation de l'échelle SUS | 113 |
| A.1 | Dispositif de contrôle radio | 117 |
| A.2 | Dispositif de contrôle switch | 118 |
| A.3 | Joystick Logitech X52 PRO | 119 |
| A.4 | Manette de gaz Logitech X52 PRO | 120 |
| A.5 | Clavier standard 108 touches | 121 |
| C.1 | Première partie du modèle de tâche décrivant l'activité de ravitaillement en vol | 134 |
| C.2 | Seconde partie du modèle de tâche décrivant l'activité de ravitaillement en vol | 135 |
| D.1 | Première partie du questionnaire TAM adapté au modèle PRISME | 136 |
| D.2 | Seconde partie du questionnaire TAM adapté au modèle PRISME | 137 |

LISTE DES TABLEAUX

| | | |
|-----|--|----|
| 4.1 | Résultats aux questions portant sur la Facilité d'Utilisation Perçue (FUP) | 95 |
| 4.2 | Résultats aux questions portant sur l'Utilité Perçue (UP) | 99 |

INTRODUCTION

Le cycle de vie d'un système complexe (ex. une plate-forme opérateur, un cockpit d'avion ou un tableau de bord de voiture), se décompose selon Walden *et al.* [130] en cinq phases (cf. Figure 1).

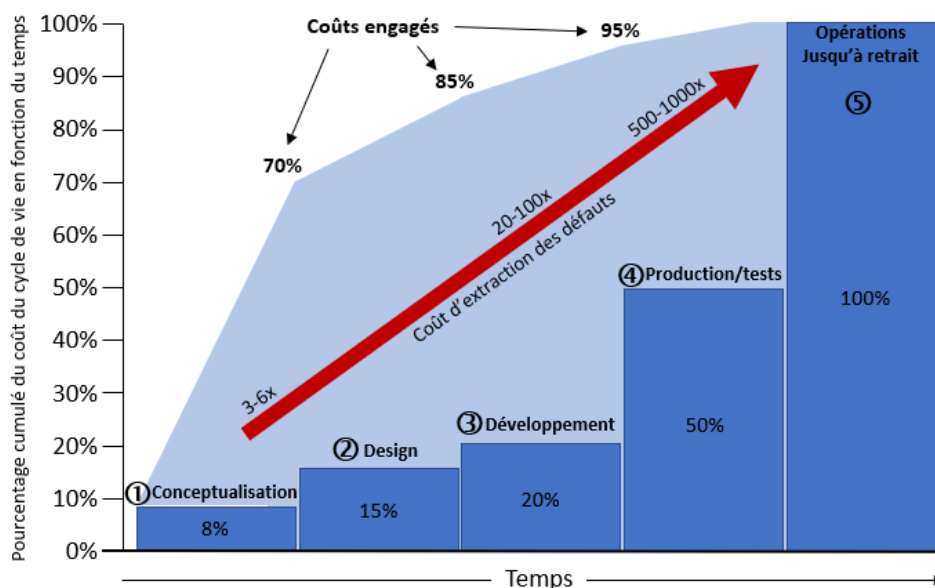


FIGURE 1 – Cycle de vie et évolution des coûts engagés et des coûts d'extraction des défauts d'un système (adapté de Walden *et al.* [130]).

Ces cinq phases couvrent la vie complète du système, de sa conceptualisation jusqu'à son retrait définitif et peuvent être définies ainsi :

1. La **Conceptualisation** consiste, en partant d'une problématique, à définir la solution générale d'un projet, établir son périmètre fonctionnel, identifier ses acteurs, *etc.*
2. Le **Design** consiste à décrire précisément le projet et ses fonctionnalités. C'est dans cette phase que l'on définit l'ergonomie du produit fini, son efficacité et son efficience.
3. Le **Développement**, qui consiste à implémenter les fonctionnalités identifiées dans les phases précédentes.

4. La/Les **Production/tests** consiste(nt) à créer des prototypes à taille réelle afin de les évaluer.
5. La dernière phase de ce cycle de vie comprends l'**exploitation en production** du système jusqu'à son **retrait**.

L'impact financier (coût d'extraction des défauts) engendré par une détection tardive des erreurs augmente au fil des quatre premières étapes du cycle de vie d'un système (flèche rouge sur la Figure 1). Cette dernière illustre l'impact financier (coût d'extraction des défauts) engendré par une détection tardive des erreurs durant les quatre premières phases du cycle de vie d'un système. Par exemple, une erreur identifiée en phase de développement (phase 3) engendrerait une perte allant de 20 jusqu'à 100 fois les coûts investis. Ce facteur d'augmentation pourrait être multiplié par dix durant la phase de production (phase 4).

C'est pourquoi viser à diminuer les coûts engagés des premières étapes, mais aussi optimiser la détection des erreurs de conception, et donc éviter les coûts engendrés par la détection tardive de ces erreurs, reste un objectif majeur du domaine industriel. De plus, c'est en particulier sur les deux premières étapes (70% à étape 1 : Conceptualisation et 85% à étape 2 : Design) que les coûts engagés sont les plus élevés.

C'est dans cet objectif de réduction et d'optimisation des coûts de conception que nos travaux de thèse s'inscrivent et en particulier sur la phase de Design (phase 2).

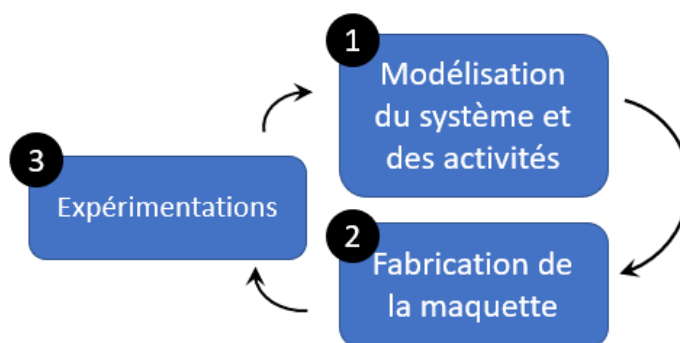


FIGURE 2 – Focus sur le cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système).

En général chaque étape est itérative selon [23] et, plus spécifiquement, dans le cas de la conception (design) de système technique où l'ergonome suit un cycle d'analyse

du problème, de proposition et de validation de la solution. Toujours dans le contexte de la conception de système technique, ce cycle est constitué d'une phase de modélisation du système et des activités, d'une phase de fabrication de maquettes basse fidélité et d'une phase de validation par l'expérimentation. L'ergonome analyse les résultats de l'expérimentation et reprend un nouveau cycle si besoin. Ce cycle est illustré par la Figure 2.

La première étape du cycle de conception d'un système comprend la modélisation du système et des activités associées (étape 1 sur la Figure 2). La modélisation du système est réalisée au moyen d'une description explicite et formelle des éléments qui constituent ce système et des liens qui les relient. Plusieurs langages existent dans la littérature pour décrire un système (UML/SysML, Grafcet, etc.). Nous les étudions dans le chapitre 1 de ce manuscrit. En complément, les activités sont également modélisées à l'aide de langages plus ou moins formels (K-MAD, CTT, UML...) que nous étudions également dans le chapitre 1. Dans nos travaux de thèse, nous avons choisi de nous appuyer sur UML/SysML, car ce langage permet de décrire, dans un même formalisme, l'aspect structurel et dynamique du système (activités) et dispose d'une sémantique précise qui permet de l'opérationnaliser dans un environnement de simulation.



FIGURE 3 – Exemple de maquette (ici un cockpit d'avion) à taille réelle



FIGURE 4 – Maquette de cockpit d'avion de modèle CESSNA

Le modèle produit permet la fabrication de la ou des maquette(s) (étape 2 sur la Figure 2) à taille réelle (*cf.* Figure 3 et 4) sur lesquelles sont réalisées des expérimentations (étape 3 sur la Figure 2). Lors de l'expérimentation, l'ergonome suit la réalisation des procédures par un opérateur sur la maquette. L'ergonome recueille des indicateurs clés (KPI, *Key Performance Indicator* en anglais) tels que le nombre d'erreurs

par rapport à l'activité prescrite et les temps de réalisation.

Si l'évaluation du système conclut à un défaut d'ergonomie, l'ergonome doit être en mesure de modifier le système. Par conséquent, les erreurs identifiées lors des expérimentations peuvent conduire à des modifications du modèle et donc à la construction d'une nouvelle maquette. C'est cette nouvelle itération sur la maquette qui implique une augmentation importante des coûts et des délais.

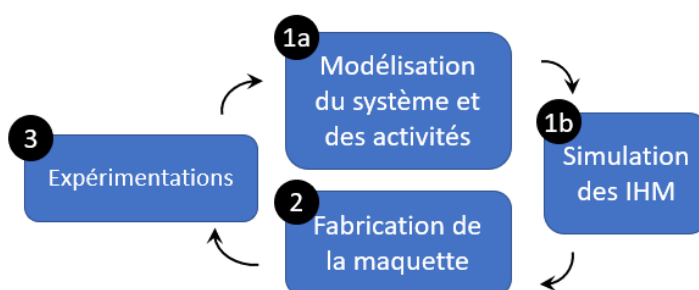


FIGURE 5 – Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM

De plus, la création de maquettes physiques permet une bonne représentation du positionnement des interfaces mais ne permet pas, ou peu, la réalisation ou la simulation des comportements du système. C'est pourquoi l'ajout d'une étape de simulation des Interfaces Humain-Machine (IHM) entre la modélisation et le maquettage (cf. étape 1b sur la Figure 5) permet de tester et simuler l'activité humaine réalisée sur la plate-forme. Cela réduit le risque d'erreur pouvant être rencontré lors de l'étape d'expérimentation des maquettes et, par conséquent, leurs coûts.

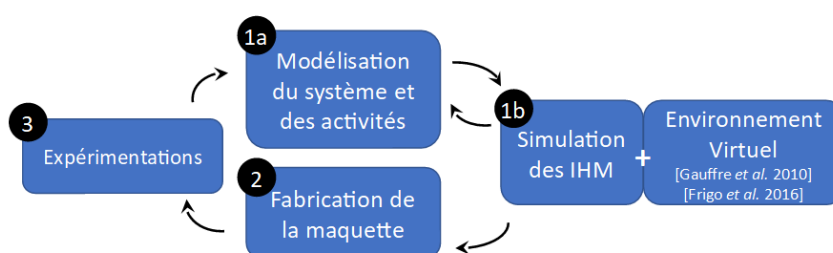


FIGURE 6 – Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM en Environnement Virtuel

Afin de permettre une simulation des systèmes lors des expérimentations (étape

3), nous proposons, dès l'étape de simulation, d'utiliser la Réalité Virtuelle (RV) ou la Réalité Mixte (RM) comme environnement de simulation de ces IHM (cf. Figure 6). En effet, divers travaux, comme ceux présentés par Gauffre *et al.* [45], Frigo *et al.* [44], Dubois *et al.* [33] ou encore Lewkowick *et al.* [82] mettent en avant les avantages de l'utilisation des Environnements Virtuels (EV) au sein d'un processus de conception en tant qu'environnement de simulation. Cette solution permet non seulement un prototypage rapide, mais également une meilleure immersion de l'opérateur dans l'environnement simulé. De plus, les coûts de conception d'un EV, de part sa ré-utilisabilité et sa flexibilité, sont moins importants que ceux qui pourraient être investis dans la fabrication d'un prototype physique.

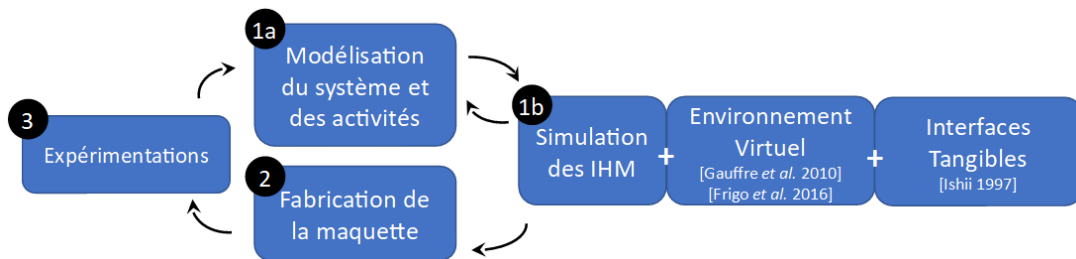


FIGURE 7 – Cycle en 3 étapes de la conception/design (phase 2 du cycle de vie d'un système) avec ajout d'une sous-étape (1b) de simulation des IHM en Environnement Virtuel et des Interfaces Tangibles



FIGURE 8 – Exemple d'environnement virtuel sans TUI



FIGURE 9 – Exemple d'environnement virtuel avec TUI

Cependant, les EV souffrent d'un manque de naturalité des interactions selon Fiebrink *et al.* [39]. En effet, si un EV permet une immersion très réaliste, l'utilisation de

manettes spécifiques ne permet pas d'évaluer l'interaction telle qu'elle devrait-être réalisée sur le système réel (cf. Figure 8). C'est pourquoi, afin d'étendre les possibilités d'interactions (cf. Figure 9) des utilisateurs, nous proposons d'ajouter le concept des interfaces tangibles (TUI, *Tangible User Interface* en anglais) à ces EV (cf. Figure 7).

La difficulté de l'ajout de l'étape de simulation des IHM en EV et des Interfaces Tangibles est que cela implique du développement informatique spécifique qui peut être rendu complexe par la variété des interacteurs proposés (nous définissons comme interacteur tout dispositif permettant d'interagir avec le système, comme une manette). Ceci n'est pas faisable par un ergonomiste et allonge alors les temps de la phase de conception et donc les coûts. C'est pourquoi nous proposons de générer l'EV à partir du modèle produit par l'ergonomiste (structure, comportement du système et activités des opérateurs) ainsi qu'un modèle générique d'interaction en EV qui permet de faire le lien entre l'utilisation des interacteurs et la réalisation des actions composant les activités, sans intervention de code spécifique. Nous émettons donc l'hypothèse que :

Un modèle générique d'interaction permettant de reconnaître automatiquement les actions métiers à partir des interactions de l'utilisateur, permettrait une plus grande agilité lors de la conception des systèmes techniques par les ergonomistes.

De cette hypothèse, nous en déduisons trois questions de recherche :

- **Q1** : Comment décrire de manière générique les différents types d'interacteurs d'EV et Tangibles qui composeront les IHM simulées ?
- **Q2** : Comment décrire la plate-forme et les activités de l'opérateur ?
- **Q3** : Comment reconnaître automatiquement l'action dans l'activité à partir des interactions utilisateur ?

L'objectif de nos travaux est double. D'une part, nous visons à proposer une typologie d'interacteurs permettant à un ergonomiste de décrire la grande majorité des dispositifs utilisés en EV. D'autre part, nous visons à présenter un modèle générique d'interaction permettant à des ergonomistes de réaliser le suivi de l'activité humaine au sein d'un EV, en ayant peu ou pas de développement informatique à prévoir. De plus, nous visons à ce que le modèle permette également d'utiliser l'EV pour créer les ac-

tivités prescrites en amont de l'expérimentation, par un expert du domaine en situation.

Ce manuscrit est composé de cinq chapitres.

Après une introduction générale, nous exposons dans le premier chapitre un état de l'art sur les modèles génériques d'interaction en nous attardant plus spécifiquement sur les paradigmes d'interaction en EV, puis les modèles abstraits d'interaction. Nous revenons sur la notion d'interfaces tangibles et nous explorons comment ces dernières peuvent être représentées et utilisées en EV. Dans un second temps, nous présentons les travaux relatifs aux modèles d'activités selon l'angle de l'ergonomie puis des Environnements Virtuels.

En nous appuyant sur les définitions et travaux présentés dans le premier chapitre, nous présentons en chapitre 2, notre modèle appelé PRISME : un modèle permettant de lier les actions issues d'un modèle d'activité et les interactions d'un utilisateur au sein d'un EV. Nous présentons tout d'abord notre typologie d'interacteur permettant de décrire la grande majorité des interacteurs présents en réalité mixte, décomposés en cinq catégories d'interaction : les périphériques d'entrées, les composants WIMP (interfaces utilisateur graphique), les gestes, la reconnaissance vocale et les objets tangibles. Nous détaillons ensuite comment réaliser la description du système et des activités grâce au méta-modèle MASCARET, puis nous terminons ce chapitre en détaillant comment notre modèle PRISME permet reconnaître l'action réalisée dans l'activité depuis les interactions de l'utilisateur.

Nous présentons dans le chapitre trois un exemple d'implémentation de notre modèle PRISME dans un cas d'usage aéronautique. Nous présentons en début de chapitre la plate-forme opérateur représentant un cockpit d'avion et les interacteurs tangibles qui la composent. Nous présentons ensuite, depuis les interfaces de supervision de PRISME, comment décrire les interactions permettant de reconnaître les actions de l'activité. Pour clore ce chapitre, nous détaillons comment un ergonome réalise le suivi ou la création d'activités en temps réel depuis les interfaces de supervision de PRISME.

Nous présentons dans le chapitre quatre l'évaluation qui nous a permis de mesurer l'acceptation de notre proposition. Cette évaluation place un ergonome en situation d'évaluation de l'ergonomie d'un cockpit d'avion. Il utilise pour cela les outils mis à disposition dans PRISME afin de suivre l'activité d'un opérateur réalisant une procédure de ravitaillement en vol. L'ergonome a suivi en temps réel le déroulement de l'activité,

puis a redéfini deux interactions afin de constater l'impact de ces modifications sur l'efficacité et l'efficacité de l'opérateur. Nous présentons par la suite les résultats obtenus et nous concluons ce chapitre par une analyse de ces résultats.

Pour terminer, nous présentons nos conclusions, et nous présentons les perspectives et limites de notre modèle.

ÉTAT DE L'ART

1.1 Introduction

Dans le cadre de cette thèse, notre objectif est de concevoir un modèle générique d'interaction qui permet à des ergonomes de concevoir un Environnement Virtuel (EV) sans devoir faire intervenir un informaticien pour créer le lien entre les interacteurs (rappelons que nous définissons un interacteur comme tout dispositif permettant d'interagir avec un système comme, par exemple, une manette ou un objet physique) et les actions métiers. Il s'agit de proposer aux ergonomes un moyen de spécifier, par exemple, que l'appuie sur une touche d'une manette associée à un casque de Réalité Virtuelle (RV) provoque le début de l'exécution de l'ouverture d'une vanne au sein d'un EV. Ceci permettrait à l'ergonome de suivre en temps réel l'évolution de la procédure métier sur le système en cours de conception, voire de créer / modifier / adapter la procédure.

Afin de proposer ce modèle générique d'interaction, il nous faut tout d'abord une typologie d'interacteurs permettant d'instancier tout type de périphérique de Réalité Augmentée (RA), de Réalité Virtuelle (RV) ou plus globalement de Réalité Mixte (RM). Plusieurs modèles génériques ou typologies ont déjà été proposés dans la littérature. Nous les présentons en détail dans la section 1.2. Cependant, ces typologies ne prennent pas en compte la manipulation d'objets physiques (et donc les interacteurs tangibles) qui offrent selon Fiebrink *et al.* [39], dans la plupart des cas, des interactions plus naturelles selon la tâche à exécuter en EV qu'une manette par exemple. Nous présentons dans ce chapitre les propriétés de ce type d'interacteur et les modèles permettant leur intégration dans un Environnement Numérique ou Virtuel.

Par ailleurs, les typologies présentées dans ce chapitre ne permettent pas de décrire de manière générique, l'impact de l'utilisation du périphérique sur le système simulé. Il n'est, par exemple, pas possible de faire le lien avec les activités et les actions métiers (l'appui d'un bouton lance l'action d'ouverture d'une vanne). Pour cela, un

modèle de description de ces actions et activités doit être utilisé. Aujourd'hui, plusieurs modèles existent et nous les présentons dans la section 1.3 de ce chapitre. Notons que certains de ces modèles sont issus du domaine de la conception en ergonomie. Ces modèles permettent une bonne description des actions et de leurs enchaînements au sein d'une activité. Ils permettent également de spécifier des propriétés profitables pour les ergonomes (ex : temps, type...). Toutefois, la plupart de ces modèles ne sont pas suffisamment précis sur les effets des actions sur le système afin de permettre immédiatement une exécution en EV. C'est pourquoi nous présentons également dans ce chapitre les modèles permettant de représenter les activités en EV.

1.2 Modèles génériques d'interaction

Dans cette section, nous présentons tout d'abord des paradigmes d'interaction utilisés tout particulièrement en RV, puis des modèles abstraits d'interaction en RM permettant la réalisation de ces paradigmes d'interaction.

1.2.1 Paradigmes d'interaction en Réalité Virtuelle

Les travaux fondateurs sur les typologies d'interactions en RV sont ceux de Bowman *et al.* [16]. Ces travaux mettent l'accent sur la méthode à adopter afin de constituer un EV interactif. L'idée majeure de ces travaux est qu'une distinction doit être faite entre les périphériques et les techniques d'interaction, les périphériques n'étant que les outils permettant de mettre en œuvre les interactions selon leurs degrés de liberté.

Trois grandes classes d'interaction en EV ont été identifiées par Bowman :

1. La navigation

Elle correspond à l'action de l'utilisateur la plus courante dans un EV. Elle consiste à mettre en place des méthodes de déplacement au sein de l'environnement, et est généralement divisée en trois catégories :

- **L'exploration** : l'utilisateur navigue sans autre but que la découverte de l'environnement
- **La recherche** : l'utilisateur se déplace vers un point défini
- **La manœuvre** : l'utilisateur réalise une action de précision de courte portée afin de changer son point de vue (pour réaliser une tâche par exemple)

Quelle que soit la catégorie ou l'objectif de l'utilisateur, la navigation dans un EV peut être réalisée de cinq manières différentes :

- (a) *Le déplacement physique* : l'utilisateur se déplace dans l'espace délimité de l'EV
- (b) *La manipulation manuelle de points de vue* : l'utilisateur peut par exemple "attraper l'air" avec sa main et le ramener à lui pour provoquer le déplacement
- (c) *Le pilotage* : l'utilisateur donne une direction (par le regard ou par un contrôleur) et une impulsion de déplacement
- (d) *La sélection d'un point d'arrivée* : l'utilisateur désigne un point d'arrivée et le système génère le déplacement par transition ou téléportation
- (e) *La planification* : l'utilisateur prévoit un chemin spécifique pour son déplacement, qui sera délégué au système. Cela permet à l'utilisateur de pouvoir réaliser des actions librement durant ce déplacement

Toujours selon Bowman, la navigation doit prendre en charge la conscience spatiale, c'est-à-dire la connaissance et/ou la représentation de l'environnement réel entourant l'utilisateur, et qui n'est pas toujours représenté dans un EV (ex : une grille apparaît lorsque l'utilisateur s'approche d'un mur réel). La navigation doit également être simple à réaliser afin de permettre à l'utilisateur de se concentrer sur des tâches plus importantes, et se sentir plus efficace et confortable, afin d'éviter l'apparition possible de cinétose [3].

2. La sélection et la manipulation

Ces types d'interaction permettent de réaliser au moins une de ces actions :

- La sélection d'un objet : L'utilisateur choisit l'objet avec lequel il va interagir
- Le positionnement d'un objet : l'utilisateur déplace un objet, par exemple poser un objet sur un autre
- La rotation d'un objet : l'utilisateur applique une rotation à un objet afin, par exemple, de tourner un objet face à lui

La sélection et la manipulation impactent grandement la qualité de l'application virtuelle. En effet, ces interactions sont très proches des interactions qu'un utilisateur opère dans le monde réel. Une des méthodes classiques pour réaliser ces

types d'interaction est de proposer une approche visant à proposer à l'utilisateur une "main virtuelle" dont les mouvements reflètent les mouvements de la main de l'utilisateur. Une autre approche largement utilisée pour sélectionner un objet est également d'utiliser un rayon (de type laser) dirigé par l'utilisateur, pointant sur l'objet à sélectionner.

3. Le contrôle du système

Ces interactions consistent à appliquer une commande qui modifiera l'état du système. Elles sont particulièrement adaptées pour les environnements WIMP (Windows, Icons, Menus, Pointers) et restent délicates à mettre en place dans un EV. Toujours selon Bowman *et al.* [16], la difficulté n'étant pas le passage à la 3D, mais la différence d'efficacité des dispositifs de pointage.

Il existe quatre groupes de type de contrôles :

- *Les menus graphiques* : ce sont des représentations visuelles des commandes (Menus "Fichier", "Editer", *etc.* que l'on retrouve dans la grande majorité des applications WIMP)
- *Les commandes vocales* : les commandes sont accessibles par la voix ("Valider fréquence", "Ouvrir vanne 3", *etc.*)
- *Les commandes gestuelles* : les commandes sont accessibles par des gestes prédéfinis (écartier les doigts pour agrandir une image, glisser un doigt pour faire défiler un texte, *etc.*)
- *Les « props »* : des objets du monde réel que l'on intègre dans l'EV et qui ont donc une utilisation naturelle et inée pour l'utilisateur (un clavier, un crayon, une bouteille, *etc.*)

Bowman propose également des recommandations pour le choix des interactions lors de la conception d'un EV :

- Envisager des interactions "magiques" plutôt que "naturelles" lorsque les tâches exigent de la productivité et de l'efficacité. L'interaction naturelle devrait être utilisée lorsqu'une réplique du monde physique est importante (par exemple modifier la couleur d'un mur en le pointant plutôt qu'en le peignant réellement)
- Choisir des techniques d'interaction basées sur les exigences de l'application. Le même ensemble de techniques ne fonctionnera pas correctement dans toutes les situations. Par exemple, une interaction permettant d'ouvrir une porte battante ne fonctionnera pas correctement pour ouvrir un tiroir coulissant.

- Limiter autant que possible les degrés de liberté et définir des contraintes physiques ou virtuelles pour guider l'utilisateur.
- Tirer parti de la complexité offerte par la diversité des éléments d'interaction (ex. mouvements du corps entier, interactions multimodales, nouveaux dispositifs, *etc.*), et donner à l'utilisateur une structure et un soutien pour gérer cette complexité accrue.
- Proposer une information en retour (feedback) afin de notifier à l'utilisateur que l'action réalisée s'est correctement déroulée, ou non.

Raimbaud *et al.* [103], qui s'appuient sur les travaux de Bowman, proposent de décomposer la tâche utilisateur en EVI (par exemple : "changer une roue de voiture") en tâches primitives afin de supprimer leur caractère spécifique et ainsi les rendre génériques (par exemple : "se déplacer à un endroit spécifique", "orienter un objet", *etc.*).

1.2.2 Modèles abstraits d'interaction

La plupart des modèles abstraits d'interaction en RV proposés dans la littérature s'appuient sur les travaux de Bowman *et al.* [16]. Dans leurs travaux, les auteurs présentent les périphériques d'interaction (*e.g.*, contrôleurs, casque, *etc.*) comme des outils permettant de mettre en œuvre des interactions selon leurs degrés de liberté. Ainsi, les interactions décrites dépendent des possibilités d'interactions offertes par ces interacteurs. Nous présentons dans cette section quelques modèles abstraits fondateurs ou récents. Nous faisons également un focus sur les modèles liés aux interactions tangibles.

Le Modèle VRPN : (*Virtual Reality Peripheral Network*)

Le modèle VRPN (Virtual Reality Peripheral Network) proposé par Taylor *et al.* [115] et son extension proposé par Steindecker *et al.* [112] proposent un modèle abstrait dans lequel tout périphérique d'interaction peut être représenté par un ensemble d'axes et/ou de boutons. Plus récemment, Lacoche [64] a proposé un modèle plus complet qui ajoute les nouvelles technologies de RM et qui s'appuie toujours sur la notion d'axes / boutons. Un extrait de ce modèle est présenté sur la Figure 1.1.

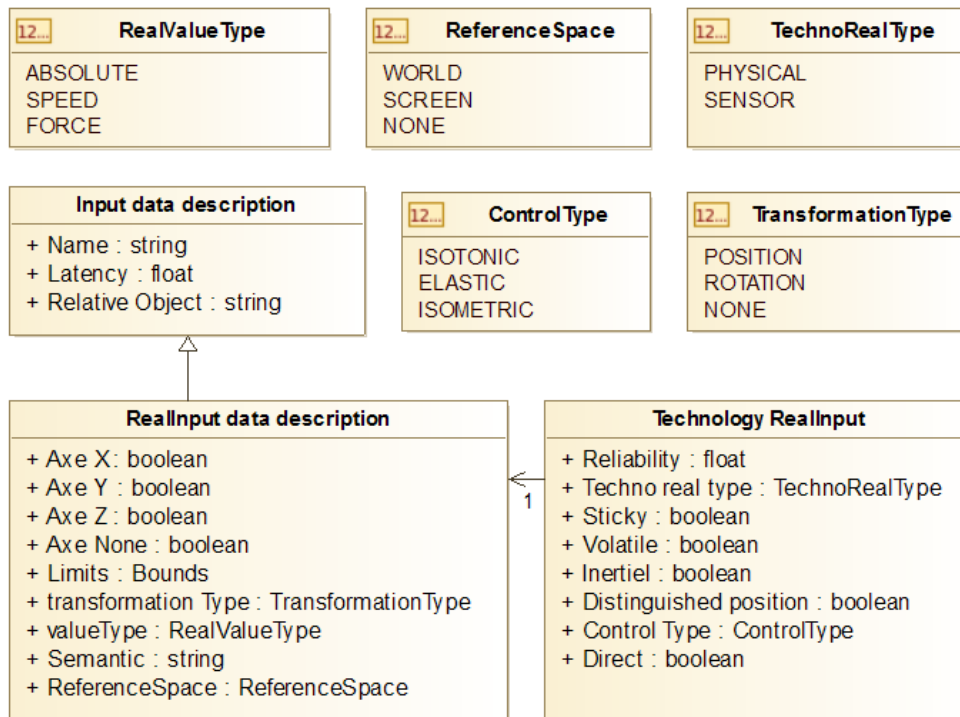
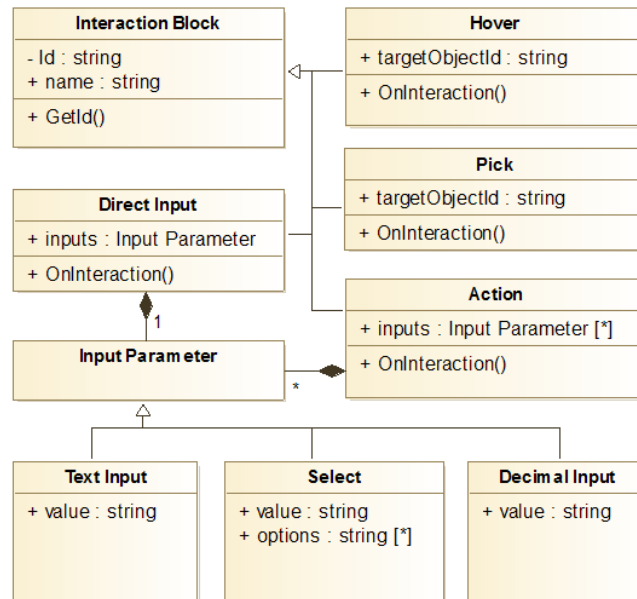


FIGURE 1.1 – Modèle abstrait d'interaction en EV selon Lacoche [64]

Dans ce modèle, une interaction est décrite au travers de deux classes. La classe *RealInputDataDescription* représente les valeurs, propriétés et limitations de l'interaction. La classe *TechnologyRealInput* décrit comment les données sont acquises depuis le périphérique d'entrée lors de cette interaction. Les effets de l'interaction sur l'environnement est uniquement représentée par le champ *Semantic* de la classe *RealInputDataDescription*. Ce champ est une chaîne de caractères et est donc très faiblement formalisé et ne permet donc pas de faire un lien avec les actions des activités simulées.

Casarin *et al.* [22] vont plus loin dans la formalisation de l'effet de l'interaction sur l'EV et proposent un modèle abstrait permettant de faire le lien entre les interacteurs et les actions de bas niveau de l'utilisateur dans une application (voir Figure 1.2). Ce modèle propose de représenter les interactions en RV selon deux niveaux d'abstraction.

Le premier niveau d'abstraction se situe entre les environnements virtuels et les périphériques d'interaction représentés par la classe (*Interaction Block*). Cette classe permet de concevoir des interactions indépendantes de la nature des périphériques et donc facilite la portabilité des EV. Ainsi, un EV conçu pour un dispositif en particu-

FIGURE 1.2 – Modèle abstrait de périphériques d'entrée selon Casarin *et al.* [22]

lier pourra, grâce à cette abstraction, être porté pour un dispositif différent. Le second niveau d'abstraction permet de faire le lien entre les interactions et leurs effets sur l'EV par les classes (*Action*, *Pick*, *ect.*). Ces classes s'appuient sur les paradigmes d'interaction proposés par Bowman *et al.* [16]. L'utilisateur, en interagissant avec le périphérique, active des blocs d'interaction et les actions associées sont exécutées.

Cependant, les actions proposées (*e.g.* : *Hover*, *Pick*, *TextInput*, *etc.*) sont certes d'un niveau d'abstraction supérieur à celui des périphériques (*InteractionBlock*) mais ne permettent pas de faire le lien automatique (sans codage) avec les actions (*e.g.*, changer la fréquence radio) décrites dans un modèle d'activité (voir section 1.3). De plus, les modèles d'interaction en RV présentés ici ne tiennent pas compte des interactions tangibles pourtant de plus en plus utilisées en EV. C'est pourquoi nous proposons de les intégrer dans notre contribution. Nous en présentons le modèle de référence dans la section suivante.

Le Modèle MCRit : *Model-Controller-Representations intangible and tangible*

Le concept d'interface tangible TUI (*Tangible User Interface*, en anglais) a été proposé par Ishii *et al.* [58] en s'inspirant des travaux de Fitzmaurice *et al.* [40]. Ce concept désigne les interfaces permettant de manipuler un contenu numérique via des objets physiques.

Dans le domaine des TUI, un des modèles d'interaction de référence est le modèle MCRit (Model-Controller-Representations intangible and tangible) proposé en 2002 par Ullmer *et al.* [118] puis repris par en 2008 par Ishii *et al.* [54]. Ce modèle peut être considéré comme une analogie au modèle MVC (Modèle-Vue-Contrôleur) propre aux interfaces graphiques (GUI - Graphical User Interface).

Cette approche se distingue du modèle MVC par deux principes. Le premier principe est que le composant "Vue" du modèle est scindé en deux : une partie intangible qui s'implémente dans l'environnement numérique, mais qui peut avoir un impact également sur le monde physique (projection, son, *etc.*) et une partie tangible qui est exclusivement dans le monde physique. Le second principe est que la représentation tangible (appartenant au composant "Vue") est fortement lié au composant "Contrôleur".

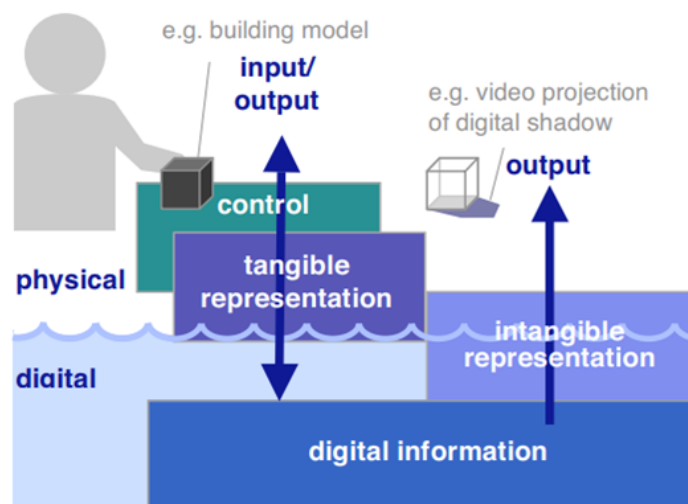


FIGURE 1.3 – Modèle d'interaction des TUI : MCRit proposé par Ulmer *et al.* [118] et Ishii *et al.* [54]

De cette proposition, les auteurs fournissent quatre propriétés principales permettant de caractériser les interfaces tangibles :

- Les représentations tangibles doivent être couplées à l'information numérique sous-jacente
- Les représentations tangibles incarnent les mécanismes de contrôle interactif
- Les représentations tangibles sont couplées avec les représentations numériques

- L'état physique des artefacts d'interface incarne partiellement l'état numérique du système

En 2008 Hiroshi Ishii [53], modélise les interactions tangibles selon trois boucles de rétroaction (cf. Figure 1.4) :

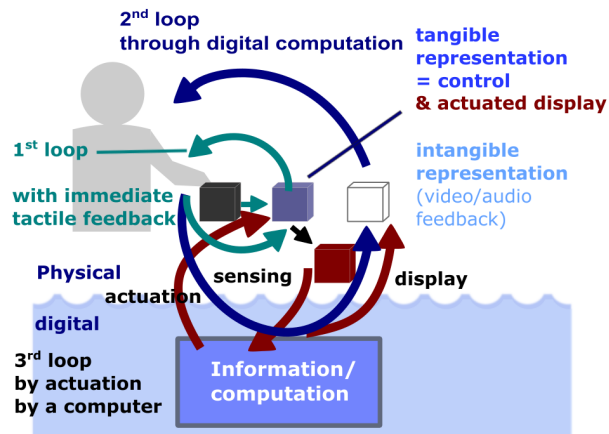


FIGURE 1.4 – Modélisation d'une TUI et des boucles de rétroaction associées proposée par Ishii [53]

1. La première est une boucle de rétroaction haptique passive qui fournit à l'utilisateur une confirmation immédiate qu'il a saisi l'objet physique ou modifié sa position (voir Figure 1.5). Cette boucle existant de manière intrinsèque dans le monde physique, ne nécessite aucune détection ou traitement numérique. L'utilisateur peut manipuler l'objet comme il le souhaite sans avoir à attendre la deuxième boucle de rétroaction et la confirmation visuelle de l'interface (contrairement à l'utilisation d'une GUI (Graphical User Interface) où l'utilisateur doit attendre un retour visuel (2ème boucle) pour terminer une action).

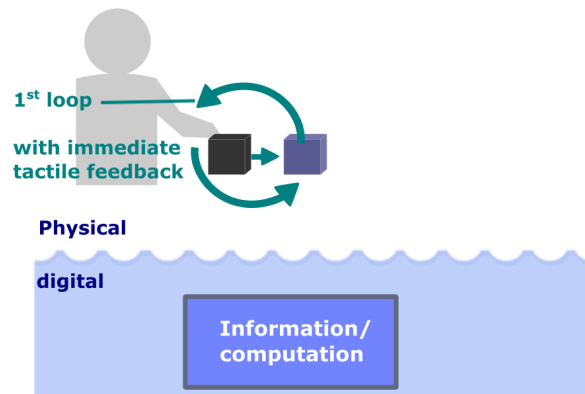


FIGURE 1.5 – Premier cycle de rétroaction d'une TUI proposé par Ishii [53]

2. La seconde boucle est une boucle de rétroaction numérique qui nécessite de détecter le déplacement d'objets physiques par les utilisateurs (voir Figure 1.6). Elle consiste à interpréter numériquement les données détectées et à en afficher les résultats sous forme de rétroaction visuelle (voire auditive). À la différence de la première boucle, celle-ci nécessite un traitement numérique et prend donc plus de temps que la première.

