



HAL
open science

Proposition d'un modèle de données pour l'édition et l'exploitation d'Environnements Virtuels ou Augmentés dans un contexte industriel.

Killian Richard

► **To cite this version:**

Killian Richard. Proposition d'un modèle de données pour l'édition et l'exploitation d'Environnements Virtuels ou Augmentés dans un contexte industriel.. Informatique. HESAM Université, 2022. Français. NNT : 2022HESAE046 . tel-03811289

HAL Id: tel-03811289

<https://pastel.hal.science/tel-03811289>

Submitted on 11 Oct 2022

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.

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
[LINEACT CESI – Campus de Paris]

THÈSE

présentée par : **Killian RICHARD**

soutenance prévue le : **07 septembre 2022**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Spécialité : **Réalité Virtuelle et Augmentée**

Proposition d'un modèle de données pour l'édition et l'exploitation d'Environnements Virtuels ou Augmentés dans un contexte industriel.

THÈSE dirigée par :

M. BAUDRY David

et co-encadrée par :

M. HAVARD Vincent

Jury

M. Simon RICHIR, Professeur, LAMPA, ENSAM

Mme Domitile LOURDEAUX, HDR, HEUDIASYC, UTC

M. Ronan QUERREC, Professeur, Lab-STICC, ENIB

M. Jean-Pierre JESSEL, Professeur, IRIT, Univ. Paul Sabatier

M. David BAUDRY, Directeur de Recherche, LINEACT, CESI

M. Vincent HAVARD, Enseignant/Chercheur, LINEACT, CESI

Président

Rapportrice

Rapporteur

Examineur

Examineur

Examineur

« Le virtuel est ce qui nous aide à faire advenir ce que nous ne sommes pas encore. »

- Philippe Quéau

Table des matières

Remerciements	7
Résumé.....	8
Abstract.....	9
Glossaire.....	10
Introduction	11
1 Contexte du projet.....	13
1.1 Contexte scientifique - les usages de la réalité virtuelle et augmentée dans l'industrie	13
1.1.1 L'industrie 4.0.....	13
1.1.2 Définition de la réalité virtuelle	15
1.1.3 Usages de la réalité virtuelle.....	16
1.1.4 Définition de la réalité augmentée	19
1.1.5 Usages de la réalité augmentée en Industrie	19
1.1.6 Synthèse	22
1.2 Contexte industriel - Étude et analyse au sein d'OREKA Ingénierie	24
1.2.1 Définitions et objectifs de l'analyse	24
1.2.2 Résultats obtenus.....	25
1.2.3 Synthèse	29
1.3 Synthèse des observations	30
2 État de l'art	32
2.1 Introduction	32
2.2 Modèles et Frameworks pour la réalité virtuelle et augmentée.....	33
2.2.1 Modèles orientés réalité virtuelle.....	33
2.2.1.1 Modèles d'environnements.....	33
2.2.1.2 Framework et modèles de scénarios	39
2.2.2 Modèles orientés réalité augmentée	47
2.2.3 INOOVAS : Un framework compatible RA/RV.....	49
2.2.4 Synthèse des modèles	51
2.3 Formalisation du séquençement sous forme de graphe	54
2.3.1 Introduction aux représentations sous forme de graphe orienté.....	55
2.3.2 Machine à états	55

2.3.3	Diagramme d'activité (UML)	56
2.3.4	Les réseaux de Petri	57
2.3.5	Grafcet	59
2.3.6	Grafchart	61
2.3.7	CTT	63
2.3.8	AND/OR Graphs	64
2.3.9	Représentations propriétaires	65
2.3.10	Synthèse	67
2.4	Problématique scientifique	70
3	INTERVALES : Proposition du modèle	72
3.1	Introduction	72
3.2	INTERVALES : Architecture et workflow	72
3.2.1	INTERVALES pour la création de contenu	73
3.2.2	INTERVALES pour la scénarisation en réalité virtuelle et augmentée	74
3.3	INTERVALES : Un modèle orienté "Features-Relations"	76
3.3.1	Définition d'environnements	76
3.3.1.1	Entity et Feature	76
3.3.1.2	Action et Relation	79
3.3.2	Définition de scénarios et procédures	80
3.3.2.1	Scenario, Objective et Task	80
3.3.2.2	INTERVALES : Formalisation du séquençement	81
3.4	Architecture et développement du framework INTERVALES	86
3.4.1	Librairies INTERVALES	87
3.4.2	Intégration dans Unity 3D : INTERVALES.UnityPackage	90
3.5	Synthèse	94
4	INTERVALES : Mise en application	96
4.1	Introduction	96
4.2	INTERVALES : Utilisation du modèle au sein du processus de conception	96
4.3	Cas d'usages industriels	97
4.3.1	LeanManufacturing VR : Assemblage sur postes manuels	98
4.3.1.1	Description du cas d'usage	98
4.3.1.2	Architecture utilisée avec INTERVALES	99
4.3.1.3	Création de l'environnement au formalisme INTERVALES	101
4.3.1.4	Édition de la procédure d'assemblage par un développeur	101

4.3.1.5	Validation de la procédure d'assemblage par le ScenarioManager en réalité virtuelle	102
4.3.1.6	Mise en place de la réalité augmentée	103
4.3.1.7	Synthèse.....	104
4.3.2	Mises en application dans les projets de l'entreprise.....	105
4.3.2.1	Tutoriel OREKA VR : Escape Game d'initiation à la réalité virtuelle.....	105
4.3.2.2	Salle virtuelle pour la formation aux mesures en physique nucléaire.....	106
4.3.2.3	Application de formation à l'assemblage d'un équipement en réalité virtuelle	107
4.3.3	Synthèse	107
5	INTERVALES : Édition de scénarios et de procédures métiers	109
5.1	Introduction	109
5.2	Édition de scénarios	109
5.2.1	Définition des besoins	109
5.2.2	Premiers développements réalisés	109
5.2.3	Étude préliminaire.....	111
5.2.4	Synthèse	112
5.3	INTERVALES : vers un outil d'édition par démonstration (Authoring-by-doing).....	113
5.4	Développement d'un nouvel éditeur 2D et d'un éditeur en authoring-by-doing	116
5.5	Expérimentations et résultats	118
5.5.1	Objectif de l'expérimentation.....	118
5.5.2	Définition du protocole expérimental	119
5.5.2.1	Variables étudiées.....	119
5.5.2.2	Participants	119
5.5.2.3	Cas d'usages utilisé et découpage de l'expérimentation.....	119
5.5.2.4	Déroulement de l'expérimentation.....	121
5.5.3	Méthodes d'évaluations utilisées.....	123
5.5.3.1	Pré-questionnaire d'évaluation du niveau des participants	123
5.5.3.2	Évaluation de l'édition de scénario	123
5.5.3.3	Évaluation de l'expérience utilisateur	126
5.5.3.4	Évaluation du retour d'expérience	127
5.5.4	Analyse des résultats obtenus et discussions	127
5.5.4.1	Résultats du pré-questionnaire d'évaluation du niveau des participants	127
5.5.4.2	Remarques sur les résultats entre les deux passages de l'expérience 1	128
5.5.4.3	Résultats de l'édition de scénario.....	135

5.5.4.4	Résultats du questionnaire sur l'expérience utilisateur.....	157
5.5.4.5	Analyse des retours d'expérience.....	162
5.6	Conclusion.....	163
6	Conclusion Générale.....	165
6.1	Travaux réalisés.....	165
6.2	Perspectives.....	168
7	Publications.....	171
8	Table des figures.....	172
9	Liste des tableaux.....	180
10	Annexes.....	185
10.1	Groupes technologiques et les ressources support de l'industrie 4.0.....	185
10.2	Fiches d'instructions d'assemblage.....	186
10.3	Questionnaire System-Usability-Scale.....	189
10.4	Questionnaire UEQ.....	190
10.5	Questionnaire final.....	191
11	Bibliographie.....	193

Remerciements

Je souhaite, tout d'abord, remercier CESI et plus particulièrement son laboratoire de recherche LINEACT, et son directeur de la recherche et de l'innovation, Bélahcène MAZARI, ainsi qu'Anne LOUIS, responsable régionale du département recherche pour m'avoir permis d'effectuer mes travaux de recherche au sein du laboratoire de Rouen.

Je tiens également à grandement remercier Didier DUFFULER et Pascal POMMIER pour m'avoir accueilli au sein de leur entreprise, OREKA Ingénierie, et pour m'avoir donné l'opportunité de réaliser une thèse CIFRE dans un contexte industriel concret. Je les remercie de m'avoir fait confiance durant ces 3 ans de thèse.

Je souhaite exprimer également une grande gratitude envers mon équipe d'encadrement, mon directeur de thèse, David BAUDRY, ainsi que mon co-encadrant, Vincent HAVARD pour avoir cru en moi en me proposant ce sujet de thèse. Ces 3 années de thèse ont été une expérience très forte, tant sur le plan professionnel et scientifique que sur le plan personnel et il n'est pas exagéré de dire que leur soutien, leur accompagnement, leur disponibilité ont contribué à l'aboutissement de ce projet de thèse. J'ai apprécié vos retours constructifs, votre écoute et votre bienveillance, rendant toujours nos échanges agréables.

Je souhaite aussi remercier tous mes collègues et amis, aussi bien de l'équipe CESI que de l'équipe OREKA Ingénierie pour leur soutien et leur bonne humeur. J'ai fortement apprécié travailler dans ce contexte aussi bien professionnel, qu'amical.

Enfin, je souhaite remercier avec beaucoup d'amour ma famille, mes parents, Sophie et Patrice ainsi que mon frère et ma sœur, Titouan et Morgane pour leur soutien durant ces trois années de thèse et sans qui, je n'en serais pas là.

Résumé

Dans le contexte de l'industrie du futur, les évolutions technologiques et numériques ont permis de mettre en place de nouvelles manières de concevoir, de collaborer ou encore de se former. Le développement des outils de CAO (Conception Assistée par Ordinateur), des outils de simulations métiers et plus récemment des Jumeaux Numériques de systèmes industriels permettent de travailler sur des modèles numériques du système, donnant une représentation multi-physiques de celui-ci. Ces modèles numériques peuvent être intégrés et interfacés avec des environnements en Réalité Virtuelle ou Augmentée (RV/RA). Ils permettent ainsi d'améliorer l'efficacité des parties prenantes dans les différentes phases de conception, d'exploitation ou de maintenance d'un système telles que, par exemple, la revue de conception, l'étude de processus, l'assistance ou la formation des opérateurs. Enfin, avec l'arrivée massive des casques immersifs et de nouveaux dispositifs de RA, ces technologies se démocratisent dans les entreprises et deviennent accessibles aux PME qui veulent pouvoir en tirer parti. Cependant, l'intégration de procédures métiers et de scénarios industriels dans les environnements virtuels et augmentés reste encore un processus complexe et peu industrialisé nécessitant des compétences diverses telles que : des compétences métiers pour la définition des procédures et scénarios, des compétences informatiques pour générer l'environnement virtuel et gérer les interactions de l'utilisateur, ... Par ailleurs, la diversité des appareils de rendu et des interactions associés complexifie d'autant plus le processus de création impliquant des temps de développement long et coûteux.

Afin de proposer des éléments de réponses à ces différentes problématiques industrielles, nous avons tout d'abord dressé un état de l'art scientifique et technologique portant sur les modèles UML et les approches basées sur les ontologies pour la formalisation de processus ou de scénarios métiers. En s'appuyant sur cette analyse et sur l'expertise et les retours d'expériences d'OREKA ingénierie, les travaux de cette thèse ont permis de proposer le modèle de données INTERVALES (INTERActive Virtual and Augmented framework for industrial Environments and Scenarios). Ce modèle permet de formaliser la création des contenus et interactions en réalité virtuelle et en réalité augmentée liés aux procédures et scénarios métiers de systèmes industriels. Une attention particulière a été portée sur l'architecture logicielle du framework INTERVALES afin de permettre une intégration avec les outils du jumeaux numériques (moteur de rendu 3D type Unity, outils de CAO et de simulation, ...). Des cas d'applications intégrant le framework INTERVALES ont été étudiés et montrent notamment la diversité des usages auxquels le framework peut répondre : implantation d'atelier de production et revue de conception de poste cobotique en RV, assistance ou formation à des gammes d'assemblages en RA/RV, formations dans le domaine du nucléaire,

Enfin, la dernière contribution porte sur les interfaces graphiques du framework INTERVALES permettant d'éditer les procédures et scénarios en définissant les tâches attendues et les paramètres associés. Ces travaux ont comparé deux éditeurs distincts : une interface utilisateur 2D basée sur des graphes de nœuds et une interface utilisateur en RV permettant l'édition par démonstration (authoring-by-doing). Les résultats des études expérimentales et des analyses conduites permettent de conclure sur l'efficacité de l'édition par démonstration pour des procédures séquentielles et de l'intérêt d'une approche hybride pour des procédures plus complexes.

Mots-clés : Réalité virtuelle, Réalité augmentée, Industrie du futur, Modélisation UML, Édition de scénario et procédures métiers, Édition par la démonstration (Authoring-by-doing)

Abstract

In the context of the factory of the future, technological and digital developments have enabled new ways of designing, collaborating and even training. The development of CAD (Computer Aided Design) tools, business simulation tools and more recently Digital Twins of industrial systems, make it possible to work on digital models of the system, giving a multi-physical representation of it. These digital models can be integrated and interfaced with Virtual or Augmented Reality (VR/AR) environments. They thus make it possible to improve the efficiency of the stakeholders in the various phases of design, operations, or maintenance of a system, such as design review, process study, assistance, or operator training. Finally, with the mass arrival of immersive headsets and new AR devices, these technologies are becoming increasingly democratic within companies and becoming accessible to SMEs who want to be able to benefit from them. However, the integration of business procedures and industrial scenarios in virtual and augmented environments remains a complex and scarcely industrialized process requiring various skills such as: business skills for the definition of procedures and scenarios, computer skills to generate the virtual environment and manage user interactions, Furthermore, the diversity of rendering devices and associated interactions further complicates the creation process involving long and costly development times.

In order to propose elements of answers to these various industrial problems, we first drew up a state of the scientific and technological art relating to UML models and approaches based on ontologies for the formalization of processes or business scenarios. Based on this analysis and on the expertise and feedback from OREKA Ingenierie, the work of this thesis has made it possible to propose the INTERVALES data model (INTERactive Virtual and Augmented framework for industrial Environments and Scenarios). This model makes it possible to formalize the creation of content and interactions in virtual and augmented reality linked to the procedures and business scenarios of industrial systems. Particular attention has been paid to the software architecture of the INTERVALES framework to allow for integration with digital twin tools (Unity-type 3D rendering engine, CAD and simulation tools, etc.). Cases of applications integrating the INTERVALES framework have been studied and demonstrate the diversity of uses with which the framework can respond, such as: the establishment of production workshop and design reviews of cobotic stations in VR, providing assistance or training in assembly ranges in AR/VR, supplementary training in the nuclear field, etc.

Finally, the last contribution concerns the graphic interfaces of the INTERVALES framework which allows users to directly edit procedures and scenarios by defining the expected tasks and associated parameters. This work compared two distinct editors: a 2D user interface based on node graphs and a VR user interface allowing authoring-by-doing. The results of the experimental studies and subsequent analyses make it possible to conclude on the effectiveness of editing by demonstration for sequential procedures and the interest of a hybrid approach for more complex procedures.

Keywords: Virtual reality, Augmented reality, Factory of the future, UML Modelling, Scenario authoring and job procedures, Authoring-by-doing

Glossaire

CAO	Conception assistée par ordinateur
CAVE	Cave Automatic Virtual Environment ou salle immersive
Cobotique	Robotique collaborative
EVAH	Environnement Virtuel d'Apprentissage Humain
INTERVALES	INTER active V irtual and A ugmented framework for industrial L Environments and S cenarios
RA (AR)	Réalité Augmentée (Augmented Reality)
RV (VR)	Réalité Virtuelle (Virtual Reality)
EV (VE)	Environnement Virtuel (Virtual Environment)

Introduction

Dans le contexte de l'industrie 4.0 et des systèmes de production cyber-physique anthropocentriques (Rauch et al., 2020), l'intégration d'interfaces utilisateur centrées sur l'humain (Raisamo et al., 2019), basées sur les technologies de réalité étendue (XR) telles que la réalité virtuelle (RV) et la réalité augmentée (RA) - fournit une aide cognitive, une assistance numérique et des outils collaboratifs. Ces technologies peuvent avoir diverses applications dans les processus de conception, de planification, de production et de maintenance industrielle (Damiani et al., 2018; Geng et al., 2020; Rauch et al., 2020). Les applications RV, avec leurs capacités immersives, créent la sensation de présence de l'utilisateur dans l'environnement virtuel (VE) et la possibilité d'interagir avec le jumeau numérique (Havard, Jeanne, et al., 2019). Ces technologies sont utilisées dans diverses activités des processus industriels en tant qu'aide cognitive pour : la conception et la planification de produits et de processus (Damiani et al., 2018; Rauch et al., 2020), la collaboration à distance (Galambos et al., 2015), l'évaluation de la sécurité et de l'ergonomie des postes de travail cobotiques (Havard, Jeanne, et al., 2019; Pérez et al., 2019), le partage des connaissances et la formation (Berg & Vance, 2017; Stone, 2001), ou encore la sensibilisation des utilisateurs aux procédures de sécurité (Matsas & Vosniakos, 2017). En complément, la capacité de la RA à ajouter du contenu virtuel à l'espace physique est utilisée pour fournir une assistance visuelle à un opérateur dans le cas d'une gamme de maintenance, de contrôle qualité, d'assistance au montage et de logistique (Funk, 2016; Havard et al., 2016; Henderson & Feiner, 2011; Vanneste et al., 2020). En effet, ces technologies permettent d'améliorer la qualité et le partage des connaissances entre le formateur et l'apprenant d'une part, et entre l'expert et l'opérateur d'autre part. Cependant, la création de contenu pour ces outils nécessite des informations et des connaissances approfondies de l'entreprise, des compétences métiers (par exemple, des experts des processus industriels et des formateurs) (Badets et al., 2020), et de solides compétences en développement informatique. Ainsi, il devient impératif de définir des modèles et outils pour faciliter la création de contenu, que ce soit pour définir le niveau d'interaction de l'environnement, ou pour orchestrer des tâches dans des scénarios de formation et d'activité tels que l'assemblage, la maintenance ou encore l'inspection.

Dans ce mémoire de thèse, une étude du contexte scientifique et du contexte industriel des usages de la réalité virtuelle et augmentée permet de positionner les travaux de recherche et les besoins industriels. Cette thèse s'inscrivant dans le cadre d'une collaboration avec l'entreprise OREKA Ingénierie, cette étude s'intéresse notamment aux besoins « proches du terrain » pour le développement des applications RA et RV. Ensuite, une analyse des modèles et frameworks existants et permettant la conception des environnements et scénarios en RA/RV est effectuée en croisant avec les besoins de l'industrie. Cette analyse identifie les exigences et les limites de ces modèles. Certains modèles existants, bien qu'avantageux à certains égards, ne sont pas forcément évolutifs lorsque l'on tente d'ajouter de nouvelles fonctionnalités telles que des modes collaborations en RA/RV ou l'intégration d'agents autonomes liés au jumeau numérique du

système étudié. De même, la plupart des modèles de scénarios examinés sont dédiés à des scénarios procéduraux (c'est-à-dire, étape par étape). En général, il reste également le besoin d'un modèle de création de contenu prenant en charge la réalité virtuelle et la réalité augmentée. Par conséquent, la principale contribution est de proposer un nouveau framework INTERVALES (**INTER**active **V**irtual and **A**ugmented framework for industrial **E**nvironments and **S**cenarios), qui :

- (i) permet la création de contenu pour les applications RV/RA ;
- (ii) facilite la transformation des connaissances existantes en scénarios compatibles RV/RA ;
- (iii) est interopérable avec des outils tiers.

Pour illustrer ce cadre, plusieurs cas d'utilisation industriels sont présentés comme par exemple l'utilisation d'interfaces en RV/RA couplées au jumeau numérique d'un processus d'assemblage industriel intégrant des postes de travail cobotiques (robotique collaborative) et manuels. Enfin, ce travail s'intéresse au processus de création de scénario à la fois via une interface graphique utilisateur ainsi qu'en utilisant l'« Authoring-by-doing » (édition par démonstration) qui consiste à créer un scénario en le réalisant.

Ce manuscrit est divisé en cinq chapitres :

- Le chapitre **1** pose le contexte général du projet de recherche.
- Le chapitre **2** propose une étude des travaux existants sur la création et l'édition d'environnements et de scénario en RV et/ou RA.
- Le chapitre **3** présente le framework INTERVALES
- Le chapitre **4** détaille l'utilisation de notre modèle dans plusieurs cas d'usages industriels.
- Le chapitre **5** détaille le processus d'édition de scénarios et de procédures métiers puis l'analyse aux travers de plusieurs expérimentations.

Ces chapitres sont suivis d'une conclusion et de propositions de perspectives de recherches.

1 Contexte du projet

Dans ce premier chapitre, afin de poser le contexte de cette thèse, nous allons présenter dans une première section 1.1 une introduction à l'industrie 4.0, une définition de ce que sont la réalité virtuelle et la réalité augmentée ainsi que les catégories d'usage de ceux-ci en industrie et en formation. De plus, ce travail rentrant dans le cadre d'une convention CIFRE, cela nous permet ainsi de bénéficier de l'expérience de l'entreprise OREKA Ingénierie. En effet, elle répond depuis plusieurs années à la demande du marché en termes d'applications de réalité virtuelle et augmentée pour les secteurs industriels (Agricole, Maritime, Militaire, Énergétique, etc.). Cela nous a permis d'effectuer une première analyse « proche du terrain » ainsi que de préciser nos axes de recherche. Cette analyse est présentée en détail dans la section 1.2.

1.1 Contexte scientifique - les usages de la réalité virtuelle et augmentée dans l'industrie

1.1.1 L'industrie 4.0

Au milieu du XX^e siècle, l'industrie a vécu une 3^{ème} révolution grâce à l'électronique et notamment l'arrivée de l'informatique. De nos jours, suite à l'émergence des nouvelles technologies numériques, l'industrie débute une 4^{ème} révolution, appelé « Industrie 4.0 » ou « Industrie du futur ». Le terme « **Industrie 4.0** » est apparu initialement en 2011 en Allemagne (Drath & Horch, 2014) dans le but d'encourager le développement numérique dans l'industrie. Il est alors repris partout dans le monde dans la communauté scientifique et prend différentes appellations : **Industry of the future**, **Smart Factory / Industry**, **Digital Factory**, etc. Dans (Danjou et al., 2017a), les auteurs ont démontré qu'il y a, en effet, un engouement croissant ces dernières années pour l'Industrie 4.0. Pour cela, ils ont réalisé un inventaire de la littérature en lien avec cette thématique de 2006 à 2015 (Figure 1).

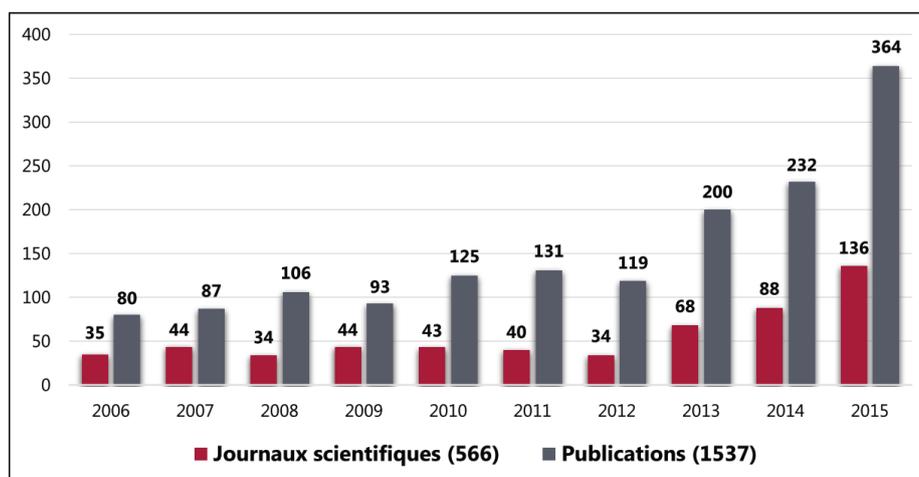


Figure 1 : Distribution par année des publications et journaux répertoriés en lien avec l'Industrie 4.0 (Danjou et al., 2017b, 2017a)

Un des grands axes de cette numérisation est l'interconnexion des données et l'optimisation de la chaîne de valeur. L'industrie 4.0 peut alors se résumer à un système global interconnecté dans lequel les machines, les systèmes de gestion (ERP : **E**nterprise **R**essource **P**lanning) et les produits sont connectés et échangent des données. Elle regroupe alors un ensemble de technologies et de concepts dont le but est la réorganisation de la chaîne de valeur (Hermann et al., 2016) et les flux d'informations.

Nous trouvons ainsi dans la littérature scientifique de nombreux travaux adressant les technologies mises en œuvre dans l'Industrie 4.0 (Dopico et al., 2016; Ernst & Frische, 2015; Keller et al., 2014). En reprenant (Danjou et al., 2017b), ces mêmes auteurs proposent de regrouper ces technologies en 10 groupes technologiques et développent ainsi la proposition de (Rüßmann, Michael Lorenz et al., 2015) qui proposait 9 piliers technologiques.

Nous retrouvons alors dans ces 10 groupes présentés sur la figure 2 :

- Le Big Data,
- L'intelligence artificielle,
- Le Cloud Computing,
- L'Internet des objets (IoT),
- Les systèmes cyber-physiques (Cyber-Physical Systems, CPS),
- La cybersécurité, la robotique / machines autonomes,
- La communication inter-machines (Machine-to-Machine, M2M),
- **La simulation (réalité virtuelle, jumeaux numériques),**
- **La réalité augmentée.**

Ce sont ces dernières technologies que sont la réalité virtuelle et la réalité augmentée que nous définirons plus en détail dans les sections suivantes. Nous nous intéresserons également au jumeau numérique (réplique numérique d'un objet, d'un processus ou d'un système) interfacé en réalité virtuelle et augmentée à travers l'un des cas d'usage présenté dans la section 4.3.1.



Figure 2 : Les nouvelles technologies de l'industrie 4.0 selon (Danjou et al., 2017b) (Voir Annexe : Figure 143)

1.1.2 Définition de la réalité virtuelle

Utilisée dans l'industrie 4.0 dans le cadre de simulations, l'expression « réalité virtuelle » fut déjà employée en 1938 par Antonin Artaud (Artaud, 1938) pour décrire le théâtre comme une « réalité virtuelle ». Mais c'est le terme (RV ou VR pour *Virtual Reality*) qui est repris en 1985 par Jaron Lasnier, développeur de jeux vidéo des studios ATARI pour désigner la réalité virtuelle telle qu'on la connaît aujourd'hui, que l'on retiendra comme origine. En effet, son entreprise « VPL Research » est la première à commercialiser des systèmes de réalité virtuelle clés en main en 1988, contribuant alors au développement de ce domaine de recherche.

Ainsi, la réalité virtuelle peut se définir de plusieurs manières d'autant plus que sa définition évoluera aux grés des technologies qui la composent. Nous retiendrons celle proposée par Philippe Fuchs, expert en réalité virtuelle, qui définit la réalité virtuelle comme ceci :

« La réalité virtuelle est un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de stimuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par l'intermédiaire de canaux sensori-moteurs » (Fuchs et al., 2006).

La définition de la réalité virtuelle peut alors se décomposer en deux principes :

- **Immersion dans un environnement virtuel d'un ou de plusieurs utilisateurs.**
- **Interaction en temps réel.**

Il existe aujourd'hui de nombreux dispositifs de réalité virtuelle répondant à cette définition. Ces interfaces ont tous pour but d'immerger et de permettre d'interagir avec un environnement virtuel, que ce soit sous forme de casques immersifs portés par l'utilisateur (Oculus, HTC Vive, Valve Index, Cardboard etc....) (Figure 3 – à gauche) ou sous forme de CAVE (**C**ave **A**utomatic **V**irtual **E**nvironment) où l'affichage est projeté sur les murs, sol et parfois plafond en temps réel (Figure 3 – à droite).



Figure 3 : Réalité virtuelle via casque immersif (à gauche) (*Réalité Virtuelle : Industriels, Architectes, Formateurs... Pourquoi Tous Les Pros Vont s'y Mettre*, 2016) et via un CAVE (à droite) (*Applications Du CAVE*, 2017).

1.1.3 Usages de la réalité virtuelle

Les applications de réalité virtuelle, avec la capacité d'immerger un ou plusieurs utilisateurs, leur donnent ainsi un sentiment de présence dans l'environnement virtuel (EV) et la capacité d'interagir avec un jumeau numérique (Havard, Jeanne, et al., 2019). Ils sont ainsi utilisés sur diverses activités dans des processus industriels comme aide cognitive. Dans (Damiani et al., 2018), les auteurs font état de la littérature et notamment des applications pratiques de la réalité virtuelle / augmentée dans l'industrie. De nombreuses applications concernent la formation et principalement dans le but de partager des connaissances et d'améliorer la sécurité des systèmes de production. Ils regroupent également d'autres types d'applications comme l'aide à la prise de décision, la sécurité, la logistique, la manutention, la conception et reconfiguration de produits/processus, la production et l'assemblage.

Dans une autre analyse (Rauch et al., 2020), les auteurs ont identifié 6 clusters de recherches, chacun répondant à un besoin parmi les 3 étapes d'un système de production industriel : **la planification** (conception technique et organisationnelle), **l'exécution** et **la maintenance**. Dans leurs rapports, la réalité virtuelle se place également comme aide cognitive pour la phase de planification d'un système de production avec, par exemple, l'aide à la prise de décision et durant la phase d'exécution avec des cas comme la formation à l'assemblage (Gorecky et al., 2017).

Afin d'illustrer ces usages, nous allons passer en revue les principaux exemples de cas d'usages tirés de la littérature dans le contexte de l'industrie 4.0.

Conception / Planification de produits et de processus :

Dans (Havard, Trigunayat, et al., 2019), un outil appelé « vPerf » est présenté (voir Figure 4). Il s'agit d'une application en réalité virtuelle collaborative dont le but est d'améliorer la prise de décision pour l'aménagement d'un atelier de production. Dans un contexte de planification d'usine, l'application permet de déplacer des machines et de faire varier en temps réel le parcours d'un ou de plusieurs produits et de calculer et visualiser des indicateurs de performances comme la distance parcourue ou encore le volume traité. Dans leur évaluation, les résultats révèlent que l'outil permet ainsi de trouver de meilleure solution d'aménagement d'usine et d'également prendre en compte les problématiques de sécurité (passages piétons, manque de visibilité, ...)

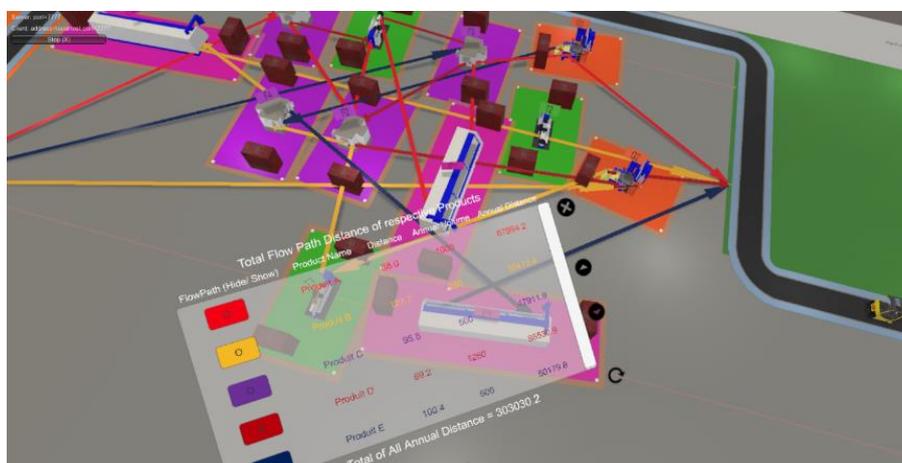


Figure 4 : Outil collaboratif à distance pour la planification d'agencement d'une usine (Havard, Trigunayat, et al., 2019)

Collaboration à distance :

La réalité virtuelle permet à plusieurs utilisateurs de collaborer à distance dans un environnement virtuel. Dans (Galambos et al., 2015), l'article présente un système appelé « VirCA » (Virtual Collaboration Arena). Il s'agit d'un framework dédié à la collaboration en réalité virtuelle/augmentée qui possède des fonctions permettant une interopérabilité entre différents domaines et technologies. Dans les cas d'usages présentés, il offre la possibilité d'une collaboration entre plusieurs utilisateurs, soit pour la formation de nouveaux employés aux environnements de ligne de production comme montré à gauche dans la figure 5, soit avec un utilisateur présent sur le système réel en RA et un autre utilisant le jumeau numérique via un CAVE (Figure 5 – à droite). Il est alors possible de participer à une consultation technique et à une formation pratique entre un expert en robotique sur le système réel et les utilisateurs finaux distants en virtuel.

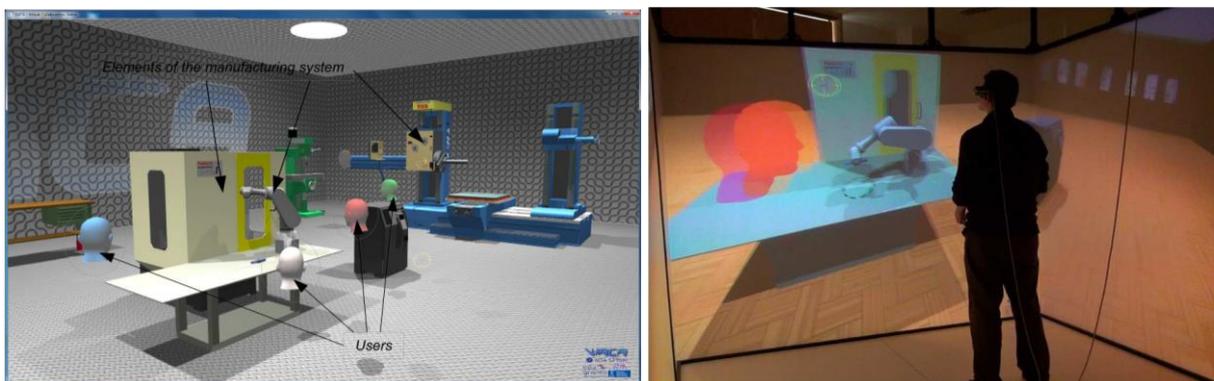


Figure 5 : (à gauche) Une session VirCA multi-utilisateurs avec quatre participants. Chaque utilisateur est virtuellement représenté par une tête colorée qui indique la direction du regard et son mouvement. (à droite) Autre cas d'usage, un des utilisateurs utilise un CAVE et la tête rouge symbolise l'opérateur assis à côté du vrai robot (Galambos et al., 2015)

Sécurité & évaluation ergonomique du poste de travail cobotique (collaboration avec un robot) :

Dans (Havard, Jeanne, et al., 2019), les auteurs proposent une architecture de cosimulation en temps réel entre un jumeau numérique et un environnement en réalité virtuelle. Le cas d'usage présenté porte sur la mise en place d'un bras cobotique UR10 sur un poste manuel afin d'aider l'opérateur lors d'assemblages industriels. L'association de la réalité virtuelle avec le système réel a notamment permis l'évaluation de la sécurité de l'opération et de l'ergonomie du poste comme montré dans la figure 6.

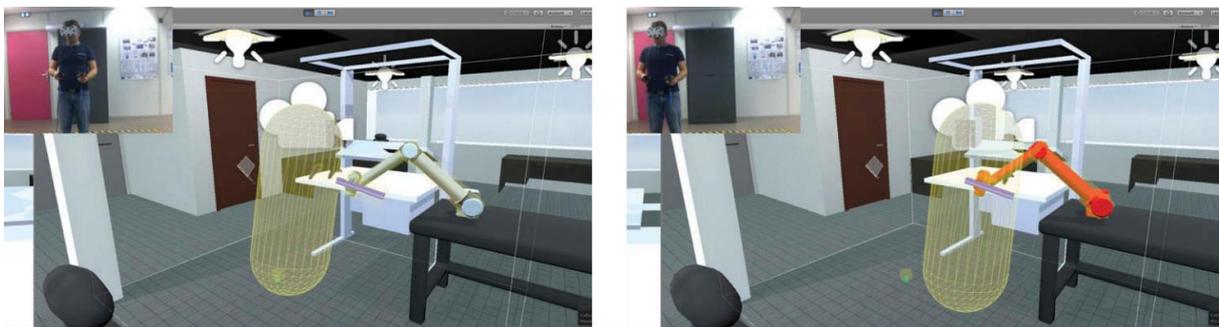


Figure 6 : Contrôle de sécurité en réalité virtuelle grâce à la détection de collision (Havard, Jeanne, et al., 2019).

Partage des connaissances et formation :

Le partage de connaissances et la formation étant l'un des apports clés de la réalité virtuelle, c'est dans ce domaine que l'on retrouve le plus de cas d'usages (Damiani et al., 2018). Déjà en 2001, (Stone, 2001) montrait que la réalité virtuelle répondait à ce besoin. L'auteur présente plusieurs cas d'usages dont une formation dans le domaine de la métallurgie sur l'usage d'une forge Vulcan ou bien dans le domaine chirurgical. En plus de former à concevoir, la réalité virtuelle peut former à maintenir un système industriel. Dans (Havard et al., 2017), une formation en réalité virtuelle à une maintenance industrielle sur une chaîne de production est présentée (Voir Figure 7).

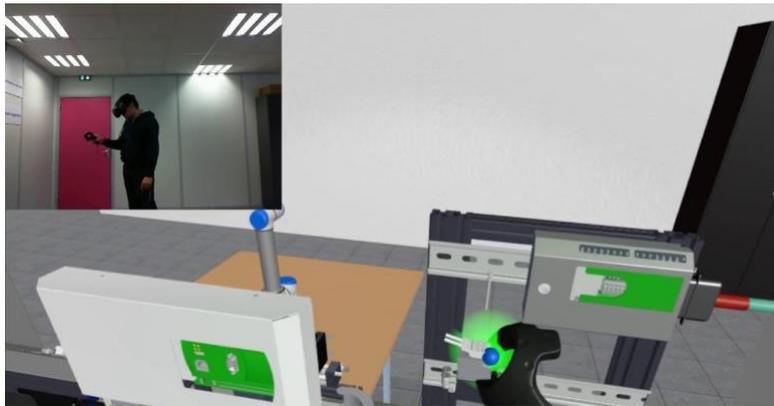


Figure 7 : Formation à une maintenance industrielle sur une chaîne de production en réalité virtuelle (Havard et al., 2017)

Sensibilisation des utilisateurs aux procédures de sécurité :

La réalité virtuelle permet également de sensibiliser les utilisateurs à la sécurité sans pour autant les mettre en danger. Dans (Matsas & Vosniakos, 2017), les auteurs présentent un EVAH (**E**nvironnement de réalité **V**irtuelle pour l'**A**pprentissage **H**umain) nommée « beWare of the robot » (Méfiez-vous du robot, en français). Il s'agit d'un serious game dont le but est d'effectuer des tâches de fabrication en collaboration avec un robot tout en sensibilisant aux problèmes de sécurité et aux problèmes de vigilances vis-à-vis du robot en déplacement (Figure 8).

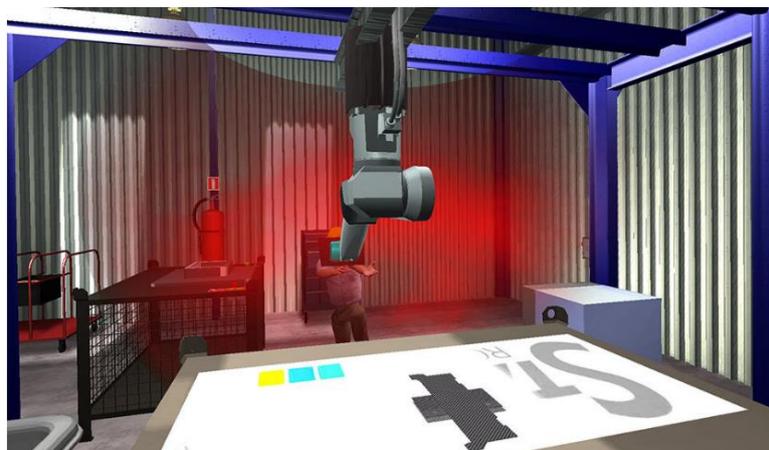


Figure 8 : Dans cette scène, l'alarme est activée, avertissant l'utilisateur qu'il est entré dans l'espace de travail du robot (Matsas & Vosniakos, 2017)

Nous avons ainsi défini et présenté les cas d'usages de la réalité virtuelle. Nous ferons de même pour la réalité augmentée dans la section suivante.

1.1.4 Définition de la réalité augmentée

À la différence de la réalité virtuelle, la réalité augmentée se rapproche du réel. Historiquement, c'est en 1992 que le terme « Réalité augmentée » (RA ou AR pour *Augmented Reality*) est introduit par (Caudell & Mizell, 1992) pour désigner des systèmes interactifs combinant la visualisation en temps réel d'objets virtuels et d'éléments physiques dans un environnement réel. L'auteur prototype un casque de type « Head-up display » permettant de connaître la position de la tête dans le monde réel et d'afficher des contenus virtuels à travers un système see-through (auquel on peut voir au travers) en temps réel. Puis, en 1994, Milgram propose une définition plus précise pour la réalité augmentée (Milgram & Kishimo, 1994) en émettant l'hypothèse d'un continuum entre le réel et le virtuel qu'il appelle la « réalité mixte » (voir Figure 9) composé de la réalité, la réalité augmentée, la virtualité augmentée et le virtuel.

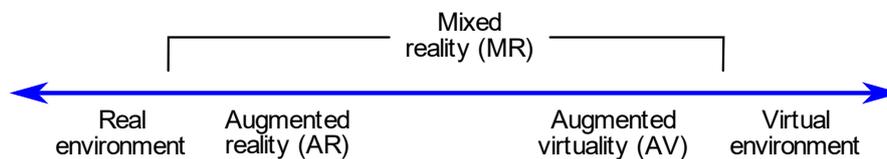


Figure 9 : Continuum entre le monde réel et le monde virtuel (Milgram & Kishimo, 1994)

En 1997, (Azuma, 1997) a voulu définir les différences opposant la réalité virtuelle et la réalité augmentée. Il propose ainsi une définition plus structurée de la réalité augmentée qui ne dépend pas de la technologie d'affichage employée. Il met en avant 3 caractéristiques que doivent respecter tous systèmes de réalité augmentée :

- **Le système doit pouvoir combiner le réel et le virtuel.**
- **Le système doit être interactif en temps réel.**
- **Le système doit être synchronisé en 3D.**

Aussi, pour qu'un système de réalité augmentée soit fonctionnel, il est nécessaire de connaître la position et l'orientation de la caméra filmant la scène, mais aussi de pouvoir traquer l'objet réel que l'on souhaite augmenter (image, marqueur, reconnaissance de l'objet 3D ...).

La réalité augmentée, permettant de mixer du contenu réel et virtuel trouve de nombreux domaines d'applications (divertissement, médecine, éducation, jeux vidéo et maintenance...). Nous allons voir plusieurs cas d'usages dans la section suivante.

1.1.5 Usages de la réalité augmentée en Industrie

Comme pour la réalité virtuelle, nous retrouvons la réalité augmentée dans (Damiani et al., 2018; Rauch et al., 2020) comme aide cognitive dans les différentes étapes d'un système de production industriel. Pour rappel, ces étapes sont la **planification**, l'**exécution** et la **maintenance**. La réalité augmentée, avec la particularité d'ajouter du contenu virtuel à l'espace physique, est alors davantage utilisée pour fournir une assistance visuelle à un opérateur et notamment dans les phases d'exécution et de maintenance.

Dans (Röltgen & Dumitrescu, 2020), les auteurs classifient les cas d'usages industriels de la réalité augmentée et les divisent en deux groupes : ceux pour l'instruction et ceux pour l'illustration (Figure 10). Le premier vise à fournir une information contextuelle à l'utilisateur lors de l'exécution de tâches tandis que l'illustration se concentre sur l'utilisation de la réalité augmentée pour la visualisation. L'objectif est d'élargir l'imagination de l'utilisateur, permettant des décisions plus éclairées. Pour ce qui est de l'instruction, nous retrouvons notamment l'aide à la maintenance, à l'assemblage / désassemblage, à la préparation de commande, au contrôle qualité ou encore l'assistance à distance.

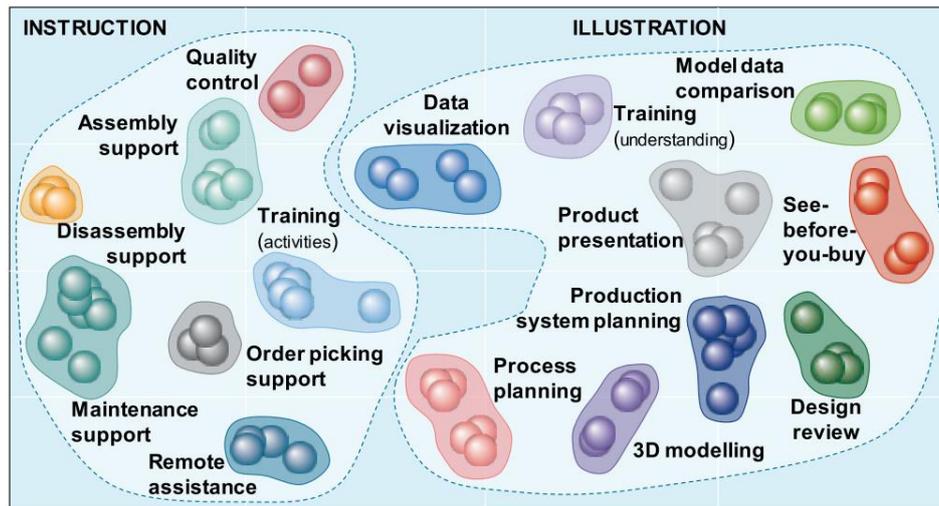


Figure 10 : Mise à l'échelle multidimensionnelle des exemples d'application en réalité augmentée présentés dans (Röltgen & Dumitrescu, 2020)

Pour la suite, voici quelques exemples des principaux cas d'usages de manière plus détaillée et tirés de la littérature concernant la réalité augmentée dans le contexte de l'industrie 4.0.

Assistance visuelle à la maintenance :

Dans le cas d'une gamme de maintenance, la réalité augmentée est particulièrement utile. Dans (Henderson & Feiner, 2011) les auteurs décrivent un prototype d'application en réalité augmentée avec un casque (video see-through) pour l'entretien et la maintenance d'un véhicule militaire (Figure 11). Pour leur expérience, les auteurs comparent la réalité augmentée avec un système d'écran déporté décrivant la tâche à effectuer ainsi qu'un système HUD (Head-up display) n'utilisant pas la réalité augmentée. Les résultats montrent que la réalité augmentée accélère jusqu'à deux fois plus la localisation des tâches à effectuer par rapport aux deux autres systèmes et réduit ainsi le nombre de mouvements de la tête lors de la localisation.

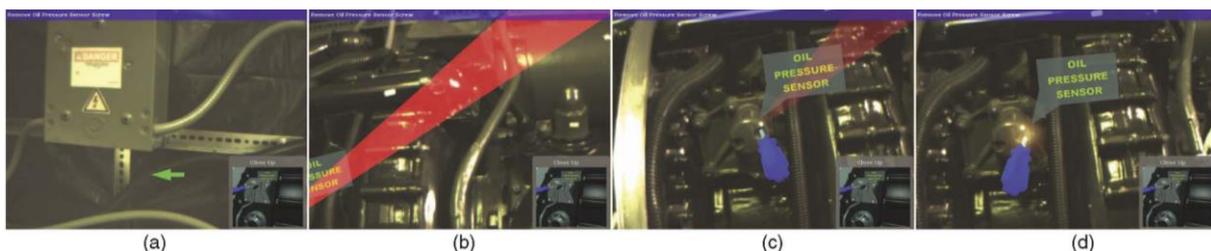


Figure 11 : Une séquence de localisation. (a) Une flèche fixe à l'écran indique la distance de rotation la plus courte jusqu'à la cible. (b) Lorsque l'utilisateur s'oriente sur la cible, une flèche 3D semi-transparente pointe vers la cible. (c) Lorsque l'utilisateur atteint la

cible, la flèche 3D commence un fondu progressif jusqu'à une transparence totale. (d) Lorsque la flèche s'est complètement estompée, un bref effet de surbrillance marque l'emplacement précis de la cible. (Henderson & Feiner, 2011)

Dans un cas d'usage similaire, (Havard et al., 2016) ont développé une application permettant la création de maintenance en réalité augmentée appelée « mProd ». En plus de l'assistance visuelle, l'application permet d'apporter une assistance audio et textuelle de la tâche à réaliser. Afin de tester leurs travaux, les auteurs ont effectué une expérience sur un cas de maintenance sur une chaîne de production (Figure 12) avec différents supports : papier, vidéo, et RA sur tablette ou smart glasses. Suite à cela, les résultats ont montré que pour une même durée, il y avait moins d'erreurs commises avec les supports en RA qu'avec le papier. Dans une seconde expérimentation, (Havard et al., 2021), les auteurs montrent que le temps de consultation du support en réalité augmentée sur tablette est réduit de 34% par rapport à son équivalent en document électronique. Par ailleurs, les auteurs montrent que les erreurs liées à la confusion d'objets similaires sont réduites puisque l'information en réalité augmentée est placée sur l'objet réel concerné.

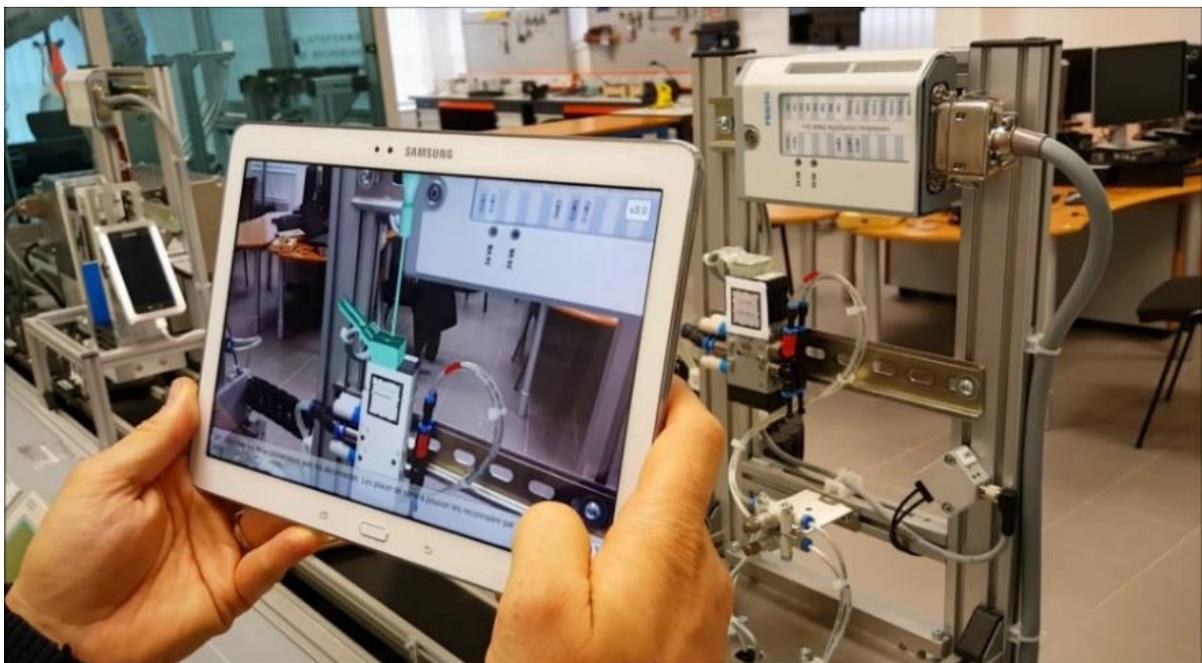


Figure 12 : Gamme de maintenance industrielle en réalité augmentée sur une chaîne de production (Havard et al., 2016, 2017; Richard et al., 2016)

Assistance visuelle au montage et à la logistique :

Dans sa thèse (Funk, 2016), l'auteur met en place un système de réalité augmentée projetée afin d'assister des opérateurs inexpérimentés / expérimentés ou ayant des troubles cognitifs lors des tâches d'assemblages. Un projecteur et une caméra de profondeur sont alors placés au-dessus de postes d'assemblages (Figure 13 – à gauche) et permettent de projeter des instructions sur les pièces (Figure 13 – à droite). De plus, une architecture logicielle a été développée et les prototypes d'application d'assistance prennent en charge à la fois l'assemblage manuel et la préparation des commandes en indiquant, par exemple, quelle pièce prendre.

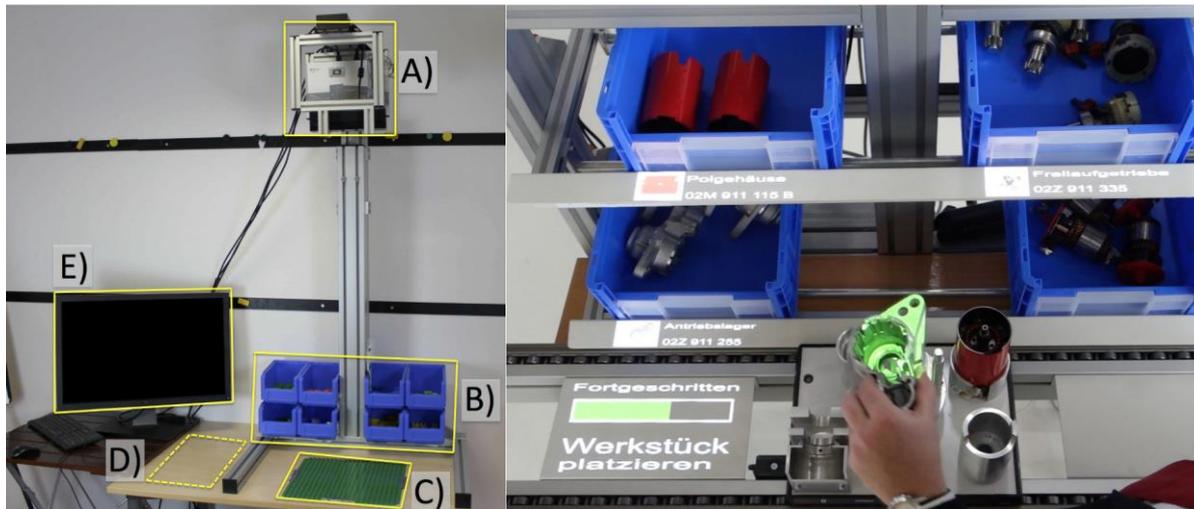


Figure 13 : (à gauche) Le type d'installation utilisé par Funk avec un projecteur et une caméra de profondeur en A. (à droite) La rétroaction projetée peut être enrichie d'informations supplémentaires. (Funk, 2016)

Plus récemment, dans (Vanneste et al., 2020), les auteurs ont comparé différents supports d'instructions pour des tâches d'assemblage : Oral, Papier et Réalité augmentée basée sur la projection. Ils ont montré à l'aide d'une expérimentation que cette dernière était adaptée aux tâches d'assemblage. Notamment le fait de laisser les mains libres à l'opérateur et sa capacité à superposer spatialement des instructions sur des pièces d'assemblage. Cette étude confirme également que l'usage de la réalité augmentée est meilleur que d'autres supports en termes de qualité d'instruction, réduisant ainsi le nombre d'erreurs de l'opérateur.

1.1.6 Synthèse

Nous avons vu dans les parties précédentes que la réalité virtuelle et la réalité augmentée, avec leurs avantages respectifs, pouvaient être appliquées à différents cas d'usages dans l'industrie. Les cas d'usages les plus répandus étant la formation pour la réalité virtuelle et l'aide à la maintenance et à l'assemblage pour la réalité augmentée. Pour ce qui est de la réalité virtuelle, des recherches ont déjà prouvé et validé son efficacité en termes d'apprentissage et d'impact positif sur les performances (Borsci et al., 2015; Feng et al., 2020; Merchant et al., 2014; C.-H. Su et al., 2019). Comme nous l'avons vu dans les cas d'usages présentés, la réalité augmentée a démontré son efficacité en termes de performances d'assistance et de guidage (Havard et al., 2016; Henderson & Feiner, 2011; Z. Wang et al., 2021), notamment dans un contexte d'assemblage ou de maintenance industrielle.

En effet, ces technologies permettent d'améliorer la qualité et le partage des connaissances entre l'expert et l'opérateur d'une part ou le formateur et l'apprenant d'autre part. Cependant, la création de contenu pour ces outils nécessite diverses informations et connaissances de l'entreprise, des compétences métiers (experts des procédés industriels, experts méthodes, formateurs) (Badets et al., 2020) mais également de fortes compétences en développement informatique. Dans le cas de la RA, ceci est par ailleurs identifié comme challenge par (de Souza Cardoso et al., 2020), qui expliquent que « des outils de développement de la RA plus intuitifs pour créer l'expérience augmentée sont nécessaires pour promouvoir la technologie. ». Concernant la RV, dans (Patle et al., 2019), les auteurs démontrent que « maintenir et/ou mettre à jour les VR-OTS (Virtual Reality

Operator Training Simulator) pour répondre aux demandes en constante évolution est l'un des défis majeurs » car cela nécessite des experts RV/RA. De plus, (Malik et al., 2020) expriment la nécessité de coupler RV/RA/JN (Jumeau Numérique) pour l'étude de cas d'usage industriel et notamment dans des contextes de collaboration humain robot car « l'utilisation de simulations continues basées sur le temps peut offrir un espace virtuel sûr pour les tests et la validation, facilitant ainsi la conception de systèmes HRC complexes (Human-Robot Collaboration, Collaboration Homme-Robot en français) ».

Ces avantages qu'offre la réalité virtuelle et la réalité augmentée dépendent avant tout de la notion de présence, de l'interactivité du monde virtuel et augmenté et de la manière dont elle a été conçue pour partager les connaissances. C'est pourquoi une attention particulière doit être portée à la création de l'environnement virtuel et augmenté, ainsi qu'aux actions réalisables au sein de celui-ci. Comme expliqué par Nebeling et al. (Nebeling & Speicher, 2018), il y a *"un manque d'outils pour prototyper et tester rapidement et facilement de nouvelles expériences utilisateur RA/RV"*. Il est également intéressant de noter dans le cadre du développement que, malgré les avancées de la recherche, les développeurs observent encore fréquemment un décalage entre les solutions proposées dans la littérature et la réalité au sein des entreprises de services du numérique (ESN).

Il est alors nécessaire de définir des modèles pour faciliter la création de contenu, que ce soit pour définir le niveau d'interaction de l'environnement, pour orchestrer des tâches dans des scénarios de formation ou d'activités, comme l'assemblage, la maintenance ou l'inspection... ou encore permettre le couplage avec d'autres outils, tels que les outils de CAO (Conception Assistée par Ordinateur) ou encore les outils PLM (Product Lifecycle Management). C'est pourquoi la prochaine section présentera une analyse des cas d'usages de la réalité virtuelle et augmentée rencontrés au sein d'OREKA ingénierie afin d'identifier les premiers besoins auxquels devront répondre ces modèles.

1.2 Contexte industriel - Étude et analyse au sein d'OREKA Ingénierie

Cette thèse se déroulant en collaboration avec l'entreprise OREKA Ingénierie, nous avons pu ajouter un réel contexte industriel lié aux activités de l'entreprise. En effet, cette entreprise qui existe depuis 2010, répond depuis plusieurs années à la demande en termes d'applications de réalité virtuelle et augmentée pour différents secteurs industriels. Cette section constitue une analyse des affaires auxquelles a répondu l'entreprise durant ces dernières années.

1.2.1 Définitions et objectifs de l'analyse

Le but de cette analyse est de faire ressortir les facteurs communs entre les différents projets de développement afin de capitaliser sur leur processus de création. Pour cela, nous disposons de plusieurs données d'entrée :

- Les fichiers Excel listant les affaires pour chaque année avec notamment :
 - Le client,
 - L'objet de l'affaire,
 - Le secteur du client,
 - L'état de l'affaire (Gagné, En attente, Perdu) ;
- Les dossiers des affaires (proposition commerciale, appel d'offres, facture, donnée d'entrée, livrable etc...),
- Les connaissances des équipes projets.

Avec ces données, nous avons réalisé une matrice des affaires sur un historique de 5 années, de **2015** à juin **2019**.

Pour cela, nous avons établi plusieurs classifications afin de pouvoir mieux regrouper et comparer les différents projets. Les affaires ont été ainsi analysées avec les éléments suivants : les types de dispositifs utilisés (casque de réalité virtuelle, de réalité augmentée, de réalité mixte, et « autre » regroupant PC, tablette, mobile etc...) ainsi qu'une classification par type d'usage, par contextes et pour finir par niveaux d'interactions.

Pour les types d'usages, nous les avons regroupés en 3 groupes :

- **Les outils de formation / Serious Game** : Ils concernent les applications ayant un but de formation, d'acquisition de connaissance. Leur but est d'immerger un apprenant dans un environnement artificiel visant à lui enseigner le fonctionnement d'un procédé complexe.
- **Les outils de démonstration / Support de communication** : Ils concernent les applications ayant purement un intérêt de visualisation (Ex : visualisation d'une animation d'un modèle 3D), de démonstration ou servant de support de communication.
- **Les outils d'aide et d'assistance** :
 - À la conception.
 - À la mise en œuvre.
 - À l'exploitation et à la maintenance.