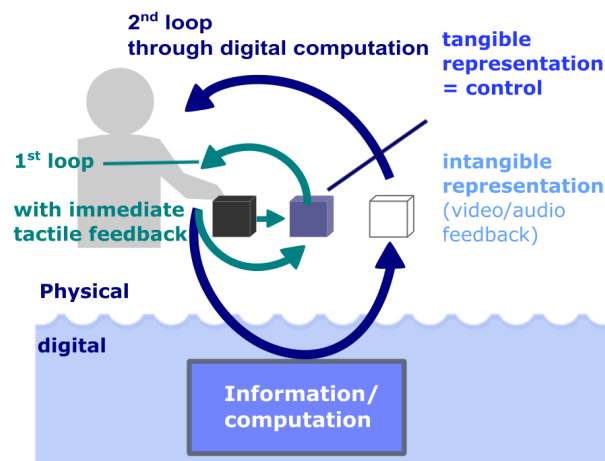


FIGURE 1.6 – Second cycle de rétroaction d'une TUI proposé par Ishii [53]

3. La troisième boucle permet au système de donner un retour sur l'état de l'information numérique lorsque le système modifie lui-même le positionnement d'un objet physique (voir Figure 1.4).

D'une manière générale, ces trois boucles de rétroactions peuvent être comparées aux trois niveaux de retour d'information de l'interaction instrumentale proposés par Beaudouin-Lafon en 1997[10] (voir figure 1.7).

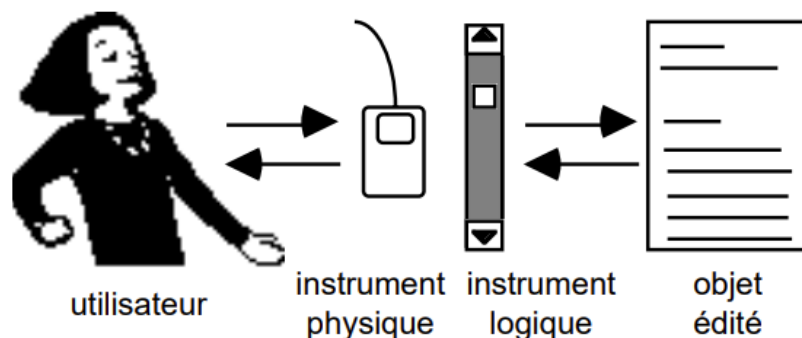


FIGURE 1.7 – L'instrument, médiateur entre l'utilisateur et l'objet selon Beaudouin-Lafon [10]

Le premier niveau correspond au retour de l'action sur l'instrument physique, le second au retour visuel via la représentation de l'instrument logique et le troisième au retour de l'effet de l'interaction sur l'objet édité.

Dans l'exemple de la figure 1.7, l'utilisateur fait défiler un ascenseur logique avec une souris. Le retour d'information sur la souris est proprioceptif (position physique de la souris). Le retour d'information de l'instrument est fourni par la position de la barre de défilement et le retour de l'effet de l'interaction sur l'objet édité est le défilement du document.

À partir du modèle MCRit, nous proposons d'intégrer les notions de représentation tangible ("rep-t") et de représentation intangible ("rep-i") dans notre modèle générique d'interaction (présenté chapitre 2). Ceci permet de considérer les objets tangibles au même titre que tous les autres interacteurs.

Nous proposons de détailler, dans la suite, la partie "modèle" issue du modèle MVC (Modèle, Vue, Contrôleur) et conservée dans MCRit, afin de lier ces interactions aux modèles d'activité.

1.3 Modèles d'activité

Nous avons présenté dans la section précédente différents modèles permettant la description des interactions que l'utilisateur peut effectuer lors de la simulation d'un

système en EV. Afin de faire le lien entre ces interactions et les actions métiers, nous présentons dans cette section différents modèles de description des actions et activités. Ils permettent une description détaillée des actions et de leurs enchaînements.

1.3.1 Modèles d'activité en ergonomie

Dans le contexte de la modélisation de plate-forme, les ergonomes utilisent des modèles de représentation des tâches (ou activités). Les modèles les plus connus sont CTT (ConcurTaskTrees) proposé par Paternò *et al.* [95], K-MAD (Kernel of Model for Activity Description) proposé par Baron *et al.* [8] et HAMSTERS, proposé par Barboni *et al.* [7]. Tous ces modèles sont des modèles de représentation graphique et hiérarchique, c'est-à-dire qu'une activité est composée de sous-activités et d'actions. Ces modèles définissent des opérateurs (*e.g.*, avant, pendant, itération, parallèle, *etc.*) pour agencer les activités ou les actions entre elles. Ces modèles partagent également la notion de rôles et de ressources qui réalisent ou servent de support aux activités. Nous détaillons ici ces trois principaux modèles.

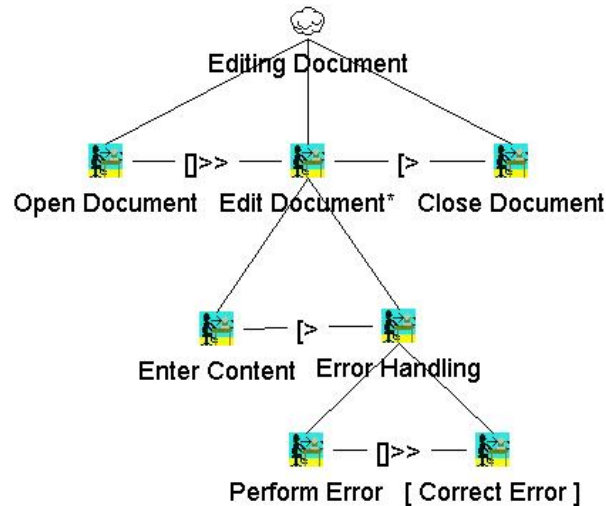


FIGURE 1.8 – Modèle de tâche issu de ConcurTaskTrees [95]

CTT [95] permet aux ergonomes de décrire un système en se focalisant sur les activités des utilisateurs. Dans CTT, une action peut être abstraite, automatique (réalisée par le système), interactive (à l'initiative de l'utilisateur avec le système) ou exclusivement réalisée par l'utilisateur (sans le système). Dans ce modèle, une activité peut

être collaborative et les actions sont organisées entre elles par des opérateurs temporels tels que : entrelacement (les activités peuvent être réalisées dans n'importe quel ordre), activation (la fin d'une activité rend possible la réalisation d'une autre), itération, optionnel, *etc.*. Un exemple de description d'activité utilisant CTT est présenté sur la figure 1.8.

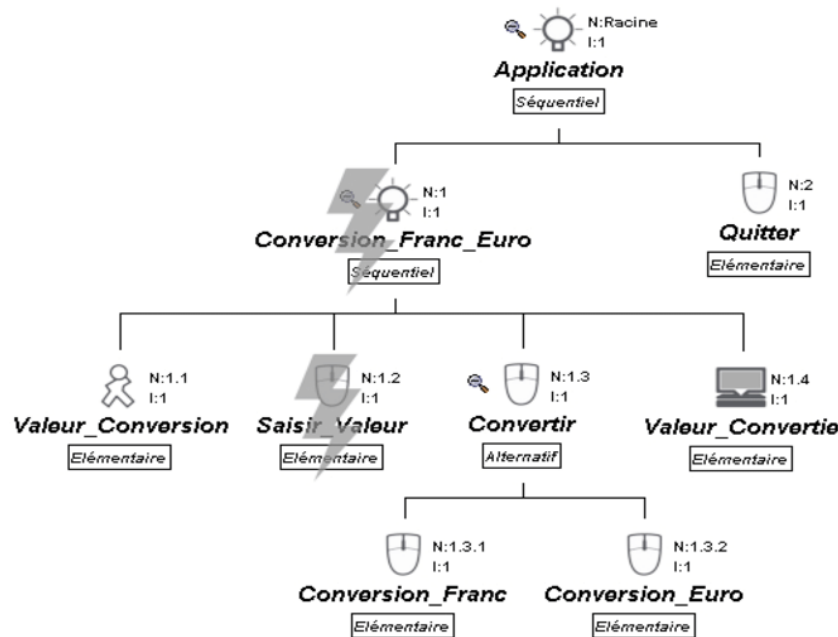


FIGURE 1.9 – Modèle de tâche issu de K-MAD [8]

Comparativement à CTT, K-MAD selon Baron *et al.* [8] raffine le concept d'action. En effet, une action est représentée par sa durée estimée, son but, des observables (effets de l'action sur le système), sa fréquence d'occurrence, son importance dans la réalisation de la tâche globale, les ressources du système utilisées pour la réalisation de l'action, son exécutant (système, utilisateur ou organisation) et sa modalité (sensori-motrice ou cognitive). Les actions sont agencées entre elles par des opérateurs comparables à ceux de CTT. Un exemple de description d'activité utilisant K-MAD est présenté sur la figure 1.9.

La figure 1.9 illustre un ensemble d'actions agencées par des opérateurs. Les opérateurs (encadrés sous l'action), définissent l'agencement d'exécution des actions qui la composent. Par exemple, l'action `Conversion_Franc_Euro`, qui possède l'opérateur `Sequentiel`, est composée des actions `Valeur_Conversion`, `Saisir_Valeur`, `Convertir` et `Valeur_Convertie`. Ces quatre actions seront exécutées de manière séquen-

tielle et devront être réalisées afin de réaliser l'action mère *Conversion_Franc_Euro*. Dans cet exemple, deux autres opérateurs sont présents. L'opérateur *Elémentaire* signifie que l'action ne contient pas de sous-actions et l'opérateur *Alternatif* spécifie que seulement l'une des sous-actions doit être réalisée afin de réaliser l'action mère.



FIGURE 1.10 – Modèle de tâche issu de HAMSTERS selon Barboni *et al.* [7]

HAMSTERS, proposé par Barboni *et al.* [7] s'appuie également sur un langage graphique qui permet de décrire les activités de manière hiérarchique comme CTT et K-MAD. La formalisation des activités est proche de celle de K-MAD et les opérateurs d'agencement sont sensiblement les mêmes, comme nous le montre la figure 1.10.

| | Abstract | Input | Output | I/O | Processing |
|-------------|--------------------------|-------------------|-------------------|-------------------|-------------------|
| Abstract | Abstract | Not Applicable | Not Applicable | Not Applicable | Not Applicable |
| User | User abstract | Perceptive | Motor | User | Cognitive |
| Interactive | Abstract interactive | Input | Output | Input/Output | Not Applicable |
| System | Abstract system | Output | Input | Input/Output | System |

FIGURE 1.11 – Stéréotypes d'action suivant la notation HAMSTER selon Barboni *et al.* [7]

Dans HAMSTERS, il est possible de mieux spécifier les ressources dont l'utilisateur a besoin pour réaliser la tâche. Ces ressources portent sur des objets ou sur des connaissances. Une autre spécificité de HAMSTERS est de permettre de stéréotyper les activités (ou les actions). Elles peuvent être de type *perceptive* (recueil d'informations par l'utilisateur), *motrice* (*Motor* en anglais, réalisation d'une action par l'utilisateur), *cognitive* (raisonnement, mémorisation...) ou *utilisateur* (*perceptive*, *motrice* et *cognitive*) (voir la ligne *User* du tableau des stéréotypes dans la figure 1.11).

Ces modèles de représentation des tâches sont centrés sur la description de l'activité et non sur le système. Si cette approche semble pertinente pour les ergonomes, elle empêche toutefois une simulation précise des effets des activités sur le système (e.g., changement d'état, de valeur de propriété).

Notre hypothèse est que l'intégration d'une phase de simulation en RV lors de la conception d'un système technique pourra raccourcir les délais et réduire les coûts. Nous présentons dans la suite les formalismes de représentation du système et des activités permettant une exécution dans les environnements virtuels.

1.3.2 Modèles d'activité en Réalité Mixte

#FIVE (Framework for Interaction in Virtual Environments) est un modèle permettant de décrire et de raisonner sur des objets pouvant participer à la réalisation d'une action. Il est inspiré des travaux de Luna *et al.* [85] permettant la modélisation des interactions collaboratives. #FIVE se concentre particulièrement sur les concepts de *Relation*, d'*Interaction* inter-objets et de *Réalisation*.

Un objet #FIVE est une entité de l'EV ne possédant pas nécessairement de représentation virtuelle. Cet objet est défini par un *Patron* d'objet qui décrit un ensemble de types (par exemple, un scalpel peut être représenté par les types *Instrument* et *Tranchant*). La notion de *Relation*, qui modélise une action, permet de définir les contraintes appliquées aux types. La notion d'*interactions*, définie selon les concepts d'attributs contrôlables et d'interacteurs, permet quant à elle de lier un ou plusieurs attributs de différents objets par des fonctions. Enfin, une *Réalisation*, qui est une instance de *Relation*, s'assure que les objets sont dans un état permettant la réalisation de l'action. Une illustration de la description d'objets et de leurs actions associées sous #FIVE est représentée en Figure 1.12.

#SEVEN, proposé par Claude [25] (voir Figure 1.13), est un moteur de scénarii in-

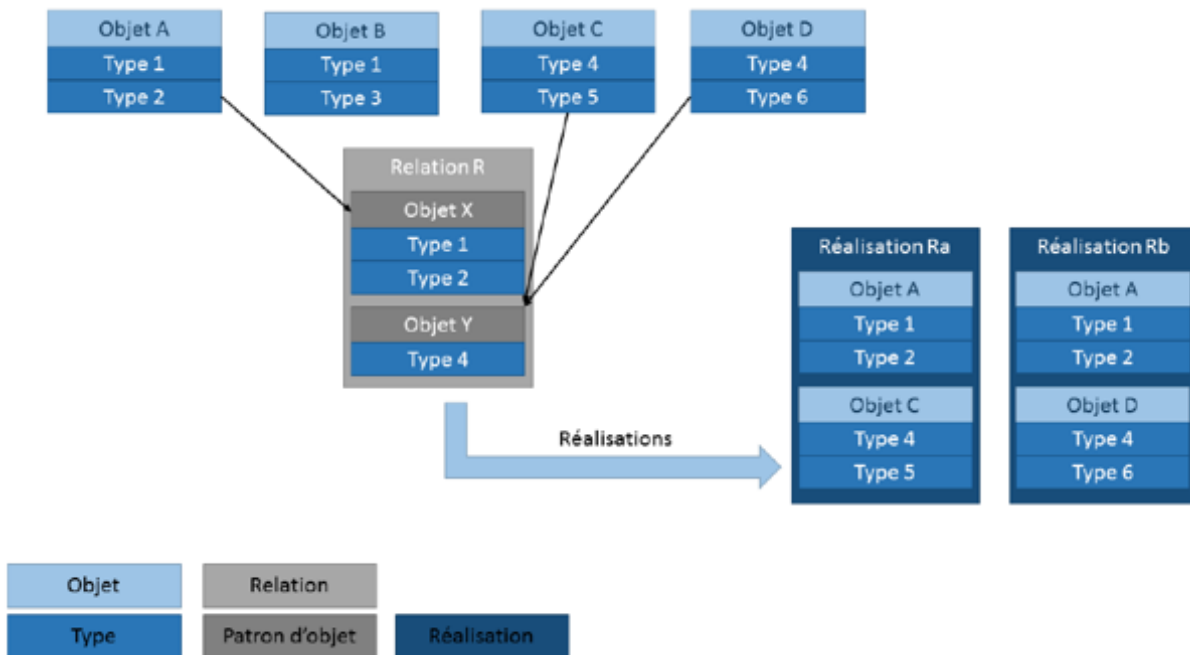


FIGURE 1.12 – Moteur de relations permettant de raisonner sur les actions possibles (#FIVE), selon Bouville *et al.* [14]

tégré dans un EV. Les activités au sein de ces scenarii sont décrites avec des réseaux de Pétri (voir Figure 1.14). Le moteur échange avec l'environnement au moyen d'interfaces d'entrées et de sorties. Les interfaces d'entrées renseignent le moteur sur l'état de l'environnement, tandis que les sorties permettent au moteur d'agir sur celui-ci.

La représentation interne de #SEVEN fonctionne suivant la boucle de perception (ou boucle PDA - perception, décision, action) définie par Mallot [87]. Afin de réaliser le fonctionnement de la boucle PDA, #SEVEN utilise trois classes d'entité. La partie **Perception** de la boucle est assurée par la classe d'entité *Senseur* (en gris foncé sur la partie basse de la figure) qui vérifie une condition sur l'état d'un élément de l'environnement. La partie **Décision** est quant à elle réalisée par la classe d'entité *Séquencement* (en bleu sur la partie basse de la figure) qui prendra les décisions d'actions selon les informations perçues par les senseurs. Enfin, la partie **Action** de la boucle est réalisée par la classe d'entité *Effecteur* (en gris clair sur la partie basse de la figure) qui déclenchera les comportements dans l'environnement.

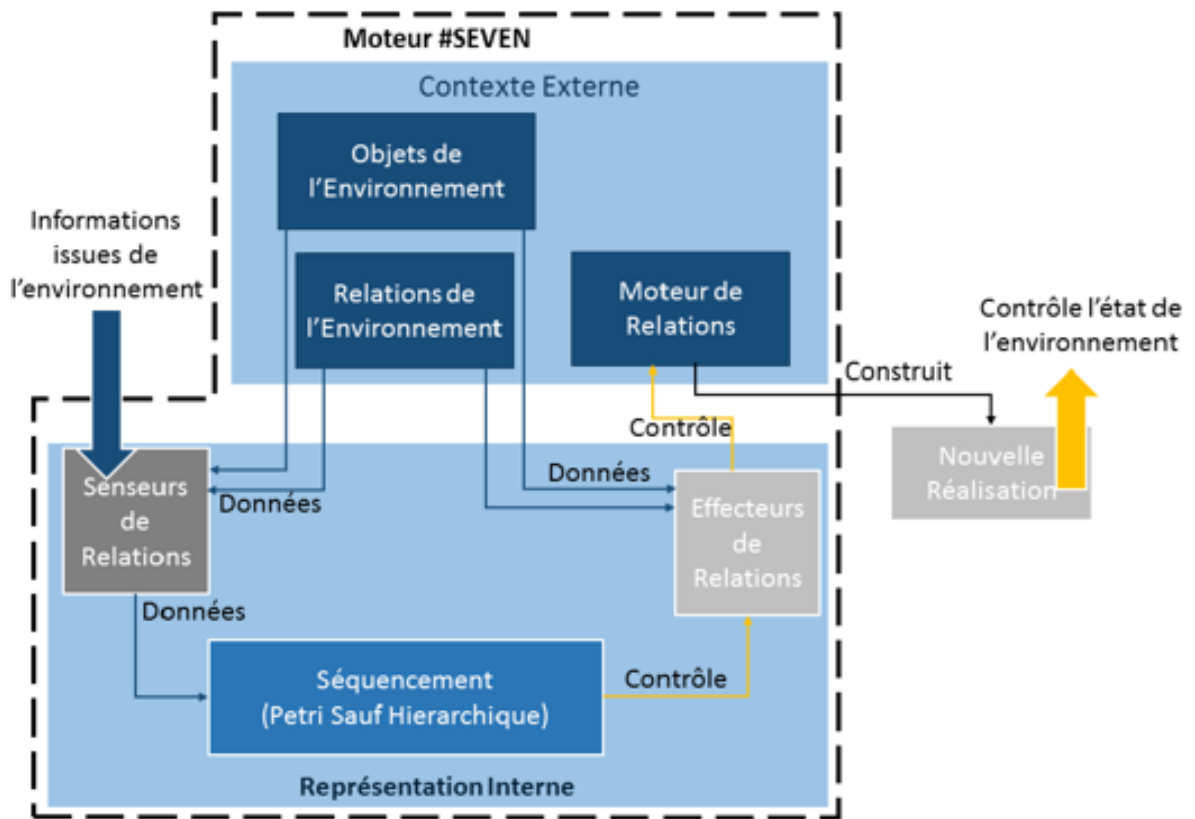


FIGURE 1.13 – #SEVEN, selon Claude [25]

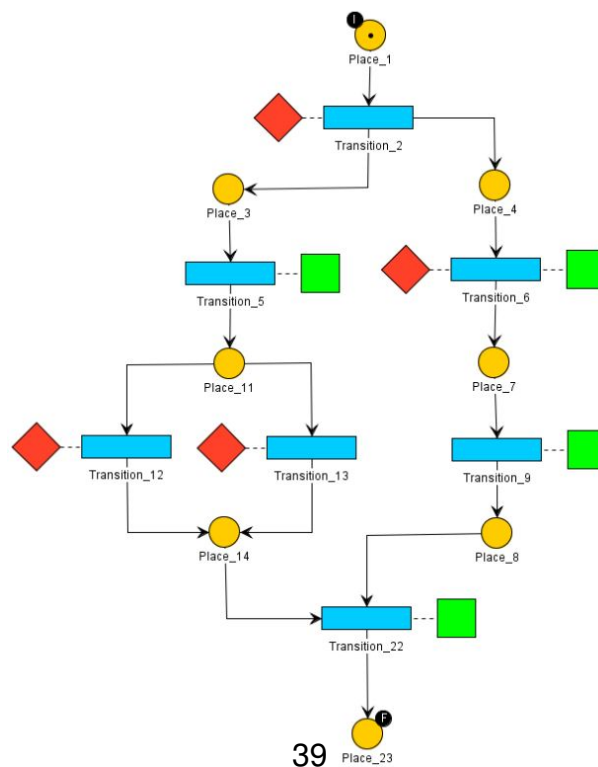


FIGURE 1.14 – Exemple de réseaux de Petri utilisés dans #FIVE et #SEVEN

L'exploitation d'un moteur #SEVEN est réalisé dans un environnement implémentant #FIVE. Elle repose sur trois classes d'entités principales.

- La classe de contexte externe, qui fournit les entrées et sorties permettant un interfaçage avec un type précis de l'environnement. Les entrées permettent d'accéder aux entités de l'environnement, et ainsi connaître leur état courant. Les sorties émettent des commandes à ces entités afin de contrôler leur comportement.
- Les senseurs de relation, qui s'assurent qu'une réalisation répondant à un ensemble de critères a eu lieu (c'est la partie **Perception** de la boucle PDA définie plus haut)
- Les Effecteurs de relation, qui permettent d'agir sur l'environnement (c'est la partie **Action** de la boucle PDA définie plus haut).

D'un autre côté, la suite logicielle HUMANS (HUMan Models based Artificial eNvironments Software platform), présentée par Lourdeaux *et al.* [83] (voir figure 1.15), est un système de scénarisation visant à favoriser la liberté d'action pour l'apprentissage, le contrôle des scénarii et l'adaptation du système nécessaires à la variabilité des scénarii. Il est composé de quatre moteurs décisionnels.

- REPLICANT, qui permet de gérer les personnages virtuels autonomes
- SELDON qui, au travers des sous-moteurs TAILOR et DIRECTOR, gère la scénarisation
- WORLD MANAGER, qui permet l'exécution de l'environnement
- MONITOR, qui gère les traces de l'apprenant

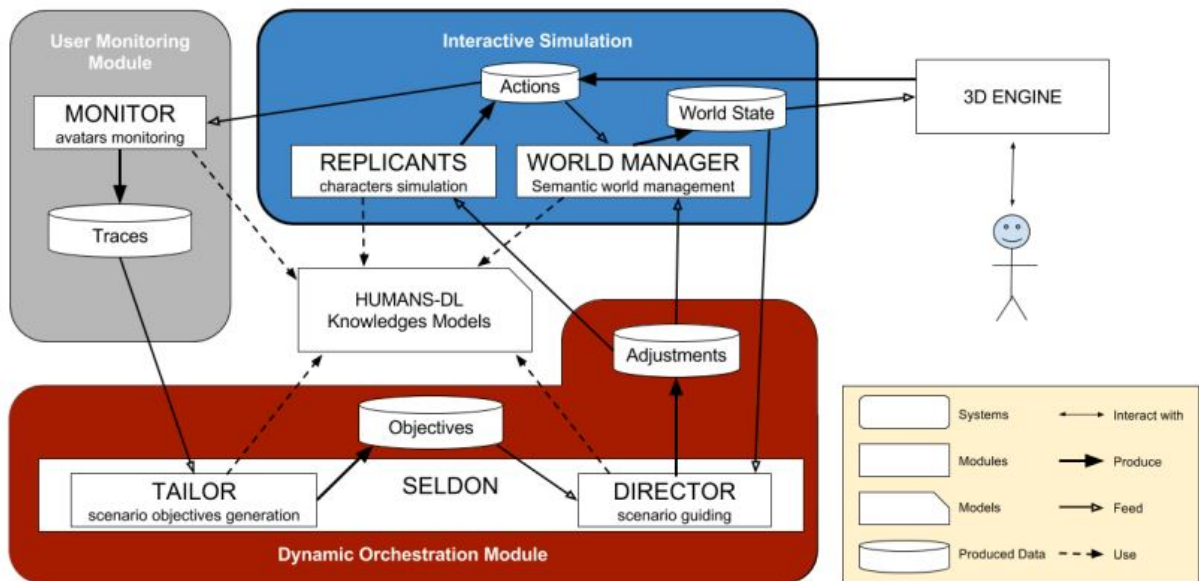


FIGURE 1.15 – Organisation générale de HUMANS selon Lourdeux *et al.* [83]

La formalisation de l'activité dans HUMANS est inspirée des langages de modélisation ergonomiques tels que MAD proposé par Perret *et al.* [100] ou GTA proposé par Van der Veer *et al.* [124]. Les activités sont représentées sous forme d'arbre dont les feuilles correspondent aux actions réalisables dans l'environnement. Leur description s'appuie sur le modèle ACTIVITY-DL (voir figure 1.16).

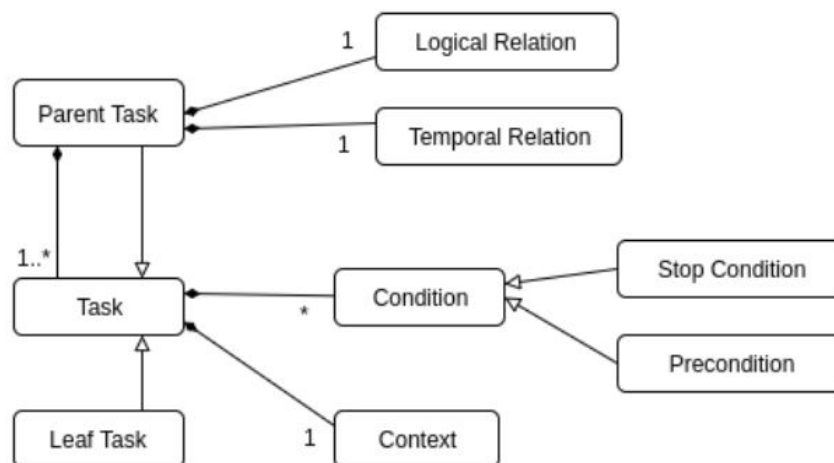


FIGURE 1.16 – Modèle ACTIVITY-DL selon Lourdeux *et al.* [83]

Dans la figure 1.16, une tâche (Task) compose une tâche mère (Parent Task), elle-même composée d'une relation logique (Logical Relation) et d'une relation tem-

porelle (Temporal Relation). Ces deux relations définissent la hiérarchie de l'activité. Une tâche est également composée d'un contexte (Context) et de conditions (Condition) pouvant être des préconditions (Precondition) ou des conditions d'arrêt (Stop Condition). La première est une condition à remplir afin d'entamer la réalisation de la tâche et la seconde une condition indiquant la fin de la réalisation de la tâche.

Le système sur lequel vont s'exécuter les activités est quant à lui représenté grâce au second modèle WORLD-DL de Lourdeaux *et al.* [83]. Il s'appuie sur les standard de représentation AWL proposé par Ivanov *et al.* [59] et RDF proposé par Brickley *et al.* [17], qui sont des formalismes informatiques, rendant la représentation du système interprétable (voir figure 1.17).

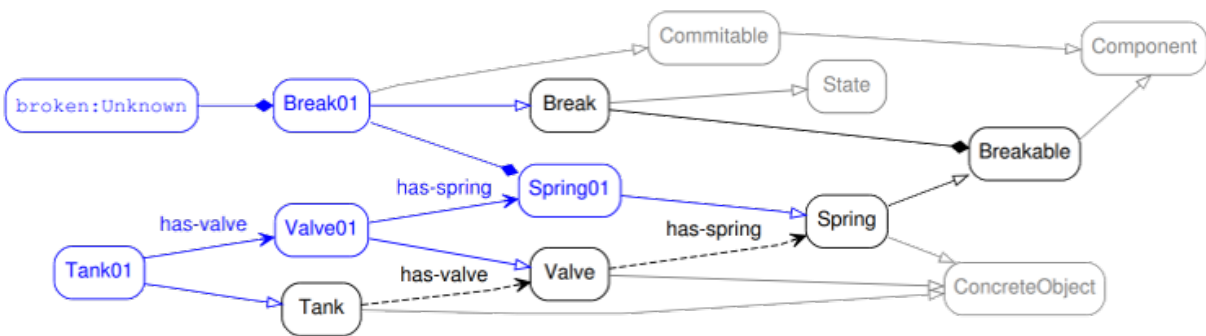


FIGURE 1.17 – Exemple de modèle décrit par WORLD-DL de Lourdeaux *et al.* [83]

Enfin, un troisième modèle, CAUSALITY-DL, permet de spécifier les relations de causalité entre les événements d'intérêt survenant dans l'EV, ainsi que les éléments bloquant la propagation de la causalité entre ces événements (voir figure 1.18).

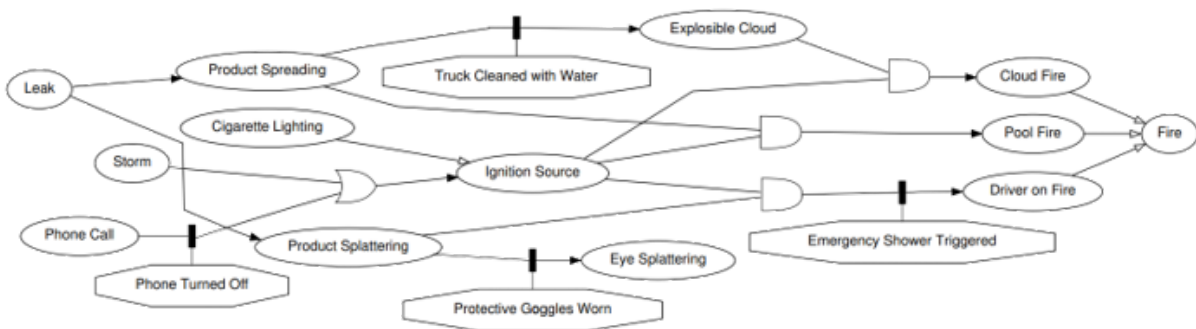


FIGURE 1.18 – Exemple représenté en CAUSALITY-DL selon Lourdeaux *et al.* [83]

INTERVALLES (INTERActive Virtual and Augmented framework for industrial Envi-

ronment and Scenarios), présenté par Richard *et al.* [104], est un framework qui permet de créer des contenus scénarisés pour des applications en RM. Il facilite la transformation de connaissances existantes en scénarii et est interopérable avec des outils tiers. Comme présenté en figure 1.19, ce modèle est composé de cinq modules différents : le module Entité-Fonctionnalité (INTERVALES Entity-Feature module), qui définit pour chaque élément actif sa représentation au sein de l'EV (entité) et sa partie fonctionnelle. Le module Interaction (INTERVALES Interactions module) gère toutes les interactions effectuées par l'utilisateur. Le module État de la scène (INTERVALES Scene State Module) prend en charge toutes les Actions/Relations effectuées. Le module Scenario Authoring (INTERVALES Scenario Authoring module) permet d'éditer des scénarios, et enfin, le module Import/Export (INTERVALES Import/Export module) est responsable de l'enregistrement et du chargement de l'état de la scène.

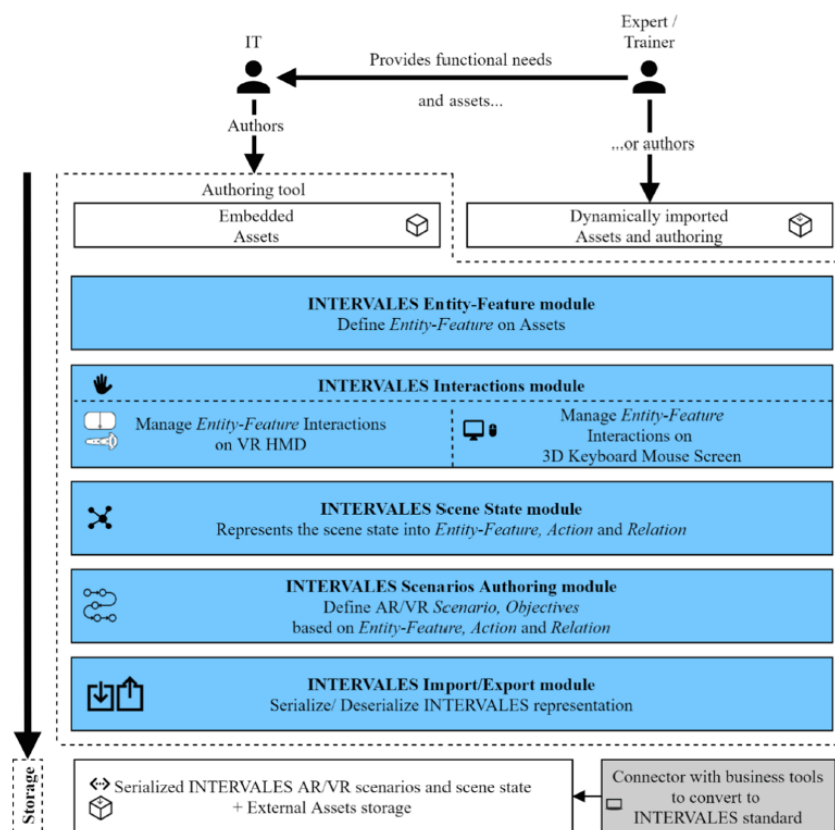


FIGURE 1.19 – Fonctions de création de l'architecture INTERVALES selon Richard *et al.* [104]

Lors de l'utilisation de ce framework et afin de réaliser un environnement de RM scénarisé (voir figure 1.20), les experts métiers (Expert dans la figure 1.20) auront le

choix de s'appuyer sur l'équipe informatique (IT dans la figure) ou de travailler en autonomie. Dans le premier cas, ils définiront ensemble les scénarii (décrits en utilisant le langage formalisé UML) qui seront joués dans l'environnement. Les experts métiers devront ensuite identifier les exigences de l'environnement afin que le service informatique puisse réaliser les développements spécifiques associés. Enfin, ils prépareront les modèles 3D grâce aux concepts Entité/Fonctionnalité (voir figure 1.20 en violet) en utilisant les interfaces utilisateur prévues à cet effet. Si aucun développement spécifique n'est nécessaire, les experts peuvent alors choisir de travailler en autonomie. Ils auront alors la charge de configurer les modèles 3D et de décrire les scénarii (voir figure 1.20 en bleu).

L'application est ensuite validée et utilisée en production (voir figure 1.20 en vert).

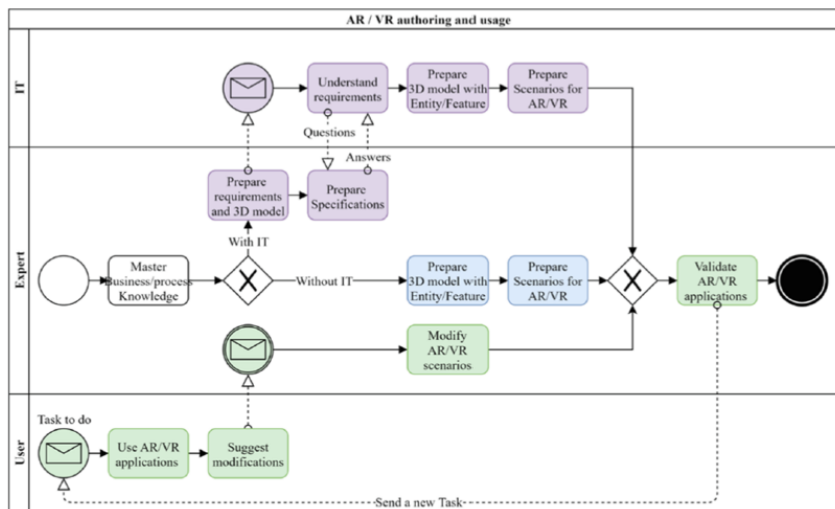


FIGURE 1.20 – Utilisation du framework INTERVALES selon Richard et al. [104]

#FIVE et #SEVEN permettent de décrire précisément les comportements d'objets interactifs et de simuler des scénarii en EV. Chaque description de comportement nécessite un développement spécifique et les scénarii simulés sont séquencés par des réseaux de Petri. HUMANS proposé par Lourdeaux *et al.* [83] représente quant à lui l'activité par des graphes de causalité (voir figure 1.18).

Cependant, si ces modèles s'appuient sur des langages formels permettant leur simulation, ils ne sont pas appropriés aux ergonomes de par le niveau de connaissances nécessaires et le manque de naturalité du langage.

Pour résoudre ces problèmes, d'autres modèles s'appuient sur une description des activités sous forme de diagrammes d'activité UML [11]. Ce langage est formel et

connu de la plupart des ergonomes. C'est le cas de MASCARET, proposé par Querrec *et al.* [102] (Multi Agent System for Collaborative, Adaptive & Realistic Environments for Training).

MASCARET, proposé par Querrec *et al.* [102] est une approche basée sur la modélisation pour la conception d'EV Intelligent orientée système et contenu. Il s'appuie sur le langage formel SysML, issue de la norme UML particulièrement adapté à la modélisation de systèmes industriels. Ce langage est suffisamment détaillé pour permettre la description des activités des utilisateurs, mais également le système sur lequel ses activités sont appliquées. Ainsi, MASCARET permet la simulation en temps réel des activités métiers décrites, et également les effets de cette activité sur le système. La modélisation du système et des activités est réalisée grâce à tout logiciel tiers capable d'exporter les modèles créés au format XMI¹. Ainsi, un même langage est utilisé par un expert métier pour décrire le système et par un ergonomiste pour décrire les activités.

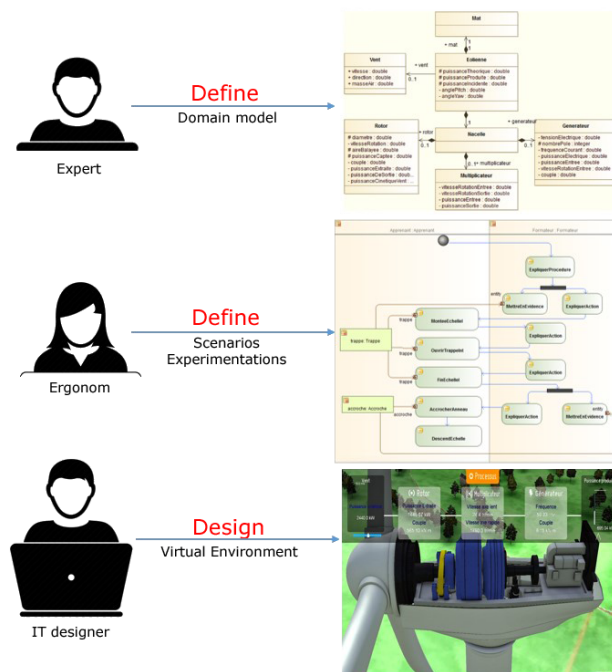


FIGURE 1.21 – Workflow de conception d'un environnement interactif avec MASCARET

1. XML Metadata Interchange. Ce format facilite l'échange standardisé de modèles d'objets et de métadonnées entre des entités travaillant dans des environnements de développement différents. XMI est basé sur trois normes industrielles : XML, UML et MOF (une norme de modélisation de groupe d'objets et de référentiel de métadonnées standard). Ainsi, son architecture permet de partager des métadonnées à l'aide d'interfaces XML.

Comme indiqué dans la figure 1.21 présentant le workflow de conception d'un EV intelligent avec MASCARET, lors de la phase de conception d'un système, le fonctionnement de celui-ci est décrit par l'expert métier sous forme de diagrammes de classes et de machines à états (en haut dans la figure 1.21) et les activités humaines sont décrites par les ergonomes sous forme de diagrammes d'activité (au milieu dans la figure 1.21). Les classes et les activités ainsi créées forment le modèle qui peut s'exécuter dans un EV conçu par un designer informaticien (en bas dans la figure 1.21).

La description des activités en MASCARET, s'appuie exclusivement sur le formalisme des diagrammes d'activité d'UML. Nous ne le détaillons pas ici, mais comme nous proposons d'utiliser MASCARET, nous montrons dans le chapitre suivant comment notre modèle d'interaction est lié au modèle de représentation de l'activité.

1.4 Conclusion

Notre objectif est de concevoir un modèle générique d'interaction qui permet à des ergonomes de concevoir un EV sans devoir faire intervenir un informaticien pour créer les liens entre les interacteurs et les actions. Dans ce chapitre, nous avons présenté les principaux travaux relatifs à cette problématique. Ainsi, nous avons vu qu'il existe des typologies d'interaction en RV, comme celle présentée par Bowman *et al.* [16]. Toutefois, ces typologies sont d'un niveau d'abstraction trop élevé pour être implémentées directement dans un EV. S'inspirant de ces typologies d'interaction en EV, il existe des typologies d'interacteurs, comme celle proposée par Lacoche [67] ou Casarin *et al.* [22]. Toutefois, bien qu'elles soient assez génériques pour prendre en compte les interacteurs courants en EV, elles ne prennent pas en compte les interactions tangibles. Pourtant, il a été montré que les interfaces tangibles (et leurs interactions/interacteurs associés) amélioreraient l'engagement de l'utilisateur selon Fiebrink *et al.* [39]. Enfin, le lien entre l'interacteur et l'action dans l'EV doit faire l'objet d'un développement spécifique, ce qui complexifie grandement la création de ces environnements pour un ergonome n'ayant pas de compétences particulières en développement.

Les interactions effectuées avec des interacteurs ont pour objectif de réaliser des actions que ce soit en Environnement Numérique ou Virtuel. En ce sens, nous avons exposé dans ce chapitre que ces actions dans un EV peuvent être représentées par des modèles d'activités. Pour les ergonomes, différents modèles d'activité existent et sont utilisés comme par exemple CTT proposé par Mori *et al.* [95], K-MAD proposé

par Baron *et al.* [8] ou HAMSTERS proposé par Barboni *et al.* [7]. Toutefois, ces modèles ne permettent pas de réaliser un lien suffisamment formel vers une description du système, ce qui empêche leur exécution dans un EV.

C'est pourquoi, en parallèle de ces premiers modèles, des travaux ont visé à proposer des modèles de représentation de l'activité spécifiquement en EV. Ces modèles sont, dans la très grande majorité, associé à un formalisme peu ou pas accessible aux ergonomes, ce qui dégrade la flexibilité nécessaire aux cycles de conception de systèmes complexes. Ce n'est pas le cas de MASCARET, proposé par Querrec *et al.* [102], qui utilise le langage formel (UML) accessible aux ergonomes et qui est suffisamment précis pour être exécuté en EV. Malheureusement, même MASCARET ne permet pas de faire de lien avec les interacteurs.

C'est pourquoi nous proposons de nous appuyer sur MASCARET pour décrire le système et les activités en EV. Nous proposons également une nouvelle typologie d'interacteurs s'inspirant de celles existantes. Nous proposons d'y intégrer la notion d'interacteur tangible en nous inspirant du modèle MCRit présenté par Ullmer *et al.* [118] et Ishii *et al.* [58].

Notre apport majeur est de proposer un modèle générique d'interaction qui fait le lien entre les interacteurs décrits à l'aide de notre typologie et les actions du modèle d'activité formalisé via MASCARET. Nous émettons alors l'hypothèse que cette démarche de conception devrait permettre (nous l'évaluerons en chapitre 4) à un ergonomiste de concevoir un EV sans faire intervenir un développeur informatique pour créer les liens entre les interacteurs et les actions métiers.

Dans le chapitre suivant, nous présentons notre modèle générique d'interaction que nous intégrons dans MASCARET, qui permet de faire le lien avec le système et les actions des activités. Notre modèle s'appuie sur une nouvelle proposition de typologie d'interacteurs intégrant les interacteurs tangibles que nous présentons également dans le chapitre suivant.

PRISME : UN MODÈLE LIANT ACTIVITÉ MÉTIER ET INTERACTION EN ENVIRONNEMENT VIRTUEL

Notre proposition a pour objectif de permettre aux ergonomes de concevoir, modéliser, suivre, voire de créer en temps réel lors de la phase de Design (*cf.* Figure 1), les activités utilisateur en EV sans développements informatiques spécifiques. Notre proposition vise donc à permettre la création d'un lien automatique entre les interactions réalisées par un opérateur via des interacteurs (périphériques d'entrées), et les actions métiers décrites dans les activités. Pour cela, nous proposons un modèle générique d'interaction appelé PRISME qui s'articule à la fois autour d'une typologie d'interacteurs de RM et tangibles adaptée aux EV que nous présentons dans la section 2.1, mais aussi d'un modèle d'interaction que nous présentons en section 2.2, et qui permet de relier un (ou plusieurs) interacteur(s) aux actions métier.

2.1 Proposition d'une typologie d'interacteurs de Réalité Mixte et Tangibles adaptée aux EV

Nous abordons ce chapitre par la proposition d'une typologie d'interacteurs. Notre proposition étend et enrichit les travaux antérieurs (*cf.* Section 1.2). Dans notre typologie, nous découpons plus formellement la description de l'interacteur et son effet. Ceci nous permet de proposer un mécanisme pour faire le lien entre l'utilisation d'un interacteur et les actions d'un modèle d'activité. En complément des travaux antérieurs, nous ajoutons les interacteurs tangibles. Notre typologie permet ainsi la description de la plus grande majorité des interacteurs existants et pouvant être utilisés en EV.

Nous découpons notre typologie en l'associant à cinq catégories :

1. les périphériques d'entrées,
2. les composants WIMP,
3. les gestes,
4. la reconnaissance vocale,
5. les objets tangibles.

Une vue générale de notre typologie est présentée par le diagramme de classes de la Figure 2.1.

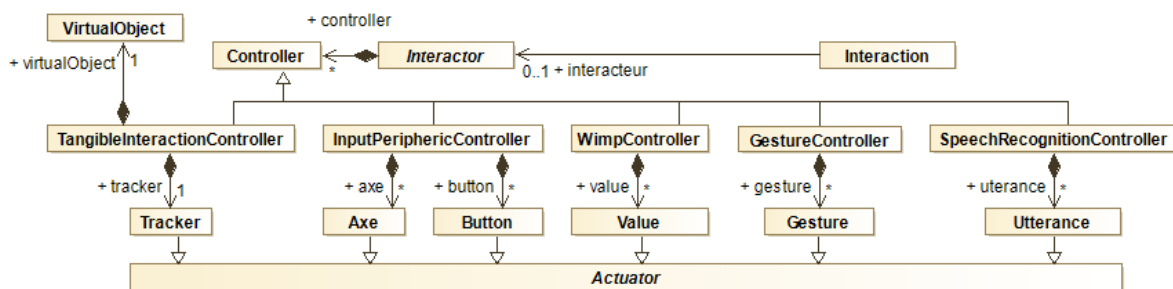


FIGURE 2.1 – Modèle de classes comportant nos cinq catégories issues de notre proposition de typologie d'interacteurs

Dans notre modèle (*cf.* Figure 2.1), un interacteur (*Interactor*) est composé d'un ensemble de contrôleurs (*Controller*), et chaque type d'interacteur en est une spécialisation. Chaque type d'interacteur est composé d'actionneurs (*Actuator*) qui leur sont propres. Par exemple, les interacteurs tangibles, représentés par la classe *TangibleInteractionController* sont composés d'objets virtuels (*VirtualObject*) et de dispositifs de localisation spatiale (*Tracker*).

Nous considérons que tout dispositif de RM et tangible peut être représenté par une composition des éléments de notre typologie.

Par exemple, un casque de RA pourrait être constitué d'un contrôleur de reconnaissance vocale (de type *SpeechRecognitionController*), d'un tracker permettant d'obtenir sa position dans l'espace, faisant du casque lui-même un objet, voire un interacteur tangible (de type *TangibleInteractionController*), de boutons poussoirs (de type *InputPeriphericsController*) et d'un contrôleur de reconnaissance de gestes (de type *GestureController*).

Nous proposons de détailler chacune des cinq catégories de notre typologie dans la suite de ce chapitre.

2.1.1 Les objets tangibles (TangibleInteractionController)

Pour définir cette catégorie, nous nous appuyons sur la définition d'Ishii et Ullmer [58]. Plus concrètement, nous incluons dans cette catégorie les objets physiques du monde réel qui peuvent représenter et manipuler des informations numériques dans un EV. Pour représenter cette catégorie, nous nous sommes également appuyés sur le modèle MCRit (voir chapitre 1.2.2 et Figure 1.3) pour lequel la partie *intangible representation* est décrite dans notre typologie par la classe *VirtualObject* (et pouvant être comparée à un jumeau numérique dont le concept a été initié par Grieves *et al.* [46][47]) et la partie *control / tangible representation* est décrite dans notre typologie par la classe *Tracker*. Le *Tracker* permet d'associer l'objet physique à un système de suivi (*e.g.*, QR code, tracker HTC Vive¹, tag RFID, *etc.*) nécessaire pour déterminer sa position dans l'espace réel et son équivalent dans l'EV. Nous modélisons les interacteurs du type *TangibleInteractionController* dans la Figure 2.2.

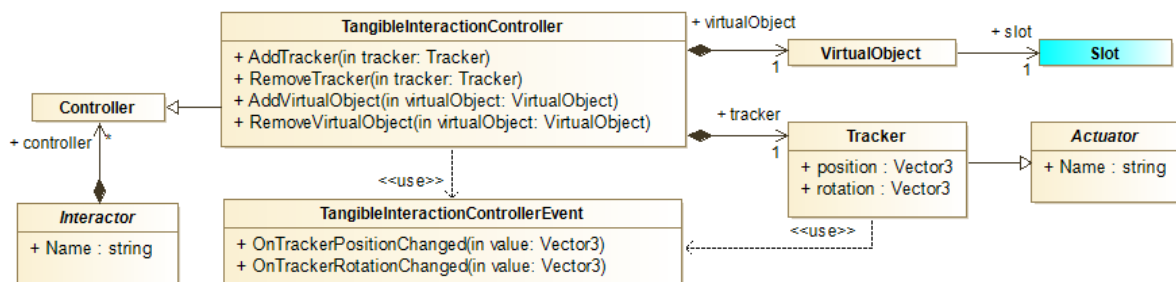


FIGURE 2.2 – Modélisation des interacteurs de type *TangibleInteractionController*

Selon ces définitions et ces modèles, un objet tangible (ou interacteur tangible) représente de manière directe une donnée du monde numérique dans le monde réel. Dans le cadre de la RM, ceci signifie que l'objet possède une représentation dans le monde réel qu'il faut être capable de localiser et une représentation dans l'EV. Cette représentation dans l'environnement est liée à la donnée manipulée par l'objet tangible. Nous avons choisi de représenter l'EV à l'aide de MASCARET, car il permet de modéliser les activités des opérateurs mais également le système technique opéré dans ces activités. Nous considérons donc que l'objet tangible manipule une donnée du système représenté par MASCARET dans l'EV. Dans la figure 2.3 nous présentons une version simplifiée du modèle MASCARET qui traite de la modélisation du système.

1. <https://www.vive.com/fr/vive-tracker/>

La conception du système est modélisée sur deux niveaux, le niveau modèle (Model) et le niveau instanciation (Environnement). Le modèle déclare des classes (Class) qui définissent des opérations (Operation) et des propriétés (Property). L'environnement référence les entités (Entity) de l'EV, qui sont des instances (Instance) des classes définies dans le modèle. Chaque instance d'une classe possède une valeur propre (ValueSpecification) pour chacun de ses attributs. Le lien entre l'attribut défini dans la classe et la valeur dans l'instance se fait à l'aide d'un Slot. Nous considérons donc que la partie intangible (VirtualObject) est liée à un Slot. Afin de permettre à l'ergonome de définir ce lien sans code informatique, nous avons proposé une interface telle que présentée dans la section 3.3 du chapitre 3.

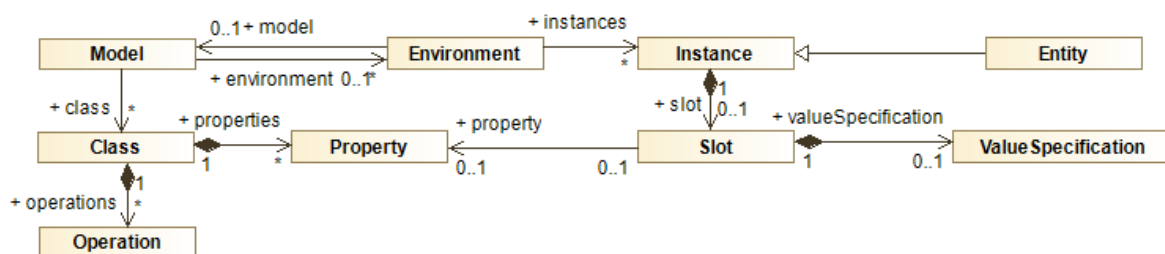


FIGURE 2.3 – Vue simplifiée du méta-modèle MASCARET permettant la définition du système.

Sur la Figure 2.2, l'actionneur de type Tracker possède deux attributs `position` et `rotation`, tous deux de type `Vector3`. Ils renseignent respectivement la position et l'inclinaison de l'objet tangible auquel le Tracker est associé. Ces données de positionnement et d'inclinaison sont appliquées en temps réel à l'objet virtuel `VirtualObject` afin de faire correspondre la représentation tangible avec la représentation intangible de l'objet.

Avec les Figures 2.4a et 2.4b, nous présentons deux exemples d'interacteurs de type `TangibleInteractionController`. Le développeur d'application de RM qui souhaite ajouter ce type d'interacteur instancie alors un interacteur de type `TangibleInteractionController`. Il instancie ensuite un actionneur de type `Tracker` ainsi qu'un actionneur de type `VirtualObject` et les ajoute à l'interacteur.

Contrairement aux TUIs, les périphériques d'entrées ne sont pas directement liés à une donnée du système simulé. L'utilisateur doit réaliser une action via ce type de périphérique pour modifier l'état du système. Nous présentons ce type d'interacteur dans la section suivante.

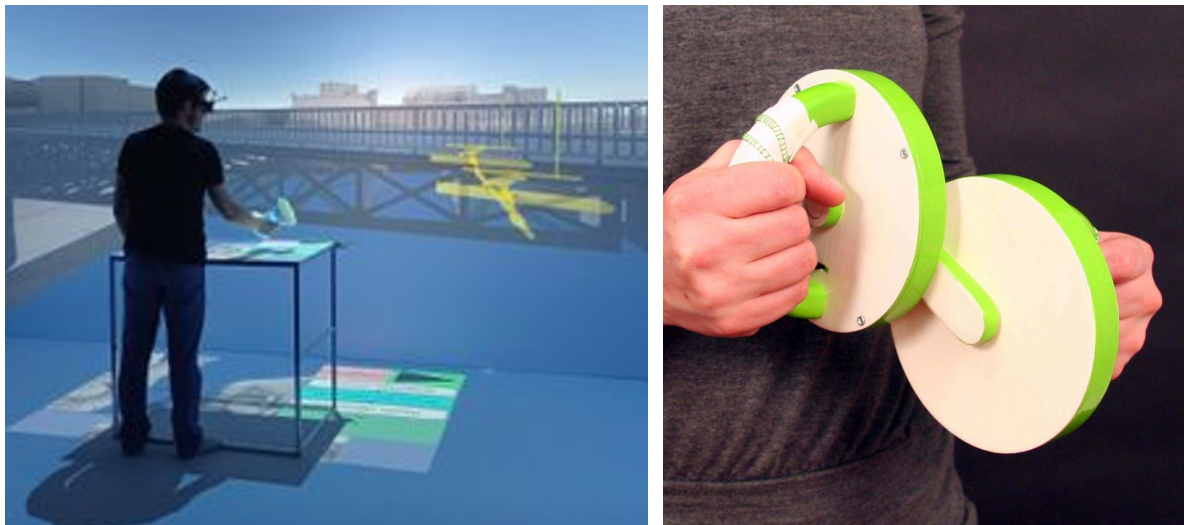


FIGURE 2.4 – Exemples d'interacteurs de type `TangibleInteractionController` : A droite, SABLIER, un interacteur tangible pour naviguer à travers l'espace et le temps en EV proposé par Mahieux *et al.* [86] et à gauche, un exemple d'objet tangible sonore proposé par Antle *et al.* [4]

2.1.2 Les périphériques d'entrées (`InputPeripheralController`)

Cette catégorie regroupe les périphériques pouvant être composés de boutons et d'axes tels que les manettes de jeu, souris, claviers, *etc.* Pour décrire ce type d'interacteur (*cf.* Figure 2.5), nous nous appuyons sur le modèle de Lacoche [64] présenté en section 1.2.2 et en Figure 1.1, où chaque *périphérique / contrôleur* peut être décrit par un ensemble d'axes et/ou de boutons.

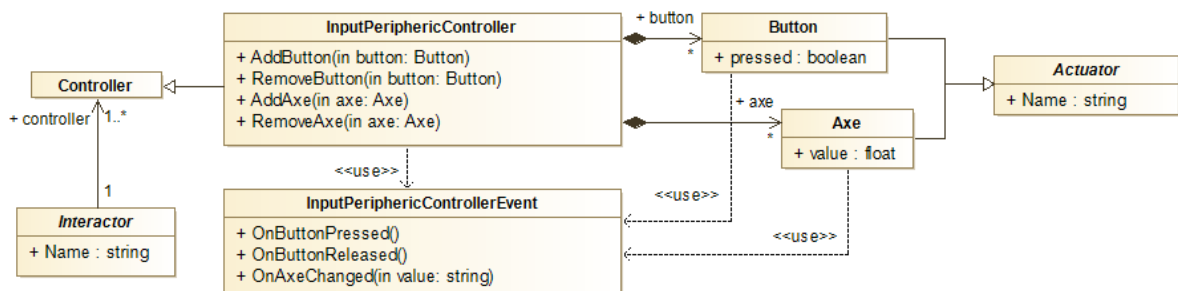


FIGURE 2.5 – Modélisation des interacteurs de type `InputPeripheralController`

Dans ce modèle (*cf.* Figure 2.5), un interacteur de type `InputPeripheralController` est composé de deux ensembles d'actionneurs (`Actuator`). Les actionneurs de

type boutons (`Button`) possèdent un attribut logique `pressed` et représentent les boutons à deux positions présents sur l'interacteur. Les actionneurs de type `Axe` représentent les actionneurs renvoyant une valeur analogique discrète ou continue (tels que le manche d'une manette, un bouton rotatif, une molette de souris, *etc.*). Lors de l'ajout d'une instance de bouton ou d'axe à une instance d'interacteur de type `InputPeripheralController`, l'instance d'interacteur s'abonne aux événements émis par ceux-ci. Ainsi, quand un bouton est appuyé ou qu'un axe est modifié, l'interacteur est notifié et émet à son tour un événement de même type. Remarquons que cette démarche (ajout d'une instance, abonnement aux événements émis, notification de l'interacteur et émission d'un événement) est semblable pour chaque catégorie d'interacteur (ex. composants WIMP, gestes) que nous nous détaillons plus bas.

Nous présentons en Figure 2.6 deux exemples d'interacteur de type `InputPeripheralController`. Le développeur d'applications de RM qui souhaite ajouter ces périphériques instancie alors deux interacteurs de type `InputPeripheralController`. Dans le cas de l'interacteur présenté en Figure 2.6a), il instancie également six axes (`Axe`) respectivement notés de 1 à 6 sur la figure et deux boutons (`Button`) respectivement notés 7 et 8 sur la figure. Les axes et les boutons sont ensuite ajoutés à l'interacteur associé.

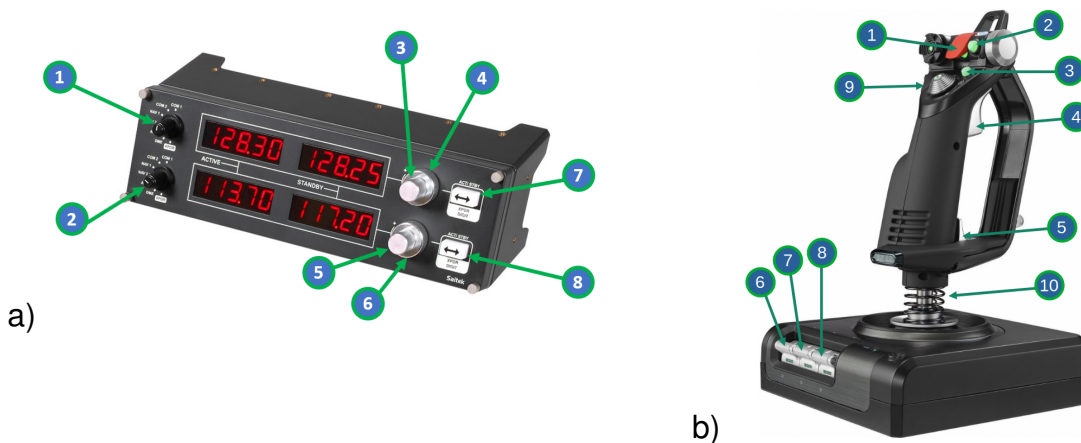


FIGURE 2.6 – Exemples d'interacteurs de type `InputPeripheralController` : a) un dispositif de contrôle radio et b) un joystick X52

2.1.3 Les composants WIMP (WIMPController)

Cette catégorie comprend les différents types de composants formant une interface WIMP (Windows, Icons, Menus, Pointing device), comme les menus, les icônes, les champs texte, les listes déroulantes *etc.*. Afin de définir cette catégorie (*cf.* Figure 2.7), nous nous sommes appuyés sur le modèle de Casarin *et al.*, [22] présenté en section 1.2.2 et en Figure 1.2.

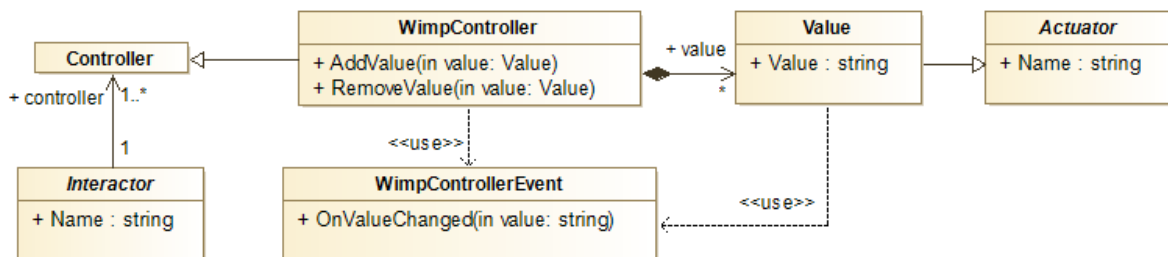


FIGURE 2.7 – Modélisation des interacteurs de type WIMPController

Dans ce modèle (*cf.* Figure 2.7), un interacteur de type WIMPController est composé d'un ensemble d'actionneurs (Actuator) de type Value. Ce type d'actionneur possède un attribut Value de type String. Cette valeur représente la valeur convertie en texte de tout élément pouvant composer une interface WIMP. Par exemple, l'index de l'élément sélectionné d'une liste déroulante sera convertie de sa valeur numérique (1, 2, 3, *etc.*) à sa valeur textuelle ("1", "2", "3", *etc.*).

Sur la Figure 2.8, nous présentons une interface utilisateur pouvant être décrit comme un interacteur de type WimpController.

Dans cet exemple, un développeur informatique pourrait instancier un interacteur de type WimpController, puis instancier un actionneur de type Value pour chaque bouton présent dans l'interface, puis les ajouter à l'instance d'interacteur.

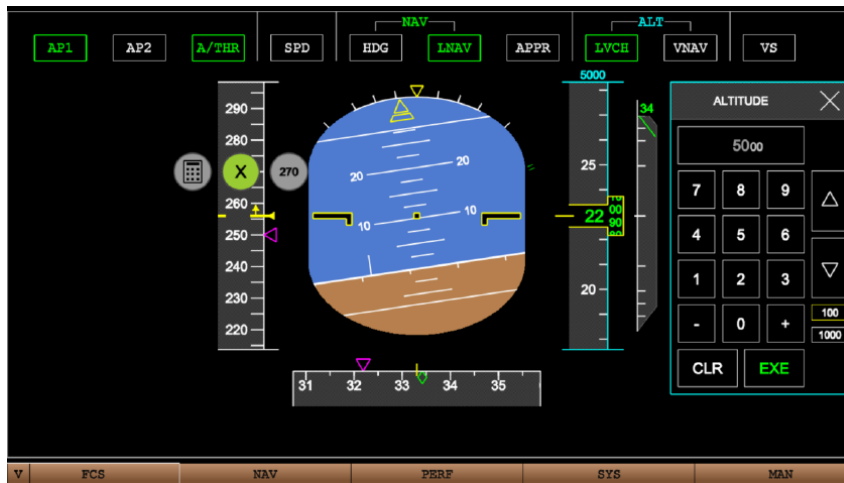


FIGURE 2.8 – Exemples d’interacteurs (composants d’interface) de type WIMPController

2.1.4 Les gestes (GestureController)

Nous rassemblons dans cette catégorie (cf. Figure 2.9) tous les périphériques permettant de réaliser de la capture gestuelle. Nous proposons de couvrir à la fois les gestes des doigts et des mains (qui peuvent être captés à l’aide de périphériques tels que le Leap Motion² ou les caméras des casques de RM) mais aussi les mouvements de l’intégralité du corps (qui peuvent être captés à l’aide de dispositifs de motion capture par exemple³).

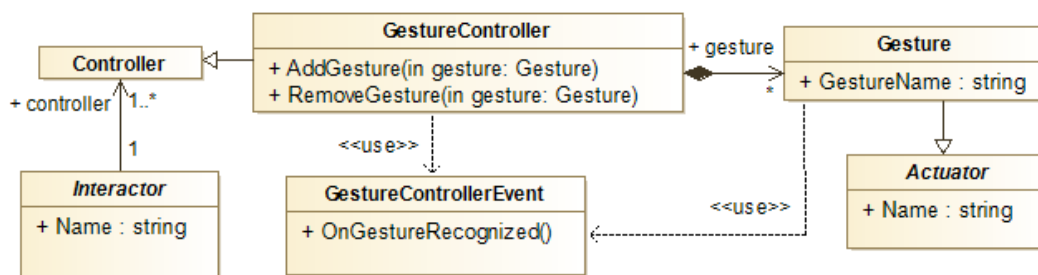


FIGURE 2.9 – Modélisation des interacteurs de type GestureController

Dans le modèle proposé en Figure 2.9, un interacteur de type `GestureController` est composé d’un ensemble d’actionneurs (`Actuator`) de type `Gesture`. Ce type

2. <https://www.ultraleap.com/product/leap-motion-controller/>

3. <https://www.xsens.com/motion-capture>

2.1. Proposition d'une typologie d'interacteurs de Réalité Mixte et Tangibles adaptée aux EV

d'actionneur possède un attribut `Name` de type `String` représentant le nom du geste reconnu.

La Figures 2.10a et 2.10b présentent deux exemples d'interacteurs de type `GestureController`. Le développeur d'applications de RM souhaitant ajouter ce type d'interacteurs à son environnement instancie un interacteur de type `GestureController` puis un actionneur de type `Gesture` (dont le nom pourrait être "Agrandir" pour la Figure 2.10a ou "Marcher" pour la Figure 2.10b). Il doit ensuite les ajouter à son interacteur préalablement instancié.

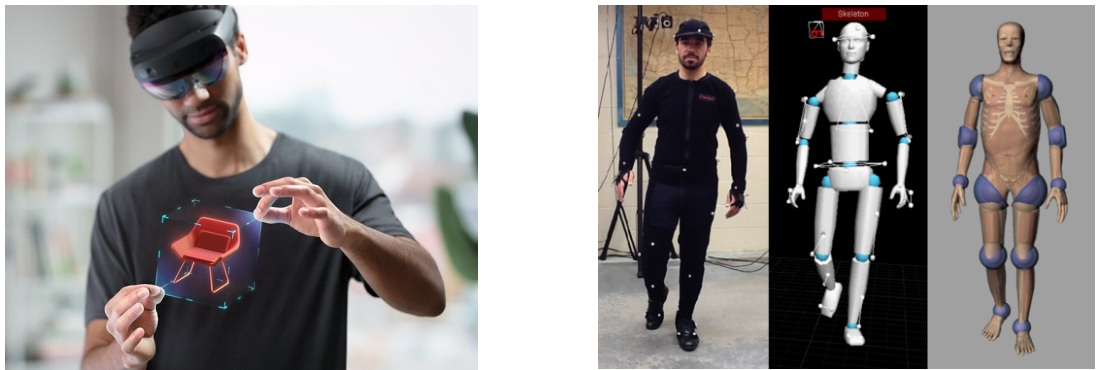


FIGURE 2.10 – Exemples d'interacteurs de type `GestureController` dont le périphérique de capture est basé sur un casque de Réalité Augmentée (à droite) et sur un dispositif de capture de mouvements (à gauche)

2.1.5 La reconnaissance vocale (`SpeechRecognitionController`)

Ce type d'interacteur permet des saisies vocales sous forme de commandes (e.g., ouvrir, fermer) ou de phrases en langage naturel.

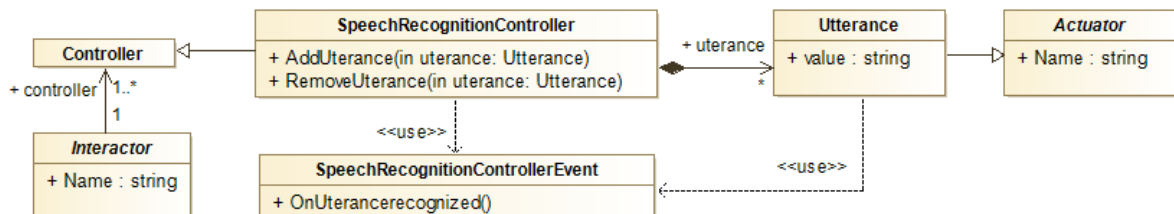


FIGURE 2.11 – Modélisation des interacteurs de type `SpeechRecognitionController`

Le texte ainsi capté est transmis au système sous forme de chaînes de caractères et peut être alors comparé ou interprété si le système dispose d'un interpréteur sé-

mantique. Nous modélisons cette catégorie d'interacteurs par la Figure 2.11. Dans cette figure, un interacteur de type `SpeechRecognitionController` est composé d'un ensemble d'actionneurs (`Actuator`) de type `Utterance` (expression en français). Ce type d'actionneur possède un attribut de type texte `value`. Il correspond au mot ou à la phrase prononcée par l'utilisateur. Lors de l'ajout d'une instance de `Utterance` à une instance d'interacteur de type `SpeechRecognitionController`, l'instance d'interacteur s'abonne aux événements émis par ceux-ci. Ainsi, quand l'utilisateur prononce un mot ou une phrase, l'interacteur est notifié et émet à son tour un événement de même type.



FIGURE 2.12 – Exemple d'interface pouvant utiliser un interacteur de type `SpeechRecognitionController` dont le périphérique de capture pourrait être ici basé sur le micro d'un casque de Réalité Mixte

A l'aide de la Figure 2.12 nous présentons un exemple d'interface pouvant utiliser un interacteur de type `SpeechRecognitionController` dont la capture pourrait être basé sur le micro d'un casque de Réalité Mixte.

2.2 Modèle d'interaction

En nous appuyant sur notre typologie d'interacteurs proposée et présentée précédemment, mais aussi sur le méta-modèle MASCARET qui permet la description des systèmes et activités en EV, nous proposons dans cette section un modèle d'interaction permettant de lier les interacteurs à l'exécution des actions d'une activité métier.

Afin de lier les interactions (`Interaction`) aux interacteurs (`Interactor`), une interaction référence un ou plusieurs interacteurs. Un interacteur est composé d'un ou plusieurs contrôleurs (`Controller`). Ce référencement permet de spécifier le type d'in-

teracteur (*Interactor*) utilisé pour réaliser une interaction (*Interaction*), comme indiqué dans la Figure 2.13.

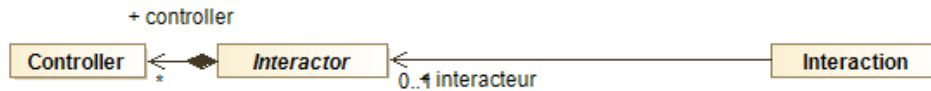


FIGURE 2.13 – Représentation de PRISME avec un focus sur la liaison entre Interaction et Interacteur

Dans notre modèle, les interactions sont spécialisées selon les quatre grandes catégories d'interaction définie par Bowman *et al.* [16] (*cf.* Figure 2.14).

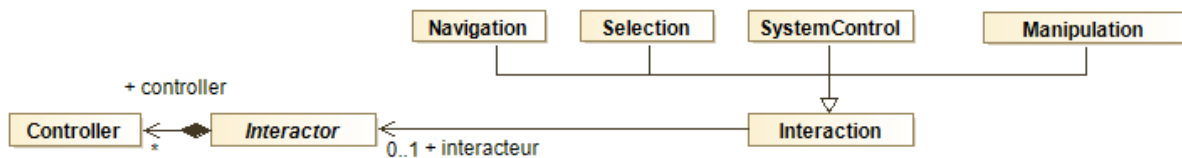


FIGURE 2.14 – Représentation de PRISME avec un focus sur les types d'interaction selon Bowman *et al.* [16]

Dans nos travaux, nous nous intéressons particulièrement aux interactions de type *Manipulation*. En effet, nous estimons que la majorité des interactions avec un élément d'un EV résultent d'une manipulation de l'utilisateur. Dans les travaux antérieurs, les actions de manipulation se limitent à la modification de la position et de l'orientation des objets de l'environnement virtuel. Nous considérons que les actions de manipulation couvrent également la modification de l'état interne d'un objet, l'activation d'une capacité qui lui est propre (appuyer sur un bouton virtuel pour exécuter l'effet lié à cet appui, ou encore faire entrer en collision deux objets virtuels pour activer un effet lié à cette collision) ou encore le changement d'une propriété du modèle. Classiquement, ces effets sur l'environnement sont considérés dans la catégorie *SystemControl*.

Nous n'avons pas formalisé de type d'interactions lié à la sélection, mais PRISME nous permet d'identifier quelques stratégies génériques pour lier la sélection d'entités et leur manipulation. Par exemple, un utilisateur pourrait vouloir utiliser un tournevis en tant qu'interacteur tangible afin de visser une vis virtuelle. Le tournevis pourrait être décrit dans PRISME comme étant une instance d'interacteur tangible de type

(TangibleInteractorController) et la vis comme étant une instance d'une classe Vis. La sélection pourrait être réalisée par contact entre la représentation virtuelle du tournevis et la vis virtuelle. Nous n'avons pas non plus traité les actions de type Navigation, car d'une part, dans notre contexte, il ne s'agit pas d'un enjeu qui porte sur l'utilisation du système technique, et d'autre part, ce type d'action est déjà largement étudié [71][52].

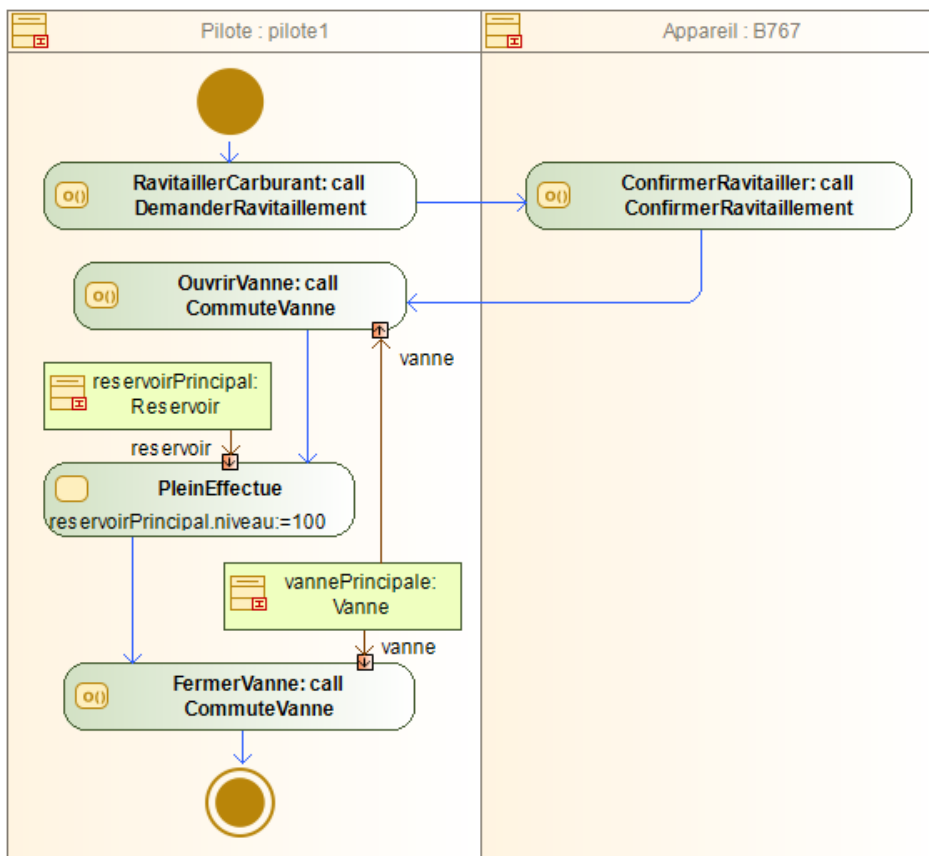


FIGURE 2.15 – Description d'une activité ravitaillement d'un avion par un diagramme d'activités

Nous considérons donc qu'une interaction de type Manipulation à un effet sur le système. Les effets attendus par l'opérateur sur ce système sont décrits, ou sont à concevoir, à l'aide d'activités ou de procédures par l'ergonome. Dans le cadre de ce travail, nous proposons que l'ergonome décrive les activités sur le système à l'aide de MASCARET sous forme de diagramme d'activités UML. Nous présentons un exemple de ce type d'activité qu'un ergonome pourrait rédiger sur la figure 2.15. Dans cette

activité, plusieurs rôles interviennent : un pilote et l'équipage d'un appareil de ravitaillement. Le pilote réalise des actions en séquence en réalisant soit des communications avec l'avion de ravitaillement, soit en manipulant des éléments du système (avion) comme le réservoir et la vanne.