Ensuite, nous avons défini des familles de contexte projet :

Maintenance	Marketing	Éducation et formation professionnelle
Assemblage	Logistique	Production
Ergonomie	Visite virtuelle	Simulation

Nous indiquons également le niveau d'interaction de l'application tels que :

- « **Aucune** » interaction correspond uniquement à de la visualisation (Ex : Vidéo 360),
- « **Faible** » pour tout ce qui est visite virtuelle, vidéo 360 ou encore manipulation d'un modèle 3D en réalité augmentée.
- « **Complet** » pour les applications disposants d'un niveau plus avancé d'interactions ou intégrant des règles métiers. Par exemple, pour une formation procédurale, il est nécessaire de pouvoir définir une succession d'étapes, ou encore un scénario (multi-objectifs de type Serious Game).

À cette classification, nous avons ajouté des critères sous forme de questions :

- Est-ce que l'application développée **est collaborative** ?
- Est-ce que l'application développée **est scénarisée** ?
- Y a-t-il besoin de **décrire un système** (interactions faisables) ?
- Y a-t-il des **objectifs** définis ?
- Les actions réalisées sur un objet ont-elles une conséquence sur d'autres (**causalité**) ?
- Y a-t-il une **définition de rôles** ?
- Y a-t-il des notions de **suivi de l'apprenant** ?
- Est-ce que l'application développée est utilisable sur différents types d'appareils (**multi-appareils** : VR, AR, MR, PC) ?

L'ensemble de ces critères a pour objectif d'identifier les demandes ainsi que les besoins actuels du marché étudié. Nous devrions pouvoir alors être en mesure de faire naître une problématique que nous affinerons lors de notre état de l'art, d'y apporter des ajustements ou encore de l'affirmer comme étant un besoin concret.

1.2.2 Résultats obtenus

Grâce à cette étude, une matrice des affaires a été réalisée permettant de mettre en évidence les types d'usage industriel et leurs tendances au sein de l'entreprise OREKA. Nous avons choisi de présenter les résultats avec deux facteurs différents : le nombre d'affaires et la répartition en pourcentage en termes de chiffre d'affaires. Le premier est purement quantitatif tandis que l'autre est un bon indicateur du volume de chaque projet. De plus, nous n'avons pris parmi les affaires, uniquement celles ayant été réalisées et gagnées.

Types de dispositifs utilisés :

Pour ce qui est des dispositifs utilisés, nous pouvons tout d'abord observer (voir Figure 14) la démocratisation de la RV, avec une augmentation importante de 2015 à 2016, s'expliquant notamment par la sortie du casque HTC Vive. Les demandes d'applications sont alors, avant 2016, essentiellement orientées sur des appareils plus conventionnels (PC, Tablette etc.) et

moins sur de la réalité virtuelle et peu de réalité augmentée. On observe également l'arrivée de la réalité mixte (RM ou MR, Mixed Reality en anglais) avec la sortie de l'HoloLens en 2017. La réalité mixte fusionne le virtuel et le réel et permet, contrairement à la RA, de placer des objets virtuels dans un environnement réel inconnu et sans la nécessité de tracker un objet existant.

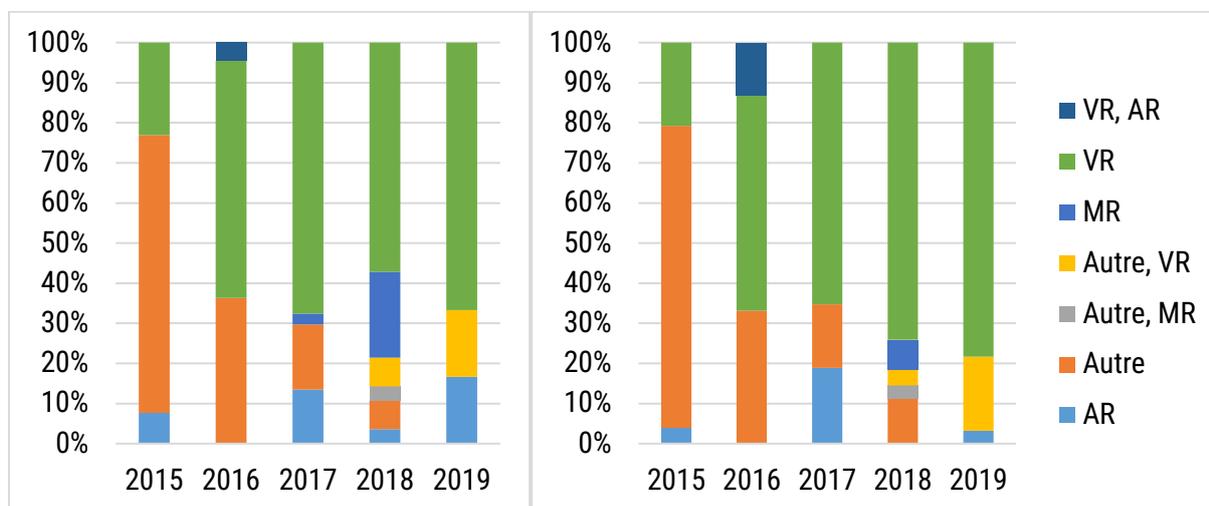


Figure 14 : Répartition des affaires en fonction des types de dispositifs utilisés (à gauche), répartition en nombre, (à droite), répartition en chiffre d'affaires.

Il est intéressant de noter que sur les 3 dernières années de l'analyse (2017-2019), les demandes de projets utilisant plusieurs technologies se démocratisent. Nous remarquons alors des projets multi-appareils, utilisant RV et RA ou encore PC et RV.

De manière globale, les affaires les plus importantes sont celles utilisant de la réalité virtuelle, les technologies classiques (PC Clavier-Souris, Tablette, Écran tactile etc...) et la réalité augmentée (voir Figure 15).

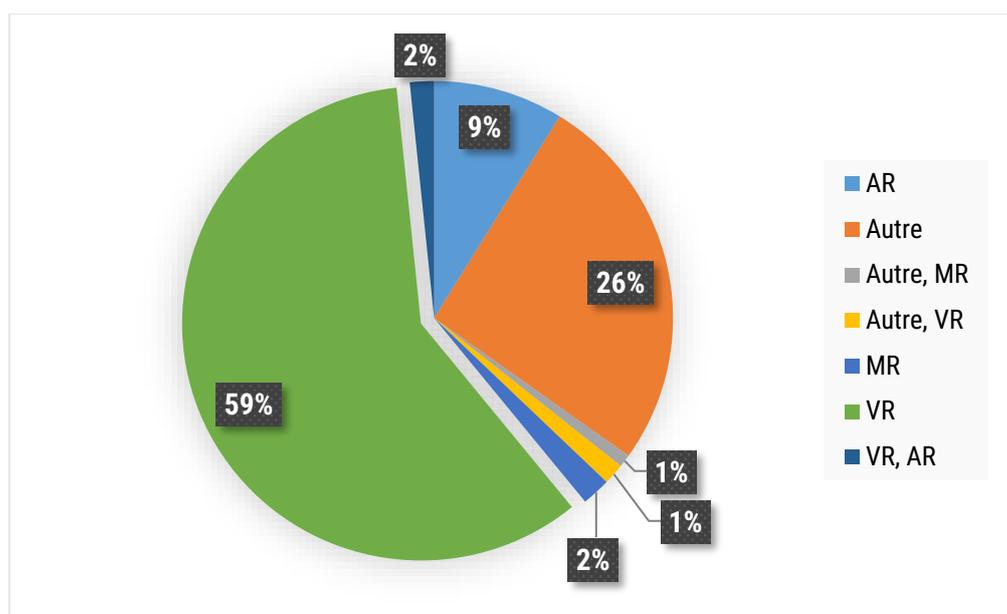


Figure 15 : Répartition en % des montants HT entre les affaires (2015-2019)

Types d'usage :

Les types d'usage, quant à eux, se portent principalement sur des outils de formations / Serious Game à **66%** suivis par les outils de démonstration et support de communication à **29%** (voir Figure 16). Nous pouvons noter également l'apparition d'outils d'aide et d'assistance en 2017 et 2018 correspondant à une hausse de la demande suite à la sortie de l'HoloLens.

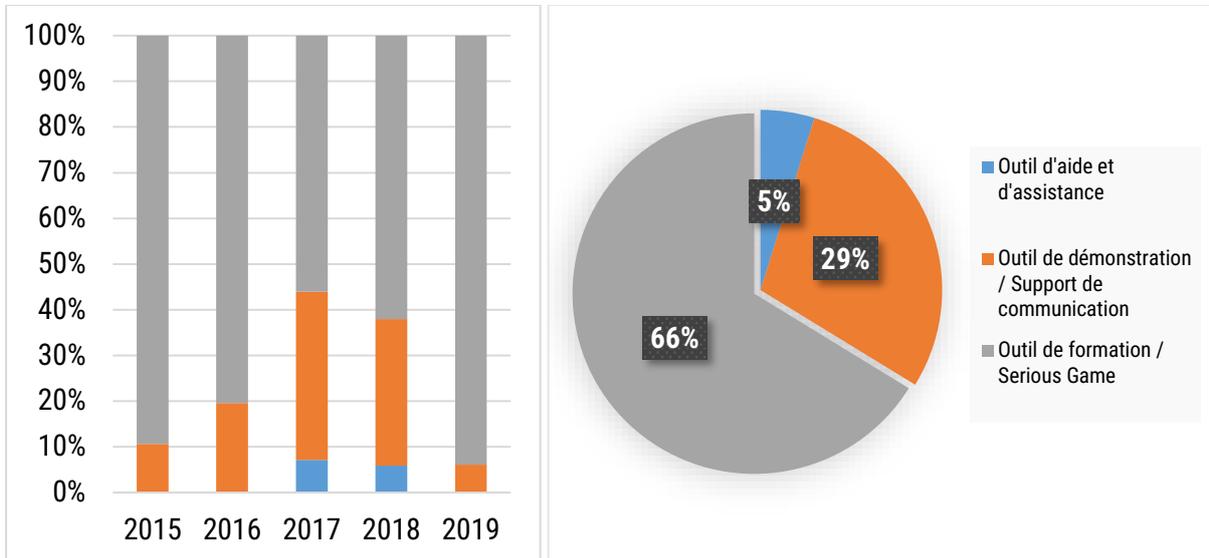


Figure 16 : Répartition des types d'usage par année (à gauche) et sur la période complète 2015-2019 (à droite).

Niveaux d'interaction :

Ensuite, en termes d'interaction, la demande la plus importante (71%) se porte pour un niveau d'interaction « complet » puis faible (28%) (voir Figure 17), ce qui concorde avec les résultats précédents. En effet, la majorité des applications développées au sein de l'entreprise sont des applications de réalité virtuelle et des outils de formations. Or, cette technologie offre justement un niveau d'interaction élevé entre l'utilisateur et l'environnement virtuel tandis que les outils de formations sont souvent sujets à une interactivité avancée pour l'apprenant, puisque ces formations doivent apprendre des concepts liés aux métiers appris.

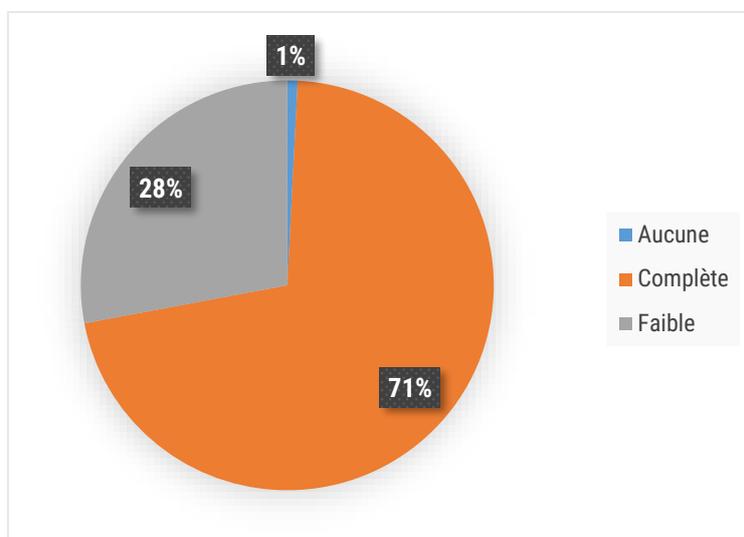


Figure 17 : Répartition niveaux d'interaction (2015-2019)

Contextes des projets :

Parmi les contextes d'application définis dans la partie précédente, quatre sont les plus notables (voir Figure 18). Ainsi **30%** des applications développées servent dans un contexte de production, **19%** servent dans le cadre de maintenances (Formation, Outils d'assistance), **17%** pour le marketing et **16%** dans le cadre de visite virtuelle. Ces résultats permettent de se rendre compte de la diversité des demandes, des buts et des contextes des applications développées sur la période d'analyse.

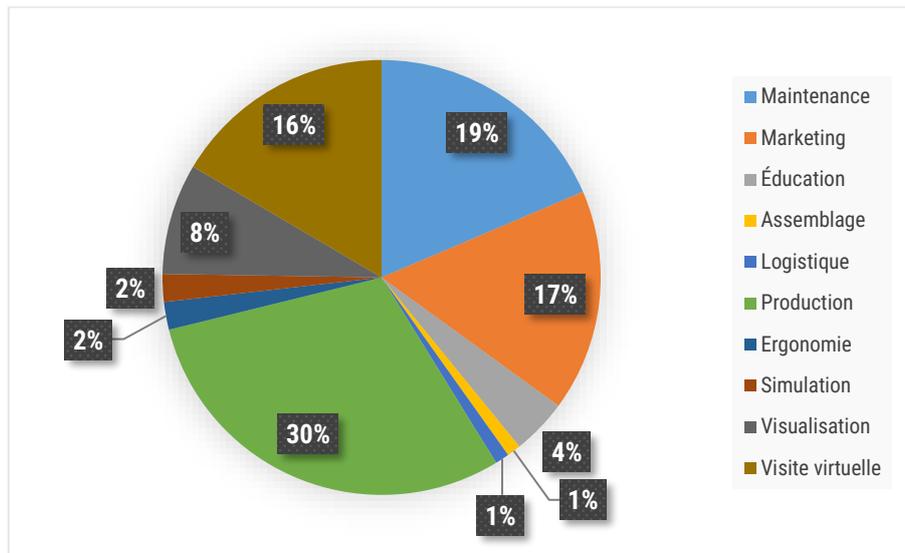


Figure 18 : Répartition contextes en nombre d'affaire (2015-2019)

Critères :

Cette section présente comment sont répartis les critères présentés précédemment parmi les applications développées sur la période 2015-2019 (Figure 19). Nous pouvons observer que les plus présents sont les applications qui requièrent la description d'un système (environnement virtuel et/ou augmenté), qui sont scénarisées, avec définitions d'objectifs à réaliser et exprimant des principes de causalité (une action sur une partie du système à une conséquence sur une autre partie). En effet, en détaillant l'analyse (voir Figure 20), on note que **80%** des applications développées requièrent la description d'un système, **71%** des applications possèdent un scénario et **69%** ont des objectifs définis. Ces applications étant les plus communément développées, c'est donc sur ces principaux axes que nous élargirons par la suite l'état de l'art et que nous chercherons à optimiser afin d'accélérer le temps de création et d'édition. Cela confirme la nécessité de formaliser un modèle de données permettant de scénariser un environnement virtuel ou augmenté.

A contrario, on observe des critères moins notables tels que la définition de rôles, le suivi de l'apprenant, si l'application est compatible sur différentes technologies (Ex : RV et RA) ou encore si l'application est collaborative.

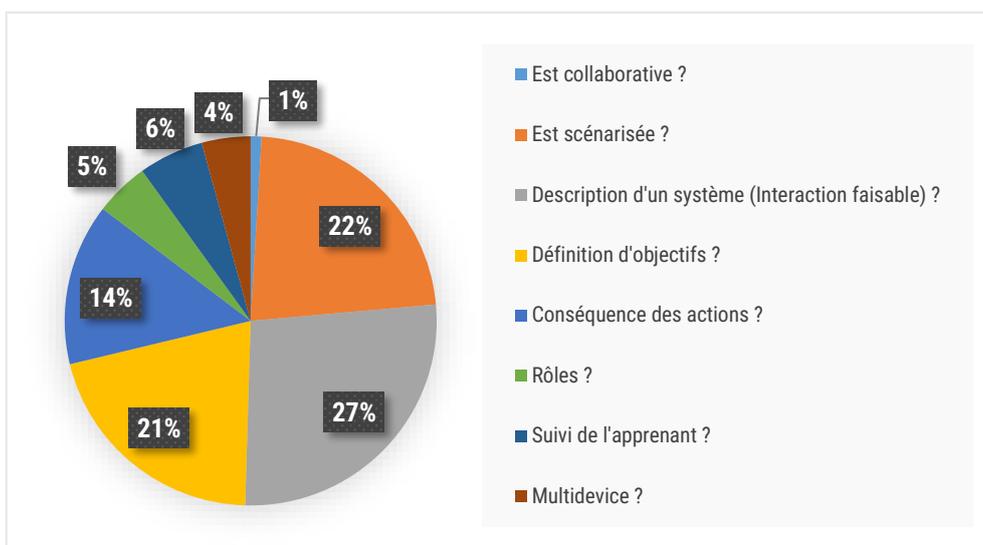


Figure 19 : Répartition des critères (2015-2019)

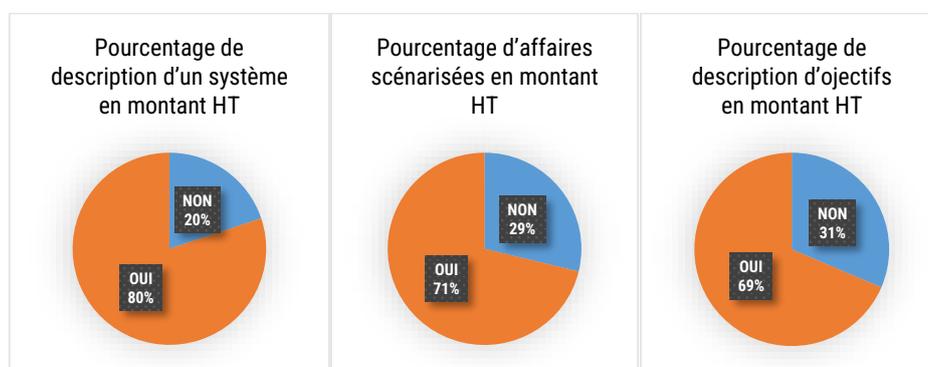


Figure 20 : Pourcentage (suivant le montant HT) des 3 principaux critères pour toutes les affaires

1.2.3 Synthèse

En conclusion de tous ces résultats, une part importante des réalisations de l'entreprise sont des applications de réalité virtuelle, d'outils de formations et avec un niveau d'interaction « complet ». En ce qui concerne les usages, les résultats obtenus corroborent l'utilisation des technologies de réalité virtuelle et de réalité augmentée présentées dans la section 1.1.2. Ensuite, concernant les critères représentant les applications de RV/RA, nous pouvons supposer, même si ces besoins sont moins présents actuellement, que la formalisation d'un framework et d'un modèle de données associé pourrait faciliter la mise en place d'applications multi-appareils ou collaborative et ainsi permettre le développement de ce type d'application.

Ces travaux permettront notamment d'orienter l'entreprise OREKA Ingénierie dans sa stratégie de coupler des compétences RA/RV et des compétences métiers tels que la performance industrielle (Ex : simulation de flux) et de développer des outils répondant aux besoins identifiés. Ainsi, à partir de cette analyse, nous allons notamment pouvoir extraire les critères minimaux qu'il faudra inclure dans notre proposition de modèle. Nous les élargirons par la suite lors de l'état de l'art dans le chapitre suivant.

1.3 Synthèse des observations

De ces observations, qu'elles soient issues de la littérature scientifique ou issues de l'analyse des usages industriels réalisés par l'entreprise, nous pouvons confirmer que l'arrivée des nouvelles technologies RV/RA dans l'industrie offre de nouvelles opportunités et fait apparaître de nouveaux besoins. Que les opérateurs utilisent la RV pour s'entraîner à travailler en milieu hostile en toute sécurité, ou qu'ils utilisent la RA pour bénéficier d'une assistance et faciliter la maintenance des machines, de nombreux cas d'usage sont déjà observables tant dans l'industrie que dans la littérature. Ces différents cas d'utilisation ont été regroupés comme suit :

- **Outils de conception, de planification ou d'assistance :**
 - À la conception (Damiani et al., 2018; Galambos et al., 2015; Havard, Jeanne, et al., 2019; Rauch et al., 2020).
 - À la mise en œuvre (Badets et al., 2020; Havard, Jeanne, et al., 2019).
 - Pour l'exploitation et la maintenance (Damiani et al., 2018; Funk, 2016; Geng et al., 2020; Havard et al., 2016, 2017; Henderson & Feiner, 2011; Rauch et al., 2020; Vanneste et al., 2020).
- **Outils de formation / Serious Game :** Leur objectif est d'immerger les apprenants dans un environnement artificiel conçu pour leur apprendre le fonctionnement d'un processus complexe (Berg & Vance, 2017; Matsas & Vosniakos, 2017; Pérez et al., 2019; Stone, 2001).
- **Outils de démonstration / Support de communication :** Il s'agit d'applications ayant un pur intérêt de visualisation (par exemple, l'animation d'un modèle 3D), une utilisation de démonstration, ou servant de support de communication (Matsas & Vosniakos, 2017).

La réalité virtuelle comme la réalité augmentée sont critiques dans ce contexte, et des besoins spécifiques peuvent apparaître parmi ces différents types. Citons par exemple, la possibilité de travailler avec plusieurs agents, soit dans une application collaborative avec plusieurs opérateurs pour l'aménagement d'un atelier (Galambos et al., 2015; Havard, Trigunayat, et al., 2019), ou dans le cadre de la conception ergonomique d'un poste de travail, ou encore dans une formation au montage avec un agent autonome tel qu'un bras robotisé couplé à un jumeau numérique (Havard, Jeanne, et al., 2019). Cela peut également inclure la possibilité de donner plus de liberté et d'implication aux utilisateurs dans le scénario en leur permettant de faire des erreurs (Stone, 2001) ou attacher des conséquences à leurs actions (c'est-à-dire le principe de causalité) (Lanquepin et al., 2013). Pour répondre aux besoins de ces environnements avancés, certains modèles et frameworks peuvent également inclure d'autres caractéristiques, telles que des modèles de rôle (Claude et al., 2014), des modèles de suivi de l'apprenant (Lanquepin et al., 2013), ou la compatibilité multi-appareils.

Ainsi, à partir des premiers challenges discutés dans la section 1.1.6 et des besoins identifiés par OREKA Ingénierie pour un framework permettant d'accélérer la création et la modification d'environnements RA/RV dans un contexte industriel, nous avons pu faire naître et positionner la problématique principale de cette thèse. Nous détaillerons cette problématique dans le chapitre suivant d'état de l'art. En effet, l'intérêt de l'industrie pour ces technologies a suscité de nombreux

travaux de recherche autour des méthodes et procédés pour la mise en place d'environnement et de scénario. Ainsi, le chapitre 2 fait état des modèles d'environnement et de scénario existants ainsi que les frameworks existants en RA et RV.

2 État de l'art

2.1 Introduction

Concernant l'existant en termes de modèles pour des applications RA/RV, deux principales approches de modélisation existent : l'une est basée sur le langage UML, comme MASCARET (Buche et al., 2011), #FIVE (Bouville et al., 2017), et (Havard et al., 2015; Martínez et al., 2014; Querrec et al., 2013); ou bien basée sur des ontologies comme (Hervás et al., 2010), HUMANS (Lanquepin et al., 2013) ou INOOVAS (Havard et al., 2017). Nous verrons certains de ces travaux plus en détail par la suite.

D'un côté, la modélisation UML permet un formalisme structuré représentant l'environnement mais nécessite que l'expert s'y conforme. À l'inverse, les ontologies peuvent être plus facilement adaptées aux vocabulaires des experts (Sanfilippo & Terkaj, 2019) et permettront, de fait, aux experts métiers d'exprimer plus facilement les connaissances qu'ils souhaitent partager. Cependant, en voulant produire des modèles avec un vocabulaire et une logique spécifique, cela rend plus difficile la formalisation de scénarios complexes. De plus, dans le domaine de l'industrie, ainsi que dans celui du développement logiciel, les experts métiers sont plus souvent confrontés au type de représentation UML, avec une représentation comme les diagrammes de classes, ou de dérivés tels que SysML (Lutfi & Valerdi, 2020), ou encore avec des représentations graphiques pour la programmation d'automates (Grafcet, Grafchart, etc...); ces diverses représentations facilitent le développement et l'intégration des règles métiers dans les applications de RV/RA. C'est la raison pour laquelle, il est intéressant d'étudier ces types de représentation existants et qui sont couramment utilisés dans les entreprises travaillant dans un contexte industriel, pour mettre en œuvre des outils XR, tout en permettant à partir de concepts simples de pouvoir générer des environnements augmentés et virtuels.

Dans ce chapitre, nous verrons une analyse des travaux de recherche portant sur la scénarisation (planification détaillée d'un séquençement d'actions) et l'orchestration d'actions (disposer dans un certain ordre) dans les environnements virtuels ou augmentés. Nous entamerons avec la section 2.2.1 par les modèles orientés réalité virtuelle avec en 2.2.1.1, les modèles d'environnements permettant de décrire un environnement ainsi que les interactions possibles dans celui-ci (Bouville et al., 2017; Chevaillier et al., 2009; Lanquepin et al., 2013). Ensuite, nous nous intéresserons dans la section 2.2.1.2 aux modèles de scénarios permettant notamment d'orchestrer des actions et de décrire un scénario (Buche et al., 2011; Chevaillier et al., 2012; Claude, 2016). Nos travaux s'ouvrant à la RA, nous nous intéresserons ensuite, dans la section 2.2.2, aux modèles orientés réalité augmentée. Puis pour finir, nous présenterons INOOVAS dans la section 2.2.3, un framework compatible RA/RV issue de précédents travaux de recherche (Havard et al., 2017). Enfin, nous ferons état des représentations graphiques liées à la scénarisation dans la section 2.3.

2.2 Modèles et Frameworks pour la réalité virtuelle et augmentée

L'étude proposée dans cette section sur les modèles et frameworks existants est analysée comme suit (Figure 21) : modèle d'environnement, modèle de scénario, et framework, ce dernier étant un regroupement de ces deux premiers. Par ailleurs, d'autres types de modèle complémentaires seront intégrés dans certains cas, tels qu'un modèle de suivi de l'apprenant ou de gestion des rôles. Les modèles d'environnements (Figure 21 : Modèle d'environnement), sont divisés en trois catégories comme proposée par #SEVEN (Claude et al., 2014) : orientés « comportement », orientés « objets synoptiques » et orientés « objets-relations ». La définition de ces termes ainsi que les modèles présentés sont passés en revue en 2.2.1.1. Les modèles de scénario peuvent permettre trois niveaux de liberté en termes de scénarisation : procédural (pas à pas), scénario ouvert avec objectifs (des lignes directrices), et scénario complètement ouvert (simulation). En effet, les scénarios peuvent aller de scénarios métier tels que des gammes opératoires ou de maintenance à des scénarios d'apprentissage. Des modèles de scénarios appartenant à différents frameworks sont passés en revue dans la section 2.2.1.2. De plus, plusieurs caractéristiques identifiées à partir des besoins industriels discutés dans le chapitre précédent seront considérées pour chaque modèle, notamment la technologie ciblée (RA ou RV), s'ils mentionnent la collaboration ou la compatibilité avec des agents autonomes, ou encore le principe de causalité.

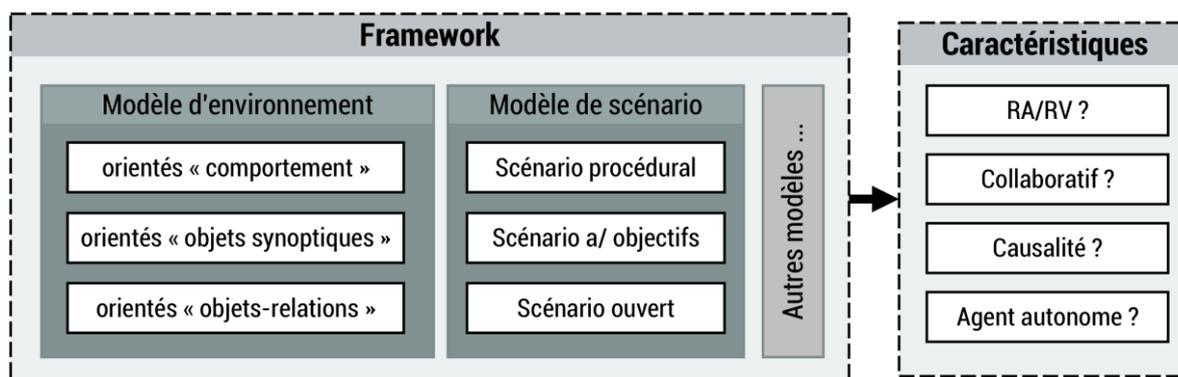


Figure 21 : Organisation utilisée pour la comparaison des modèles

2.2.1 Modèles orientés réalité virtuelle

2.2.1.1 Modèles d'environnements

L'un des premiers objectifs d'un modèle d'environnement est de formaliser les concepts permettant de décrire et d'utiliser les environnements virtuels et augmentés. Cette formalisation à travers un framework permet aux développeurs et utilisateurs finaux de réduire le temps de création de contenu. Par ailleurs, la formalisation permet de donner un cadre conceptuel permettant de faire évoluer le framework plus rapidement. Une fois le modèle créé, et à condition que celui-ci utilise des concepts réutilisables, il devient plus facile pour les développeurs d'implémenter les scénarios produits par les experts métiers. Cela peut également permettre aux experts métiers n'ayant pas de connaissance en développement, via une interface simplifiée représentant ces concepts, de faire évoluer ces scénarios. Par exemple, un expert doit facilement

pouvoir mettre en place des interactions d'une manière qui lui permette de produire des scénarios pour partager ses connaissances plus rapidement. Des frameworks comme UMI3D (Casarin et al., 2018) ou D3PART (Lacoche et al., 2017) sont de bons exemples de cette tendance permettant de faciliter le processus de création de contenu du point de vue des développeurs et de mise en place de multimodalités d'interaction (casques RV, PC clavier/souris, ...) que nous présenterons dans le détail dans notre proposition du chapitre 3. Comme notre cible applicative porte sur des cas d'usages industriels impliquant ces deux modalités RV/RA, nous étudierons dans la section suivante les modèles existants pour la mise en place d'environnements en réalité virtuelle et nous nous focaliserons sur la réalité augmentée dans une prochaine section (2.2.2).

S'agissant de la réalité virtuelle, que ce soit pour un **Environnement de réalité Virtuelle** pour les **Apprentissages Humains** (EVAH, VRLE en anglais), un système de formation en réalité virtuelle (VRTS en anglais) ou encore de simulation métiers avec un jumeau numérique, les modèles définissent comment les actions, au sein de l'environnement, fonctionnent et comment les apprenants ou utilisateurs, les objets virtuels et les interactions sont gérés. Ainsi, il existe plusieurs façons de représenter des environnements virtuels interactifs et plusieurs types de modèles avec leur propre logique. Dans la littérature, ils sont regroupés selon trois manières principales de représenter l'EV (environnement virtuel) et les interactions : les modèles orientés comportement (Lamarche & Donikian, 2002), les modèles orientés « Objets synoptiques » (Badawi & Donikian, 2004) et les modèles orientés « Objets-Relation » (Barot, 2014; Lanquepin et al., 2013; Mollet et al., 2007), détaillés ci-dessous.

2.2.1.1.1 Modèles orientés comportement

Tout d'abord, l'interaction au sein d'un environnement virtuel peut être représentée sans entité spécifique, mais plutôt comme un comportement ou une fonction. Parmi ces modèles orientés comportement, nous retrouvons le framework HCSM (Cremer et al., 1995) ou encore le langage HPTS++ (Lamarche & Donikian, 2002) qui est une extension de HPTS. Ces derniers utilisent des formalismes de machines à états hiérarchiques. Ce formalisme utilise des états, représentés par des bulles, et des transitions entre ces états, représentées par des flèches. Il existe deux types de notation, les machines à états de Moore (Moore, 1956) ainsi que les machines à états de Mealy (Mealy, 1955). Leurs notations sont similaires, la différence principale est que dans les machines de Moore les sorties possibles ne dépendent que de l'état présent tandis que dans les machines de Mealy, elles dépendent également des entrées. C'est cette version de Mealy qu'utilisent les auteurs. Prenons avec la figure 22, l'exemple d'un comportement d'un déplacement d'objet. Le formalisme se compose d'états et de transitions. Nous avons pour chaque transition, une condition d'entrée (Ex : « true ») et une valeur de sortie (Ex : 1). Pour arriver à l'état « TakeObject », la valeur d'entrée de « GetHand » doit être *true* et la sortie sera donc de 1. Ensuite, lorsque l'état d'entrée de « TakeObject » devient « endOfMovement », et donc que l'objet a été pris, il y aura un passage à l'état « MoveAndWatch » et « MoveObject ». Pour finir, lorsque ces deux mouvements sont terminés alors il sera possible de passer à l'état « PutObject ».

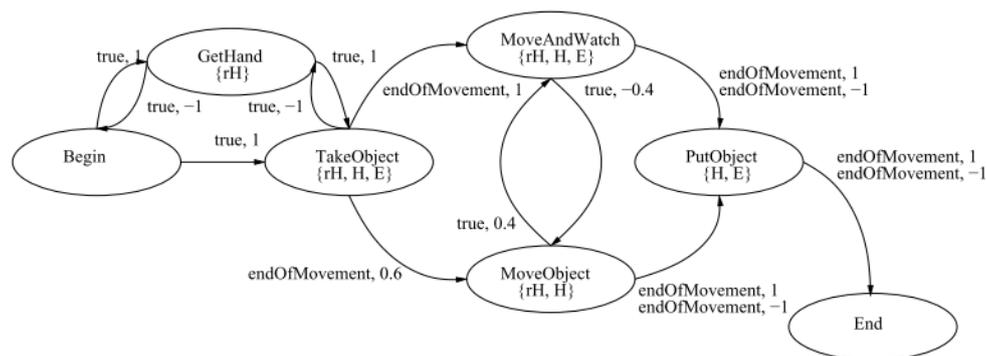


Figure 22 : Exemple d'une formalisation machine à états pour le comportement d'un déplacement d'objet (HPTS++ (Lamarche & Donikian, 2002))

L'avantage de ce type de modèle est leur capacité à observer l'état de l'environnement et à réagir à toute action. Cependant, du fait que ces frameworks utilisent des entités spécialisées pour les objets et pour les interactions des agents (autonomes ou non), cela implique d'adapter le comportement de chaque nouvel objet ou agent virtuel. Même si, de ce fait, ils peuvent en général être facilement adaptés à n'importe quel contexte d'application, cette approche est globale. Ce type de modèle est très spécifique et dépendant du contexte d'application et devient donc peu modulable et réutilisable. Par conséquent, si la mise en œuvre d'un autre modèle est nécessaire (par exemple, modèle de collaboration ou de gestion des rôles), cela nécessite un développement supplémentaire pour l'adapter et le faire fonctionner avec l'environnement existant. Aussi, les concepts ne peuvent être utilisés de manière unitaire et sont donc difficilement modulables.

2.2.1.1.2 Modèles orientés « Objets synoptiques »

À l'inverse des modèles présentés précédemment, les interactions possibles peuvent être contenues dans les objets eux-mêmes. Ces « objets synoptiques » utilisent l'objet virtuel comme son propre gestionnaire d'actions. Lorsqu'un acteur souhaite interagir avec un objet, celui-ci propose une liste des interactions qu'il permet. Dans Starfish (Badawi & Donikian, 2004), l'interactivité des objets est basée sur deux composants simples : les actions et les surfaces interactives. Les actions sont soit des actions basiques qui sont au minimum des « actions atomiques primitives » prédéfinies (par exemple donner, déplacer, attraper ou saisir), ou une composition de ces actions prédéfinies. En reprenant l'exemple des auteurs, l'action « ouvrir une porte » peut être ainsi facilement décomposée en actions basiques : Se **déplacer** à la porte ; **Bouger** le bras vers la poignée ; **Saisir** la poignée ; **Déplacer** la main pour tourner la poignée ; **Déplacer** la porte ; **Relâcher** la poignée. Les surfaces interactives sont généralement un volume par zone qui définit les parties de l'objet qui correspondent à l'exécution d'une action. Dans le framework Marigold, (Willans & Harrison, 2001) les auteurs utilisent une représentation hybride pour exprimer des comportements d'objet principalement constitués par des entrées / sorties continues et discrètes ainsi que des réseaux de Petri Sauf (détaillé dans la section 2.3.4) (Petri, 1962, 1966). L'action « ouvrir une porte » vu avant est illustrée dans la figure 23 en suivant cette représentation.

Cette approche orientée objet est très efficace lorsqu'il s'agit de créer des actions d'objet unique, comme ouvrir / fermer une porte ou encore activer des boutons, mais cela arrive rapidement à

ses limites pour des actions d'objet multiples comme lors d'une opération d'assemblage. Dans ce cas, chaque action doit être définie sur chaque objet en fonction de chaque combinaison possible. De plus, il est d'autant plus difficile lorsqu'il s'agit d'applications collaboratives et de la gestion de plusieurs instances d'objets.

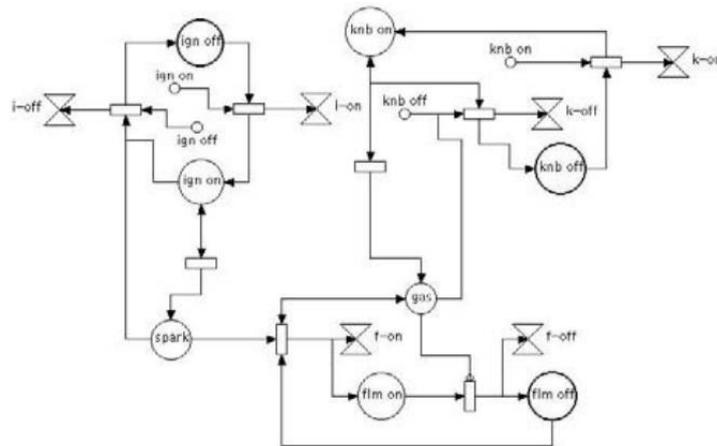


Figure 23 : Exemple de comportement d'un objet synoptique présenté dans (Willans & Harrison, 2001)

2.2.1.1.3 Modèles orientés « Objets-Relations »

Ensuite, l'interaction peut également être représentée sur la base d'une logique combinatoire appelée « Objets-Relations ». En effet, en ajoutant une autre entité appelée « Relation » en plus des objets, ces modèles parviennent à représenter plusieurs actions d'objet. Une relation consiste en des comportements impliquant différents objets avec des conditions préalables et des conditions évoluant au cours de l'exécution. Si une condition est atteinte, les comportements modifient l'état des objets (Lugrin & Cavazza, 2006). Une condition peut être induite à partir de la définition des « Types » d'objets, eux-mêmes définis dans la relation, de leurs états (par exemple, une modification de propriété) ou de leur placement dans l'environnement virtuel. Cela permet une manière plus flexible et générique de décrire les interactions, ce qui facilite encore plus la gestion de plusieurs instances. En effet, certains modèles « Objets-Relations » peuvent même gérer plusieurs acteurs (réels et virtuels) interagissant dans un environnement commun, tel que par exemple dans l'approche STORM (Mollet et al., 2007). Dans la littérature, des frameworks comme **MASCARET** (Chevaillier et al., 2012) ou **HUMANS** (Barot, 2014; Lanquepin et al., 2013) intègrent respectivement leur modèle orienté « Objets-Relation », **VEHA** pour MASCARET et **Domain-DL** pour HUMANS.

VEHA (Virtual Environment supporting Human Activities) (Chevaillier et al., 2009) est un modèle basé sur UML qui permet de représenter et décrire les entités disponibles et l'interaction à l'intérieur de l'EVAH. Il définit : les types d'objets, leurs structures, leurs comportements et les relations logiques entre eux, ainsi que leurs topologies. Il permet également de représenter des types communs de relations telles que la relation physique ou la relation de composition d'objet comme un livre à l'intérieur d'un bureau. La relation de composition peut également être utilisée pour représenter un objet sous la forme d'une composition de différents objets, ce qui est un avantage pour les opérations d'assemblage. Les comportements sont représentés, soit à l'aide de la machine-états de VEHA (Figure 24), soit par un « comportement opaque » (Figure 24 -

Behavior), c'est-à-dire un morceau de code spécifique pour lequel aucune représentation sémantique n'est donnée. VEHA fournit l'abstraction nécessaire à la modélisation d'entités spécifiques aux domaines d'activité de l'application. VEHA permet à plusieurs acteurs d'utiliser des comportements différents d'un même objet ou de faire de la « co-manipulation ». Cependant, comme les auteurs le mentionnent, la difficulté réside dans la synchronisation des différents comportements d'un objet complexe à l'aide de machines-états.

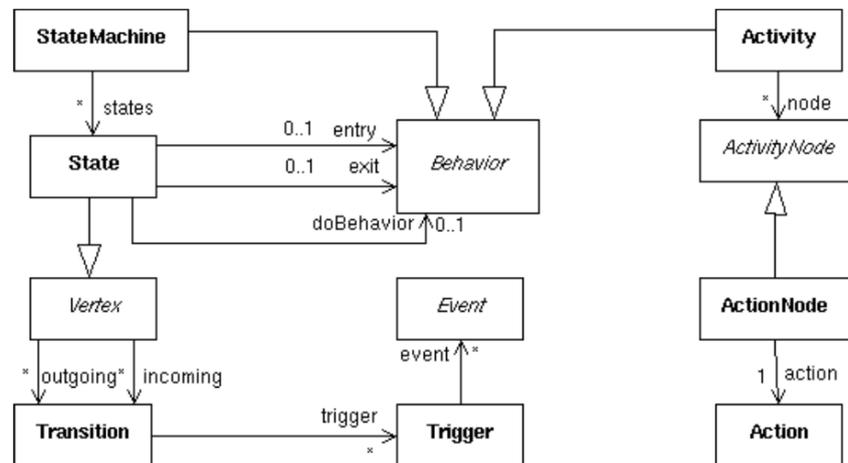


Figure 24 : Diagramme de classes du méta-modèle VEHA : machine à états comportementale (Chevaillier et al., 2009)

Domain-DL (Barot, 2014; Carpentier et al., 2013) est basé sur un ensemble d'entités associées : objets, actions, comportements et événements (Figure 25). Ils décrivent les objets comme un ensemble de propriétés. Le formalisme proposé décrit le monde virtuel avec une ontologie (basée sur le langage OWL-DL (Bechhofer et al., 2004)). Les objets peuvent être des entités concrètes (e.g. modèle 3D, agents) ou abstraites (e.g. objectif, tâche) qui peuvent être décrites ou manipulées dans l'environnement virtuel. Ils peuvent également être des agents capables d'interagir de manière autonome avec leur environnement. Les actions permettent à ces agents (réels ou simulés) de réaliser des relations entre objets uniquement si les conditions préalables en fonction de l'état de l'environnement virtuel sont validées. Lorsqu'une action est réalisable, les comportements correspondants sont créés. Le comportement peut également être créé par un événement. Une fois déclenché, un comportement modifie l'état d'un ou plusieurs objets.

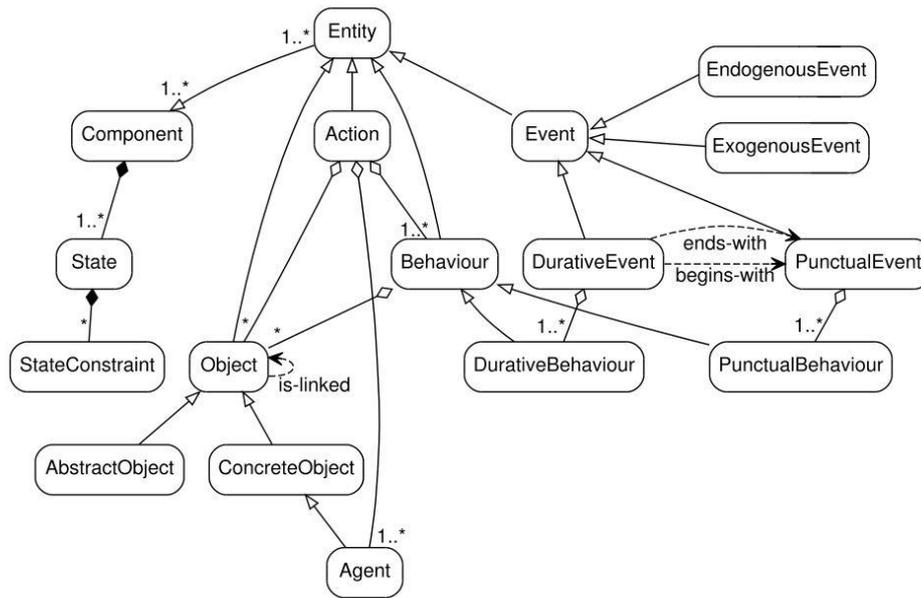


Figure 25 : Méta-modèle ontologique de DOMAIN-DL (Barot, 2014)

Dans (Bouville et al., 2017), les auteurs présentent **#FIVE**, un modèle qui se concentre sur deux concepts: les objets et les relations. Le principe de base du modèle consiste en la définition de types d'objets (voir Figure 26 - Object), regroupés en pattern d'un ou plusieurs types d'objets. Ensuite, pour chaque pattern, #FIVE permet de définir les relations possibles entre eux et les interactions utilisateurs réalisables (voir Figure 26 – Relation). Ce modèle étend également cette idée de définition des types d'objets en définissant des rôles aux acteurs. Cette approche simplifie donc le développement de l'environnement virtuel et permet d'optimiser les interactions ainsi que la création de contenu en catégorisant les éléments dans un ensemble de types d'objets. Cette approche pour définir la scène virtuelle sur lequel l'utilisateur doit interagir est plus proche de la manière dont on interagit naturellement avec le monde.

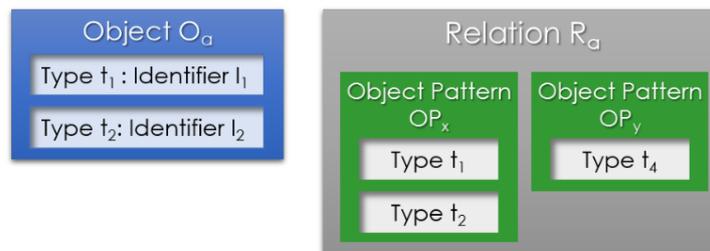


Figure 26 : Concepts d'Objet-Relation extrait de #FIVE (Bouville et al., 2017)

Ces concepts ont été étendus à une approche RV et RA dans (Richard et al., 2018) avec notamment la possibilité de préciser un système de suivi utilisé (Reconnaissance d'images, de modèle 3D etc....). L'objectif était de rendre n'importe quel objet reconnaissable pour faire de la réalité augmentée. Les auteurs ont pu mettre en place d'une part, la formation d'un opérateur télécom en réalité virtuelle, et d'autre part, l'assistance de l'opérateur en réalité augmentée avec une même description de l'environnement et un contenu commun (voir Figure 27).



Figure 27 : Application d'assistance à l'opérateur en réalité augmentée (en haut) et formation en réalité virtuelle (en bas) basées sur le modèle extrait de (Richard et al., 2018)

2.2.1.1.4 Synthèse

Tout bien considéré, les modèles « Object-Relation » offrent les mêmes possibilités que les autres modèles en termes de comportement ou de représentation de mondes virtuels interactifs, mais ils permettent également d'autres possibilités comme la co-manipulation ou encore de définir des actions et relations plus simplement et qui peuvent être facilement éditées. Il est en effet plus facile de créer / éditer une relation par pattern d'objet, ce qui rendra interactif n'importe quel objet correspondant dans l'environnement, plutôt que par objets comme pour les objets synoptiques par exemple. De plus, cette logique est très naturelle pour représenter, créer et éditer des environnements interactifs car elle se rapproche de la manière naturelle dont le monde peut être interprété. Le monde qui nous entoure, ainsi qu'un environnement virtuel, peut se résumer, en termes d'interactions, car il est peuplé d'objets et d'acteurs. Dans un EV, les acteurs peuvent être contrôlés par les utilisateurs ou être autonomes. L'environnement virtuel peut être modifié lors de la simulation, par des objets et des acteurs, via des actions.

2.2.1.2 Framework et modèles de scénarios

Dans la section précédente, nous avons identifié des modèles permettant de définir un environnement virtuel ainsi que les interactions qui y sont faisables. Cependant, pour qu'un environnement virtuel soit pleinement efficace afin de transmettre l'information souhaitée à l'utilisateur final, il est nécessaire que les actions qu'il réalise entraînent une conséquence sur l'environnement ou qu'il permette de valider la réalisation d'un objectif défini. Ainsi, nous nous intéresserons dans cette partie à la scénarisation d'actions dans un environnement virtuel. Après, une introduction dans la section 2.2.1.2.1, nous présenterons 3 frameworks existants ainsi que leur moteur de scénario.

2.2.1.2.1 Introduction aux modèles de scénarios et frameworks existants

En réalité virtuelle, comme nous le verrons plus tard en réalité augmentée, le besoin d'orchestrer des actions est présent, que ce soit pour une démarche procédurale (ex : gammes de maintenance ou d'assemblage) ou encore pour définir des objectifs de formation industrielle. Au vu des observations, aussi bien de la littérature scientifique que des problématiques

industrielles (chapitre 1), la complexité du scénario peut être définie comme une procédure (étape par étape), un scénario ouvert (simulation) ou un scénario ouvert avec des objectifs conducteurs. Un modèle de scénario doit donc proposer des concepts permettant au créateur de contenu, aux développeurs informatiques ou aux experts métiers des scénarios de formations ou des scénarios d'étude, d'implantation ou encore d'optimisation de processus industriel au sein de l'environnement virtuel ou réel. Les solutions existantes ont déjà proposé des concepts de scénarisation simples (scénarisation uniquement) ou des concepts plus avancés (avec par exemple des concepts pédagogiques et de suivi de l'apprenant, la définition du rôle de l'opérateur ou encore des scénarios de collaboration homme-robot (Ottogalli et al., 2019)). Nous allons détailler dans les prochaines sections les moteurs de scénarios mis en place dans 4 frameworks existants : #SEVEN (Claude et al., 2014), HUMANS (Lanquepin et al., 2013), MASCARET (Chevaillier et al., 2012) orientés RV. Nous présenterons enfin une synthèse des modèles et frameworks identifiés dans la littérature.

2.2.1.2.2 #SEVEN

Dans (Claude et al., 2014), le moteur #SEVEN fournit, en plus d'un moteur d'environnement et de modèles de rôle, un modèle de scénario basé sur les réseaux de Petri (Petri, 1966). Tout d'abord une des caractéristiques observées du moteur de scénario est qu'il est plutôt orienté développeur dans sa mise en place et qu'il est centré scénario pédagogique. Le moteur est indépendant du modèle d'environnement et permet d'exporter / importer un scénario en utilisant le formalisme XML. Il peut ainsi être utilisé avec le moteur d'environnement des mêmes auteurs, à savoir #FIVE (vu dans la section 2.2.1.1.3) ou bien un autre.

Pour ce qui est de l'architecture, le moteur de scénario #SEVEN se découpe en 2 parties : Le « contexte externe » qui gère les entrées / sorties et fait l'interface avec l'environnement ; La « spécification des scénarios » qui correspond à l'état initial de la représentation interne des scénarios possibles (Figure 28).

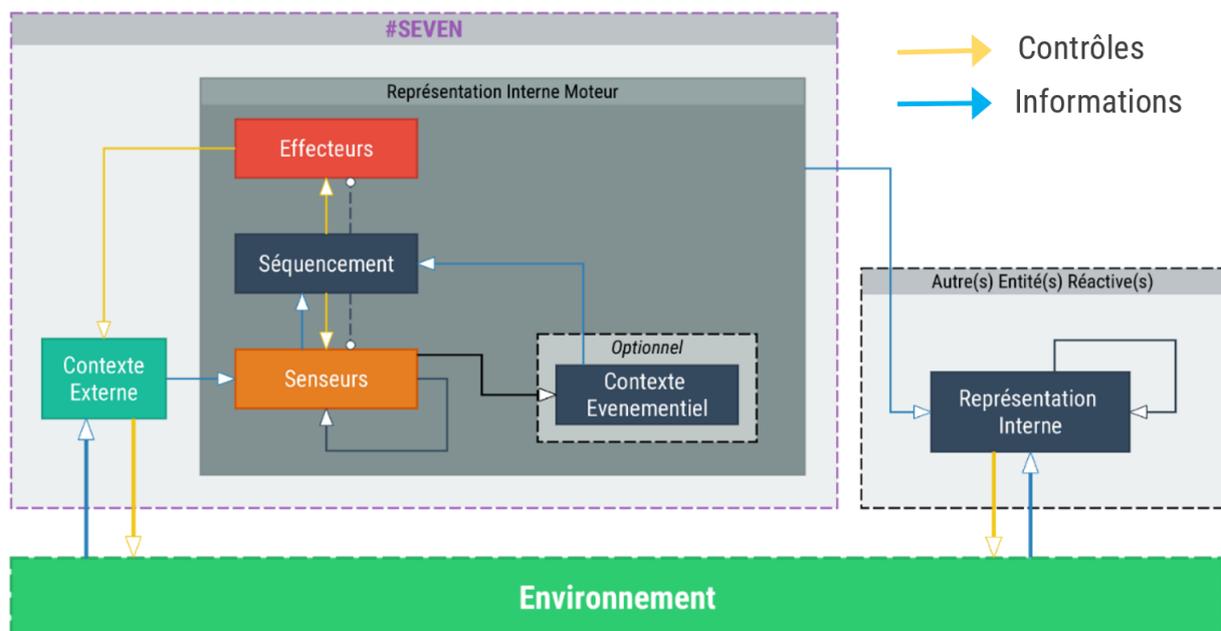


Figure 28 : Fonctionnement des composants principaux de la bibliothèque #SEVEN (Claude et al., 2014)

Dans le contexte externe, les entrées ont accès à toutes les entités de l'environnement tandis que les sorties peuvent envoyer des commandes aux entités. Chaque entité dispose de senseurs, d'un processus interne gérant la prise de décisions et d'effecteurs. Les senseurs servent à l'acquisition des données dans l'environnement et attendent que le contexte externe transmette les conditions souhaitées. Ce contexte de validation est appelé « contexte évènementiel ». Pour ce qui est des effecteurs, ils permettent de déclencher un comportement dans l'environnement au travers du contexte externe. Senseurs et Effecteurs sont représentés par une classe qui définit leur fonctionnement et leurs paramètres de configuration.

Ensuite pour la gestion du séquençement, #SEVEN se base sur les réseaux de Petri Sauf Hiérarchique. Les réseaux de Petri sont constitués de places, de transitions et de jetons permettant l'avancement du scénario. Nous verrons plus en détail le fonctionnement des réseaux de Petri dans la section 2.3.4. Cette représentation permet d'exprimer la séquentialité (l'ordonnancement des étapes), la concurrence et le parallélisme (Figure 29). Elle permet aussi de structurer hiérarchiquement un scénario avec la possibilité de le découper en sous-réseaux.

Dans les réseaux de Petri, on dit qu'une transition est sensibilisée lorsque toutes les places qui la précèdent possèdent un jeton. À cela #SEVEN rajoute leur concept de senseur et d'effecteur permettant d'ajouter une condition de validation en plus de la sensibilisation avec le senseur ou encore le déclenchement d'une action dans l'environnement avec l'effecteur lorsque cette même condition est validée. Les auteurs étendent également le concept des jetons en les transformant en collections de variables pouvant évoluer, récupérer et transmettre des informations au fur et à mesure de l'avancement du scénario.

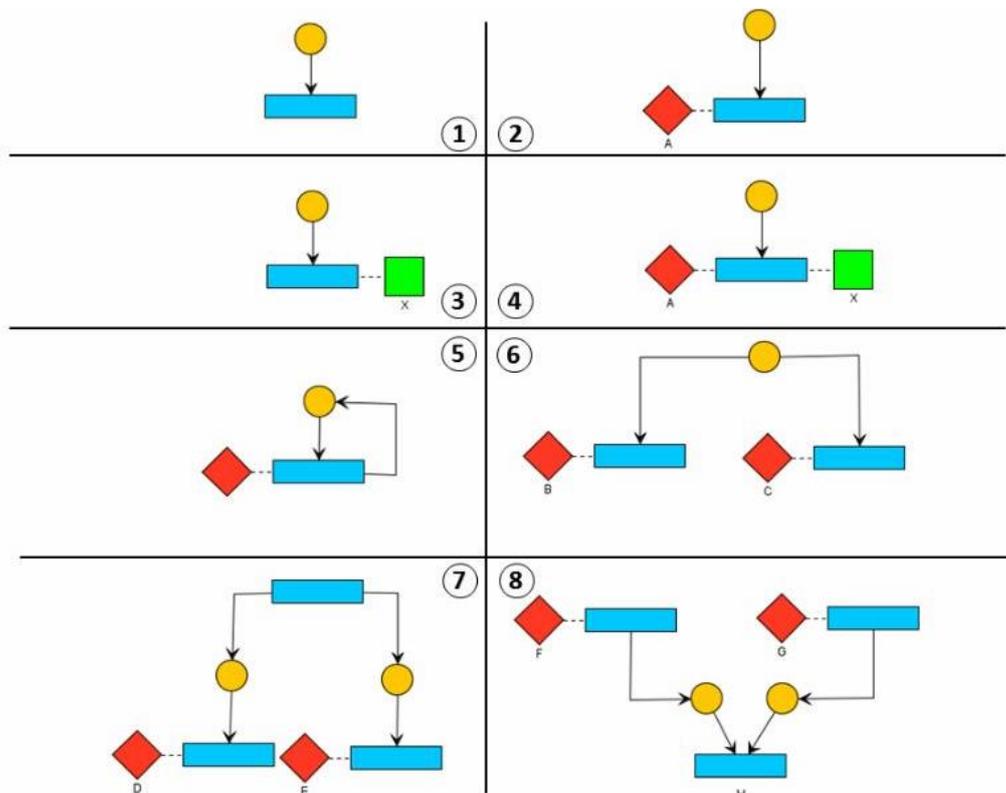


Figure 29 : Exemples de structures de base pour réaliser des ensembles de scénarios complexes avec : les places (en jaune) ; les transitions (en bleus) ; les senseurs (en rouge) et les effecteurs (en vert) (Claude et al., 2014)

Les auteurs présentent aussi une interface permettant de créer et d'éditer des scénarios sous la forme de réseaux de Petri (Figure 30).

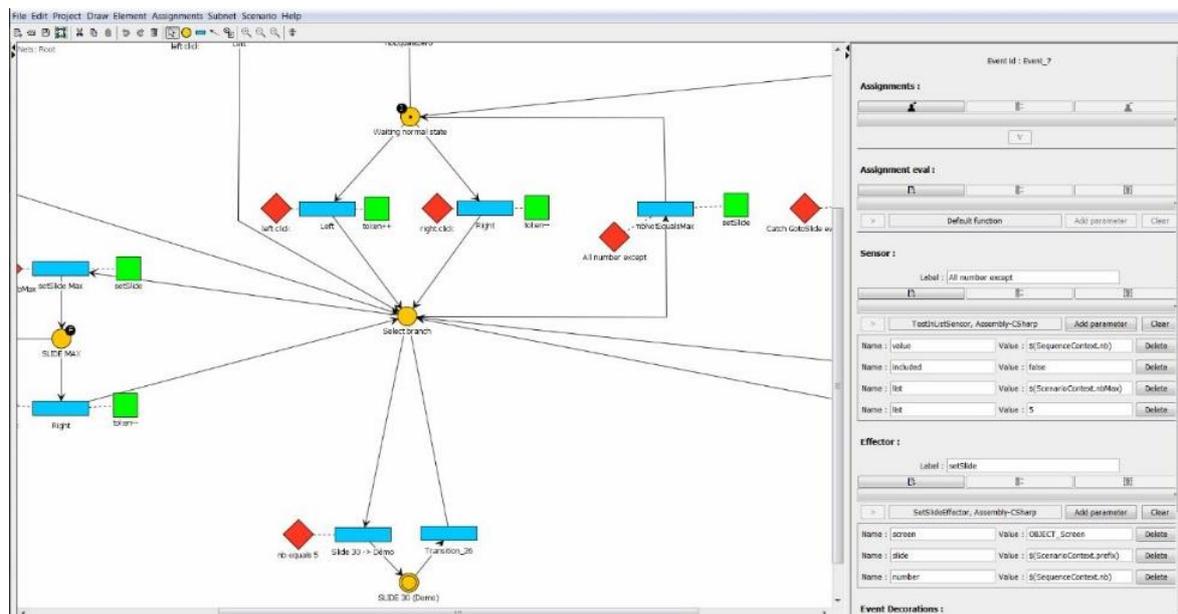


Figure 30 : Interface de l'éditeur de réseaux #SEVEN (Claude et al., 2014)

Pour conclure sur #SEVEN, l'utilisation de Petri Net permet une représentation claire du scénario et de son déroulement. Cependant, dans un scénario à grande échelle, tel qu'un assemblage industriel complexe, cela peut rapidement rendre cette représentation compliquée. Une autre limite devient claire si, dans le cas d'une action non sollicitée comme un démontage, le scénario ne peut revenir en arrière que si ce cas a été défini au préalable. De même, dans le cas d'un environnement entièrement ouvert avec la possibilité de faire des erreurs imprévues dans le scénario de base, ce modèle nécessitera que chaque éventualité soit prévue lors de la rédaction du scénario. De plus, comme le contexte de ce framework est la formation, il se concentre uniquement sur la réalité virtuelle, et ne prend pas en compte la possibilité d'avoir un scénario commun entre la réalité virtuelle et la réalité augmentée.