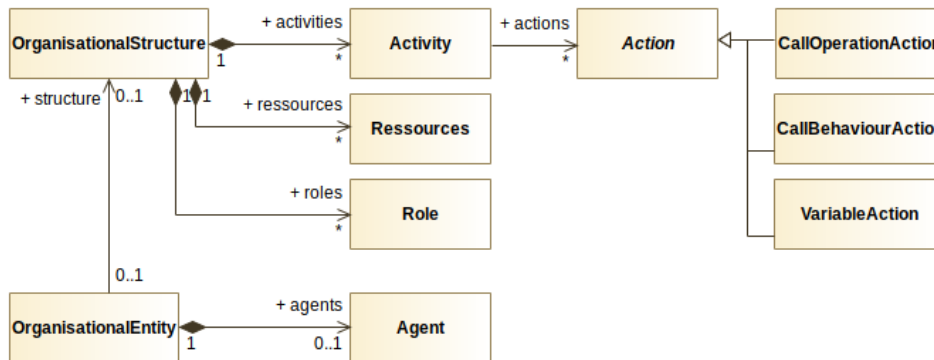


FIGURE 2.16 – Vue simplifiée du méta-modèle MASCARET permettant la description d'une activité.

Afin de formaliser le lien entre les interactions décrites dans PRISME et les actions (*Action*) de MASCARET, une interaction (*Interaction*) de type manipulation (*Manipulation*) référence une action (*Action*), comme le montre la figure 2.17.

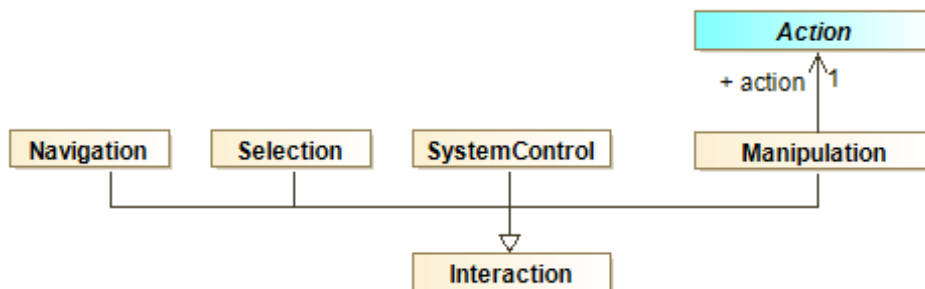


FIGURE 2.17 – Lien entre une Manipulation de notre modèle et une Action de MASCARET

Dans MASCARET, et comme le montre la Figure 2.18, une activité (*Activity*) agence des actions (*Action*) à l'aide d'opérateurs (*e.g.*, séquence, conditionnel, itération, *etc*).

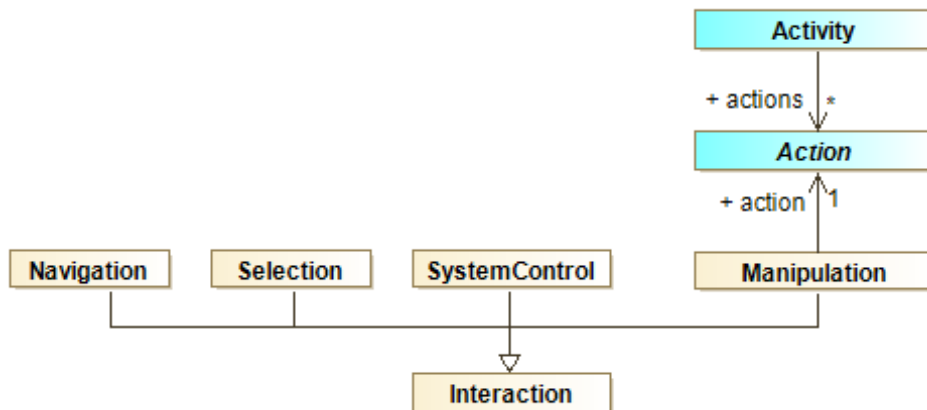


FIGURE 2.18 – Liaison entre une Action et une Activity

MASCARET s'appuie sur UML qui définit de manière exhaustive des types d'actions tels que ceux réalisant des opérations (`CallOperationAction`), ceux modifiant des valeurs du système (`VariableAction`) ou ceux gérant la hiérarchie des activités en appelant d'autres activités (`CallBehaviorAction`). De plus, MASCARET ajoute d'autres types d'actions tels que ceux lançant des animations au sein d'un E.V. (`PlayAnimationAction`) ou ceux gérant la communication inter-agent (`CommunicationAction`) (voir figure 2.19).

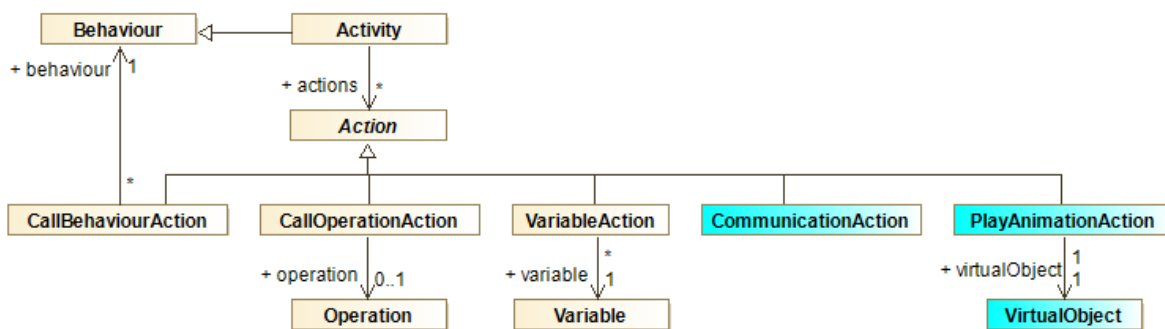


FIGURE 2.19 – Exemple de types d'action présents dans la norme UML (en jaune) et deux exemples de type d'action ajouté par MASCARET (en bleu)

Comme présenté dans la Figure 2.20, par la suite, nous illustrons le principe de lien avec les actions uniquement avec les actions de type `CallBehaviourAction`, `VariableAction` et `CallOperationAction` dont les `Operation` qu'elles référencent, peuvent avoir un effet sur l'EV via les entités passées en paramètres (`Entity`). Toutefois, le principe reste le même pour toutes les autres actions.

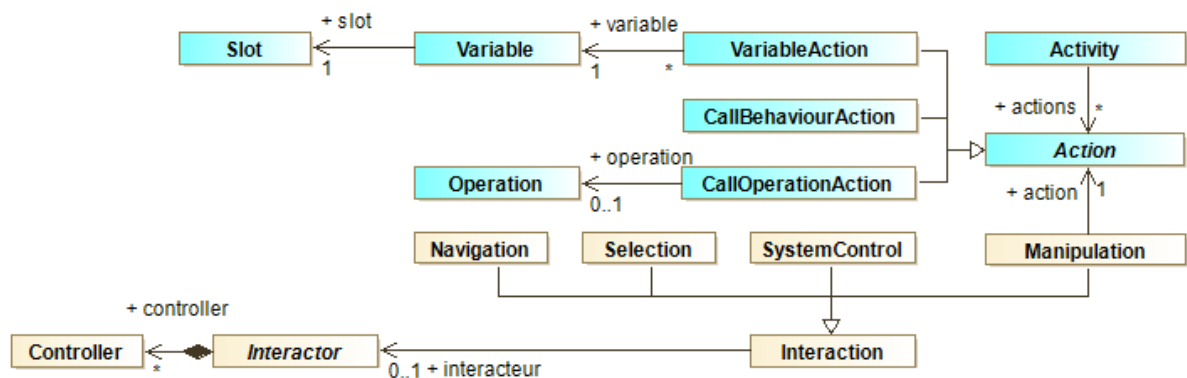
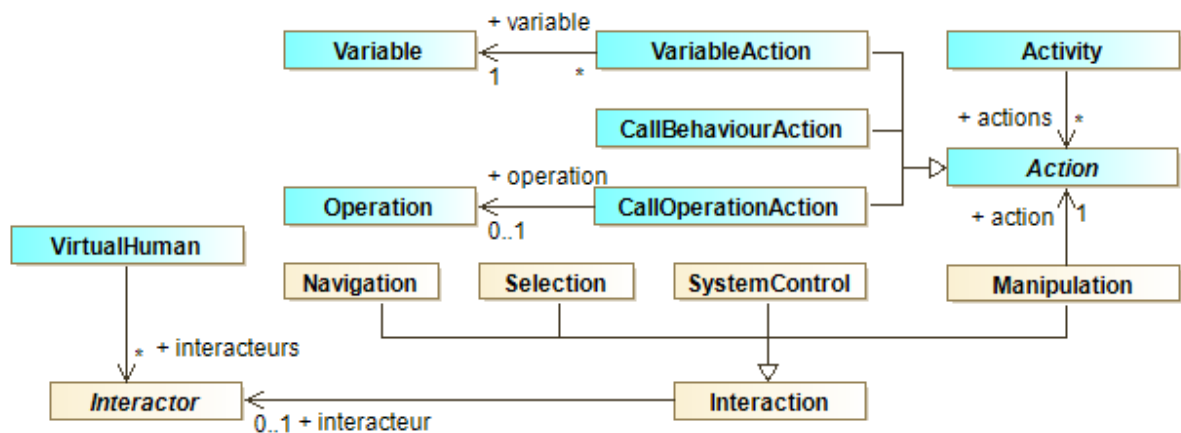


FIGURE 2.20 – Intégration des types d’actions dans notre proposition

Enfin, un interacteur est référencé par un `VirtualHuman`, comme présenté dans la Figure 2.21. Un `VirtualHuman` peut représenter un agent autonome incarné dans l’EV ou un utilisateur manipulant les interacteurs à sa disposition.

FIGURE 2.21 – Modèle présentant la liaison entre les interacteurs (`Interactor` manipulés par l’utilisateur et les actions (`Action`) issues d’une activité (`Activity`)

2.3 Conclusion

Dans ce chapitre, nous avons présenté comment notre modèle d’interaction, que nous avons appelé PRISME, permet à un ergonomiste de lier les interactions réalisées

par un utilisateur et les actions métier décrites dans une activité. Afin de réaliser ce lien, un ergonome réalise la description du système, des activités qui s'y rapportent et des interacteurs impliqués en utilisant notre modèle PRISME. Il décrit ensuite les interactions liant les interacteurs décrits aux actions de l'activité. Une fois ces descriptions réalisées, le système instancie ensuite automatiquement les éléments décrits par l'ergonome. Ainsi, une interaction spécifique décrite par un ergonome permet, lorsqu'elle est reconnue, d'exécuter l'action de l'activité qui lui est associée.

Nous avons, en début de chapitre, proposé une typologie permettant de décrire la grande majorité des interacteurs pouvant être utilisés en Environnements Virtuels. Notre typologie se compose de cinq types d'interacteurs : les périphériques d'entrées, les composants WIMP, les gestes, la reconnaissance vocale et enfin les objets tangibles. Dans la suite de ce chapitre, nous avons présenté la réalisation de la description d'un système et des activités qui lui sont liées grâce au méta-modèle MASCARET. Dans un environnement MASCARET, le système est décrit en utilisant les concepts de classes, de propriétés et d'opérations. Les activités sont décrites par des diagrammes d'activités en utilisant les notions d'actions, de ressources et de rôles. L'intégralité des modèles conceptualisant un système ainsi que les interacteurs décrits grâce à notre typologie sont ensuite instanciés par le système au moment de son exécution sans avoir recours à l'intervention d'une équipe informatique. Cela permet à un ergonome de réaliser des modifications dans le modèle ou l'activité en toute autonomie. Pour finir, nous avons détaillé comment nous avons intégré les notions d'interacteur et d'interaction au méta-modèle MASCARET afin de lier les interacteurs décrits par notre typologie aux actions d'un modèle d'activités, une interaction étant composée d'un ensemble d'interacteurs et d'une action.

Dans le chapitre suivant, nous présentons le cas applicatif dans lequel nos travaux de thèse s'inscrivent, et comment nous y avons instancié notre modèle.

UTILISATION DE NOTRE TYPOLOGIE ET IMPLÉMENTATION DE PRISME SUR UN CAS D'USAGE EN AÉRONAUTIQUE

Nos travaux de thèse s'inscrivent dans un projet financé par l'ANR (Agence Nationale de la Recherche)¹ et porté par l'IRT (Institut de Recherche Technologique)² b<>com. Ce projet, nommé VUXIA pour *Virtual platform for rapid UX design and IA data collection* a pour objectif le développement de nouveaux outils et de plateformes permettant de raccourcir les coûts et les délais des processus de conception de systèmes complexes (ex. un cockpit d'avion ou un tableau de bord de voiture). Pour cela le projet s'appuie sur l'utilisation des interfaces tangibles mêlées à la Réalité Mixte. Plusieurs partenaires prennent part à ce projet dont HDG (Human Design Group) qui est une société de conception d'interfaces utilisateur employant tout particulièrement des ergonomes. Dans le contexte du projet, nous avons choisi d'aborder un cas d'étude lié à la conception d'un cockpit d'avion dont les éléments nécessaires au pilotage (*e.g.*, manche, manette des gaz, écrans tactiles, écrans d'affichage, *etc.*) sont des éléments tangibles de l'interface (nous présentons plus en détails notre plateforme dans la section suivante). Dans ce projet, la conception de la plateforme est réalisée en deux phases :

1. Une phase de design/conception qui consiste, d'une part, à décrire le système ainsi que les éléments constituant la plateforme et, d'autre part, à positionner en réalité mixte et de façon tangible ces différents éléments. Par exemple, un cockpit d'avion peut être composé d'un écran de contrôle. Le concepteur / ergonomes peut, par exemple, choisir de positionner l'écran (objet physique/tangible) face au pilote, à sa droite ou à sa gauche.

1. <https://anr.fr/>

2. <https://b-com.com/>

2. La seconde phase est une phase d'évaluation de la configuration du système (positionnement des éléments de l'interface). Elle consiste à mesurer l'efficacité des opérateurs sur la plateforme lors de la réalisation d'activités. Durant cette phase, les ergonomes en charge de la conception suivent en temps réel le déroulement de l'activité et enregistrent des indicateurs clés (KPI : *Key Performance Indicator*) définis au préalable. Pour reprendre notre exemple précédent, un ergonome pourrait donc mesurer si la position de l'écran de contrôle est préférable en face, à gauche ou à droite du pilote.

Une fois ces deux phases terminées, un éventuel nouveau cycle de conception est lancé jusqu'à l'obtention d'une configuration adéquate.

Nos travaux visent à contribuer tant à la phase de conception qu'à la phase d'évaluation. En effet, PRISME permet d'une part de décrire le système et les activités qui lui sont associées, mais également de réaliser le suivi, voire la création des activités en EV. Dans ce chapitre, nous présentons premièrement notre plateforme opérateur représentant un cockpit d'avion ainsi que les éléments qui la composent. Dans un second temps, nous présentons comment un ergonome peut décrire les interactions liées aux actions de l'activité, suivre ou créer les activités grâce à notre modèle PRISME et, *in fine*, évaluer sa conception.

3.1 Vers une plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en Environnement Virtuel

Notre cas d'usage vise à apporter un contexte dans lequel la conception d'une plateforme (dans notre cas, un cockpit d'avion) serait à l'étude par un ergonome. A l'aide de cette plateforme, il serait en mesure d'étudier l'impact du placement des éléments de l'interface sur la performance de l'opérateur (un pilote de ligne).

L'idée ici n'est pas de concevoir une maquette haute définition du système (cockpit), car cela engendrerait des temps et des coûts supplémentaires tels que nous l'avons expliqué en introduction générale. Notre objectif est plutôt de concevoir une plateforme générique constituée d'interacteurs tangibles. Ces derniers sont augmentés par l'usage de la Réalité Mixte afin d'améliorer l'interaction et l'immersion d'un opérateur

3.1. Vers une plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en Environnement Virtuel

dans un EV. Un opérateur (ici un pilote de ligne) interagit alors via cette plateforme dans l'EV afin de réaliser des activités. Un ergonomiste suit ces activités via une interface qui lui est dédiée (un client web que nous avons développé) afin de paramétrer et d'évaluer l'interface (les interactions, la position des interacteurs, la réalisation des activités, etc.). Nous illustrons le principe global par la Figure 3.1.



FIGURE 3.1 – Principe global de notre plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en EV.

De manière générique, notre plateforme physique (*cf.* Figure 3.2) est composée de 3 plans sur lesquels sont "projetés" les interfaces du pilote. Celle-ci est également composée de dispositifs aéronautiques (interacteurs tangibles) spécifiques à notre cas d'usage (*i.e.*, manche, radio et gestionnaire de carburant).

Un panneau de supervision mobile ② peut être placé dynamiquement sur ces plans. Ce panneau est décrit par le type d'interacteur `TangibleInteractorController` de notre typologie, composé d'un objet virtuel de type `VirtualObject` et d'un dispositif de tracking de type `Tracker`. La position de cet interacteur est connue en temps réel. Un évènement (`Event`) est envoyé lors d'un mouvement. Aucun développement spécifique n'a été nécessaire ici, car nous utilisons des trackers standards (*e.g.*, HTC Vive) appartenant à une bibliothèque d'interacteurs "standards".

La plateforme est également composée de deux interacteurs spécifiques au métier étudié dans notre cas d'usage. Il s'agit d'un dispositif de gestion des niveaux de carburant ① et d'un dispositif de changement de fréquence radio ③. Ces interacteurs sont

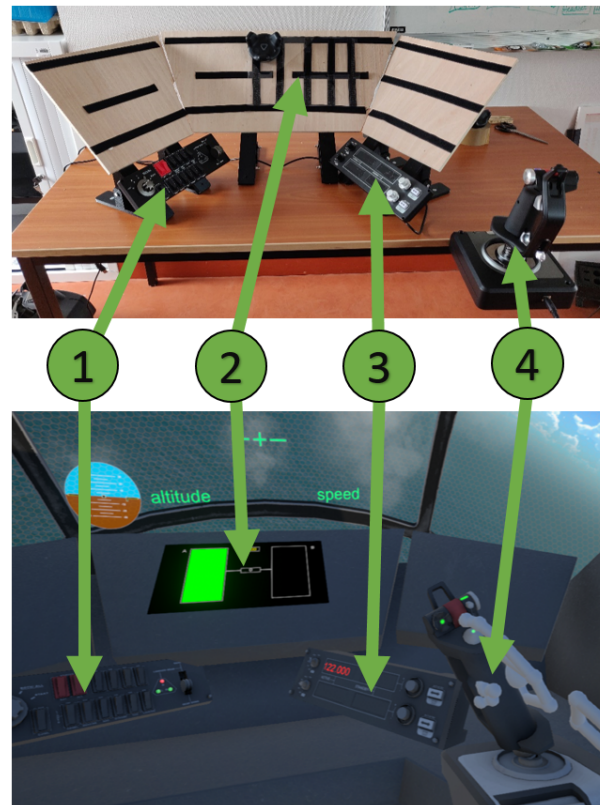


FIGURE 3.2 – Notre plateforme physique constituée d’interacteurs tangibles (en haut) et l’EV (ici projeté dans un casque de RV) représentatif d’un cockpit d’avion (en bas)

constitués d’un `InputPeripheralController`, composé de plusieurs boutons. Un événement est envoyé pour chaque changement d’état d’un bouton. Ils sont également constitués d’un `TangibleInteractorController`, composé d’un objet virtuel et d’un dispositif de tracking. Un événement est envoyé lorsqu’un début ou une fin de mouvement est détectée ou lorsque la position de l’interacteur atteint un seuil prédéfini. Ces interacteurs sont très spécifiques et directement liés au domaine métier (aéronautique) de notre plateforme. C’est pourquoi nous avons dû faire des développements dédiés. Cependant, ces développements ont uniquement consisté à lier le driver de ces interacteurs aux classes et au système d’évènements de PRISME.

Enfin, la plateforme est constituée d’un manche ④, décrit par un interacteur de type `TangibleInteractorController` (lui aussi composé d’un objet virtuel et d’un dispositif de tracking) et par un interacteur de type `InputPeripheralController` (composé de plusieurs boutons et plusieurs axes). Comme mentionné pour les dispositifs ci-dessus, un événement (`Event`) est envoyé lors d’un mouvement et à chaque changement d’état

3.1. Vers une plateforme générique de simulation mêlant Réalité Mixte et interacteurs tangibles en Environnement Virtuel

d'un bouton. Aucun développement spécifique n'a été nécessaire ici.

L'EV représente le modèle 3D du cockpit d'avion en cours d'étude par l'ergonome. L'opérateur (le pilote) est immergé dans cet EV à l'aide d'un casque de RV (cf. Figure 3.2) ou d'un casque de RA (cf. Figure 3.3) selon les besoins.



FIGURE 3.3 – Notre plateforme physique et l'EV (ici projeté dans un casque de RA). Les écrans sont placés en fonction de la position du tracker et le pilote peut contrôler l'avion à l'aide des interacteurs tangibles et de la voix.

Ces dispositifs (casques de RA ou de RV) sont équipés d'un micro qui sert au pilote à donner des instructions vocales (comme lors d'un vol). Le micro est représenté dans la plateforme par un interacteur de type `SpeechRecognitionController` et émet un événement (`Event`) lorsqu'un mot ou une phrase sont prononcés. Dans notre cas d'étude, l'avion est en vol et les différents instruments de vol (e.g., horizon, altitude, etc.) sont simulés. Plus précisément, dans l'activité liée à notre cas d'étude, le panneau de supervision ② présente les informations sur les niveaux de carburant. Il s'agit d'un des éléments du cockpit dont l'ergonome veut évaluer la position. Point important de notre apport, durant la simulation, l'utilisateur (le pilote) a la possibilité de modifier l'emplacement de l'ensemble de ces interacteurs et écrans afin de les positionner à sa convenance pour raison ergonomique ou pratique par exemple. Dans ce cas, le déplacement de l'élément tangible (`TangibleInteractorController`) du cockpit (e.g., panneau de supervision) est répercuté dans l'EV avec le déplacement de son homologue virtuel (`VirtualObject`) sans qu'aucun développement ne soit nécessaire.

Dans la section suivante, nous présentons comment un ergonome réalise la description des interacteurs d'une plateforme présents dans un EV.

3.2 Description des interacteurs constitutifs d'une plateforme à l'aide de PRISME

Lors de l'acquisition d'un nouvel interacteur, et afin de rendre cet interacteur exploitable dans PRISME, il est nécessaire de réaliser sa description. Cette description s'appuie sur notre typologie d'interacteurs et permet d'exploiter les données fournies par le nouvel interacteur. Elle est réalisée par une équipe informatique sous la forme d'un code logiciel liant les données fournies par le périphérique (appui sur un bouton, valeur d'un axe, texte reconnu, *etc.*) aux actionneurs de notre typologie, selon leur nature (Button, Axe, Utterance, *etc.*). Le code suivant (*cf.* Figure 3.1) est un exemple d'instanciation d'un interacteur (CustomInteractor) contenant un contrôleur de type SpeechRecognitionController et contrôleur de type InputPeriphericController.

Listing 3.1 – Exemple de description d'interacteur

```
1 public class CustomInteractor
2 {   public CustomTangibleInteractor(string name)
3     {   Interactor cInteractor = new Interactor(name);
4
5         Controller IPController = new InputPeriphericController();
6         IPController.AddButton(new Button("Button 1", KeyCode.
7             JoystickButton2));
8         IPController.AddButton(new Button("Button 2", KeyCode.
9             JoystickButton5));
10        IPController.AddAxe(new Button("Slider 1", "Axis 8"));
11        IPController.AddAxe(new Button("Slider 2", "Axis 9"));
12        cInteractor.AddController(IPController);
13
14        Controller SController = new SpeechRecognitionController();
15        SController.AddUtterance(new Uretance());
16        cInteractor.AddController(SController);
17
18        PRISME.AddInteractor(cInteractor);
19    }
20 }
```

Cet exemple montre comment un interacteur ainsi que les contrôleurs et leurs actionneurs qui le composent sont instanciés et, comment le nouvel interacteur est ajouté à une instance de PRISME. Un interacteur ainsi décrit pourra, par la suite, être utilisé selon les besoins des futurs projets, sans autre intervention de l'équipe informatique.

Nous venons de présenter comment décrire un interacteur grâce à notre typologie d'interacteurs. Dans la section suivante, nous expliquons comment nous lions les interactions décrites par notre modèle et les actions décrites dans le modèle d'activité. Nous terminons ce chapitre par la présentation du suivi et de la création d'une activité.

3.3 Définition des interactions dans PRISME

Notre modèle d'interaction PRISME permet à un ergonome de lier les effets des interacteurs disponible dans l'EV et les actions métier décrites dans une activité. Pour cela, tant les interacteurs présents dans l'environnement, que le système sur lequel les activités seront appliquées mais aussi les activités elles-mêmes, doivent être instanciés dans notre modèle PRISME. Ainsi, une interaction spécifique décrite par un ergonome permet, lorsqu'elle est reconnue, d'exécuter l'action de l'activité qui lui est associée (ex. "tourner un bouton spécifique" permet de "changer la fréquence radio").

La description d'une interaction se réalise en deux étapes :

1. Spécifier le type d'action souhaitée (ex. modifier un attribut, exécuter une opération, lancer une animation, *etc.*),
2. Spécifier l'actionneur qui est référencé pour cette interaction.

Pour effectuer cela dans PRISME, une interface dédiée aux ergonomes a été développée (*cf.* Figure 3.4). Cette interface permet de décrire une interaction et de l'instancier dans PRISME. Celle-ci s'enregistre alors aux événements des actionneurs qu'elle référence. Lorsque ces événements sont émis, l'interaction est déclenchée et l'action correspondante peut alors être réalisée. Cette interface est générée par l'EV et est accessible depuis un navigateur. Elle communique en temps réel avec l'environnement et permet la modification ou l'instanciation des interactions, le suivi de l'évolution des indicateurs clés sélectionnés par l'utilisateur ainsi que le suivi ou la création de l'activité.

Cette interface se découpe en trois parties distinctes :

1. Une partie *Profils* qui permet à un ergonome de sélectionner le profil dans lequel il souhaite sauvegarder les interactions qu'il spécifie (*cf.* Figure 3.5). Ainsi, pour une même simulation, un ergonome pourra tester différentes configurations sans devoir relancer toute la simulation. Elle permet également de sélectionner le type d'action à lier à l'interaction (ex. exécuter une opération, modifier la valeur

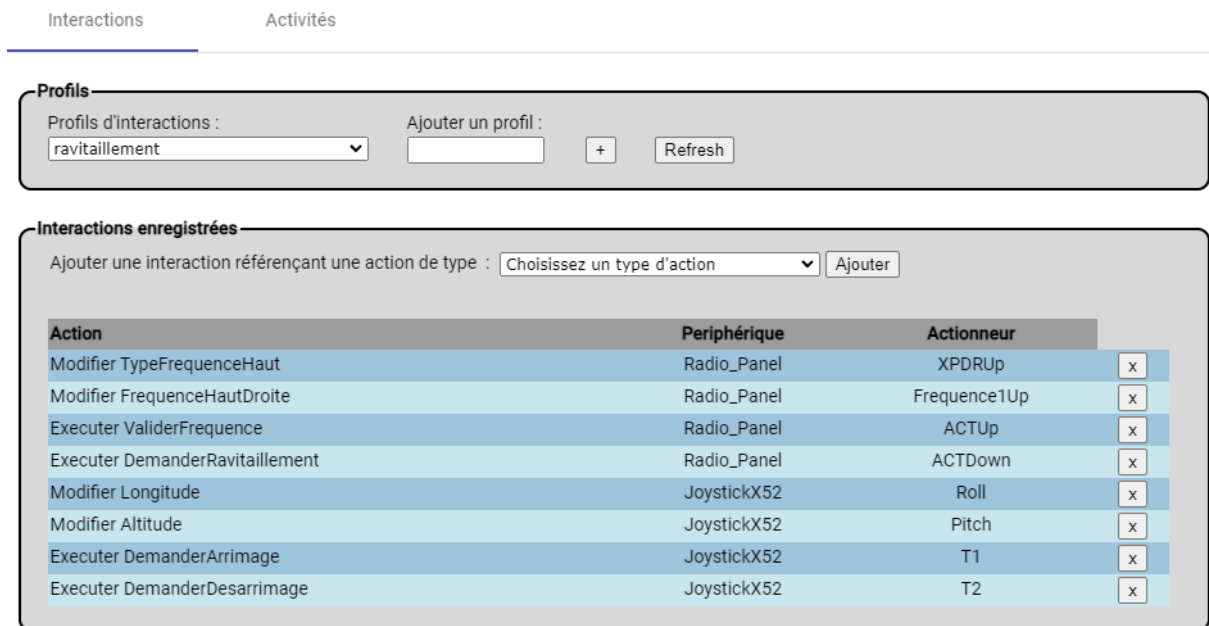


FIGURE 3.4 – Interface dédiée à la spécification des interactions

d'un attribut, etc.). Une fois l'action sélectionnée, l'interface affiche une section dédiée selon le type de l'action.



FIGURE 3.5 – Interface dédiée à la gestion des profils et à la sélection d'un type d'action

2. Une partie dédiée au type d'action sélectionnée. Lorsque l'ergonome choisit un type d'action dans la partie `Profils`, une section dédiée au type d'action sélectionné apparaît. Nous détaillons dans la suite de ce chapitre les différents types d'action que nous avons implémentés ainsi que leurs interfaces respectives.
3. Une partie `Interactions enregistrées` qui recense les actions associées aux périphériques et actionneurs préalablement créés. (cf. Figure 3.6)

Nous présentons par la suite la partie de l'interface dédiée à l'instanciation d'une nouvelle interaction. Cette partie étant générée en fonction du type d'action sélectionnée par l'utilisateur, nous ne présentons que deux types d'actions issues de ceux présentés dans notre chapitre 2 : `CallOperationAction` et `VariableAction`

Interactions enregistrées

Ajouter une interaction référençant une action de type :

| Action | Periphérique | Actionneur | |
|---------------------------------|--------------|--------------|----------------------------------|
| Modifier TypeFrequenceHaut | Radio_Panel | XPDRUp | <input type="button" value="x"/> |
| Modifier FrequenceHautDroite | Radio_Panel | Frequence1Up | <input type="button" value="x"/> |
| Executer ValiderFrequence | Radio_Panel | ACTUp | <input type="button" value="x"/> |
| Executer DemanderRavitaillement | Radio_Panel | ACTDown | <input type="button" value="x"/> |
| Modifier Longitude | JoystickX52 | Roll | <input type="button" value="x"/> |
| Modifier Altitude | JoystickX52 | Pitch | <input type="button" value="x"/> |
| Executer DemanderArrimage | JoystickX52 | T1 | <input type="button" value="x"/> |
| Executer DemanderDesarrimage | JoystickX52 | T2 | <input type="button" value="x"/> |

FIGURE 3.6 – Interface dédiée à la gestion des interactions

Spécification d'une interaction pour exécuter une opération

La partie de l'interface dédiée à l'exécution d'une opération (Exécuter une opération) permet de spécifier une interaction qui référence une opération et un actionneur.

Profils

Profils d'interactions : Ajouter un profil :

Executer une operation

Rôle : Operation : Interacteur : Actionneur :

FIGURE 3.7 – Interface dédiée aux actions de type CallOperationAction

Afin de créer une interaction liant un interacteur et une opération, l'ergonome choisit l'opération à réaliser. Pour cela, il sélectionne avec la première liste déroulante le rôle (Role) qui référence l'opération voulue, puis, avec la seconde liste déroulante, l'opération Operation particulière (cf. Figure 3.8). La liste de ces rôles et opérations est récupérée par introspection du modèle.

L'ergonome choisit ensuite l'interacteur et l'actionneur qui permettront de réaliser l'action. La liste de ces interacteurs et de ces actionneurs est également récupérée par introspection du modèle. Ne sont listés ici que les interacteurs composés d'au moins un actionneur éligible au lancement de l'exécution d'une opération. Nous avons défini que seuls les actionneurs retournant une valeur booléenne, lorsqu'ils sont actionnés,

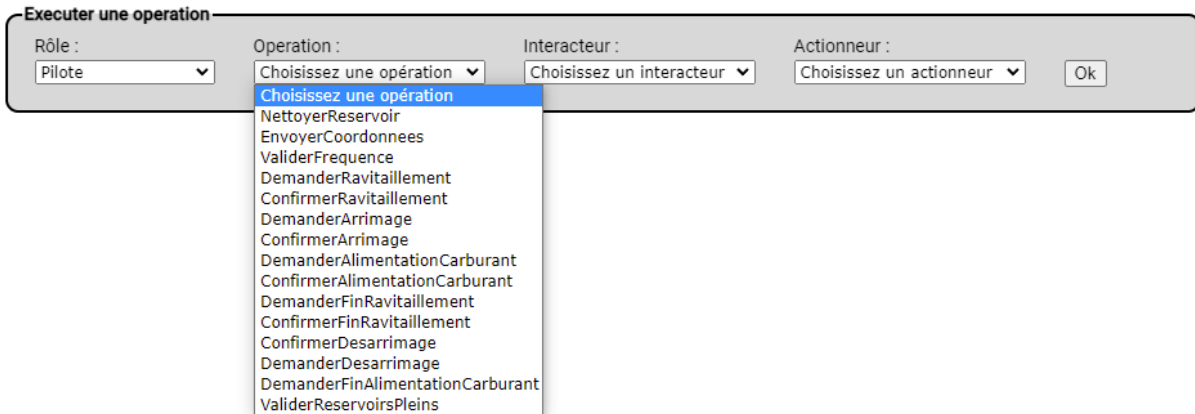


FIGURE 3.8 – Exemple des opérations qu’il est possible de sélectionner lors du choix d’une opération

étaient éligibles à ce cas de spécification d’interaction. Ainsi, seuls les actionneurs de type Button, Gesture et Utterance seront listés dans cette liste déroulante (cf. Figure 2.1, section 2.1) comme illustré dans la Figure 3.9. En effet, ces actionneurs renvoient une valeur booléenne *Vrai* si respectivement un bouton est pressé ou si un geste ou une phrase est reconnu. Les actionneurs de type Axe, Value et Tracker renvoient quant à eux la valeur de l’actionneur (une valeur de type float pour les actionneurs de type Axe, string pour les actionneurs de type Value et Vector3 pour les actionneurs de type Tracker).



FIGURE 3.9 – Exemple d’actionneurs compatibles avec l’exécution d’une opération

Spécification d'une interaction pour modifier la valeur d'un attribut

La partie de l'interface dédiée aux actions de type `VariableAction` (Modifier la valeur d'un attribut) permet de lier la valeur d'un attribut du modèle à la valeur émise par un actionneur (cf. Figure 3.10).

The dialog box titled "Modifier la valeur d'un attribut" contains the following fields:

- Type de ressource : Choisissez un type de ressource
- Ressource : Choisissez une ressource
- Attributs : Choisissez un attribut
- Transformation : Choisissez une action
- Options : Choisissez une opération
- Valeur : (empty text box)
- Interacteur : Choisissez un interacteur
- Actionneur : Choisissez un actionneur
- Ok

FIGURE 3.10 – Interface dédiée aux actions de type `VariableAction`

Dans notre modèle, l'attribut à sélectionner est contenu dans l'entité affectée à une ressource. Une ressource est affectée à une instance de classe représentée dans l'EV et disponible dans PRISME.

L'ergonome choisi dans un premier temps l'attribut dont la valeur doit être modifiée. Pour cela il sélectionne, grâce à la première liste déroulante, le type de ressource contenant l'attribut à modifier puis, l'instance particulière possédant l'attribut à modifier avec la seconde liste déroulante et, enfin, il sélectionne l'attribut qu'il souhaite lier à un actionneur avec la troisième liste déroulante. Chaque attribut est désigné par son nom ainsi que son type (cf. Figure 3.11).

The dialog box titled "Modifier la valeur d'un attribut" shows the following selections:

- Type de ressource : Radio
- Ressource : radioAvion
- Attributs : Choisissez un attribut (dropdown menu open)
- Transformation : Choisissez une action
- Options : Choisissez une opération
- Valeur : (empty text box)
- Interacteur : Choisissez un interacteur
- Actionneur : Choisissez un actionneur
- Ok

The open dropdown menu for "Attributs" lists the following options:

- Choisissez un attribut
- Choisissez un attribut
- FrequenceHautDroite (integer)
- FrequenceHautGauche (integer)
- FrequenceBasDroite (integer)
- FrequenceBasGauche (integer)
- TypeFrequenceHaut (integer)
- TypeFrequenceBas (integer)
- ValidationHaut (boolean)
- ValidationBas (boolean)

FIGURE 3.11 – Exemple de sélection d'un type de ressource, ressource et attribut pour la modification de la valeur d'un attribut

L'ergonome choisit ensuite le type de transformation à appliquer à la valeur de l'attribut. Nous avons défini des règles de modifications couvrant la majorité des besoins selon les cas d'usages testés. Toutefois, ces règles ne sont pas exhaustives et peuvent être adaptées au besoin.

La modification de la valeur d'un attribut consiste premièrement à choisir un type de transformation (cf. Figure 3.12) puis d'y appliquer une option de transformation particulière (cf. Figure 3.13).

FIGURE 3.12 – Exemple de sélection d'une transformation associée à un attribut en cours de modification

Nous avons défini cinq types de transformations que nous présentons ici avec les options de transformation qui leur sont liées :

- **Incrémenter de** : permet d'incrémenter la valeur de l'attribut d'une valeur définie
- **Décrémenter de** : permet de décrémenter la valeur de l'attribut d'une valeur définie
- **Multiplier par** : permet de multiplier la valeur de l'attribut selon un facteur défini
- **Diviser par** : permet de diviser la valeur de l'attribut selon un facteur défini
- **Valeur du contrôleur** : permet d'assigner la valeur du contrôleur à la valeur de l'attribut

Les quatre premiers types de transformations comportent deux options (cf. Figure 3.13).

FIGURE 3.13 – Exemple de sélection d'une option de transformation

La première option consiste à appliquer une valeur fixe comme facteur de transformation (renseignée alors dans la zone de saisie Valeur, cf. Figure 3.13). L'autre option

de transformation consiste à utiliser la valeur que renvoie l'actionneur comme facteur de modification. Ainsi, si l'ergonome décide d'incrémenter la valeur de l'attribut (premier type de transformation) de la valeur du contrôleur, alors à chaque utilisation de l'actionneur, la valeur de l'attribut sera incrémentée de la valeur renvoyée par l'actionneur.

Le cinquième type de transformation (pour rappel : "Valeur du contrôleur") permet d'assigner la valeur renvoyée par le contrôleur, modifiée par une des options de transformation disponible (cf. Figure 3.14).

The image shows a dialog box titled "Modifier la valeur d'un attribut". It contains several dropdown menus and a text input field. The "Options" dropdown is open, showing a list of operations: "Choisissez une opération", "valeur exacte", "incrémenté de", "décrémenté de", "multiplié par", and "divisé par". The "Valeur" field is empty, and there is an "Ok" button.