2.2.1.2.3 HUMANS

Dans (Lanquepin et al., 2013), les auteurs ont présenté **HUMANS (HUMAN Models based Artificial eNvironments Software platform)** comme un framework utilisant des ontologies, destiné à l'entraînement dans un environnement virtuel. Le framework est centré sur l'apprenant. Il est composé de 4 moteurs décisionnels : **REPLICANTS** pour la gestion des personnages virtuels ; **SELDON** pour la scénarisation ; **WORLD MANAGER** pour la gestion de l'exécution du monde et **MONITOR** pour la gestion du suivi de l'apprenant (Figure 31). Ces 4 moteurs communiquent avec les données **HUMANS-DL** composé de 3 modèles de connaissance que l'on peut aussi définir comme langage ontologique : **ACTIVITY-DL**, **WORLD-DL/DOMAIN-DL** (présenté dans la section précédente) et **CAUSALITY-DL**.

Côté scénarisation, SELDON se compose de 2 sous-modèles : **TAILOR** et de **DIRECTOR**. TAILOR s'occupe de la génération de situation d'apprentissage en sélectionnant les objectifs pertinents adaptés à la ZPD (Zone Proximale de Développement) pour garantir un apprentissage efficace.

Tandis que DIRECTOR planifie la génération de scénarios correspondant aux objectifs scénaristiques et aux données apportées par TAILOR. Pour cela, il utilise un ensemble d'ajustements :

- Les Happenings : Déclenchement d'événements ponctuels et exogènes ou activation / désactivation d'événements duratifs.
- Le Late Commitment : Permet l'ajout pendant la simulation de reprécisions de l'état initial de l'environnement. (Des états peuvent être initialement inconnus et se préciser durant la simulation)
- Les Contraintes d'occurrence : Ajout de contrainte sur des comportements probabilistes (Ex : probabilité d'une fuite d'eau)

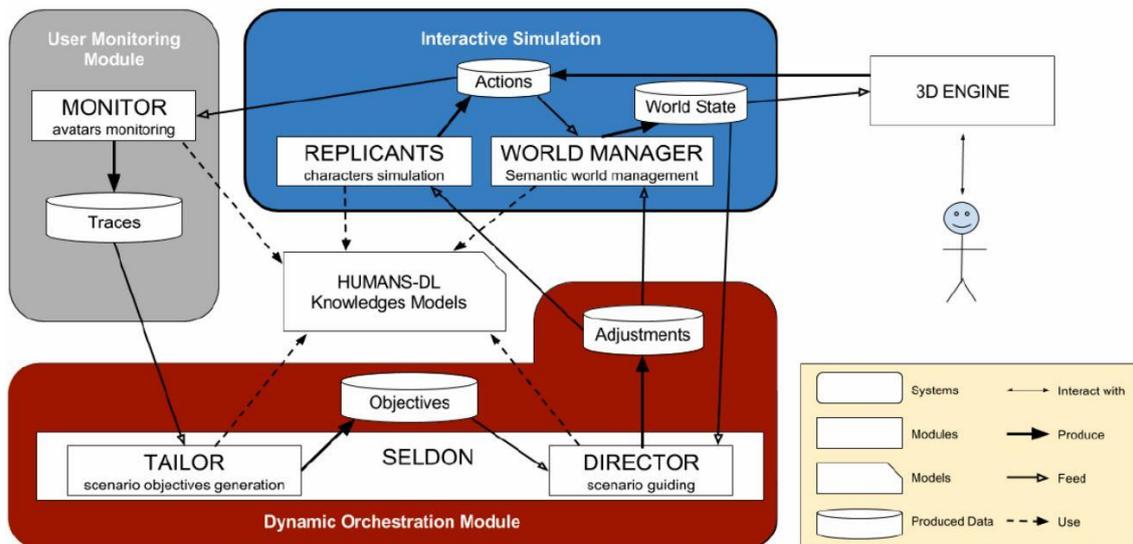


Figure 31 : Architecture de la plateforme HUMANS

Dans la gestion de la scénarisation, ce sont ensuite les langages ontologiques ACTIVITY-DL et CAUSALITY-DL qui vont nous intéresser. ACTIVITY-DL (Figure 33 et Figure 34) permet l'organisation hiérarchique d'un ensemble de tâches, dont les propriétés se réfèrent à des entités décrites dans DOMAIN-DL (vue dans la section précédente 2.2.1.1.3). Comme montré dans le méta-modèle (Figure 33), une tâche est décomposable en sous-tâche. Chaque tâche peut être liée à une opération qui correspond à une action sur une ou plusieurs ressources. Une tâche peut également posséder des relations temporelles, des conditions, un contexte ainsi que des tags. La relation temporelle (entre tâches fille d'une tâche mère commune) est représentée par des relations d'Allen, qui correspondent à des relations de temporalité entre tâches filles d'une même tâche mère (Ex Figure 32 : la tâche A commence après la tâche B). Ensuite, une condition peut être soit : une précondition, une condition de satisfaction ou bien une condition d'arrêt. Pour finir, le contexte se constitue des ressources associées aux tâches et se transmet de tâche mère en tâche fille. Un exemple d'utilisation du modèle est illustré dans la figure 34.

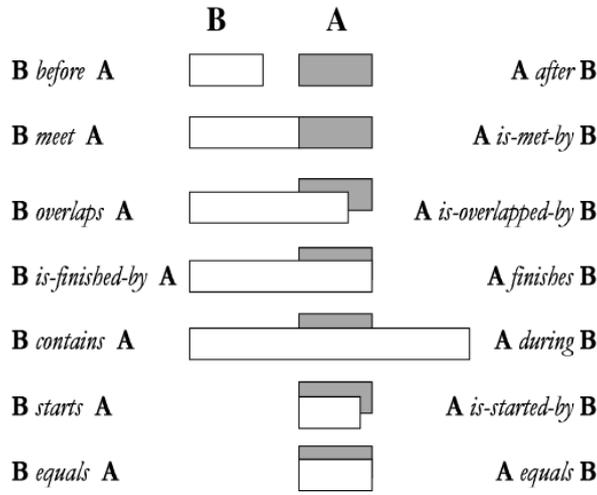


Figure 32 : Relations temporelles d'Allen (Allen, 1983)

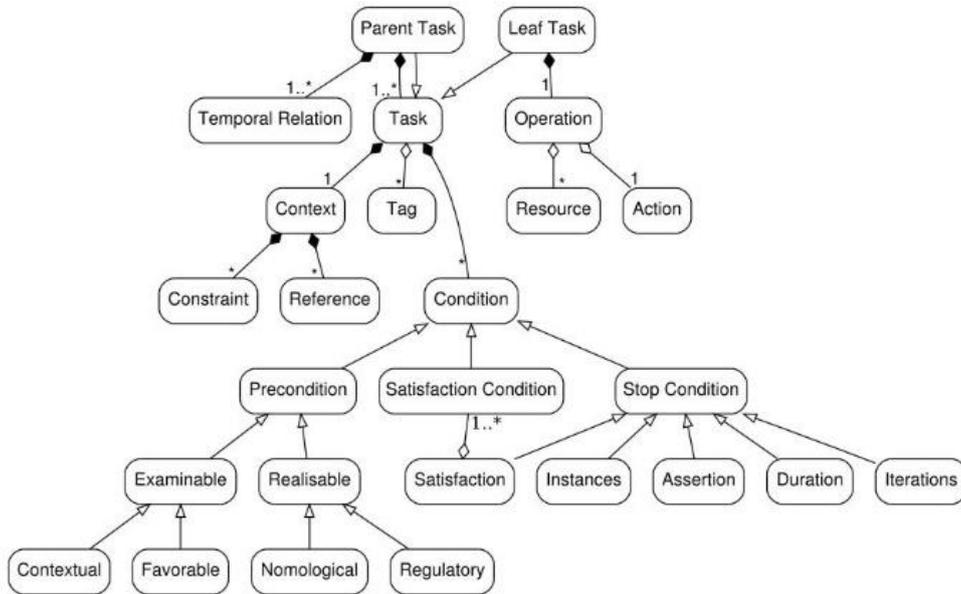


Figure 33 : Méta-modèle ontologique de ACTIVITY-DL

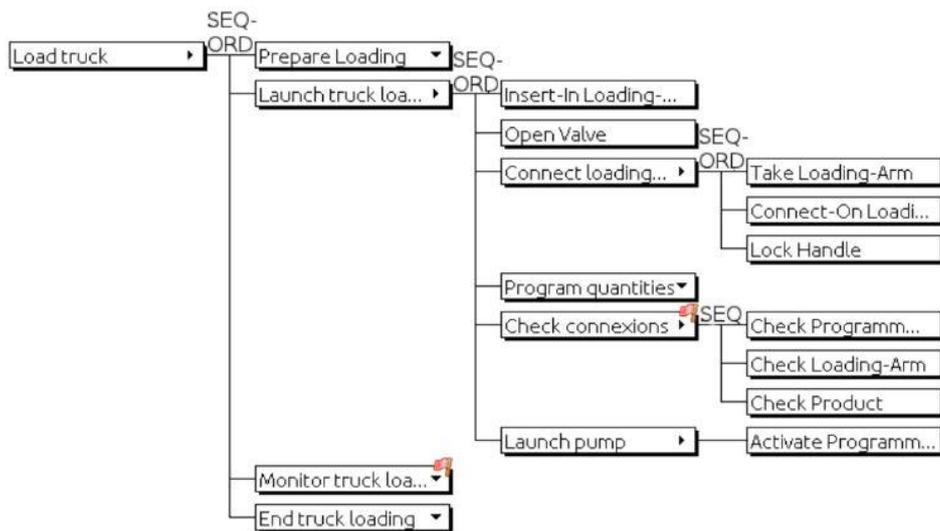


Figure 34 : Exemple de modèle représenté en ACTIVITY-DL

Tandis que ACTIVITY-DL permet de définir le séquençage, CAUSALITY-DL (Figure 35), comme son nom l'indique, permet de gérer le principe de causalité et se représente sous la forme d'un graphe orienté acyclique (Figure 36). Comme ACTIVITY-DL, celui-ci s'appuie sur les données de DOMAIN-DL. À partir de l'environnement, il est possible de spécifier des événements dans CAUSALITY-DL sur lequel une ou plusieurs relation(s) de causalité peuvent être liées. La relation utilise les portes logiques « ET » et « OU » pour effectuer respectivement, une conjonction ou une disjonction. Par ailleurs, des « barrières » de prévention ou de protection peuvent être associées à un évènement et peuvent interrompre la causalité. Cela permet de préciser un évènement comme ponctuel (activable une seule fois) si la barrière est « active » ou durable (répétable) si la barrière est passive.

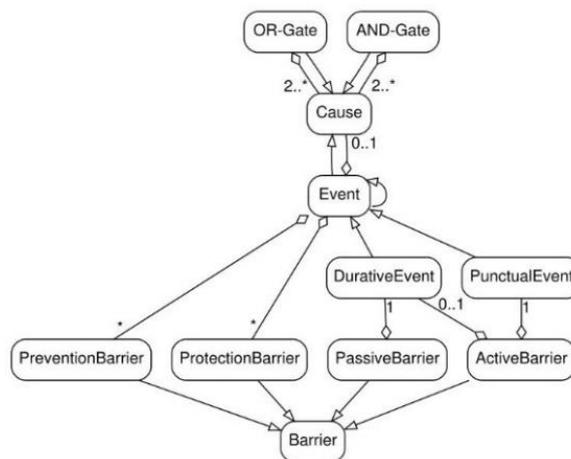


Figure 35 : Méta-modèle ontologique de CAUSALITY-DL

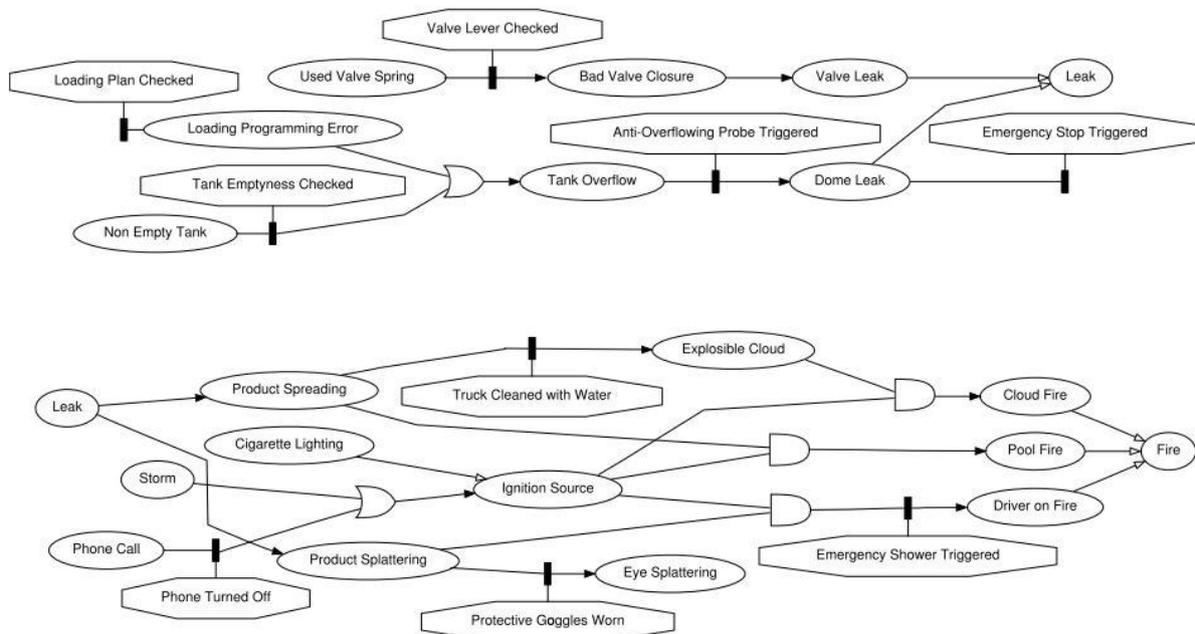


Figure 36 : Exemple de modèle représenté en CAUSALITY-DL

Pour conclure, HUMANS fournit un moteur de scénarios (principalement un modèle d'activité) appelé SELDON, un moteur d'environnement (monde et causalité) et un moteur de suivi des apprenants. Il offre également des possibilités avancées pour les avatars autonomes. Le cadre

est vaste et modulaire. SHELDON permet à l'utilisateur de faire des erreurs, car il utilise la zone de développement proximal (ZPD) pour assurer un apprentissage efficace. Le concept sous-jacent est que les objectifs d'apprentissage s'adaptent et ne sont toujours ni trop simples ni trop difficiles. Cependant, bien que ce modèle soit complet, il est très hétérogène en termes de représentations. Ces différentes représentations des données pour chaque moteur obligent à passer par différents modèles pour créer l'environnement, et nécessitent donc une connaissance de chacun. De plus, les représentations se veulent interprétables et orientées métier/formateur, mais du point de vue de la prise en main (sans interface user-friendly en sur-couche), ces notions peuvent être plus compliquées à appréhender pour un non-initié et une personne non-technique.

2.2.1.2.4 MASCARET

Dans (Chevaillier et al., 2012), les auteurs présentent **MASCARET (Multi Agent System for Collaborative, Adaptive & Realistic Environments for Training)**. Il s'agit d'une plateforme logicielle et d'un modèle générique qui est orienté pour les systèmes industriels. La plateforme utilise le langage UML dans sa représentation et se base sur deux méta-modèles (Figure 37) : un moteur d'environnement (VEHA) que nous avons décrit dans la section 2.2.1.1.3 et un moteur d'activité « HAVE ». HAVE décrit les interactions et les activités possibles dans l'environnement par les utilisateurs et les agents autonomes. De plus, MASCARET intègre « MATS », un sous-système dédié aux fonctionnalités de tutorat dans un EVAH.

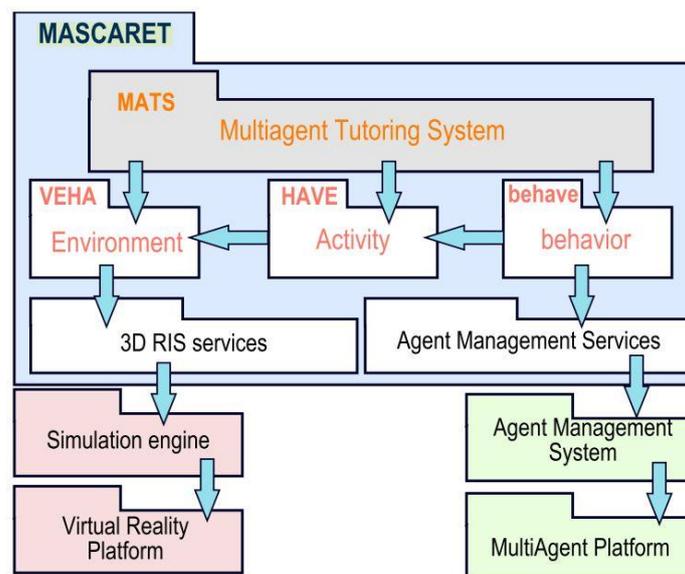


Figure 37 : Architecture de la plateforme MASCARET

En effet HAVE (Human Activities in a Virtual Environment) a pour but de décrire les activités d'un collectif d'agents (humains ou autonomes). Une activité est définie dans le modèle comme étant « *un ensemble d'enchaînements possibles d'actions à réaliser par une structure organisationnelle afin d'atteindre un objectif* ». La structure organisationnelle est un concept défini dans HAVE et correspond à la description abstraite d'un collectif d'agents et définit leur rôle « *roleClass* » (Figure 38 – à gauche). Une action, quant à elle, est définie par des conditions de faisabilité avec des expressions logiques sur les entités de l'environnement de VEHA. Pour finir, le séquençage est formalisé sous forme d'automates (Figure 38 – à droite).

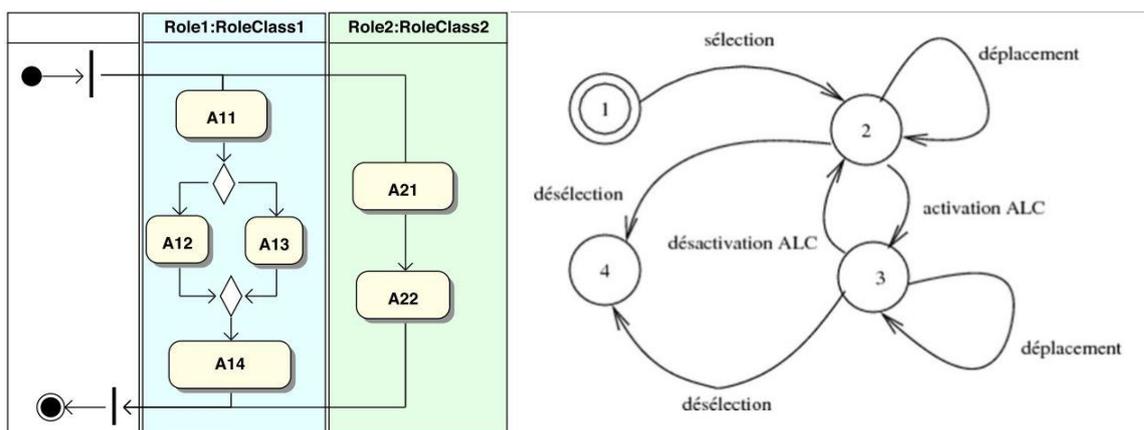


Figure 38 : (à gauche) Exemple d'automates définis suivant HAVE ; (à droite) Exemple d'agencement d'actions assignées à différents collectifs.

Comme #SEVEN, il utilise un langage unifié qui est compréhensible par les développeurs en particulier et facile à traduire dans une interface utilisateur. Comme HUMANS, grâce au modèle proposé, le framework offre une liberté à l'utilisateur en permettant au participant de faire des erreurs. De plus, le modèle fournit des notions de traces de l'apprenant avec MATS.

Cependant HAVE ne semble pas prévu pour pouvoir déclencher des événements dans l'environnement et s'arrête à l'agencement d'action. Comme pour #SEVEN et HUMANS, MASCARET est plutôt axé sur un usage orienté réalité virtuelle et ne mentionne pas d'usages en réalité augmentée.

2.2.2 Modèles orientés réalité augmentée

Lorsque l'on se concentre sur la réalité augmentée, qui est une partie importante de la digitalisation industrielle (Masood & Egger, 2020), nous constatons que deux grandes phases doivent être définies : la reconnaissance de l'environnement réel puis la définition du modèle permettant d'éditer une procédure faisable en réalité augmentée. Cependant, dans les modèles identifiés, ces deux phases sont très liées, c'est pourquoi nous avons décidé de ne pas les séparer dans l'analyse comme cela est fait pour la réalité virtuelle.

En ce qui concerne la reconnaissance de l'environnement pour permettre à une application de faire de la réalité augmentée, il existe quatre grandes méthodes selon (Havard, 2018): la reconnaissance basée marqueur, la reconnaissance basée points d'intérêts, la reconnaissance basée modèle 3D et la reconnaissance basée SLAM (Simultaneous Localization And Mapping). Pour les trois premières méthodes, il s'agit de reconnaître un objet connu dans la scène réelle, alors que pour la dernière méthode il s'agit de construire la forme en trois dimensions d'un environnement réel inconnu. Au-delà des avantages et inconvénients de chacune de ces méthodes, il est intéressant d'en définir le principe général. En effet, l'objectif des algorithmes de réalité augmentée est de définir la manière de reconnaître l'environnement réel, soit grâce à un marqueur placé dans la scène, soit en se basant sur les points d'intérêts, tels que SIFT, SURF ou ORB (Ruble et al., 2011) de l'objet à reconnaître, soit en faisant de la reconnaissance de modèle 3D (Park et al., 2020), ou soit en estimant la 3D de la scène réelle à partir des capteurs de la caméra. Au final, les paramètres (type d'algorithme et leurs configurations associées) permettant

de faire la reconnaissance pour faire de la réalité augmentée font partie du modèle d'environnement RA.

L'étude de la littérature montre en effet que les modèles RA sont fortement couplés entre modèles d'environnements et modèles de scénarios. Ainsi, dans (Friedrich et al., 2002), les auteurs proposent une architecture basée sur le Web et un modèle UML de RA pour la création et le déploiement de contenu définissant des procédures de maintenance ou d'assemblage, étape par étape. De même, (Lee & Rhee, 2008) détaille un outil « conscient du contexte », basé sur une ontologie, qui permet d'adapter l'application RA à l'expérience de l'utilisateur et aux capacités des appareils utilisés. De la même manière, (C.-J. Su et al., 2012) propose une architecture associée à un modèle UML pour décrire les opérations et la maintenance (O&M). Afin de faciliter l'édition de contenu (Zhu et al., 2013) propose un outil auteur permettant aux experts métiers et opérateurs du domaine d'éditer une procédure.

Cependant, dans ces travaux, la manière de définir et reconnaître l'objet et la manière de définir la procédure de maintenance sont décrites avec un modèle de données unique. Par conséquent, dès qu'une procédure change, il est nécessaire de la rééditer entièrement ; il n'est pas possible d'utiliser les caractéristiques de l'objet virtuel qui seraient définies par un modèle d'environnement commun à toutes les procédures, pour adapter une maintenance en réalité augmentée ou en créer une nouvelle.

Quelques approches tendent à distinguer les deux modèles. (Martínez et al., 2014) propose de modulariser la manière de reconnaître l'objet pour la réalité augmentée et de décrire la procédure. Il propose ainsi les modules de reconnaissance des marqueurs, de vision par ordinateur et de rendu 3D qui pourront être mis à jour en fonction des évolutions technologiques. En ce qui concerne la procédure, il définit également 3 concepts : **Procedure**, **Job** et **Task** (voir Figure 39). La Procedure représente l'objectif global (i.e. « réparer l'écran ») le Job représente des groupes de Task permettant de réaliser la Procedure. De cette manière le Job peut être réutilisé dans plusieurs procédures de maintenance. Le concept de Task représente le plus petit geste effectué dans une maintenance. Grâce à cette proposition, il est possible d'utiliser des Job communs dans différentes Procedure. Ainsi, le Job ne doit être modifié qu'une seule fois pour être mis à jour au sein de chaque Procedure l'utilisant. Le principal inconvénient est que le concept de Task ne sépare pas l'action de l'objet sur lequel il est effectué, comme, par exemple, une action de dévissage. Il n'existe donc pas de concept d'action réutilisable sur différents objets. (Havard et al., 2017) renforcent la proposition de (Martínez et al., 2014) en séparant la description de l'action, de l'objet sur lequel elle agit. De plus, leur proposition, permettant de décrire une procédure de réalité augmentée et une procédure de réalité virtuelle avec une seule édition, est intéressante. Les auteurs séparent bien, la description de l'objet, le modèle d'environnement, des actions attendues sur ces objets ainsi que de leur ordre, i.e. modèle de scénarios.

Bien que la modélisation pour la création en RA puisse être complexe, elle reste plus simple que celle pour la RV, puisque la réalité augmentée est principalement faite pour présenter de l'information tandis que dans la réalité virtuelle, il est nécessaire de prendre en compte les interactions faisables par l'utilisateur sur les éléments virtuels, les agents virtuels...

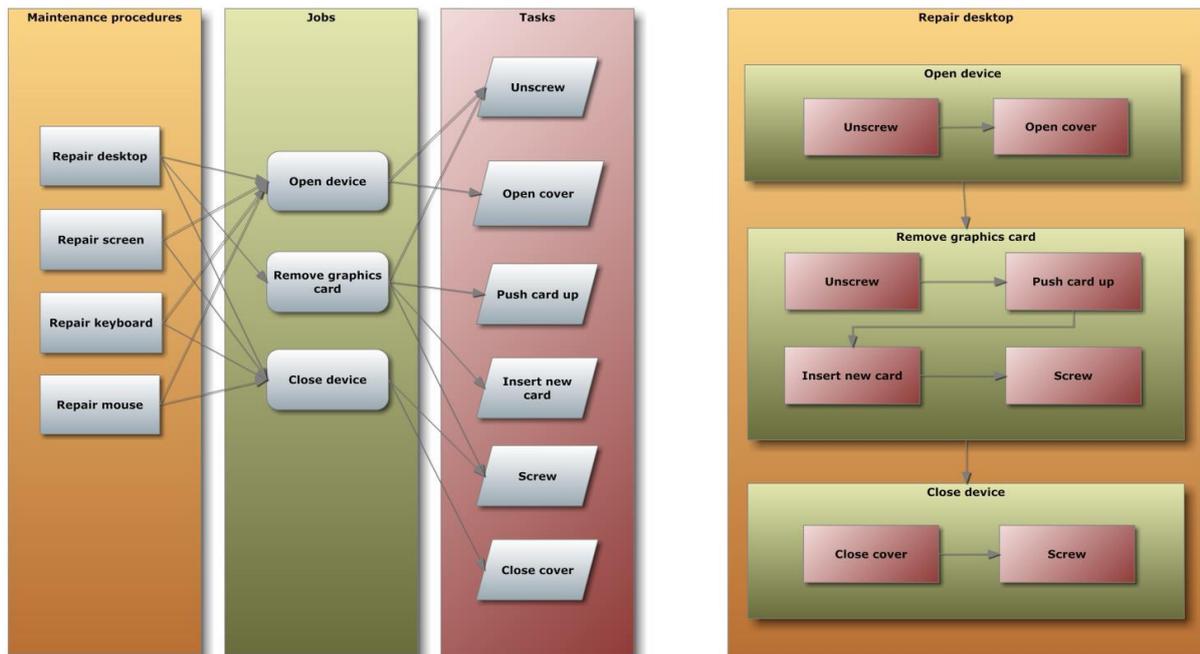


Figure 39 : Illustrations du modèle de données basé sur les Procédure, Job et Task. Extrait de (Martínez et al., 2014).

2.2.3 INOOVAS : Un framework compatible RA/RV

Concernant les modèles destinés à être utilisés aussi bien en RV qu'en RA, (Havard et al., 2017) a proposé **INOOVAS** (**IND**ustrial **Ontology** for **Operation** in **Virtual** and **Augmented** **Scenes**), une ontologie qui vise à permettre à un expert métier de créer et d'éditer à la fois un guide en réalité augmentée et une formation de maintenance en réalité virtuelle. Ce modèle se concentre sur la création de contenu pour ces deux objectifs dans le cadre d'opérations de maintenance industrielle.

Pour permettre la définition de scénario, l'ontologie qui y est consacrée (Figure 40) définit un scénario comme étant une opération « **AVROperation** ». Chaque AVROperation est composée d'étapes « **AVROperationStep** ». Une étape correspond à une ou plusieurs actions « **AVRActions** » qu'il faudra réaliser afin de la valider. Dans le cas de la réalité augmentée, il est également possible de définir quel algorithme de suivi devra être utilisé pour localiser l'objet sur lequel interagir. L'ordonnancement des étapes se fait alors en utilisant des listes d'étapes et d'actions.

L'avantage de cette ontologie est son processus de création de contenu par un expert métier dont l'usage est ensuite commun en RA ou RV. Il permet donc à l'expert de définir une unique fois les opérations que l'opérateur aura à effectuer afin qu'il soit utilisable dans les deux environnements RA/RV (voir Figure 41).

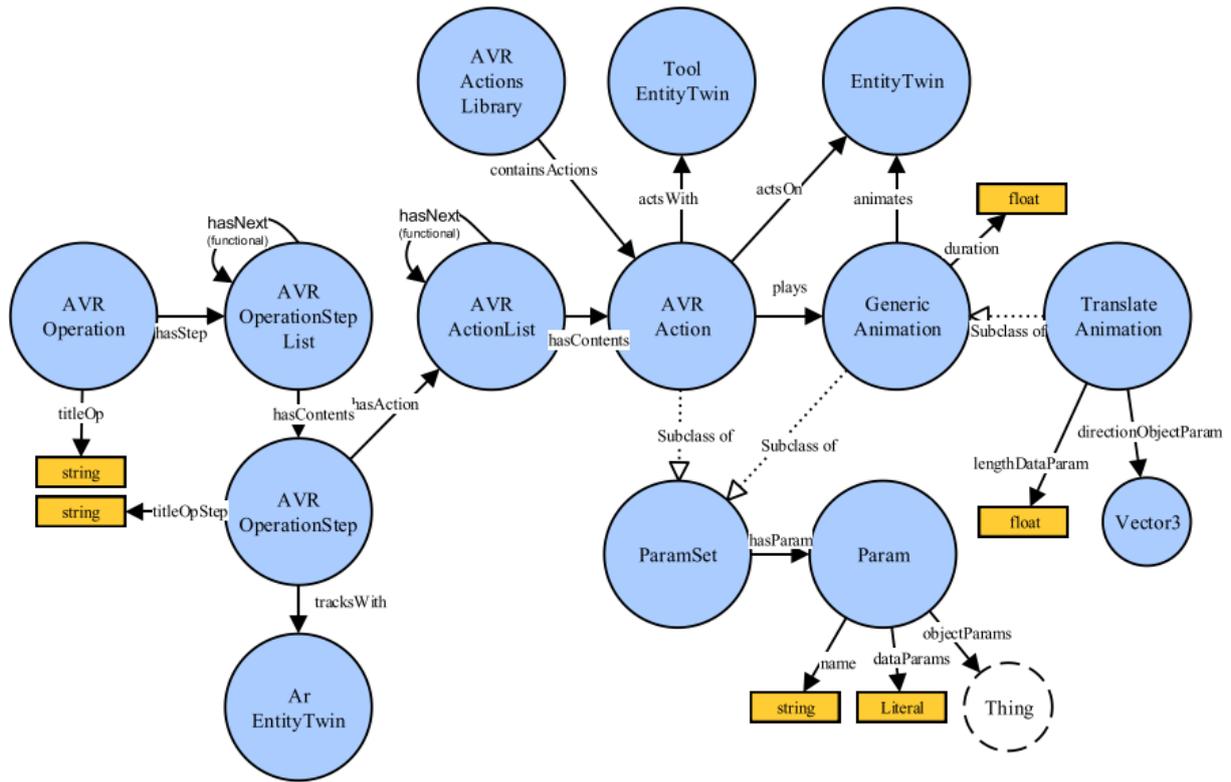


Figure 40 : Ontologie des opérations sur des systèmes industriels proposée par INOOVAS



Figure 41 : Processus de création de contenu et d'usage en RA/RV via l'ontologie INOOVAS, extrait de (Havard et al., 2017)

En ce qui concerne la partie édition de contenu, elle est faisable au sein de la réalité virtuelle. L'expert s'immerge en réalité virtuelle face au système sur lequel il doit éditer une procédure. Puis, il utilise les contrôleurs pour sélectionner chaque composant sur lesquels agir (voir Figure 42 – a). Ensuite, il sélectionne le type d'opération à effectuer sur le composant (voir Figure 42 – b), pour enfin définir les paramètres de l'opération sélectionnée (voir Figure 42 – c).

Cependant, la traduction de cette ontologie dans une application RV/RA peut être compliquée par sa représentation spécifique des données qui est orientée triplet et non orientée objet.

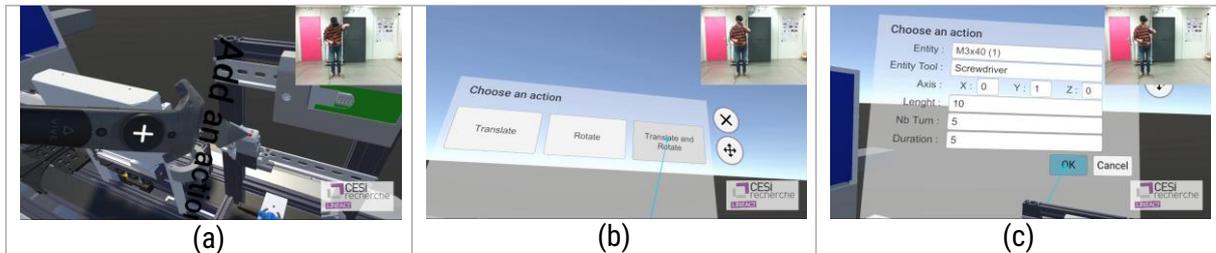


Figure 42 : Étape d'édition de procédure en réalité virtuelle basée sur le framework INOVAS. (a) l'expert sélectionne le composant, (b) l'expert sélectionne le type d'opération, (c) l'expert paramètre l'opération.

2.2.4 Synthèse des modèles

Afin de conclure sur les modèles d'environnements et de scénarios RA/RV, nous avons synthétisé dans le tableau 1 les différents modèles présentés précédemment ainsi que d'autres modèles identifiés dans la littérature en fonction des propriétés définies en 2.2. Les modèles existants sont donc comparés suivant ces caractéristiques :

- S'ils sont utilisables en réalité augmentée ou / et en réalité virtuelle,
- S'ils sont collaboratifs,
- S'ils permettent une orchestration d'objectifs simples ou de tâches (les uns après les autres, comme une procédure de maintenance par exemple) ou des scénarios plus avancés (formation d'apprentissage),
- S'ils ont le principe de causalité (i.e les actions effectuées sur une des entités du monde ont des conséquences sur d'autre(s)),
- S'ils permettent de contenir la description de l'environnement,
- Le domaine des cas d'usages présentés.

Tableau 1 : Synthèse des modèles existants

Modèle	Technologie ? (AR/VR)	Framework						Collaboratif ?	Agents autonomes ?	Causalité ?	Cas d' usages industriels ?	Cas d'usages	
		Modèle d'env.			Modèle de Scénario								
		Comportement	Objets synoptiques	Objets-Relations	Scénario procédural	Scénario a/ objectifs	Scénario ouvert						
#FIVE (Bouville et al., 2017)	Réalité virtuelle			X				X				Formation de préparation à la chirurgie, démonstrateur d'interaction	
#SEVEN (Claude et al., 2014)					X	X		X	X	X	X	Scénarios d'apprentissage pour la chirurgie, la formation pour l'entretien d'une presse à injecter.	
HCSM (Cremer et al., 1995)		X							X	X		Simulation de trafic	
HPTS++ (Lamarche & Donikian, 2002)		X							X	X		Simulation d'interaction et de comportement	
HUMANS (Lanquepin et al., 2013)				X	X	X	X	X	X	X	X	X	Application d'apprentissage (ARAKI : conception d'environnement virtuel pour la formation aux activités de maintenance, NIKITA, SIMADVF)
Marigold (Willans & Harrison, 2001)			X										Générer un environnement virtuel et un prototype d'interaction
MASCARET (Chevaillier et al., 2012)				X	X	X	X	X	X	X	X	X	SÉCURÉVI : Scénario pour la formation des sapeurs-pompiers à la gestion opérationnelle et au commandement, et GASPARET : Management de l'Aviation Sur Porte-Avions par Réalité Virtuelle, réalisent des études de dimensionnement, et d'ergonomie
Smart-Objects (Kallmann & Thalmann, 1999)			X						X				Simulation interactive de bureau intelligent
Starfish (Badawi & Donikian, 2004)			X						X				Démonstrateur d'interaction simple sur Objet Synoptique (tirer une porte)
STORM (Mollet et al., 2007)				X								X	Formation industrielle
(Martínez et al., 2014)	Réalité augmentée				X						X	Maintenance du système générique étape par étape en RA	
(Gajšek et al., 2020)					X			X	X		X	Maintenance du système industriel pas à pas avec HoloLens	
ARAUM (Erkoyuncu et al., 2017)					X							X	Réparation de moteur étape par étape en RA
ARML (OGC, 2015)													Décrire une scène de réalité augmentée, afficher des actifs RA dans le monde
COIVA (Hervás et al., 2013)					X								Trouver des objets en RA dans un bâtiment pour des besoins spécifiques selon le contexte de l'utilisateur.
Context-aware AR (Zhu et al., 2013)					X				X			X	Maintenance du système industriel pas à pas et éditable par les opérateurs
EASE-R ³ (Lamberti et al., 2014)					X				X			X	Maintenance du système en mode guidé ou assisté par un expert
MAR (ISO/IEC, 2014)					X							X	Décrivez la cible, le suivi et l'actif RA à ajouter. Utilisé dans la maintenance, la médecine, la localisation RA, les jeux. Toute la logique est codée en dur.
OARP (C.-J. Su et al., 2012)					X							X	Maintenance étape par étape de la voiture en RA

Modèle	Technologie ? (AR/VR)	Framework						Collaboratif ?	Agents autonomes ?	Causalité ?	Cas d' usages industriels ?	Cas d'usages
		Modèle d'env.			Modèle de Scénario							
		Comportement	Objets synoptiques	Objets-Relations	Scénario procédural	Scénario a/ objectifs	Scénario ouvert					
U-Car (Lee & Rhee, 2008)										X	Maintenance étape par étape de la voiture en RA	
(Ottogalli et al., 2019)	Les deux				X	X		X	X		X	Simulateur de chaîne d'assemblage final d'avion, simulateur de cellule haute tension, simulateur de machine-outil.
D3PART (Lacoche et al., 2017)					X			X				Aménagement collaboratif d'une salle
INOOVAS (Havard et al., 2017)					X	X			X		X	Scénario de formation sur une maintenance pas à pas en RV et guidage opérateur en RA
UMI3D (Casarin et al., 2018)								X	X			Jeu de construction collaboratif

Ainsi, certaines tendances concernant les modèles de la littérature peuvent déjà être observées. Les modèles sont pour la plupart conçus spécifiquement pour la RV, ou spécifiquement pour la RA, ce qui limite leurs utilisations. Pourtant, bien qu'une partie des données nécessaires à la mise en place de scénarios pour ces deux environnements comportent des données communes, il y a un manque de modèles permettant d'être utilisés à la fois en RV et en RA. Nous pouvons également souligner que, comme expliqué dans la section 1.1, les cas d'utilisation sont principalement des applications dédiées au partage de connaissance et d'apprentissage pour la RV et des applications de description de procédure (maintenance, assemblage, ...) pour la RA. Or, il est possible de retrouver un socle commun entre la RA et la RV autour de la description de procédure qu'il serait intéressant d'avoir afin de permettre l'usage dans les deux modalités.

Des limites peuvent également être observées. Concernant les modèles d'environnement, les modèles orientés comportement sont limités par la nécessité de réadapter chaque objet lorsque de nouveaux objets ou agents autonomes sont implémentés, puisqu'ils doivent savoir interagir entre eux. Pour les modèles orientés « objets synoptiques », la principale limitation réside dans la nécessité de définir plusieurs fois les mêmes actions sur plusieurs objets dans le cas d'opérations d'assemblage ou d'actions collaboratives. Bien qu'ils ne soient pas sans avantages, ils ne sont guère évolutifs.

En synthèse, nous constatons que les modèles « objets-relations » offrent les mêmes possibilités que les autres modèles en termes de comportement ou de représentation des mondes virtuels interactifs, tout en permettant aussi des capacités complémentaires (co-manipulation, actions simples facilement éditables). De plus, cette logique est efficace pour représenter, créer et éditer des environnements interactifs car elle se rapproche de la manière dont le monde peut naturellement être interprété. Le monde physique, ainsi qu'un environnement virtuel, peut se résumer en termes d'interactions, car il est peuplé d'objets et d'agents. Dans un EV, les agents

peuvent être contrôlés par des utilisateurs ou être autonomes. L'environnement virtuel peut être modifié au cours de la simulation par des objets et des agents via des actions. Comme, parmi tous les modèles présentés, les modèles « objets-relations » sont les plus complets, cette direction a été choisie pour le modèle proposé dans le chapitre 3.

Concernant les modèles de scénarios, la plupart de ceux passés en revue permettent la création de scénarios procéduraux (étape par étape). Pour ceux qui offrent plus de liberté en termes de scénarisation, ils restent toutefois limités en ce qui concerne la possibilité pour les utilisateurs de faire des erreurs, la faisabilité de définir des scénarios ouverts. Par exemple, la possibilité de revenir en arrière et d'annuler une action n'est pas intégrée sauf si elle est prévue dans le scénario. Par ailleurs, ils se concentrent uniquement sur la réalité virtuelle.

Finalement, à partir de cette recherche et en regroupant les points positifs ainsi que les limites observées, plusieurs critères ont été définis pour le modèle à mettre en œuvre :

- Permettre de créer un environnement virtuel et/ou augmentée (interactions réalisables, rôles, conséquences des interactions dans l'environnement ...), indépendamment du dispositif d'affichage,
- Permettre de définir des objectifs d'apprentissage et des scénarios avec une orchestration d'objectifs.
- Rester indépendant du nombre d'agents, qu'ils soient contrôlés par une personne réelle ou simulée,
- Permettre l'extensibilité et l'évolutivité du modèle.

Notre contribution est de proposer un modèle qui répond à ces critères et facilitera ainsi la mise en place d'un environnement aussi bien en RV qu'en RA, allant d'une simple procédure étape par étape à un scénario d'apprentissage complexe impliquant des objectifs plus globaux.

Puisque, parmi tous les modèles présentés, les modèles « Objets-Relations » sont les plus complets dans la manière de représenter l'environnement, cette direction a été choisie pour le modèle proposé. Nous avons également vu différents modèles de scénario ayant chacun leur formalisme permettant de représenter la séquentialité. Cependant, afin de permettre de rendre accessible l'édition de scénarios à des personnes sans compétences en développement informatique, il est indispensable de fournir une représentation graphique du scénario. C'est pourquoi, nous ferons état des différents langages graphiques pouvant accompagner notre proposition dans la section suivante.

2.3 Formalisation du séquençement sous forme de graphe

Même si les parties précédentes ont permis d'identifier les limites et d'identifier plusieurs critères pour le modèle à mettre en œuvre, toute la partie de scénarisation devra se baser sur une représentation graphique pour symboliser le séquençement afin de rendre accessible cette édition de scénarios à des experts métiers. C'est pourquoi, il est important d'identifier les formalismes graphiques existants. Dans cette section nous nous intéresserons à des

représentations graphiques liées à la scénarisation d'environnements virtuels et/ou augmentés ou encore utilisées dans un contexte industriel.

2.3.1 Introduction aux représentations sous forme de graphe orienté

Dans un moteur de scénario, l'utilisation d'une représentation graphique ou d'un langage graphique pour la scénarisation est pratiquement nécessaire afin de fournir une vue d'ensemble claire et structurée des agencements de séquencements souvent complexes. Une représentation claire du comportement du système permet non seulement une compréhension améliorée (Sugiyama et al., 1981) mais permet aussi d'étendre son utilité, au-delà de la simple représentation, à la manipulation pour la création et l'édition du système, que ce soit de la part des développeurs ou des experts. Cependant, un langage graphique pour la scénarisation doit répondre à plusieurs critères. Il doit permettre d'exprimer la séquentialité, la concurrence et le parallélisme. À cela nous ajoutons qu'il doit permettre la réutilisation de séquencement répétitif ou générique et donc être hiérarchique pour découper un scénario en sous-scénario.

Différents modèles de scénario vus dans la section précédente, comme #SEVEN (Claude, 2016) par exemple, propose l'utilisation de réseaux de Petri (Petri, 1962, 1966). D'autres comme LORA++ (Gerbaud et al., 2007; Mollet & Arnaldi, 2006) s'inspirent de langages industriels comme le Grafcet (David & Alla, 1992) ou utilisent une représentation sous forme de Machine à états (Cremer et al., 1995). Dans les sections suivantes, nous allons donc faire un focus sur certains des langages graphiques existants, ainsi que leurs variantes, pouvant être utilisés dans la scénarisation.

2.3.2 Machine à états

Une machine à états finis ou un automate fini sont plus communément utilisés en mathématiques ou en informatique. Ils sont utilisés pour représenter un séquencement ou un comportement. C'est un modèle constitué d'un nombre d'**états** finis et de **transitions** (Figure 43). Une machine à états peut passer d'un état à un autre en réponse à certaines entrées. Une transition correspond au passage d'un état à un autre. Une **entrée** est simplement une condition qui déclenche une transition.

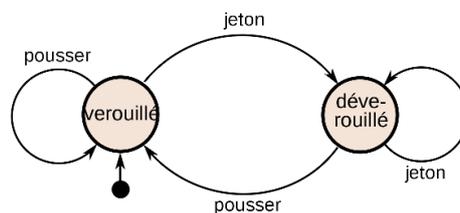


Figure 43 : Exemple simple d'un automate fini appliqué sur un portillon de métro.

Comme déjà vu dans la section 2.2.1.1.1, certains modèles comme HCSM (Cremer et al., 1995), HPTS++ (Devillers & Donikian, 2003; Lamarche & Donikian, 2002) utilisent les machines à états ou automates finis pour définir les différents enchaînements possibles d'actions au niveau du comportement des éléments d'un environnement virtuel. En effet, ces machines à états basées UML introduisent des concepts d'états hiérarchiquement imbriqués et de parallélisme. Cette représentation répond aux besoins de séquentialité, de parallélisme, de concurrence et de

hiérarchie. Leurs avantages sont qu'ils permettent de gérer plusieurs états et transitions en même temps ainsi que le retour à un état précédent.

Cependant lorsque le nombre d'actions et transitions augmente, cela peut rapidement devenir illisible. Car, même si la décomposition hiérarchique et l'utilisation de plusieurs automates en parallèles simplifient la tâche de création de scénario ou de comportement, l'effort à fournir reste important car il est nécessaire d'explicitier toutes les transitions possibles. De plus, ce type de formalisme comme les machines de Moore et de Mealy, étant orienté mathématique et informatique, sont difficiles à utiliser pour les non-initiés. En effet, la représentation graphique est peu lisible d'autant plus que des notions telles que « ET » ou « OU » ne peuvent être exprimées graphiquement.

2.3.3 Diagramme d'activité (UML)

En termes de formalisme de base concernant la représentation d'un processus, le diagramme d'activité répond aussi aux différents critères. Ce type de diagramme UML permet de représenter une suite d'évènements qui vont se déclencher en fonction des états du système. Comme pour les autres formalismes, il est très proche des diagrammes d'état-transitions et va permettre de représenter la séquentialité, la concurrence et le parallélisme.

Les composants de base d'un diagramme sont similaires aux autres formalismes (Figure 42). Un diagramme d'activités se décompose en activités. Une activité est un comportement décrit par un séquençement de nœuds et de transitions. Une transition représente simplement le passage d'une activité à une autre. Il est également possible de regrouper les activités en groupe. Ensuite, concernant les nœuds, il en existe 3 types : les nœuds d'exécutions, les nœuds de contrôle et les nœuds d'objets.



Figure 44 : Représentation graphique des nœuds d'activité

- Les nœuds d'exécutions sont des nœuds que l'on peut exécuter. Le principal nœud étant les nœuds d'action avec une action liée. Une action peut être, par exemple, l'affectation ou l'accès à d'une valeur (set/get), un calcul, l'émission / la réception d'un signal etc.
- Les nœuds de contrôle permettent eux de contrôler le flux d'activités. Il en existe plusieurs comme les nœuds initiaux ou de fin d'activités, les nœuds de fusion / bifurcation que nous pouvons retrouver dans les langages Graftet et Grafchart. Il dispose également d'un nœud de décision pour représenter une condition booléenne, suivie de deux connecteurs respectivement OUI ou NON.
- Les nœuds d'objet permettent de représenter des objets qui vont pouvoir être impactés par une action de par leurs entrées / sorties.

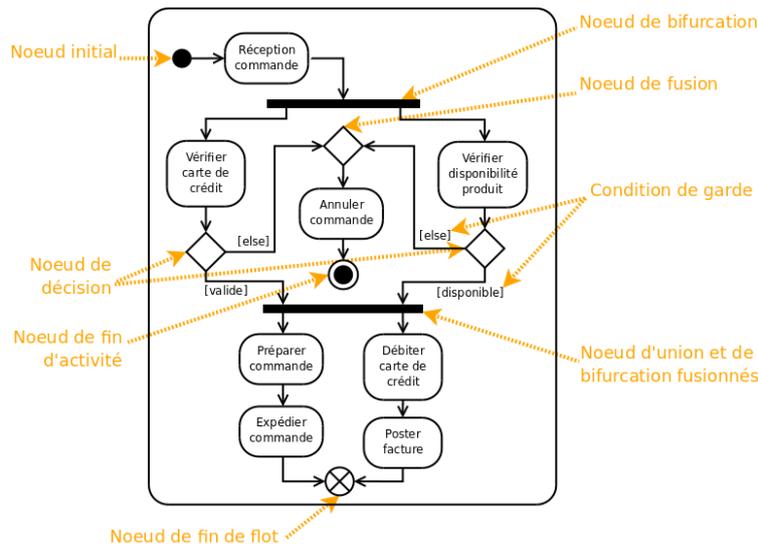


Figure 45 : Exemple de diagramme d'activité¹.

L'avantage de ce formalisme est qu'il est complet et répond aux critères pour représenter un scénario industriel. Il a été utilisé dans des frameworks comme MASCARET avec HAVE (Chevaillier et al., 2012) (voir 2.2.1.2.4) ou POSEIDON (Marion, 2010).

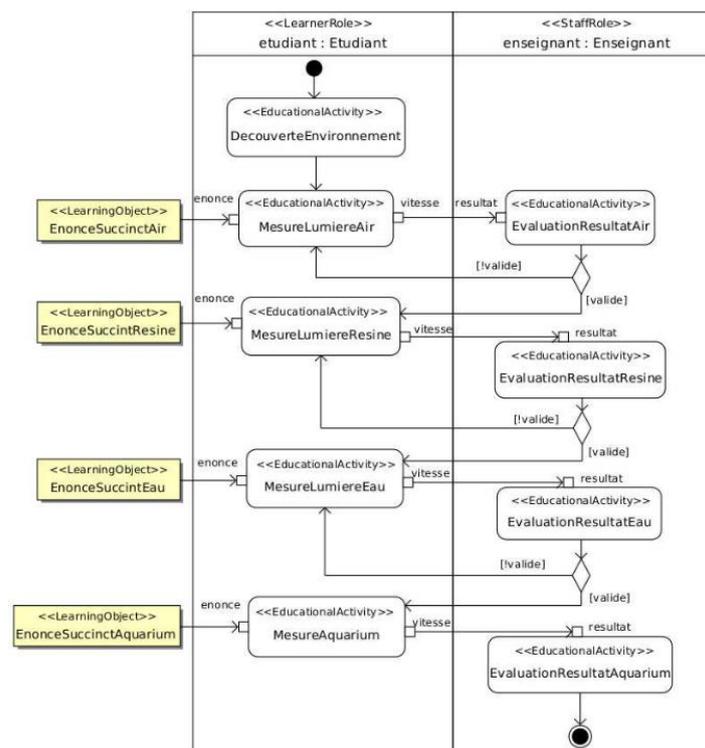


Figure 46 : Exemple d'activité HAVE (Chevaillier et al., 2012)

2.3.4 Les réseaux de Petri

Depuis la parution des réseaux de Petri en 1962 (Petri, 1962), plusieurs variantes ont vu le jour, comme les réseaux de Petri dit « Haut niveau » et les réseaux de Petri dit « Interprétés »

¹ <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>

que nous détaillerons par la suite. Cette représentation a déjà été utilisée dans la scénarisation d'environnement virtuel comme dans #SEVEN (Claude, 2016) ou dans (Brom et al., 2007; Brom & Abonyi, 2006). Ce sont des graphes orientés composés de **Place** et de **Transitions**. Les deux éléments sont reliés entre eux par des arcs, représentés par des flèches. Un arc a un poids (un entier numérique), conditionnant alors le passage de l'un à l'autre. Ainsi, chaque place a des transitions et chaque transition a des places.

Une place peut être en amont ou en aval d'une ou plusieurs transitions. Elle peut contenir plusieurs **jetons** sauf exception dans le cas de réseaux de Petri dit « bornés » (où le nombre de jetons est fini) ou bien « sauf » lorsque le nombre de jetons maximum par places est de 1 (dans ce cas précis, les réseaux de Petri saufs sont équivalents à des machines à états finis). Dans le cas de réseau de Petri « Haut niveau », le jeton peut contenir des informations. Une place peut être initiale ou finale (voir Figure 47).

Ensuite, une transition correspond à un évènement ayant lieu dans le système. Elle est « sensibilisée » lorsque toutes les places en amont possèdent un nombre de jetons au minimum égal au poids de l'arc. Si l'arc a un poids de 2 alors 2 jetons seront nécessaires pour la sensibiliser (voir Figure 47 – Place 1). Si une place est sensibilisée alors une transition peut être déclenchée. Si telle est le cas, elle consomme les jetons des places en amont et produit un ou des jetons dans les places en aval. Le réseau de Petri devient « interprété » avec l'ajout de prédicat (condition) aux transitions.

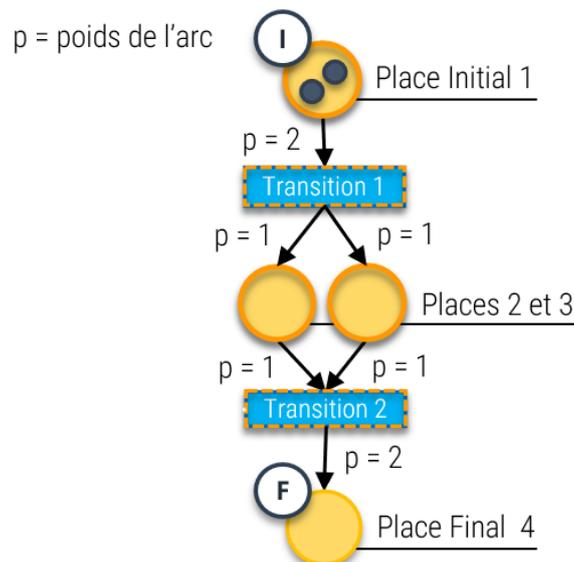


Figure 47 : Exemple d'un réseau de Petri simple

Les places représentent les états du système, et une transition le passage d'une place à une autre. Une place peut également renvoyer vers un autre sous-réseau permettant une hiérarchisation des différentes séquences. Les composants de base des réseaux de Petri sont donc simples. La séquentialité, la concurrence et le parallélisme nécessaire pour pouvoir représenter un scénario industriel peuvent être exprimés de par la structure du réseau de Petri. Prenons l'exemple de quatre vis A, B, C et D ne pouvant être vissées qu'en diagonale. Le réseau se représenterait ainsi (voir Figure 48). Des jetons sont présents dans les places initiales P1 et

P2, si l'opérateur commence par la vis A, la condition de la transition T1 est validée, les jetons des places en amont P1 et P2 sont donc consommés et un nouveau jeton est créé de P3. La prochaine sera obligatoirement la vis C. Si l'opérateur tente de visser la vis B par exemple, le scénario n'avancera simplement pas car aucun jeton n'aurait été recréé dans une des places en amont (voir Figure 48 - P1).

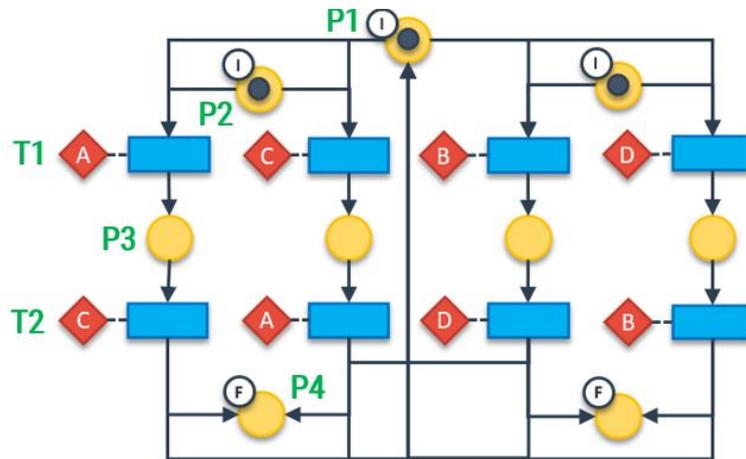


Figure 48 : Exemple de séquençage en réseau de Petri Sauf

Nous pouvons, en effet, entrevoir un problème si l'on veut concevoir un scénario prenant en compte les erreurs. Dans un tel cas, chaque erreur possible devra être prise en compte dès la création du scénario initial.

2.3.5 Grafcet

Le Grafcet (David & Alla, 1992) (GRAPhe Fonctionnel de Commande par Etapes et Transitions) est un outil graphique permettant de décrire les différents comportements de l'évolution d'un système. C'est un langage principalement utilisé dans l'industrie et dans l'automatisme.

Il comprend des **étapes** associées à des actions, des **transitions** associées à des **réceptivités** et des **liaisons orientées** reliant étapes et transitions.

Une étape correspond à un état ou une partie de l'état du système. Il a deux états possibles : **actif**, représenté par un jeton dans l'étape ou **inactif**. Chaque étape est repérée numériquement et possède une variable booléenne valant **1** si l'étape est active et **0** sinon. L'état du système initial est indiqué pour une étape initiale représentée par un carré double (voir Figure 49).

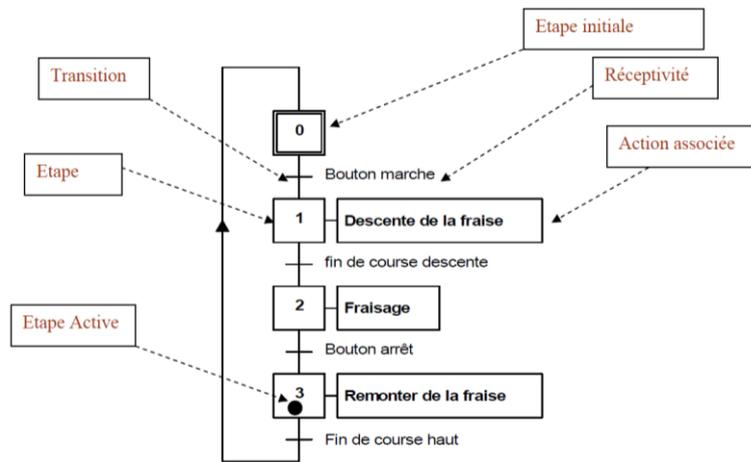


Figure 49 : Exemple de grafcet dans l'industrie

À chaque étape sont associée une ou plusieurs action(s) mais il est possible d'avoir des étapes sans action, appelé étape « vide ». L'action va faire référence à un ordre ou alors à d'autres grafcets. Une même action peut être associée à plusieurs étapes. Une action peut être de 3 types :

- Continue, l'ordre est émis de façon continue tant que l'étape est active.
- Conditionnelle, l'ordre est émis que si l'étape est active et si la condition associée est vraie.
- Ou mémorisée. L'ordre est émis et est maintenu sur l'activation de plusieurs étapes consécutives.

Une transition correspond aussi au passage entre deux étapes. À chaque transition est associée une condition logique appelée réceptivité. C'est la condition nécessaire pour passer d'une étape à une autre.

Pour le séquençage, Grafcet intègre différentes notions comme le saut d'étape ou la reprise d'une étape (par exemple pour effectuer une boucle si une condition n'est pas obtenue). On retrouve également les mêmes concepts de divergence et convergence en « ET » et « OU » (voir Figure 50) présentés dans les réseaux de Petri (voir 2.3.4).