FIGURE 3.14 – Exemple de sélection d'une option de transformation

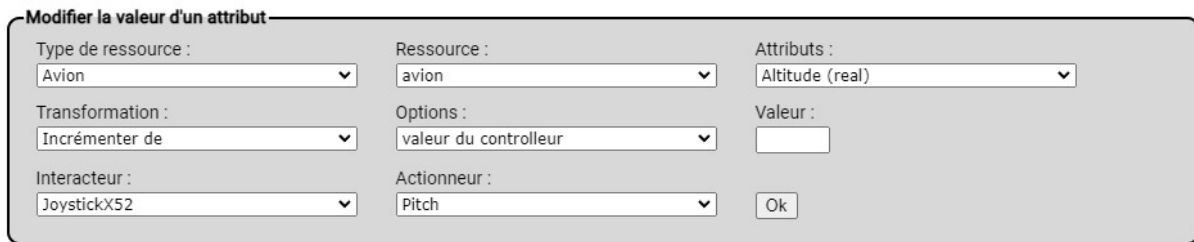
La première option permet d'assigner à la valeur de l'attribut, la valeur renvoyée par l'actionneur, sans autres modifications. Les quatre autres options de transformation modifieront la valeur renvoyée par l'actionneur selon l'option choisie et la valeur renseignée dans le champ de saisie `Valeur` avant de l'assigner à la valeur de l'attribut.

Une fois la transformation de la valeur de l'attribut définie, l'ergonome choisit l'actionneur qu'il souhaite lier à cette interaction. Pour cela, il doit premièrement sélectionner l'interacteur qu'il souhaite lier dans la liste déroulante `Interacteur` puis l'actionneur de son choix dans la liste déroulante `Actionneur`.

Une fois l'interaction validée, une spécification d'interaction est créée, enregistrée dans le profil sélectionné et instanciée. Elle est alors prête à être exploitée dans un EV.

Afin d'illustrer par un exemple ces différentes étapes, nous proposons de revenir sur notre exemple du cockpit d'avion. Dans cet exemple, la valeur de l'attribut `Altitude` de la ressource `avion` sera modifiée selon le type de transformation ("Incrémenter de") et l'option de transformation ("Valeur du contrôleur").

Ainsi, quand l'utilisateur actionne l'axe `Pitch` (c'est-à-dire l'axe vertical) du joystick `JoystickX52`, la valeur de l'attribut `Altitude` de l'avion sera modifiée, comme illustré dans la Figure 3.15.



Modifier la valeur d'un attribut

| | | |
|---------------------|----------------------|-----------------|
| Type de ressource : | Ressource : | Attributs : |
| Avion | avion | Altitude (real) |
| Transformation : | Options : | Valeur : |
| Incrémenter de | valeur du controleur | |
| Interacteur : | Actionneur : | Ok |
| JoystickX52 | Pitch | |

FIGURE 3.15 – Exemple de description d'interaction permettant de modifier l'attribut Altitude de la ressource avion grâce à l'actionneur Pitch de l'interacteur JoystickX52

Dans cet exemple, la valeur de l'attribut Altitude de la ressource avion sera modifiée selon l'interaction de l'utilisateur tout au long de la simulation.

Par cette démarche, nous démontrons ici que PRISME permet à un ergonomiste de lier les interacteurs aux activités d'un système ou d'une plateforme sans l'intervention d'un développeur informatique.

Dans les sections précédentes, nous avons présenté comment un ergonomiste pouvait réaliser la modélisation de l'ensemble des interacteurs nécessaires à la réalisation de ces activités. Cette modélisation, ou plus exactement ces modèles créés à l'aide de PRISME, sont instanciés par l'environnement au moment de son exécution. Les éléments virtuels et tangibles de l'environnement sont alors associés aux instances des éléments des modèles selon leur nature, conformément à la description faite par l'ergonomiste. Ils peuvent alors, durant l'exécution de l'EV, jouer le rôle qu'il leur a été assigné au sein du modèle. Nous avons également présenté comment un ergonomiste pouvait associer les interacteurs préalablement définis aux actions d'une activité. Ces associations permettent de réaliser en temps réel l'activité décrite dans l'EV. Les interfaces utilisateur de PRISME, que nous avons présenté précédemment, permettent cependant de suivre également la réalisation d'une activité. Nous proposons de détailler les concepts de suivi et de création d'activités à l'aide de PRISME dans la section suivante.

3.4 Suivi et création d'une activité à l'aide de PRISME

Durant l'étude d'un système impliquant des activités humaines, tel qu'une plateforme opérateur, l'observation de ces activités réalisées par des experts métiers est

un apport majeur. À ce jour, lorsqu'un ergonome souhaite réaliser un suivi de l'activité, il a le choix de simuler l'activité au sein d'un outil de suivi d'activité tel que ceux présentés en section 1.3, ou de retranscrire cette activité sur tout autre support selon ses observations. Notre proposition, PRISME, permet à un ergonome de réaliser un suivi automatique ou une création de l'activité en temps réel au sein d'un environnement virtuel. Lors du suivi de l'activité, les interactions de l'opérateur réalisant les actions de l'activité sont automatiquement reconnues par PRISME et les effets de ces actions sont automatiquement appliqués au système. Ainsi, un ergonome peut librement observer le comportement de l'opérateur et du système, tandis que PRISME offre un retour visuel du déroulement de cette activité depuis ses interfaces de supervision. Dans la suite de ce chapitre, nous présentons comment un ergonome réalise le suivi ou la création de l'activité depuis les interfaces de supervision de PRISME.

3.4.1 Suivi d'une activité

Dans le cas d'un suivi de l'activité, l'ergonome décrit au préalable une activité à l'aide de MASCARET (via un modèleur UML) et les interactions grâce à notre modèle PRISME. Afin d'illustrer notre propos, nous nous appuyons sur un exemple d'activité et un ensemble d'interactions prédéfinis.

La Figure 3.16 présente les trois premières actions d'une activité complète décrite dans le modèle présenté en annexe B. Dans cet exemple, les actions en attente de réalisation sont visibles dans l'interface que nous avons développée et dédiée au suivi et à la création d'une activité par un ergonome.

L'interacteur `RadioPanel` a été préalablement décrit selon sa configuration (détaillée dans le document présenté en annexe B). Les interactions `Modifier TypeFrequenceHaut`, `Modifier FrequenceHautDroite` et `Executer ValiderFrequence`, illustrées dans la Figure 3.17 ont également été définies à l'aide de PRISME en suivant le processus détaillé dans la section 3.3. Remarquons que dans la Figure 3.17, la première interaction indique qu'une manipulation de l'actionneur `XPDRUp` de l'interacteur `RadioPanel` modifie la valeur de l'attribut `TypeFrequenceHaut` et ainsi de suite pour les autres interactions.

Comme le détaille l'algorithme 1, lors de l'exécution de l'EV, et lorsque l'utilisateur manipule les interacteurs à sa disposition, ceux-ci émettent des événements.

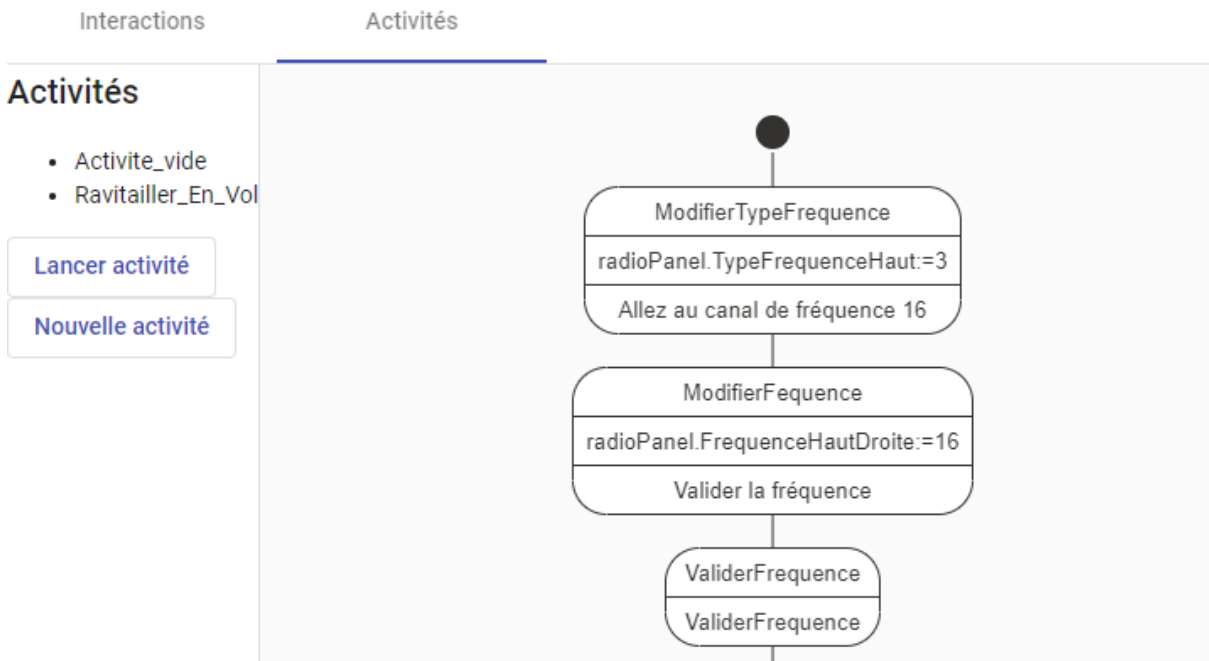


FIGURE 3.16 – Exemple de suivi d'activité

Interactions enregistrées

Ajouter une interaction référençant une action de type :

| Action | Periphérique | Actionneur | |
|---------------------------------|--------------|--------------|---|
| Modifier TypeFrequenceHaut | Radio_Panel | XPDRUp | x |
| Modifier FrequenceHautDroite | Radio_Panel | Frequence1Up | x |
| Executer ValiderFrequence | Radio_Panel | ACTUp | x |
| Executer DemanderRavitaillement | Radio_Panel | ACTDown | x |
| Modifier Longitude | JoystickX52 | Roll | x |
| Modifier Altitude | JoystickX52 | Pitch | x |
| Executer DemanderArrimage | JoystickX52 | T1 | x |
| Executer DemanderDesarrimage | JoystickX52 | T2 | x |

FIGURE 3.17 – Exemples d'interactions enregistrées dans PRISME

Dès qu'un de ces évènements est émis, PRISME récupère par introspection les interactions qui référencent l'actionneur manipulé. Pour chacune de ces interactions, il compare, toujours par introspection, la nature de l'action référencée par cette interaction (`CallOperationAction`, `VariableAction`, *etc.*) à celle attendue dans l'activité. Pour chaque correspondance trouvée, les paramètres de l'action référencée et ceux de l'action attendue sont comparés selon la nature de l'action. Si une correspondance

Algorithm 1: Algorithme de suivi d'activité dans PRISME**Un interacteur émet un évènement****ActAttendue** = Action attendue dans l'activitéPour chaque interaction enregistrée **Inter****Act** = Action référencée dans **Inter**Si **Act** est de même type que **ActAttendue**Si **Act** est de type `CallOperationAction`**Oper** = Operation référencée dans **Act****OperAttendue** = Operation référencée dans **ActAttendue**Si **Oper** = **OperAttendue**Si Paramètres de **Oper** = Paramètres de **OperAttendue****ActAttendue** est réaliséeSi **Act** est de type `VariableAction`Si attribut référencé dans **Act** = attribut référencé dans **ActAttendue**Si Condition de réalisation référencée dans **Act** est remplie**ActAttendue** est réalisée

Mise à jour des KPI

est trouvée, et si les conditions de réalisation de l'action attendue sont remplies, alors l'action est réalisée. Dans tous les cas, les données clés (KPI) sont mises à jour.

Les effets des interactions sur l'activité décrite dépendent du mode (de suivi ou de création) de l'activité définie par l'ergonome avant l'exécution de l'environnement.

Lors du suivi de l'activité, un opérateur réalise une activité précise en interagissant avec des objets tangibles et virtuels au sein d'un EV. PRISME détecte par ces interactions qu'une action a été réalisée et en informe l'ergonome par un retour visuel. À droite sur la figure 3.18, l'opérateur a réalisé l'action `ModifierTypeFrequence`. Prisme a détecté la réalisation de cette action au travers l'interaction de l'opérateur et l'ergonome a pu en parallèle observer l'opérateur réaliser cette interaction.

Toutefois, lors du suivi d'une activité, que ce soit par erreur ou par choix, l'opérateur peut être amené à réaliser une interaction qui n'est pas attendue dans la hiérarchie de l'activité. L'interaction sera prise en compte par l'environnement, mais l'action dans l'activité ne sera pas réalisée. Ainsi, si un interacteur est lié à un attribut particulier d'une instance de ressource, les manipulations réalisées sur cet interacteur en modifieront sa valeur. Les effets de ces modifications sur le système seront appliqués, sans que l'action soit réalisée.

Par exemple, dans le cas décrit à gauche sur la Figure 3.18, si l'utilisateur tente de réaliser la seconde action (`ModifierFrequence`) avant de réaliser la première action

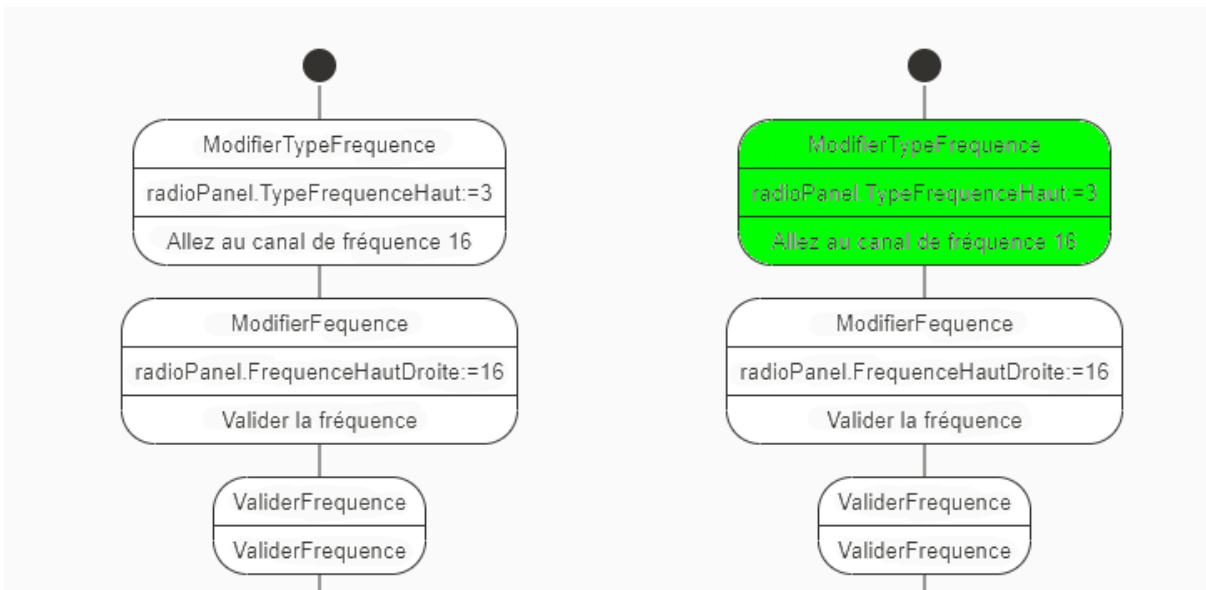


FIGURE 3.18 – Exemple de suivi d’activité. À gauche, une activité nouvellement débutée. À droite, la première action a été réalisée.

(*ModifierTypeFrequence*), l’attribut *FrequenceHautDroite* sera tout de même modifié, mais l’action *ModifierFrequence* ne sera considérée comme réalisée que lorsque l’action *ModifierTypeFrequence* aura été réalisée.

En addition du retour visuel indiquant la réalisation d’une action, PRISME offre la possibilité à l’ergonome de suivre l’évolution en temps réel d’indicateur clés (KPI : Key Performance Indicator) définis tels que le temps pris par l’opérateur pour effectuer une action, le nombre d’essais réalisés par l’opérateur pour modifier la valeur d’un attribut. Si certains de ces indicateurs sont présents dans le modèle, l’ergonome peut vouloir ajouter des indicateurs spécifiques. PRISME ayant une connaissance précise de l’état du système, il est possible de définir un nouvel indicateur basé sur l’état d’un élément du modèle métier instancié et de l’afficher dans l’activité, comme présenté dans la figure 3.19.

3.4.2 Création d’activité

Afin de décrire l’activité, l’ergonome peut soit décrire cette activité à l’aide d’un diagramme d’activité, soit, créer l’activité en enregistrant en temps réel les interactions réalisées dans un EV. Ainsi, un ergonome pourra faire appel à un expert métier afin qu’il réalise une activité particulière et enregistrer cette activité. Celle-ci pourra, par la

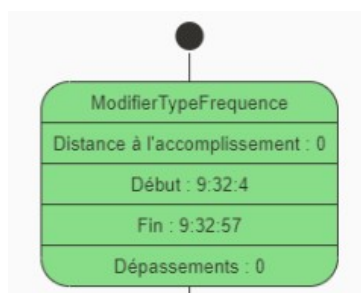


FIGURE 3.19 – Exemple d'action avec visualisation d'indicateurs clés

suite, être utilisée en tant qu'activité prescrite lors de futures études ou être comparée à d'autres activités ainsi enregistrées dans le but d'étudier un comportement plutôt qu'une ergonomie matérielle.

Dans le cas de la création d'une activité, si une manipulation de l'utilisateur remplit les conditions d'exécution d'une action, une action similaire sera créée et ajoutée à l'activité. Ainsi, toute manipulation créera une action marquée comme "réalisée" dans l'activité, selon l'ordre des manipulations réalisées.

3.5 Conclusion

Dans ce chapitre, nous avons présenté comment nous avons instancié notre modèle en nous appuyant sur un cas d'usage aéronautique. Ce cas d'usage illustre la possibilité de réaliser une étude ergonomique d'un cockpit d'avion sans développements spécifiques. Dans cette étude, un ergonome observe le comportement d'un pilote d'avion aux commandes d'un appareil en vol réalisant une procédure de ravitaillement en vol. L'ergonome suis en temps réel le déroulement de l'activité et capture des indicateurs clés (KPI) sur des interfaces de supervision prévues à cet effet.

Après avoir présenté notre plateforme physique et les interacteurs qui la composent (un manche à balai, une manette des gaz, un dispositif radio et un panneau de commutation), nous avons détaillé comment l'ergonome réalisait la description des interacteurs tangibles grâce à notre typologie. Nous avons expliqué ensuite en détail comment l'ergonome, depuis les interfaces de configuration de PRISME, créait les interactions qui associent les actionneurs de ces interacteurs aux actions issues du modèle d'activité décrivant la procédure de ravitaillement en vol.

Enfin, nous avons présenté comment l'ergonome peut réaliser le suivi ou la création

de l'activité du pilote en temps réel.

Dans le chapitre suivant, nous présentons l'évaluation mise en place afin de déterminer l'utilisabilité de notre proposition.

ÉVALUATION DE PRISME PAR DES ERGONOMES POUR DE LA CONCEPTION DE PLATEFORMES

Pour rappel et comme détaillé en introduction de ce manuscrit, le processus de validation d'un système tel qu'une plate-forme opérateur comprend plusieurs étapes [130]. L'une d'elles consiste à simuler le système. Cette simulation permet aux concepteurs (ergonomes) d'observer les activités d'un (ou des) opérateur(s) en situation. Ces observations permettent aux ergonomes de mesurer, par exemple, l'efficacité¹ ou l'efficacité² du système étudié.

Avec notre proposition de modèle, que nous avons appelé PRISME, nous visons à permettre aux ergonomes d'évaluer un système dans son contexte d'utilisation en environnement virtuel et tangible. Plus exactement, nous leur offrons la possibilité de réaliser un suivi des activités, voire de créer des procédures métier, sans développements spécifiques supplémentaires. PRISME s'apparente donc aux outils mis actuellement à disposition des ergonomes dans et pour leurs tâches d'évaluation. Plus concrètement, et pour rappel, nous visons avec ces travaux à répondre aux trois questions de recherche ci-dessous :

- **Q1** : Comment décrire de manière générique les différents types d'interacteurs d'EV et Tangibles qui composeront les IHM simulées ?
- **Q2** : Comment décrire la plate-forme et les activités de l'opérateur ?
- **Q3** : Comment reconnaître automatiquement l'action dans l'activité à partir des interactions utilisateur ?

Dans ce chapitre, nous présentons une expérimentation que nous avons menée

1. Précision et degré d'achèvement avec lesquels l'utilisateur atteint des objectifs spécifiés
2. Rapport entre les ressources utilisées (min) et les résultats obtenus (max)

avec des ergonomes afin d'évaluer l'acceptation de PRISME au sein de leurs activités de conception. Pour cela, nous proposons de mesurer la perception que le participant a de son efficacité et de son efficience lors de l'utilisation de notre modèle. Dans cette expérimentation, les participants/ergonomes ont eu pour objectif principal d'évaluer un système à l'aide de PRISME. Remarquons que dans la suite de ce chapitre, nous considérerons l'expérimentation de notre modèle par l'unique terme *expérimentation* et la tâche expérimentale d'évaluation du système par l'unique terme *évaluation*.

Lors de sa tâche d'évaluation, le participant observe des variables objectives (KPI : *Key Performance Indicator*). Il est à noter que ces observations se rapportent à la tâche expérimentale des participants, et non à l'expérimentation mesurant l'acceptation de PRISME.

Nos observations ont été faites au travers d'une tâche expérimentale réalisée par des participants ayant tous un profil d'ergonome. Cette tâche consiste à réaliser une partie de l'évaluation d'un cockpit d'avion en nous appuyant sur le concept du Magicien d'Oz [2]. Plus concrètement, et dans notre cas, l'opérateur manipulant les interacteurs du cockpit étudié sera un compère et non un pilote certifié. Le système et les modèles de tâche liés ont été décrits lors d'une phase précédente au processus de validation du système. Lors de l'évaluation, le compère immergé dans un EV manipulera les différents interacteurs (en majorité tangibles) de la plate-forme afin de réaliser une procédure métier décrivant une activité de ravitaillement en vol (voir section 4.1.3, Figure C.1). Une interface dédiée et alimentée par notre modèle PRISME (voir section 4.1.2 et Figure 4.2) permettra au participant/ergonome de suivre en temps réel la réalisation de cette procédure ainsi que l'évolution des critères de performances (KPI).

4.1 Méthode / Protocole

4.1.1 Population

Cette étude a été menée avec une population d'ergonomes. Sur dix-sept participants, neuf étaient des hommes et huit des femmes. Quatre étaient étudiants dans un domaine lié à l'ergonomie (niveau Master 2), et treize étaient des ergonomes professionnels.

Les passations ont été effectuées sur deux sites distincts. Cinq ergonomes ont participé à l'étude au sein du laboratoire Human Factors Technologies de l'Institut de

Recherche Technologique b<>com, domicilié au Centre Européen de Réalité Virtuelle à Brest, et douze ergonomes au sein des locaux d'une société partenaire de b<>com, Human Design Group, spécialisée dans le domaine de l'ergonomie, située à Toulouse.

4.1.2 Matériel

Dans cette section, nous détaillons les interacteurs constitutifs de la plate-forme utilisée lors de l'expérimentation, le modèle de tâches décrivant l'activité de ravitaillement en vol (activité réalisée par un compère, pour rappel) ainsi que les interfaces de supervision permettant aux ergonomes de suivre en temps réel l'activité réalisée par le compère.

Le contenu expérimental dépeint une situation d'étude d'un cockpit d'avion. La plate-forme physique représentant ce cockpit est simulée dans un EV. Les participants avaient accès à deux interfaces de supervision. La première est une interface de visualisation et d'édition des liaisons entre les actions du modèle de tâche et les interacteurs (voir Figure 4.2). La seconde interface permet de visualiser en temps réel d'une part le suivi de l'activité sous la forme de diagrammes d'activité, et d'autre part l'évolution des critères de performance liées à l'activité réalisée par le compère (voir Figure 4.7). Ces deux interfaces ont été décrites en détails dans le chapitre 3.3. Toutefois, nous proposons une description plus courte de ces interfaces plus loin dans ce chapitre afin de faciliter la compréhension de la tâche de l'ergonome lors de l'étude.

Description de la plate-forme opérateur

Dans notre cas d'usage, le système à évaluer est une plate-forme représentant un cockpit d'avion, simulé en EV. Cette plateforme est composée de cinq interacteurs tangibles dont quatre possèdent leurs jumeaux numériques représentés dans l'EV (voir Figure 4.1) :

1. **Un dispositif de contrôle radio** : Voir annexe A, 1
2. **Un dispositif de contrôle switch** : Voir annexe A, 2
3. **Une manette de jeu Logitech X52 PRO** : Voir annexe A, 3
4. **Une manette de gaz Logitech X52 PRO** : Voir annexe A, 4
5. **Un clavier standard** : Voir annexe A, 5



FIGURE 4.1 – Plate-forme opérateur composée des quatre interacteurs

Description des interfaces de supervision dédiées aux ergonomes

Pour suivre l'exécution des activités (dans notre cas, la procédure de ravitaillement en vol), les participants ont eu accès à deux interfaces de supervisions. Ces interfaces sont générées et alimentées par PRISME en temps réel.

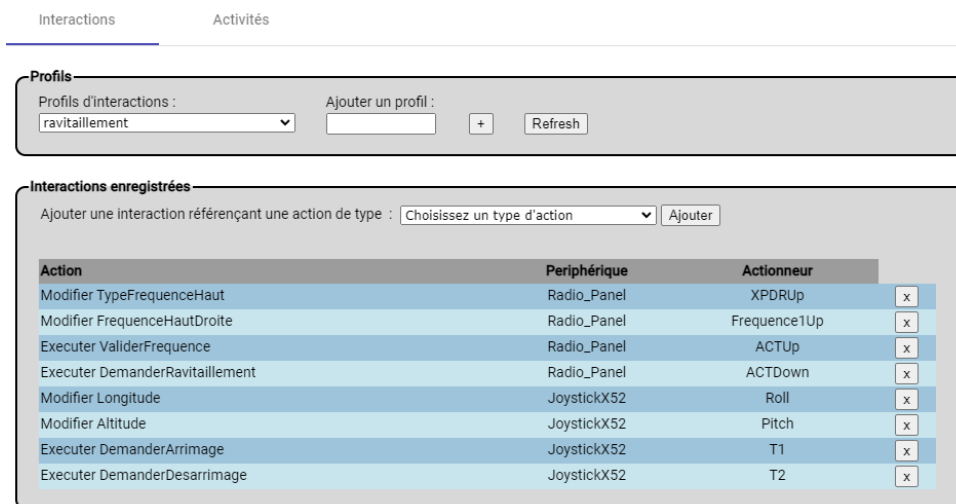


FIGURE 4.2 – Interface permettant de lier les interacteurs aux actions issues du modèle d'activité

La première interface permet de décrire les liens entre les interacteurs et les actions du modèle de tâche (voir Figure 4.2). Pour rappel, elle se découpe en quatre parties :

- **Profils** permet d'enregistrer des profils d'interactions. Cela consiste à pouvoir facilement enregistrer et rappeler une configuration d'interactions particulière (voir Figure 4.3).

FIGURE 4.3 – Interface permettant d'enregistrer et de charger un ensemble d'interactions

- **Exécuter une opération** permet de lier un interacteur à une action dont la réalisation lance l'exécution d'une opération (voir Figure 4.4).

FIGURE 4.4 – Interface permettant de définir une interaction réalisant une action de type `CallOperationAction`

- **Modifier la valeur d'un attribut** (voir Figure 4.5) permet de modifier la valeur de l'attribut d'une instance de classe particulière, selon le type de l'attribut et la nature de l'interacteur. Par exemple, un attribut de type booléen pourra être modifié avec un actionneur de type `[bouton]` et un attribut de type entier ou réel pourra être modifié avec un actionneur de type `[axe]` (voir chapitre 2.1, Figure 2.1).

FIGURE 4.5 – Interface permettant de définir une interaction réalisant une action de type `VariableAction`

- **Interactions enregistrées** permet d'ajouter une nouvelle interaction au profil sélectionné, et également de visualiser les liaisons entre les interacteurs et les actions issues du modèle d'activité, enregistrées dans ce même profil. À ce jour,

il est possible de supprimer une interaction, mais pas de la modifier (voir Figure 4.6).

Interactions enregistrées

Ajouter une interaction référençant une action de type :

| Action | Périphérique | Actionneur | |
|---------------------------------|--------------|--------------|----------------------------------|
| Modifier TypeFrequenceHaut | Radio_Panel | XPDRUp | <input type="button" value="x"/> |
| Modifier FrequenceHautDroite | Radio_Panel | Frequence1Up | <input type="button" value="x"/> |
| Executer ValiderFrequence | Radio_Panel | ACTUp | <input type="button" value="x"/> |
| Executer DemanderRavitaillement | Radio_Panel | ACTDown | <input type="button" value="x"/> |
| Modifier Longitude | JoystickX52 | Roll | <input type="button" value="x"/> |
| Modifier Altitude | JoystickX52 | Pitch | <input type="button" value="x"/> |
| Executer DemanderArrimage | JoystickX52 | T1 | <input type="button" value="x"/> |
| Executer DemanderDesarrimage | JoystickX52 | T2 | <input type="button" value="x"/> |

FIGURE 4.6 – Interface listant les interactions enregistrées dans un profil particulier

La seconde interface permet de suivre en temps réel la réalisation des activités (dans notre cas le ravitaillement de l'avion) au sein de l'EV, et de visualiser l'évolution des indicateurs clés de performance, KPI (voir Figure 4.7). Globalement, les indicateurs clés sont :

- **Le temps de réalisation de l'action** qui correspond au temps entre le début de l'interaction nécessaire à la réalisation de l'action et la fin de la réalisation de celle-ci.
- **Le temps pour atteindre les coordonnées de ravitaillement** qui correspond au temps que le compère va mettre pour atteindre les coordonnées du lieu de rendez-vous avec l'appareil ravitailleur.
- **Le nombre de dépassements** qui correspond au nombre de dépassements positifs ou négatifs effectués par le compère lorsque celui-ci tente d'atteindre une valeur spécifique avec un interacteur.
- **La distance à l'accomplissement** qui correspond à la valeur à atteindre soustraite de valeur actuelle.
- **L'enchaînement des actions réalisées**

Enfin, sur cette interface et plus exactement dans le diagramme d'activité, une action peut prendre différentes couleurs selon son état :

- **Blanc** : lorsque l'action n'a pas encore été réalisée
- **Bleu** : lorsque l'action est en cours de réalisation
- **Vert** : lorsque l'action est réalisée

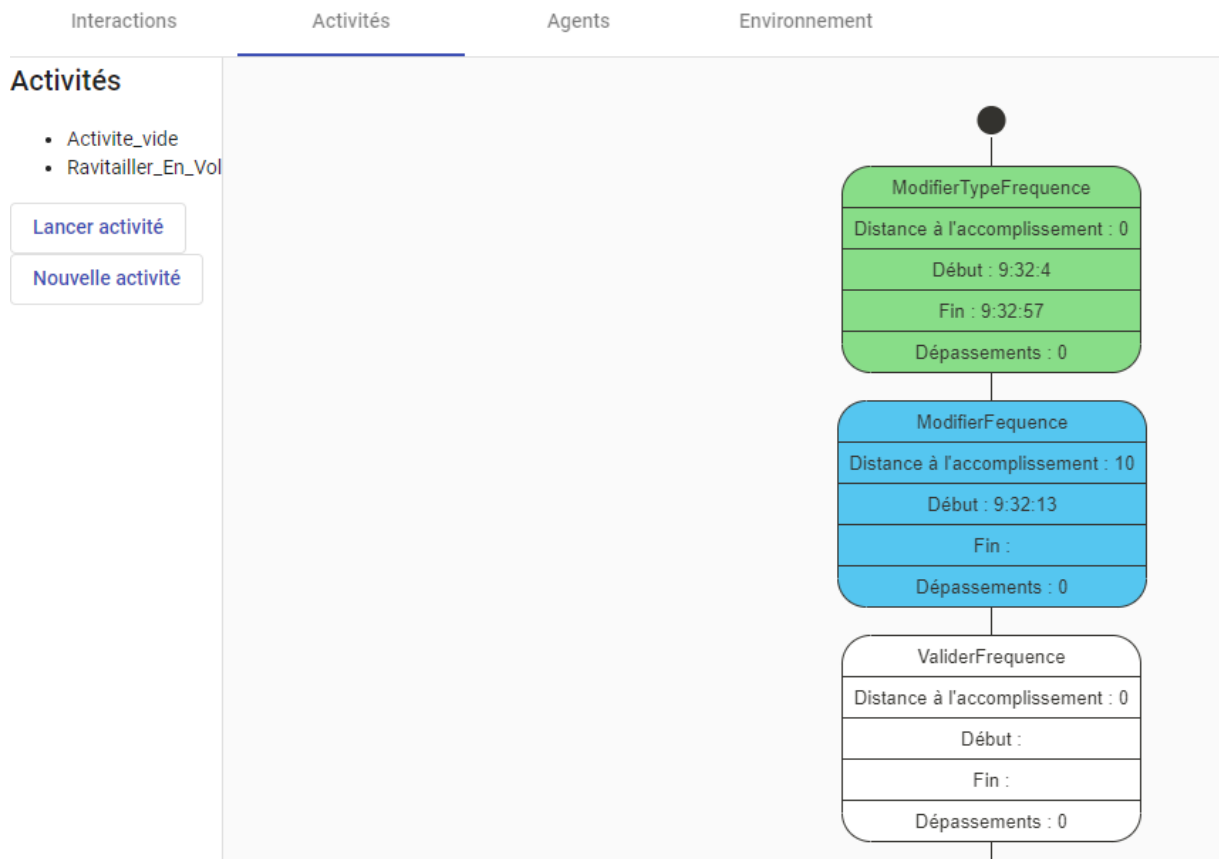


FIGURE 4.7 – Interface permettant de visualiser le suivi de l'activité et des KPI en temps réel. Ici, l'action `ModifierTypeFrequence` a été réalisée, l'action `ModifierFrequence` est en cours de réalisation et l'action `ValiderFrequence` n'a pas encore débuté.

4.1.3 protocole

Nous présentons dans cette section notre protocole d'évaluation ainsi que les données analysées par le participant durant la tâche expérimentale.

La procédure expérimentale s'est déroulée en quatre phases. La première phase était une phase de familiarisation où le système et le contexte de l'étude était présenté

au participant. Durant la seconde phase, les participants ont observé une première exécution de la procédure métier en suivant l'évolution de celle-ci sur les interfaces de supervision. Ils ont également observé l'évolution des indicateurs clés (KPI) automatiquement générés par PRISME. Au cours de la troisième phase, il a été demandé aux participants de modifier au préalable des interactions permettant la réalisation de deux actions spécifiques : `ModifierFrequence` et `ValiderFrequence`. Les participants ont ensuite observé sur les interfaces de supervision les effets de ces modifications sur l'exécution de la procédure métier. Durant la quatrième et dernière phase, les participants ont rempli un questionnaire TAM, permettant d'évaluer l'acceptation de notre modèle dans le cadre de leur tâche d'évaluation.

Nous proposons de détailler plus précisément les éléments de ces quatre phases dans les sous-sections ci-dessous.

Phase n°1

Lors de cette première phase, il s'agissait dans un premier temps de présenter aux participants le contexte de l'évaluation, dans un second temps la tâche expérimentale qu'ils auraient à accomplir, le rôle qu'ils détiendraient durant cette tâche. Pour cela, les participants devaient prendre connaissance d'un document explicatif (voir annexe B), sous la supervision de l'expérimentateur. Ce document présente les informations contenues dans les modèles UML décrivant la plate-forme physique, les interacteurs qui la composent ainsi que l'activité de ravitaillement en vol. Plus précisément, ce document expose :

- Les rôles acteurs de la procédure, leur nature, et quelles sont leurs capacités, c'est-à-dire les opérations qu'ils peuvent exécuter,
- Les types d'actions constituant la procédure. Nous en avons implémenté deux (voir chapitre 3) : les actions permettant d'exécuter des opérations et les actions permettant de modifier des valeurs d'attributs,
- Les ressources de la procédure, leur sémantique et leurs attributs,
- Les interacteurs présents sur la plate-forme physique ainsi que leur composition.

Enfin, nous avons présenté aux participants les deux interfaces de supervision qui leur permettraient de réaliser l'étude du système.

Phase n°2

Durant la seconde phase de l'expérimentation, les participants étaient amenés à observer et suivre le compère réaliser la procédure de ravitaillement en vol. Pour cela, ils pouvaient suivre en temps réel l'exécution de l'activité et l'évolution des indicateurs clés (KPI) grâce aux deux interfaces de supervision présentées précédemment. À l'issue de cette phase, les participants ont reporté sur une grille d'observation les indicateurs clés qu'ils ont pu observer durant le suivi de la procédure métier afin de pouvoir les comparer ensuite aux résultats issus de la phase 3.

Phase n°3

La troisième phase de l'évaluation est composée de deux étapes. La première consiste à modifier les liens entre les actions du modèle métier et les interacteurs présents durant la réalisation de la procédure métier. Pour cela, les participants ont utilisé l'interface prévue à cet effet (voir Figure 4.2). Il leur a été demandé de modifier les interactions réalisant les actions `ModifierFrequence` (touche W du clavier pour diminuer et X du clavier pour augmenter la fréquence) et `ValiderFrequence` (touche C du clavier) afin qu'elles soient liées aux actionneurs du dispositif de gestion radio (actionneur `Frequence1Up` pour l'action `ModifierFréquence` et `ACTUp` pour l'action `ValiderFrequence`). Durant la seconde étape de cette phase, les participants ont une nouvelle fois suivi en temps réel l'exécution de l'activité de ravitaillement en vol ainsi que l'évolution des indicateurs clés sur les interfaces de supervision. À l'issue de cette étape, les participants ont reporté sur une nouvelle grille d'observation les indicateurs clés qu'ils ont pu observer durant le suivi de la procédure métier. Nous avons ensuite demandé aux participants de comparer les données reportées sur les deux grilles d'observation, et de nous indiquer s'ils remarquaient une évolution de la performance du compère concernant la réalisation de la partie de l'activité dédiée à la modification de la fréquence radio. Notre but étant de vérifier que notre proposition permet bien à un ergonome de constater les impacts des modifications réalisées sur la plateforme ou l'activité entre deux évaluations d'un même système.

Phase n°4

Durant cette phase, il a été demandé aux participants de renseigner un questionnaire TAM (voir Annexe Figure D). Le Modèle d'Acceptation des Technologies (TAM, Technology Acceptance Model) [27][28] est un modèle prédominant pour évaluer l'acceptation des Technologies de l'Information et de la Communication. Selon Davis [27][28], l'acceptation par les utilisateurs d'une nouvelle technologie ou d'un nouveau dispositif est mesurée selon deux échèles : l'Utilité Perçue (UP) et la Facilité d'Utilisation Perçue (FUP). L'Utilité Perçue représente "*le degré selon lequel une personne pense qu'utiliser un système particulier améliorerait ces performances*" et la Facilité d'Utilisation Perçue représente "*le degré selon lequel une personne pense qu'utiliser un système particulier se ferait sans effort*". Nous nous sommes donc appuyés sur ce modèle afin d'évaluer l'acceptation de PRISME au sein du métier de l'ergonome. Notons que nous avons choisi de préserver la langue d'origine du questionnaire (anglais). Toutefois, afin de faciliter la compréhension des questions, nous avons décidé d'ajouter une traduction française sous la version anglaise de chaque question.