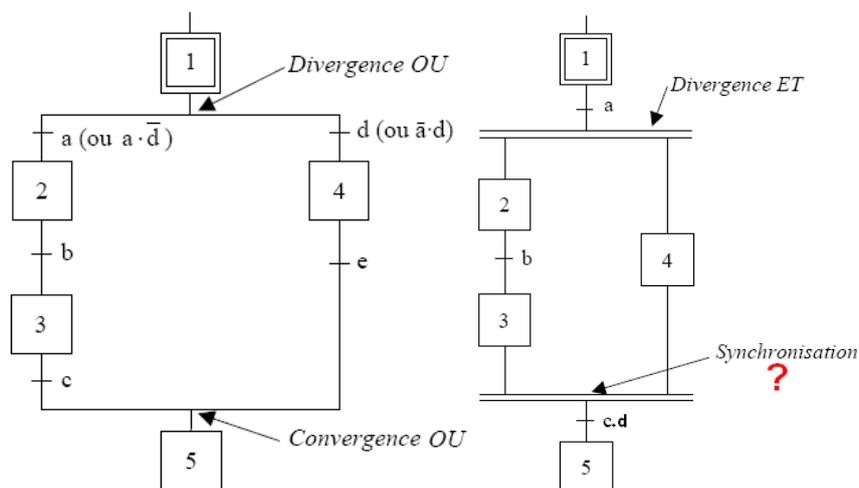


Figure 50 : Exemple en Grafcet de divergence et convergence « OU » à gauche, et d'une divergence / convergence « ET » à droite.

Ainsi, Grafcet répond aux besoins présentés en introduction en termes de séquentialité, de concurrence et de parallélisme. Il permet aussi une structure hiérarchique lorsqu'une action fait référence à un autre grafcet. De plus, le formalisme Grafcet, principalement dédiés à l'automatisme, permet d'éditer des procédures séquentielles au moyen de concepts avec lesquels le monde industriel est habitué.

2.3.6 Grafchart

Selon (Johnsson & Årzén, 1998), il y a deux versions différentes de Grafchart. Une version originale qui est principalement basée sur Grafcet et la 2^{ème}, appelée « High-Level Grafchart », qui inclut les principes des réseaux de Petri haut niveau. De plus Grafchart est un langage graphique s'adaptant facilement au contexte industriel car utilisé dans ce secteur.

La syntaxe de Grafchart est en effet similaire à Grafcet et se constitue principalement d'**étapes** et de **transitions**. Les **étapes** représentent les états et les **transitions** représentent les conditions à remplir pour passer d'une étape à une autre. Le langage Grafchart supporte les chemins alternatifs, exprimant la concurrence, et les chemins parallèles, le parallélisme (Figure 51). De la même manière que pour Grafcet et les réseaux de Petri, une étape est « active » lorsqu'elle contient un jeton et une transition est dite « activées » lorsque toutes les étapes qui la précèdent sont actives. Dans la version ordinaire, les jetons sont seulement des indicateurs booléens tandis que dans la version haut niveau, comme pour la version haut niveau des réseaux de Petri, les jetons peuvent également contenir de l'information via des attributs.

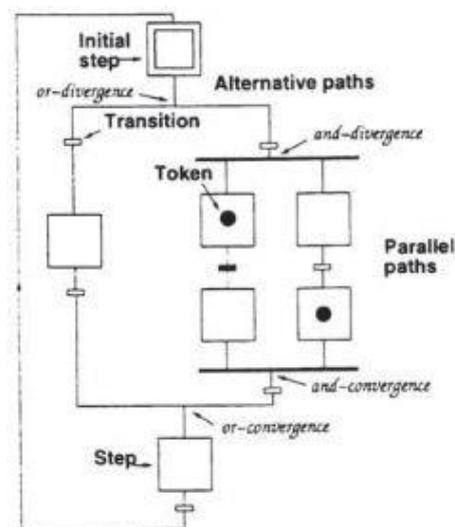


Figure 51 : Extrait de syntaxe graphique de Grafchart (Johnsson & Årzén, 1998)

La syntaxe de Grafchart est constituée de plusieurs types d'éléments (Figure 52) : les étapes, les transitions, les processus, les procédures. La version haut niveau intègre également les notions de macro-étapes, d'étapes de procédure, d'étapes de processus et de transitions d'exceptions. Chaque élément peut être associé à un sous-espace de travail (« subworkspace »), hiérarchisant ainsi le séquençement.

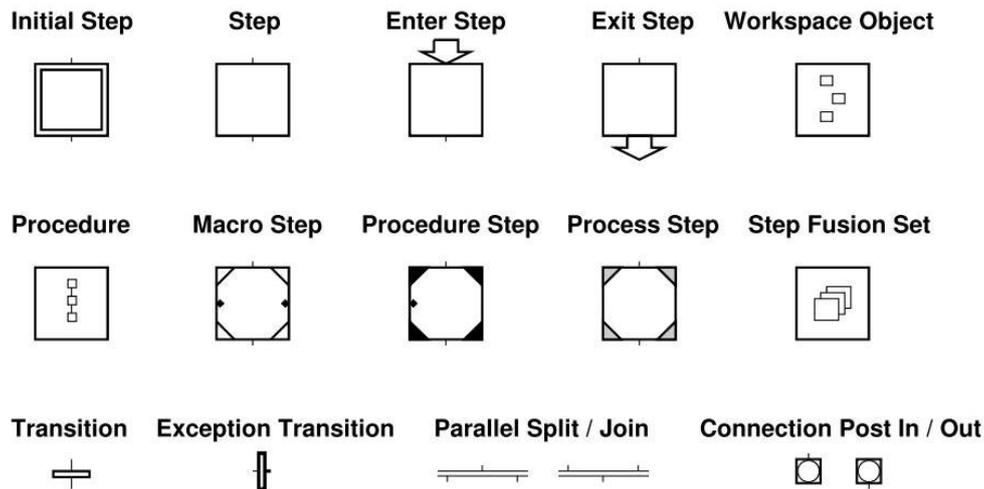


Figure 52 : Les principaux éléments constituant la syntaxe de Grafchart (Theorin, 2013)

Comme expliqué, une étape représente un état du système. Elle regroupe notamment des actions qui sont déclenchées quand celle-ci est active (Figure 53). Ces actions sont similaires à celles présentées dans les réseaux de Petri interprétés (Moalla, 1985) bien qu'elles ne se limitent qu'à un booléen. Dans Grafchart, ces actions peuvent être conditionnelles ou inconditionnelles et quatre types d'actions sont supportés :

- Permanente (always actions) : Les actions sont exécutées de manière cyclique tant que l'étape est active.
- Initiale (initially actions) : Les actions sont exécutées lorsque l'étape est activée.
- Finale (finally actions) : Les actions sont exécutées lorsque l'étape est désactivée.
- Interruptive (abortive action) : Les actions sont exécutées juste avant que l'étape devienne interrompue lors du déclenchement d'une transition d'exception.

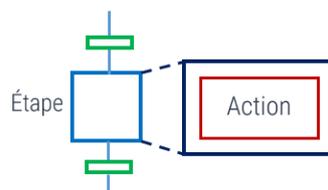


Figure 53 : Une étape Grafchart disposant d'une action

Les transitions sont utilisées pour passer d'un état à un autre. Chaque transition est associée à une ou plusieurs « réceptivité(s) » avec un évènement et une condition. Lorsque la réceptivité est vérifiée et que la condition est vraie, la transition est déclenchée. Les étapes précédentes sont alors désactivées et les suivantes activées.

Pour une représentation plus claire et pour hiérarchiser le séquençement, Grafchart permet d'encapsuler un séquençement. Soit de manière ponctuelle avec l'utilisation de **processus**, soit pour permettre une réutilisation externe et répétée avec l'utilisation de **procédures**. En effet, un processus permet d'encapsuler un séquençement d'étapes et de transitions décrivant une fonction dans son sous-espace de travail. La fonction contenue doit obligatoirement commencer par une étape initiale. Une procédure est un séquençement, contenu dans un sous-espace, qui

peut être exécuté à plusieurs endroits. Elle commence toujours avec une étape d'entrée et finit avec une étape de sortie (voir Figure 52). Elle permet notamment les appels récursifs ou externes.

Le Grafchart haut niveau vient donc compléter le Grafchart dit « ordinaire » avec les notions de **macro-étapes**, **d'étapes de procédure**, **d'étapes de processus** et de **transitions d'exceptions**, détaillées dans les paragraphes suivants. Les avantages de cette version haut niveau dans le cadre d'un usage industriel sont démontrés par (Johnsson & Årzén, 1996) pour la création d'un système de gestion de contrôle de lots. L'auteur montre, via leur cas d'usage, qu'utiliser les jetons et un graphique multidimensionnel comme Grafchart augmente les possibilités de construction. En effet, dans la version haut niveau, de même que pour les réseaux de Petri haut niveau, les jetons peuvent transporter de l'information et possèdent des attributs.

Les macro-étapes : Elles représentent des étapes qui ont une structure interne dans leur sous-espace de travail. Cette structure interne doit forcément commencer par des états d'entrées et de sorties. Comme les étapes, elles peuvent disposer d'actions.

Les étapes de procédure : Elles représentent l'appel d'une procédure et possèdent comme attribut le nom de la procédure à appeler.

Les étapes de processus : Elles permettent de lancer une procédure dans un thread séparé de la même manière qu'un processus. La transition suivant l'étape de processus est activée dès que l'exécution de la procédure a débuté. Un jeton blanc est montré tant que le processus s'exécute.

Les transitions d'exception : Les transitions d'exception peuvent uniquement être connectées à des macro-étapes ou des étapes de procédure. Elle est activée tant que l'étape est active. Si la transition est validée, alors l'étape qui était en cours est annulée et les actions abortives, s'il y en a, sont exécutées. Les étapes retiennent leurs états d'exécution. Il est alors possible de les relancer à l'endroit où elles s'étaient arrêtées. Cette variante a été proposée pour la première fois par (Årzén, 1991) et permet d'intégrer une gestion des erreurs dans la scénarisation.

De plus, les transitions peuvent avoir plusieurs réceptivités ce qui permet une description très compacte du système contrairement aux réseaux de Petri qui requièrent une transition pour chaque réceptivité.

Ainsi, comme pour le Grafcet, le Grafchart répond aux besoins en termes de séquentialité, de concurrence et de parallélisme. Il permet aussi une structuration hiérarchique. De plus, le Grafchart, principalement dédié au secteur industriel, utilise un langage avec lesquels le monde industriel est habitué. Cependant, nous pouvons supposer que la syntaxe Grafchart nécessite un temps d'apprentissage pour des personnes n'ayant pas cette connaissance industrielle, notamment dû faite du nombre de blocs de construction possible.

2.3.7 CTT

En plus des formalismes de base vue ci-dessus, il est intéressant de noter qu'il existe une multitude d'autres représentations basées graphe pour représenter un scénario. La notation CTT (ConcurTaskTrees) (Paternò & Santoro, 2003) est un formalisme graphique qui s'interface avec des modèles de domaine représentés en UML. Pour représenter le séquençement, il organise les

tâches de manière hiérarchique, les tâches d'un même niveau étant ordonnées grâce à des opérateurs exprimant la séquentialité ou encore l'alternative. Dans la notation de base, il existe 6 opérateurs différents :

- Le choix entre deux tâches ($T1 \text{ [] } T2$) ;
- La concurrence, c'est-à-dire que les tâches peuvent s'exécuter en parallèle ($T1 \text{ ||| } T2$) ;
- L'indépendance d'ordre où l'ordre d'exécution des tâches n'a pas d'importance ($T1 \text{ |=} T2$) ;
- La désactivation / activation, qui a pour effet de stopper ($T1 \text{ [> } T2$) ou d'exécuter une tâche ($T1 \text{ >> } T2$) ;
- La suspension / reprise permet de suspendre ou reprendre l'exécution d'une tâche ($T1 \text{ [> } T2$) ;
- L'itération lorsqu'une tâche peut s'exécuter en boucle.

Bien que CTT ait été conçu à l'origine pour servir de support à la conception d'interfaces hommes-machines, il a été utilisé dans EMSAVE (Figure 54) (Buttussi et al., 2013; Vidani & Chittaro, 2009) pour représenter les procédures dans un environnement virtuel pour l'apprentissage humain, dans une optique de suivi de l'apprenant et de génération dynamique de menus. Bien que la logique de ce formalisme soit minimaliste mais puissante, elle reste moins connue des industriels par rapport à d'autre langage. Par ailleurs, sa représentation hiérarchique pour définir une procédure semble peu simple à utiliser pour un non-initié.

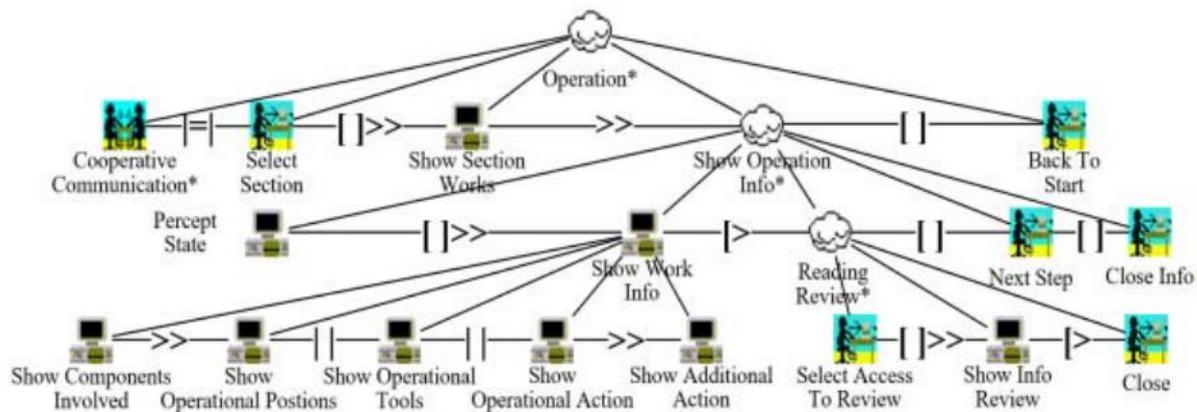


Figure 54 : Exemple de représentation CTT (Buttussi et al., 2013; Vidani & Chittaro, 2009)

2.3.8 AND/OR Graphs

Le formalisme AND/OR Graphs a été conçu, à l'origine, pour la planification d'assemblage industriel (De Mello & Sanderson, 1990). Ce formalisme sous forme de NodeGraph (Graphe de nœud) a une forte capacité à représenter l'exécution parallèle des actions d'assemblage (AND) ainsi que l'indépendance temporelle des actions parallèle, c'est-à-dire que l'ordre n'a pas d'importance (OR). La figure 55 montre par exemple un scénario d'assemblage à respecter entre 4 éléments A, B, C et D.

Cette représentation est intéressante à étudier car très efficace lorsqu'il s'agit d'assemblage, d'autant plus que son principe est très simple à comprendre du fait de n'utiliser que des nœuds AND ou OR. Nous retrouvons alors la séquentialité, le parallélisme et la concurrence. Cependant

son usage reste restreint à l'assemblage et peut rapidement trouver ses limites dans un autre contexte.

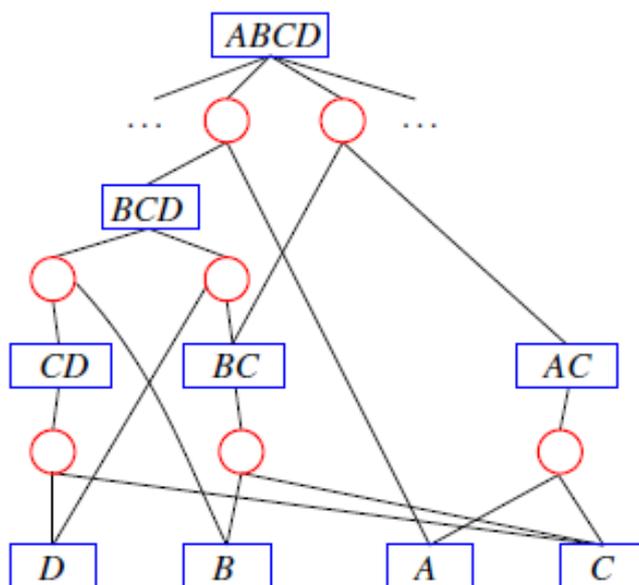


Figure 55 : Exemple de AND/OR Graphs (Johannsmeier & Haddadin, 2017) avec les rectangles bleus étant les OR et les cercles rouges les AND.

2.3.9 Représentations propriétaires

Il est intéressant aussi d'aller étudier les solutions logicielles existantes permettant la création de scénario. Les informations concernant les formalismes et les architectures qu'ils utilisent étant peu accessibles, nous ne pouvons pas nous baser que la visualisation du résultat. C'est pourquoi, nous avons catégorisé ces outils comme ayant une représentation propriétaire.

Par exemple, nous pouvons citer la solution WEAVR Creator² proposé par Unity³ (Figure 56). WEAVR Creator est l'un des modules de leur plateforme Pacelab WEAVR qui permet de créer des programmes de formation en réalité étendue (RV/RA). Il s'agit d'un outil basé sur des nœuds, eux-mêmes composé d'étapes, et de connexions les reliant permettant de créer des contenus de formation sans codage.

Une autre solution similaire est proposée par Light & Shadow (voir Figure 57) avec un plugin Unity appelé « Interact ». Interact permet de créer des simulations industrielles et centrées sur l'homme en RV. Sur le même principe que WEAVR Creator, l'éditeur de scénario proposé au sein du plugin est basé sur des nœuds représentant des étapes et relier entre eux permet également de créer un scénario sans passer par du développement informatique.

² <https://unity.com/fr/products/weavr>

³ <https://unity.com/>

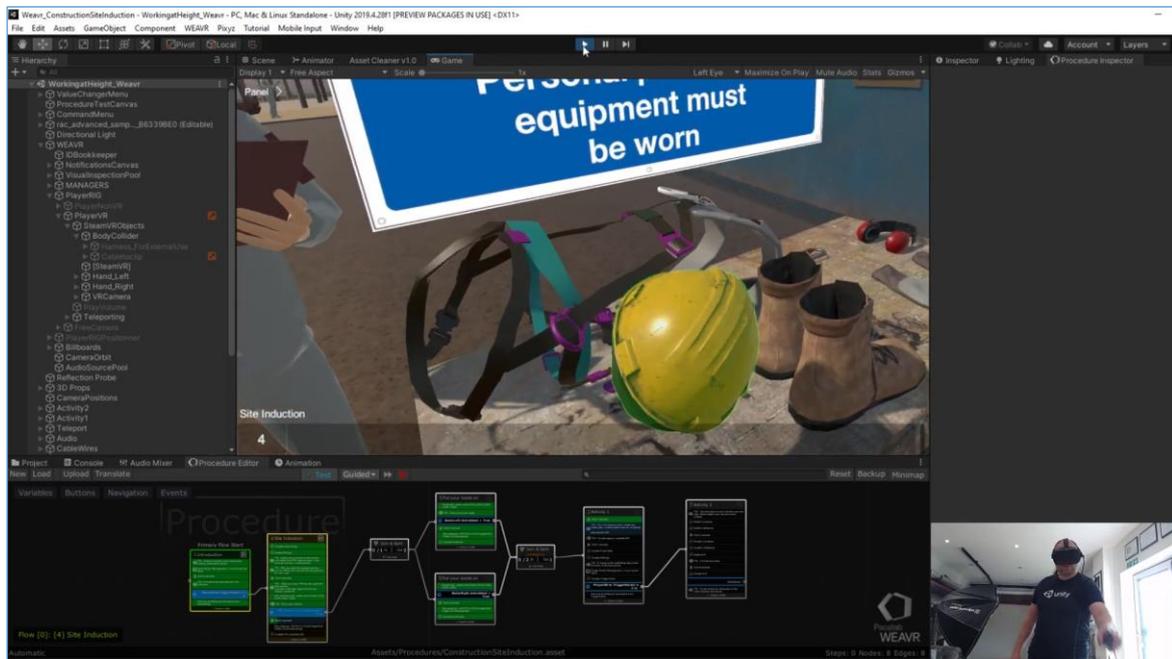


Figure 56 : Interface WEAVR dans Unity3D

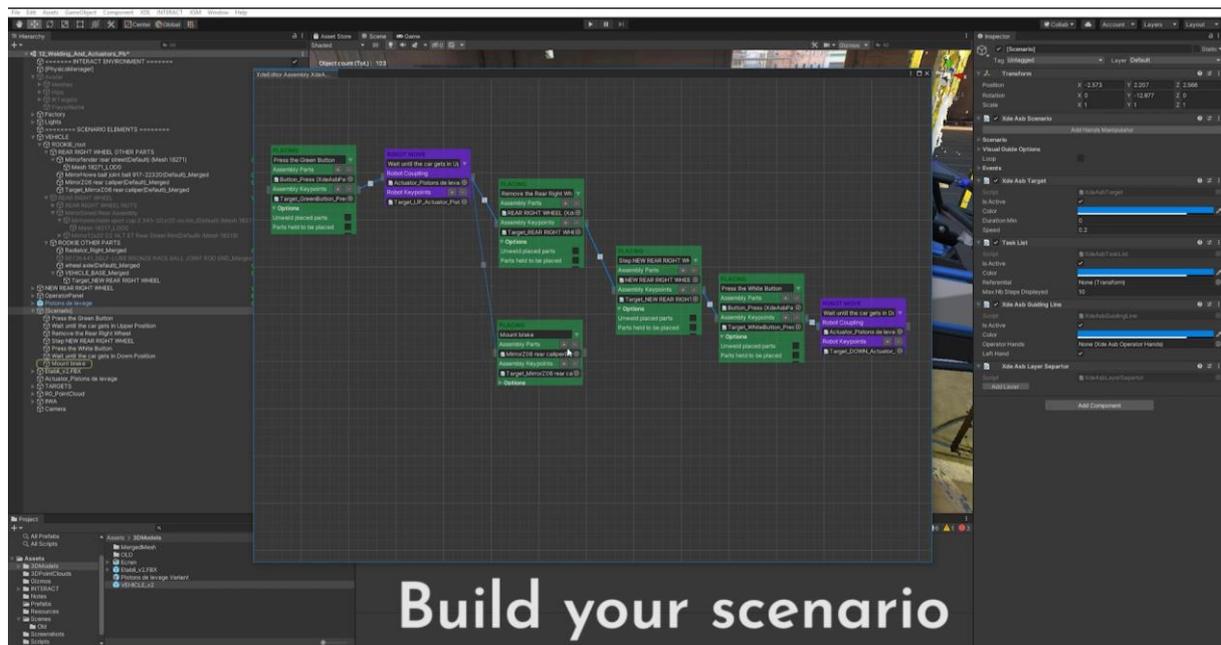


Figure 57 : Interface de l'éditeur de scénario d'Interact dans Unity développé par Light & Shadow⁴.

Plus spécifique à la réalité augmentée, la compagnie Diota a développé un éditeur « DiotaEditor⁵ » pour développer des instructions de travail étape par étape et les organiser (Figure 58). Il peut s'agir d'instruction de montage dans le cas d'un assemblage ou d'une gamme de maintenance ou d'instructions de contrôle sur une sélection de pièces dans le cas d'un contrôle qualité.

⁴ <https://www.ls-group.fr/interact>

⁵ <https://www.diota.com/solutions-products/our-products/diotaeditor/>

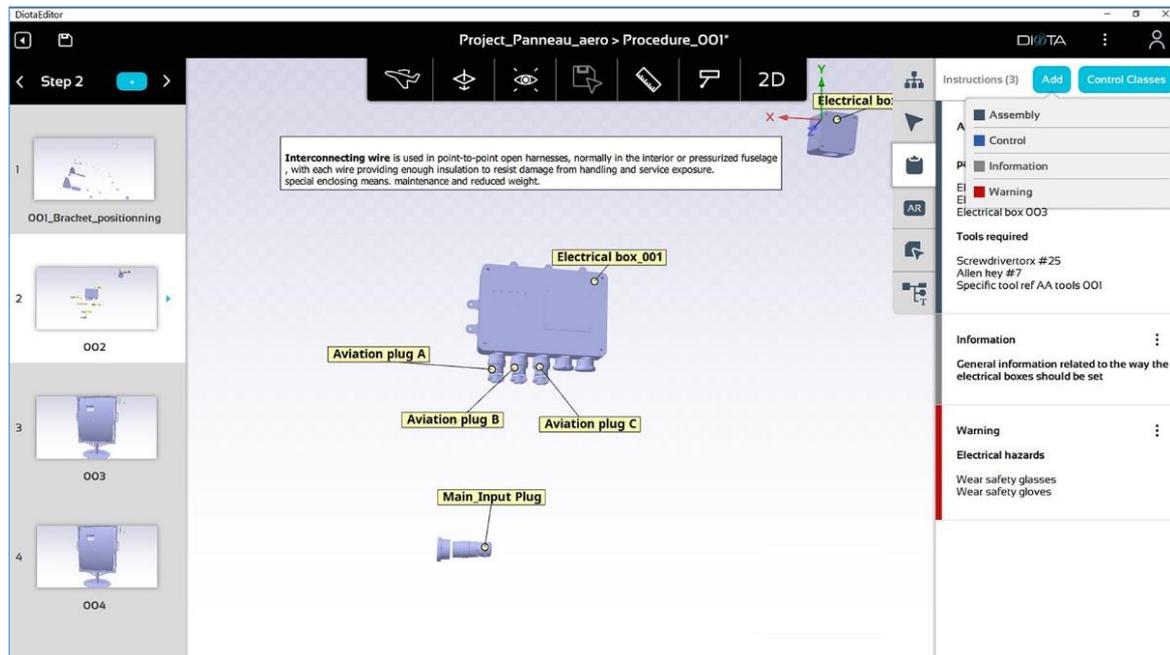


Figure 58 : Interface de création de procédure en réalité augmentée avec DiotaEditor

2.3.10 Synthèse

Dans cette section, nous avons étudié différentes représentations graphiques pouvant être utilisées pour la scénarisation d'environnement virtuel ou augmenté. Nous avons notamment présenté les réseaux de Petri, réseau de Petri de haut niveau (utilisé par #SEVEN (Claude, 2016)), Grafcet et pour finir Grafchart. Le Grafchart se basant sur le grafcet et les réseaux de Petri ainsi que sur des éléments de la programmation orientée objets, les différences et similitudes ont été soulignées. De plus, du fait de son orientation pour le développement, nous pouvons noter que Grafchart comprend plus de blocs de construction que les autres modèles permettant ainsi une représentation plus modulable.

Afin de mieux se rendre compte de l'importance de bien définir et formaliser la représentation du séquençage, le tableau suivant (Tableau 2) liste différents modèles de scénario et solutions techniques regroupés par formalisme de base (Petri, Grafcet, Machine état, ...) dont le formalisme utilisé par le modèle est inspiré. Parmi les critères, nous avons retenu : le contexte d'usages du formalisme : Générique (G), Industriel (I), Formation (F), Narration (N) ; Le type d'agencement que celui-ci permet d'exprimer : Séquentialité, Parallélisme, Concurrence, Hiérarchie. Nous indiquons également, de manière à identifier les langages plus complets que d'autres en termes de blocs de construction et d'éléments de langage, si celui-ci à un niveau normal (+) ou avancé (++). De plus, les formalismes étant utilisés dans un contexte de scénarisation, nous avons précisé s'ils prennent en compte les ressources disponibles dans l'environnement ? S'ils permettent de représenter plusieurs types d'agents (réel ou autonome) et en collaboration ? Pour finir, nous précisons si une interface / un logiciel d'édition de scénario en lien avec la solution existe.

Ce que nous pouvons observer c'est qu'il existe une multitude de formalismes et de variantes dont les modèles s'inspirent. Nous retrouvons les réseaux de Petri, Grafcet et Grafchart, détaillés précédemment, ainsi que les machines à états (Cremer et al., 1995), les diagrammes d'activité

(Chevaillier et al., 2012), ou encore les formalismes HTC (Mohseni-Kabir et al., 2015), CC-HTC (Hayes & Scassellati, 2016), AND/OR Graphs (Johannsmeier & Haddadin, 2017), CTT (Buttussi et al., 2013; Vidani & Chittaro, 2009), Tous remplissent leur rôle en ce qui concerne la représentation de la séquentialité mais peu vont plus loin dans les possibilités d'agencement comme la possibilité de mettre des tâches en parallèle ou en concurrence. Il va de même pour la notion de « sous-scénario » avec la hiérarchisation. Ce sont les formalismes utilisés dans un usage RA/RV qui montre un niveau de langages plus complet. Cependant, la gestion des ressources de l'environnement, l'intégration de différents types d'agents (réel ou autonome) ainsi que la collaboration sur un même scénario n'est pas systématique.

Tableau 2 : Synthèse des représentations graphiques du séquençage et leurs variantes utilisées par les modèles de scénario. « ? » indique que les données n'ont pas pu être vérifiées dû à un souci d'accessibilité.
Colonnes Usages : Générique (G), Industriel (I), Formation (F), Narration (N)

Solution	Formalisme de base	Usages	Appareils ciblés	Agencement				Niveau du langage	Gestion des assets ?	Multi-agent ?	Collaboration ?	Interface(s) édition existante(s)
				Séquentialité	Parallélisme	Concurrence	Hierarchique					
Réseau de Petri (Petri, 1962, 1966; J. Wang et al., 2017)	Réseau de Petri	G	N/A	X	X	X	X	+				PIPE2 , Yasper
#SEVEN (Claude, 2016; Lécuyer et al., 2020; Terrier et al., 2020)		G	RV	X	X	X	X	++	X	X	X	#SEVEN Scenario editing tool with Unity + éditeur #SEVEN
IVE (Brom et al., 2007; Brom & Abonyi, 2006)		N	RV	X	X	X		+				Aucun trouvé
Grafcet (David & Alla, 1992)	Grafcet	I	N/A	X	X	X	X	++				GoJS , Grafcet-Studio
LORA++ (Gerbaud et al., 2007; Mollet & Arnaldi, 2006) avec Generic Virtual Trainer (OpenMask) (Gerbaud, 2008; Gerbaud & Arnaldi, 2009)		F	RV	X	X	X		++	X			Aucun trouvé
Grafchart (Johnsson & Årzén, 1998, 1999; Theorin, 2013)	Grafchart	I	N/A	X	X	X	X	++				JGrafchart
Machine états	Machine états	G	N/A	X	X	X	X	+				Draw.io
HCSM (Cremer et al., 1995)		G	RV	X	X	X	X	+				Aucun trouvé
HPTS++ (Devillers & Donikian, 2003; Lamarche & Donikian, 2002)		G	RV	X	X	X	X	+		X		Aucun trouvé
Story Nets (Hill et al., 2003; Rickel et al., 2002; Swartout et al., 2004, 2006; Traum et al., 2003)		F	RV	X		X		+		X	X	Aucun trouvé
ABL (Mateas & Stern, 2002)		N	RV	X		X		+				Aucun trouvé
Amazon Sumerian		N / F	RV	X				+	X			Amazon Sumerian
DRAKON		G	RV	X	X	X		+				DRAKON
Diagramme d'activité		D	G	N/A	X	X	X		+			

Solution	Formalisme de base	Usages	Appareils ciblés	Agencement				Niveau du langage	Gestion des assets ?	Multi-agent ?	Collaboration ?	Interface(s) édition existante(s)
				Séquentialité	Parallélisme	Concurrence	Hierarchique					
HAVE (Chevaillier et al., 2012)		F	RV	X	X	X		+	X	X	X	Suite logicielle
IMS-LD (Lejeune, 2004)		F	RV	X	X	X		+		X	X	Aucun trouvé
YALTA (Burkhardt et al., 2009) (Support HPTS++)		F	RV	X	X	X	X	+		X		Aucun trouvé
POSEIDON (MASCARET) (Marion, 2010)		P	RV	X	X	X	?	+		X		Aucun trouvé
WEAVR Creator	Propriétaire	F / N	RV	X	X	?	?	++	X			WEAVR (Video)
VISC		G	RV	X	?	?	?	++				VISC
XR Twin		G	RV	?	?	?	?	++	X			XR Twin
DIOTA Editor		I	RA	X	?	?	?	++	X			DiotEditor
HTN (Mohseni-Kabir et al., 2015)	HTN	G	N/A	X				X	+		X	Une interface graphique d'édition pour les professeurs
Clique/Chain HTN (Hayes & Scassellati, 2016)	CC-HTN	G	N/A	X					+		X	Aucun trouvé
AND/OR graphs (Johannsmeier & Haddadin, 2017)	AND/OR graphs	G	N/A	X	X	X		+				Aucun trouvé
FORMID (Echterbille, 2014; Lejeune, 2011; Létondor, 2016)	Suite d' étapes	F	N/A	X					+	X	X	FORMID (Suite logicielle)
U-Director (Mott & Lester, 2006) (Réseaux bayésiens) (Amokrane, 2010)	Réseaux bayésiens	N	RA / RV	X		X			+			Aucun trouvé
Mimesis (M. Riedl et al., 2003; M. O. Riedl & Young, 2010)	Plans partiellement ordonnés	G	N/A	X					+			Aucun trouvé
EMSAVE (Buttussi et al., 2013; Vidani & Chittaro, 2009)	Formalisme CTT	F	RV	X	X	X			+			Aucun trouvé
(Gong et al., 2019)		G	N/A	X	X	X			+			Aucun trouvé
(Aehnel & Bader, 2018; Bader & Aehnel, 2014)		G	N/A	X	X	X			+			Aucun trouvé
PAPOUS (Silva et al., 2003)	Graph	N	N/A	X		X			+			Interface de création d'histoire

Solution	Formalisme de base	Usages	Appareils ciblés	Agencement				Niveau du langage	Gestion des assets ?	Multi-agent ?	Collaboration ?	Interface(s) édition existante(s)
				Séquentialité	Parallélisme	Concurrence	Hierarchique					
ICT Leaders (Gordon et al., 2004)		N	N/A	X		X		+				Aucun trouvé

2.4 Problématique scientifique

Pour conclure ce chapitre, nous avons, dans cette partie, passé en revue plusieurs modèles d'environnements, de scénarios et des frameworks permettant de créer et d'éditer des contenus en réalité virtuelle et/ou en réalité augmentée. Nous avons ensuite étudié plusieurs langages graphiques pouvant représenter le séquençement de scénario. Nous pouvons faire ressortir plusieurs axes sur lequel orienter notre proposition de modèle.

Nous constatons, tout d'abord, un manque de modèles permettant une création de contenu unifiée pour des usages en RV et en RA dans un contexte industriel. La raison principale étant leurs usages spécifiques très différents : principalement orientés formation pour la RV ou assistances pour la RA. Pourtant, ces deux cas d'usages peuvent utiliser un même moteur de scénario ainsi que des données et connaissances métiers communes. La différence étant qu'en réalité virtuelle, un scénario sera plus évolué et ouvert tandis qu'en réalité augmentée, un scénario sera majoritairement procédural (une étape après l'autre, comme une gamme de maintenance par exemple) et nécessitera donc une interprétation de la part du lecteur de scénario. Une des premières contributions du modèle sera donc de permettre une création commune de contenus RV/RA définissant les interactions possibles. Pour cela l'approche des modèles « Objet-Relation » nous semble la plus pertinente car, en plus d'avoir les avantages des autres approches, elle est très modulable et plus facilement interprétable par un expert n'ayant pas de connaissance en développement. Nous allons donc prendre cette direction dans la proposition de notre modèle.

De plus, nous constatons que la majorité des modèles dispose d'un moteur de scénario permettant au minimum la définition d'objectifs de manière procédurale (une étape après l'autre). Ils sont cependant moins nombreux à laisser la possibilité à l'utilisateur de réaliser un scénario complexe en autonomie en lui laissant la possibilité de commettre des erreurs.

En outre, notre analyse nous a permis de lister les caractéristiques à prendre en compte lors de la conception du modèle. Pour rappel, il devra :

- Permettre de créer un environnement virtuel et/ou augmentée (interactions réalisables, rôles, conséquences des interactions dans l'environnement ...), indépendamment du dispositif d'affichage,
- Permettre de définir des objectifs d'apprentissage et des scénarios avec une orchestration d'objectifs.

- Rester indépendant du nombre d'agents, qu'ils soient contrôlés par une personne réelle ou simulée,
- Permettre l'extensibilité et l'évolutivité du modèle.

Il serait également intéressant d'intégrer à notre modèle la définition de rôles aux agents (virtuels ou réels), de préciser des conséquences aux interactions (principe de causalité) et d'intégrer les notions de suivi de l'apprenant.

Pour finir, nous pouvons relever des limitations pour la représentation du séquençement. La première limitation étant que nous trouvons soit une représentation légère et claire mais qui peut être vite limitée et restreindre la mise en place d'un scénario complexe. La seconde limitation étant l'usage de plusieurs méta-modèles et langages différents à apprendre pour pouvoir prendre en main l'outil de scénarisation. Pour nous, cela peut freiner la facilité de prise en main du framework. C'est pourquoi, nous nous orientons vers un langage unifié et disposant de plusieurs éléments de construction à la manière de Grafchart, permettant ainsi d'optimiser la représentation d'un scénario complexe tout en conservant suffisamment de clarté.

Nous souhaitons finalement que le framework proposé fasse abstraction de la technologie utilisée que ce soit RV/RA ou PC Clavier/Souris et du nombre d'agents. D'autant plus que le modèle doit garantir son extensibilité et son évolutivité pour l'ajout de futurs modules. Cette proposition est détaillée dans le chapitre 3 qui suit. Nous verrons ensuite plusieurs mises en application dans le chapitre 4. Enfin dans le chapitre 5, nous présenterons l'approche d'éditeur 2D basé graphe et sur l'authoring-by-doing (ou édition par démonstration) permettant l'édition de scénario par un expert métier. Ce chapitre présentera de plus une expérimentation de ces éditeurs.

3 INTERVALES : Proposition du modèle

3.1 Introduction

L'approche proposée est basée sur les concepts d'« Entity-Feature » similaire aux modèles « Objet-Relation ». Elle a pour but de répondre aux besoins des usages de l'industrie 4.0, présentée en partie 1.1, en permettant par exemple la définition d'opérations complexes d'assemblage ou de maintenance tout en répondant aux critères identifiés en 2.2.4 et en restant compatible à la fois en réalité virtuelle et augmentée. Ainsi, nous proposons **INTERVALES : INTERactive Virtual and Augmented framework for industrial Environments and Scenarios**.

Le framework proposé est le résultat d'une analyse théorique des différents modèles existants dans le domaine des environnements virtuels et de la réalité augmentée présentés dans le chapitre précédent, combinée à des travaux de recherche antérieurs portant sur des cas d'usage de différentes situations industrielles et de formation dans des opérations de maintenance et d'assemblage en réalité virtuelle et augmentée (Badets et al., 2020; Havard et al., 2015, 2016; Havard, Jeanne, et al., 2019; Havard, Trigunayat, et al., 2019). À la suite de cette analyse, et comme nous le détaillerons dans la section 3.2, le framework a été décomposé en plusieurs modules afin de répondre aux exigences identifiées :

- Décrire les caractéristiques de chaque entité de la scène ;
- Définir les actions réalisables et les paramètres associés à chacune lors de l'interaction avec les entités ;
- Gérer et vérifier les procédures et scénarios pendant la réalisation de l'activité ;
- Enfin, exporter le scénario et l'état de la scène afin d'étendre le framework INTERVALES avec des outils externes.

Afin de rester génériques, les modules du méta-modèle INTERVALES sont formalisés en UML. Dans cette section, nous présentons tout d'abord l'architecture INTERVALES qui permet, d'une part, la création de scénarios ou de procédures et, d'autre part, leur utilisation en RA ou en RV. Ensuite, les différents modules du framework INTERVALES sont présentés et le modèle de données UML associé est décrit.

3.2 INTERVALES : Architecture et workflow

L'architecture INTERVALES a été construite, d'une part, pour aider à la création de contenus et de scénarios par des développeurs et des experts métiers ou formateurs (3.2.1), et d'autre part, pour fournir des activités en réalité virtuelle, ou assistance en réalité augmentée, aux utilisateurs finaux, tels que les apprenants ou les opérateurs (3.2.2). L'architecture du framework INTERVALES, permettant de décrire ces deux processus (i.e. création et utilisation), est décrite dans les deux sections suivantes.

3.2.1 INTERVALES pour la création de contenu

Du côté de la création (voir Figure 59), INTERVALES peut être utilisé par deux rôles différents : les développeurs informatiques et l'expert métier / formateur (voir Figure 59). La section suivante explique d'abord l'utilisation du framework par des développeurs informatiques, puis illustre comment ces fonctions peuvent être étendues par un expert ou un formateur d'un domaine spécifique et sera plus amplement détaillée dans le chapitre 5.

Toutes les applications de réalité augmentée ou de réalité virtuelle reposent sur des ressources (aussi appelées assets), principalement des modèles 3D, sur lesquels des fonctionnalités (Feature) sont ajoutées pour permettre aux utilisateurs d'interagir avec eux et de rendre les objets interactifs les uns avec les autres. Pour atteindre cet objectif, INTERVALES s'appuie sur plusieurs modules distincts illustrés dans la figure 59. Le premier est le **module Entity-Feature**. Ce module permet de définir les caractéristiques de chaque asset grâce aux concepts d'« **Entity** » et de « **Feature** ». L'**Entity** représente l'objet 3D. Sur celle-ci sont placées une ou plusieurs **Feature**. Chaque Feature permet de définir son comportement et comment elle doit interagir avec d'autres Features. La section 3.3 expliquera plus en profondeur ces concepts.

Fondé sur ces deux concepts, le **module « INTERVALES Interaction »** permet de gérer toutes les interactions effectuées par l'utilisateur sur chaque Entity et Feature. Ce module est capable de prendre en compte les appareils utilisés (e.g., contrôleur RV ou clavier / souris) afin d'adapter les interactions faites par les auteurs de contenu et de scénario. Lors de l'interaction avec l'**Entity**, et en fonction des **Features** dont elle dispose, le framework INTERVALES envoie alors des événements concernant une **Action** effectuée sur une Entity/Feature ou une Relation créée entre deux Entity/Feature.

Le **module « INTERVALES Scene state »** (voir Figure 59), est chargé d'enregistrer toutes les actions/rerelations effectuées entre les **Features** placées sur chaque **Entity**. Ce module est donc capable de sauvegarder et de charger l'état d'une scène, quelles que soient les interactions faites par l'utilisateur.

Basé sur l'état de la scène, le **module « INTERVALES Scenario Authoring »** permet au développeur ou à l'expert métier / formateur de définir des scénarios grâce à des objectifs basés sur les actions que l'utilisateur final doit effectuer (voir Figure 59). Une fois défini, le **module INTERVALES Import/Export** est en charge de la sérialisation / désérialisation des scénarios et de l'état de la scène dans un fichier au format JSON, basé sur le modèle détaillé dans la section 3.3. De plus, comme les scénarios et l'état de la scène sont basés sur un modèle formel UML, il est possible que des outils externes, tels que les outils de CAO (Conception Assistée par Ordinateur) et PLM (Product Lifecycle Management), intègrent un plugin respectant le formalisme d'INTERVALES afin de produire des scénarios (voir Figure 59 - **Connector with business tools**). Cela permettrait donc au framework INTERVALES d'être interopérable avec

d'autres outils et de se connecter, par exemple, à des outils comme FlexSim⁶ (gestion de flux) ou Webots⁷ (simulation robotique).

Une fois les scénarios définis, ils sont utilisés pour former des personnes ou assister les opérateurs. La section suivante décrit comment les scénarios exportés sont utilisés par le lecteur INTERVALES en RA ou en RV.

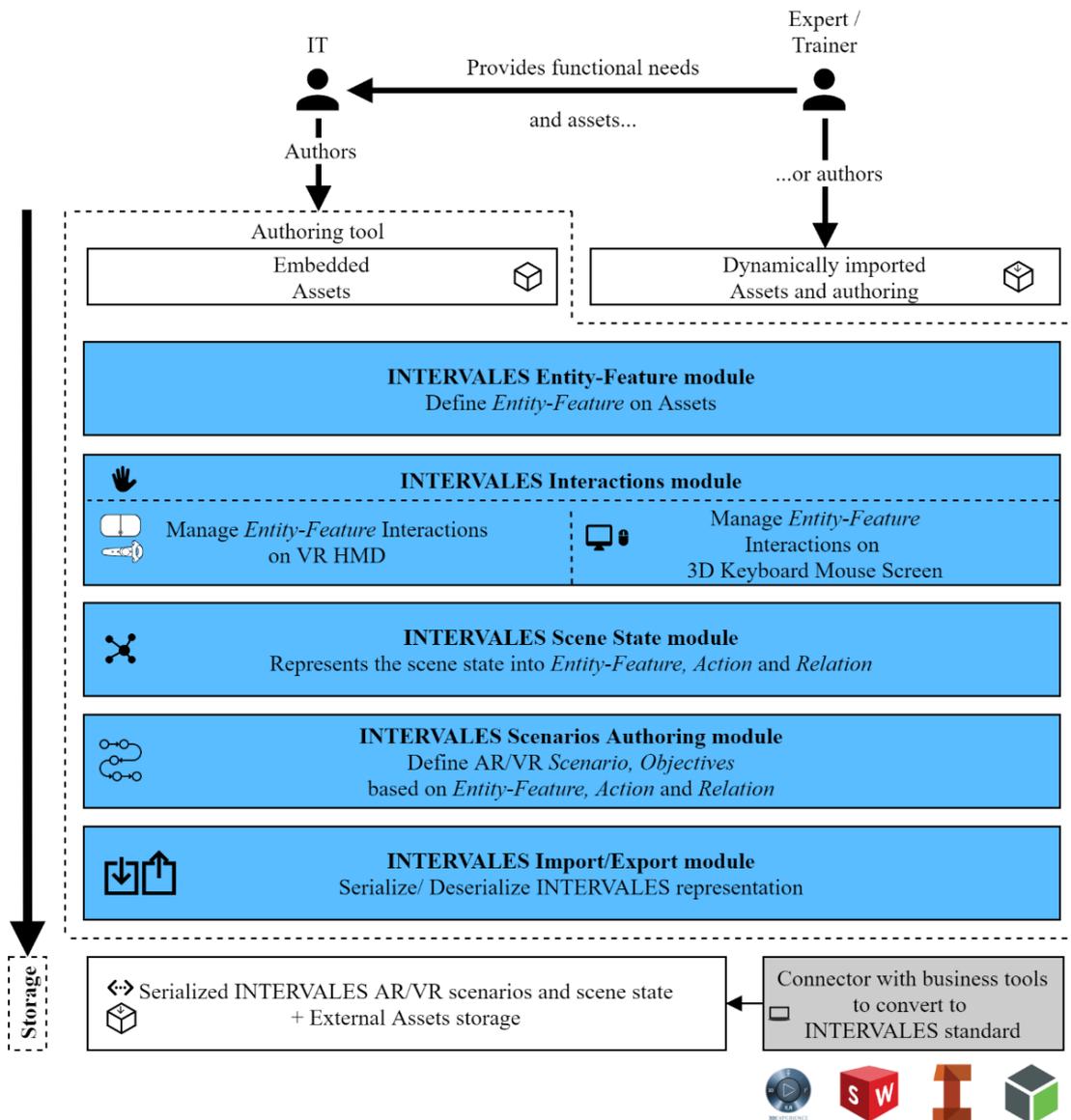


Figure 59 : L'architecture INTERVALES : Partie création

3.2.2 INTERVALES pour la scénarisation en réalité virtuelle et augmentée

Le lecteur RA ou RV, basé sur INTERVALES, est utilisé dans un flux inverse du flux de création, en partant des données stockées par l'utilisateur final (voir Figure 60). Le lecteur commence donc par importer les scénarios dans le lieu de stockage. Ensuite, le module

⁶ <https://www.flexsim.com/fr/>

⁷ <https://cyberbotics.com/>

INTERVALES Import/export (voir Figure 60) gère la conversion du scénario sérialisé en un état de scène et des objectifs pour l'utilisateur. À partir de là, l'utilisateur peut interagir dans la scène virtuelle en toute liberté sur les objets composés d'*Entity* et de *Feature*. Les modules **INTERVALES Interactions** et **Scene state** convertissent chaque interaction en une action/relation entre *Feature* et *Entity*. À partir de cette représentation, le module **INTERVALES Scenarios Checker** compare les objectifs définis par les auteurs à l'état actuel de la scène et les Actions/Relations réalisées par l'utilisateur. Comme chaque Objectif spécifique représente une Action/Relation spécifique avec une valeur de tolérance, le module Scenarios Checker s'assure que chaque Action réalisée par l'utilisateur correspond à un Objectif prévu dans le scénario. Ainsi, il valide que les scénarios définis sont correctement exécutés par l'utilisateur final (voir Figure 60). Un exemple sera détaillé dans la section 3.3.2.1. En ce qui concerne la RA, un scénario est une succession d'étapes que l'utilisateur final doit effectuer sur le système physique. Chaque Action/Relation est représentée par un objet animé en réalité augmentée pour assister ou guider les utilisateurs dans leur tâche.

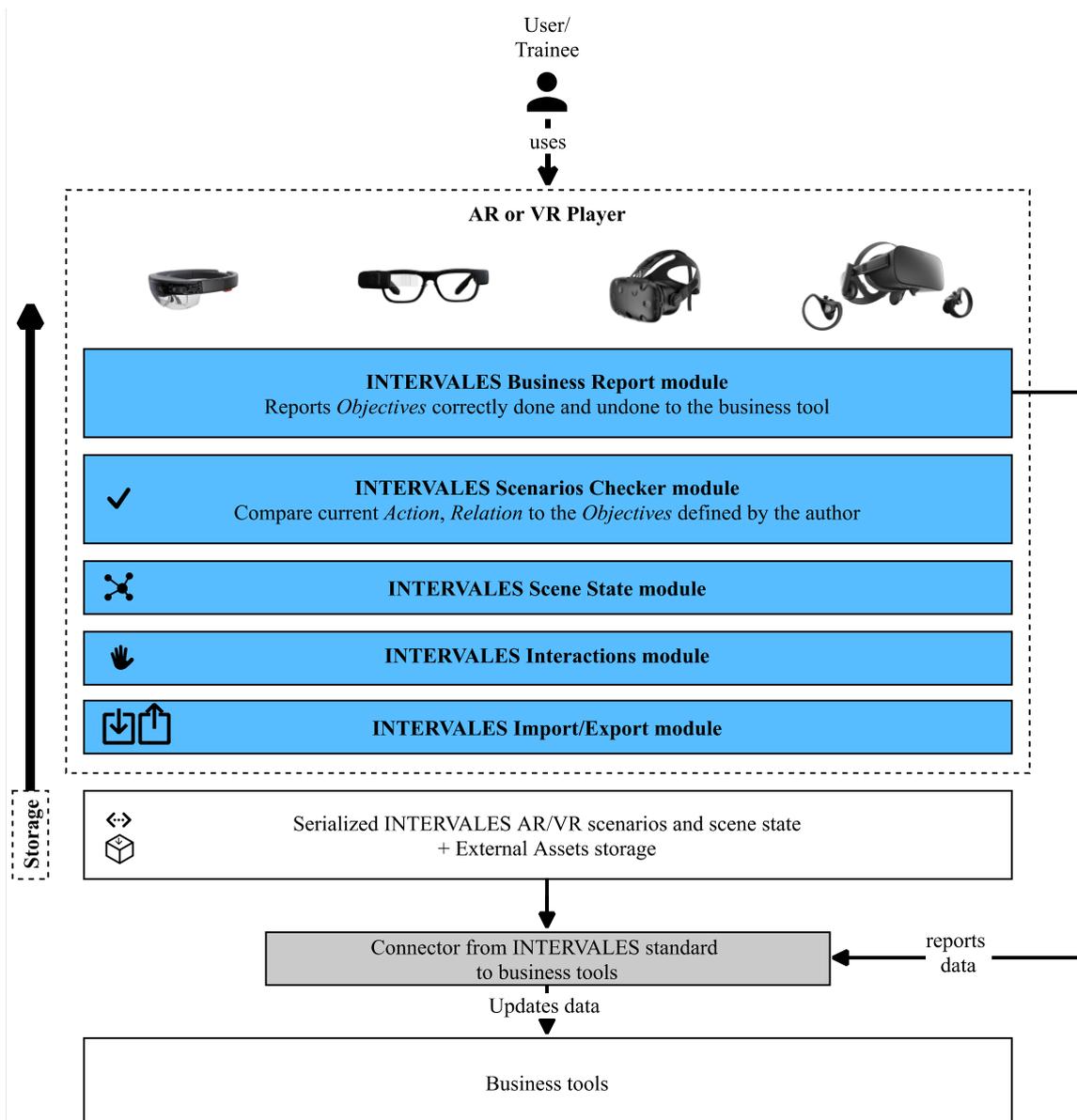


Figure 60 : Architecture INTERVALES : Partie usage de l'utilisateur final.

Enfin, le module **INTERVALES Business Report** peut rapporter chaque étape correctement effectuée par l'utilisateur à l'outil métier pour suivre ce qui se passe dans les ressources métier, comme une maintenance sur une machine ou une formation effectuée par un opérateur pour une machine spécifique.

Afin de détailler les modules présentés, la section suivante présente les différents concepts utilisés dans ceux-ci, tels que *Entity*, *Feature*, *Action*, *Relation* et *Objective*.

3.3 INTERVALES : Un modèle orienté “Features-Relations”

3.3.1 Définition d’environnements

Tout d’abord, il est important de décrire une scène virtuelle avec des concepts génériques qui peuvent être spécialisés pour des besoins spécifiques propres à des métiers. C’est pourquoi, le modèle INTERVALES est basé sur de l’UML avec héritage. Le modèle INTERVALES définit le monde virtuel comme une composition de ces deux types de concepts ; **Entities** et **Features** (voir Figure 61). Grâce à cela, il est possible de définir des **Actions** (effectuées sur une Entity ou une Feature), ou des **Relations** (effectuées entre deux Features) et de créer des interactions. Chaque Entity ou Feature du monde virtuel ou augmenté hérite de la classe **ObjectIdentityTransform** pour définir la pose de l’objet 3D dans la scène. Cette dernière classe hérite de la classe **ObjectIdentity**, qui permet d’identifier rapidement un objet dans la scène à l’aide d’un identifiant et d’un chemin unique.

Afin d’illustrer les concepts détaillés, un exemple de montage sur poste de travail manuel sera utilisé. Plus précisément, une opération d’assemblage comprenant des vis et des inserts d’écrous de différentes tailles (M6, M7, M8 correspondants respectivement au diamètre 6, 7 et 8 mm) sera utilisée. Comme une **Entity** est un objet existant dans l’environnement virtuel, elle a une représentation géométrique, référencée par *ModelName*. Par exemple, il peut s’agir du modèle 3D d’une vis, d’un écrou, d’un outil.

3.3.1.1 Entity et Feature

Afin de rendre les **Entities** interactives, une liste des **Features** associées est définie (voir Figure 61 : *FeatureList*). Elle permet de définir les interactions que l'utilisateur peut faire avec cette Feature et les relations réalisables entre les Features à l’aide de règles de compatibilité. Chaque Feature est située sur l’entité dans l’espace 3D sur trois axes (x, y, z). De plus, deux listes de propriétés permettent de définir avec quelles autres Features une Feature peut correspondre. Pour ce faire, deux listes de propriétés ont été intégrées : *OwnPropertyList*, qui comprend les spécifications des Features, et *CompatibilityPropertyList*. Cette dernière liste de propriétés contient les propriétés requises que doit posséder une autre entité pour être compatible et donc associable au sein d’une relation.

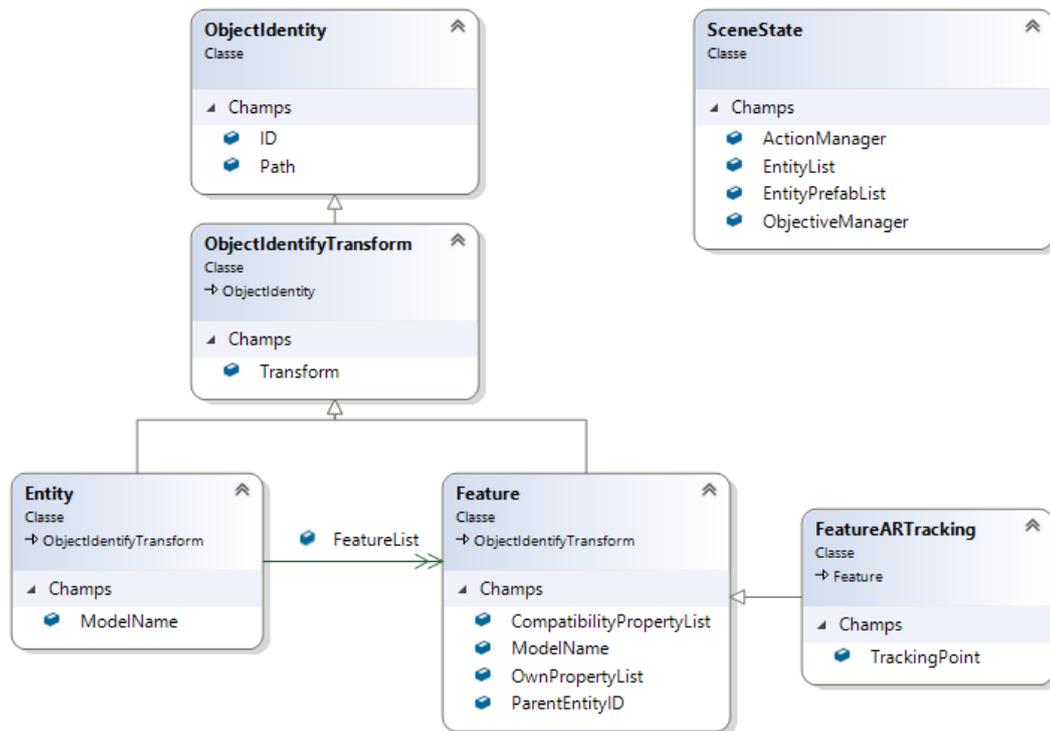


Figure 61 : Diagramme de classe du moteur d'Entity-Feature

Par exemple, si une Feature « Vis_M8 » a une *OwnPropertyList* = ["M8_Male"] et une *CompatibilityPropertyList* = ["M8_Femelle"], par conséquent, cette Feature ne peut être associée qu'à une autre Feature, par exemple « Insert_M8 », qui a *OwnPropertyList* = ["M8_Femelle"] et *CompatibilityPropertyList* = ["M8_Male"].

Il est important de noter que lorsqu'une Relation est établie entre deux Features, l'Entity contenant chaque Feature se déplacera pour représenter la Relation. Étant donné que les Features possèdent leur référentiel 3D, aucune confusion d'orientation n'est possible lorsqu'elles correspondent entre elles. En effet, quand deux Features sont appariées, elles sont orientées pour que l'axe y soit en sens opposé (voir figure 62).

Par ailleurs, puisqu'une Entity peut avoir plusieurs Features, il est possible de décrire des Entities complexes dans un contexte d'assemblage qui pourraient avoir plusieurs fois les mêmes features. Par exemple, dans la figure 62, deux Entity sont présentes et des « FeatureSnap » (Snap = Accrocher) ont été disposées sur chacune d'elles. Le principe d'une « FeatureSnap » est de permettre ainsi de lier deux Entities entre elles au niveau de la position des « FeatureSnap ». Ainsi, à la sélection d'une « FeatureSnap » puis d'une deuxième, une Relation est créée entre les deux Entities.

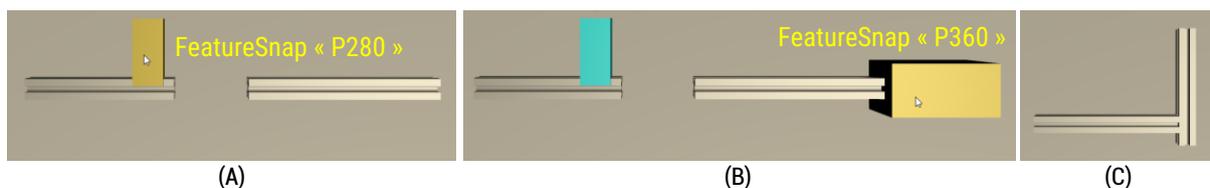


Figure 62 : Création d'une relation depuis une preuve de concept du framework INTERVALES. (A) Sélection de la première feature à connecter sur l'Entity « P280 » (en jaune) ; (B) Sélection de la seconde figure à connecter sur l'entity « P360 » (en jaune), la première feature sélectionnée apparaît en bleu ; (C) Résultat de la relation mise en place.

De plus, concernant la RA, il est possible de définir une classe fille héritant de Feature pour permettre de faire de la réalité augmentée (voir Figure 61 : FeatureARTracking). Cette FeatureARTracking permet de définir la configuration de suivi RA et de l'associer à une Entity spécifique. Dans le cas d'une procédure d'assemblage, il sera alors possible de récupérer la FeatureARTracking permettant de reconnaître l'Entity réelle afin d'afficher en réalité augmentée l'information indiquant l'action à réaliser. Plus précisément, le modèle INTERVALES permettra de définir différents types de FeatureARTracking en fonction du type de reconnaissance à utiliser : FeatureARTrackingImage pour une reconnaissance par image ou FeatureARTrackingModel pour une reconnaissance basée sur un modèle 3D. Une mise en application des FeatureARTracking sera présentée plus en détail dans le chapitre 4 (4.3.1.6).

Il convient alors de noter que chaque Entity peut contenir plusieurs types de Features ; par exemple une Entity « P360 » peut avoir une FeatureSnap prévue pour une Entity « P280 » ainsi qu'une FeatureGrab (« Attrapable ») ce qui permet de créer des objets complexes à partir de ce concept.

De plus, puisque nous voulons donner la possibilité aux utilisateurs de se tromper lors d'une session de formation en réalité virtuelle, des cas de compatibilité plus complexes sont supportés par le modèle INTERVALES. Par exemple, un scénario prévoyant d'associer une vis M8 dans un insert M8, permet aussi aux auteurs de rendre possible le fait d'y placer une vis M7 ou M6 dans cet insert, même si cela représente une erreur. Pour cela, les associations de chacun d'entre eux doivent se faire en définissant indépendamment les règles de compatibilité.

Du fait que les règles de compatibilité peuvent devenir plus verbeuses au fur et à mesure qu'il y en a beaucoup à définir, la compatibilité exclusive « OR », en plus de « AND » a été incluse, permettant d'apporter les compatibilités de types XOR, NOT, NAND, NOR, XNOR en reprenant le principe des portes logiques. De plus, en permettant l'utilisation d'expression régulière (Regex) comme règle de compatibilité, le modèle permettra ainsi d'accélérer le processus de création et d'optimisation pour les cas complexes de compatibilité. Prenons l'exemple de l'interaction « visser » entre une vis M8 et un insert M8 dans un assemblage spécifique où il est nécessaire de préciser que seules les têtes plates et rondes sont concernées. En utilisant la *CompatibilityPropertyList* de base de l'insert M8, deux entrées de compatibilité doivent alors être respectivement définies lorsque le type de la tête est spécifié, *CompatibilityPropertyList* = ["M8 SCREW FLATHEAD", "M8 SCREW ROUNDHEAD"]. Tandis que la liste de compatibilités des vis à tête plate et cruciforme correspondant à ["M8_NUT"] est la même (voir Figure 63 - a).

Pour cela, des caractères spécifiques sont pris en charge : « & » = AND; " | " = OR; « ^ » = XOR. En utilisant une liste de compatibilités complexes, les propriétés de la vis M8 et de l'insert M8 peuvent être croisées et la liste de compatibilités sera ["M8 & SCREW & (FLATHEAD | CROSSHEAD)"] (voir Figure 63 - b). Une autre façon de permettre des erreurs serait de ne pas spécifier certaines propriétés (voir Figure 63 - c).

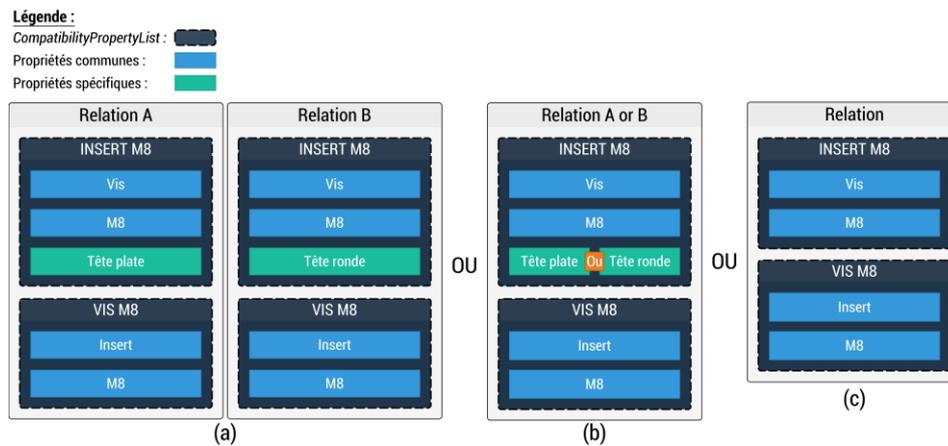


Figure 63 : (a) Déclaration de deux relations sans règles de compatibilité complexes. (b) Déclaration d'une unique relation en utilisant une règle de compatibilité complexe. (c) Déclaration d'une unique relation mais en ne spécifiant aucune propriété.

De plus, dans un cas d'utilisation où toutes les entités du monde virtuel ont une relation définie avec une entité spécifique sauf une, il serait nécessaire de recréer chaque règle de compatibilité indépendamment. L'idée d'inclure une compatibilité XOR "tout sauf..." comme, par exemple, *CompatibilityPropertyList = ["^ FLATHEAD"]* est donc intéressante.

Grâce à cela, les développeurs et les experts métier / formateurs peuvent décrire les *Features* de manière plus réaliste en précisant avec quelles autres propriétés elles sont compatibles. Cela permet de gérer une interaction même s'il y a théoriquement des erreurs (permettre de mettre une vis M6 dans un insert M8 plus gros par exemple). Le modèle peut ensuite gérer quelles erreurs sont autorisées ou non.

3.3.1.2 Action et Relation

Après avoir décrit les concepts d'Entity-Feature, cette section se concentre sur les **Actions** et les **Relations** qui s'appliquent entre Features. L'objectif de cette partie du modèle UML est de suivre chaque action effectuée lors de la session RV ou RA. Elle produit une liste complète des actions effectuées qui est notamment utilisée pour la création et la vérification du scénario. Cette fonctionnalité est détaillée dans le chapitre 4 au travers d'un cas d'usage. Les Actions (voir Figure 64 – b) sont définies par un *ActionType* et une *EntityFeatureReference*. Une action concerne une seule Entity, comme une modification de propriétés ou bien une instanciation. Une Relation, quant à elle, implique deux Features sur deux Entity différentes (voir Figure 64 - c). *EntityFeatureReference* correspond aux Entities et Features impliquées dans une action ou une relation. L'héritage est utilisé pour spécialiser une Relation et y ajouter des informations personnalisées à son sujet. Par exemple, la relation « **RelationSlider** » (voir Figure 64 - d) contient la distance depuis le point de départ du slider (Figure 65).

Afin de lister toutes les actions ou relations faites, il est nécessaire d'avoir un gestionnaire les recensant. C'est le rôle de l'**ActionManager** (voir Figure 64 - a). L'*ActionManager* contient une *ActiveActionList* permettant d'avoir l'état actuel de la scène avec la liste d'Actions et de Relations en cours. La liste *AllActionDoneList* recense toutes les actions créées et retirées depuis le début de l'activité (RV ou RA).

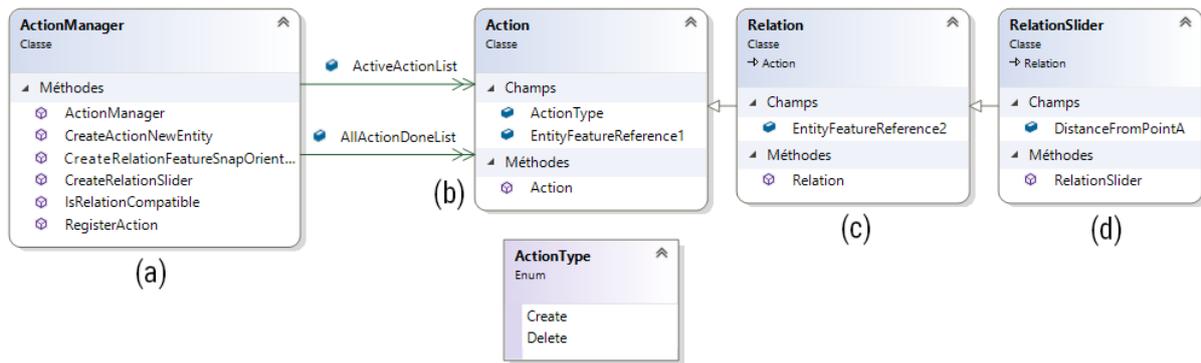


Figure 64 : Diagramme de classe du moteur de relation.

Grâce à ces quatre concepts Entity-Feature-Action-Relation, le module d'interaction INTERVALES gère les interactions faites par l'utilisateur. À titre d'exemple, lorsqu'un utilisateur attrape deux entités et rapproche deux FeatureSlider l'un de l'autre, il peut voir dans l'environnement virtuel une pièce fantôme montrant où la première entité sera placée si la RelationSlider est créée lorsqu'il relâchera l'entité. La figure 65 illustre cette interaction que nous reverrons dans la section 4.3-Cas d'usages industriels.

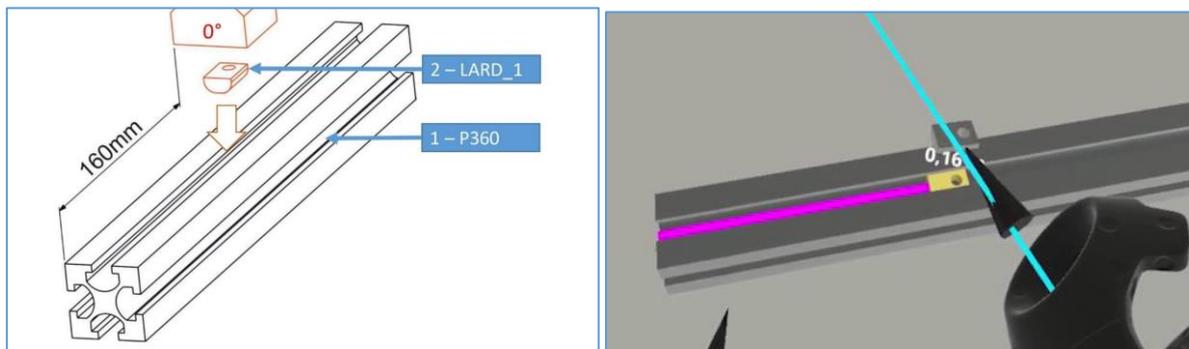


Figure 65 : Exemple de RelationSlider entre une Entity « P360 » et « LARD_1 ». (à gauche) la feuille d'instruction ; (à droite) la réalisation de la relation en RV, avec la pièce fantôme en jaune.

3.3.2 Définition de scénarios et procédures

Avec cette connaissance de l'environnement et de l'état des relations entre les entités, les **Scenario** peuvent être créés plus facilement en définissant les changements attendus de l'état de l'environnement. Les sections suivantes décriront les concepts pour la définition de scénarios (3.3.2.1) et le formalisme mis en place (3.3.2.2).

3.3.2.1 Scenario, Objective et Task

Un scénario est divisé en une série d'Objectifs (**Objective**) à atteindre (voir Figure 66 - Objectif). Un objectif peut être une tâche (**Task**) unique (voir Figure 66 : UnitTask) à effectuer ou un regroupement de tâches (voir Figure 66 : GroupTask) à effectuer dans n'importe quel ordre ou de manière séquentielle. Une Tâche est en elle-même une Action qui doit être réalisée par un **Agent**, c'est-à-dire un acteur de l'environnement. Il peut être réel (ex : opérateur) ou autonome (ex : robot manipulateur mobile). De toute évidence, chaque Action (ou Relation) a une tâche correspondante. Par exemple, la relation « RelationSlider » (visible dans la figure 65) a la tâche correspondante TaskSlider. En définissant le seuil de tolérance de l'action attendue, il est ainsi

possible de définir comment cette tâche peut être considérée comme « effectuée ». Dans l'exemple du RelationSlider, en définissant la distance $DistanceFromPointA = 25$ mm et la $ValidationRules = [“DistanceFromPointA.Tolerance = \pm 1”]$, le *TaskSlider* est alors considéré comme correctement réalisée par le module **INTERVALES Scenarios Checker** si les éléments sont glissés entre [24mm, 26mm]. Il est également possible de définir une tâche comme nécessaire, optionnelle dans l'objectif, voire critique.

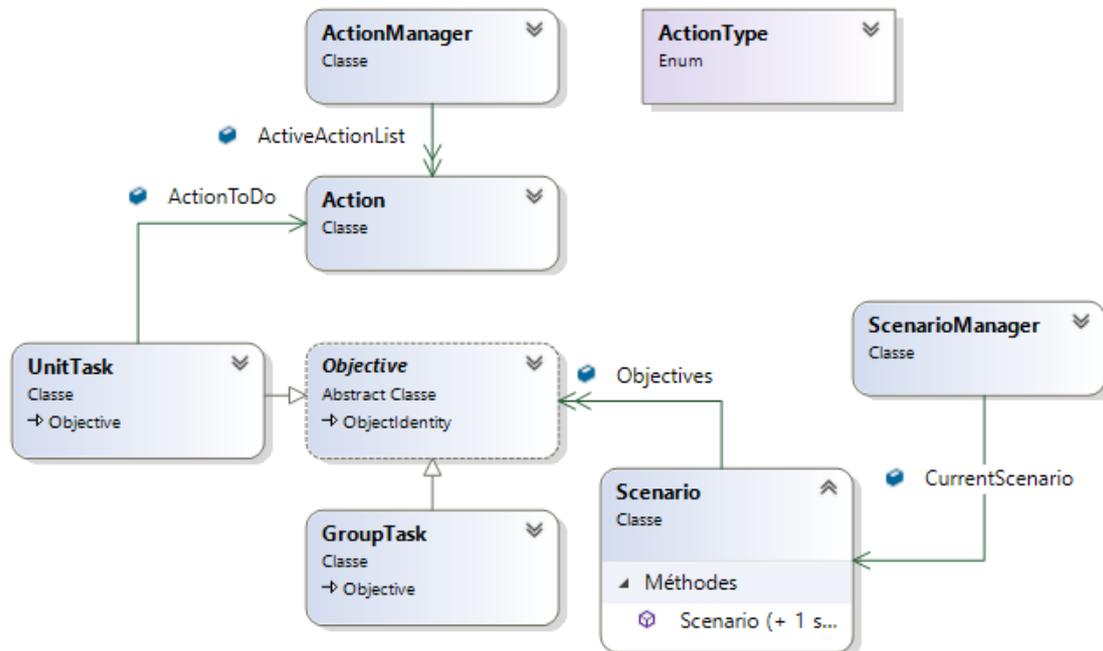


Figure 66 : Modèle d'orchestration des tâches et objectifs.

Enfin, grâce à ces trois concepts Scénario-Objectif-Tâche, cette disposition des objectifs et des tâches permet de gérer des scénarios procéduraux ou des scénarios plus complexes avec des tâches qui pourraient être effectuées dans des ordres différents. Le module de scénario d'INTERVALES permet donc, par exemple, de définir un scénario procédural pour une application de maintenance en RA avec des actions à effectuer dans un ordre spécifique, ou bien de définir un processus d'assemblage en RV contenant plusieurs tâches à exécuter sans ordre spécifique. D'autres cas d'applications d'INTERVALES seront présentés dans le chapitre 4 et en perspective de ces travaux (6.2).

3.3.2.2 INTERVALES : Formalisation du séquençement

Grâce aux concepts de bases définis dans la section précédente, notre scénario se formalisait donc hiérarchiquement en une suite d'Objectifs (**Objective**). Un objectif pouvant être une tâche unitaire (**UnitTask**) ou un regroupement d'objectifs (**GroupTask**) à effectuer dans n'importe quel ordre ou l'une après l'autre, soit respectivement, en parallèle ou de manière procédurale. Et une tâche est, comme expliqué dans la section précédente, une **Action** qui doit être réalisée. Ainsi, le séquençement peut être géré par la récursivité des regroupements d'objectifs (GroupTasks). Nous avons alors quatre blocs de construction principaux : InitialTask (Tâche initiale) ; EndTask (Tâche finale) ; UnitTask (Tâche) ; GroupTask procédural et parallèle (Figure 67).

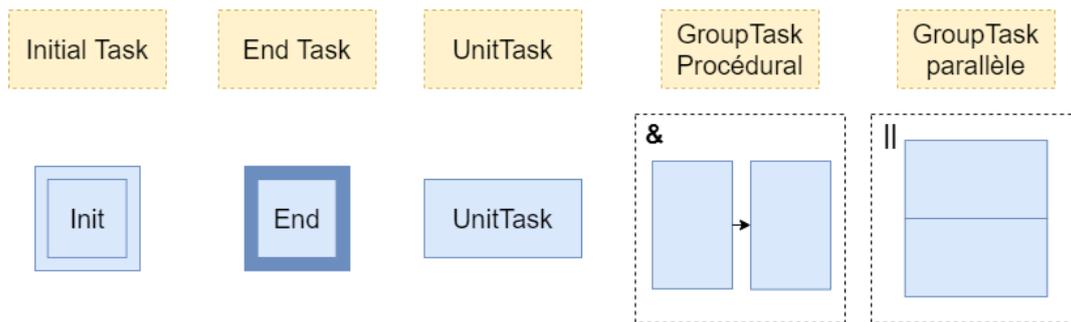


Figure 67 : Éléments du premier formalisme

Afin de donner un exemple, nous utiliserons le scénario suivant pour toute la suite de cette section. Imaginons le remplacement d'une roue où il est nécessaire de visser quatre boulons A, B, C et D en diagonales, soit deux paires d'actions $\left(\frac{A}{C}\right)$ et $\left(\frac{B}{D}\right)$ pouvant être effectuées dans n'importe quel ordre (Ex : (A ou C) ou (C ou A)). Les conditions de réalisations peuvent s'écrire comme suit : $\left(\left(\frac{A}{C}\right) \& \left(\frac{B}{D}\right)\right) \parallel \left(\left(\frac{B}{D}\right) \& \left(\frac{A}{C}\right)\right)$. En prenant, le formalisme expliqué précédemment, nous arriverons à la représentation suivante (Figure 68 – à gauche). Cependant, nous devrions alors contraindre la définition du scénario en imposant de commencer par la paire A/C par exemple. Pour donner le libre de choix de la 1^{ère} paire à réaliser, nous pouvons nous rendre compte qu'un nouvel élément de langage est alors nécessaire (Figure 68 – « ? » - à droite) afin d'exprimer un « OU » exclusif.

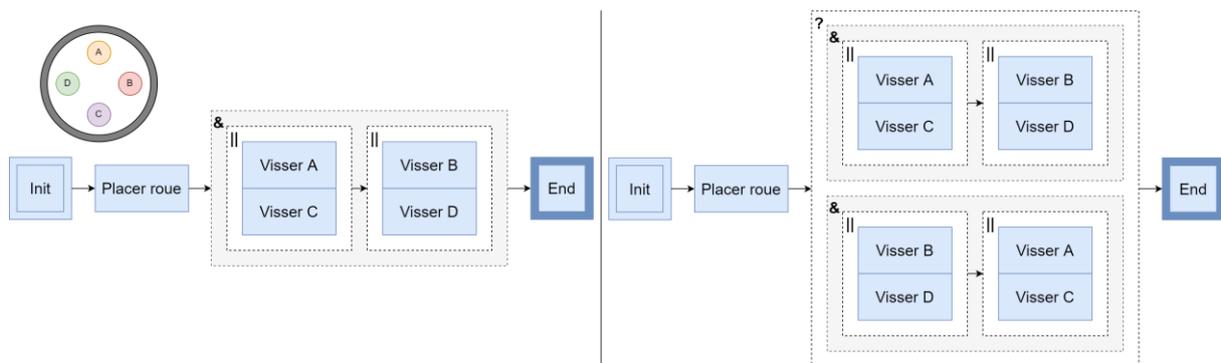


Figure 68 : À gauche, exemple de scénario avec seulement les concepts de bases. À droite, le même scénario avec la nécessité d'ajouter un nouvel élément de langage.

En se basant seulement sur les concepts de bases proposés dans le moteur de scénario INTERVALES, ce formalisme composé de tâches et de groupes reste très basique. En effet, bien que celui-ci réponde aux critères de séquentialité, il reste limité en termes de concurrence et devient confus à la lecture dans le cas de scénarios complexes. Ainsi, comme explicitée dans la section 2.3, nous retrouvons l'importance de basculer à une représentation en graphe, plus claire et permettant d'exprimer des scénarios plus complexes.

C'est pourquoi nous avons fait évoluer le moteur de scénario INTERVALES (Figure 69) en ajoutant le concept de **ScenarioObject**. Notre formalisme s'inspire de formalismes industriels, utilisés dans la programmation d'automate par exemple, comme le Grafcet (2.3.5), le Grafchart (2.3.6) ou le Flowchart. Nous retrouvons notamment le concept de nœuds (Figure 69 - « ObjectiveNode », « ConditionalNode », etc) et de connexions (Figure 69 – Connexion).

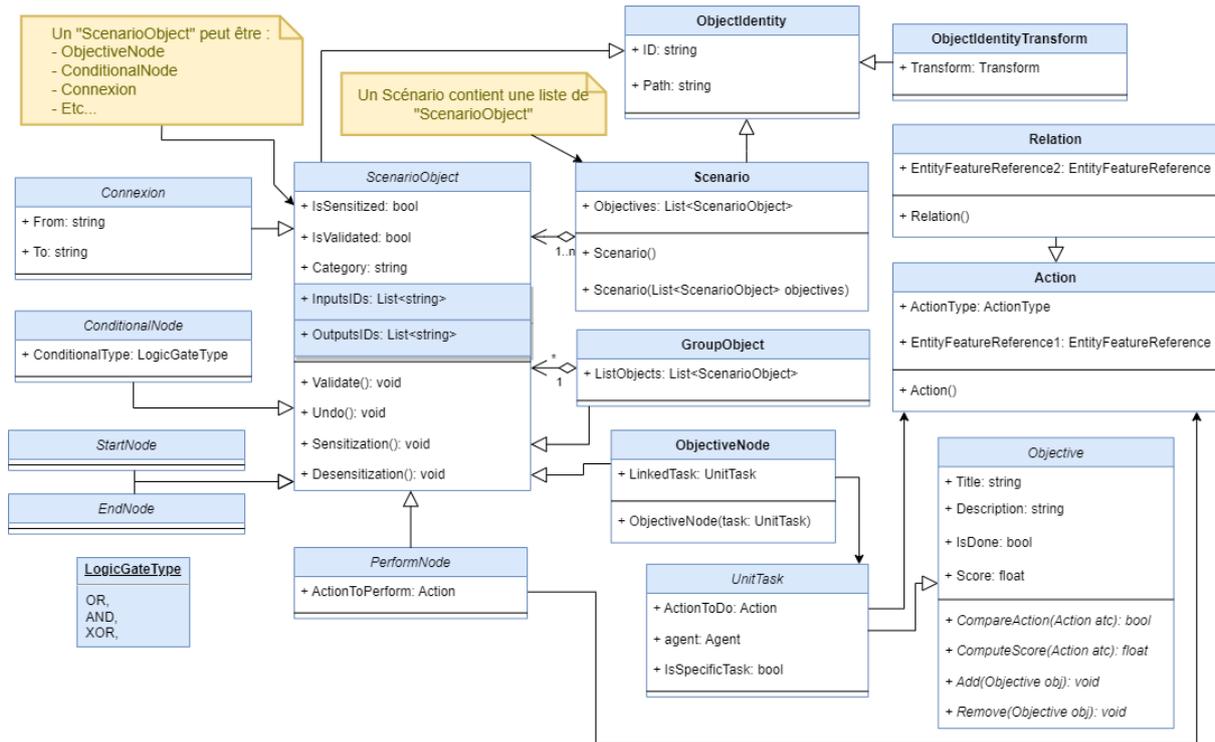


Figure 69 : Formalisme proposé appliqué au moteur de scénario INTERVALES

Un **Scenario** n'est alors plus directement constitué d'une liste d'Objective mais est constitué d'une liste d'éléments « **ScenarioObject** ». Un **ScenarioObject** est une classe abstraite dont héritent tous les blocs de construction d'un Scenario. Un Scenario va principalement être constitué d'**ObjectiveNode** et de **Connexion**. Un **ObjectiveNode** correspond à une tâche (UnitTask) à effectuer tandis qu'une **Connexion** relie deux ScenarioObject entre eux (voir Figure 69 - *Connexion : From / To*). Chaque ScenarioObject a connaissance des ScenarioObject entrants (InputIDs) ou sortants (OutputsIDs). Un ScenarioObject peut être sensibilisé et/ou validé (voir Figure 69 - *ScenarioObject : IsSensitized, IsValidated*). Il est dit « Sensibilisé » lorsque celui-ci peut être validé et qu'il devient alors « en attente de validation ». Nous allons alors avoir une fonction de sensibilisation « *Sensitization()* », désensibilisation « *Desensitization()* », de validation « *Validate()* » et d'annulation « *Undo()* » déclinable pour chaque spécification de ScenarioObject. La fonction « Undo » permet notamment de remonter un scénario étape par étape.

Nous avons ainsi défini plusieurs blocs de construction héritant de ScenarioObject (Figure 70) :

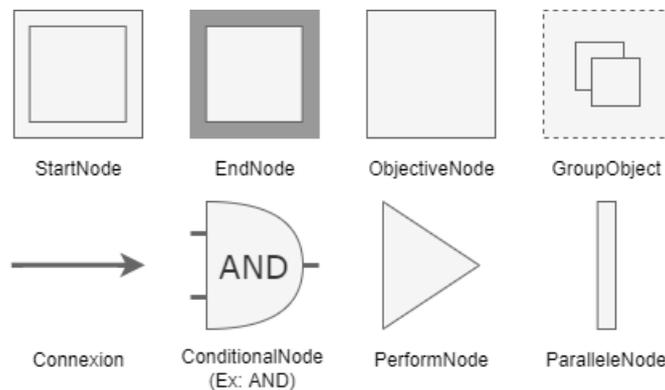


Figure 70 : Blocs de construction du formalisme INTERVALES

StartNode : Ce bloc est en charge d'initialiser un scénario ou un réseau de nœuds au sein d'un scénario. Pour cela, il est validé par défaut au démarrage et va donc sensibiliser toutes les connexions connectées à sa sortie.

EndNode : À l'inverse ce bloc permet d'indiquer la fin d'un scénario un d'un réseau de nœuds. Il se valide automatiquement lorsque toutes les connexions entrantes sont sensibilisées.

ObjectiveNode : Il s'agit du bloc de construction principal. Il correspond à une tâche à effectuer et est donc rattaché à une action définie. Il se sensibilise lorsqu'une des connexions entrantes est sensibilisée. Le scénario est alors en attente de la réalisation de la tâche correspondante. Lorsque celle-ci est faite, le bloc est alors validé. Lors de la validation, le bloc va valider les connexions entrantes qui étaient sensibilisées, afin de marquer par où est passé l'utilisateur, et sensibiliser les connexions sortantes.

GroupObject : Un groupObject contient une liste de ScenarioObject. Un groupObject étant aussi un ScenarioObject, la récursivité est possible. Ce bloc de construction permet de créer des sous-réseaux (Ex : Procédure). Lorsque le groupe est sensibilisé, le premier élément de la liste est sensibilisé également. Dans le cas d'un sous-scénario ou d'une procédure, le StartNode est validé. Le groupObject est validé lorsque le dernier bloc de la liste est validé.

Connexion : Une connexion permet de relier une entrée et une sortie de deux ScenarioObject. Elle peut être sensibilisée/désensibilisée ou validée. Sa sensibilisation entraîne la sensibilisation du bloc suivant. De la même manière, la désensibilisation entraîne la désensibilisation du bloc suivant si celui-ci n'a pas déjà été validé. Par défaut, seulement une connexion sortant d'un ScenarioObject peut être validée à la fois. En effet, lorsqu'une connexion est validée (Figure 71 - 1), les autres connexions connectées à la même sortie sont désensibilisées (Figure 71 - 2). Le bloc sortant est alors lui aussi désensibilisé (Figure 71 - 3).

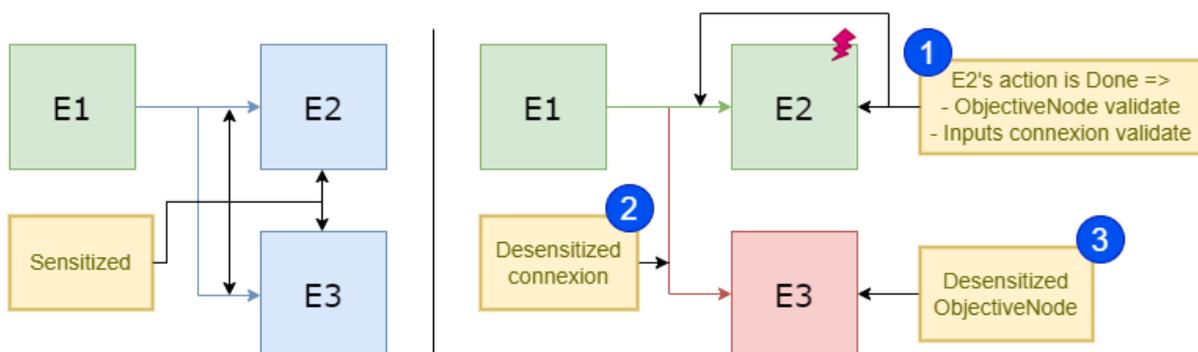


Figure 71 : Cas d'exemple dans le cas de plusieurs connexions sortantes

ConditionalNode : Ce bloc reprend le principe des portes logiques et permet d'appliquer une condition logique à deux connexions entrantes. Par défaut, un ScenarioObject se comporte comme un OR, c'est-à-dire qu'il suffit qu'une seule des connexions soit sensibilisée pour qu'il soit sensibilisé à son tour. Grâce à ce bloc, les conditions AND, NOR, NAND et XOR sont également possibles. Lorsque les conditions sont réunies, le bloc est validé et les connexions sortantes sont sensibilisées.

PerformNode : Le PerformNode permet de déclencher automatiquement une action lorsque ce bloc est sensibilisé. Si l'action est possible, le bloc est alors directement validé et sensibilise donc les connexions sortantes. Par exemple, dans un scénario d'assemblage cobotique, cela permet, de déclencher une ActionSnap sur le robot lorsque la pince de ce dernier se ferme.

ParalleleNode : Ce bloc agit comme une divergence et permet au scénario d'avoir plusieurs chemins parallèles. Pour cela, ce bloc permet la validation de plusieurs connexions sortantes contrairement aux autres blocs. Il se sensibilise lorsqu'une connexion entrante est sensibilisée et se valide lorsqu'au moins une des connexions sortantes est validée.

Afin de mieux comprendre notre proposition de formalisme, nous reprendrons le même exemple vu précédemment en simulant l'exécution d'un scénario tel que les boulons C puis A, suivis de B et D, seront vissés (Figure 72).

Au démarrage du scénario, la *StartNode* va alors sensibiliser les connexions sortantes. Ces connexions vont elles-mêmes sensibiliser les *ScenarioObjects* sortants. Les deux *ParalleleNode* vont ainsi sensibiliser les connexions sortantes. Nous avons alors comme état de départ les deux paires d'opérations sensibilisées et en attente de réalisation (Figure 72 - 1). Lorsque le boulon C est vissé, l'*ObjectiveNode* est alors validé et va valider la connexion entrante. Le *ParalleleNode* va se valider et l'autre connexion vers la paire B et D est alors désensibilisée. En effet, ce scénario a été défini de la sorte pour garantir que si l'on commence par une paire A/C par exemple, il sera possible de valider l'autre paire B/D que lorsque la paire A/C sera validée (Figure 72 - 2). Ensuite lorsque le boulon A est vissé, la connexion entrante va aussi être validée et la connexion sortante sensibilisée (Figure 72 - 3). La *ConditionalNode* est ensuite validée car toutes les connexions entrantes sont sensibilisées et vont permettre de resensibiliser le *ParalleleNode* pointant vers B et D (Figure 72 - 4). L'*ObjectiveNode* B pourra alors être validé (Figure 72 - 5) et de même pour la D (Figure 72 - 5). La *ConditionalNode* est alors validée. Puis les deux paires étant validées la dernière condition est alors validée également et le scénario est complété.

En synthèse, nous avons traité, dans cette section, de la représentation graphique d'un scénario / procédure au sein du framework INTERVALES. Nous aborderons plus tard, dans le chapitre 5, l'interface graphique de création et d'édition ainsi que d'autres exemples de scénarios.

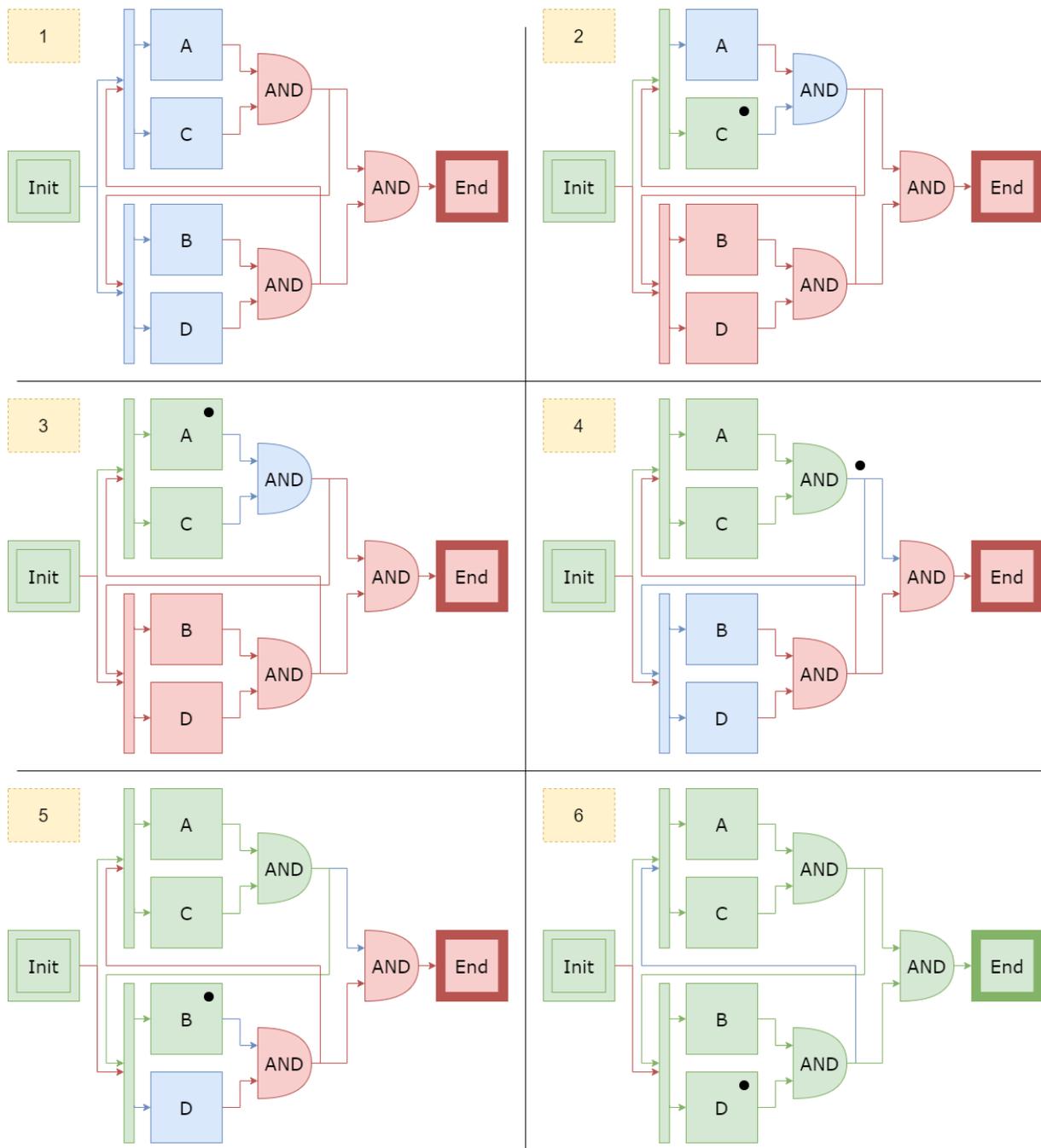


Figure 72 : Exemple de scénario en utilisant le formalisme INTERVALES (Vert : Validé, Bleu : Sensibilisé, Rouge : Désensibilisé)

3.4 Architecture et développement du framework INTERVALES

Nous avons vu dans les sections précédentes notre proposition de modèle avec son architecture et ses concepts. Nous présenterons dans cette section le développement du framework INTERVALES et son intégration au moteur de jeu Unity 3D, moteur de rendu 3D temps réel très représenté dans les publications scientifiques et dans l'industrie et donc utilisé chez OREKA.

Le framework INTERVALES a été développé en langage C# et se divise donc en 2 parties : la librairie au format **.dll** et un package Unity 3D (Figure 73).

- La librairie **INTERVALES_Library.dll** correspond au modèle INTERVALES. Elle se découpe en deux blocs : **Core** et **Process**. La partie **Core** contient la définition des classes de base en reprenant les concepts d'INTERVALES. C'est le méta modèle. La partie **Process**, va permettre de spécialiser les concepts suivant un besoin métier identifié, pour par exemple, spécialiser le concept de Feature en FeatureSlider ou FeatureSnapOriented ainsi que leurs Relation, Task associées.
- Ensuite, le package Unity **INTERVALES_UnityPackage** permet ainsi de faire le lien entre le modèle (la librairie) et son implémentation pour le moteur Unity. Ce découpage permet donc de dissocier les comportements et les interactions en fonction des moteurs de rendus ou outils implémentant le framework INTERVALES. Il fait l'interface entre les données et sa mise en œuvre dans le moteur 3D ou logiciel utilisé. Pour faire cela, le framework utilise un gestionnaire d'évènement que nous détaillerons plus loin dans le chapitre 5 et le concept de Maker que nous introduirons dans la section 3.4.2. Ainsi, il est possible de faire un équivalent sur un autre moteur comme Unreal Engine ou dans un outil de CAO et de rendre donc le modèle INTERVALES interopérable.

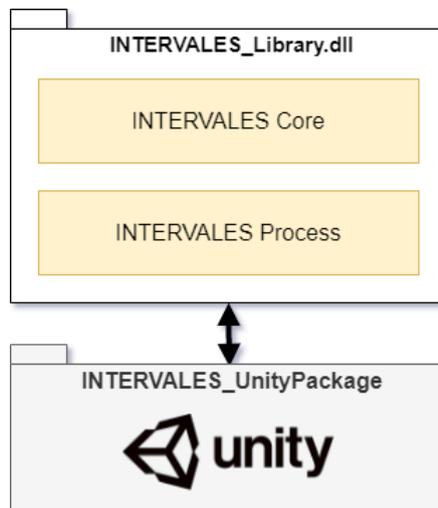


Figure 73 : Organisation globale des développements du framework INTERVALES

3.4.1 Librairies INTERVALES

La librairie **INTERVALES_Library.dll** se décompose en 2 sous-parties : **Core** et **Process** (Figure 75). **INTERVALES_Core** contient donc les classes basées sur concepts pour la description de l'environnement, avec l'*Entity*, *Feature* et *Action/Relation*, et également de scénario, *Scenario*, *ScenarioObject*, *GroupObject*, *Objective/UnitTask*. Il s'agit alors principalement de classes abstraites dont vont pouvoir hériter les classes spécialisées dans **INTERVALES_Process**. Cette partie **Core** contient aussi les classes permettant d'appliquer un identifiant unique, un chemin de référence ou une position à n'importe quel élément avec *ObjectIdentity* et *ObjectIdentityTransform*. Il a fallu pour cela redéfinir ce qu'est une *Transform* avec une position (*Vector3*) et une rotation (*Quaternion*). Nous avons également choisi d'y placer le gestionnaire d'évènement afin que celui-ci puisse être accessible et utilisé par l'ensemble des autres éléments devant déclencher ou écouter un évènement. Nous reviendrons sur ce gestionnaire d'évènement de manière plus détaillée dans la section 5.3.

INTERVALES_Process va ainsi se baser et hériter d'**INTERVALES_Core** pour se spécialiser. Il s'agit de sa première fonction. Nous retrouvons, par exemple dans la figure 75, la Feature « *FeatureSnapOriented* » (Figure 75 – en vert) qui est utilisée lors de l'action « *RelationSnapOriented* » (Figure 75 – en orange). Afin de créer ces actions, une classe « *Factory* » est définie pour chaque spécialisation, ici « *RelationSnapOrientedFactory* » (Figure 75 – en rouge). Une fabrique (« *Factory* » en anglais) nous permet de créer une instance d'action pour chaque spécialisation. Enfin, la UnitTask « *TaskSnapOriented* » (Figure 75 – en violet) est utilisée pour définir ce qui est attendu et la tolérance acceptée pour la Task créée.

Concernant la RA, nous retrouvons également la Feature spécialisée « *FeatureARTracking* ». Cette dernière, une fois ajoutée à une entity dans Unity 3D permet de placer un point de référence pour la reconnaissance et le suivi en réalité augmentée. Dans la figure 74, nous pouvons voir le placement d'une *FeatureARTracking* sur une Entity « *P360* » via l'interface d'Unity 3D. Nous reviendrons sur l'implémentation en RA d'INTERVALES au travers un cas d'usage présenté en 4.3.1.6.

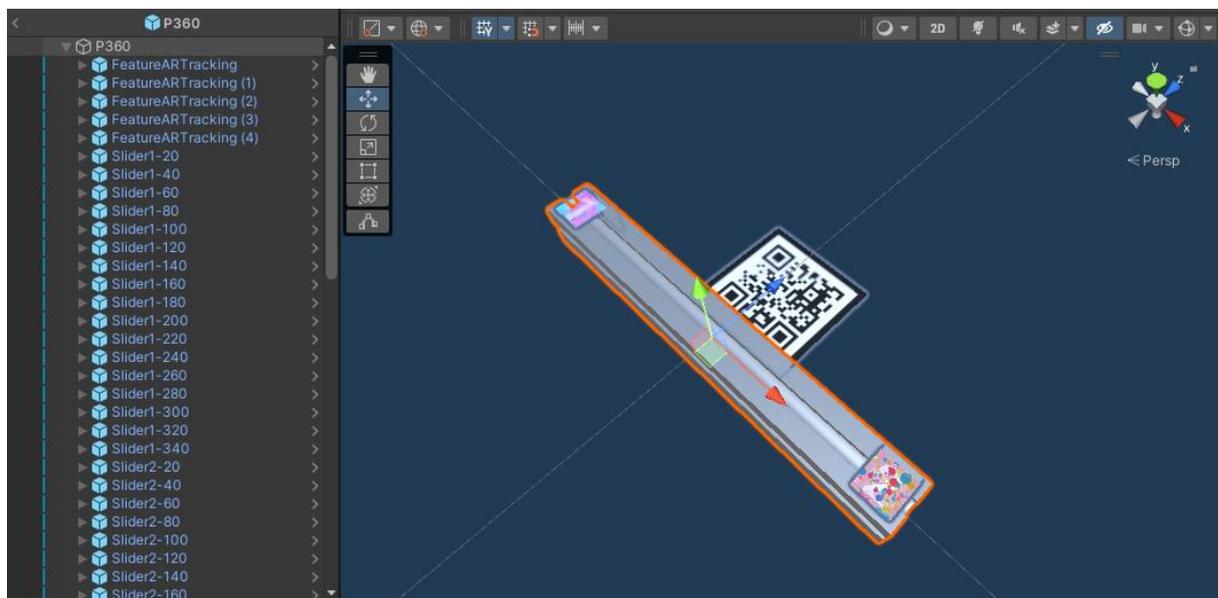


Figure 74 : Placement d'une *FeatureARTracking* sur une Entity « *P360* » au sein d'Unity 3D.

Par ailleurs, afin de pouvoir représenter le scénario sous forme graphique, les différents *Nodes* sont spécialisés (Ex : *ObjectiveNode*, *ConditionalNode*, ...).

Pour finir, nous retrouvons dans **INTERVALES_Process** la classe « *SceneState* » qui va pouvoir être sérialisée pour stocker/charger l'état d'une scène à un instant t . Cette classe comprend la liste des Entity présentes, les actions qui ont été réalisées ou bien le scénario en cours.

INTERVALES

v0.3.4_alpha

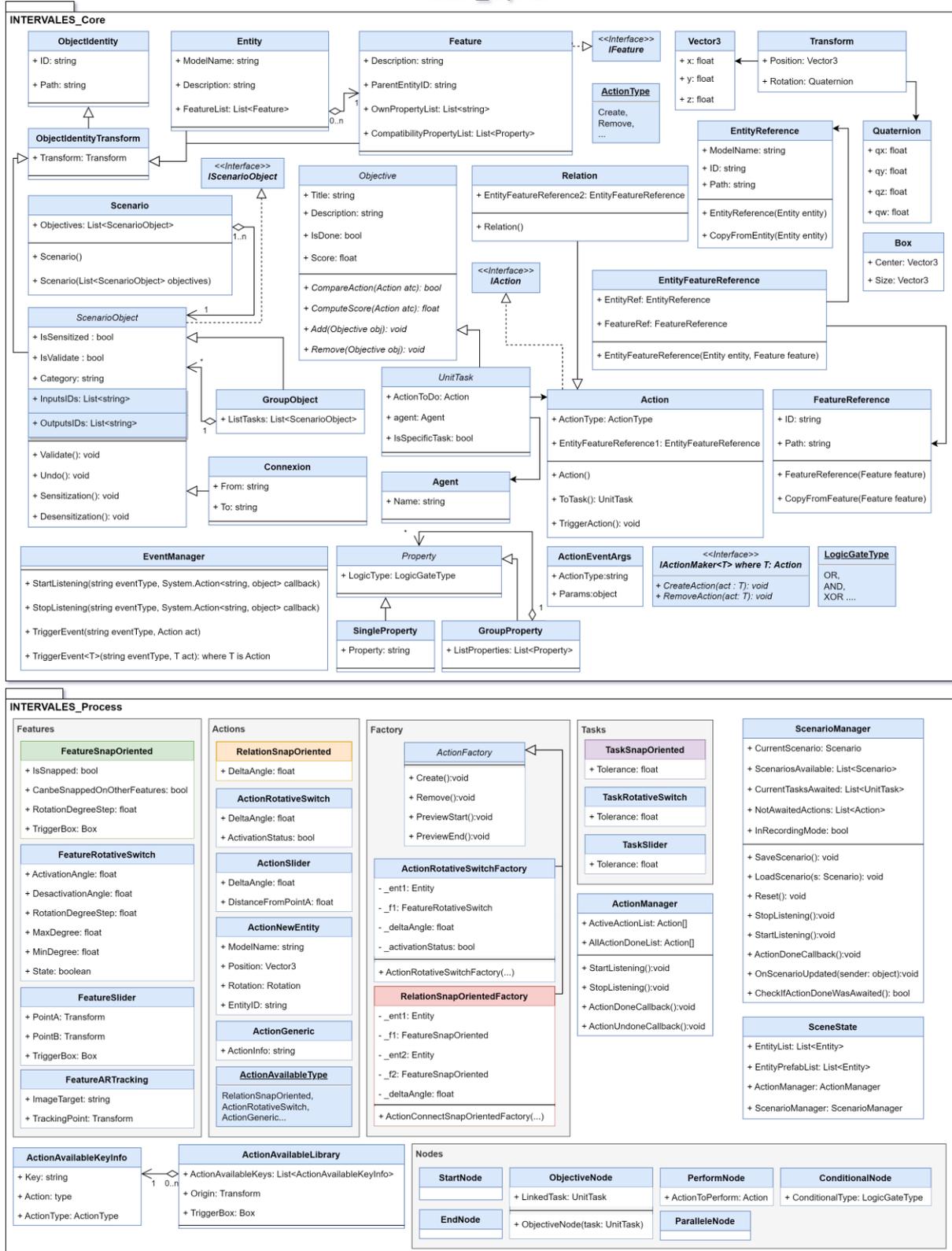


Figure 75 : Diagramme de classe de la librairie INTERVALES_Library.dll

Cette section a permis de présenter la librairie représentant le modèle de données INTERVALES. Afin d'être utilisé concrètement dans un environnement virtuel ou augmenté, il est nécessaire de

représenter chacun de ces concepts au sein du moteur de rendu ciblé. La section suivante détaille l'implémentation dans le moteur Unity 3D.

3.4.2 Intégration dans Unity 3D : INTERVALES.UnityPackage

Dans cette partie, nous allons décrire comment a été intégrée la librairie INTERVALES au moteur Unity. Nous avons choisi Unity pour développer notre prototype INTERVALES car c'est le moteur qui est le plus souvent utilisé en recherche actuellement dans le secteur de la réalité virtuelle/augmentée. Par ailleurs ce moteur est aussi le moteur principalement utilisé dans l'industrie. Unity 3D permet également une plus grande ouverture et s'adapte plus facilement à des besoins spécifiques à la recherche, comme l'implémentation de librairie, de module (Ex : Webots), ou de protocole (Ex : OPC-UA).

Afin d'intégrer INTERVALES, nous avons dû développer un package Unity **INTERVALES_UnityPackage** (voir Figure 76). Le package se découpe en deux parties : **Runtime** et **Editor**. La partie Runtime gère le fonctionnement du package lorsque l'application est en cours d'exécution. La partie Editor gère le fonctionnement seulement au sein de l'interface Unity et lors du développement et de l'édition de la scène virtuelle.

Le premier besoin auquel doit répondre le package est de permettre au moteur 3D d'interpréter les informations de la librairie et inversement. Unity étant basé sur une logique de composants (composants) placés sur des GameObject (objet dans la scène Unity), le package intègre donc les composants basés sur les classes définies dans la librairie INTERVALES. Par exemple, la classe « *INTERVALES_Entity* » reprend les informations définies dans la librairie et le composant va pouvoir être placé sur un objet 3D. Il en va de même pour les Features, les Actions et les ScenarioObject. Pour simplifier l'intégration de ces nouvelles classes spécialisées, nous avons mis en place des classes abstraites génériques (« Template »), servant de modèle et qui vont permettre aux nouvelles classes, créées lors de l'ajout d'une nouvelle Feature par exemple, de s'intégrer directement dans le framework INTERVALES et dans Unity. Dans cette version simplifiée du diagramme de classe (Figure 76), nous pouvons ainsi voir *INTERVALES_FeatureTemplate*, *INTERVALES_ActionTemplate* et *INTERVALES_ScenarioObjectTemplate*.

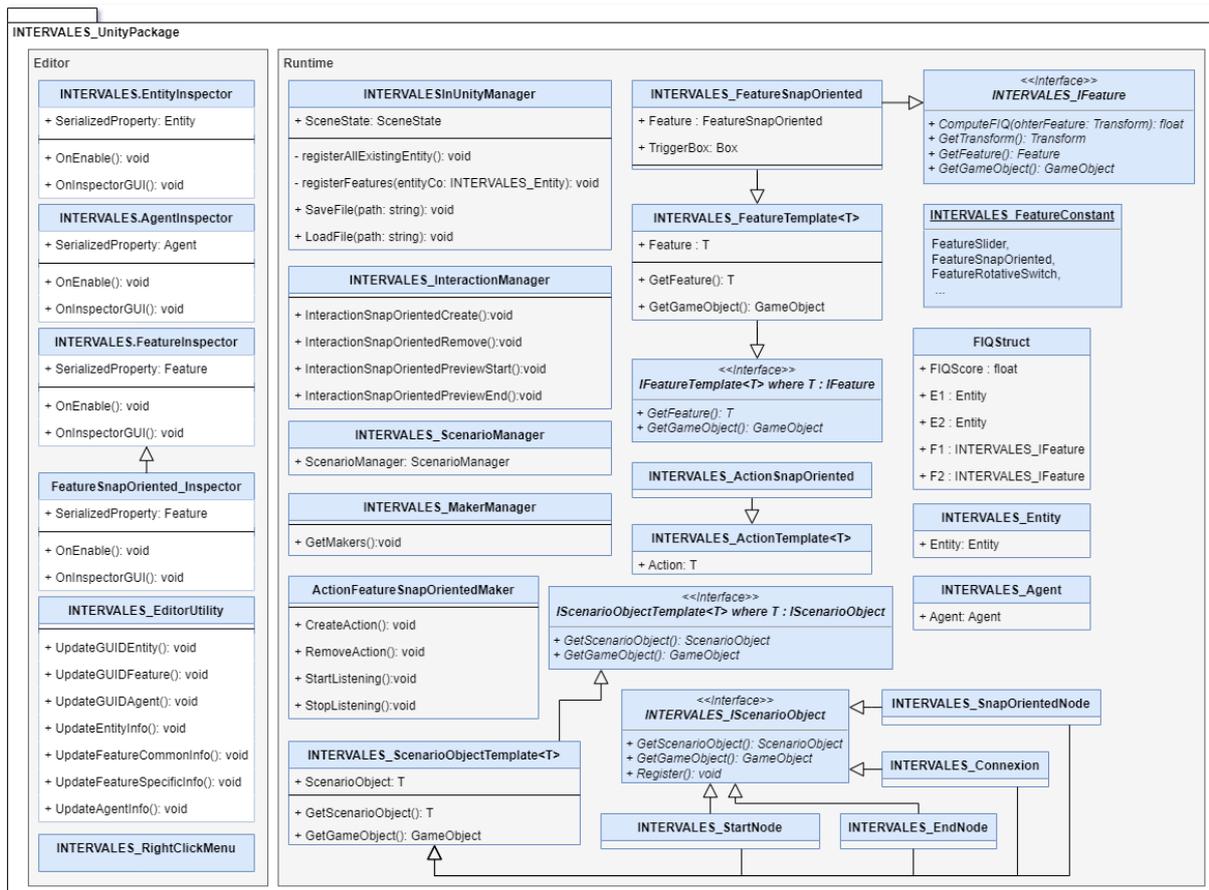


Figure 76 : Diagramme de classe simplifié d'INTERVALES.UnityPackage

Ensuite, chaque action peut disposer d'un **Maker** associé. Le rôle du Maker est de représenter visuellement dans le moteur de rendu utilisé, ici Unity, l'action résultant du déclenchement d'un évènement d'action. Par exemple, la figure 77 illustre le fonctionnement du Maker lors d'une action d'assemblage « RelationSnapOriented ». Ainsi, lorsqu'un utilisateur approche deux Entity, le Maker va interroger les données INTERVALES afin de savoir si l'action est faisable. Si tel est le cas, le Maker affiche une prévisualisation (Fantôme) de ce que la relation donnera une fois réalisée. Si l'utilisateur relâche l'Entity, le Maker envoie l'information au framework INTERVALES qui va ensuite déclencher un évènement « SnapOriented ». C'est ensuite le Maker qui est en charge de visuellement attacher les deux entités au moyen des classes à disposition dans le moteur Unity (RigidBody, Collider, FixedJoint,...). Nous reviendrons sur le fonctionnement des Makers et du gestionnaire d'évènement dans la section 5.3 du chapitre 5.

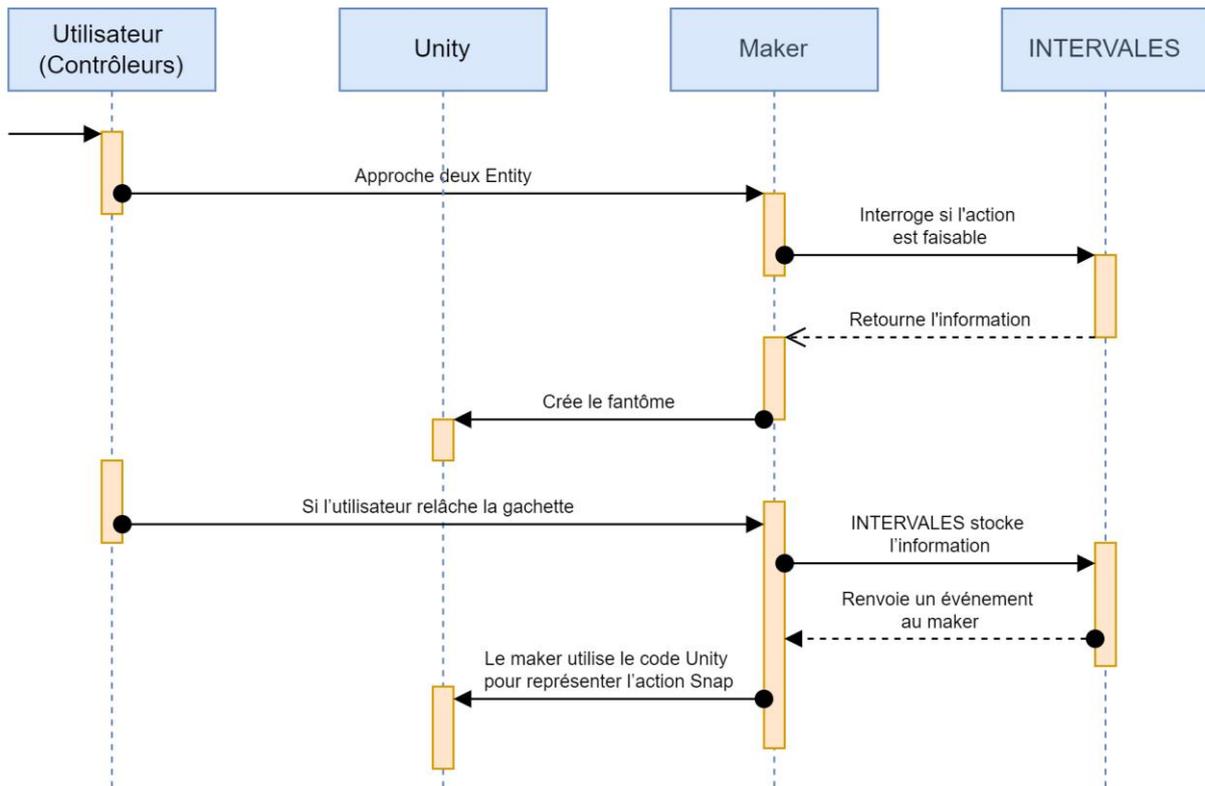


Figure 77 : Diagramme de séquence du fonctionnement du Maker RelationSnapOrientedMaker.

Globalement, le fonctionnement en runtime est géré par les différents managers. *INTERVALESInUnityManager* gère les informations de la scène. C'est aussi lui qui va permettre la sauvegarde / le chargement d'un fichier JSON de l'état de la scène à un instant t . Il contient toutes les données INTERVALES en temps réel. Il y a ensuite *INTERVALES_InteractionManager* qui va contenir des fonctions permettant de déclencher une action suite à une interaction, par exemple lorsque 2 entity sont reliées par une *FeatureSnapOriented*. *INTERVALES_ScenarioManager* va gérer le scénario et *INTERVALES_MakerManager* va gérer l'initialisation des différents makers pour chaque type d'action.

Ensuite, pour la partie Editor, l'objectif a été en plus d'intégrer les concepts INTERVALES, d'incorporer également INTERVALES à l'interface Unity et de proposer des automatismes dans le but de participer à l'un des objectifs d'INTERVALES, à savoir la simplification de la mise en place d'un environnement virtuel et de scénario. Par exemple, nous avons développé des interfaces de configuration personnalisées pour l'inspecteur Unity concernant les Entity (Figure 78) ou les Features (Figure 79). Nous présentons également un prototype d'interface d'édition de scénario au sein de l'éditeur Unity dans le chapitre 5.

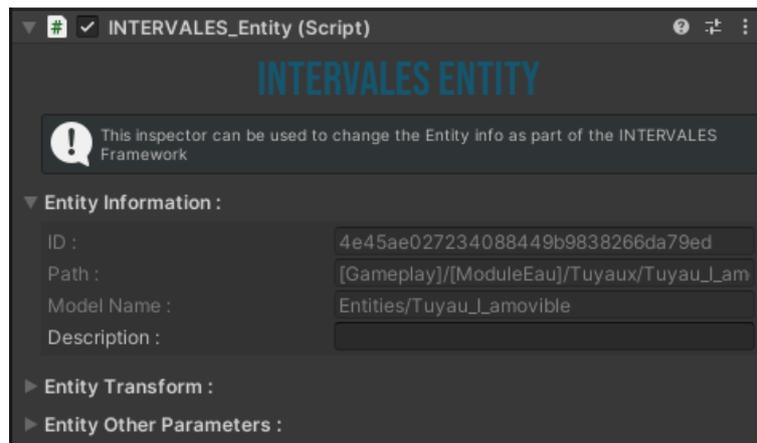


Figure 78 : Inspecteur permettant la configuration d'une Entity

Dans la figure 79 par exemple, le développeur peut ajouter le composant « INTERVALES_FeatureSnapOriented » et configurer la feature via l'inspecteur Unity. Nous pouvons visualiser les informations communes comme l'ID ou le chemin de la feature dans l'arborescence de la scène. Il est ensuite possible de modifier les paramètres spécifiques à la feature ou encore dans notre cas, d'ajouter des propriétés pour définir sa compatibilité avec d'autre feature du même type (Figure 79 – Own Properties List et Compatibility Properties List).