4.2 Résultats

Nous présentons dans cette section l'analyse des résultats issus du questionnaire TAM [27][28] rempli par les participants lors de la phase n°4 de l'expérimentation. Pour rappel, les questions du TAM sont réparties dans deux sections distinctes. La première section porte sur la facilité d'utilisation perçue (FUP), et la seconde section sur l'utilité perçue (UP). Par la suite, nous présentons l'analyse de la pertinence de l'utilisation de PRISME lors de la phase de design d'un système par des ergonomes. Pour cette analyse, nous avons considéré les réponses allant de 1 à 3 (*Extremely likely*, *Quite Likely* et *Slightly likely*) comme positives, les réponses de valeur 4 (*Neither likely or unlikely*) comme neutres et les réponses allant de 5 à 7 (*Extremely unlikely*, *Quite unlikely* et *Slightly unlikely*) comme négatives.

4.2.1 Facilité d'Utilisation Perçue (FUP) de PRISME

La Facilité d'Utilisation Perçue (FUP) du questionnaire TAM concerne les questions Q1 à Q10. Elles visent à mesurer la perception de l'utilisateur sur sa facilité à utiliser PRISME. Nous résumons les réponses à ces questions sur le tableau 4.1.

| Question | Cardinalité | | | | | | | Médiane | Moy. | Écart-Type | Tx de rép. pos. |
|------------|-------------|---|---|---|---|---|---|---------|------|------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | |
| Q1 | 7 | 6 | 2 | 1 | 1 | 0 | 0 | 2 | 2,00 | 1,17 | 88% |
| Q2 | 5 | 9 | 3 | 0 | 0 | 0 | 0 | 2 | 1,88 | 0,70 | 100% |
| Q3 | 6 | 9 | 2 | 0 | 0 | 0 | 0 | 2 | 1,76 | 0,66 | 100% |
| Q4 | 6 | 6 | 3 | 1 | 1 | 0 | 0 | 2 | 2,12 | 1,17 | 88% |
| Q5 | 5 | 8 | 3 | 1 | 0 | 0 | 0 | 2 | 2,00 | 0,87 | 94% |
| Q6 | 2 | 9 | 3 | 2 | 1 | 0 | 0 | 2 | 2,47 | 1,07 | 82% |
| Q7 | 4 | 3 | 3 | 1 | 1 | 3 | 2 | 3 | 3,53 | 2,21 | 59% |
| Q8 | 4 | 6 | 3 | 3 | 1 | 0 | 0 | 2 | 2,47 | 1,23 | 76% |
| Q9 | 4 | 6 | 5 | 2 | 0 | 0 | 0 | 2 | 2,29 | 0,99 | 88% |
| Q10 | 4 | 4 | 3 | 3 | 2 | 1 | 0 | 3 | 2,88 | 1,58 | 65% |
| FUP | | | | | | | | 2,2 | 2,34 | 1,16 | 84% |

TABLE 4.1 – Résultats aux questions portant sur la Facilité d'Utilisation Perçue (FUP)

Question Q1

La perception de facilité d'utilisation de PRISME (voir Figure 4.8) a été évalué positivement par 88% des participants (15 sur 17), de manière neutre par un participant, et négativement par un participant. La réponse moyenne des participants pour cette question est de 2 avec un écart type de 1.17 (voir tableau 4.1).

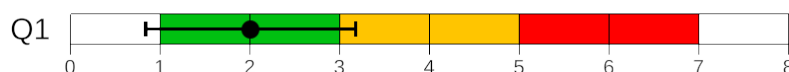


FIGURE 4.8 – Réponse moyenne et écart-type pour la question n°1
Je pense qu'utiliser PRISME serait facile pour moi

Question Q2

La totalité des participants ont estimé que leurs interactions avec PRISME serait claire et compréhensive (voir Figure 4.9). La réponse moyenne des participants pour

cette question est de 1.88 avec un écart type de 0.7 (voir tableau 4.1).



FIGURE 4.9 – Réponse moyenne et écart-type pour la question n°2
Je pense que mon interaction avec PRISME serait claire et compréhensible

Question Q3

La totalité des participants ont estimé qu'il leur serait facile de devenir habile à utiliser PRISME (voir Figure 4.10). La réponse moyenne des participants pour cette question est de 1.76 avec un écart type de 0.66 (voir tableau 4.1).



FIGURE 4.10 – Réponse moyenne et écart-type pour la question n°3
Je pense qu'il serait facile de devenir habile à utiliser PRISME

Question Q4

88% des participants (15/17) ont estimés que PRISME serait adaptable dans ses interactions (voir Figure 4.11). Un participant a donné une réponse neutre et un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.12 avec un écart type de 1.17 (voir tableau 4.1).

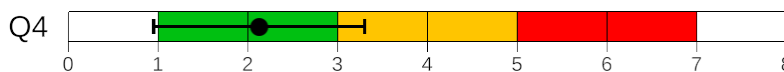


FIGURE 4.11 – Réponse moyenne et écart-type pour la question n°4
Je trouverais PRISME flexible pour interagir avec

Question Q5

94% des participants (16/17) ont estimés qu'ils leur seraient facile d'apprendre à utiliser PRISME (voir Figure 4.12). Un participant a donné une réponse neutre. La

réponse moyenne des participants pour cette question est de 2 avec un écart type de 0.87 (voir tableau 4.1).

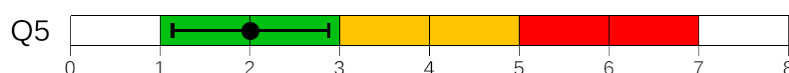


FIGURE 4.12 – Réponse moyenne et écart-type pour la question n°5
Apprendre à utiliser PRISME serait facile pour moi

Question Q6

82% des participants (14/17) ont estimés qu'ils leur seraient facile d'amener PRISME à réaliser ce qu'ils souhaitent (voir Figure 4.13). Deux participants ont donné une réponse neutre et un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.47 avec un écart type de 1.07 (voir tableau 4.1).

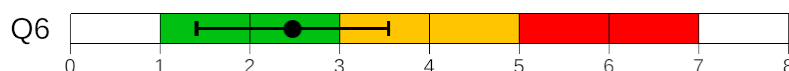


FIGURE 4.13 – Réponse moyenne et écart-type pour la question n°6
Ce serait facile pour moi d'amener PRISME à faire ce que je veux faire

Question Q7

59% des participants (10/17) ont estimé que leur expérience limitait leur capacité à déterminer la facilité d'utilisation de PRISME. Un participant a donné une réponse neutre et six participants ont donné une réponse négative. La réponse moyenne des participants pour cette question est de 3.53 avec un écart type de 2.21 (voir tableau 4.1).

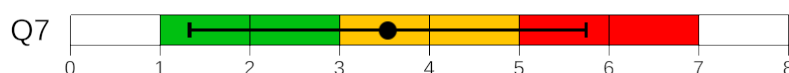


FIGURE 4.14 – Réponse moyenne et écart-type pour la question n°7
Je pense que ma capacité à déterminer la facilité d'utilisation de PRISME est limitée par mon manque d'expérience

Question Q8

76% des participants (13/17) ont estimé que PRISME leur permettrait d'exécuter leurs tâches plus rapidement (voir Figure 4.15). Trois participants ont donné une réponse neutre et un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.47 avec un écart type de 1.23 (voir tableau 4.1).

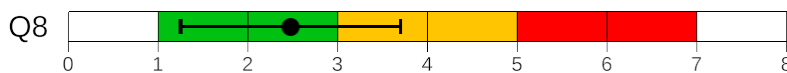


FIGURE 4.15 – Réponse moyenne et écart-type pour la question n°8
Utiliser PRISME dans mon travail me permettrait d'accomplir des tâches plus rapidement

Question Q9

88% des participants (15/17) ont estimé que PRISME améliorerait leurs performances professionnelles (voir Figure 4.16). Deux participants ont donné une réponse neutre. La réponse moyenne des participants pour cette question est de 2.29 avec un écart type de 0.99 (voir tableau 4.1).



FIGURE 4.16 – Réponse moyenne et écart-type pour la question n°9
L'utilisation de PRISME améliorerait mes performances professionnelles

Question Q10

65% des participants (11/17) ont estimé que PRISME augmenterait leur productivité. Trois participants ont donné une réponse neutre et trois participants ont donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.88 avec un écart type de 1.58 (voir tableau 4.1).

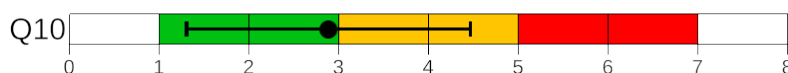


FIGURE 4.17 – Réponse moyenne et écart-type pour la question n°10
Utiliser PRISME dans mon travail augmenterait ma productivité

4.2.2 Utilité Perçue (UP) de PRISME

Cette partie du questionnaire TAM concerne les questions Q11 à Q20. Elle mesure l'utilité perçue de PRISME par les utilisateurs.

| Question | Cardinalité | | | | | | | Médiane | Moy. | Écart-Type | Tx de rép. pos. |
|-----------|-------------|----|---|---|---|---|---|---------|------|------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | |
| Q11 | 4 | 6 | 3 | 2 | 2 | 0 | 0 | 2 | 2,53 | 1,33 | 68% |
| Q12 | 5 | 3 | 7 | 1 | 1 | 0 | 0 | 3 | 2,41 | 1,18 | 79% |
| Q13 | 6 | 5 | 6 | 0 | 0 | 0 | 0 | 2 | 2,00 | 0,87 | 89% |
| Q14 | 7 | 10 | 0 | 0 | 0 | 0 | 0 | 2 | 1,59 | 0,51 | 89% |
| Q15 | 7 | 8 | 2 | 0 | 0 | 0 | 0 | 2 | 1,71 | 0,69 | 89% |
| Q16 | 8 | 7 | 1 | 1 | 0 | 0 | 0 | 2 | 1,71 | 0,85 | 84% |
| Q17 | 5 | 3 | 4 | 4 | 0 | 1 | 0 | 3 | 2,65 | 1,46 | 63% |
| Q18 | 5 | 5 | 6 | 0 | 0 | 1 | 0 | 2 | 2,29 | 1,26 | 84% |
| Q19 | 4 | 3 | 4 | 4 | 1 | 1 | 0 | 3 | 2,88 | 1,50 | 58% |
| Q20 | 5 | 7 | 4 | 1 | 0 | 0 | 0 | 2 | 2,06 | 0,90 | 84% |
| UP | | | | | | | | 2,3 | 2,18 | 1,05 | 88% |

TABLE 4.2 – Résultats aux questions portant sur l'Utilité Perçue (UP)

Question Q11

76% des participants (13/17) ont estimé que PRISME augmenterait leur efficacité (voir Figure 4.18). Deux participants ont donné une réponse neutre et deux participants ont donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.53 avec un écart type de 1.33 (voir tableau 4.2).

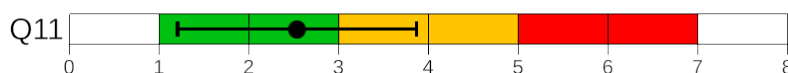


FIGURE 4.18 – Réponse moyenne et écart-type pour la question n°11
L'utilisation de PRISME améliorerait mon efficacité au travail

Question Q12

88% des participants (15/17) ont estimé que PRISME faciliterait leur travail (voir Figure 4.19). Un participant a donné une réponse neutre et un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.41 avec un écart type de 1.18 (voir tableau 4.2).

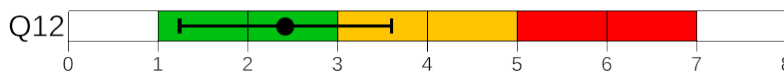


FIGURE 4.19 – Réponse moyenne et écart-type pour la question n°12
Utiliser PRISME faciliterait mon travail

Question Q13

La totalité des participants ont estimé que PRISME serait utile dans leur travail (voir Figure 4.20). La réponse moyenne des participants pour cette question est de 2 avec un écart type de 0.87 (voir tableau 4.2).

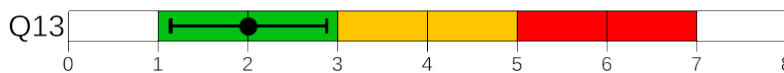


FIGURE 4.20 – Réponse moyenne et écart-type pour la question n°13
Je trouverai PRISME utile dans mon travail

Question Q14

La totalité des participants ont estimé que l'utilisation de PRISME est une bonne idée (voir Figure 4.21). La réponse moyenne des participants pour cette question est de 1.59 avec un écart type de 0.51 (voir tableau 4.2).

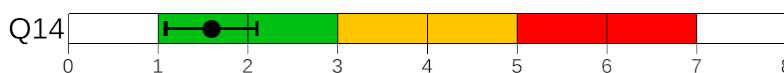


FIGURE 4.21 – Réponse moyenne et écart-type pour la question n°14
Je pense que c'est une bonne idée d'utiliser PRISME

Question Q15

La totalité des participants ont aimé l'idée d'utiliser PRISME (voir Figure 4.22). La réponse moyenne des participants pour cette question est de 1.71 avec un écart type de 0.69 (voir tableau 4.2).

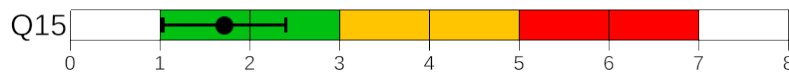


FIGURE 4.22 – Réponse moyenne et écart-type pour la question n°15
J'aime l'idée d'utiliser PRISME

Question Q16

94% des participants (16/17) estiment que l'idée d'utiliser PRISME est une bonne idée (voir Figure 4.23). Un participant a donné une réponse neutre. La réponse moyenne des participants pour cette question est de 1.71 avec un écart type de 0.85 (voir tableau 4.2).

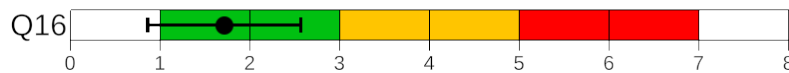


FIGURE 4.23 – Réponse moyenne et écart-type pour la question n°16
Utiliser PRISME est une bonne idée

Question Q17

71% des participants (12/17) prévoient d'utiliser PRISME dans l'avenir (voir Figure 4.24). Quatre participants ont donné une réponse neutre et un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.65 avec un écart type de 1.46 (voir tableau 4.2).

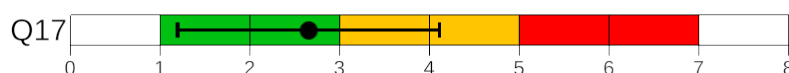


FIGURE 4.24 – Réponse moyenne et écart-type pour la question n°17
Je prévois d'utiliser PRISME à l'avenir

Question Q18

94% des participants (16/17) ont l'intention d'utiliser PRISME s'ils en ont accès (voir Figure 4.25). Un participant a donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.29 avec un écart type de 1.26 (voir tableau 4.2).

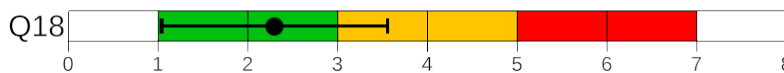


FIGURE 4.25 – Réponse moyenne et écart-type pour la question n°18
En supposant que j'aie accès à PRISME, j'ai l'intention de l'utiliser

Question Q19

65% des participants (11/17) ont estimé que l'utilisation de PRISME est importante (voir Figure 4.26). Quatre participants ont donné une réponse neutre et deux participants ont donné une réponse négative. La réponse moyenne des participants pour cette question est de 2.88 avec un écart type de 1.5 (voir tableau 4.2).

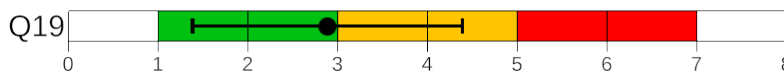


FIGURE 4.26 – Réponse moyenne et écart-type pour la question n°19
Dans mon travail, l'utilisation de PRISME est importante

Question Q20

94% des participants (16/17) ont estimé l'utilisation de PRISME pertinente (voir Figure 4.27). Un participant a donné une réponse neutre. La réponse moyenne des participants pour cette question est de 2.88 avec un écart type de 0.9 (voir tableau 4.2).

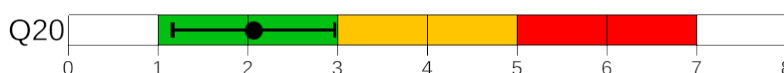


FIGURE 4.27 – Réponse moyenne et écart-type pour la question n°20
Dans mon métier, l'usage de PRISME est pertinent

4.2.3 Alignement des résultats TAM sur l'échelle SUS

Nous proposons d'aligner les résultats obtenus aux questionnaires TAM selon les métriques liées aux échelles SUS. En effet, l'échelle SUS (System Usability Scale) a été proposée par Brooke[18] afin d'évaluer rapidement l'utilité d'un système. Ce système de notation permet d'assigner un score compris entre 0 et 100, 100 étant la note maximale, permettant de considérer le système évalué comme parfait. Cette notation subjective permet donc une représentation plus évocatrice qu'une donnée numérique. Ainsi, en plus d'analyser l'Utilité Perçue et la Facilité d'Utilisation Perçue, Lewis[81] propose trois formules permettant d'aligner les résultats d'un TAM selon les métriques liées aux échelles SUS.

Nous avons ainsi calculé deux valeurs : $PRISME_{FUP}$ et $PRISME_{UP}$, qui sont ensuite agrégées afin d'obtenir une note globale : $PRISME_{TAM}$ (cf. Équation 4.28, 4.29 et 4.30).

$$PRISME_{FUP} = 100 - (MOY(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10) - 1) * (100/6)$$

FIGURE 4.28 – Formule de calcul de la variable $PRISME_{FUP}$

$$PRISME_{UP} = 100 - (MOY(Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20) - 1) * (100/6)$$

FIGURE 4.29 – Formule de calcul de la variable $PRISME_{UP}$

$$PRISME_{TAM} = MOY(PRISME_{FUP}, PRISME_{UP})$$

FIGURE 4.30 – Formule de calcul de la variable $PRISME_{TAM}$

Nous obtenons ainsi : $PRISME_{FUP} = 77.65$ et $PRISME_{UP} = 80.29$. La moyenne de ces valeurs, et donc le score global $PRISME_{TAM}$ est de 78.97, ce qui place PRISME dans la plage Acceptable de l'échelle SUS avec une note de C (voir Figure 4.31).

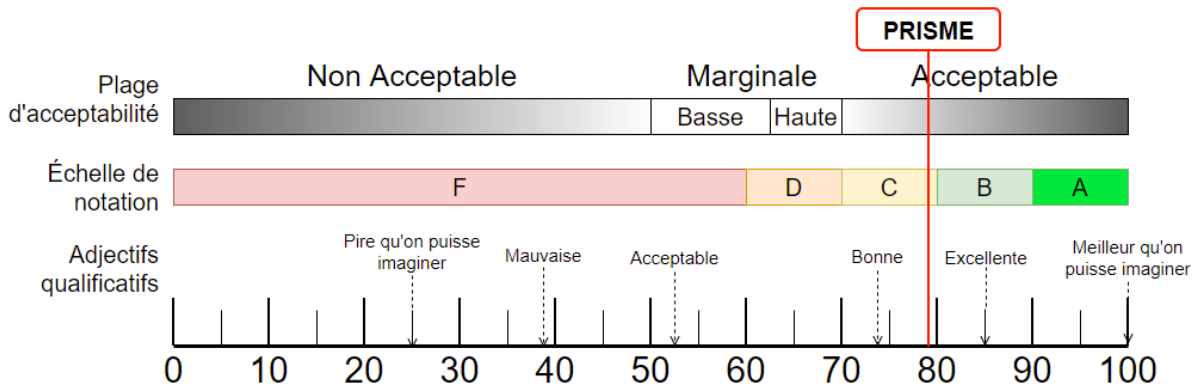


FIGURE 4.31 – Positionnement de PRISME sur l'échelle SUS selon les formules d'alignement entre TAM et SUS proposées par Lewis[81]

4.3 Discussion

4.3.1 Acceptation de PRISME

L'évaluation de l'Utilité Perçue (UP) de notre proposition a donné des résultats majoritairement positifs. 80% des participants ont perçu notre proposition comme étant utile. Notons cependant que l'importance de l'utilisation de PRISME et la prévision de l'utilisation de PRISME à l'avenir sont les variables pour lesquelles les résultats d'évaluation sont les plus faibles. Cela peut être dû au fait que les participants n'avaient pas tous une expérience élevée dans le domaine de l'évaluation ergonomique, certains étant encore étudiants. Le contexte d'utilisation de notre proposition mettait en scène une évaluation ergonomique par l'observation d'un utilisateur réalisant une procédure métier particulière. De ce fait, le participant était immergé dans un rôle d'évaluateur, sans doute peu connu pour un étudiant. Concernant la comparaison avec d'autres systèmes, nous n'avons pas fourni aux participants d'autres solutions à évaluer, certains n'ont pas pu se souvenir d'un système équivalent qu'ils auraient utilisé précédemment (P6).

La Facilité d'Utilisation Perçue (FUP) a également été évaluée très positivement par les participants de l'évaluation, 79% des participants ont jugé que leurs interactions avec notre proposition étaient claires et compréhensives et qu'il leur serait facile de devenir habile à son utilisation. Ils ont également considéré qu'il serait simple d'apprendre à utiliser PRISME. Les résultats de l'évaluation de l'acceptation de PRISME présentés dans la Section 4.2 ont montré que notre modèle est perçue comme ac-

ceptable par les utilisateurs (voir Figure 4.31). Rappelons qu'en complément de nos questionnaires quantitatifs, nous avons également demandé aux participants (P*) d'exprimer à voix-haute leurs réflexions durant toute l'évaluation en suivant le protocole Think Aloud, présenté par Ericsson *et al.* [34]. Nous abordons et discutons certains résultats de ces données qualitatives dans la section suivante.

4.3.2 Retours qualitatifs

Lors de l'évaluation, les participants (P*) ont indiqué (protocole Think Aloud) que PRISME *"favorise l'observation directe"*(P8), que *"Permettre de mesurer automatiquement le temps de réalisation des actions est génial, ça permet de se concentrer sur le comportement plutôt que le chronomètre"*(P8). Ils ont indiqué également que *"Grace à PRISME, j'ai beaucoup moins le stress du chrono ou de la tâche faite ou pas, et je peux me focaliser sur les observables importants"*(P8), que *"J'ai toujours eu des difficultés à suivre les activités dans mes évaluations, à cause du papier et du crayon, si ç'avait été automatisé, c'est sûr que ç'aurait été mieux"*(P17) et *"Avec Prisme, on pourrait presque se passer d'observateur"*(P15). Ces remarques confirment l'utilité et le gain d'efficacité d'un suivi de l'activité en temps réel. Ces commentaires appuient les résultats positifs de l'évaluation de l'utilité perçue (cf. Section 4.2.2), et de la Facilité d'Utilisation Perçue (cf. Section 4.2.1). De leur point de vue d'ergonome, les participants ont estimé que *"Si je peux avoir l'activité réelle de l'utilisateur retranscrite en diagramme d'activité, c'est juste magique"*(P8), que *"C'est génial de pouvoir enregistrer l'activité de l'utilisateur et de pouvoir la comparer à une activité prescrite"*(P16), *"Enregistrer l'activité me permet de montrer par la suite que l'utilisateur à réaliser telle action avant telle autre, et c'est génial."*(P8). Ils ont également estimé que *"Sur un modèle de tâche complexe, connaître le chemin pris par l'utilisateur et mesurer sa performance par rapport à un modèle prescrit est très intéressant"*(P12), *"Pouvoir enregistrer l'activité de l'utilisateur en temps réel est un gain de temps énorme"*(P13). Ces commentaires confirment l'utilité de la création d'activité en temps réel. Ils confirment également le gain de temps et d'efficacité dans les tâches d'évaluation ergonomiques de cette fonctionnalité.

La possibilité de capturer en temps réel l'évolution d'indicateurs clé (KPI) est une fonctionnalité appréciée des participants, qui ont estimé que *"Pouvoir paramétrer l'affichage afin de gérer les buts d'observation est très intéressant"*(P8), *"Enregistrer les KPI pour analyser les données a posteriori permet de se focaliser sur le comporte-*

ment de l'utilisateur"(P8), et que "Pouvoir déterminer quel KPI est enregistré est extrêmement intéressant"(P17). Toutefois, certains ont estimé que "PRISME augmente la capacité analytique, et donc permet des analyses plus fines, mais pas forcément une analyse plus rapide"(P15). Cela est dû aux possibilités étendues de capture de données (données physiologiques, comportementales, orales, etc.) durant l'exécution des procédures en EV, qui nécessitent par la suite une analyse plus complexe de ces données.

4.4 Conclusion

Dans ce chapitre, nous avons présenté l'expérimentation que nous avons mise en place afin d'évaluer l'acceptation et la validité de notre proposition de modèle d'interaction. Afin d'évaluer l'acceptation, nous nous sommes appuyés sur le modèle TAM proposé par Davis *et al.* [27]; Haugstvedt *et al.* [48]; Lewis *et al.* [81] en nous concentrant sur les critères d'Utilité Perçue (UP) et de Facilité d'Utilisation Perçue (FUP). D'après le modèle TAM, l'Utilité Perçue et la Facilité d'Utilisation Perçue sont les deux principaux facteurs impactant l'intention d'utiliser un système ou une nouvelle technologie, selon Davis *et al.* [27]. Cette expérimentation a eu lieu dans deux lieux distincts : 1) le laboratoire Human Factors Technologies de l'Institut de Recherche Technologique b<>com et 2) la société HDG (Human Design Group). En tout, dix-sept personnes ont participé à cette étude. L'expérimentation s'est déroulée en quatre phases. Lors de la première phase, nous avons présenté le contexte de l'expérimentation, la tâche expérimentale ainsi que le rôle du participant durant cette tâche. Les participants ont pu étudier un document relatant les détails de l'expérimentation durant le temps qu'ils souhaitaient. Lors de la seconde phase, les participants ont observé un compère réaliser une procédure de ravitaillement en vol. Ils ont été libres de suivre d'une part l'exécution de l'activité et l'évolution d'indicateurs clés sur les interfaces de supervision de PRISME en temps réel, et d'autre part les manipulations réalisées par le compère. Lors de la troisième phase de l'évaluation, les participants ont utilisé les interfaces de PRISME afin de modifier les liens entre les actions du modèle métier et les interacteurs présents durant la réalisation de la procédure métier. La dernière phase de l'évaluation consistait à renseigner un questionnaire composé de deux parties : 1) Utilité Perçue ; 2) Facilité d'Utilisation Perçue, ayant pour objectif d'évaluer l'acceptation de notre proposition de modèle. Afin de recueillir des retours qualitatifs en complément des évaluations quanti-

tatives nous avons demandé aux participants d'exprimer à voix haute leurs remarques durant tout le déroulé de l'évaluation.

Les résultats des deux premières parties de notre questionnaire (UP et FUP) ont montrés que notre proposition de modèle d'interaction est considérée comme étant acceptable par les participants. De plus, un alignement du résultat de l'évaluation de l'Utilité Perçue et de la Facilité d'Utilisation Perçue avec l'échelle SUS octroie le score de 78,97 à PRISME ce qui le classe dans la partie "acceptable" de la plage d'acceptation de l'échelle SUS et lui associe le qualificatif de "bonne". Les résultats de cette analyse nous permettent de conclure que PRISME est acceptable pour les tâches d'évaluation en ergonomie. D'une manière générale, les commentaires et remarques que les participants ont exprimées durant l'expérimentation appuient cette analyse. Il est à noter que la fonctionnalité la moins bien perçue est celle correspondante à l'enregistrement des indicateurs clé (KPI). En effet, les participants ont mis en avant que si la possibilité de multiplier les données captées lors des évaluations permettait une analyse plus fine, elle en augmentait également sa complexité et donc sa durée. Bien que positive, notre évaluation pourrait être complétée par une étude comparative. Cette seconde évaluation aurait pour objectif de mesurer l'impact de PRISME sur la complexité et la rapidité de conception d'un environnement virtuel d'évaluation de système complexe.

CONCLUSION

Nous avons vu en introduction de ce manuscrit que le cycle de vie d'un système complexe (ex. une console technique, un poste d'atelier, etc.), se décompose selon Walden *et al.* [130] en plusieurs phases (voir figure 5.1, pour rappel). L'une d'elle est la phase de design (phase n°2 sur la Figure 5.1) et consiste à décrire précisément le système et ses fonctionnalités. C'est en particulier dans cette phase que l'ergonomie du produit / système, son efficacité et son efficience sont définis.

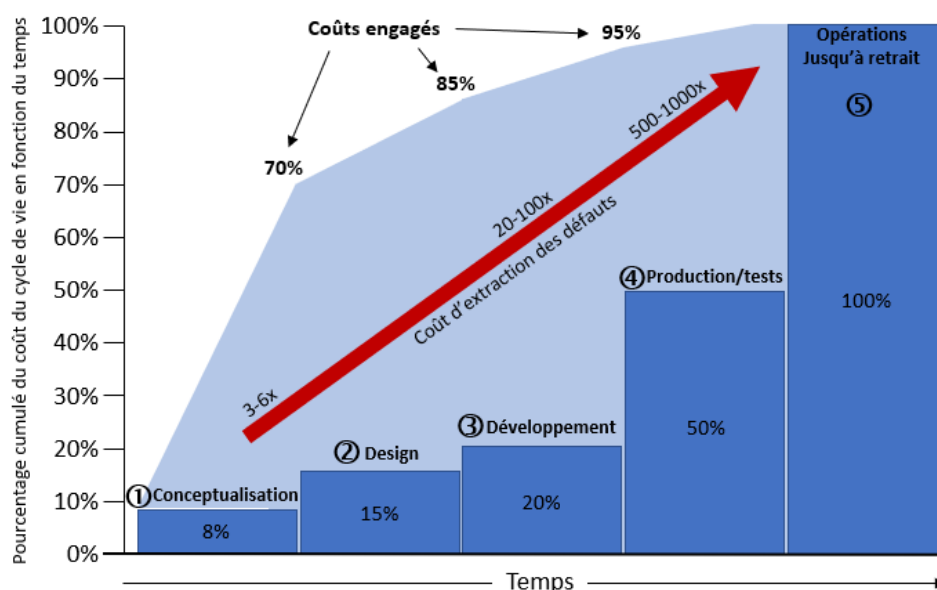


FIGURE 5.1 – Cycle de vie et évolution des coûts engagés et des coûts d'extraction des défauts d'un système (adapté de Walden *et al.* [130]).

Toujours selon Walden *et al.* [130], l'impact financier dû au coût d'extraction des défauts engendrés par une détection tardive des erreurs pourrait s'élever, dans le pire des cas, jusqu'à mille fois le montant des coûts investis. C'est dans cette optique de réduire ou limiter les coûts de conception d'un système que nos travaux de thèse se sont inscrits. Ainsi, l'objectif de nos travaux était de contribuer à la diminution des coûts

engagés lors des premières étapes de conception, mais aussi d'optimiser la détection des erreurs lors de ces étapes, et donc d'éviter les coûts engendrés par la détection tardive des erreurs.

Le résultat de nos travaux vise donc à faciliter le travail des équipes d'ergonomes dans leurs tâches de conception. Pour cela, nous avons proposé de redécouper la phase de design en quatre étapes cycliques : 1) la modélisation du système et des activités, 2) la simulation des IHM en Environnement Virtuel et tangible, 3) la fabrication de maquettes et 4) les expérimentations (voir Figure 5.2).

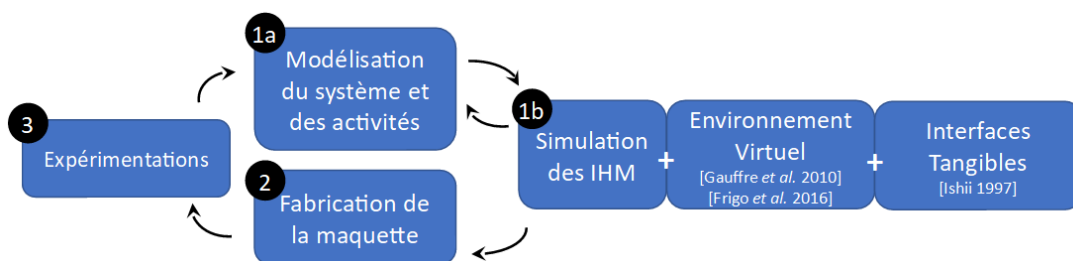


FIGURE 5.2 – Notre proposition de cycle d'étude et d'évaluation d'un système complexe découpé en 4 étapes

Pour l'étape de modélisation du système (étape **1a** dans la Figure 5.2), nous avons proposé d'effectuer cette dernière à l'aide de diagrammes de classes UML (Unified Modeling Language) [11], et de décrire l'activité humaine réalisée par l'opérateur sur ce système au moyen de diagrammes d'activités métier. Le choix de ce langage se justifie par le fait qu'il peut être interprété et exécuté de manière automatique dans l'EV. De plus, c'est actuellement un langage accessible et utilisé par les ergonomes et, plus globalement, connu de la majorité des concepteurs industriels. En ce qui concerne l'étape de simulation des IHM (étape **1b** dans la Figure 5.2), nous avons proposé de nous appuyer sur les Environnements Virtuels et les interactions tangibles. L'utilisation de ce couple permet non seulement un prototypage rapide, mais également une immersion de l'opérateur dans un environnement simulé accentuée par l'utilisation d'objets physiques et tangibles rendant ainsi les interactions plus adaptées et naturelles. Grâce à MASCARET, un ergonome peut simuler en temps réel une activité en environnement virtuel. Toutefois, il ne peut pas faire le lien entre les manipulations des interacteurs par l'utilisateur et les actions de l'activité, sans développements informatiques spécifiques.

Pour ces travaux, nous avons émis l'hypothèse qu'un modèle générique d'interactions permettant de reconnaître automatiquement les actions métiers à partir des

interactions de l'utilisateur, permettrait une plus grande agilité lors des expérimentations des prototypes par des ergonomes. De cette hypothèse, nous avons déduit trois questions de recherches :

- **Q1** : Comment décrire de manière générique les différents types d'interacteurs de Réalité Mixte (RM) et Tangibles qui composeront les IHM simulées ?
- **Q2** : Comment décrire la plate-forme et les activités de l'opérateur ?
- **Q3** : Comment reconnaître automatiquement l'action dans l'activité à partir des interactions utilisateur ?

En réponse à la première question, nous avons proposé une nouvelle typologie d'interacteurs en nous appuyant et en augmentant les typologies déjà existantes. À ce titre, nous avons en particulier intégré dans notre proposition la notion d'interacteur tangible en nous appuyant sur modèle MCRit présenté par Ullmer [118] et Ishii [58]. Au final, notre typologie qui comprend cinq grandes catégories d'interacteurs (voir Figure 5.3) permet de décrire la grande majorité des interacteurs présents dans les Environnements Virtuels.

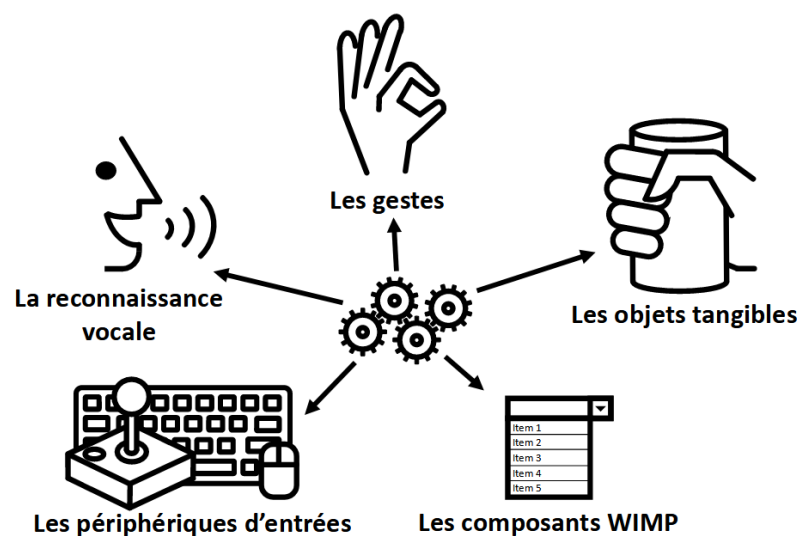


FIGURE 5.3 – Illustration synthétique de notre typologie d'interacteurs

Afin de répondre à notre seconde question de recherche, nous nous sommes appuyés sur le méta-modèle MASCARET, proposé par Querrec *et al.* [102]. MASCARET permet de décrire le système et les activités qui s'y rapporte dans un langage formel (UML) accessible aux ergonomes et suffisamment précis pour être exécutable en EV.

Dans notre proposition, nous avons augmenté MASCARET et y avons intégré la notion d'interacteur.

Enfin, en réponse à la troisième question de recherche, nous avons proposé un modèle d'interaction que nous avons appelé PRISME. PRISME permet de faire lien entre les interacteurs décrits à l'aide notre typologie et les actions d'un modèle d'activité formalisé grâce à MASCARET.

Ainsi, notre contribution permet à un ergonomiste de concevoir un EV sans faire intervenir un développeur pour créer les liens entre les interacteurs et les actions métiers. Elle permet également de suivre ou de créer en temps réel l'activité d'un utilisateur ainsi que les effets de cette activité sur le système simulé.

Nous poursuivons cette conclusion par une présentation des principaux résultats obtenus. Enfin, nous la terminons en exposant un ensemble de perspectives de nos travaux.

5.1 Principaux résultats obtenus

Les travaux de thèse que nous avons présentés dans ce manuscrit portent sur l'utilisation de la Réalité Virtuelle pour la conception de plateformes opérateur industrielles, par les ergonomes, afin d'en limiter les coûts de développement. Dans ce contexte, notre apport scientifique majeur est un modèle d'interaction permettant de faire le lien automatique (c'est-à-dire sans développement informatique spécifique) entre les interactions réalisées par l'opérateur (via des périphériques ou des objets tangibles) et les activités métiers.

Notre modèle appelé PRISME se compose d'une typologie d'interacteurs génériques permettant aux ergonomes de concevoir des situations de simulation en EV en exploitant tous types d'interacteurs. Dans notre typologie, cinq types d'interacteurs ont été proposés. Avec ces cinq types d'interacteurs, nous estimons que tout dispositif ou système de RM et tangible peut être représenté. Comparativement aux typologies existantes, notre typologie apporte une augmentation sur deux points : 1) la prise en considération des interactions tangibles et 2) la séparation des types d'interacteurs des types d'interactions, ce qui facilite leur utilisation dans un modèle d'interaction lié à un modèle métier.

Afin de valider l'acceptation de notre modèle, nous avons mené une expérimentation de PRISME avec dix-sept d'ergonomes (étudiants et professionnels) qui, pour

rappel, sont les utilisateurs cibles de notre proposition. Pour cela, nous nous sommes appuyés sur le modèle TAM [28] Haugstvedt *et al.* [48]; Alharbi *et al.* [1]; [81] en nous concentrant sur les critères de Facilité d'Utilisation Perçue (FUP) et d'Utilité Perçue (UP).

Lors de cette expérimentation, nous avons demandé aux participants / ergonomes d'évaluer un système complexe (dans notre cas, un cockpit d'avion) au moyen de notre modèle PRISME et de ses interfaces dédiées. Le participant avait pour tâche d'observer un compère (pilote de l'avion) réaliser une procédure de ravitaillement en vol à l'aide d'interacteurs tangibles. Le participant pouvait ainsi observer la réalisation de l'activité en temps réel, l'évolution d'indicateurs clés (KPI) et les interactions du pilote sur les interacteurs tangibles.