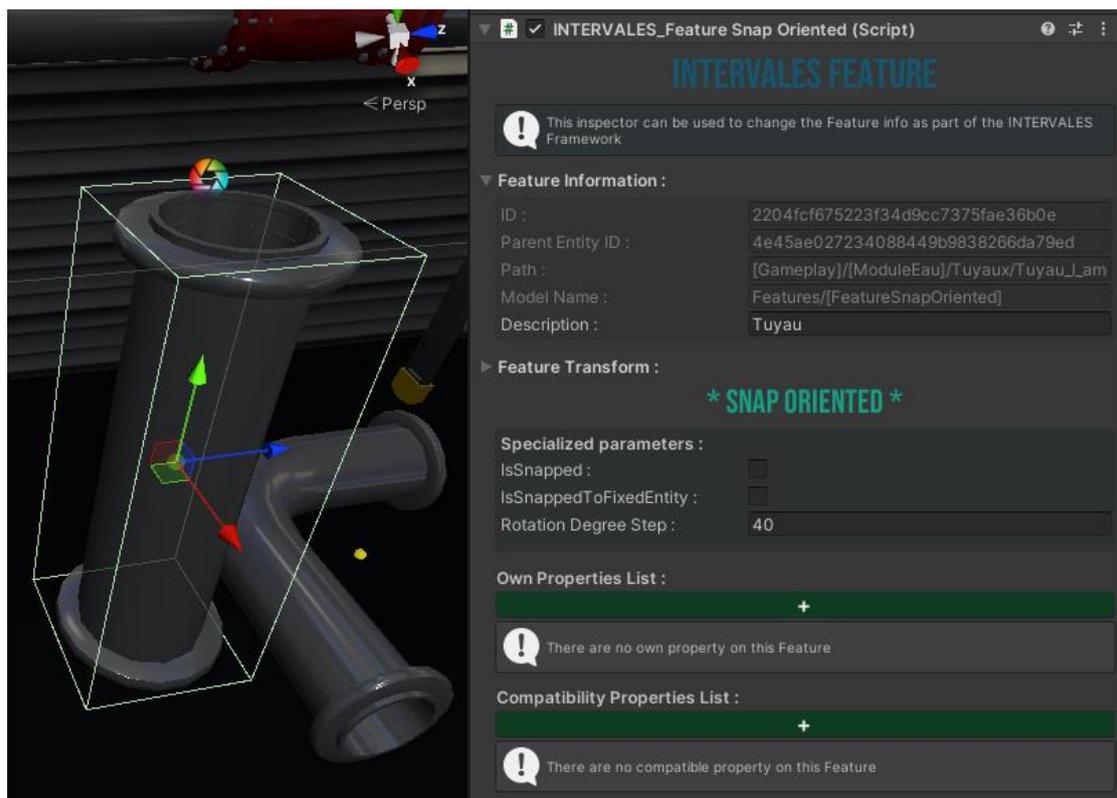


Figure 79 : Inspecteur permettant la configuration d'une FeatureSnapOriented

D'autres automatismes comme un menu rapide ont été mis en place (Figure 80), permettant par exemple de configurer une scène par défaut avec les éléments d'INTERVALES. INTERVALES s'intègre alors facilement et rapidement dans l'environnement de travail Unity et dans un nouveau

projet. Le tout toujours dans l'optique de simplifier son usage et d'avoir un package « Plug and play ».

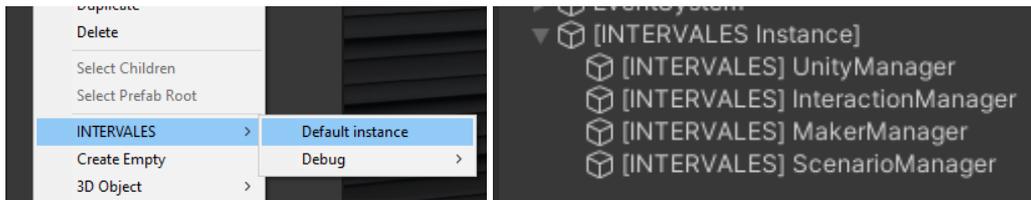


Figure 80 : À gauche, le menu rapide pour instancier INTERVALES. À droite, l'instance d'INTERVALES.

3.5 Synthèse

Pour conclure, nous avons présenté dans ce chapitre une proposition d'un framework nommé INTERVALES (INTERActive Virtual and Augmented framework for industrial Environments and Scenarios). L'architecture et ses différents modules impliqués ont été présentés dans le but de permettre la création de scénarios en réalité virtuelle et/ou augmentée soit par des développeurs informatiques, soit par des experts métier. Le framework proposé est ainsi composé de cinq modules différents : le module « Entity-Feature », pour définir les caractéristiques de chaque élément ; le module d'interaction afin de gérer toutes les interactions faites par l'utilisateur ; le module « Scene state », pour gérer les actions et relations effectuées ; le module « Scenario Authoring », pour définir et éditer des scénarios ; et le module d'import/export, pour enregistrer et charger l'état de la scène à un instant t . De ce fait, ces différents modules rendent possible la création et l'édition de l'environnement et des interactions au sein de celui-ci et permettent de définir des tâches et des objectifs. L'architecture ainsi décrite montre qu'INTERVALES a la capacité de décrire des scénarios RA ou RV dans un langage formel et peut ensuite être utilisé dans des applications RA ou RV. De plus, la formalisation UML du modèle, exportable en JSON, permet à d'autres outils de décrire ou de mettre à jour des scénarios, ce qui rend le modèle interopérable.

De plus, ce chapitre a décrit les classes INTERVALES et comment elles s'architecturent pour définir, par exemple, aussi bien un scénario procédural qu'un scénario d'opérations d'assemblage complexe qui laisse de la place aux erreurs de l'utilisateur. En effet, toutes les actions sont enregistrées qu'elles soient attendues par le scénario en cours ou non. Nous avons également présenté qu'il était possible de remonter le scénario pas à pas car chaque tâche peut être défaire (« undo »).

Ensuite, nous avons proposé un formalisme sous forme de graphe de nœud s'inspirant de représentation industrielle (Ex : Grafcet/Grafchart) pour la représentation de scénario. Notre formalisme utilise le concept de ScenarioObject pouvant être sensibilisé et validé et pouvant être dérivé en différents blocs de construction.

Pour finir, nous avons présenté le développement qui a été réalisé sur le framework INTERVALES et son intégration au moteur Unity 3D. INTERVALES s'intègre alors comme package et offre des automatismes de développement toujours dans une optique de simplification du processus de création d'application en réalité virtuelle et augmentée. Nous illustrerons, dans le chapitre 4,

comment le formalisme ainsi que le package INTERVALES a été mis en application dans plusieurs cas d'usages industriels. Puis, dans le chapitre 5, nous présenterons comment le framework INTERVALES a été utilisé afin de créer un outil d'édition via une interface 2D ainsi qu'un outil d'édition par démonstration (authoring-by-doing).

4 INTERVALES : Mise en application

4.1 Introduction

Nous nous intéressons dans ce chapitre, à la mise en application d'INTERVALES sur différents cas d'usages. Nous présenterons, dans la section 4.2, comment le modèle INTERVALES peut être utilisé au sein du processus de conception d'un projet. Puis, nous présenterons comment INTERVALES a été mise en place avec la présentation de plusieurs cas d'usages industriels dans la section 4.3, que ce soient des développements propres au laboratoire de recherche ou au sein de l'entreprise OREKA Ingénierie. Le fonctionnement d'INTERVALES étant très similaire sur les différents cas d'usages, nous le détaillerons principalement dans le premier cas d'usage en 4.3.1.

4.2 INTERVALES : Utilisation du modèle au sein du processus de conception

Dans le chapitre précédent, l'architecture technique et le modèle de données du framework INTERVALES ont été définis. Pour mieux comprendre comment peut s'intégrer INTERVALES dans un processus de conception d'application et le rôle de chaque partie prenante, nous avons formalisé le workflow d'utilisation au format BPMN (**B**usiness **P**rocess **M**odel and **N**otation) et le décrivons dans cette section. Ce workflow a été proposé notamment suite à l'observation du processus existant au sein d'OREKA Ingénierie afin de ne pas l'alourdir.

Ainsi, comme le montre la figure 81, le processus commence par l'expert métier du domaine. Il représente la personne qui maîtrise le mieux les connaissances et les processus métiers. À partir de là, l'expert a plusieurs options pour travailler avec le framework INTERVALES. Pour la première option, l'expert peut choisir de travailler avec les développeurs afin de produire les applications et les scénarios RA / RV (voir Figure 81 - en violet). Dans un tel cas, le rôle de l'expert est de rédiger les spécifications de l'application tandis que les développeurs doivent, tout d'abord, comprendre le métier et les besoins de l'expert, puis préparer les ressources en conséquence et développer les applications.

Pour la seconde option, l'expert peut choisir de travailler de manière autonome (voir Figure 81 - en bleu). Dans ce cas, il devra définir et placer les concepts d'Entity-Feature sur chacun des objets 3D impliqués dans le scénario. Cela implique à minima que l'expert métier dispose de compétences en navigation dans un environnement 3D (CAO). Une troisième option est également possible, le développeur peut préparer les modèles 3D avec les concepts d'Entity-Feature et laisser ensuite l'expert préparer les scénarios. En plus, des interfaces de configuration proposées dans le chapitre précédent avec les Figure 78 et Figure 79, la mise en place d'un outil

d'édition détaché d'Unity 3D et rendu accessible pour les experts métiers via la mise en place d'une interface user-friendly reste nécessaire.

Que ce soit l'expert et/ou l'équipe de développement qui définissent les Entity-Features, le framework INTERVALES gérant les interactions pour chacun de ces concepts permet ensuite de définir les procédures et scénarios. Une fois les scénarios définis et édités, l'application est validée et utilisée en production (voir Figure 81 - en vert).

Puisque l'application est basée sur Entity / Feature, Action / Relation et Objectifs, les scénarios sont définis de manière compréhensible avec un langage basé UML. Par conséquent, ils peuvent être adaptés pour correspondre à l'évolution des processus industriels. À titre d'exemple sur la figure 81, l'utilisateur (ou opérateur) de l'application RA / RV propose une modification de l'ordonnancement des étapes (Figure 81 – « Suggère des modifications ») et celle-ci est prise en compte par l'expert qui modifie alors le scénario précédemment défini de manière à prendre en compte le changement proposé (Figure 81 – « Modifie les scénarios RA/RV »).

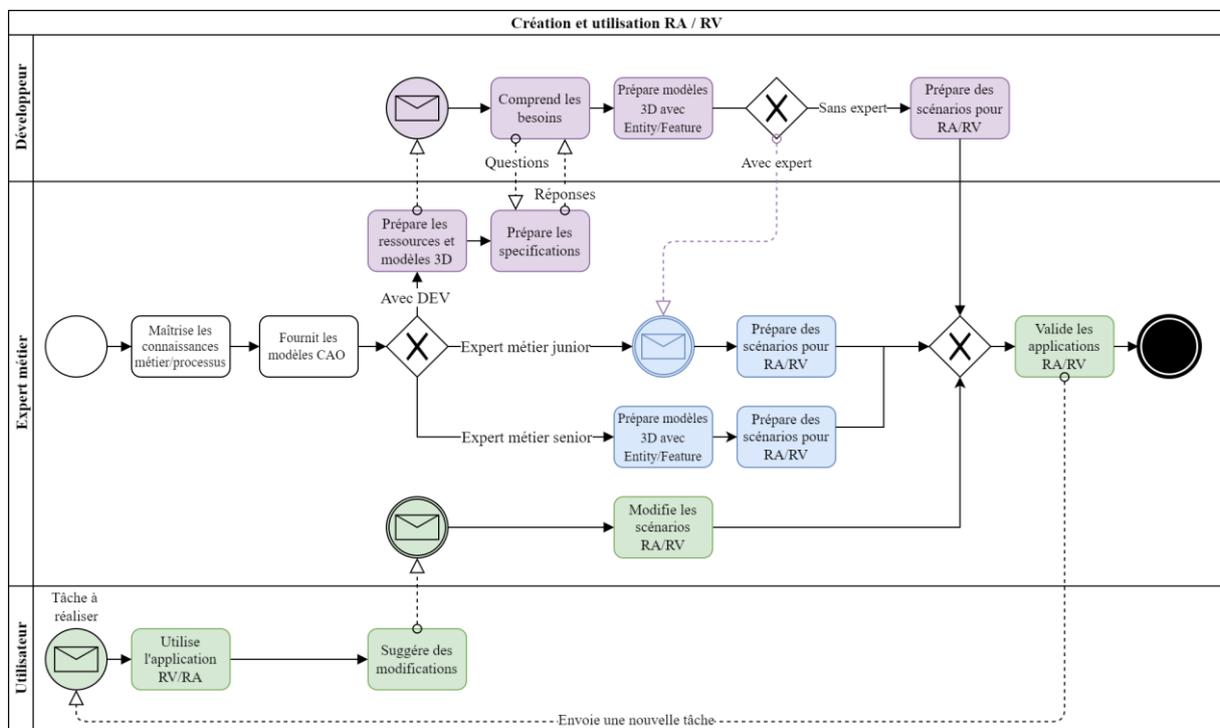


Figure 81: Workflow d'utilisation du framework INTERVALES en BPMN.

4.3 Cas d'usages industriels

Dans cette partie, plusieurs cas d'usages vont être présentés. Nous détaillerons tout d'abord un cas d'usage industriel s'appliquant à des travaux de recherche sur le LeanManufacturing dans la section 4.3.1. Le fonctionnement d'INTERVALES étant très similaire sur les différents cas d'usages, nous le détaillerons principalement dans ce premier cas d'usage. Puis, dans la section 4.3.2, nous présenterons les cas d'usages réalisés au sein de l'entreprise OREKA Ingénierie intégrant soit les concepts d'INTERVALES ou entièrement le framework.

4.3.1 LeanManufacturing VR : Assemblage sur postes manuels

4.3.1.1 Description du cas d'usage

Le cas d'usage présenté ici est composé de postes de travail manuel assistés par un bras cobotique afin d'effectuer une procédure d'assemblage (voir Figure 82). Pour cela, nous nous sommes basés sur l'atelier physique qui est composé de six postes pour l'assemblage et le contrôle qualité de produits (Figure 82 : a). Une cartographie des flux ainsi que le produit assemblé sont visibles dans la Figure 83. Les opérations sur un poste de travail sont assistées par un bras cobotique UR10 pour la manipulation des pièces et/ou un système de réalité augmentée pour l'aide à l'instruction (Figure 82: b). Ensuite, un jumeau numérique intégrant des modèles CAO 3D, un modèle comportemental de bras cobotique, des instructions d'assemblage et un scénario a été développé.

C'est basé sur ce jumeau numérique et sur le modèle INTERVALES, qu'un environnement virtuel interactif et collaboratif et un autre en réalité augmentée ont été mis en place. L'environnement en réalité augmentée est, quant à lui, un outil pour guider l'opérateur dans son processus d'assemblage manuel. Un exemple de l'environnement de l'atelier industriel est présenté dans la figure 82 : c, et une vue CAO du poste de travail cobotique dans la figure 82 : d.

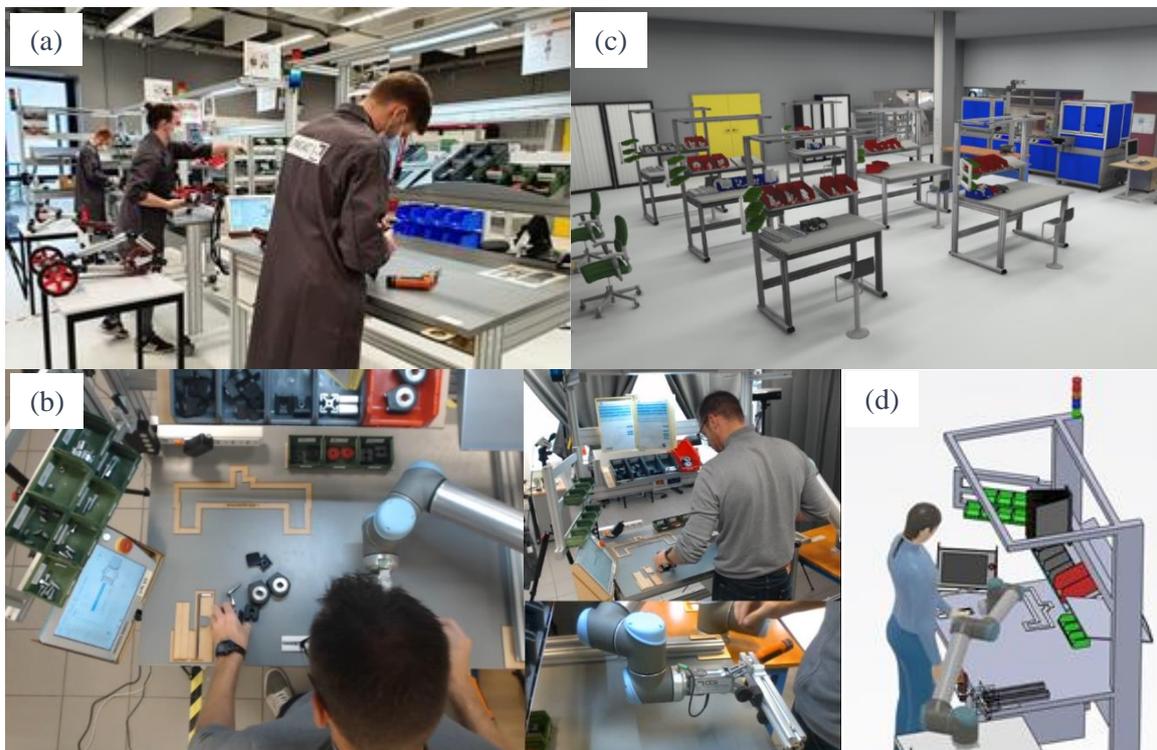


Figure 82: (a) L'atelier industriel. (b) Assemblage sur un poste de travail assisté par un bras robot UR10. (c) Vue de l'atelier industriel en RV. (d) Vue CAO du poste de travail cobotique. Toutes les photos présentées ont été prises au laboratoire de LINEACT CESI.

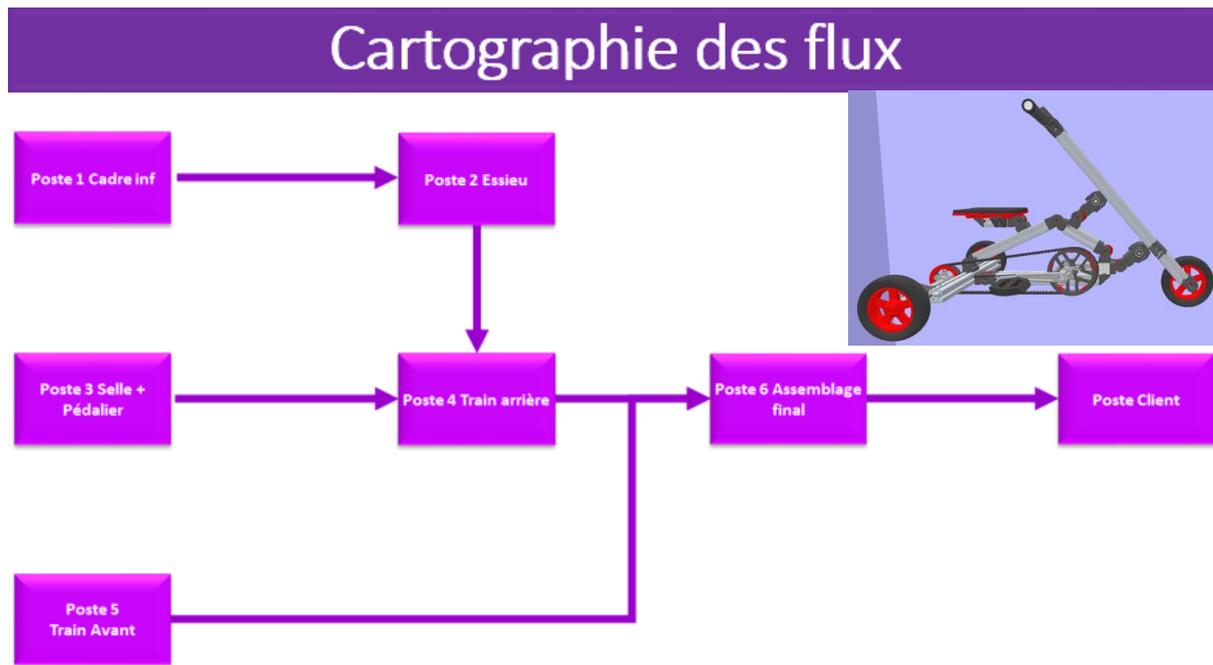


Figure 83 : Cartographie des flux d'assemblage et visuel du produit assemblé.

4.3.1.2 Architecture utilisée avec INTERVALES

Le scénario, détaillé ci-dessus, est illustré par une étude de cas sur une procédure d'assemblage assistée par un cobot en RV et RA (voir Figure 84). Le but de ce cas d'usage est de former sur les principes du Lean Manufacturing (Badets et al., 2020) ou sur la procédure d'assemblage sur des postes manuels assistés par de la cobotique (collaboration avec un robot). Dans ce deuxième cas d'utilisation, l'opérateur procède aux étapes d'assemblage assisté d'un UR10, un bras robotique à 6 degrés de liberté (DoF), qui peut maintenir une pièce pendant que l'opérateur agit dessus. Le scénario est composé de plusieurs phases. Tout d'abord, il y a une phase de préparation, réalisée avec des outils de CAO/FAO (Fabrication Assistée par Ordinateur) sur la conception des postes de travail avec leurs configurations, comprenant aussi la sélection des robots ainsi que la programmation et la création des instructions de montage (voir Figure 84 - Conception).

Ensuite, ces connaissances préparées avec les outils CAO/FAO sont exportées au format INTERVALES, avec leur Entity et Features, et intégrées dans le framework. Ensuite, et principalement grâce au module **INTERVALES Interactions**, l'environnement virtuel créé est utilisé pour valider la conception et l'étude ergonomique du poste de travail en faisant réaliser l'assemblage dans une scène RV par un opérateur (voir Figure 84 - Digital Prototype). De plus, chaque opérateur est ici représenté par une capsule jaune afin de vérifier si lui et le cobot ne se heurtent pas au cours du processus. La conception ou l'évaluation collaborative de l'aménagement de l'atelier est également réalisée dans l'environnement RV. Plusieurs utilisateurs en RV peuvent ajuster et valider la disposition des postes de travail, des zones de stockage et des zones de circulation. Une fois le prototype numérique validé, chaque processus d'assemblage est défini grâce au module de scénarios INTERVALES.

En conséquence, les scénarios de formation RV sont utilisables par les opérateurs débutants sur la base des données INTERVALES exportées des phases précédentes (voir Figure 84 - Production). De plus, sur la base des mêmes données INTERVALES et en utilisant une caméra au-dessus des postes de travail manuels et un écran dédié fixé sur le poste de travail, des instructions RA peuvent être fournies à l'opérateur en temps réel lors de l'assemblage (voir Figure 84 – Assembly assistance in AR) comme vue précédemment. Les instructions RA sont visibles depuis un écran déporté à côté de l'opérateur, tandis que la caméra filme le poste de travail depuis le dessus de l'opérateur. Dans ce cas, on choisit d'utiliser la RA avec l'affichage sur l'écran déporté pour éviter d'équiper et d'encombrer l'opérateur avec des Smart Glasses telles que les Vuzix M300 ou le casque Microsoft HoloLens. Les possibilités d'utiliser une tablette et mobile ont également été écartées, car l'opérateur doit garder les mains libres. Cependant, comme les instructions sont basées sur le modèle INTERVALES, ces nouveaux dispositifs pourraient être intégrés en développant l'application spécifique capable de les lire. Ainsi, INTERVALES couvre la conception ergonomique du poste de travail, la formation au montage avec Collaboration Homme-Robot (HRC), et l'aide au montage en réalité augmentée. La prochaine étape pour cette application serait d'offrir des fonctionnalités de jumeau numérique en temps réel, par exemple en transposant le mouvement des cobots entre les systèmes virtuels et physiques.

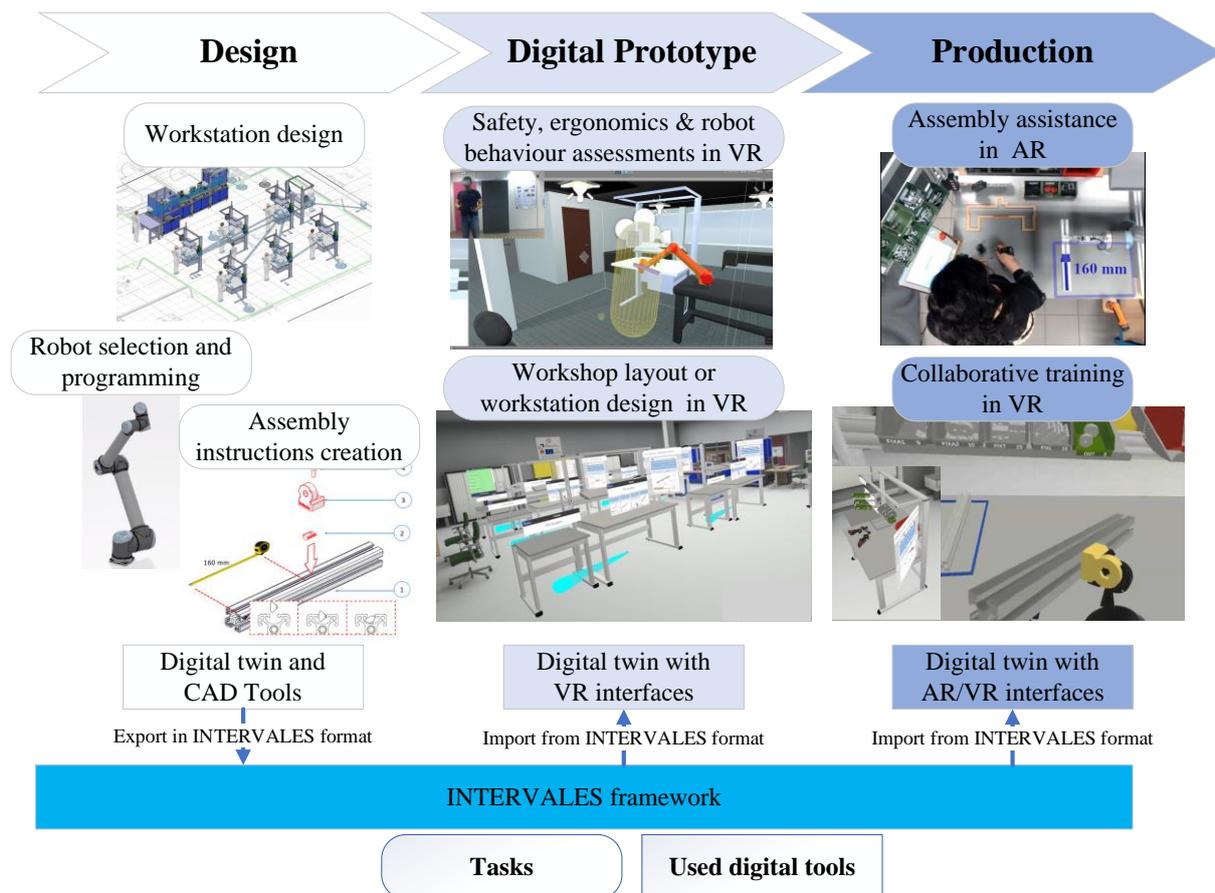


Figure 84 : Procédure d'assemblage assistée par un cobot en RV et en RA avec la conception de postes manuels, le prototype numérique et la production d'une formation d'assemblage basée sur INTERVALES et d'un assistant en RA également basé sur INTERVALES

De ce fait, lors de la phase de conception et de revue du prototype numérique, l'environnement RV est utilisé pour la conception et les évaluations ergonomiques et sécuritaires des postes de travail manuels et cobotiques. La réalité virtuelle collaborative est également utilisée pour la conception de l'aménagement de l'atelier.

Pendant la phase de production, la RV est utilisée à des fins de formation concernant les procédures d'assemblage sur les postes de travail manuels et cobotiques. Un outil RA est utilisé pour guider l'opérateur tout au long du processus d'assemblage. Nous le présenterons dans la section 4.3.1.6. Un jumeau numérique, associé aux données représentant le processus, a été défini une unique fois en suivant le formalisme INTERVALES. Il est utilisé par la suite dans les deux applications : la RV et la RA. Comme les données sont les mêmes pour les deux environnements, la suite décrira principalement comment les scénarios sont intégrés dans l'environnement virtuel.

4.3.1.3 *Création de l'environnement au formalisme INTERVALES*

À partir des modèles CAO et des gammes d'assemblage (mode opératoire) disponibles, une bibliothèque de 150 entités, représentant les pièces nécessaires à l'assemblage du produit, les pièces du poste de travail, les outils et du cobot, a été créée. Une fois leurs Features et leurs Relations définies (Figure 85), il est possible d'assembler librement chacune de ces pièces.

Pour comprendre comment formater une connaissance existante dans une scène INTERVALES, plusieurs étapes doivent être suivies. Dans notre cas d'étude, le processus d'assemblage est d'abord analysé ; chaque pièce requise (ou Entity) et les différentes contraintes (ou Feature) auxquelles il faut prêter attention doivent être définies. Concentrons-nous à titre d'exemple sur le profil de 360 mm, appelé P360 dans la figure 85. Sur cette partie, d'autres pièces telles que FIXA1 peuvent être mises de chaque côté, et les lardons (nommés LARD) peuvent être glissés dans chacune des 4 fentes de la P360. Ainsi, 2 types de Feature « FeatureSnap » et « FeatureSlider » avec leurs propriétés et leurs listes de compatibilités ont été définies (voir Figure 85 : c). Par exemple, pour le slider, une attention particulière doit être accordée à la distance du point de départ à laquelle le lardon doit être placé. Ainsi, cette propriété, appelée DistanceFromPointA, est incluse dans la classe « FeatureSlider ».

4.3.1.4 *Édition de la procédure d'assemblage par un développeur*

À partir de là, une procédure est définie par le développeur, en interaction avec l'expert métier (Figure 81 – « Préparation des spécifications »), afin de respecter le mode opératoire visible sur la figure 85 - A. Par conséquent, 3 objectifs (contenant chacun une tâche) sont programmés. Le premier est TaskSlider entre un P360 et un LARD avec une *DistanceFromPointA="160mm"* et une *Tolerance=±2*. La deuxième tâche est une TaskSnapOriented entre FIXA1 et LARD avec un *SnapOrientedAngle = 0* et une *Tolerance = 0.5*. Enfin, la troisième tâche est une TaskScrew entre B820 et LARD sans paramètre ajouté.

Du point de vue de la création, cela consiste en l'importation du modèle de chaque pièce (Figure 85 - B), au placement des Features décrites précédemment (Figure 85 - C) et à leur configuration en utilisant l'interface utilisateur (Figure 85 - D). Ces données seront ensuite converties en

données INTERVALES au format JSON qui pourront être sauvegardées ou rechargées par la suite (Figure 85 - E). Le Tableau 3 résume les features qui ont été mises en place pour ce cas d'usage.

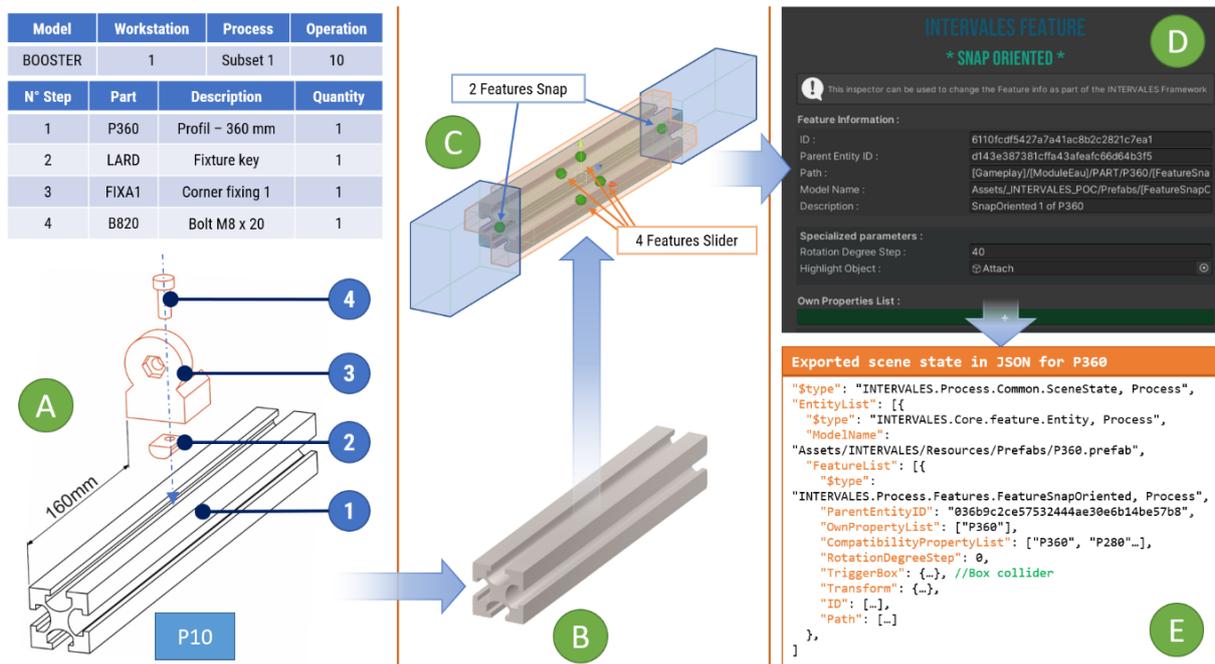


Figure 85: Placement des Features sur P360 en se basant sur la feuille d'instructions. (A) : La feuille d'instructions fournie ; (B) : Le modèle 3D de P360 ; (C) : Placement des Features dans Unity ; (D) : Configuration de l'inspecteur de la Feature SnapOriented dans Unity3D. (E) : L'état de la scène INTERVALES en JSON.

Tableau 3 : Liste des features nécessaires à la réalisation de l'assemblage présenté et leur description associée.

Feature	Description
FeatureSlider	Permet de glisser une Entity dans une autre sur une distance donnée.
FeatureSnapOriented	Permet d'accrocher deux Entity compatibles entre elles avec un degré de liberté en rotation.
FeatureScrew	Permet de visser deux Entity compatibles entre elles.

4.3.1.5 Validation de la procédure d'assemblage par le ScenarioManager en réalité virtuelle

Enfin, au lancement de l'application RV, les interactions sont gérées par le module d'interaction INTERVALES, qui permet aux utilisateurs d'interagir librement avec les Entities. En interagissant, chaque **Maker** envoie les événements d'Action et Relation fait par l'utilisateur. De cette manière, le **ScenarioManager** traite ces événements et vérifie si les Objectifs sont atteints ou non. Par exemple, la figure 86 présente les premières étapes attendues de l'assemblage. L'utilisateur doit prendre une pièce P360 (Figure 86 : 1) et effectuer une SlideRelation entre le P360 et le LARD (Figure 86 : 2) à 160 mm attendus du PointA avec une tolérance de 5 mm (Figure 86 : 3).



Figure 86 : Exemple de scénario au format de données INTERVALES en JSON

4.3.1.6 Mise en place de la réalité augmentée

En ce qui concerne la réalité augmentée, les tâches définies dans le scénario sont alors réutilisées comme un système de guidage. Afin de prendre en charge la réalité augmentée, une Feature « **FeatureARTracking** » est définie de manière à reconnaître le P360 à partir d'un marqueur. Dans le moteur Unity, les modèles 3D fournis sont importés, puis les entités et FeatureARTracking sont positionnées aux bons endroits (Figure 87).

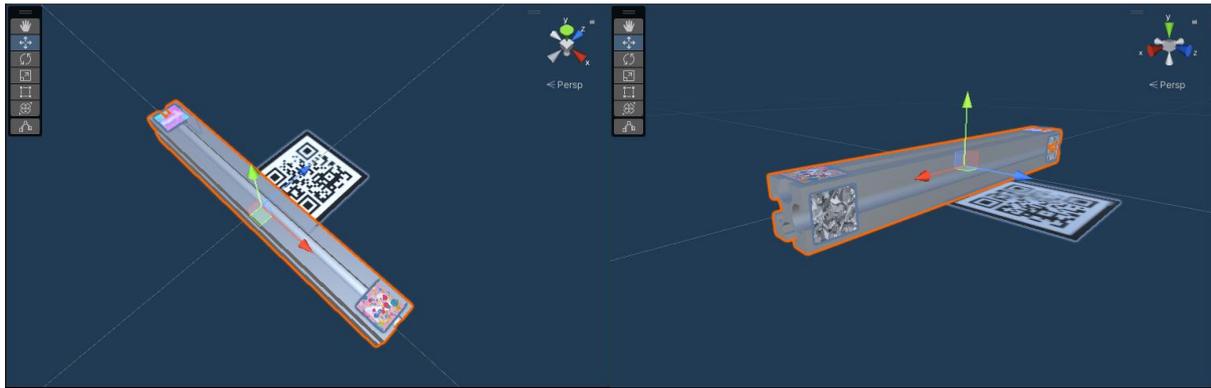


Figure 87 : Positionnement de FeatureARTracking sur l'entity P360 au sein du moteur Unity 3D.

Chaque Entity disposant d'une FeatureARTracking peut donc être utilisée pour repérer la pièce réelle à reconnaître pour faire de la réalité augmentée. À partir de la procédure définie ci-dessus, il est possible d'afficher la pièce correspondante à utiliser lors de la tâche à effectuer ainsi que des informations supplémentaires comme l'outil à utiliser ou la position souhaitée. Par exemple, lorsque l'opérateur qui effectue l'assemblage atteint une étape correspondant à l'assemblage d'un FIXA1 et d'un P360, le système RA affiche une superposition du FIXA1 virtuel sur le vrai P360 pour indiquer où il doit être placé (Figure 88). L'exemple fourni ci-dessous est une preuve de concept qui utilise un suivi par marqueur avec un smartphone déporté. Les perspectives sont d'utiliser un écran déporté ou un casque HoloLens pour afficher ce contenu en RA. Il serait aussi possible d'utiliser des algorithmes basés modèles 3D et de définir une FeatureARModel3D pour permettre la reconnaissance par modèle 3D.



Figure 88 : Preuve de concept du guidage à l'assemblage en réalité augmentée.

4.3.1.7 Synthèse

Grâce aux concepts proposés dans le framework INTERVALES, nous avons pu créer cet environnement virtuel / augmenté, avec les interactions réalisables et un scénario suivant une feuille d'instructions. Le cas d'usage industriel présenté est collaboratif et les actions peuvent être réalisées à la fois par n'importe quel utilisateur ou par des agents autonomes, ici le bras robotique. Et comme l'environnement est basé sur des données INTERVALES, il peut être

facilement capturé. Il est ainsi possible pour le formateur d'arrêter la procédure d'assemblage puis de la recharger là où l'utilisateur s'est arrêté.

Pour ce cas d'usages, nous avons présenté un cas d'édition fait par un développeur informatique. Un outil d'édition permettant aux experts métiers d'éditer un scénario sur ce même cas d'usage sera présenté dans le chapitre 5.

4.3.2 Mises en application dans les projets de l'entreprise

Cette thèse étant dans le cadre d'une convention CIFRE, les concepts définis ainsi que le framework INTERVALES ont pu être intégrés et éprouvés dans différents cas d'usages et projets de l'entreprise au cours de la thèse. Nous traiterons deux types de cas d'usages :

- L'intégration d'INTERVALES au sein d'une application déjà existante,
- L'intégration d'INTERVALES dès le début du processus de conception.

Ces projets seront décrits dans les sections suivantes.

4.3.2.1 Tutoriel OREKA VR : Escape Game d'initiation à la réalité virtuelle

Cette application est le premier projet que nous avons utilisé comme cas d'usages au sein de l'entreprise. L'objectif principal de l'application est d'initier des personnes à la réalité virtuelle et les habituer aux interactions spécifiques de cet environnement. Il s'agit d'un tutoriel sous la forme d'un « escape game » dans un environnement industriel (Figure 89) avec différents ateliers.

L'application étant déjà existante, il s'agissait de mieux intégrer le framework INTERVALES au processus de conception d'applications de l'entreprise. L'objectif de cette preuve de concept était de :

- Voir comment intégrer INTERVALES à une application existante pour permettre de récupérer les actions faites au sein d'une scène et de poser / développer les concepts du modèle INTERVALES dans Unity suivant les besoins de l'application. Par exemple, de nouvelles interactions liées à de nouvelles Features à mettre en place.
- Être capable de convertir la scène Unity en un fichier textuel JSON (Éléments présents, interactions possibles, etc ...) dans le but de pouvoir sauvegarder et recharger l'état de la scène à un instant t .
- Valider le scénario de l'application et pouvoir le réaliser.
- Tester les premiers concepts d'édition par démonstration « authoring-by-doing » détaillés plus tard dans le chapitre 5.

Ainsi, en se basant sur le scénario original de l'application, nous avons pu définir des scénarios de base afin de tester la robustesse du moteur de scénario par la même occasion. Dans la figure 89 à droite, nous pouvons voir l'atelier « Remplissage d'un container d'eau ». Ce scénario consiste à placer correctement les tuyaux manquants pour compléter la tuyauterie, ouvrir des vannes et pour finir, pomper l'eau jusqu'à remplir le container. Nous avons ainsi pu valider le moteur de scénario pour la mise en place de scénario parallèle (Ex : Compléter la tuyauterie) et étape par étape (Ex : Ouverture des vannes). Nous avons également pu valider la sauvegarde / restauration de l'environnement via cette preuve de concept.

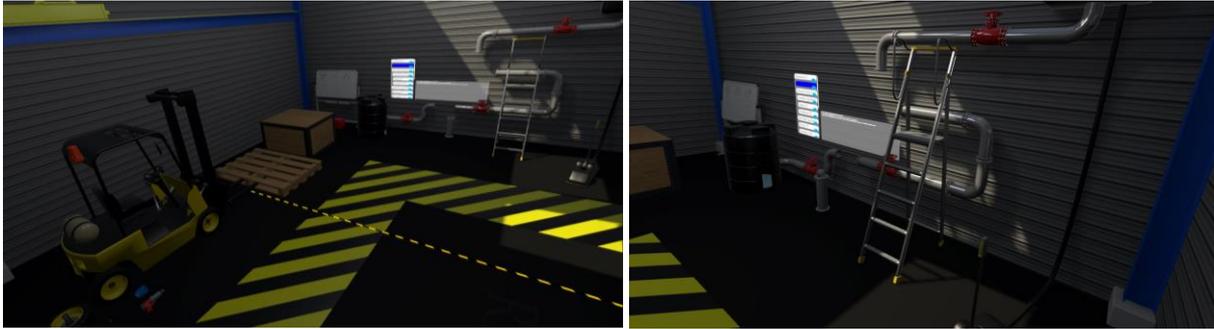


Figure 89 : Environnement virtuel de l'application « Tutoriel OREKA VR ».

Ainsi, ce premier cas d'usage au sein de l'entreprise nous a permis de tester si INTERVALES s'intégrait facilement à un projet existant. Nous avons, par exemple, dû ajouter une nouvelle feature pour l'ouverture des vannes que nous avons appelée `FeatureRotativeSwitch`. Cette feature permet d'envoyer un évènement d'activation / désactivation suivant le dépassement d'un angle minimum et maximum donné. Nous avons alors pu observer des axes d'améliorations concernant l'ajout de nouvelles features. En effet, la procédure d'ajout d'une nouvelle feature nécessite un processus manuel source d'erreurs et peut donc être améliorée, voire automatisée.

4.3.2.2 Salle virtuelle pour la formation aux mesures en physique nucléaire

Pour ce projet client, le CNAM souhaite virtualiser une salle de mesures nucléaires réelle dans laquelle ses étudiants réalisent des travaux pratiques de mesures de sources radioactives émettrices. Cette salle virtuelle devait permettre la réalisation de plusieurs travaux pratiques dans un même environnement en utilisant le clavier et la souris. C'est alors le choix d'une simulation (scenario ouvert) qui a été fait (voir Figure 90). C'est-à-dire que suivant les équipements de mesures et les sources branchés, l'environnement et les résultats de mesures devaient réagir en conséquence et simuler un comportement similaire à la réalité.



Figure 90 : Environnement virtuel de la formation aux mesures en physique nucléaire.

Plusieurs concepts d'INTERVALES ont été utilisés pour la réalisation de cette application, notamment sur la notion d'entité et d'actions génériques (Exemple dans la figure 91 : Attraper, Placer, Brancher, Configurer). Comme pour les actions INTERVALES, une action ne dépend pas de l'entité sur laquelle elle est effectuée ; celle-ci est simplement représentée par un paramètre de l'action. Le modèle INTERVALES a également été utilisé pour les réflexions autour du scénario ouvert car ses concepts permettent d'ajouter des connaissances métiers au sein de l'application.



Figure 91 : Exemple d'actions génériques mis en place dans le projet.

4.3.2.3 Application de formation à l'assemblage d'un équipement en réalité virtuelle

Dans ce projet, l'entreprise SNEF souhaite représenter les nouvelles technologies RA/RV sur une ligne d'assemblage automatique avec un jumeau numérique pour notamment, former des opérateurs à des opérations de maintenance et d'assemblage sur poste manuel en réalité virtuelle. L'opérateur devra réaliser le montage sur un poste manuel d'un équipement de type « écran de contrôle » constitué d'un écran, d'une carte mère « Arduino », d'un carter, de vis et de colonnes de fixation (voir Figure 92 – à gauche).

Le cas d'usage étant similaire à celui présenté dans la section 4.3.1, puisqu'il s'agit également d'un assemblage sur poste manuel, le choix d'utiliser le framework INTERVALES pour son développement a pu facilement être réalisé sans ajout de nouvelles features. Il a permis de facilement mettre en place les interactions d'assemblages entre les Entities/Features et de définir la procédure de montage en paramétrant les Actions / Relations possibles (Figure 92).

L'usage d'INTERVALES permet également de proposer une grande valeur ajoutée comme la sauvegarde et le chargement de la scène à une étape souhaitée de l'assemblage via un tableau de bord ou encore la possibilité de récupérer l'historique des actions réalisées par l'opérateur. Par ailleurs, l'assemblage nécessitant l'usage d'une visseuse pour certaines étapes, cela nous a amenés à faire évoluer INTERVALES afin d'intégrer la notion d'outil (voir Figure 92 – à droite).

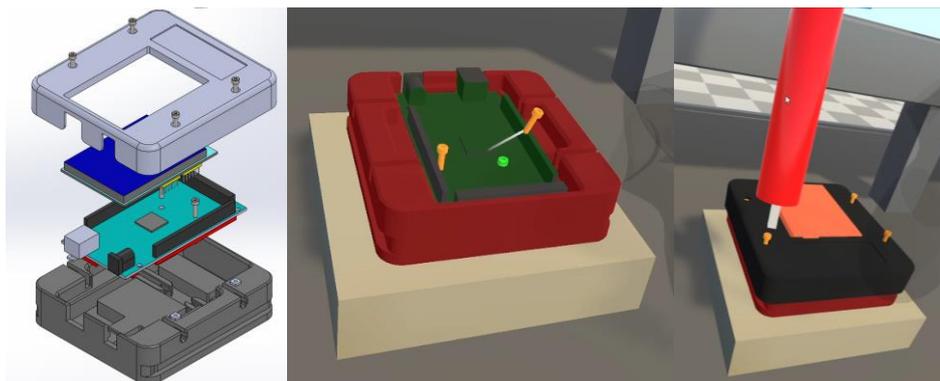


Figure 92 : (à gauche) Montage de l'écran de contrôle à réaliser. (au milieu) Placement d'une vis durant l'assemblage de l'équipement. (à droite) Utilisation d'un outil (visseuse) pour l'assemblage.

4.3.3 Synthèse

Nous avons présenté dans ce chapitre comment pouvait être utilisé INTERVALES dans un processus de conception via une proposition de workflow au format BPMN avec les différents acteurs : experts métiers, développeurs et opérateurs. Ensuite, plusieurs cas d'usages industriels ont été détaillés, qu'ils s'appliquent à la recherche, à des projets industriels ou encore à des

besoins dans le domaine de la formation. Le tableau suivant (Tableau 4) synthétise ces cas d'usages en reprenant les caractéristiques identifiées dans l'état de l'art et appliquées à ces derniers afin de démontrer les concepts pris en compte par INTERVALES.

Tableau 4 : Synthèse des caractéristiques INTERVALES appliquées aux cas d'usages présentés.

Cas d'usages	Technologie	Scénario	Scénario a/	Scénario	Collaborative?	Agent	Causalité ?
LeanManufacturing							
Conception de l'agencement d'un atelier	RV			X	X		
Évaluation de la conception et de l'ergonomie des postes de travail ; Évaluation de la sécurité et du comportement du robot				X	X	X	
Formation au montage		X	X		X	X	X
Assistance au montage	RA	X				X	
OREKA Ingénierie							
Tutoriel OREKA VR : Escape Game d'initiation à la réalité virtuelle	RV	X	X				X
Application de formation à l'assemblage d'un équipement		X	X				X
Salle virtuelle pour la formation aux mesures en physique nucléaire	PC			X			X

L'usage des concepts proposés par le framework INTERVALES et des développements réalisés a alors permis la mise en place d'environnements virtuels et augmentés, la définition des interactions possibles et la création de scénarios dans différents contextes industriels. Nous avons aussi montré qu'INTERVALES commence déjà à s'intégrer au processus de développement de l'entreprise et permet d'ores et déjà une plus-value, notamment en termes de fonctionnalités proposées. Sur le projet SNEF, par exemple, l'intégration d'INTERVALES a permis d'avoir accès aux fonctionnalités rendues possibles comme la sauvegarde / restauration d'un état de la scène ou encore l'accès à l'historique des actions réalisées. De plus, les features et interactions ainsi créées dans le cadre de ces projets permettent également de fournir une bibliothèque réutilisable par l'entreprise pour de futurs projets (Ex : FeatureRotativeSwitch pour l'ouverture des vannes du projet Tutoriel OREKA). Cependant la définition de scénario reste encore accessible uniquement à des experts informatiques car elle nécessite de passer par la formalisation des données en JSON comme nous pouvons le voir dans la figure 86.

Toutefois, le modèle UML d'INTERVALES fournit un cadre de formalisation permettant d'utiliser ou de créer des outils générant des procédures au format INTERVALES. Par conséquent, nous avons travaillé sur ce verrou et nous proposons et évaluons dans le chapitre 5 des interfaces d'éditeurs pour définir / modifier des scénarios.

5 INTERVALES : Édition de scénarios et de procédures métiers

5.1 Introduction

Dans ce dernier chapitre, nous aborderons l'édition de scénarios et de procédures métiers. Nous commencerons par préciser les besoins et objectifs de l'édition de scénarios dans une première partie 5.2. Nous y présenterons également un premier prototype d'éditeur de scénario, puis, nous détaillerons la mise en place d'un autre moyen d'édition appelé « Authoring-by-doing » dans la section 5.3. Pour finir, nous comparerons l'édition de scénario avec deux outils d'éditions à travers une expérimentation dans la section 5.5, l'un via une interface d'édition 2D et l'autre en réalité virtuelle via l'authoring-by-doing.

5.2 Édition de scénarios

5.2.1 Définition des besoins

Nous avons présenté, dans les chapitres précédents, le modèle INTERVALES et comment un scénario industriel pouvait être formalisé ainsi que différents cas d'usages. Cependant, le développement de moyen d'édition est tout aussi important dans notre objectif de simplifier et de rendre accessible aux experts métiers la création / l'édition de scénario. En effet, l'éditeur doit :

- Pouvoir prendre en compte les ressources disponibles dans l'environnement virtuel (Entity/Feature, type d'actions possibles).
- Permettre à un expert métier n'ayant pas de connaissance en développement informatique, d'éditer un scénario.
- Offrir une interface ou un usage clair.

5.2.2 Premiers développements réalisés

C'est dans ce contexte que nous avons développé un premier prototype d'éditeur de scénario (Figure 93). Cet éditeur a été développé en C# et est intégré seulement à Unity. Ce dernier n'est donc pas accessible en runtime, c'est-à-dire dans l'application exploitable par l'utilisateur final. Il se compose d'une interface utilisateur de type NodeGraph reprenant notre première formalisation, présentée dans la section 3.3.2.2, utilisant des nœuds et groupes de nœuds, respectivement UnitTask et GroupTask. Les utilisateurs ont accès à toutes les actions disponibles telles que, pour ce cas d'utilisation d'assemblage, " Pick component " (Prendre un composant), "RelationSlider" ou "RelationSnapOriented" et peuvent les configurer en conséquence. Par exemple, sur une action "Pick component", l'utilisateur peut sélectionner le

type d'entity à l'aide d'une liste déroulante et nommer la pièce. Ensuite, une liste affiche les Entités actuellement utilisées dans le scénario (Figure 93 : A). Toutes les tâches de l'interface graphique affichent un aperçu de l'entité concernée (Figure 93 : B) ou de l'emplacement de la fonctionnalité (Figure 93 : C). Pour ajouter une nouvelle tâche, l'utilisateur peut soit cliquer avec le bouton droit de la souris, soit cliquer sur la tâche précédente « out », puis la rechercher à l'aide de la barre de recherche (Figure 93 : D).

En conséquence, l'expert informatique ou métier peut éditer un scénario INTERVALES via l'interface utilisateur en définissant des tâches et en définissant ses paramètres. A titre d'exemple, les instructions d'assemblage sont présentées dans la figure 94. L'éditeur permet alors de convertir une fiche d'instruction en scénario dans INTERVALES, comme illustré à la figure 93. La procédure métier éditée correspond aux étapes suivantes : prenez une pièce P360 et une pièce LARD (Figure 93 : 1), effectuez une SlideRelation entre le P360 et le LARD (Figure 93 : 2) à 160 mm attendus du PointA avec une tolérance de 5 mm (Figure 93 : 3).

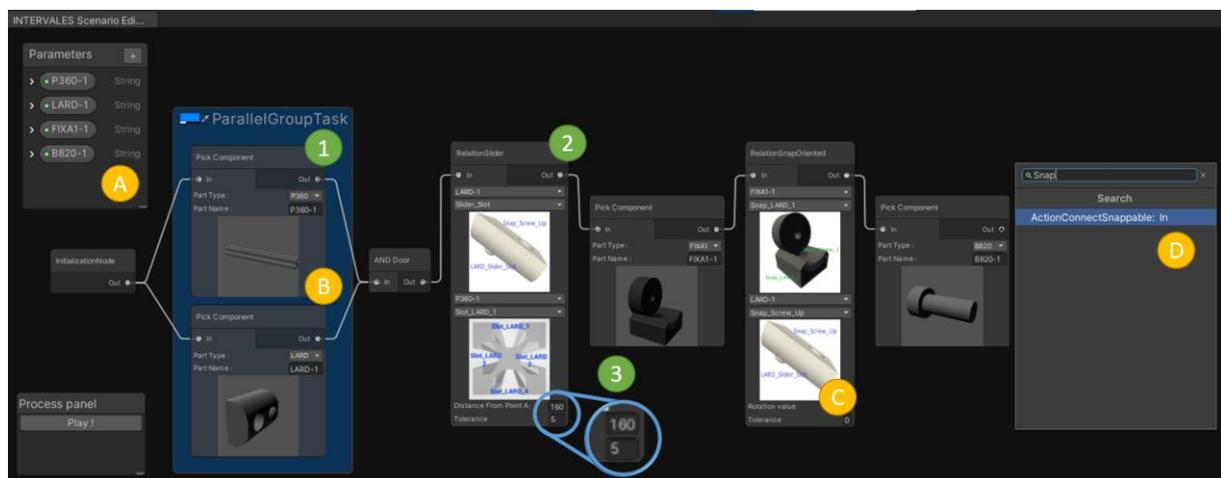


Figure 93 : Interface graphique du premier prototype d'éditeur de scénario INTERVALES

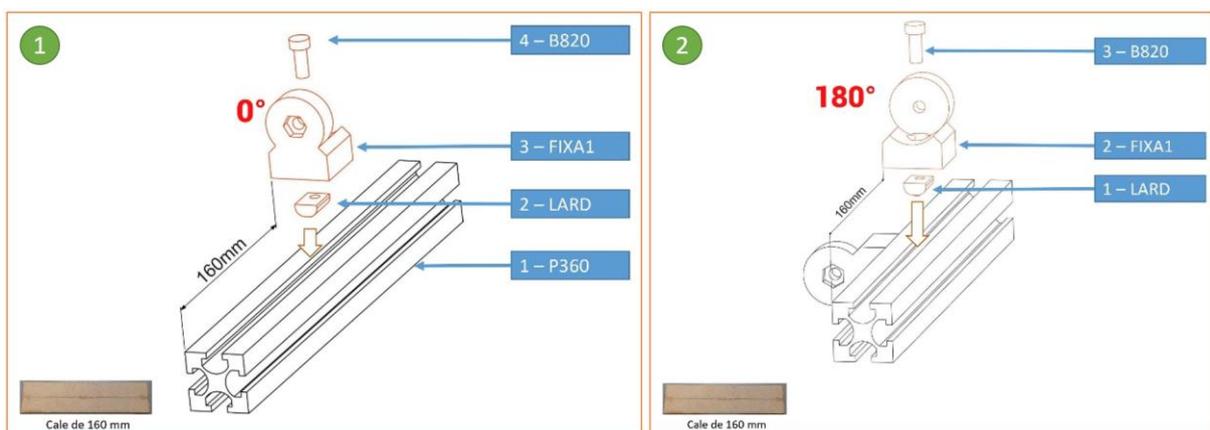


Figure 94 : Série d'opérations utilisées pour l'expérience

5.2.3 Étude préliminaire

Dans le but de recueillir des premiers retours utilisateurs et afin d'évaluer si le framework INTERVALES et cette première interface utilisateur de création de scénarios peuvent être utilisés aussi bien par des experts informatiques que par des experts métiers, sans compétences en développement, nous avons réalisé une expérimentation auprès de 11 participants (5 développeurs et 6 experts métiers sans compétences en développement) de 25 à 53 ans. Pour ce faire, ils devaient créer le scénario correspondant à une série d'opérations, illustrées à la figure 94, à l'aide de l'interface INTERVALES.

L'expérimentation se déroule comme suit : après une vidéo d'introduction expliquant le contexte et l'utilisation de l'interface, chacun a eu une phase libre pour s'exercer avec l'éditeur INTERVALES, suivie d'une phase en autonomie pour réaliser l'expérience avec leurs temps de réalisation qui ont été enregistrés. Pour finir, ils devaient répondre à un questionnaire composé d'un SUS (System-Usability-Scale (Lewis, 2018)) et d'un UEQ (User Experience Questionnaire⁸) (Laugwitz et al., 2008). Nous avons choisi ces questionnaires (voir Annexes 10.3 et 10.4) car le System-Usability-Scale fournit un outil fiable pour mesurer l'utilisabilité avec un score de 0 à 100 et l'UEQ, pour mesurer l'expérience utilisateur en utilisant 26 paires d'adjectifs opposés (ex : compliqué/facile).

La figure 95 présente le score SUS (à gauche) et le temps de réalisation (à droite) selon le profil des participants (développeur DEV et expert métier NDEV). Pour tester si les résultats sont statistiquement différents, nous avons effectué un test de Student (t-test). Premièrement, l'homogénéité des variances est testée avec le test de Levene. Les résultats renvoient une p-value de $0,395 > 0,05$ pour les scores SUS et une p-value de $0,493 > 0,05$ pour les temps d'exécution. Ensuite, la distribution de normalité des données est testée pour les deux groupes à l'aide des tests de Shapiro. Concernant le score de satisfaction, les résultats donnent une p-value de $0,317 > 0,05$ pour les développeurs et une p-value de $0,329 > 0,05$. Pour l'analyse du temps, il renvoie une p-value de $0,124 > 0,05$ et une p-value de $0,117 > 0,05$.

Étant donné que les tests d'homogénéité et de normalité sont tous deux validés, nous avons pu exécuter le t-test et les résultats du t-test fournissent une p-value de $0,079 > 0,05$ pour le score SUS et une p-value de $0,893 > 0,05$ concernant les délais d'exécution. Comme le t-test donne une p-value $> 0,05$, nous pouvons conclure que le temps de réalisation et le score SUS ne dépendent pas significativement de la catégorie des participants : experts en informatique ou experts métiers (non-développeurs).

C'est pourquoi nous pouvons conclure que l'éditeur INTERVALES permet aux informaticiens ou experts métier de créer un scénario dans une durée équivalente, soit en moyenne **522,5 secondes**. En effet, les durées moyennes pour réaliser les deux séries d'opérations sont similaires (Figure 95 – à droite).

⁸ <https://www.ueq-online.org/>

Concernant le score SUS, même s'il n'y a pas de différences statistiquement significatives, nous pouvons constater un écart entre les moyennes pour les informaticiens et les non-développeurs avec respectivement un SUS de **70,5** et **83,8**. Cela peut s'expliquer par le fait que les développeurs sont plus critiques puisqu'ils utilisent couramment des interfaces similaires alors que les experts métier ont montré un bon intérêt pour l'utilisation de cette interface et la possibilité d'éditer des scénarios avec des blocs visuels. Pour illustrer ce propos, nous pouvons citer un expert métier qui donne comme retour que l'éditeur est « très claire, efficace et user-friendly » tandis qu'un autre participant ayant un profil de développeur a donné des retours critiques et techniques sur l'interface notamment comme par exemple que « si une relation SnapOriented est défini après un « pick » d'un objet, alors la relation devrait être déterminé ». Et effectivement, l'analyse des données de l'UEQ confirme que les experts informatiques montrent moins d'attractivité que les experts métiers avec respectivement une moyenne (de -3 à 3) de *0,7* et *1,89* comme montré dans la figure 96.

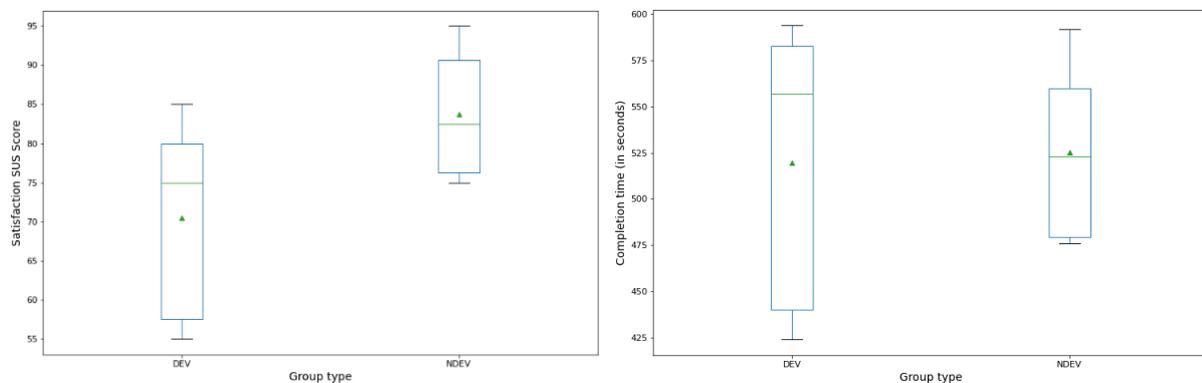


Figure 95: (À gauche) Boîte à moustaches du score SUS en pourcentage selon le type de groupe. (À droite) Boîte à moustaches des temps de réalisation selon le type de groupe.

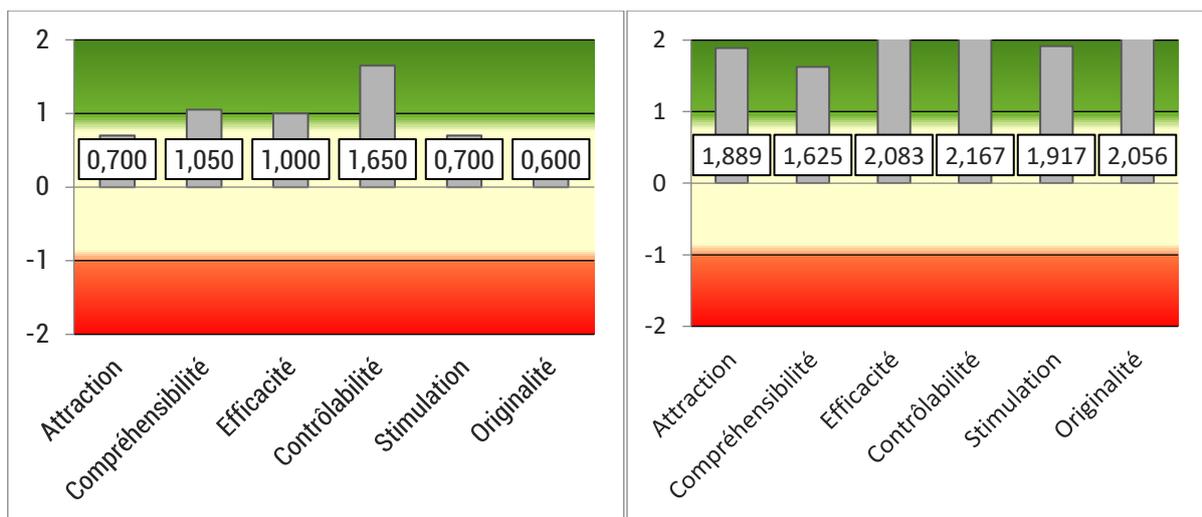


Figure 96 : Résultats fournis par l'outil d'analyse UEQ avec à gauche les développeurs, et à droite, les experts métiers.

5.2.4 Synthèse

Les résultats de l'expérience montrent qu'il existe un engouement pour ce type d'interface utilisateur d'autant plus qu'il est déjà possible pour un informaticien ou un expert métier de mettre

en place un scénario aussi rapidement et avec un même temps de pratique de cette interface. Cependant, cette interface ayant été réalisée directement pour l'éditeur Unity, elle n'était pas utilisable directement au sein d'applications RV/RA. Par ailleurs, nous souhaitons également étendre l'édition en ajoutant la possibilité de créer un scénario en le faisant une première fois en réalité virtuelle. Il s'agit de l'authoring-by-doing et nous verrons comment ce moyen d'édition a été intégré à INTERVALES dans la section suivante.

5.3 INTERVALES : vers un outil d'édition par démonstration (Authoring-by-doing)

Nous nous intéressons ici au gestionnaire d'événements et au processus de création authoring-by-doing. La proposition décrite dans cette section concerne le module INTERVALES Interactions présenté dans la figure 59 de la section 3.2.1.

L'objectif principal de cette évolution est de rendre accessible les données d'événement présentes à l'intérieur du gestionnaire d'interaction et cela afin de les réutiliser pour la création de scénarios. De plus, cela rendra le module d'interaction, présenté dans le chapitre 3.2, interopérable avec les événements d'interaction des bibliothèques externes ou d'autres cadres existants comme (Lanquepin et al., 2013).

Les deux composants principaux de cette proposition sont le gestionnaire d'événements (Figure 97 – EventManager) et le gestionnaire d'interactions (Figure 97 – InteractionManager). Le premier reçoit tous les événements d'Action/Relation effectués dans la scène. Cela lui permet, tout d'abord, d'enregistrer tous les événements d'Action/Relation effectués dans la scène, que ce soit par un utilisateur ou par un agent autonome. Par ailleurs, l'EventManager relaie ces événements d'Action/Relation auprès des Makers qui sont en charge de représenter visuellement la conséquence de l'Action/Relation (Figure 97 – ActionNewEntityMaker ou ActionFeatureSnapOrientedMaker). En ce qui concerne le gestionnaire d'interaction (InteractionManager), il est en charge de transformer les interactions faites avec les contrôleurs : clavier, souris, ou contrôleur RV, en un événement d'Action/Relation. De cette manière, il peut gérer des comportements développés tels que saisir ou déposer par exemple (Figure 97 – InteractionManager) ou alors il peut s'adapter à des bibliothèques existantes comme VRTK (Figure 97 - InteractionManagerVRTK).

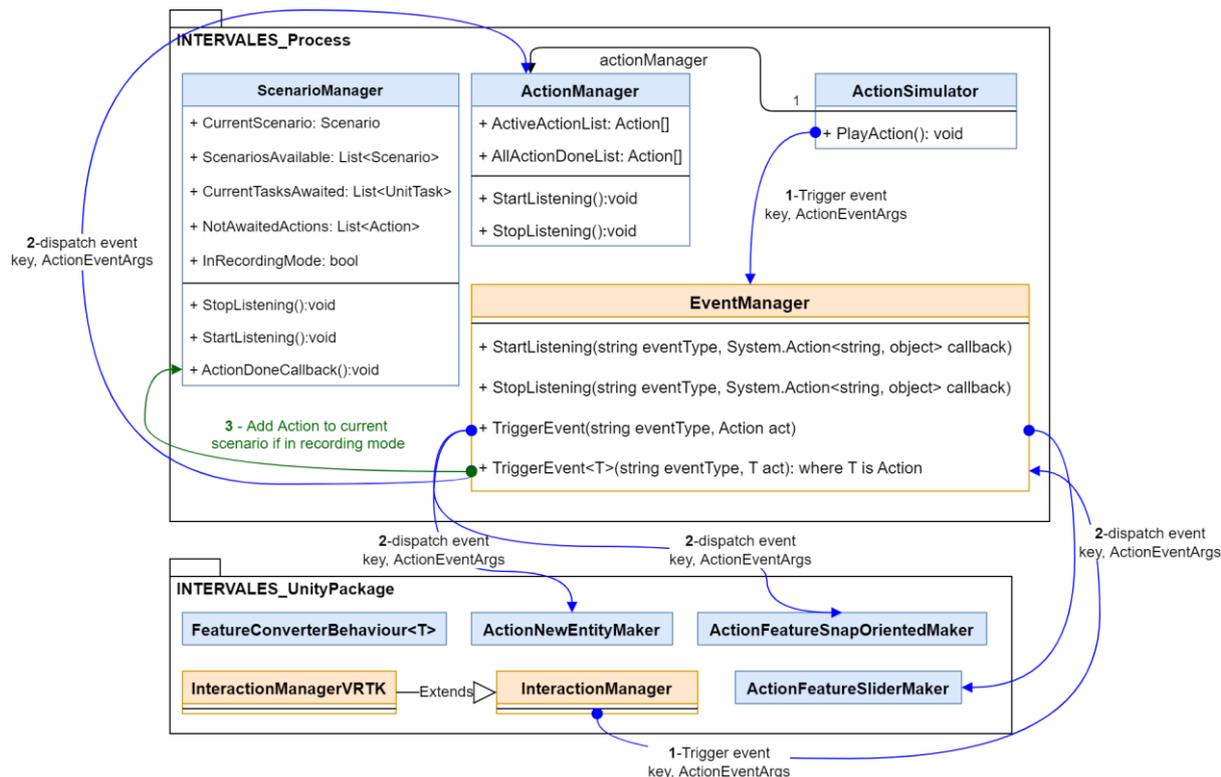


Figure 97 : Workflow d'événement proposé.

Prenons le cas d'un scénario d'assemblage où un opérateur doit visser une vis dans un écrou. À l'aide de VRTK⁹, les développeurs peuvent utiliser les interactions déjà disponibles dans ce SDK dans le moteur Unity ; cela leur permet de détecter les collisions entre eux ces deux modèles vis et écrou, puis, via un connecteur, de notifier INTERVALES InteractionManager qui transformera l'interaction en un événement pour l'EventManager.

Le framework traite les données d'événement, ici si les deux pièces peuvent faire correspondre les listes de compatibilités des features attachées, et va définir une action résultante (ActionManager). Dans ce cas, une relation "Visser" a été effectuée. Ensuite, l'EventManager est informé de cette action et envoie l'événement via le Maker correspondant.

Concernant l'authoring-by-doing, si l'application est en mode enregistrement, la tâche correspondante est alors ajoutée au scénario en cours de création (Figure 97 – événement vers ScenarioManager en vert). Pour finir, le Maker déplacera et positionnera la vis à l'intérieur de l'écrou pour représenter l'action de la vis.

Ainsi, nous avons présenté le fonctionnement de l'interaction manager et de l'event manager mis en place dans le framework INTERVALES et qu'ils permettent d'enregistrer un scénario directement en effectuant les actions dans l'environnement virtuel avec l'authoring-by-doing. Dans la suite de ce chapitre et pour illustrer le processus d'authoring by doing, nous utiliserons le cas d'usage présenté dans la section 4.3.1 avec le scénario d'assemblage suivant (Figure 98).

⁹ <https://vrtoolkit.readme.io/>

Model	Workstation	Process	Operation
BOOSTER	1	Subset 1	10
N° Step	Part	Description	Quantity
1	P360	Profil – 360 mm	1
2	LARD	Fixture key	1
3	FIXA1	Corner fixing 1	1
4	B820	Bolt M8 x 20	1

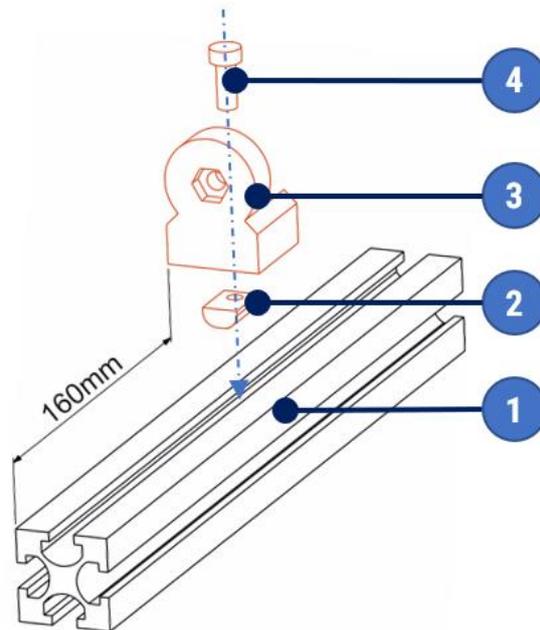


Figure 98 : Exemple de fiche d'instructions d'un poste de travail

L'expert exécute l'environnement virtuel en mode enregistrement et effectue l'assemblage en suivant la fiche d'instructions pour créer le scénario. Il prend un profil de 360 mm, glisse une clé de fixation et enclenche une fixation d'angle dessus (Figure 99 – A). Ainsi, à chaque action effectuée par l'expert métier dans l'environnement RV, un événement est déclenché et l'action est enregistrée comme une action réalisée et une tâche équivalente à l'action effectuée est ajoutée dans le scénario (Figure 99 – B). Dans l'éditeur authoring-by-doing, l'ordre d'exécution est utilisé pour créer la procédure, ce qui donne dans notre exemple (Take = ActionNewEntity ; Slide = RelationSlider ; Snap = RelationSnapOriented) avec les entités concernées et les références des fonctionnalités. Ensuite, l'action correspondante peut être convertie en une nouvelle tâche dans un scénario à l'aide du gestionnaire d'événements. Au final, un fichier JSON contenant les tâches peut être exporté (Figure 99 – C).

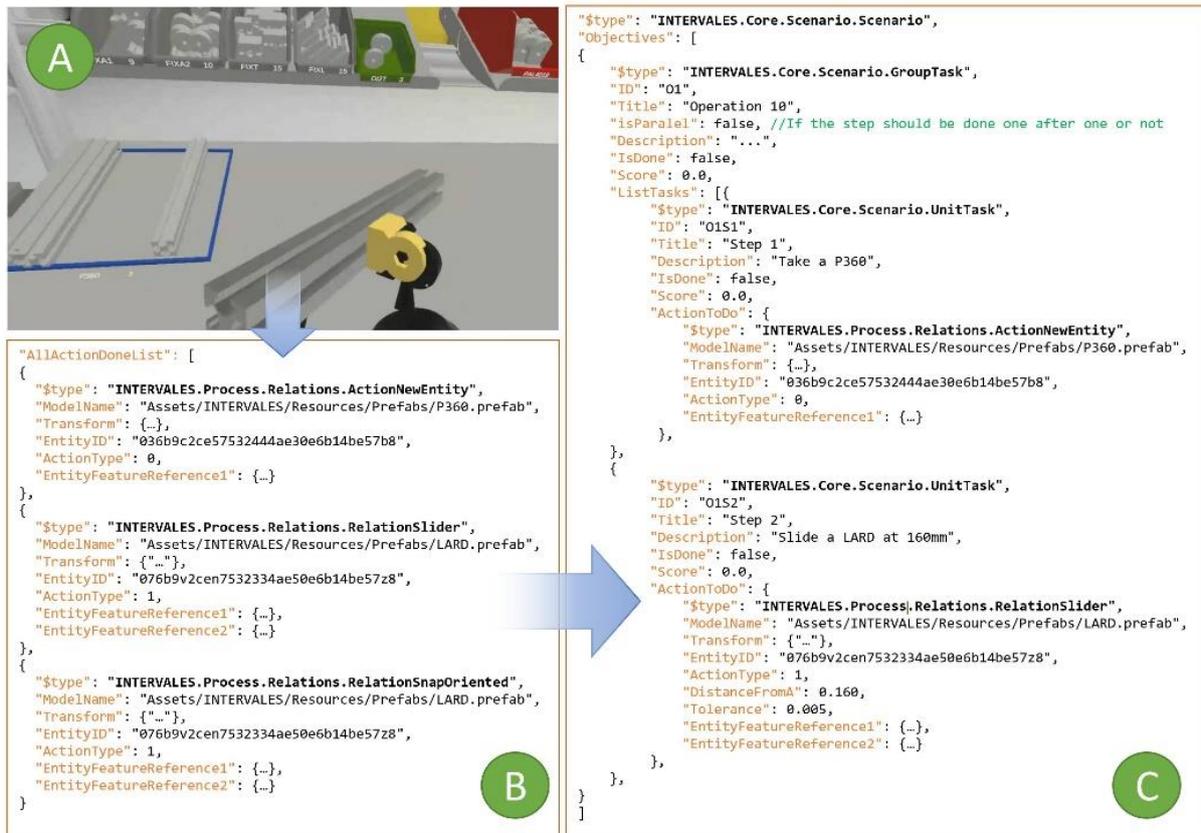


Figure 99 : (A) Actions réalisées en RV par un développeur / expert métier ; (B) Actions effectuées enregistrées ; (C) Action convertie en tâches à faire dans le scénario créé par l'authoring-by-doing.

5.4 Développement d'un nouvel éditeur 2D et d'un éditeur en authoring-by-doing

Suite à cela, nous avons développé deux nouveaux éditeurs de scénario : une interface d'édition 2D et un moyen d'édition en réalité virtuelle via l'authoring-by-doing. Comme notre première version présentée dans la section 5.2.2, l'interface d'édition 2D que nous avons développée se base sur des graphes de nœuds (Figure 100). Cette nouvelle version prend en compte le formalisme INTERVALES. Cette interface permet donc de créer des scénarios en y ajoutant des nœuds « *ObjectiveNode* » spécialisés, comme ici avec des « *SnapOrientedNode* », et en les reliant par des « *Connexion* ». Il est possible de les ajouter, ou de les supprimer, en utilisant un menu rapide en cliquant soit dans l'espace d'édition, soit sur un des éléments. Une fois un nœud créé, il est possible de configurer la tâche attendue via des listes déroulantes. Dans l'exemple d'une *SnapOrientedNode*, il va ainsi être possible de configurer les entités et les features attendues puis le degré de rotation. De plus, contrairement à notre première version, ce nouvel éditeur fonctionne directement au sein de l'application compilée, contrairement au précédent qui ne fonctionnait que dans l'éditeur Unity.

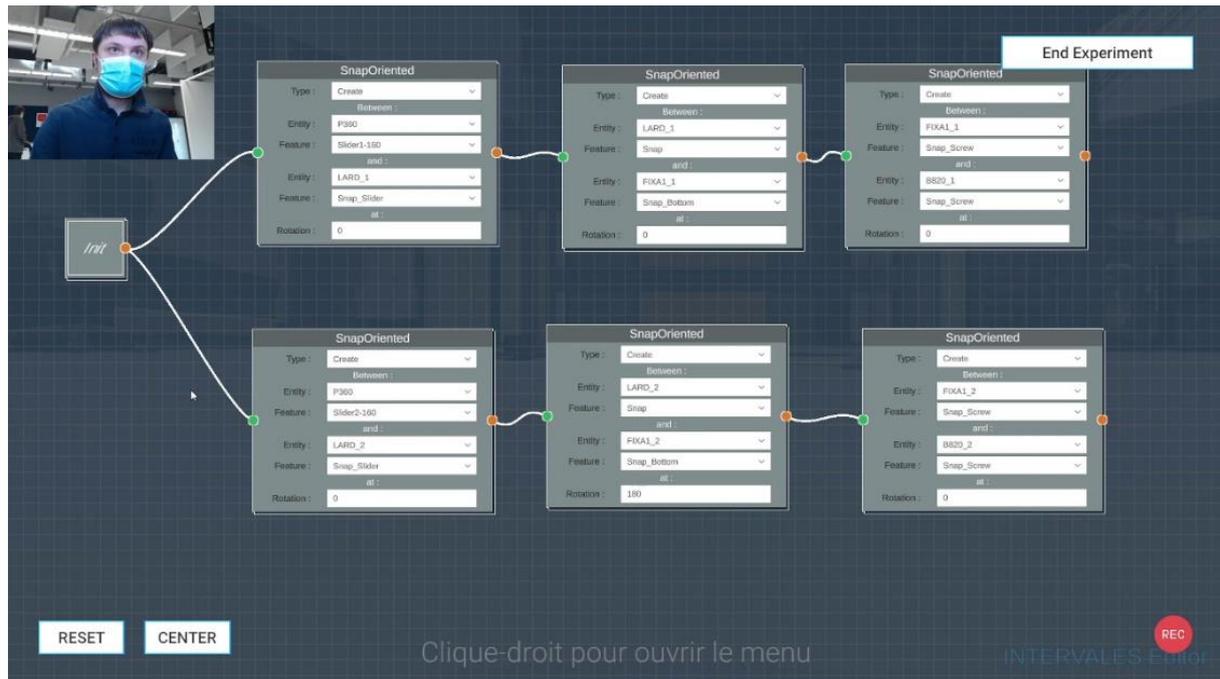


Figure 100 : Interface d'édition de scénario de l'éditeur 2D.

Nous avons également mis en place une édition directement en réalité virtuelle en se basant sur le système d'autoring-by-doing présenté dans la section précédente (5.3). Pour cela, l'utilisateur a la possibilité de passer en mode « Recording », ainsi chaque action qu'il réalise est convertie en tâche puis ajoutée au scénario en cours d'édition comme montré dans la figure 101.



Figure 101 : (en haut) L'assemblage en cours de réalisation ; (en bas) Fenêtre montrant le scénario en cours d'édition par « authoring-by-doing ».

Pour passer facilement en *authoring-by-doing*, nous avons ajouté un menu radial d'actions (Figure 102). Un expert métier peut alors directement activer / désactiver l'édition en *authoring-by-doing* depuis la réalité virtuelle. Dans le cas où l'utilisateur ajoute une action par erreur, il lui est possible via ce même menu de revenir en arrière (Figure 102 - « undo » représenté par la flèche retour) ou encore de redémarrer un nouveau scénario depuis le début (Figure 102 - « reset » représenté par la flèche faisant un tour complet).

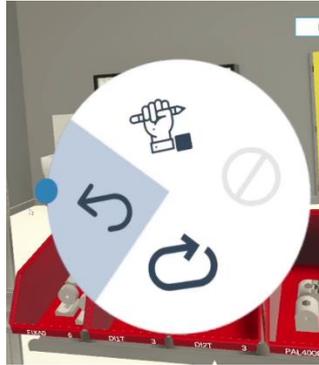


Figure 102 : Menu radial pour l'*authoring-by-doing*

Les deux éditeurs pouvant être tous les deux accessibles au sein d'une même application et ayant accès aux mêmes données de l'environnement, il est alors possible de passer d'un éditeur à l'autre. En effet, l'*authoring-by-doing* enregistre chaque action faite les unes à la suite des autres de manière procédurale. Nous pouvons imaginer, qu'un formateur voulant créer un scénario d'assemblage pourra dans un premier temps le réaliser en *authoring-by-doing* puis le retravailler en utilisant l'éditeur 2D pour le restructurer, en le rendant parallèle par exemple. Nous comparerons l'efficacité et l'expérience utilisateur ressentie des deux éditeurs dans la section suivante.

5.5 Expérimentations et résultats

5.5.1 Objectif de l'expérimentation

Comme présenté dans les sections précédentes, nous avons intégré à INTERVALES un module d'édition de scénarios avec deux outils : une interface d'édition PC en 2D et en réalité virtuelle avec « *authoring-by-doing* ». L'objectif de cette expérimentation est d'étudier deux points principaux :

- Étudier la capacité d'un expert métier, sans compétence en développement informatique, à créer / éditer un scénario d'assemblage en utilisant ces outils d'édition.
- Comparer les deux méthodes de création de scénario, en termes de performance et d'expérience utilisateur :
 - Via l'interface d'édition en 2D,
 - Via « *authoring-by-doing* » en réalité virtuelle.

L'expérimentation se déroule en utilisant le même cas d'usages que la première expérimentation, c'est-à-dire l'assemblage sur poste manuel.

5.5.2 Définition du protocole expérimental

5.5.2.1 Variables étudiées

Dans cette nouvelle expérimentation, nous souhaitons comparer des critères de performance ainsi que de satisfaction suivant le profil du participant ou de l'éditeur utilisé pour créer un scénario d'assemblage sur poste manuel. Nous nous sommes basés sur (Nielsen, 1994) pour catégoriser les critères d'évaluation en 3 catégories. Ainsi, nous avons comparé des variables liées à :

- La performance en termes d'édition de scénario :
 - La durée de l'édition,
 - Le niveau de qualité du scénario réalisé,
 - Le nombre d'erreurs,
- Le comportement durant l'édition du scénario :
 - Le nombre d'actions faites pour créer un scénario.
- La satisfaction :
 - UEQ (User Experience Questionnaire),
 - Les retours d'expériences des participants.

5.5.2.2 Participants

Pour cette expérimentation, nous souhaitons étudier deux profils de personnes : développeur informatique et industriel. En tout, trente-cinq personnes ont participé à l'expérimentation, douze ayant un profil industriel que nous noterons « INDUS », et vingt-trois ayant un profil informatique, noté « INFO ». Les participants ont entre 20 et 30 ans et viennent de différentes promotions de l'école d'ingénieurs CESI dans les domaines informatique ou génie industriel allant de BAC+3 au doctorat (30 hommes et 5 femmes). Ces participants sont ensuite séparés en deux groupes « GR1 » et « GR2 » puis chaque groupe en sous-groupe A et B comprenant chacun la moitié des profils industriels et la moitié des profils informatiques.

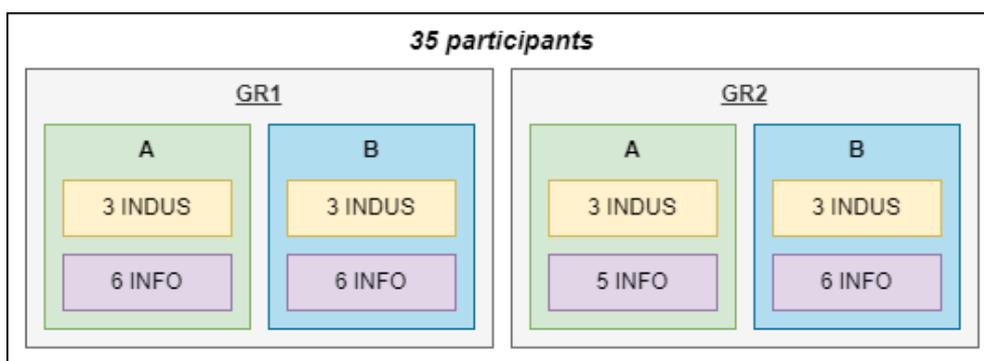


Figure 103 : Répartition des participants

5.5.2.3 Cas d'usages utilisé et découpage de l'expérimentation

L'expérimentation utilise le cas d'assemblage sur poste manuel vu dans la section 4.3.1 comme cas d'usages. Nous avons fait le choix de la découper en deux sous-expériences avec deux instructions d'assemblage dont les participants doivent réaliser le scénario. Un premier

scenario simple puis un second plus complexe. Nous comparons la création de ce scenario via l'édition 2D et l'authoring-by-doing.

Pour la première expérimentation, les participants avaient pour consigne de créer un **scénario linéaire** (procédural) correspondant aux opérations **P10** et **P20** (Figure 104, voir annexes 10.2 pour les fiches d'instructions complètes).

1. Opération « P10 » :
 - a. Placer le **lardon à 160 mm (Slider2_160)** sur le **profil**,
 - b. Placer la **fixation d'angle 1** sur le **lardon à 0°**,
 - c. Placer le **boulon M8 x 20** sur la **fixation d'angle 1**.
2. Opération « P20 » :
 - a. Placer un **lardon à 160 mm (Slider2_160)** sur le **profil**
 - b. Placer la **fixation d'angle 1** sur le **lardon (180°)**
 - c. Placer le **boulon M8 x 20** sur la **fixation d'angle 1**

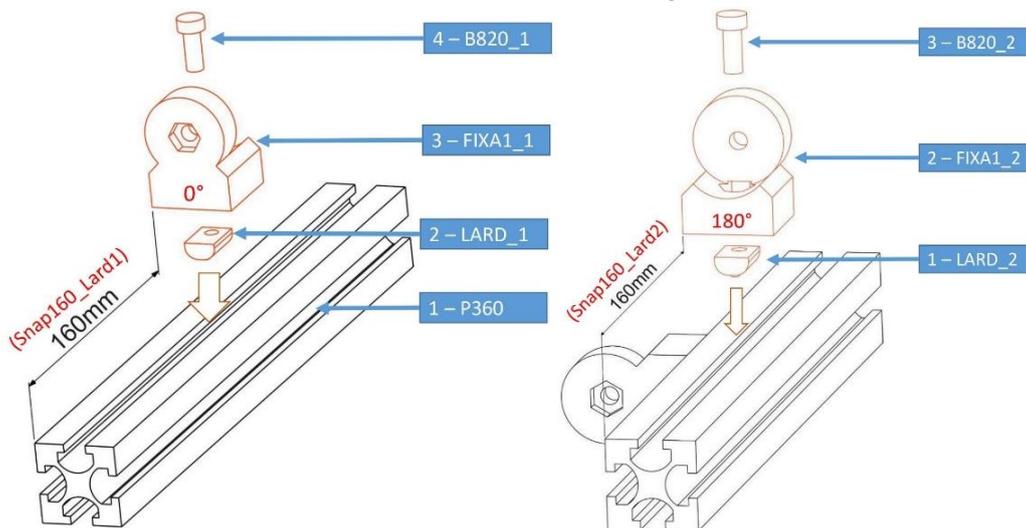


Figure 104 : Extrait des fiches d'instructions P10 et P20

Ensuite, la seconde expérimentation a pour objectif de créer un **scénario avec parallélisme** correspondant à l'opération **P30**. Nous comparons alors l'édition 2D et l'authoring-by-doing suivie d'une réorganisation via l'interface 2D.

Opération « P30 », pour le côté gauche et droit :

1. Placer un **lardon** sur le **profil**
2. Placer une **fixation en T** sur le **lardon à 90°/270°**
3. Placer un **bloc de palier 40x40x19** sur la **fixation en T**
4. Placer un **boulon M8x40** sur le **bloc de palier 40x40x19**
5. Placer un **capot** sur une extrémité du **profil**

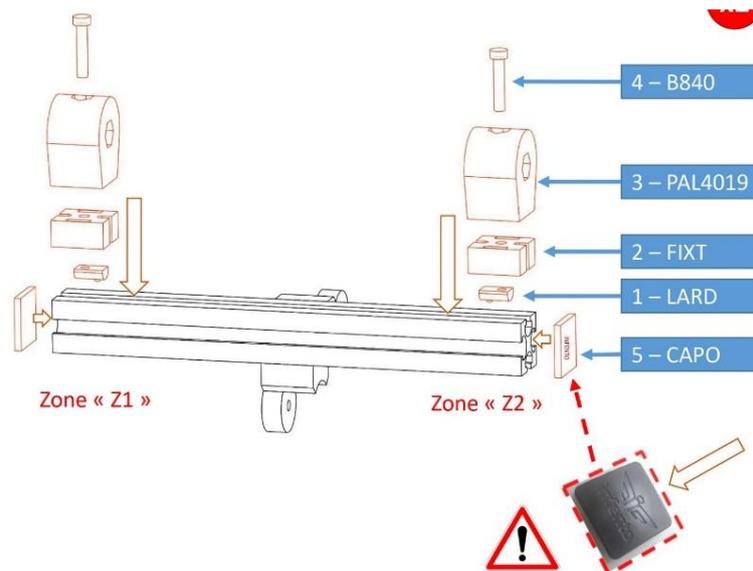


Figure 105 : Extrait de la fiche d'instructions P30

5.5.2.4 Déroulement de l'expérimentation

La figure 106 représente le déroulement de l'expérimentation. En amont de l'étude, un pré-questionnaire a été envoyé à chaque participant afin de recueillir leur niveau d'expérience à utiliser la réalité virtuelle et leur niveau d'aisance à effectuer des tâches manuelles comme l'assemblage. Ensuite, nous avons réparti les participants de manière aléatoire dans les deux groupes définis en équilibrant les profils industriels et informatiques suite au pré-questionnaire afin d'obtenir des groupes homogènes. Une fois les groupes constitués et les participants assignés à un groupe et sous-groupe, l'expérimentation a pu être planifiée. Nous avons à notre disposition 4 postes opérationnels équipés de casque de réalité virtuelle HTC Vive Pro Eye. L'expérimentation se déroule donc par session de 4 participants maximum. Chaque poste est également équipé d'une webcam et d'un logiciel de capture d'écran (Open Broadcaster Software : OBS¹⁰) pour enregistrer ce que voit l'utilisateur à l'écran ou dans son casque immersif et le voir physiquement pendant la session d'expérimentation (Figure 107).

Pour introduire l'étude de cas, une vidéo de présentation de l'expérimentation est visionnée par les participants, suivie ensuite d'une vidéo explicative de l'interface d'édition 2D puis d'une autre sur l'authoring-by-doing. Puis, une phase libre d'utilisation est réalisée avec la possibilité de demander de l'aide en cas de besoin afin de permettre aux participants de bien prendre en main les outils.

Pour le premier scénario, les participants sont assignés à un moyen d'édition suivant leur sous-groupe (A ou B), ceux-ci vont donc soit commencer par utiliser l'éditeur 2D ou l'authoring-by-doing en réalité virtuelle. Cette répartition a pour but de voir s'il y a une influence suivant l'ordre de passage du participant sur l'éditeur 2D ou l'authoring-by-doing pour un même scénario. Les participants réalisent alors le scénario demandé. Suite à ce premier passage, le participant

¹⁰ <https://obsproject.com/>

répondra à une première série de deux questionnaires propres à l'éditeur utilisé évaluant : l'expérience utilisateur avec l'UEQ (Schrepp et al., 2017) et un questionnaire composé de questions ouvertes recueillant ses retours d'expériences (Voir annexe 10.5). Ensuite les participants ayant utilisé l'éditeur 2D réalisent le même scénario en utilisant uniquement l'authoring-by-doing et inversement pour ceux ayant commencé par l'authoring-by-doing. À la fin de ce deuxième passage, les participants répondent à la même série de deux questionnaires correspondant à l'autre outil d'édition.

Ensuite, pour le deuxième scénario, les participants ont pour consigne de réaliser un scénario parallèle. Suivant leur groupe, ils devront le réaliser soit uniquement via l'éditeur 2D soit en combinant l'authoring-by-doing avec l'éditeur 2D.

La partie suivante détaillera les méthodes utilisées pour évaluer chaque critère.

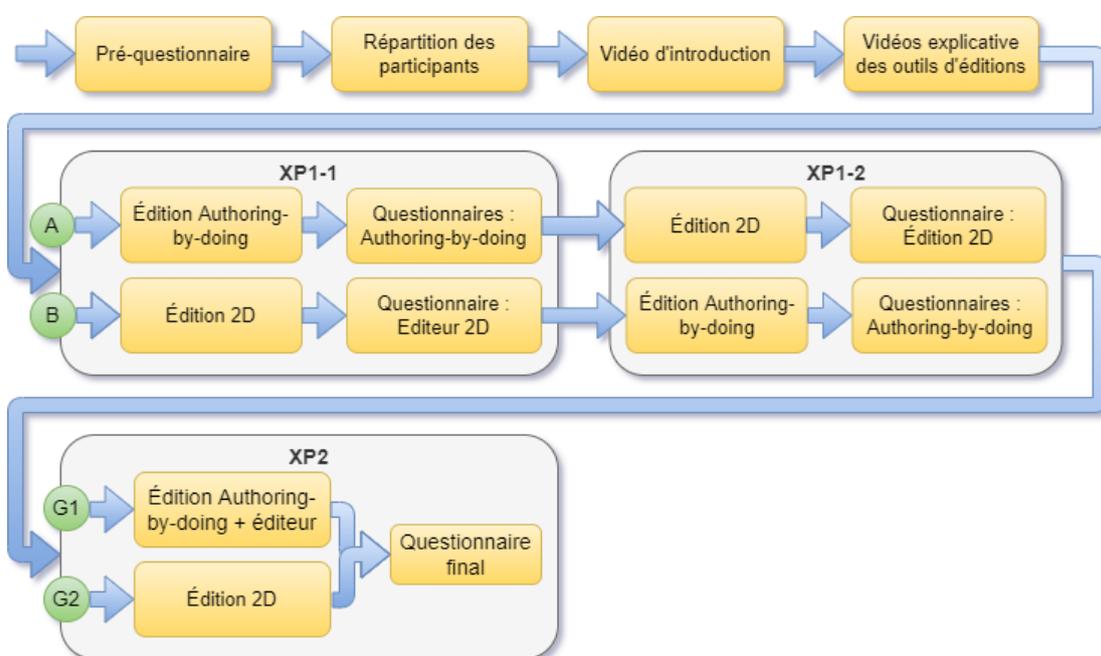


Figure 106 : Déroulement de l'expérimentation

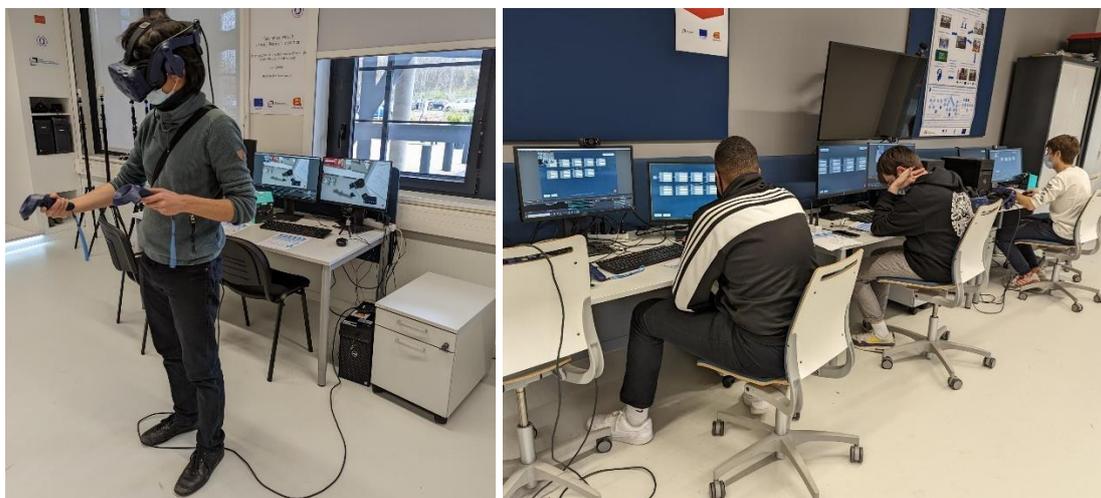


Figure 107 : Photos prises pendant la réalisation de l'expérience. Avec à gauche, via l'authoring-by-doing et à droite, via l'éditeur 2D.

5.5.3 Méthodes d'évaluations utilisées

5.5.3.1 Pré-questionnaire d'évaluation du niveau des participants

Nous avons donc envoyé en amont de l'expérimentation un pré-questionnaire aux participants. Le but de ce questionnaire était de collecter dans un premier temps des informations générales telles que la fourchette d'âge ou leur niveau d'étude. Ensuite, nous voulons évaluer le niveau des participants sur deux points : leur familiarité avec la réalité virtuelle et leur niveau d'aisance manuel. Les résultats obtenus permettent de mieux répartir les participants dans les différents groupes de manière homogène. Pour cela les questionnaires se composaient de deux questions :

- À quel niveau considérez-vous votre niveau en réalité virtuelle ?
 - Jamais utilisée
 - Utilisée seulement 1 à 2 fois
 - Utilisée une dizaine de fois
 - Habitué car utilisée régulièrement (Ex : au travail ou sur jeux vidéo)
- À quel niveau considérez-vous votre aisance manuelle (assemblage / maintenance / bricolage)
 - J'en fais 1 fois par an
 - J'en fais 3 fois par an
 - J'en fais 1 fois par mois
 - J'en fais 1 fois par semaine
 - J'en fais à mon travail

Concernant la réalité virtuelle, si le participant n'avait jamais utilisé de la réalité virtuelle ou seulement 1 à 2 fois, il était considéré comme débutant. S'il l'avait utilisée une dizaine de fois, un niveau intermédiaire et pour les habitués, un niveau expert.

Concernant l'aisance manuelle, nous voulions prendre en compte le niveau et la fréquence des participants à effectuer des tâches d'assemblages en réel.

5.5.3.2 Évaluation de l'édition de scénario

Afin de pouvoir évaluer la performance des outils en matière d'édition de scénario. Nous allons comparer plusieurs données qui seront récupérées durant l'expérimentation :

- Performance :
 - La durée mise pour créer le scénario demandé (en secondes),
 - Le niveau de qualité du scénario réalisé,
 - Le nombre d'erreurs sur le scénario final vis-à-vis du scénario demandé,
- Comportementale :
 - Le nombre d'actions faites pour créer un scénario.

Pour effectuer cette évaluation, nous avons recueilli différentes données pour chaque expérimentation et par participant :

- Une vidéo de l'édition,
- Le fichier JSON au format INTERVALES de l'état de la scène contenant le scénario réalisé,

- Un fichier texte exporté depuis l'application contenant tous les logs dont les actions faites.

Évaluation de la durée :

Concernant la durée d'édition, chaque session d'expérimentation est enregistrée et permet de comptabiliser le temps global d'édition. À noter que les périodes d'inactivité liées à un problème technique ou une demande de précision de la part du participant ne sont pas comptabilisées dans le temps final.

Évaluation du niveau de qualité :

Ensuite afin d'évaluer le niveau de qualité d'un scénario réalisé par un participant par rapport à celui attendu, le critère a été découpé en deux points :

- Le niveau de respect des étapes : c'est-à-dire sur la présence ou non des étapes attendues et le respect de leur ordre dans le scénario réalisé.
- Le niveau de paramétrage des étapes : est-ce que les étapes sont bien configurées / réalisées. Par exemple, si les pièces qui sont assemblées sont au bon endroit ou bien orientées.

Pour cela, nous avons établi 3 niveaux de classification :

- Ne réponds pas aux attentes (0) : le travail réalisé par le participant ne correspond pas aux instructions données.
- Réponds aux attentes (1) : le travail réalisé par le participant répond aux attentes.
- Dépasse les attentes (2) : le travail réalisé par le participant est supérieur aux instructions données et dépasse les attentes.

En mettant en lien, le respect des étapes et du paramétrage cela nous permet d'obtenir un niveau global en suivant la matrice ci-dessous (Tableau 5). Pour que le niveau global puisse être considéré comme répondant aux attentes, il faut à minima que le niveau de respect des étapes et du paramétrage réponde aux attentes. Ensuite pour qu'il puisse être considéré comme dépassant les attentes, il faut qu'au moins un des deux niveaux dépasse les attentes, l'autre doit obligatoirement répondre aux attentes. Ainsi le niveau global équivaut à $N_{global} = N_{étapes} * N_{paramétrage}$. Par exemple, dans le cas de la première expérimentation, nous avons considéré comme dépassant les attentes, les participants ayant réalisé un scénario parallèle alors que la consigne fournit était de réaliser un scénario linéaire (procédural). Pour la deuxième expérimentation, nous avons considéré comme dépassant les attentes les scénarios allant plus loin dans le parallélisme des étapes.

Tableau 5 : Tableau résumant les différentes possibilités du niveau global suivant le niveau de respect des étapes et du paramétrage.

Niveau étapes	Niveau paramétrage	Niveau global
0	0	0
1	0	0
2	0	0
1	1	1
2	1	2

Évaluation du nombre d'erreurs :

Les erreurs visibles sur le scénario final réalisé par le participant sont comptabilisées (Ex : manque d'une étape, mauvais paramétrage). Par exemple, si le participant place dans le mauvais ordre une étape au début de la création du scénario, mais qu'il la corrige avant qu'il ne signale que son scénario est terminé, alors cela n'est pas considéré comme une erreur. Cette autocorrection sera analysée dans la partie concernant le nombre d'actions effectuées pour l'édition du scénario.

Évaluation du nombre d'actions faites :

Grâce au fichier de log exporté à chaque session, nous avons connaissance des actions réalisées par un participant de la création d'un nouveau nœud sur l'éditeur 2D à l'assemblage d'une pièce en réalité virtuelle. Voici une liste complète des actions qui ont été comptabilisées pour les deux éditeurs :

- Éditeur 2D
 - Création d'une connexion
 - Création d'un nœud
 - Suppression d'une connexion
 - Suppression d'un nœud
- Authoring-by-doing
 - Faire un assemblage
 - Défaire un assemblage
 - Annulation d'un assemblage (Undo)

Hypothèses d'expérimentation :

L'évaluation de l'édition de scénario permet alors de répondre aux hypothèses suivantes :

- H_{1.édition} : l'éditeur 2D et l'authoring-by-doing permettent d'éditer un scénario,
- Hypothèses sur les temps d'éditations :
 - H_{2.édition} : les temps d'éditations via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique,
 - H_{3.édition} : les temps d'éditations via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique,
 - H_{4.édition} : l'authoring-by-doing réduit le temps mis pour éditer un scénario simple par rapport à l'éditeur 2D,
 - H_{5.édition} : l'authoring-by-doing réduit le temps mis pour éditer un scénario complexe par rapport à l'éditeur 2D,
- Hypothèses sur le nombre d'actions :
 - H_{6.édition} : le nombre d'actions via l'éditeur 2D est similaire entre un profil industriel et un profil informatique,
 - H_{7.édition} : le nombre d'actions via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique,
 - H_{8.édition} : l'authoring-by-doing réduit le nombre d'actions faites pour un scénario simple par rapport à l'éditeur 2D,
 - H_{9.édition} : l'authoring-by-doing réduit le nombre d'actions faites pour un scénario complexe par rapport à l'éditeur 2D,
- Hypothèses sur le nombre d'erreurs :

- H_{10.édition} : le nombre d'erreurs via l'éditeur 2D est similaire entre un profil industriel et un profil informatique,
- H_{11.édition} : le nombre d'erreurs via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique,
- H_{12.édition} : l'authoring-by-doing réduit le nombre d'erreurs commises pour un scénario simple par rapport à l'éditeur 2D,
- H_{13.édition} : l'authoring-by-doing réduit le nombre d'erreurs commises pour un scénario complexe par rapport à l'éditeur 2D.

Dans un premier temps, pour vérifier l'hypothèse H_{1.édition}, nous évaluerons le pourcentage de participants répondant aux attentes sur la qualité de leur scénario. Ensuite, pour vérifier les hypothèses H_{2.édition} et H_{3.édition}, nous comparerons les durées d'éditations en fonction des profils respectivement sur l'éditeur 2D et l'authoring-by-doing. En utilisant un test statistique (t-test/test de Student (Zabell, 2008) ou test de Mann et Whitney (Wilcoxon, 1945)), nous vérifierons qu'il n'y a pas de différence significative entre les deux profils que ce soit pour réaliser un scénario procédural ou un scénario plus complexe parallèle. Puis, pour vérifier les hypothèses H_{4.édition} à H_{13.édition}, nous comparerons la durée, le nombre d'actions et le nombre d'erreurs entre les deux outils, par profils et pour un scénario simple et complexe. De même, nous pourrions vérifier si la différence est significative avec un t-test ou un test de Mann et Whitney.

5.5.3.3 Évaluation de l'expérience utilisateur

Pour évaluer l'expérience utilisateur, nous avons utilisé la version réduite de l'UEQ (User Experience Questionnaire) (Laugwitz et al., 2008). Il s'agit d'un questionnaire regroupant 8 paires d'adjectifs opposées (Figure 108) avec une échelle de Likert de 1 à 7 pointant vers l'un des deux. Ce questionnaire est fourni avec un outil d'analyse sur deux axes : la qualité pragmatique et la qualité hédonique. Les résultats vont de -3 (très mauvais) à 3 (excellent), sachant que les valeurs entre -0.8 and 0.8 peuvent être considérées comme neutre.

Rigide	○ ○ ○ ○ ○ ○ ○ ○	Facilitant
Compliqué	○ ○ ○ ○ ○ ○ ○ ○	Simple
Inefficace	○ ○ ○ ○ ○ ○ ○ ○	Efficace
Déroutant	○ ○ ○ ○ ○ ○ ○ ○	Clair
Ennuyeux	○ ○ ○ ○ ○ ○ ○ ○	Captivant
Inintéressant	○ ○ ○ ○ ○ ○ ○ ○	Intéressant
Conventionnel	○ ○ ○ ○ ○ ○ ○ ○	Original
Habituel	○ ○ ○ ○ ○ ○ ○ ○	Avant-gardiste

Figure 108 : Version courte de l'UEQ

Nos hypothèses concernant l'expérience utilisateur que nous souhaitons vérifier sont les suivantes :

- H_{1.UEQ} : l'expérience utilisateur de l'éditeur 2D est similaire pour les profils industriels et informatiques,

- H_{2.UEQ} : l'expérience utilisateur de l'authoring-by-doing est similaire pour les profils industriels et informatiques,
- H_{3.UEQ} : l'éditeur utilisé a un impact sur l'expérience utilisateur des profils industriels,
- H_{4.UEQ} : l'éditeur utilisé a un impact sur l'expérience utilisateur des profils informatiques,

5.5.3.4 Évaluation du retour d'expérience

Enfin, pour évaluer le retour d'expérience final, chaque participant doit répondre à une question à choix unique concernant le moyen d'édition qu'il a préféré lors du questionnaire final : « Suite à cette expérience, préférez-vous l'édition 2D, l'authoring by doing ou la combinaison des deux ? ». Suivie d'une question ouverte « Pourquoi ce choix ? ».

L'hypothèse à vérifier est la suivante :

- H_{1.REX} : Les participants sont favorables à une édition en authoring-by-doing.

À la fin de chaque questionnaire, les participants ont la possibilité d'ajouter des commentaires et avis sur le ou les éditeurs utilisés pour l'expérience.

5.5.4 Analyse des résultats obtenus et discussions

5.5.4.1 Résultats du pré-questionnaire d'évaluation du niveau des participants

Pour commencer, il y a plusieurs remarques que nous pouvons noter sur les résultats de pré-questionnaire envoyé en amont de l'étude. Tout d'abord concernant l'âge (Figure 109), les participants à cette étude ont tous moins de 30 ans et **80%** d'entre eux ont moins de 25 ans. L'échantillon sur lequel nous avons réalisé l'étude est plutôt jeune et il peut être supposé qu'ils aient une prise en main facilitée des outils numériques proposés.

Répartition des âges des participants

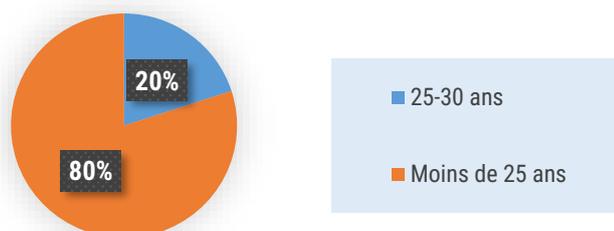


Figure 109 : Répartition des âges des participants

Ensuite, concernant leur niveau de formation, tous les participants ont un niveau de formation au minimum équivalent ou supérieur à un Bac +2 (Figure 110) dont **63%** ont un niveau d'étude au moins Bac +4 et **20%** ont un niveau équivalent et supérieur à Bac +5.

Répartition du niveau de formation

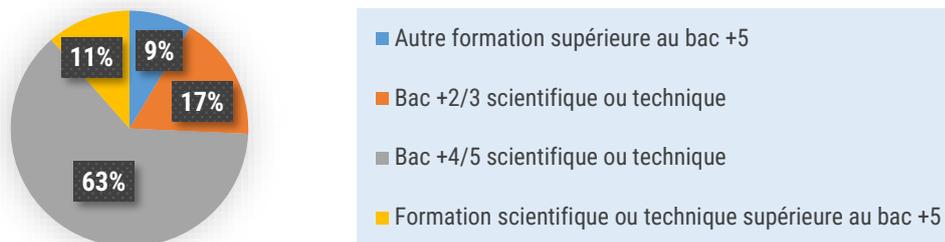
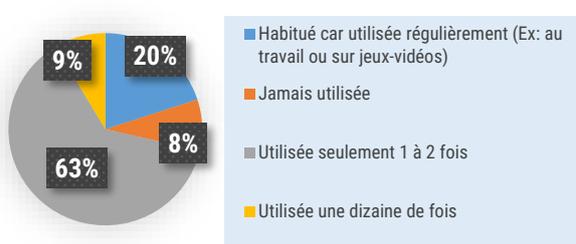


Figure 110 : Répartition du niveau de formation des participants

Puis, afin de connaître leur niveau en réalité virtuelle et en assemblage, nous leur avons demandé leur niveau d'expérience avec la réalité virtuelle et à quelle fréquence ils effectuaient des tâches manuelles (Ex : assemblage de meuble). Concernant la réalité virtuelle (Figure 111 – à gauche), seulement **8%** des participants n'en avaient jamais fait l'expérience et la majorité, **63%**, l'avait déjà utilisé 1 à 2 fois. En effet, une grande partie des participants connaissait déjà la réalité virtuelle.

Concernant l'aisance manuelle (Figure 111 – à droite), seul **17%** des participants réalisent très peu de tâches manuelles ou de bricolage. Ainsi, nous pouvons supposer qu'une grande partie des participants sont à l'aise lorsqu'il s'agit d'effectuer un assemblage.

Répartition du niveau en réalité virtuelle



Répartition du niveau d'aisance manuelle

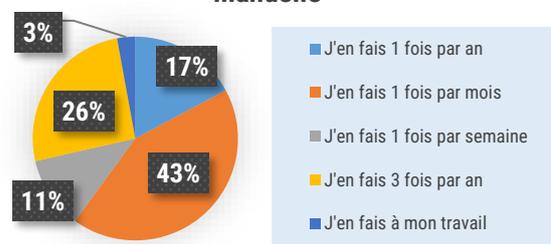


Figure 111 : À gauche, répartition du niveau en réalité virtuelle des participants. À droite, répartition du niveau d'aisance manuelle des participants

Suite à cela, les participants ont pu être répartis de manière équitable dans les différents groupes et sous-groupes détaillés dans la section 5.5.2.2.

5.5.4.2 Remarques sur les résultats entre les deux passages de l'expérience 1

L'organisation de l'expérience 1 (XP1) étant découpé en 2 passages (XP1-1 et XP1-2) afin que chaque participant puisse expérimenter l'édition d'un même scénario procédural avec les deux outils d'édition, nous souhaitons d'abord vérifier s'il y a une influence ou un biais d'apprentissage entre les deux passages XP1-1 et XP1-1. Nous allons donc nous concentrer sur la performance d'édition : le temps, le nombre d'erreurs, le nombre d'actions et le niveau du scénario réalisé. Pour cela, nous comparerons ces quatre variables à l'aide d'un t-test entre le passage XP1-1 et XP1-2 pour tester s'il y a une différence statistiquement significative entre les deux. Afin de pouvoir utiliser le t-test, nous devons vérifier les hypothèses d'homogénéité des

variances et de normalité des données. Pour cela, nous allons d'abord tester l'homogénéité des variances avec un test de Levene ainsi que l'hypothèse de normalité avec un test de Shapiro-Wilk. Dans le cas où ces hypothèses ne sont pas vérifiées, nous utiliserons un test de rang (Mann et Whitney). Pour ces tests, les données des profils industriels et informatiques ont été groupées en fonction de l'outil d'édition utilisé : Éditeur 2D (2D) et Authoring-by-Doing (AbD).

5.5.4.2.1 Le temps d'édition avec un outil d'édition en fonction du 1^{er} ou 2^{ème} passage

5.5.4.2.1.1 Analyses statistiques

Dans le Tableau 6, nous pouvons déjà observer une légère baisse du temps d'édition moyen entre le 1^{er} et le 2^{ème} passage en passant de **428 secondes à 364 secondes** avec l'édition 2D et de **175 secondes à 133 secondes** pour l'édition AbD. Soit respectivement une baisse de **~15%** pour les participants ayant utilisé l'éditeur 2D après l'authoring-by-doing et **~24%** pour ceux ayant utilisé l'authoring-by-doing après l'éditeur 2D.

Tableau 6 : Statistique générale du temps d'éditions (en secondes) pour les deux outils entre XP1-1 et XP1-2.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Statistique générale	Min = 231s Moy = 428s Max = 627s	Min = 228s Moy = 364s Max = 628s	Min = 70s Moy = 175s Max = 331s	Min = 53s Moy = 133s Max = 291s

Afin de tester ces observations, nous effectuons un t-test. Pour que le t-test soit valide, nous vérifions tout d'abord l'homogénéité des variances. En comparant XP1-1 et XP1-2, les résultats du test de Levene (Tableau 7) donnent une p-value de $0.336 > 0.05$ pour l'édition 2D et une p-value de $0.716 > 0.05$ pour l'authoring-by-doing. L'homogénéité des variances est ainsi validée.

Tableau 7 : Résultats du test de Levene pour XP1-1_2D_Temps vs XP1-2_2D_Temps (a) et XP1-1_AbD_Temps vs XP1-2_AbD_Temps (b).

	2D (a)	AbD (b)
Homogénéité des variances (Test de Levene)	F = 0.949 P _{value} = 0.336	F = 0.134 P _{value} = 0.716

Ensuite, le test de Shapiro donne avec l'édition 2D (Tableau 8), une p-value de $0.224 > 0.05$ pour l'XP1-1 et une p-value de $0.007 < 0.05$ pour l'XP1-2. Puis, pour l'édition AbD, une p-value de $0.644 > 0.05$ pour l'XP1-1 et une p-value de $0.017 < 0.05$ pour l'XP1-2. Les deux échantillons de l'XP1-2 ne suivent donc pas une loi normale.

Tableau 8 : Résultats du test de Shapiro pour les temps des deux passages. * indique que l'hypothèse de normalité n'est pas validée.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Hypothèse de normalité (Test de Shapiro-Wilk)	W = 0.933 P _{value} = 0.224	W = 0.838 P _{value} = 0.007*	W = 0.960 P _{value} = 0.644	W = 0.886 P _{value} = 0.017*

Les conditions pour effectuer un t-test ne sont donc pas valides. Nous utiliserons donc un test de Mann et Whitney (Tableau 9). Nous obtenons alors une p-value de $0.046 < 0.05$ pour l'édition 2D et une p-value de $0.028 < 0.05$ pour l'authoring-by-doing. La p-value est inférieure à 0.05 donc il y a une différence statistiquement significative au niveau du temps entre le 1^{er} et 2^{ème} passage suivant si le participant a fait 2D puis AbD ou AbD puis 2D sur le même scénario.

Tableau 9 : Résultats du test de Mann et Whitney pour les temps des deux passages. (a) XP1-1_2D_Temps vs XP1-2_2D_Temps et (b) XP1-1_AbD_Temps vs XP1-2_AbD_Temps. * représente les différences statistiquement significatives.

	2D (a)	AbD (b)
Test de Mann et Whitney	U = 101.5 P _{value} = 0.046*	U = 94 P _{value} = 0.028*

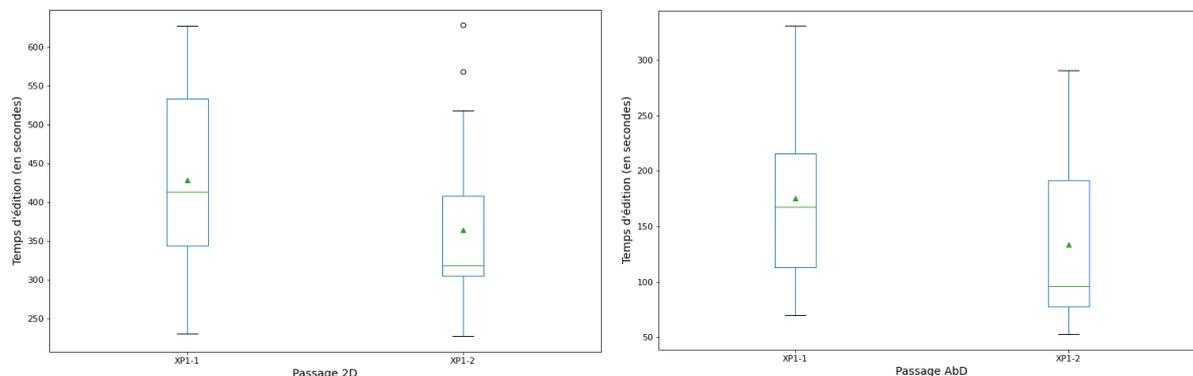


Figure 112 : (À gauche) Boîte à moustaches des temps d'édition selon le passage pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le passage pour l'édition AbD.

5.5.4.2.1.2 Discussion

Cette différence statistique montre donc qu'il y a un biais d'apprentissage au niveau du temps suivant l'ordre de l'outil utilisé lors du 1^{er} et 2^{ème} passage. Sur le 2^{ème} passage, nous pouvons confirmer, qu'en moyenne, les participants ont mis moins de temps pour réaliser le même scénario, environ **-15%** pour les participants ayant utilisé l'éditeur 2D après l'authoring-by-doing et environ **-24%** pour ceux ayant utilisé l'authoring-by-doing après l'éditeur 2D. Nous pouvons conclure que le fait d'avoir réalisé le scénario suite au 1^{er} passage a eu une influence sur le 2^{ème} passage et cela peu importe l'outil.

5.5.4.2.2 Le nombre d'erreurs avec un outil d'édition en fonction du 1^{er} ou 2^{ème} passage

5.5.4.2.2.1 Analyses statistiques

Dans le Tableau 10, nous pouvons déjà observer une augmentation du nombre d'erreur moyenne d'une part, et une baisse d'une autre, entre le 1^{er} et le 2^{ème} passage en passant de **0.28 à 0.82** avec l'édition 2D et de **0.87 à 0.33** pour l'édition AbD. Soit respectivement une hausse de **~+190%** et une baisse de **~-62%**.

Tableau 10 : Statistique générale du nombre d'erreurs pour les deux outils entre XP1-1 et XP1-2.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Statistique générale	Min = 0 Moy = 0.28 Max = 2	Min = 0 Moy = 0.82 Max = 3	Min = 0 Moy = 0.47 Max = 3	Min = 0 Moy = 0.33 Max = 2

Pour le reste de l'ensemble des analyses statistiques, nous ne détaillerons plus le processus d'analyse. Ainsi, le Tableau 11 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances n'est ainsi validée que pour l'authoring-by-doing. De plus, tous les échantillons ne suivent pas une loi normale.