Les résultats de notre expérimentation ont montré que PRISME est considéré comme étant acceptable par les participants. Plus exactement, PRISME obtient un score de Facilité d'Utilisation Perçue (FUP) de 77.65 sur 100 et un score d'Utilité Perçue (UP) de 80.29 sur cent, et donc un score global de 78.97, ce qui place PRISME dans la fourchette acceptable de l'échelle SUS avec une note de C (voir figure 5.4).

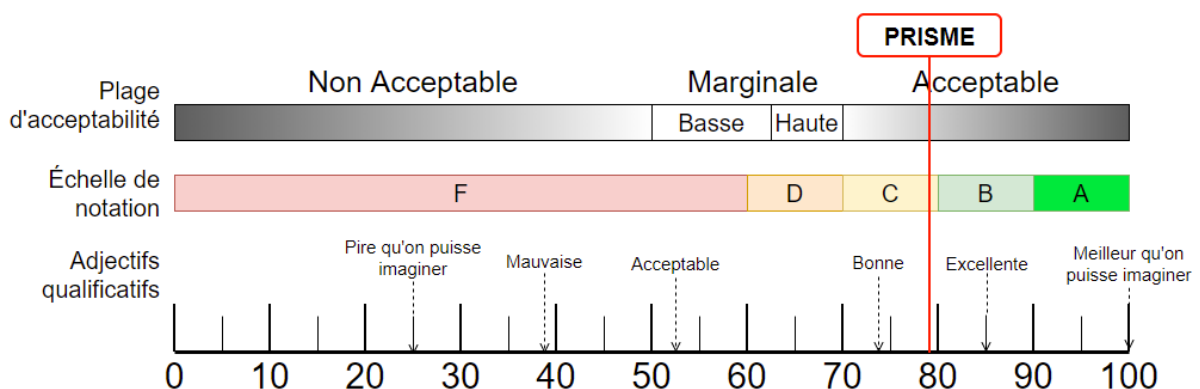


FIGURE 5.4 – Positionnement de PRISME sur la représentation de l'échelle SUS

5.2 Limites de nos travaux et perspectives

La mise en œuvre de notre modèle sur un cas d'usage a permis de révéler quelques limites de nos travaux. En effet, l'objectif de nos travaux était de permettre aux ergonomes de concevoir, modifier et de suivre en temps réel l'activité d'un opérateur sans intervention de nouveaux codages informatiques. À l'aide de notre modèle PRISME,

notre objectif est en très grosse partie atteint. Mais, en l'état, une équipe informatique doit toutefois rédiger un fichier `xm1` afin de décrire les interacteurs en présence. Bien que ces développements puissent être considérés comme infimes, ils n'en restent pas moins peu adaptés à des ergonomes et actuellement nécessaires au bon fonctionnement de notre modèle. Ainsi, afin de permettre à un ergonome d'être totalement autonome quant à la description d'un système, il serait possible de créer une interface permettant de décrire et de modifier des interacteurs.

De plus, comme expliqué dans le chapitre 2, nous n'avons implémenté dans notre reconnaissance d'action que les actions de types `CallOperationAction` et `VariableAction`. Nous couvrons ainsi une large partie des actions utilisées dans une étude ergonomique, mais limite toutefois les possibilités de description des activités de l'ergonome. Il serait intéressant d'implémenter les autres types d'actions issues de la norme UML, ainsi que les types d'actions spécifiques à MASCARET.

Dans notre modélisation des interactions, nous nous sommes focalisés sur les interactions de type `Manipulation`, un des quatre grands types d'interaction décrit par Bowman *et al.* [16], en estimant que la grande majorité des modifications apportées à l'environnement par un utilisateur résultait d'une manipulation. Nous pourrions également implémenter les actions de `Selection` et de `Navigation`, afin de couvrir l'ensemble des types d'interaction décrits par Bowman *et al.* [16].

En perspectives de nos travaux, il serait également intéressant de créer une typologie d'interactions (*e.g.*, prendre, tourner, déplacer, *etc.*) afin de simplifier la lecture des interfaces de configuration des interactions. Nous pourrions également ajouter des stéréotypes aux actions métiers (*e.g.*, action motrice, perceptive, *etc.*) et la capacité de décrire les interactions en utilisant un diagramme d'activité. Ces augmentations apportées à notre modèle devraient faciliter la spécification des interactions par l'ergonome, mais également lui permettre de la modifier en temps réel lors de la simulation (par exemple s'il perçoit une difficulté d'utilisation par l'opérateur).

Enfin, nous serait également possible de procéder à une validation expérimentale supplémentaire de notre modèle selon deux approches complémentaires. D'une part, nous serait intéressant d'évaluer la charge de développements d'un EV avec et sans notre modèle. Cela permettrait de mettre en évidence sinon valider les avantages théoriques de PRISME du point de vue du développeur informatique. D'autre part, évaluer la faisabilité de décrire un système et ses activités en EV, sans développement, avec et sans notre modèle et, cette fois-ci, du point de vue de l'ergonome, serait également

à point à traiter. Ce dernier permettrait de valider de manière formelle que PRISME réduit les délais et les coûts de développement des plate-formes opérateur.

PUBLICATIONS

Conférences internationales avec actes et comité de lecture

- **Fazzari, J. M.**, Kubicki, S., & Querrec, R. (2020, November). PRISME : An interaction model linking domain activities and mixed and tangible interactors in virtual environments. In 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct) (pp. 241-246). IEEE.
- **Fazzari, J. M.**, Kubicki, S., & Querrec, R. (2020, November). PRISME : Toward a model linking tangible interaction and domain-specific tasks in mixed reality environments. In 2020 22nd Symposium on Virtual and Augmented Reality (SVR) (pp. 210-219). IEEE.

Séminaires et autres publications

- **Fazzari, J. M.** (2019a, juillet). Modèle générique d'interaction liant modèles d'actions métier et interactions tangibles en environnement virtuel collaboratif. RJC-IHM'19, Toulouse, France
- **Fazzari, J. M.** (2019b, décembre). Modèle de Reconnaissance d'Actions Métier à partir des Interactions Utilisateur en Environnement Virtuel. Rencontres Doctorales IHM'19, Grenoble France.

LISTE DES INTERACTEURS COMPOSANT NOTRE PLATEFORME EXPÉRIMENTALE

1. Un dispositif de contrôle radio



FIGURE A.1 – Dispositif de contrôle radio

Cet interacteur est décrit dans notre typologie comme un [InputPeripheral-Controller] (voir chapitre 2.1, figure 2.1). Il est composé d'axes et de boutons décrits tels que :

- 1 **XPDRUp** : Axe renvoyant des valeurs entières comprises entre 0 et 6
- 2 **XPDRDown** : Axe renvoyant des valeurs entières comprises entre 0 et 6
- 3 **Frequence1Up** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
- 4 **Frequence2Up** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
- 5 **Frequence1Down** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
- 6 **Frequence2Down** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
- 7 **ACTUp** : Bouton poussoir renvoyant la valeur 1 si appuyée, 0 si relâché

8 **ACTDown** : Bouton poussoir renvoyant la valeur 1 si appuyée, 0 si relâché

2. Un dispositif de contrôle switch

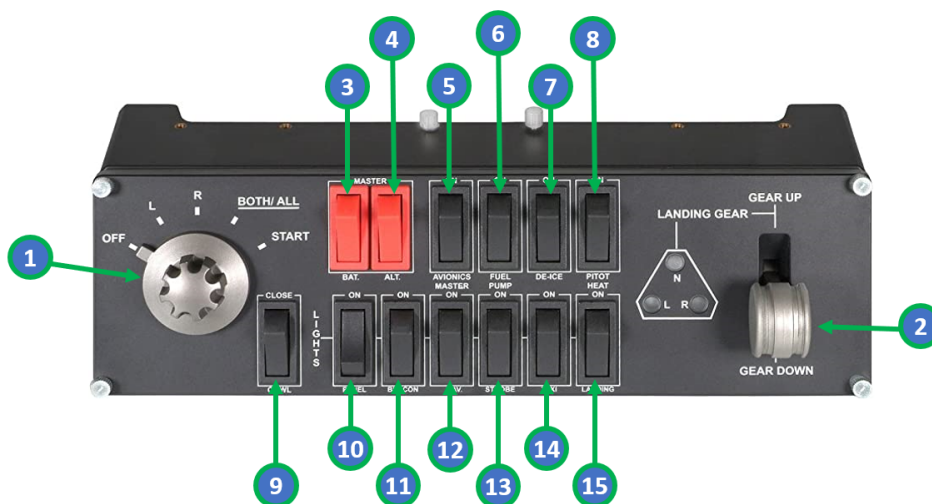


FIGURE A.2 – Dispositif de contrôle switch

Cet interacteur est décrit dans notre typologie comme un [InputPeripheral-Controller] (voir chapitre 2.1, figure 2.1). Il est composé d'axes et de boutons décrits tels que :

- 1 **RotaSwitch** : Axe renvoyant des valeurs entières comprises entre 0 et 4
- 2 **GEAR** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 3 **BAT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 4 **ALT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 5 **AVIONIC** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 6 **FUEL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 7 **CE-ICE** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 8 **PITOT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 9 **COWL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 10 **PANEL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 11 **BEACON** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 12 **NAV** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 13 **STROBE** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
- 14 **TAXI** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse

15 **LANDING** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse

3. Une manette de jeu Logitech X52 PRO



FIGURE A.3 – Joystick Logitech X52 PRO

Cet interacteur est décrit dans notre typologie comme un [InputPeripheral-Controller] (voir chapitre 2.1, figure 2.1). Il est composé d'axes et de boutons décrits tels que :

- 1 **Trigger light / Trigger hard** : Bouton (appui léger / appuis fort)
- 2 **B** : Bouton
- 3 **A** : Bouton
- 4 **Fire** : Bouton
- 5 **Cap X / Cap Y** : Axe renvoyant les valeurs entières -1 et 1
- 6 **Shift** : Bouton
- 7 **T1 / T2** : Boutons (T1 actionné vers le haut, T2 vers le bas)
- 8 **T3 / T4** : Boutons (T3 actionné vers le haut, T4 vers le bas)
- 9 **T5 / T6** : Boutons (T5 actionné vers le haut, T6 vers le bas)
- 10 **Roll / Pitch / Yaw** : Axes renvoyant des valeurs réelles comprises entre -1 et 1

4. Une manette de gaz Logitech X52 PRO



FIGURE A.4 – Manette de gaz Logitech X52 PRO

Cet interacteur est décrit dans notre typologie comme un [InputPeripheral-Controller] (voir chapitre 2.1, figure 2.1). Il est composé d'axes et de boutons décrits tels que :

- 1 **GAS** : Axe renvoyant une valeur réelle comprise entre -1 et 1
- 2 **I** : Bouton
- 3 **Rotation I** : Axe renvoyant une valeur réelle comprise entre -1 et 1
- 4 **D** : Bouton
- 5 **E** : Bouton
- 6 **Rotation E** : Axe renvoyant une valeur réelle comprise entre -1 et 1
- 7 **Slide** : Axe renvoyant une valeur réelle comprise entre -1 et 1

5. Un clavier standard

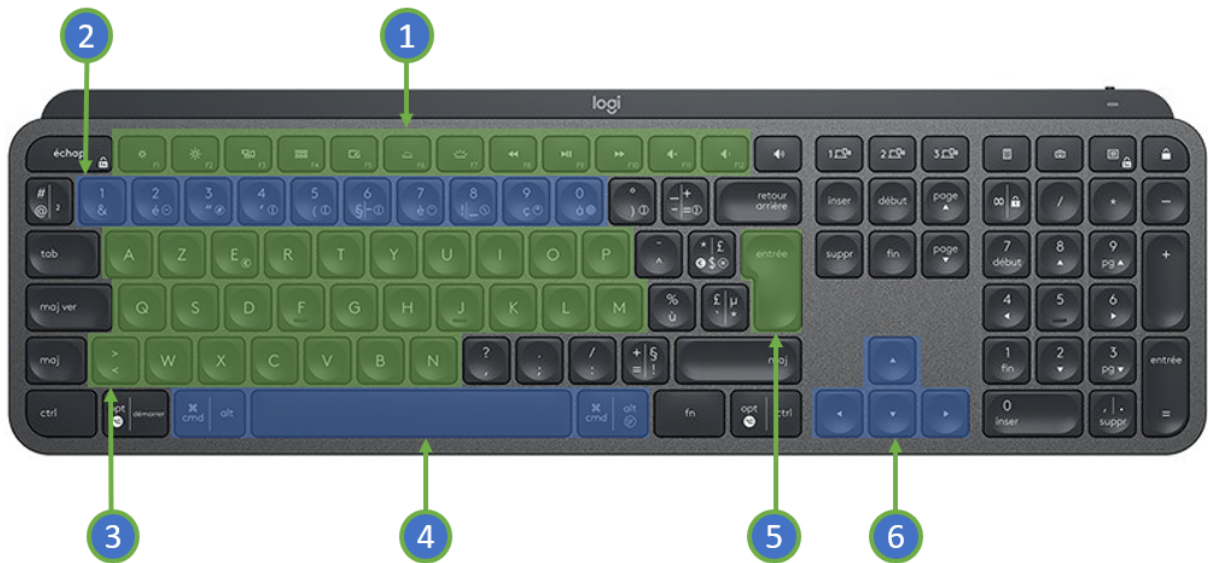


FIGURE A.5 – Clavier standard 108 touches

Cet interacteur est décrit dans notre typologie comme un [InputPeripheralController] (voir chapitre 2.1, figure 2.1). Il est composé de 57 boutons décrits tels que :

- 1 **Les touches de F1 à F12**
- 2 **Les touches de 1 à 0**
- 3 **Les touches de A à Z et la touche < >**
- 4 **Les touches ALT droite, Espace et ALT gauche**
- 5 **La touche Entrée**
- 6 **Les touches directionnelles**

DOCUMENT DE PRÉSENTATION DE L'EXPÉRIMENTATION AUX PARTICIPANTS

1.1 Contexte

Nos travaux ont pour objectif de réduire les coûts et les délais de conceptions de plateformes opérateurs. Nous émettons l'hypothèse qu'intégrer une étape de simulation en environnement virtuel dans le cycle de prototypage permettrait de réduire ces coûts et délais. En effet, la réalité virtuelle permet non seulement de prototyper rapidement les nouvelles configurations d'une plate-forme mais également d'immerger l'opérateur dans un environnement virtuel pour y réaliser les activités métiers associées. L'utilisateur ainsi immergé, réalisera les activités métiers par le biais d'Interfaces Utilisateurs Tangibles (TUI, en anglais), considérées comme plus naturelles, conviviales et intuitives pour interagir dans un Environnement Virtuel (EV).

L'objectif de nos travaux rentre dans le cadre de la conception de plate-formes opérateur. Il vise à permettre aux ergonomes de concevoir, modifier et suivre en temps réel l'activité en RM sans intervention de nouveaux codages. Plus précisément, notre apport scientifique est de faire le lien automatique (c'est à dire sans développements informatiques spécifiques) entre les interactions réalisées par l'opérateur via des périphériques ou des objets tangibles et les actions métiers décrites dans les activités.

Votre tâche expérimentale consiste à réaliser une partie de l'étude ergonomique d'un cockpit d'avion. Vous intervenez sur la phase de simulation de la plateforme opérateur en environnement virtuel et tangible. Le système et l'activité ont été décrit au préalable et vous sont présentés dans ce document.

Vous allez réaliser cette étude en magicien d'Oz. L'utilisateur manipulant les interacteurs du cockpit d'avion n'étant pas un pilote, mais un compère. Cette étude se déroulera en deux phases durant lesquelles le compère réalisera une procédure de ravitaillement en vol. Lors de chaque phase, vous réaliserez en temps réel le suivi de l'activité métier, et observerez l'évolution de KPI spécifiques sur une interface de supervision dédiée.

1.2 La procédure métier

La procédure métier est réalisée par un agent contrôlé par un humain. Cet agent à le rôle ayant le rôle de [pilote], et possède les capacités (opérations) suivantes :

- ValiderFrequence
- DemanderRavitaillement
- DemanderArrimage
- DemanderAlimentationCarburant
- DemanderFinRavitaillement
- ConfirmerFinRavitaillement
- ValiderReservoirsPleins
- DemanderFinAlimentationCarburant
- DemanderDesarrimage

Ces opérations sont référencées dans les actions éponymes du modèle d'activité présenté figure 1.

Lors de l'exécution de la procédure métier, le pilote réalisera deux types d'actions différentes :

- Les actions permettant d'exécuter une opération particulière (capacités de l'agent), comme par exemple l'action **ValiderFrequence** en figure 1
- Les actions permettant de modifier une valeur d'attribut, comme par exemple l'action **ModifierFrequence** en figure 1

Ces actions vont agir sur des ressources décrites dans le modèle métier :

- La ressource **avion**, qui représente l'appareil piloté par le pilote.
- La ressource **ravitailleur**, qui représente l'appareil approché pour le ravitaillement.

Ces ressources sont des instances de la classe [Avion] qui contiennent quatre attributs :

- Longitude : attribut publique de type entier
 - Latitude : attribut publique de type réel
 - Altitude : attribut publique de type réel
 - Vitesse : attribut publique de type réel
-
- La ressource **reservoirDroite**, qui représente le réservoir situé sous l'aile droite de l'appareil piloté par le pilote. Ce réservoir alimente le moteur.
 - La ressource **reservoirGauche**, qui représente le réservoir situé sous l'aile gauche de l'appareil piloté par le pilote. Ce réservoir alimente le moteur.
 - La ressource **reservoirCentral**, qui représente le réservoir permettant d'accueillir le carburant provenant de l'appareil ravitailleur. Le carburant qui y est déversé sera ensuite redistribué aux réservoirs droite et gauche de l'appareil.
 - La ressource **moteur**, qui représente le moteur de l'appareil piloté par le pilote. Cette ressource consomme du carburant.
 - La ressource **tanker**, qui représente le réservoir de carburant transporté par l'appareil ravitailleur.

Ces ressources sont des instances de la classe [Reservoir], qui contiennent deux attributs :

- Capacité : attribut publique de type entier
 - Niveau : attribut publique de type réel
-
- La ressource **VanneMoteurDroite**, qui représente la vanne reliant le réservoir de droite au moteur.
 - La ressource **VanneMoteurGauche**, qui représente la vanne reliant le réservoir de gauche au moteur.
 - La ressource **VanneTankerDroite**, qui représente la vanne reliant le réservoir central au réservoir de droite.
 - La ressource **VanneTankerGauche**, qui représente la vanne reliant le réservoir central au réservoir de gauche.

-
- La ressource **VanneRavitailleur**, qui représente la vanne reliant le réservoir de de l'appareil ravitailleur au réservoir central.

Ces ressources sont des instances de la classe [Vanne], qui contiens deux attributs :

- Ouverture : attribut publique de type réel
- EstAlimentee : attribut publique de type booléen

Une ressource **radioPanel**, qui représente le dispositif radio de l'appareil.

Cette ressource est une instance de la classe [RadioPanel], qui contiens huit attributs :

- FrequenceHautDroite : attribut publique de type entier
- FrequenceHautGauche : attribut publique de type entier
- FrequenceBasDroite : attribut publique de type entier
- FrequenceBasGauche : attribut publique de type entier
- TypeFrequenceHaut : attribut publique de type entier
- TypeFrequenceBas : attribut publique de type entier
- ValidationHaut : attribut publique de type entier
- ValidationBas : attribut publique de type entier

Une ressource **switchPanel**, qui représente le dispositif de gestion des vannes.

Cette ressource est une instance de la classe [SwitchPanel]

Cette classe contiens Quinze attributs :

- Switch : attribut publique de type entier
- Master_Bat : attribut publique de type booléen
- Master_Alt : attribut publique de type booléen
- Avionic_Master : attribut publique de type booléen
- Fuel_Pump : attribut publique de type booléen
- DE-ICE : attribut publique de type booléen
- Pitot_Heat : attribut publique de type booléen
- Cowl : attribut publique de type booléen
- Panel : attribut publique de type booléen
- Beacon : attribut publique de type booléen
- Navigation : attribut publique de type booléen
- Strobe : attribut publique de type booléen
- Taxi : attribut publique de type booléen
- Landing : attribut publique de type booléen
- Gear : attribut publique de type booléen

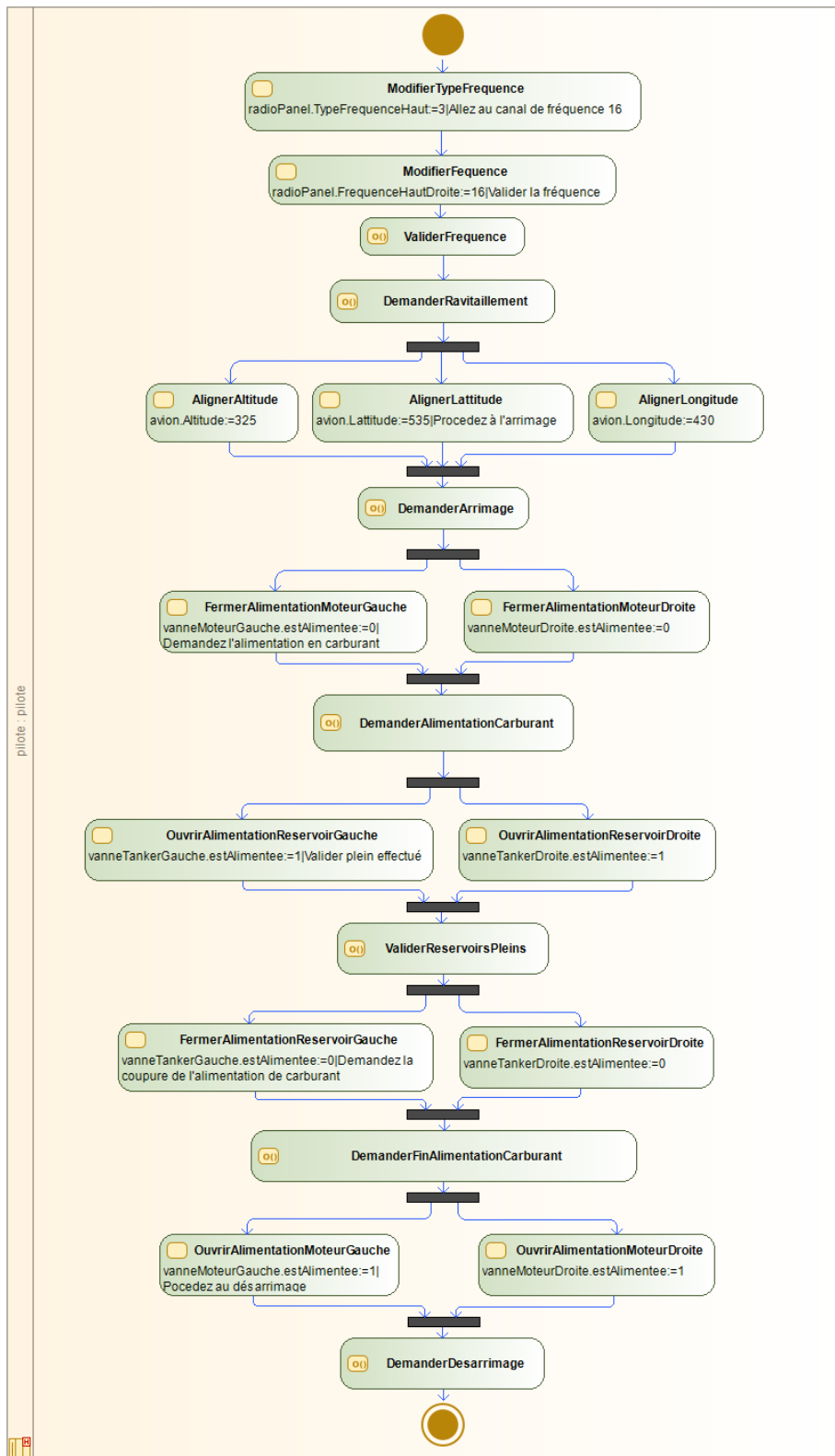


Figure 1 : Modèle d'activité métier

Afin de pouvoir suivre l'exécution de la procédure de ravitaillement en vol, vous aurez accès à deux interfaces de supervisions. La première interface permet de décrire les liens entre les interacteurs et les actions du modèle de tâche (voir figure 2).

Elle se découpe en quatre parties :

- **Profile** : permet d'enregistrer des profils d'interactions. Cela consiste à pouvoir facilement enregistrer et rappeler une configuration d'interactions particulière.
- **Exécuter une opération** : permet de lier un interacteur à une action dont la réalisation lance l'exécution d'une opération.
- **Modifier la valeur d'un attribut** : permet de modifier la valeur de l'attribut d'une instance de classe particulière, selon le type de l'attribut et la nature de l'interacteur.
- **Interactions enregistrées** : permet de visualiser les liaisons entre les interacteurs et les actions issues du modèle d'activité, enregistrées dans le profil sélectionné.

Interactions
Activités
Agents
Environnement

Profiles

Profils d'interactions : Ajouter un profil : +

Executer une operation

Operation : Interacteur : Actionneur :

Modifier la valeur d'un attribut

Type de ressource : Ressource : Attributs :

Transformation : Options : Valeur :

Interacteur : Actionneur :

Interactions enregistrées

| Action | Périphérique | Actionneur | |
|---------------------------------|--------------|--------------|----------------------------------|
| Modifier TypeFrequenceHaut | Radio_Panel | XPDRUp | <input type="button" value="x"/> |
| Modifier TypeFrequenceBas | Radio_Panel | XPDRDown | <input type="button" value="x"/> |
| Modifier FrequenceHautDroite | Radio_Panel | Frequence1Up | <input type="button" value="x"/> |
| Executer ValiderFrequence | keyboard | A | <input type="button" value="x"/> |
| Executer DemanderRavitaillement | keyboard | B | <input type="button" value="x"/> |

Figure 2 : Interface de gestion des interactions

La seconde interface permet de suivre en temps réel la réalisation de l'activité par le compère au sein de l'environnement virtuel, et de visualiser l'évolution des indicateurs clés de performance (voir figure 3).

Dans le diagramme d'activité, une action peut prendre différentes couleurs selon son état :

- **Blanc** : lorsque l'action n'a pas encore été réalisée
- **Bleu** : lorsque l'action est en cours de réalisation
- **Vert** : lorsque l'action est réalisée

Vous y trouverez également les données observables telles que :

- **Distance à l'accomplissement** : Cet entier représente la différence entre la valeur cible et la valeur actuelle de l'attribut à modifier
- **Début** : Représente l'heure de début de la réalisation de l'action
- **Fin** : Représente l'heure de fin de la réalisation de l'action
- **Overshoots** : Représente le nombre de dépassements effectués en tentant d'attendre la valeur cible

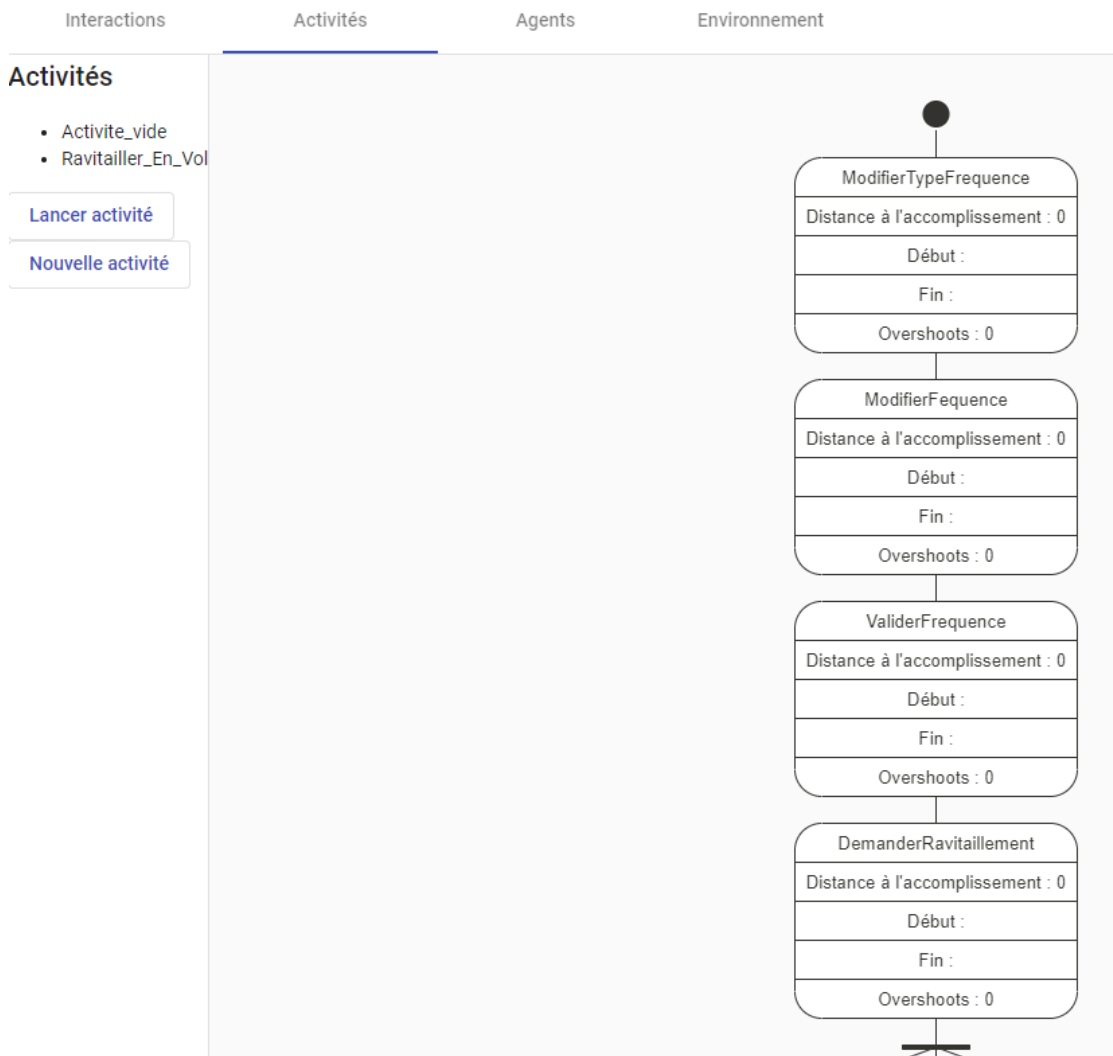


Figure 3 : Interface de supervision de l'activité

1.3 Tâche expérimentale

La procédure expérimentale que vous allez réaliser se découpe en deux phases. Durant la première phase, il vous est demandé d'observer une première exécution de la procédure métier en suivant l'évolution de celle-ci sur les interfaces de supervision. Au cours de la seconde phase, vous devrez modifier au préalable les interactions permettant la réalisation de deux actions spécifiques. Vous observerez ensuite sur les interfaces de supervision les effets de ces modifications sur l'exécution de la procédure métier.

1.3.1 Première phase

Durant la première phase de l'évaluation, vous allez observer le compère réaliser la procédure de ravitaillement en vol. Vous suivrez l'exécution de l'activité et l'évolution des indicateurs clés sur les interfaces de supervision en temps réel, et renseignerez une grille d'observation.

1.3.1 Seconde phase

La seconde phase de l'évaluation est composée de deux étapes. La première consiste à modifier les liens entre les actions du modèle métier et les interacteurs présents durant la réalisation de la procédure métier. Pour cela, vous utiliserez l'interface prévue à cet effet (voir figure 2). Vous modifierez les interactions réalisant les actions **ModifierFrequence** et **ValiderFrequence**. La description complète des interacteurs à disposition se trouvent en annexe de ce document. Vous constaterez ensuite la modification de la performance du compère, grâce aux relevés des données observables.

A l'issus de ces phases, vous renseignerez un questionnaire.

Annexe A. Les interacteurs

Annexe A.1. Commande vocale

Cet interacteur scrute en permanence le micro par défaut connecté au système, et renvoie les mots reconnus sous forme d'une chaîne de caractère.

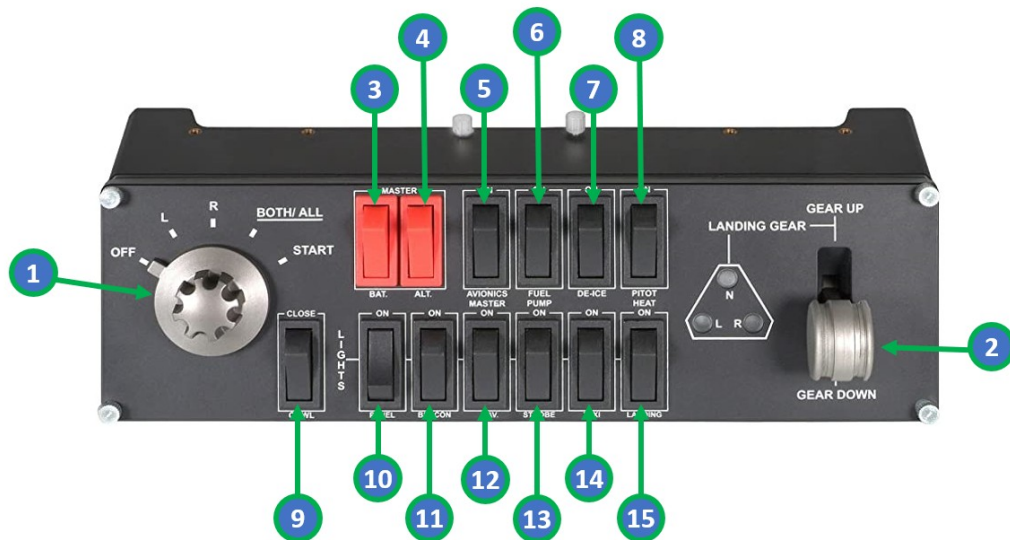
Annexe A.2. RadioPanel



Cet interacteur est composé d'axes et de boutons décrits tels que :

1. **XPRDUp** : Axe renvoyant des valeurs entières comprises entre 0 et 6
2. **XPRDDown** : Axe renvoyant des valeurs entières comprises entre 0 et 6
3. **Frequance1Up** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
4. **Frequance2Up** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
5. **Frequance1Down** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
6. **Frequance2Down** : Axe renvoyant la valeur 1 si rotation horaire, et -1 si rotation anti-horaire
7. **ACTUp** : Bouton pousseur renvoyant la valeur 1 si appuyée, 0 si relâché
8. **ACTDown** : Bouton pousseur renvoyant la valeur 1 si appuyée, 0 si relâché

Annexe A.3. SwitchPanel



Cet interacteur est composé d'axes et de boutons décrits tels que :

1. **RotaSwitch** : Axe renvoyant des valeurs entières comprises entre 0 et 4
2. **GEAR** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
3. **BAT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
4. **ALT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
5. **AVIONIC** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
6. **FUEL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
7. **CE-ICE** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
8. **PITOT** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
9. **COWL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
10. **PANEL** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
11. **BEACON** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
12. **NAV** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
13. **STROBE** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
14. **TAXI** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse
15. **LANDING** : Bouton à deux positions renvoyant la valeur 1 en position haute, 0 en position basse

Annexe A.4. Manette de gaz



Cet interacteur est composé d'axes et de boutons décrits tels que :

1. **GAS** : Axe renvoyant une valeur réelle comprise entre -1 et 1
2. **I** : Bouton
3. **Rotation I** : Axe renvoyant une valeur réelle comprise entre -1 et 1
4. **D** : Bouton
5. **E** : Bouton
6. **Rotation E** : Axe renvoyant une valeur réelle comprise entre -1 et 1
7. **Slide** : Axe renvoyant une valeur réelle comprise entre -1 et 1

Annexe A.5. Joystick



Cet interacteur est composé d'axes et de boutons décrits tels que :

1. **Trigger light / Trigger hard**: Bouton (appui léger / appui fort)
2. **B** : Bouton
3. **A** : Bouton
4. **Fire** : Bouton
5. **Cap X / Cap Y** : Axe renvoyant les valeurs entière -1 et 1
6. **Shift** : Bouton
7. **T1 / T2** : Boutons (T1 actionné vers le haut, T2 vers le bas)
8. **T3 / T4** : Boutons (T3 actionné vers le haut, T4 vers le bas)
9. **T5 / T6** : Boutons (T5 actionné vers le haut, T6 vers le bas)
10. **Roll / Pitch / Yaw** : Axes renvoyant des valeurs réelles comprises entre -1 et 1

MODÈLE DE TÂCHE DÉCRIVANT L'ACTIVITÉ DE RAVITAILLEMENT EN VOL

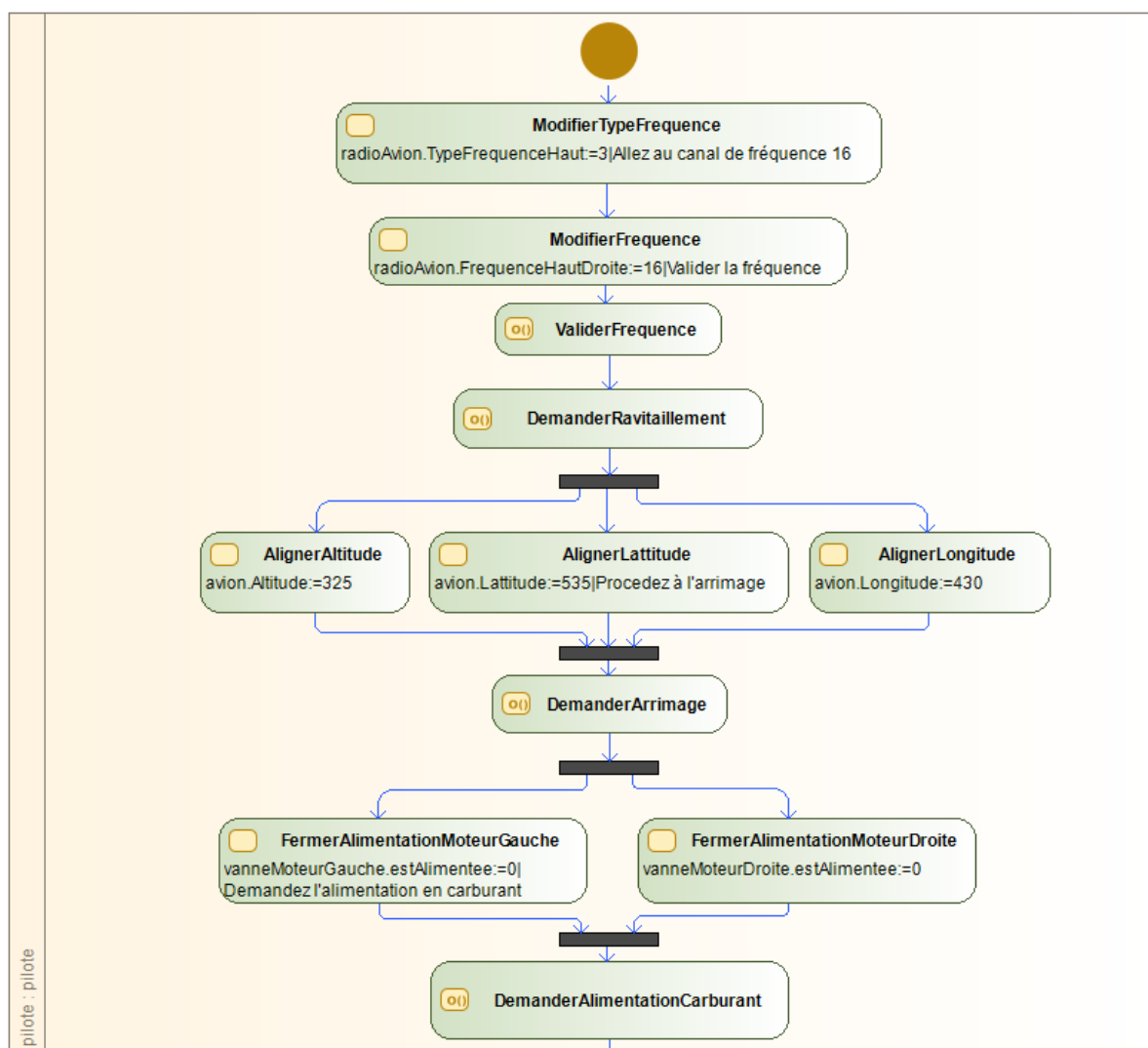


FIGURE C.1 – Première partie du modèle de tâche décrivant l'activité de ravitaillement en vol

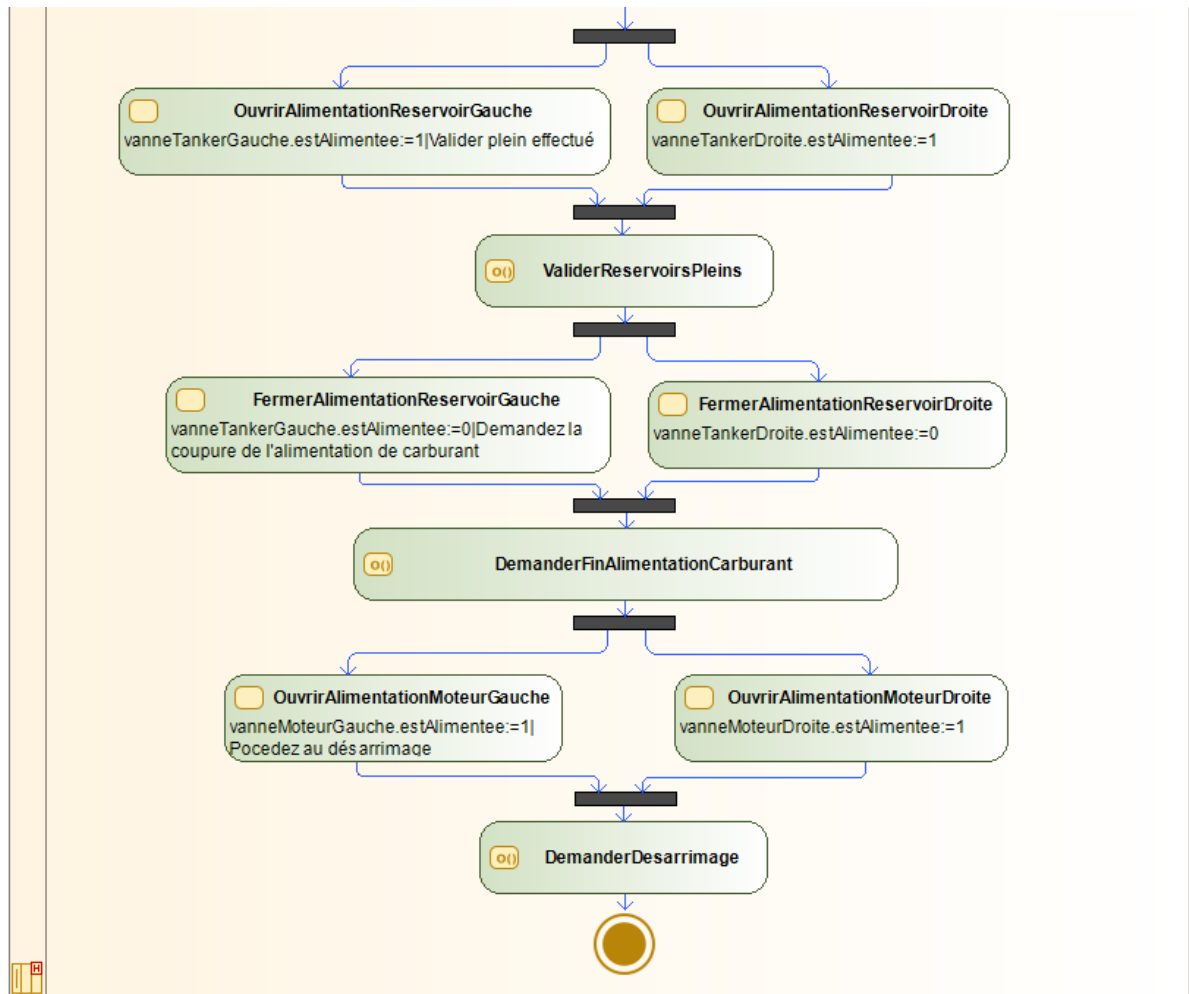


FIGURE C.2 – Seconde partie du modèle de tâche décrivant l'activité de ravitaillement en vol

QUESTIONNAIRE TAM ADAPTÉ AU MODÈLE PRISME

Participant number ____

Technology Acceptance Questionnaire

Response items use a 7 points likely **likely - unlikely** scale from extremely likely to extremely unlikely

| | Likely | | | Unlikely | | | |
|---|-----------|-------|----------|----------|----------|-------|-----------|
| | Extremely | Quite | Slightly | Neither | Slightly | Quite | Extremely |
| I feel that using PRISME would be easy for me <i>(Je pense que l'utilisation de PRISME serait facile pour moi)</i> | | | | | | | |
| I feel that my interaction with PRISME would be clear and understandable <i>(Je pense que mon interaction avec PRISME serait claire et compréhensible)</i> | | | | | | | |
| I feel that it would be easy to become skillful at using PRISME <i>(Je pense qu'il serait facile de devenir habile à utiliser PRISME)</i> | | | | | | | |
| I would find PRISME to be flexible to interact with <i>(Je trouverais que PRISME est flexible pour interagir avec)</i> | | | | | | | |
| Learning to operate PRISME would be easy for me <i>(Apprendre à opérer PRISME serait facile pour moi)</i> | | | | | | | |
| It would be easy for me to get PRISME to do what I want to do <i>(Ce serait facile pour moi de faire en sorte que PRISME fasse ce que je veux faire)</i> | | | | | | | |
| I feel that my ability to determine PRISME ease of use is limited by my lack of experience <i>(Je sens que ma capacité à déterminer la facilité d'utilisation de PRISME est limitée par mon manque d'expérience)</i> | | | | | | | |
| Using PRISME in my job would enable me to accomplish tasks more quickly <i>(Utiliser PRISME dans mon travail me permettrait d'accomplir des tâches plus rapidement)</i> | | | | | | | |

FIGURE D.1 – Première partie du questionnaire TAM adapté au modèle PRISME

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| Using PRISME would improve my job performance <i>(Utiliser PRISME améliorerait mes performances au travail)</i> | | | | | | | |
| Using PRISME in my job would increase my productivity <i>(Utiliser PRISME dans mon travail augmenterait ma productivité)</i> | | | | | | | |
| Using PRISME would enhance my effectiveness on the job <i>(Utiliser PRISME améliorerait mon efficacité au travail)</i> | | | | | | | |
| Using PRISME would make it easier to do my job <i>(Utiliser PRISME faciliterait mon travail)</i> | | | | | | | |
| I would find PRISME useful in my job <i>(Je trouverais PRISME utile dans mon travail)</i> | | | | | | | |
| I believe it is a good idea to use PRISME <i>(Je crois que c'est une bonne idée d'utiliser PRISME)</i> | | | | | | | |
| I like the idea of using PRISME <i>(J'aime l'idée d'utiliser PRISME)</i> | | | | | | | |
| Using PRISME is a positive idea <i>(Utiliser PRISME est une idée positive)</i> | | | | | | | |
| I plan to use PRISME in the future <i>(Je prévois d'utiliser PRISME dans le futur)</i> | | | | | | | |
| Assuming that I have access to PRISME, I intend to use it <i>(En supposant que j'ai accès à PRISME, j'ai l'intention de l'utiliser)</i> | | | | | | | |
| In my job, the usage of PRISME is important <i>(Dans mon travail, l'utilisation de PRISME est importante)</i> | | | | | | | |
| In my job, the usage of PRISME is relevant <i>(Dans mon travail, l'utilisation de PRISME est pertinente)</i> | | | | | | | |

FIGURE D.2 – Seconde partie du questionnaire TAM adapté au modèle PRISME

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Saleh ALHARBI et Steve DREW, « Using the technology acceptance model in understanding academics' behavioural intention to use learning management systems », in : *International Journal of Advanced Computer Science and Applications* 5.1 (2014), p. 143–155.
- [2] René AMALBERTI et Claude VALOT, « Le Magicien d'Oz », in : *CERMA, journée du PRC Rhône-Alpes* (1992).
- [3] Zineb AMALLAL, « LA CINETOSE (MAL DE MER) », in : (2021).
- [4] Alissa N ANTLE, Greg CORNESS et Milena DROUMEVA, « Human-computer-intuition? Exploring the cognitive basis for intuition in embodied interaction », in : *International Journal of Arts and Technology* 2.3 (2009), p. 235–254.
- [5] Ruth AYLETT et Marc CAVAZZA, « Intelligent Virtual Environments-A state-of-the-art report », in : *Eurographics Conference, Manchester, UK*, Citeseer, 2001.
- [6] Eric BARBONI et al., « Beyond Modelling : An Integrated Environment Supporting Co-Execution of Tasks and Systems Models », in : *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '10, Berlin, Germany : Association for Computing Machinery, 2010, p. 165–174, ISBN : 9781450300834, DOI : 10.1145/1822018.1822043, URL : <https://doi.org/10.1145/1822018.1822043>.
- [7] Eric BARBONI et al., « Beyond modelling : an integrated environment supporting co-execution of tasks and systems models », in : *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, ACM, 2010, p. 165–174.
- [8] Mickaël BARON et al., « K-MADe : un environnement pour le noyau du modèle de description de l'activité », in : *Proceedings of the 18th Conference on L'Interaction Homme-Machine*, IHM '06, Montreal, Canada : ACM, 2006, p. 287–288, ISBN : 1-59593-350-6.

-
- [9] M. BARON et al., « K-MADe : Un Environnement Pour Le Noyau Du Modèle de Description de l'activité », in : *Proceedings of the 18th Conference on l'Interaction Homme-Machine*, IHM '06, Montreal, Canada : Association for Computing Machinery, 2006, p. 287–288, ISBN : 1595933506, DOI : 10 . 1145 / 1132736 . 1132786, URL : <https://doi.org/10.1145/1132736.1132786>.
- [10] Michel BEAUDOUIN-LAFON, « Interaction instrumentale : de la manipulation directe à la réalité augmentée », in : *Actes des neuvièmes journées sur l'Interaction Homme-Machine* (1997), p. 97–104.
- [11] Grady BOOCH, James RUMBAUGH et Ivar JACOBSON, *Le guide de l'utilisateur UML*, Eyrolles, 2000.
- [12] Christophe BORTOLASO et Emmanuel DUBOIS, *LNCS 8119 - Model Assisted Creativity Sessions for the Design of Mixed Interactive Systems : A Protocol Analysis*, rapp. tech., 2013, p. 126–143.
- [13] Christophe BORTOLASO et al., « Méthode de Conception Assistée par les Modèles pour les Systèmes Interactifs Mixtes To cite this version : HAL Id : tel-00867039 », thèse de doct., juin 2013.
- [14] Rozenn BOUVILLE et al., « # FIVE : High-level components for developing collaborative and interactive virtual environments », in : *2015 IEEE 8th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, IEEE, 2015, p. 33–40.
- [15] Doug A. BOWMAN et al., « An Introduction to 3-D User Interface Design », in : *Presence : Teleoperators and Virtual Environments 10.1* (2001), p. 96–108, DOI : 10 . 1162 / 105474601750182342, eprint : <https://doi.org/10.1162/105474601750182342>, URL : <https://doi.org/10.1162/105474601750182342>.
- [16] Doug A BOWMAN et al., « An Introduction to 3-D User Interface Design », in : *Presence : Teleoperators and Virtual Environments 10.1* (2001), p. 96–108, DOI : 10 . 1162 / 105474601750182342.
- [17] Dan BRICKLEY, Ramanathan V GUHA et Andrew LAYMAN, « Resource description framework (RDF) schema specification », in : (1999).
- [18] John BROOKE, « Sus : a “quick and dirty” usability », in : *Usability evaluation in industry 189.3* (1996).

-
- [19] Gaëlle CALVARY et al., « Des widgets aux comets pour la Plasticité des Systèmes Interactifs », in : *Revue d'Interaction Homme Machine, Europa* 6. 1 (2005).
- [20] Gaëlle CALVARY et al., « Towards a new generation of widgets for supporting software plasticity : the" comet" », in : *IFIP International Conference on Engineering for Human-Computer Interaction*, Springer, 2004, p. 306–324.
- [21] Julien CASARIN, Dominique BECHMANN et Marilyn KELLER, « A Unified Model for Interaction in 3D Environment », in : *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, VRST '17*, Gothenburg, Sweden : Association for Computing Machinery, 2017, ISBN : 9781450355483, DOI : 10.1145/3139131.3139140, URL : <https://doi.org/10.1145/3139131.3139140>.
- [22] Julien CASARIN, Dominique BECHMANN et Marilyn KELLER, « A Unified Model for Interaction in 3D Environment », in : *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, VRST '17*, Gothenburg, Sweden : Association for Computing Machinery, 2017, ISBN : 9781450355483, DOI : 10.1145/3139131.3139140, URL : <https://doi.org/10.1145/3139131.3139140>.
- [23] Marjorie Barcella CHARRIER, « Ergonomie et design dans une démarche de conception de produits centrée sur les besoins des personnes », Theses, Université de Technologie de Belfort-Montbéliard, nov. 2016, URL : <https://tel.archives-ouvertes.fr/tel-01618328>.
- [24] Pierre CHEVAILLIER et al., « Semantic modeling of Virtual Environments using MASCARET », in : *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, IEEE, mar. 2012, p. 1–8, ISBN : 978-1-4673-1248-6, DOI : 10.1109/SEARIS.2012.6231174.
- [25] Guillaume CLAUDE, « Sequencement d'actions en environnement virtuel collaboratif », thèse de doct., Rennes, INSA, 2016.
- [26] Céline COUTRIX et Laurence NIGAY, « Mixed reality : a model of mixed interaction », in : *Proceedings of the working conference on Advanced visual interfaces*, 2006, p. 43–50.
- [27] Fred D DAVIS, « Perceived usefulness, perceived ease of use, and user acceptance of information technology », in : *MIS quarterly* (1989), p. 319–340.

-
- [28] Fred D DAVIS, « User acceptance of information technology : system characteristics, user perceptions and behavioral impacts », in : *International journal of man-machine studies* 38.3 (1993), p. 475–487.
- [29] Emmanuel DUBOIS, « Conception, Implémentation et Evaluation de Systèmes Interactifs Mixtes : une Approche basée Modèles et centrée sur l'Interaction », thèse de doct., 2009, p. 151.
- [30] Emmanuel DUBOIS, *IHM et Modélisation de Systèmes Interactifs Mixtes AccessiMap View project Thèse : conception et évaluation de techniques d'interaction optimisée pour la transmission d'information View project*, rapp. tech.
- [31] Emmanuel DUBOIS et Philip GRAY, « A design-oriented information-flow refinement of the ASUR interaction model », in : *IFIP International Conference on Engineering for Human-Computer Interaction*, Springer, 2007, p. 465–482.
- [32] Emmanuel DUBOIS, Philip GRAY et Laurence NIGAY, « ASUR++ : a design notation for mobile mixed systems », in : *International Conference on Mobile Human-Computer Interaction*, Springer, 2002, p. 123–139.
- [33] Emmanuel DUBOIS et al., « Design and evaluation of mixed interactive museographic exhibits. », in : *IJART* 4.4 (2011), p. 408–441.
- [34] K. Anders ERICSSON et Herbert A. SIMON, *Protocol analysis : Verbal reports as data*. Protocol analysis : Verbal reports as data. Pages : 426, Cambridge, MA, US : The MIT Press, 1984, ISBN : 0-262-55012-1 (Paperback).
- [35] Racim FAHSSI, Célia MARTINIE et Philippe PALANQUE, *HAMSTERS : un environnement d'édition et de simulation de modèles de tâches*, rapp. tech.
- [36] Racim FAHSSI, Célia MARTINIE et Philippe PALANQUE, *HAMSTERS : un environnement d'édition et de simulation de modèles de tâches*, rapp. tech.
- [37] Jean-Michel FAZZARI, Sébastien KUBICKI et Ronan QUERREC, « PRISME : Toward a model linking tangible interaction and domain-specific tasks in mixed reality environments », in : *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, IEEE, 2020, p. 210–219.

-
- [38] Rebecca FIEBRINK, Dan MORRIS et Meredith Ringel MORRIS, « Dynamic Mapping of Physical Controls for Tabletop Groupware », in : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, Boston, MA, USA : Association for Computing Machinery, 2009, p. 471–480, ISBN : 9781605582467, DOI : 10.1145/1518701.1518778, URL : <https://doi.org/10.1145/1518701.1518778>.
- [39] Rebecca FIEBRINK, Dan MORRIS et Meredith Ringel MORRIS, « Dynamic Mapping of Physical Controls for Tabletop Groupware », in : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'09, Boston, MA, USA : ACM, 2009, p. 471–480, ISBN : 978-1-60558-246-7.
- [40] G FITZMAURICE, Hiroshi ISHII et William BUXTON, « Bricks : Laying the foundations for Graspable User Interfaces. In proceedings of CHI'95 », in : *proceedings of CHI'95* (1995).
- [41] George W FITZMAURICE et William BUXTON, « An empirical evaluation of graspable user interfaces : towards specialized, space-multiplexed input », in : *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, 1997, p. 43–50.
- [42] Stéphanie FLECK et al., « Classifier les interfaces tangibles par la cladistique : critères et fiches pour un inventaire collaboratif », in : (oct. 2018), 12p.
- [43] Sean FOLLMER et al., « inFORM : dynamic physical affordances and constraints through shape and object actuation. », in : *Uist*, t. 13, 10.1145, 2013, p. 2501988–2502032.
- [44] Mauricio A FRIGO, Ethel CC da SILVA et Gustavo F BARBOSA, « Augmented reality in aerospace manufacturing : A review », in : *Journal of Industrial and Intelligent Information* 4.2 (2016).
- [45] Guillaume GAUFFRE et al., « Developing Mixed Interactive Systems : A Model-Based Process for Generating and Managing Design Solutions », in : Springer, London, oct. 2010, p. 183–208, DOI : 10.1007/978-1-84882-733-2_10.
- [46] Michael GRIEVES, « Digital twin : manufacturing excellence through virtual factory replication », in : *White paper* 1 (2014), p. 1–7.

-
- [47] Michael GRIEVES et John VICKERS, « Digital twin : Mitigating unpredictable, undesirable emergent behavior in complex systems », in : *Transdisciplinary perspectives on complex systems*, Springer, 2017, p. 85–113.
- [48] Anne-Cecilie HAUGSTVEDT et John KROGSTIE, « Mobile augmented reality for cultural heritage : A technology acceptance study », in : *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, IEEE, 2012, p. 247–255.
- [49] Vincent HAVARD, *Développement de méthodes et outils basés sur la réalité augmentée et virtuelle pour l'assistance ou l'apprentissage d'opérations dans un contexte industriel*, rapp. tech.
- [50] Eva HORNECKER et Jacob BUUR, « Getting a Grip on Tangible Interaction : A Framework on Physical Space and Social Interaction », in : *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06 (2006)*, p. 437, DOI : 10.1145/1124772.1124838.
- [51] Chun Hua HSIAO et Chyan YANG, « The intellectual development of the technology acceptance model : A co-citation analysis », in : *International Journal of Information Management* 31.2 (2011), p. 128–136.
- [52] Haochen HU et al., « Navigation in virtual and real environment using brain computer interface : a progress report », in : *Virtual Reality & Intelligent Hardware* 4.2 (2022), p. 89–114.
- [53] Hiroshi ISHII, « Tangible Bits : Beyond Pixels », in : *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, TEI '08, Bonn, Germany : Association for Computing Machinery, 2008, p. 15–25, ISBN : 9781605580043, DOI : 10.1145/1347390.1347392.
- [54] Hiroshi ISHII, « Tangible bits : beyond pixels », in : *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ACM, 2008, p. 15–25.
- [55] Hiroshi ISHII, « The tangible user interface and its evolution », in : *Communications of the ACM* 51.6 (2008), p. 32–36.
- [56] Hiroshi ISHII, Ali MAZALEK et Jay LEE, « Bottles as a minimal interface to access digital information », in : *CHI'01 extended abstracts on Human factors in computing systems*, 2001, p. 187–188.

-
- [57] Hiroshi ISHII, Sandia REN et Phil FREI, « Pinwheels : visualizing information flow in an architectural space », in : *CHI'01 extended abstracts on Human factors in computing systems*, 2001, p. 111–112.
- [58] Hiroshi ISHII et Brygg ULLMER, « Tangible bits : towards seamless interfaces between people, bits and atoms », in : *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, 1997, p. 234–241.
- [59] Sergey IVANOV et Evgeny BURNAEV, « Anonymous walk embeddings », in : *International conference on machine learning*, PMLR, 2018, p. 2186–2195.
- [60] Robert J.K. JACOB et al., « Reality-based interaction », in : *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, New York, New York, USA : ACM Press, 2008, p. 201, ISBN : 9781605580111, DOI : 10.1145/1357054.1357089.
- [61] Mustafa JARRAR, « Towards Automated Reasoning on ORM Schemes », in : *Conceptual Modeling - ER 2007*, sous la dir. de Christine PARENT et al., Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 181–197, ISBN : 978-3-540-75563-0.
- [62] Thomas [Mc Neill] JOHN S. GERO, « An approach to the analysis of design protocols », in : *Design Studies* 19.1 (1998), p. 21–61, ISSN : 0142-694X, DOI : [https://doi.org/10.1016/S0142-694X\(97\)00015-X](https://doi.org/10.1016/S0142-694X(97)00015-X).
- [63] Sébastien KUBICKI et al., « New human-computer interactions using tangible objects : Application on a digital tabletop with rfid technology », in : *International Conference on Human-Computer Interaction*, Springer, 2009, p. 446–455.
- [64] Jérémy LACOCHE, « Plasticity for user interfaces in mixed reality », Theses, Université Rennes 1, juil. 2016.
- [65] Jérémy LACOCHE, « Plasticity for user interfaces in mixed reality », Thèse de doctorat dirigée par Duval, Thierry Informatique Rennes 1 2016, thèse de doct., 2016, URL : <http://www.theses.fr/2016REN1S034>.
- [66] Jérémy LACOCHE, *Plasticity for user interfaces in mixed reality*, rapp. tech.
- [67] Jérémy LACOCHE et al., « Plasticity for 3d user interfaces : new models for devices and interaction techniques », in : *Proceedings of the 7th ACM SIGCHI symposium on engineering interactive computing systems*, 2015, p. 28–33.

-
- [68] Jérémy LACOCHE et al., « Redistribution et Plasticité pour les Interfaces Utilisateurs 3D : un Modèle Illustré », in : (oct. 2015).
- [69] Jae Yeol LEE, Dong Woo SEO et Gue Won RHEE, « Tangible authoring of 3D virtual scenes in dynamic augmented reality environment », in : *Computers in Industry* 62.1 (2011), p. 107–119, ISSN : 0166-3615, DOI : <https://doi.org/10.1016/j.compind.2010.07.003>, URL : <http://www.sciencedirect.com/science/article/pii/S0166361510001077>.
- [70] Jae Yeol LEE, Dong Woo SEO et Gue Won RHEE, « Tangible authoring of 3D virtual scenes in dynamic augmented reality environment », in : *Computers in Industry* 62.1 (2011), p. 107–119.
- [71] Jaewook LEE et al., « User Preference for Navigation Instructions in Mixed Reality », in : *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2022, p. 802–811, DOI : 10.1109/VR51125.2022.00102.
- [72] Daniel LEITHINGER et al., « Physical telepresence : shape capture and display for embodied, computer-mediated remote collaboration », in : *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, p. 461–470.
- [73] Daniel LEITHINGER et al., « Shape displays : Spatial interaction with dynamic physical form », in : *IEEE computer graphics and applications* 35.5 (2015), p. 5–11.
- [74] Dominique LENNE, Indira THOUVENIN et Stéphane AUBRY, « Supporting design with 3D-annotations in a collaborative virtual environment », in : *Research in engineering design* 20.3 (2009), p. 149–155.
- [75] Sophie LEPREUX et al., « From Centralized interactive tabletops to Distributed surfaces : the Tangiget concept », in : *International Journal of Human-Computer Interaction* 28.11 (2012), p. 709–721.
- [76] Sophie LEPREUX et al., « Interaction Tangible sur Table, définitions et modèles », in : *Journal d'Interaction Personne-Système* 5.1 (2016), p. 1–21.
- [77] Catherine LETONDAL et al., « Being in the Sky », in : *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '18*, New York, New York, USA : ACM Press, 2018, p. 656–666, ISBN :

9781450355681, DOI : 10.1145/3173225.3173229, URL : <http://dl.acm.org/citation.cfm?doid=3173225.3173229>.

- [78] Catherine LETONDAL et al., « Being in the Sky », in : *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '18*, New York, New York, USA : ACM Press, 2018, p. 656–666, ISBN : 9781450355681, DOI : 10.1145/3173225.3173229.
- [79] Catherine LETONDAL et al., « Being in the Sky : Framing Tangible and Embodied Interaction for Future Airliner Cockpits », in : (2018), DOI : 10.1145/3173225.3173229.
- [80] Catherine LETONDAL et al., « Exploring the Physical Design Space for Situation Awareness and Performance in the Interactive Cockpit », in : *TEI 2014, 8th International Conference on Tangible, Embedded and Embodied Interaction*, 2014.
- [81] James R LEWIS, « Comparison of Four TAM Item Formats : Effect of Response Option Labels and Order. », in : *Journal of Usability Studies* 14.4 (2019).
- [82] Daniel LEWKOWICZ, Guillaume CALVET et Guillaume JEGOU, « Ngagement–Rapid design and evaluation of complex systems in virtual reality settings », in : (2020).
- [83] Domitile LOURDEAUX et al., « HUMANS : suite logicielle pour la scenarisation d’environnements virtuels pour la formation à des situations socio-techniques complexes », in : -, 2017.
- [84] Vincent LUCQUIAUD, « Proposition d’un noyau et d’une structure pour les modèles de tâches orientés utilisateurs », in : *Proceedings of the 17th Conference on l’Interaction Homme-Machine*, ACM, 2005, p. 83–90.
- [85] Andrés Saraos LUNA, Valérie GOURANTON et Bruno ARNALDI, « Collaborative virtual environments for training : A unified interaction model for real humans and virtual humans », in : *E-Learning and Games for Training, Education, Health and Sports*, Springer, 2012, p. 1–12.
- [86] Pierre MAHIEUX et al., « SABLIER : a Tangible Interactor to Navigate through Space and Time », in : *CHI Conference on Human Factors in Computing Systems*, 2022, p. 1–12.

-
- [87] Hanspeter A MALLOT, « Behavior-oriented approaches to cognition : Theoretical perspectives », in : *Theory in biosciences* 116.3 (1997), p. 196–220.
- [88] Andrew MANCHES, Claire O'MALLEY et Steve BENFORD, « Physical manipulation : evaluating the potential for tangible designs », in : *Proceedings of the 3rd international conference on tangible and embedded interaction*, 2009, p. 77–84.
- [89] Ali MAZALEK et Michael NITSCHKE, « Tangible Interfaces for Real-Time 3D Virtual Environments », in : *Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE '07*, Salzburg, Austria : Association for Computing Machinery, 2007, p. 155–162, ISBN : 9781595936400, DOI : 10.1145/1255047.1255080, URL : <https://doi.org/10.1145/1255047.1255080>.
- [90] Ali MAZALEK et Michael NITSCHKE, « Tangible interfaces for real-time 3D virtual environments », in : *Proceedings of the international conference on Advances in computer entertainment technology*, ACM, 2007, p. 155–162.
- [91] G. MORI, F. PATERNO et C. SANTORO, « CTTE : support for developing and analyzing task models for interactive system design », in : *IEEE Transactions on Software Engineering* 28.8 (2002), p. 797–813.
- [92] Odile MORNAS et al., « Development of Systems Engineering People to support Major Transformation Plans in Thales (Process, Roles, Methodology & related tools) », in : *INCOSE International Symposium 22* (juil. 2012), p. 1884–1898, DOI : 10.1002/j.2334-5837.2012.tb01444.x.
- [93] Ken NAKAGAKI et al., « Materiable : Rendering dynamic material properties in response to direct physical touch with shape changing interfaces », in : *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, p. 2764–2772.
- [94] Donald A NORMAN, *The psychology of everyday things*. Basic books, 1988.
- [95] Fabio PATERNÒ, Cristiano MANCINI et Silvia MENICONI, « ConcurTaskTrees : A Diagrammatic Notation for Specifying Task Models », in : *ConcurTaskTrees : A Diagrammatic Notation for Specifying Task Models*, jan. 1997, p. 362–369, DOI : 10.1007/978-0-387-35175-9_58.

-
- [96] Sylvain PAUCHET et al., « GazeForm », in : *Proceedings of the 2018 on Designing Interactive Systems Conference 2018 - DIS '18*, New York, New York, USA : ACM Press, juin 2018, p. 1193–1205, ISBN : 9781450351980, DOI : 10.1145/3196709.3196712.
- [97] Sylvain PAUCHET et al., « GazeForm : Dynamic Gaze-adaptive Touch Surface for Eyes-free Interaction in Airliner Cockpits », in : (2018), DOI : 10.1145/3196709.3196712, URL : <https://doi.org/10.1145/3196709.3196712>.
- [98] Sylvain PAUCHET et al., « GazeForm : Dynamic Gaze-adaptive Touch Surface for Eyes-free Interaction in Airliner Cockpits », in : (2018), DOI : 10.1145/3196709.3196712.
- [99] Carl Adam PETRI, « Communication with automata », in : (1966).
- [100] Christine PIERRET-GOLBREICH et Dominique SCAPIN, « MAD : Méthode Analytique de Description des tâches », in : *Colloque sur l'ingénierie des interfaces homme-machine*, 1989.
- [101] Ronan QUERREC, Paola VALLEJO et Cédric BUCHE, « MASCARET : creating virtual learning environments from system modelling », in : *The Engineering Reality of Virtual Reality 2013*, sous la dir. de Margaret DOLINSKY et Ian E. MCDOWALL, t. 8649, International Society for Optics et Photonics, SPIE, 2013, p. 21–31, DOI : 10.1117/12.2008186.
- [102] Ronan QUERREC, Paola VALLEJO et Cédric BUCHE, « MASCARET : creating virtual learning environments from system modelling », in : *The Engineering Reality of Virtual Reality 2013*, t. 8649, 2013, p. 41–61.
- [103] Pierre RAIMBAUD et al., « A Task-Centred Methodology to Evaluate the Design of Virtual Reality User Interactions : A Case Study on Hazard Identification », in : *Buildings* 11.7 (2021), p. 277.
- [104] Killian RICHARD et al., « INTERVALES : INTERActive Virtual and Augmented framework for industrial Environment and Scenarios », in : *Advanced Engineering Informatics* 50 (2021), p. 101425.
- [105] Kimiko RYOKAI, Stefan MARTI et Hiroshi ISHII, « I/O brush : drawing with everyday objects as ink », in : *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, p. 303–310.

-
- [106] Marc Aurel SCHNABEL et Thomas KVAN, « Design, communication & collaboration in immersive virtual environments », in : (2002).
- [107] Philipp SCHOESSLER et al., « Kinetic blocks : Actuated constructive assembly for interaction and display », in : *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, p. 341–349.
- [108] Orit SHAER et Eva HORNECKER, « Tangible User Interfaces : Past, Present, and Future Directions », in : *Foundations and Trends® in Human–Computer Interaction* 3.1-2 (2009), p. 1–137, ISSN : 1551-3955, DOI : 10.1561/1100000026.
- [109] Orit SHAER et al., « The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data », in : *International Journal of Human-Computer Studies* 70.10 (2012), p. 746–764.
- [110] Orit SHAER et al., « The TAC paradigm : specifying tangible user interfaces », in : *Personal and Ubiquitous Computing* 8.5 (2004), p. 359–369.
- [111] Erik STEINDECKER et al., « Expanding VRPN to Tasks in Virtual Engineering I4D : Integrativer Informations-und Interaktionsraum für Industrie 4.0 in Dresden View project », in : (2014), DOI : 10.1115/DETC2014-34277.
- [112] *Expanding VRPN to Tasks in Virtual Engineering*, t. Volume 1B : 34th Computers and Information in Engineering Conference, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, V01BT02A026, août 2014, DOI : 10.1115/DETC2014-34277, eprint : <https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2014/46292/V01BT02A026/4254871/v01bt02a026-detc2014-34277.pdf>.
- [113] *Expanding VRPN to Tasks in Virtual Engineering*, t. Volume 1B : 34th Computers and Information in Engineering Conference, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, V01BT02A026, août 2014, DOI : 10.1115/DETC2014-34277.
- [114] Ralph STELZER et al., « Expanding vrpn to tasks in virtual engineering », in : *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection, 2014.

-
- [115] Russell M. TAYLOR II et al., « VRPN : A Device-independent, Network-transparent VR Peripheral System », in : *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '01, Baniff, Alberta, Canada : ACM, 2001, p. 55–61, ISBN : 1-58113-427-4.
- [116] Romain TERRIER et al., « Scenario-based VR Framework for Product Design », in : *VISIGRAPP, GRAPP, the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Valletta, Malta, 2020, URL : <https://hal.archives-ouvertes.fr/hal-02502648>.
- [117] David THEVENIN et Joëlle COUTAZ, « Plasticity of User Interfaces : Framework and Research Agenda. », in : *Interact*, t. 99, 1999, p. 110–117.
- [118] Brygg Anders ULLMER et Hiroshi ISHII, « Tangible interfaces for manipulating aggregates of digital information », thèse de doct., Citeseer, 2002.
- [119] Brygg ULLMER et Hiroshi ISHII, « Emerging frameworks for tangible user interfaces », in : *IBM systems journal* 39.3.4 (2000), p. 915–931.
- [120] Brygg ULLMER, Hiroshi ISHII et Robert JK JACOB, « Token+ constraint systems for tangible interaction with digital information », in : *ACM Transactions on Computer-Human Interaction (TOCHI)* 12.1 (2005), p. 81–118.
- [121] Brygg ULLMER et al., *Tangible Menus and Interaction Trays : Core tangibles for common physical/digital activities*, rapp. tech., 2008.
- [122] Brygg ULLMER et al., *Tangible Menus and Interaction Trays : Core tangibles for common physical/digital activities*, rapp. tech., 2008.
- [123] Brygg ULLMER et al., « Tangible menus and interaction trays : core tangibles for common physical/digital activities », in : *Proceedings of the 2nd international conference on Tangible and embedded interaction*, 2008, p. 209–212.
- [124] Gerrit VAN DER VEER et Martijn van WELIE, « Task based groupware design : putting theory into practice », in : *Proceedings of the 3rd conference on Designing interactive systems : processes, practices, methods, and techniques*, 2000, p. 326–337.
- [125] Jean-Luc VINOT et Sylvie ATHENES, « Legible, are you sure ? an experimentation-based typographical design in safety-critical context », in : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, p. 2287–2296.

-
- [126] Jean-Luc VINOT et al., « Could tangibility improve the safety of touch-based Interaction ? », in : (sept. 2016), URL : <https://hal-enac.archives-ouvertes.fr/hal-01367011/>.
- [127] Jean-Luc VINOT et al., « Could tangibility improve the safety of touch-based Interaction ? », in : (sept. 2016).
- [128] Jean-Luc VINOT et al., « Could tangibility improve the safety of touch-based Interaction ? Could tangibility improve the safety of touch-based Interaction ? Exploring a new Physical Design Space for Pilot-System Interfaces », in : (2016), DOI : 10.1145/2950112.2964581.
- [129] Simon VOELKER et al., « Knobology revisited : A comparison of user performance between tangible and virtual rotary knobs », in : *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, 2015, p. 35–38.
- [130] David D WALDEN et al., *INCOSE Systems Engineering Handbook version 4 : A guide for system life cycle processes and activities*, Wiley Online Library, 2015.
- [131] Xiangyu WANG et Phillip S DUNSTON, « System evaluation of a mixed reality-based collaborative prototype for mechanical design review collaboration », in : *Computing in Civil Engineering (2005)*, 2005, p. 1–9.
- [132] Michelle YEH et al., *Human factors considerations in the design and evaluation of flight deck displays and controls*, rapp. tech., John A. Volpe National Transportation Systems Center (US), 2013.
- [133] O ZUCKERMAN, « Tutorial Prototyping tangible interfaces », in : ECSCW, 2005.
- [134] Oren ZUCKERMAN et Ayelet GAL-OZ, « To TUI or not to TUI : Evaluating performance and preference in tangible vs. graphical user interfaces », in : *International Journal of Human-Computer Studies* 71.7-8 (2013), p. 803–820.

Titre : Modèle de reconnaissance d'actions à partir des interactions en Réalité Mixte et Tangibles

Mot clés : Interaction, tangible, réalité mixte, activité, modèle

Resumé : Nous présentons dans cette thèse un modèle de reconnaissance d'action métier à partir des interactions utilisateur en Environnement Virtuel et Tangible. Notre objectif est de faire le lien automatique (c'est-à-dire sans développement informatique spécifique) entre les interactions réalisées par un utilisateur via des périphériques ou des objets tangibles dans un environnement virtuel et les actions opérateurs décrites dans les activités métier. Pour cela, nous avons dans un premier temps conçu un modèle générique d'interaction en nous appuyant sur les modèles d'interaction en environnement virtuel existants, un algorithme générique permettant de reconnaître l'action métier réalisée à partir des interactions de l'utilisateur ainsi qu'une typologie d'interacteurs permettant de décrire la grande majorité des interacteurs en réalité virtuelle et augmentée. Nous avons implémenté notre modèle de reconnaissance dans un cas d'usage aéronautique où les interactions portent sur des activités de pilotage et validé celui-ci par une évaluation mesurant son acceptation auprès d'ergonomes.

Title : Action recognition model from interactions in Mixed and Tangible Reality

Keywords : Interaction, tangible, mixed reality, activity, model

Abstract : We present in this thesis a business action recognition model from user interactions in Virtual and Tangible Environment. Our objective is to make an automatic link (that is to say without specific IT development) between the interac-

tions carried out by a user via peripherals or tangible objects in a virtual environment and the operator actions described in the business activities. To do this, we first designed a generic interaction model based on existing virtual environment interaction models, a generic algorithm to recognize the business action performed from user

interactions as well as a typology of interactors allowing to describe the vast majority of interactors in virtual and augmented reality. We implemented our recognition model in an aeronautical use case where the interactions relate to piloting activities and validated it by an evaluation measuring its acceptance with ergonomists.