Tableau 11 : Résultats du test de Levene et du test de Shapiro pour le nombre d'erreurs des deux passages. * indique que l'hypothèse n'est pas validée.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Homogénéité des variances (test de Levene)	F = 5.399 P _{value} = 0.026*		F = 0.268 P _{value} = 0.608	
Hypothèse de normalité (Test de Shapiro-Wilk)	W = 0.552 P _{value} = 2.34 ⁻⁰⁶ *	W = 0.765 P _{value} = 0.0007*	W = 0.618 P _{value} = 1.570 ⁻⁰⁵ *	W = 0.543 P _{value} = 1.97 ⁻⁰⁶ *

Les conditions pour effectuer un t-test ne sont donc pas valides. Nous utiliserons donc un test de Mann et Whitney (Tableau 12). Nous obtenons alors une p-value de $0.029 < 0.05$ pour l'édition 2D et une p-value de $0.325 > 0.05$ pour l'authoring-by-doing. La p-value est inférieure à 0.05 donc il y a une différence statistiquement significative au niveau des erreurs entre le 1^{er} et 2^{ème} passage pour les participants ayant utilisé l'édition 2D. Cependant, concernant l'authoring-by-doing, la p-value est supérieure à 0.05, il n'y a donc pas de différence significative entre le 1^{er} et 2^{ème} passage.

Tableau 12 : Résultats du test de Mann et Whitney pour le nombre d'erreurs des deux passages. (a) XP1-1_2D_Erreur vs XP1-2_2D_Erreur et (b) XP1-1_AbD_Erreur vs XP1-2_AbD_Erreur. * représente les différences statistiquement significatives.

	2D (a)	AbD (b)
Test de Mann et Whitney	U = 103.5 P _{value} = 0.029*	U = 142 P _{value} = 0.325

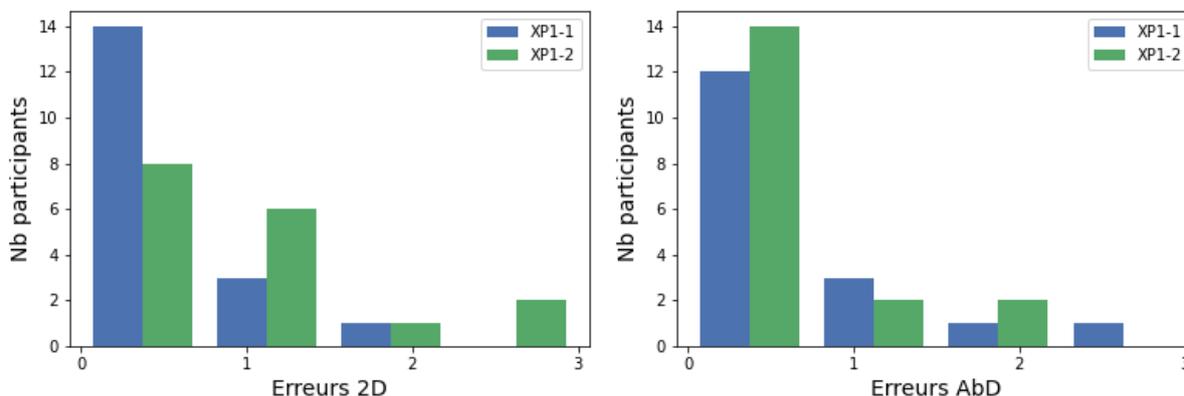


Figure 113 : (À gauche) Histogramme du nombre d'erreurs selon le passage pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le passage pour l'édition AbD.

5.5.4.2.2 Discussion

Pour conclure, cela signifie dans un premier temps que les participants ayant utilisé l'édition 2D après l'authoring-by-doing ont eu tendance à faire plus d'erreurs de manière significative. Nous pouvons supposer que cela est dû au fait que le paramétrage avec l'éditeur 2D est plus complexe qu'avec l'AbD, notamment sur l'angle d'orientation de la pièce. En effet, l'authoring-by-doing est un outil d'édition intuitif qui ne requiert qu'une compréhension visuelle globale de l'assemblage à réaliser ; seule la compréhension du dessin isométrique suffit puisque les paramètres, notamment des valeurs de rotation, sont remplis automatiquement lorsque l'utilisateur effectue

l'action dans la scène virtuelle. En passant ensuite sur le même scénario avec l'édition 2D, ces mêmes valeurs doivent être remplies par les participants. Or, les participants peuvent penser qu'avec l'éditeur 2D, ce remplissage est aussi automatique ; par manque d'attention, cela cause donc des erreurs de paramétrages supplémentaires. En outre, si les participants éditent bien le paramètre d'angle, ils doivent mentalement projeter comment la pièce se positionnera en fonction de l'angle, ce qui peut être source d'erreur aussi. Le constat concernant l'authoring-by-doing, indiquant que les participants ont fait en moyenne le même nombre d'erreurs, qu'ils l'aient utilisé au 1^{er} ou au 2^{ème} passage avec l'édition 2D, va en ce sens.

5.5.4.2.3 Le nombre d'actions avec un outil d'édition en fonction du 1^{er} ou 2^{ème} passage

5.5.4.2.3.1 Analyses statistiques

En regardant les statistiques générales (Tableau 13), nous pouvons déjà noter que le nombre d'actions moyen est très similaire entre le 1^{er} et le 2^{ème} passage sur les deux outils d'éditeurs en passant de **16.4** à **15.3** (-1.1) avec l'édition 2D et de **7.35** à **7.77** (+0.42) avec l'authoring-by-doing.

Tableau 13 : Statistique générale du nombre d'actions pour les deux outils entre XP1-1 et XP1-2.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Statistique générale	Min = 12 Moy = 16.4 Max = 21	Min = 12 Moy = 15.3 Max = 19	Min = 6 Moy = 7.35 Max = 19	Min = 6 Moy = 7.77 Max = 13

Le Tableau 14 donne les résultats du test de Levene et du test de Shapiro. Bien que l'homogénéité des variances soit validée pour l'édition 2D et l'authoring-by-doing, trois des quatre échantillons ne suivent pas une loi normale.

Tableau 14 : Résultats du test de Shapiro pour le nombre d'actions des deux passages. * indique que l'hypothèse n'est pas validée.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Homogénéité des variances (Test de Levene)	F = 3.774 P _{value} = 0.061		F = 0.011 P _{value} = 0.915	
Hypothèse de normalité (Test de Shapiro-Wilk)	W = 0.886 P _{value} = 0.033*	W = 0.912 P _{value} = 0.109	W = 0.49 P _{value} = 1.08 ⁻⁰⁶ *	W = 0.828 P _{value} = 0.004*

Les conditions pour effectuer un t-test ne sont donc pas valides. Nous utiliserons donc un test de Mann et Whitney (Tableau 15). Il n'y a donc pas de différence statistiquement significative au niveau du nombre d'actions réalisées entre les participants ayant utilisé l'éditeur 2D au 1^{er} passage et ceux l'ayant utilisé après l'authoring-by-doing. Cependant, concernant l'authoring-by-doing, la p-value est inférieure à 0.05 donc il y a une différence statistiquement significative entre le 1^{er} et 2^{ème} passage.

Tableau 15 : Résultats du test de Mann et Whitney pour le nombre d'actions des deux passages. (a) XP1-1_2D_Action vs XP1-2_2D_Action et (b) XP1-1_AbD_Action vs XP1-2_AbD_Action. * représente les différences statistiquement significatives.

	2D (a)	AbD (b)
Test de Mann et Whitney	U = 122.5 P _{value} = 0.159	U = 103 P _{value} = 0.039*

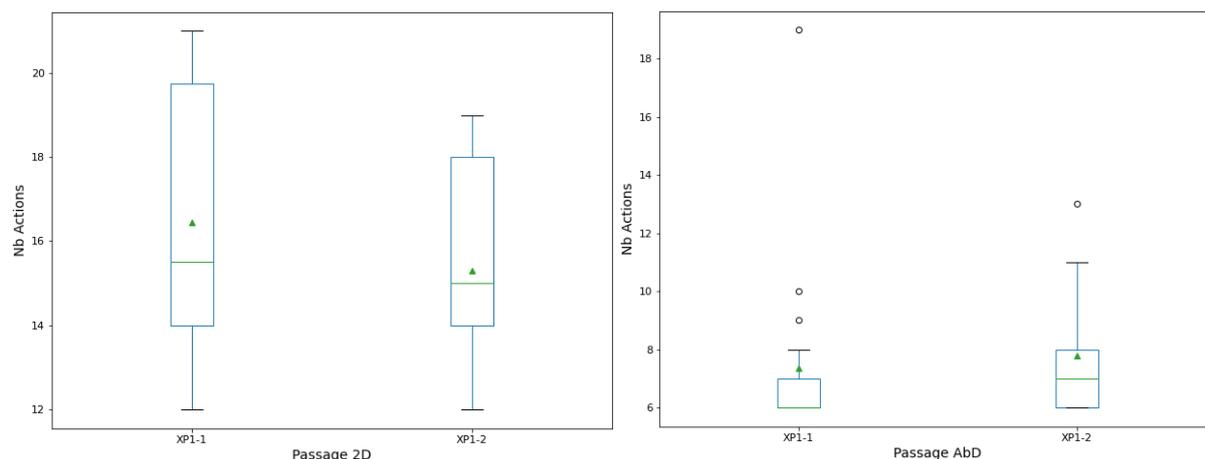


Figure 114 : (À gauche) Boîte à moustaches du nombre d'actions selon le passage pour l'édition 2D. (À droite) Boîte à moustaches du nombre d'actions selon le passage pour l'édition AbD.

5.5.4.2.3.2 Discussion

Nous avons un nombre d'actions similaires sur le 2^{ème} passage avec l'éditeur 2D mais une différence statistiquement significative sur le 2^{ème} passage avec l'éditeur AbD. Cela montre que les participants ayant utilisé l'éditeur 2D puis l'authoring-by-doing ont fait en moyenne plus d'actions.

Cette différence peut s'expliquer par le fait qu'en édition 2D, il fallait paramétrer chaque action. De ce fait, sur l'éditeur AbD, même si instinctivement le participant aboutit à un scénario correct avec peu d'actions, il était plus vigilant quant aux actions réalisées afin de s'assurer de l'orientation d'une pièce par exemple. Cette observation est en corrélation avec l'analyse précédente qui concernait le nombre d'erreurs.

5.5.4.2.4 Le niveau du scénario final avec un outil d'édition en fonction du 1^{er} ou 2^{ème} passage

5.5.4.2.4.1 Analyses statistiques

Pour ce qui est du niveau général du scénario obtenu sur les différents outils en fonction du 1^{er} et 2^{ème} passage, rappelons que : (0) correspond à un scénario ne répondant pas aux attentes, (1) à un scénario répondant aux attentes et (2) à un scénario dépassant les attentes. Rappelons également que les scénarios créés à partir de l'authoring-by-doing ne peuvent au maximum qu'atteindre le niveau « répond aux attentes ». En regardant les statistiques générales (Tableau 16), nous pouvons voir que le niveau moyen des scénarios réalisés avec l'édition 2D a baissé de **0.83** à **0.59** entre le 1^{er} et le 2^{ème} passage tandis qu'il a légèrement monté avec l'authoring-by-doing en passant de **0.71** à **0.78**.

Tableau 16 : Statistique générale du niveau du scénario pour les deux outils entre XP1-1 et XP1-2.

	2D		AbD	
	XP1-1	XP1-2	XP1-1	XP1-2
Statistique générale	Min = 0 Moy = 0.83 Max = 2	Min = 0 Moy = 0.59 Max = 2	Min = 0 Moy = 0.71 Max = 1	Min = 0 Moy = 0.78 Max = 1

En regardant la répartition du niveau des scénarios édités montrée dans la figure 115, nous remarquons qu'en effet, le pourcentage de participant ayant créé un scénario ne correspondant pas aux attentes passe de **33%** à **58%**. Tandis que pour l'authoring-by-doing, il diminue en passant de **29%** à **22%**. Comme pour le nombre d'erreur, nous pouvons supposer que cela est dû au passage de l'authoring-by-doing, qui est intuitif et donc requière moins d'effort de lecture des fiches d'instructions (voir annexe : 10.2), par rapport à l'édition 2D qui demande plus de précision. Les participants faisaient ainsi moins attention aux instructions écrites et donc commettaient des erreurs de paramétrage.

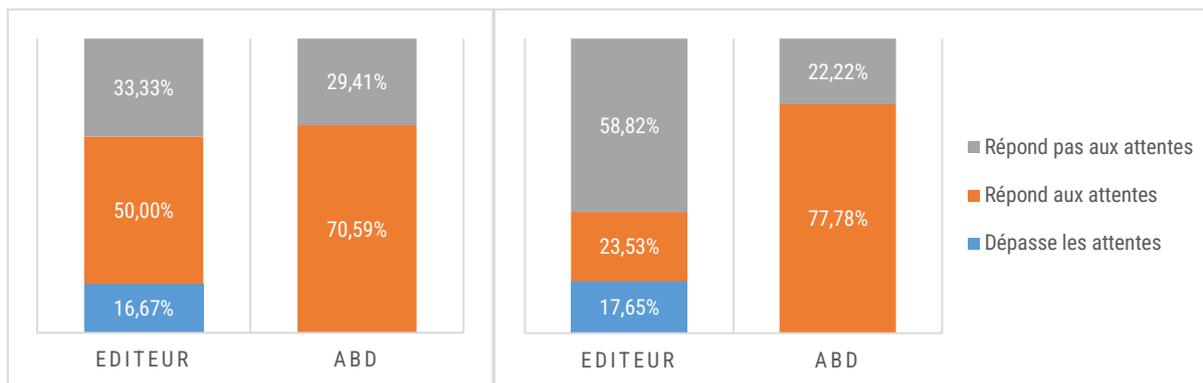


Figure 115 : Répartition du niveau des scénarios édités suivant l'outil d'édition en pourcentage, avec à gauche le 1^{er} passage et à droite, le 2^{ème}.

Vérifions ces observations de manière statistique. Pour cela, nous utiliserons un test de Mann et Whitney (Tableau 17). Il n'y a donc pas de différence statistiquement significative sur le niveau moyen des scénarios réalisés entre les participants ayant utilisé l'éditeur 2D au 1^{er} passage et ceux l'ayant utilisé après l'authoring-by-doing ou inversement.

Tableau 17 : Résultats du test de Mann et Whitney pour les niveaux des scénarios finaux des deux passages. (a) XP1-1_2D_ Niveau vs XP1-2_2D_ Niveau et (b) XP1-1_AbD_ Niveau vs XP1-2_AbD_ Niveau. * représente les différences statistiquement significatives.

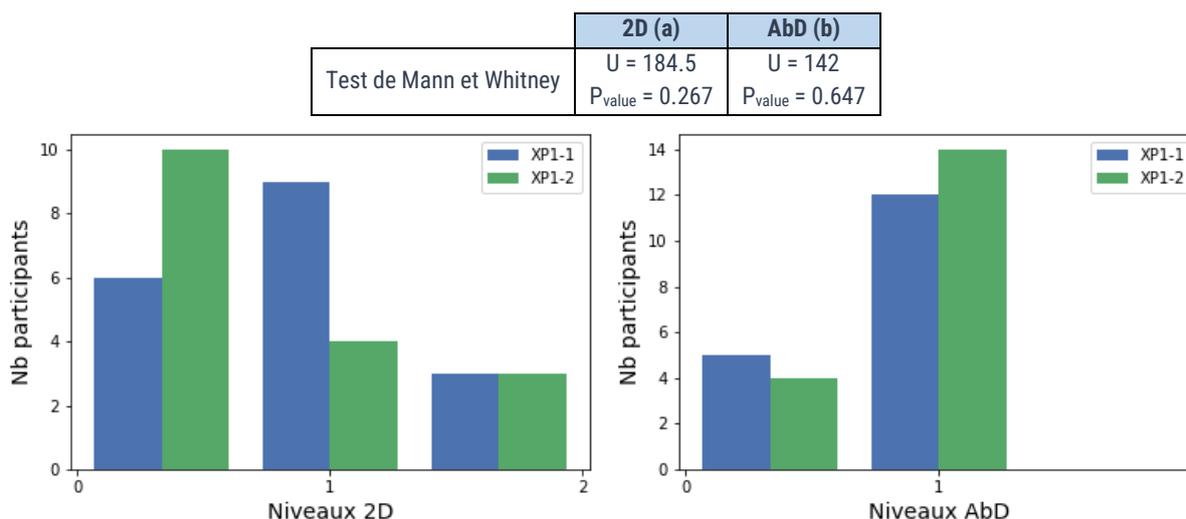


Figure 116 : (À gauche) Histogramme des niveaux d'édition selon le passage pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le passage pour l'édition AbD.

5.5.4.2.4.2 Discussion

Ainsi, même si nous pouvons observer une différence sur la figure 116 entre le 1^{er} et 2^{ème} passage, celle-ci n'est finalement pas statistiquement significative. Cela rejoint en partie notre analyse concernant le nombre d'erreurs. En effet, celle-ci n'est également pas statistiquement significative concernant le 1^{er} et 2^{ème} passage AbD. Cependant concernant l'éditeur 2D, le fait qu'il y ait une différence statistiquement significative sur le nombre d'erreurs ne semble pas avoir eu d'influence sur la différence de qualité entre le 1^{er} et 2^{ème} passage.

5.5.4.2.5 Synthèse

Pour conclure sur l'analyse de l'influence de l'ordre des outils d'éditeurs utilisés, le tableau 18 synthétise les résultats obtenus. Nous souhaitons notamment vérifier s'il y a eu un biais d'apprentissage entre les deux passages XP1-1 et XP1-2. Si l'on regarde le temps d'édition moyen, alors nous pouvons, en effet, conclure qu'il y a un biais d'apprentissage car celui-ci est plus faible durant le 2^{ème} passage et que la différence est statistiquement significative. Cependant, concernant le 2^{ème} passage sur l'édition 2D, nous pouvons nuancer ces propos en notant que les participants ayant utilisé l'authoring-by-doing au 1^{er} passage ont fait plus d'erreurs pour moins d'actions et pour un niveau moyen d'attente du scénario plus faible en utilisant l'édition 2D. Ainsi, ce gain de temps peut être dû aux participants ayant réalisé moins d'actions et n'ayant pas pris le temps de bien les paramétrer (comme expliqué dans la section 5.5.4.2.2). Le gain de temps peut aussi être lié aux temps de consultation des instructions d'assemblage moins important par rapport aux participants ayant commencé par l'édition 2D au 1^{er} passage.

À la différence, nous pouvons observer que les participants étant passé de l'éditeur 2D à l'authoring-by-doing ont fait moins d'erreurs mais plus d'actions pour un niveau global moyen plus élevé.

Tableau 18 : Synthèse des différences entre le 1^{er} et 2^{ème} passage pour l'édition 2D et l'authoring-by-doing. * représente les différences statistiquement significatives.

	2D			AbD		
	1 ^{er}	2 ^{ème}	Diff	1 ^{er}	2 ^{ème}	Diff
Temps moyen	428s	364s	-64s ↓*	175s	133s	-42s ↓*
Nombre d'erreurs moyennes	0.28	0.82	+0.54 ↑*	0.87	0.33	-0.54 ~
Nombre d'actions moyennes	16.4	15.3	-1.1 ~	7.35	7.77	+0.42 ↑*
Niveaux moyens	0.83	0.59	-0.24 ↓	0.71	0.78	+0.07 ↑

Afin de lisser l'influence de l'ordre de passages des outils d'éditeurs et le biais d'apprentissage, les résultats des deux passages de l'XP1 seront regroupés dans l'analyse présentée dans la section suivante.

5.5.4.3 Résultats de l'édition de scénario

Dans cette section, nous allons comparer la performance d'édition : le temps, le nombre d'erreurs, le nombre d'actions et le niveau du scénario réalisé en fonction du profil des participants. Pour chacune de ces quatre variables, nous comparerons les résultats entre les profils industriels et informatiques pour les deux moyens d'édition sur l'XP1 et l'XP2 puis entre l'édition 2D et l'authoring-by-doing pour l'XP1 puis entre l'édition 2D et l'authoring-by-doing

combiné à l'édition 2D pour l'XP2. Cette analyse nous permettra de répondre aux hypothèses $H_{2,édition}$ à $H_{9,édition}$.

5.5.4.3.1 Le niveau de qualité du scénario réalisé

5.5.4.3.1.1 Observations générales

Concernant le niveau de qualité du scénario, dans le tableau 19, nous pouvons observer que les moyennes entre les profils informatiques et industriels semblent proches uniquement sur l'XP1 avec l'authoring-by-doing. Pour le reste, les profils informatiques semblent réaliser, en moyenne, un scénario de meilleure qualité avec, pour l'édition 2D de l'XP1, **0.869** contre **0.417**. Puis sur l'XP2, **1.09** contre **0.5** pour l'édition 2D et **1.25** contre **0.833** pour l'authoring-by-doing combiné à l'édition 2D. Cependant, concernant la différence entre les outils d'édition, celle-ci est moins notable.

Tableau 19 : Statistique générale sur le niveau de qualité pour l'XP1 et l'XP2 par profil et outil d'édition.

XP1				XP2			
2D		AbD		2D		AbD + 2D	
INFO	INDUS	INFO	INDUS	INFO	INDUS	INFO	INDUS
Min = 0	Min = 0	Min = 0	Min = 0	Min = 0	Min = 0	Min = 1	Min = 0
Moy = 0.869	Moy = 0.417	Moy = 0.739	Moy = 0.75	Moy = 1.09	Moy = 0.5	Moy = 1.25	Moy = 0.833
Max = 2	Max = 2	Max = 1	Max = 1	Max = 2	Max = 1	Max = 2	Max = 2

Avant de vérifier si nos observations sur les différences et ressemblances sont statistiquement significatives, nous utiliserons un test de Mann et Whitney pour les tests concernant ces échantillons car il s'agit de comparer des valeurs allant de 0 à 2. Nous comparerons d'abord les deux profils de participants dans la section 5.5.4.3.1.2 puis les outils d'édition en 5.5.4.3.1.3.

5.5.4.3.1.2 Comparaison du niveau de qualité entre les profils industriels et informatiques

5.5.4.3.1.2.1 Pour un scénario simple (XP1)

Analyses statistiques :

Les résultats (Tableau 20) montre que pour l'édition 2D sur l'XP1, nous obtenons entre les profils industriels et informatiques une p-value de $0.083 > 0.05$ et pour l'authoring-by-doing, $0.963 > 0.05$.

Tableau 20 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les profils INFO et INDUS pour l'XP1.
* représente les différences statistiquement significatives.

XP1	
2D - INFO vs INDUS	AbD - INFO vs INDUS
U = 184.5	U = 136.5
P _{value} = 0.083	P _{value} = 0.963

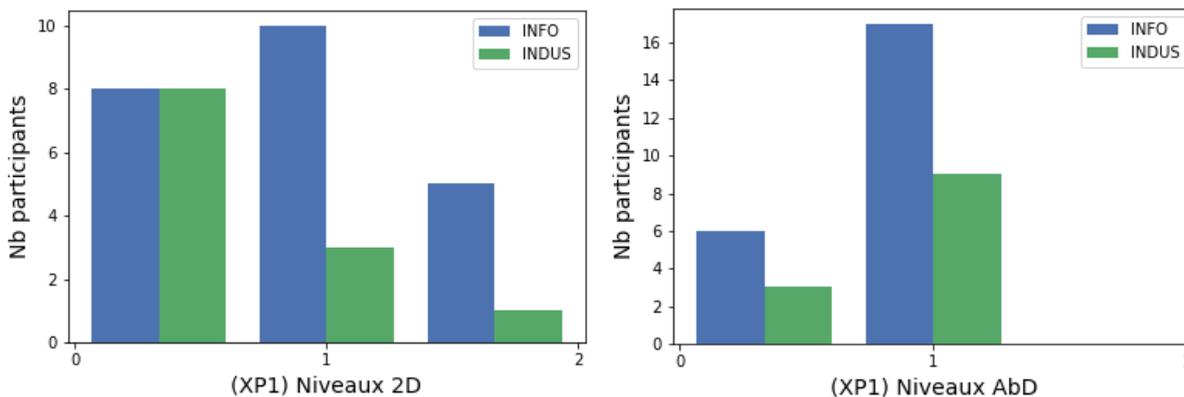


Figure 117 : Résultats de l'XP1. (À gauche) Histogramme des niveaux d'édition selon le profil pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le profil pour l'édition AbD.

Discussion :

En conclusion, notre observation sur le fait que les profils informatiques ont plus répondu aux attentes que les profils industriels sur l'édition d'un scénario simple en utilisant l'éditeur 2D n'est donc pas statistiquement valide. Nous pouvons considérer que concernant l'édition d'un scénario simple (XP1), les profils informatiques ou industriels ont répondu aux attentes avec un niveau similaire que ce soit avec l'éditeur 2D ou bien l'AbD.

5.5.4.3.1.2.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant l'XP2 avec un scénario complexe, les résultats (Tableau 21) donnent une p-value de $0.108 > 0.05$ pour l'édition 2D et une p-value de $0.196 > 0.05$ pour l'authoring-by-doing combiné à l'édition 2D. Les p-values sont supérieures à 0.05 (pas de différence significative).

Tableau 21 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les profils INFO et INDUS pour l'XP2. * représente les différences statistiquement significatives.

XP2	
2D - INFO vs INDUS	AbD+2D - INFO vs INDUS
U = 48	U = 48
P _{value} = 0.108	P _{value} = 0.196

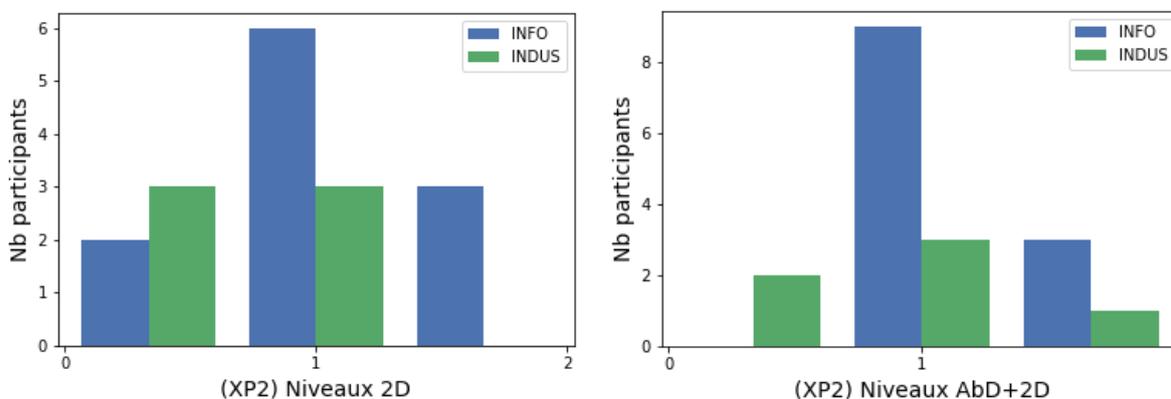


Figure 118 : Résultats de l'XP2. (À gauche) Histogramme des niveaux d'édition selon le profil pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le profil pour l'édition AbD+2D.

Discussion :

Nous pouvons considérer que, concernant l'édition d'un scénario complexe (XP2), les profils informatiques ou industriels ont répondu aux attentes avec un niveau similaire que ce soit avec l'éditeur 2D ou bien l'AbD + 2D. Les conclusions étant les mêmes pour un scénario simple, cela nous montre d'ores et déjà qu'un profil industriel arrive à répondre aux attentes concernant l'édition d'un scénario complexe aussi bien qu'un profil ayant des compétences en développement informatique.

Cependant, nous pouvons observer sur la figure 118, que seuls les profils informatiques ont proposé un scénario « dépassant les attentes » avec l'éditeur 2D et, de la même manière, ils sont majoritaires avec l'authoring-by-doing combiné à l'éditeur 2D. Cela nous amène à nuancer notre propos, car en effet, même si un profil industriel arrive à éditer un scénario donné en utilisant les outils d'édition, seules des connaissances en algorithmie permettent d'aller plus loin en termes de qualité et complexité de scénario.

5.5.4.3.1.3 Comparaison du niveau de qualité entre les outils d'édition

5.5.4.3.1.3.1 Pour un scénario simple (XP1)

Analyses statistiques :

Intéressons-nous maintenant à la comparaison entre l'édition 2D et l'authoring-by-doing pour un scénario simple. Comme pour la comparaison entre les profils, nous utiliserons donc un test de Mann et Whitney (Tableau 22). Pour les profils INFO sur l'XP1, nous obtenons entre l'édition 2D et l'authoring-by-doing une p-value de $0.634 > 0.05$, et pour l'authoring-by-doing, une p-value de $0.102 > 0.05$. Les p-values sont toutes supérieures à 0.05.

Tableau 22 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les différents outils d'édition pour l'XP1. * représente les différences statistiquement significatives.

XP1	
INFO - 2D vs AbD	INDUS - 2D vs AbD
U = 284.0	U = 46.5
P _{value} = 0.634	P _{value} = 0.102

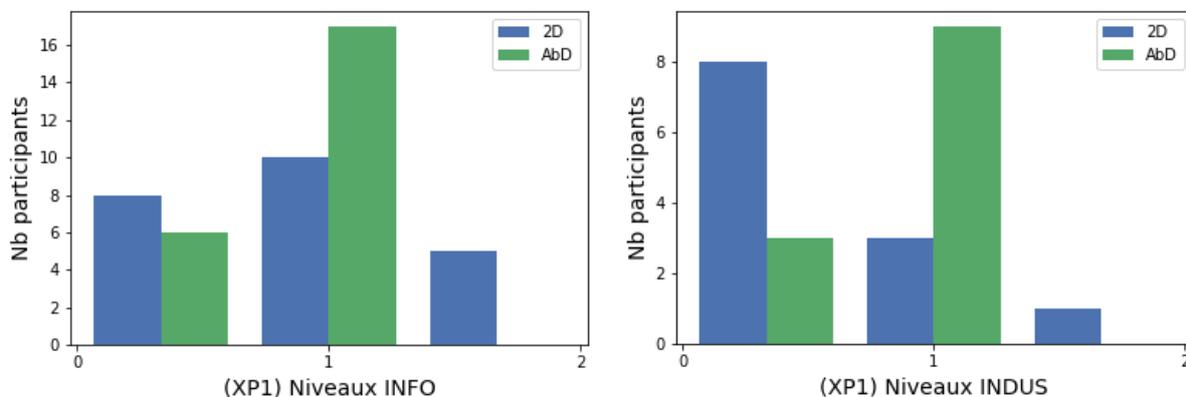


Figure 119 : Résultats de l'XP1. (À gauche) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INFO. (À droite) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INDUS.

Discussion :

Ainsi, concernant la comparaison des scénarios simples réalisés avec les différents outils d'éditeurs, il n'y a pas de différence statistiquement significative sur le niveau du scénario final réalisé. Il faut tout de même rappeler que seulement l'éditeur 2D permet une édition avancée offrant la possibilité de dépasser les attentes sur le scénario réalisé. En effet, même si la figure 119 montre qu'il y a plus de participants (INFO et INDUS) qui ont réalisé un scénario conforme (1) en utilisant l'AbD que l'éditeur 2D, nous arrivons à un équivalent en tenant compte des participants ayant dépassé les attentes.

Nous pouvons en conclure que les deux outils d'éditeurs ont permis d'éditer un scénario simple avec une qualité attendue similaire.

5.5.4.3.1.3.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre l'édition 2D et l'authoring-by-doing pour un scénario complexe, les résultats du test de Mann et Whitney dans le Tableau 23 donnent une p-value de $0.608 > 0.05$ pour les profils INFO et une p-value de $0.476 > 0.05$ pour les profils INDUS.

Tableau 23 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les différents outils d'édition pour l'XP2. * représente les différences statistiquement significatives.

XP2	
INFO - 2D vs AbD+2D	INDUS - 2D vs AbD+2D
U = 58.5	U = 13.5
P _{value} = 0.608	P _{value} = 0.476

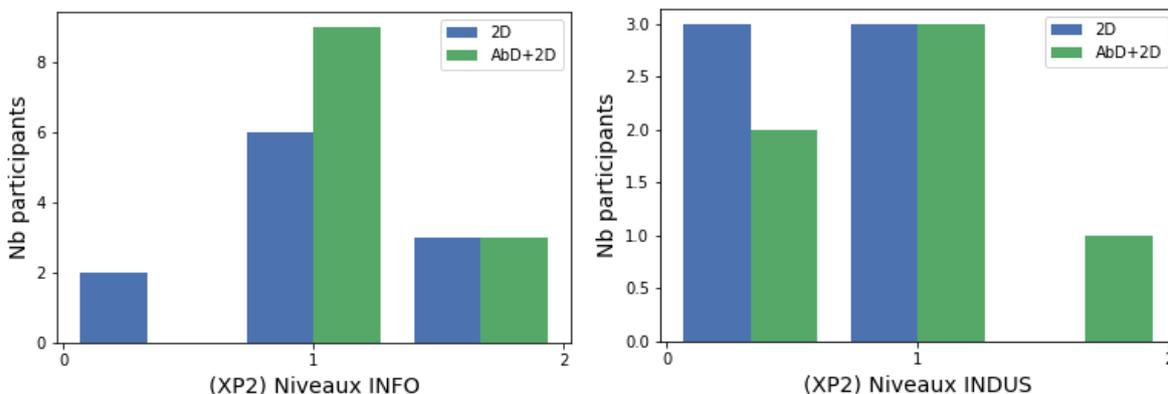


Figure 120 : Résultats de l'XP2. (À gauche) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INFO. (À droite) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INDUS.

Discussion :

Comme pour le scénario simple, nous pouvons noter que les p-values sont également toutes supérieures à 0.05. Ainsi, concernant la comparaison des scénarios complexes réalisés avec les différents outils d'éditeurs, il n'y a pas de différence statistiquement significative sur le niveau du scénario final réalisé. Nous pouvons en conclure que les deux outils d'éditeurs ont permis d'éditer un scénario complexe avec un niveau similaire.

Ainsi, que ce soit pour un scénario simple ou complexe, ces outils d'édition ont permis d'écrire un scénario répondant aux consignes pour la majorité des participants. L'hypothèse $H_{1, \text{édition}}$: « l'éditeur 2D et l'authoring-by-doing permettent d'écrire un scénario » peut donc être validée.

5.5.4.3.2 Les erreurs commises sur le scénario final

5.5.4.3.2.1 Observations générales

Le niveau de qualité analysé dans la section précédente est étroitement lié aux erreurs commises, c'est pourquoi nous poursuivons avec l'analyse du nombre d'erreurs. Dans le tableau 24, nous pouvons déjà noter, qu'en moyenne, les profils industriels ont fait plus d'erreurs sur le scénario final à l'XP1 et l'XP2. Sur l'XP1, les profils industriels ont fait en moyenne **1.08** erreurs sur l'édition 2D contre **0.26** pour les profils informatiques et **0.5** erreurs sur l'authoring-by-doing contre **0.348**. De même que sur l'XP2, ils ont fait en moyenne **1.166** erreurs sur l'édition 2D contre **0.636** pour les profils informatiques et **0.666** sur l'authoring-by-doing combinée à l'édition 2D contre **0.333**.

Concernant les outils d'édition, nous noterons que sur l'XP1, les profils informatiques ont fait légèrement plus d'erreurs en utilisant l'authoring-by-doing (**0.348**) que l'édition 2D (**0.26**). Concernant les profils industriels, eux ont fait en moyenne moins d'erreurs avec l'authoring-by-doing (**0.5**) qu'avec l'édition 2D (**1.08**). Sur l'XP2, nous passons de **0.636** pour les profils informatiques ayant utilisé l'éditeur 2D à **0.333** en AbD+2D. De même pour les profils industriels qui passent de **1.166** à **0.666**, soit dans les deux cas, une réduction quasiment de moitié.

Tableau 24 : Statistique générale sur le nombre d'erreurs pour l'XP1 et l'XP2 par profil et outil d'édition.

XP1				XP2			
2D		AbD		2D		AbD + 2D	
INFO	INDUS	INFO	INDUS	INFO	INDUS	INFO	INDUS
Min = 0	Min = 0	Min = 0	Min = 0	Min = 0	Min = 0	Min = 0	Min = 0
Moy = 0.26	Moy = 1.08	Moy = 0.348	Moy = 0.5	Moy = 0.636	Moy = 1.166	Moy = 0.333	Moy = 0.666
Max = 3	Max = 3	Max = 2	Max = 3	Max = 4	Max = 4	Max = 4	Max = 2

Pour la suite, nous comparerons d'abord les deux profils de participants dans la section 5.5.4.3.2.2 puis les outils d'édition en 5.5.4.3.2.3.

5.5.4.3.2.2 Comparaison des erreurs commises pour chaque éditeur en fonction des profils des participants

5.5.4.3.2.2.1 Pour un scénario simple (XP1)

Analyses statistiques :

Concernant la comparaison entre les profils industriels et informatiques pour un scénario simple, le tableau 25 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances est validée. Cependant, deux des échantillons ne suivant pas une loi normale, les conditions pour effectuer un t-test ne sont donc pas valides. Nous utiliserons donc un test de Mann et Whitney.

Tableau 25 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP1 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 1.763 P _{value} = 0.192		F = 0.298 P _{value} = 0.589	
Hypothèse de normalité (Test de Shapiro)	W = 0.441 P _{value} = 2.4e-8*	W = 0.865 P _{value} = 0.056	W = 0.590 P _{value} = 0.07	W = 0.962 P _{value} = 7e-7*

Les résultats du test de Mann et Whitney (Tableau 26) retournent une p-value inférieure à 0.05 uniquement sur la comparaison entre les profils ayant utilisé l'éditeur 2D. Il y a ainsi une différence statistiquement significative entre le nombre d'erreurs commises par les profils informatiques et industriels pour la réalisation d'un scénario simple avec l'éditeur 2D.

Tableau 26 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs pour l'XP1 entre les profils INFO et INDUS pour chaque outil d'édition. * représente les différences statistiquement significatives.

XP1	
2D - INFO vs INDUS	AbD - INFO vs INDUS
U = 58 P _{value} = 0.0006*	U = 135.5 P _{value} = 0.464

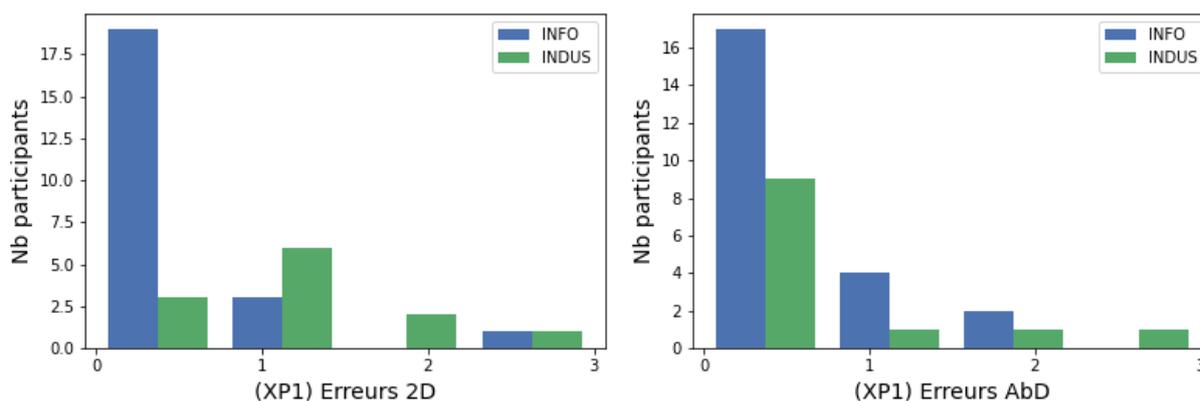


Figure 121 : Résultats de l'XP1. (À gauche) Histogramme du nombre d'erreurs selon le profil pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le profil pour l'édition AbD.

Discussion :

Ainsi, nous pouvons affirmer que les profils industriels ont, en moyenne, commis plus d'erreurs sur un scénario simple réalisé avec l'éditeur 2D par rapport aux profils informatiques. L'édition authoring-by-doing semble alors plus recommandée pour les profils industriels. Comme expliqué dans la section 5.5.4.2.2.2, les paramètres des étapes sont complétés automatiquement lors de la réalisation de l'action en AbD tandis qu'avec l'édition 2D, ces mêmes valeurs doivent être remplies manuellement. Cela cause alors des erreurs. Pour régler ce problème, l'éditeur 2D doit évoluer afin de mieux accompagner l'édition avec notamment un pré-remplissage des paramètres ainsi qu'une visualisation des actions paramétrées.

5.5.4.3.2.2.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre les profils industriels et informatiques pour un scénario complexe, les résultats du test de Levene et du test de Shapiro présentés dans le Tableau 27

montrent que les conditions pour réaliser un t-test ne sont pas remplies. Nous utiliserons donc un test de Mann et Whitney.

Tableau 27 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP2 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD+2D	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 0.685 P _{value} = 0.421		F = 0.355 P _{value} = 0.559	
Hypothèse de normalité (Test de Shapiro)	W = 0.592 P _{value} = 2 ^e -5 *	W = 0.809 P _{value} = 0.07	W = 0.327 P _{value} = 1 ^e -6 *	W = 0.640 P _{value} = 0.001 *

Les résultats du test de Mann et Whitney (Tableau 28) montrent qu'il n'y a pas de différence statistiquement significative en termes d'erreurs d'édition.

Tableau 28 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs pour l'XP2 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.

XP2	
2D - INFO vs INDUS	AbD+2D - INFO vs INDUS
U = 25.5 P _{value} = 0.204	U = 28 P _{value} = 0.139

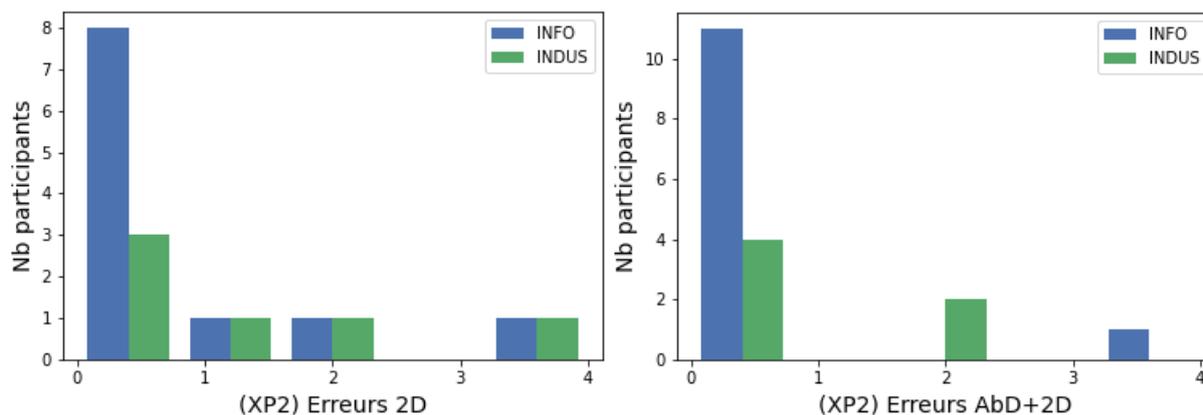


Figure 122 : Résultats de l'XP2. (À gauche) Histogramme du nombre d'erreurs selon le profil pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le profil pour l'édition AbD+2D.

Discussion :

Contrairement au scénario simple, nous notons que les moyennes du nombre d'erreurs sont similaires entre un profil industriel et un profil informatique que ce soit sur l'éditeur 2D ou l'AbD+2D. Cependant, nous ne pouvons alors que valider partiellement l'hypothèse H_{10,édition} « le nombre d'erreurs via l'éditeur 2D est similaire entre un profil industriel et un profil informatique » car même si elle est validée pour un scénario complexe, celle-ci n'est pas valide pour un scénario simple. Cependant, l'hypothèse H_{11,édition} « le nombre d'erreurs via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique » est qu'en à elle validée pour un scénario simple et complexe.

5.5.4.3.2.3 Comparaison des erreurs commises pour chaque profil de participants en fonction des outils d'édition

5.5.4.3.2.3.1 Pour un scénario simple (XP1)

Analyses statistiques :

Concernant la comparaison entre les outils d'éditions pour un scénario simple, les résultats du test de Levene et du test de Shapiro présentés dans le tableau 29 montrent que les conditions pour réaliser un t-test ne sont pas remplies. Nous utiliserons donc un test de Mann et Whitney.

Tableau 29 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP1 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD	2D	AbD
Homogénéité des variances (Test de Levene)	F = 0.195 P _{value} = 0.661		F = 0.057 P _{value} = 0.812	
Hypothèse de normalité (Test de Shapiro)	W = 0.441 P _{value} = 2.4e-8 *	W = 0.590 P _{value} = 0.07	W = 0.865 P _{value} = 0.056	W = 0.962 P _{value} = 7e-7 *

Les résultats du test de Mann et Whitney (Tableau 30) nous donnent donc pour les profils INFO sur l'XP1, entre l'édition 2D et l'autoring-by-doing une p-value de $0.245 > 0.05$, et pour les profils INDUS, une p-value de $0.028 < 0.05$.

Tableau 30 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs entre les différents outils d'édition pour chaque profil. * représente les différences statistiquement significatives.

XP1	
INFO - 2D vs AbD	INDUS - 2D vs AbD
U = 241.5 P _{value} = 0.245	U = 41 P _{value} = 0.028*

La p-value est inférieure à 0.05 uniquement sur la comparaison entre les outils d'édition pour les profils INDUS de l'XP1. Donc concernant la comparaison entre les outils d'éditions, notre remarque sur le fait que les profils industriels ont fait en moyenne moins d'erreurs avec l'autoring-by-doing qu'avec l'édition 2D est statistiquement valide.

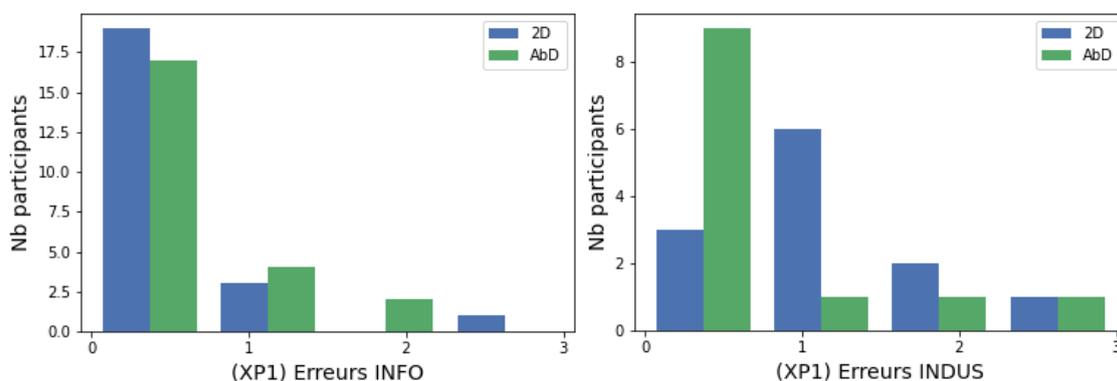


Figure 123 : Résultats de l'XP1. (À gauche) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INFO. (À droite) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INDUS.

Discussion :

Ce que nous pouvons en conclure est que l’authoring-by-doing se démarque pour les profils industriels lors de l’édition d’un scénario simple. En effet, ces derniers ont commis moins d’erreurs en moyenne avec l’AbD qu’avec l’éditeur 2D et cette différence est statistiquement significative. A contrario, les profils informatiques étant plus habitués à des interfaces similaires à l’édition, la différence est beaucoup moins notable, voir similaire comme le montre la figure 123.

L’hypothèse $H_{12.edition}$ « l’authoring-by-doing réduit le nombre d’erreurs commises pour un scénario simple par rapport à l’éditeur 2D » ne peut donc être que partiellement validée.

5.5.4.3.2.3.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Ensuite, concernant la comparaison entre l’édition 2D et l’authoring-by-doing pour un scénario complexe, les résultats présentés dans le tableau 31 montrent que les conditions pour réaliser un t-test ne sont pas validées.

Tableau 31 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d’erreurs de l’XP2 entre les différents outils d’édition. * indique que l’hypothèse n’est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD+2D	2D	AbD+2D
Homogénéité des variances (Test de Levene)	F = 0.354 P _{value} = 0.558		F = 0.592 P _{value} = 0.459	
Hypothèse de normalité (Test de Shapiro)	W = 0.592 P _{value} = 2e-5*	W = 0.327 P _{value} = 1e-6*	W = 0.809 P _{value} = 0.07	W = 0.640 P _{value} = 0.001*

Comme pour la comparaison entre les profils, nous utiliserons donc un test de Mann et Whitney (Tableau 32). Les résultats donnent des p-values qui sont toutes supérieures à 0.05.

Tableau 32 : Résultats du test de Mann et Whitney pour les moyennes du nombre d’erreurs entre les différents outils d’édition pour l’XP2. * représente les différences statistiquement significatives.

XP2	
INFO - 2D vs AbD+2D	INDUS - 2D vs AbD+2D
U = 77.5 P _{value} = 0.305	U = 21 P _{value} = 0.652

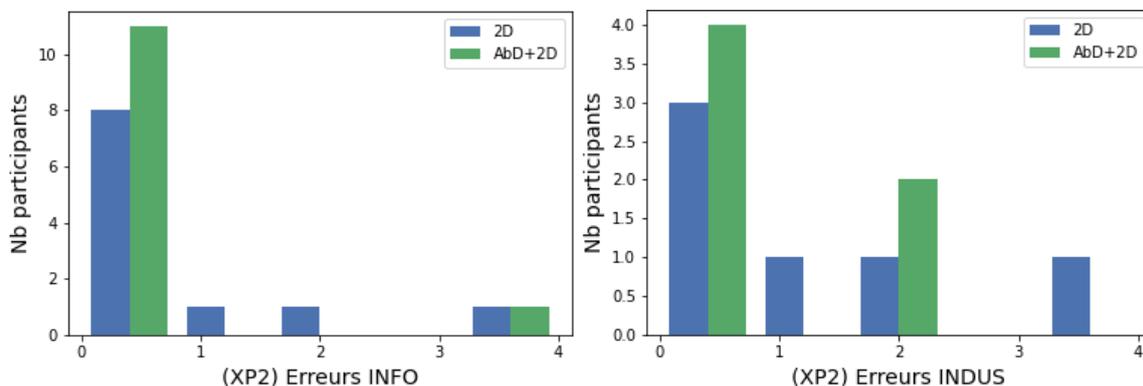


Figure 124 : Résultats de l’XP2. (À gauche) Histogramme du nombre d’erreurs selon l’outils d’édition pour les profils INFO. (À droite) Histogramme du nombre d’erreurs selon l’outils d’édition pour les profils INDUS.

Discussion :

Concernant les scénarios complexes, la différence entre les outils d'éditions n'est pas statistiquement significative. L'hypothèse $H_{13.édition}$ « l'authoring-by-doing réduit le nombre d'erreurs commises pour un scénario complexe par rapport à l'authoring-by-doing associé à l'éditeur 2D » ne peut donc pas être validée. Ainsi, nous pouvons dire que la multimodalité (AbD+2D) n'a pas créé de confusion supplémentaire vis-à-vis de l'utilisation uniquement de l'édition 2D pour réaliser un scénario complexe.

5.5.4.3.3 Le temps d'édition**5.5.4.3.3.1 Observations générales**

Commençons par l'analyse des durées d'édition. Dans le tableau 33, nous pouvons d'ores et déjà observer que les moyennes entre le profil informatique et industriel sont proches sur l'XP1 avec une différence de **18s** sur l'édition 2D (5%) et de **13s** pour l'AbD (9%). Elles sont plus éloignées sur l'XP2 avec une différence de **81s** sur l'édition 2D (21%) et de **73s** pour l'AbD+2D (20%).

Aussi, concernant les outils d'édition, nous pouvons noter qu'il y a un gain de temps moyen en faveur de l'AbD comparé à l'édition 2D. Sur l'XP1, nous passons de **391s** pour les profils informatiques ayant utilisé l'éditeur 2D à **158s** en AbD, soit **un gain de -233s** ($\approx 60\%$) et un temps **divisé par ≈ 2.5** et de **409s** à **145s** pour les profils industriels, soit **un gain de -264s** ($\approx 65\%$) et un temps d'édition **divisé par ≈ 2.8** . Sur l'XP2, nous passons de **456s** pour les profils informatiques ayant utilisés l'éditeur 2D à **279s** en AbD+2D, soit **un gain de -177s** ($\approx 40\%$) et un temps **divisé par ≈ 1.6** . Cependant, les moyennes de temps pour les profils industriels passent de **375s** à **352s**, soit un de gain seulement **-23s** ($\approx 6\%$).

Tableau 33 : Statistique générale sur le temps d'édition pour l'XP1 et l'XP2 par profil et outil d'édition.

XP1				XP2			
2D		AbD		2D		AbD + 2D	
INFO	INDUS	INFO	INDUS	INFO	INDUS	INFO	INDUS
Min = 228s	Min = 262s	Min = 53s	Min = 65s	Min = 332s	Min = 149s	Min = 126s	Min = 228s
Moy = 391s	Moy = 409s	Moy = 158s	Moy = 145s	Moy = 456s	Moy = 375s	Moy = 279s	Moy = 352s
Max = 628s	Max = 624s	Max = 331s	Max = 259s	Max = 638s	Max = 542s	Max = 478s	Max = 502s

Pour la suite, nous comparerons tout d'abord les deux profils de participants dans la section 5.5.4.3.3.2 puis les outils d'édition en 5.5.4.3.3.3.

5.5.4.3.3.2 Comparaison du temps d'édition entre les profils industriels et informatiques**5.5.4.3.3.2.1 Pour un scénario simple (XP1)****Analyses statistiques :**

Concernant la comparaison entre les profils industriels et informatiques pour un scénario simple, le tableau 34 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances est ainsi validée. Concernant l'hypothèse de normalité, seul l'échantillon « 2D_INFO » ne suit pas une loi normale. Nous utiliserons donc un test de Mann et Whitney pour les tests concernant cet échantillon.

Tableau 34 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP1 entre les profils INFO et INDUS par outil d'édition. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 0.11 P _{value} = 0.739		F = 3.25 P _{value} = 0.08	
Hypothèse de normalité (Test de Shapiro)	W = 0.912 P _{value} = 0.045*	W = 0.865 P _{value} = 0.056	W = 0.922 P _{value} = 0.072	W = 0.962 P _{value} = 0.811

Les conditions pour effectuer un t-test sont donc valides sauf pour « 2D_INFO ». Nous utiliserons donc un test de Mann et Whitney pour ce dernier (Tableau 35). Aussi bien pour le test de Mann et Whitney que pour le t-test, nous obtenons entre les profils industriels et informatiques des p-values supérieures à 0.05. Il n'y a donc pas de différence statistiquement significative au niveau du temps d'édition entre les profils industriels et informatiques pour la réalisation d'un scénario simple que ce soit sur l'édition 2D ou l'authoring-by-doing.

Tableau 35 : Résultats du test de Mann et Whitney et du t-test pour les temps d'édition entre les profils INFO et INDUS pour l'XP1.

Mann et Whitney	T-test
2D - INFO vs INDUS	AbD - INFO vs INDUS
U = 121.5	T = 0.487
P _{value} = 0.289	P _{value} = 0.629

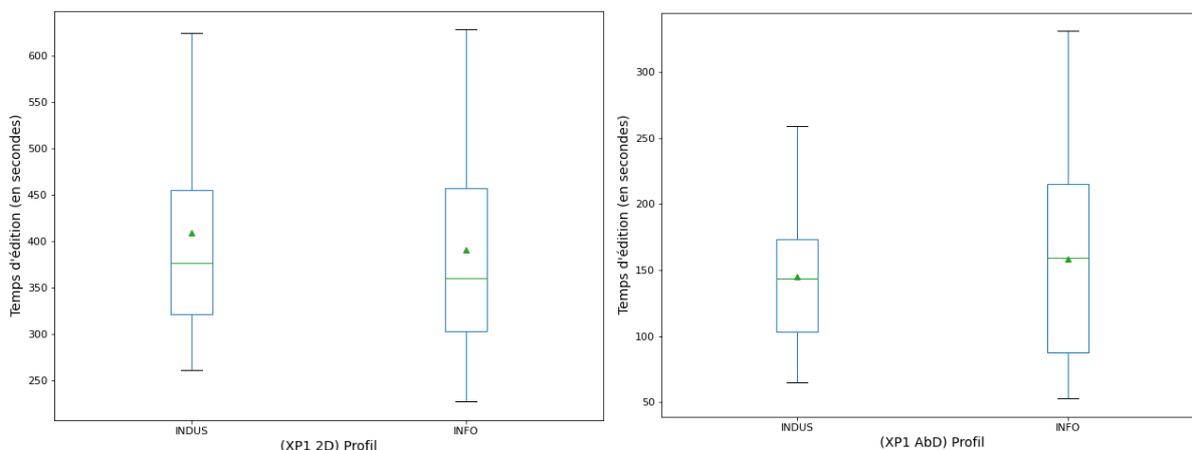


Figure 125 : Résultats de l'XP1. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition AbD.

Discussion :

Comme il n'y a pas de différence statistiquement significative, les temps d'édition peuvent être considérés comme les mêmes entre INFO et INDUS. Les hypothèses $H_{2.édition}$ « les temps d'éditions via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique » et $H_{3.édition}$ « les temps d'éditions via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique » sont donc valides concernant un scénario simple. Cela montre que ces outils rendent accessible la création de procédure à des non-développeurs, les rendant ainsi capables d'éditer un scénario dans un temps similaire à des profils informatiques.

5.5.4.3.3.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre les profils industriels et informatiques pour un scénario complexe, le tableau 36 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances et l'hypothèse de normalité sont validées.

Tableau 36 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP2 entre les profils INFO et INDUS par outil d'édition. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD + 2D	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 0.68 P _{value} = 0.42		F = 1.60 P _{value} = 0.223	
Hypothèse de normalité (Test de Shapiro)	W = 0.953 P _{value} = 0.684	W = 0.947 P _{value} = 0.716	W = 0.921 P _{value} = 0.294	W = 0.897 P _{value} = 0.356

Les conditions pour effectuer un t-test sont donc valides. Les résultats présents dans le tableau 37 indique qu'il n'y a pas de différence statistiquement significative au niveau du temps d'édition entre les profils industriels et informatiques pour la réalisation d'un scénario complexe que ce soit sur l'édition 2D ou l'authoring-by-doing.

Tableau 37 : Résultats du t-test pour les temps d'édition entre les profils INFO et INDUS pour l'XP2.

T-test	
2D - INFO vs INDUS	AbD+2D - INFO vs INDUS
T = 1.494 P _{value} = 0.156	T = 1.316 P _{value} = 0.207

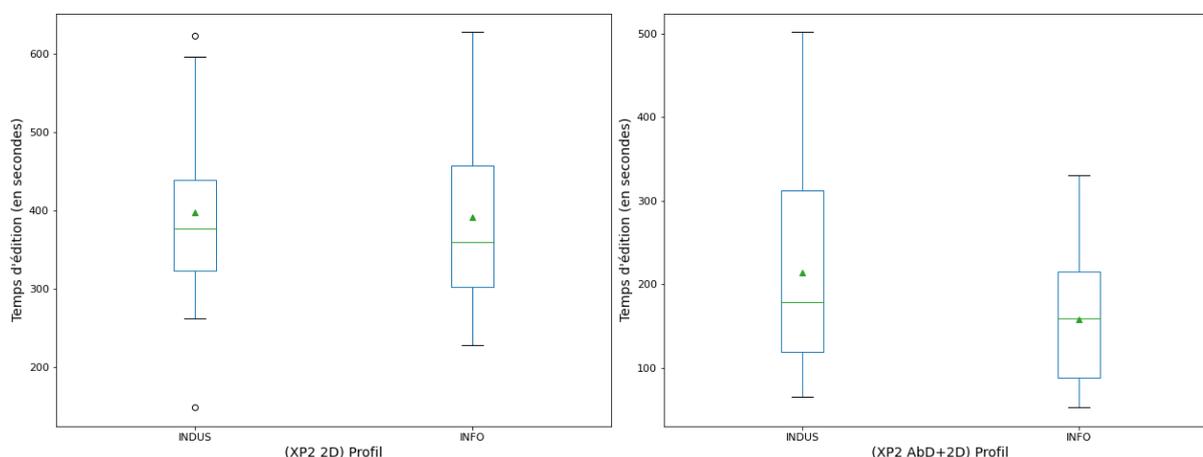


Figure 126 : Résultats de l'XP2. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition AbD+2D.

Discussion :

Comme il n'y a pas de différence statistiquement significative, les hypothèses H_{2,édition} « les temps d'éditions via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique » et H_{3,édition} « les temps d'éditions via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique » sont donc valides concernant un scénario complexe également.

5.5.4.3.3.3 Comparaison du temps d'édition entre les outils AbD et 2D

5.5.4.3.3.3.1 Pour un scénario simple (XP1)

Analyse statistique :

Concernant la comparaison entre l'édition 2D et l'authoring-by-doing, les résultats des tests de Levene et de Shapiro (Tableau 38) montrent que l'homogénéité des variances est validée et que seule l'hypothèse de normalité n'est pas valide concernant l'échantillon « 2D_INFO ».

Tableau 38 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP1 entre les différents outils d'édition par profils. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD	2D	AbD
Homogénéité des variances (Test de Levene)	F = 2.605 P _{value} = 0.114		F = 2.634 P _{value} = 0.119	
Hypothèse de normalité (Test de Shapiro)	W = 0.912 P _{value} = 0.045*	W = 0.922 P _{value} = 0.072	W = 0.865 P _{value} = 0.056	W = 0.962 P _{value} = 0.811

Comme pour la comparaison entre les profils, les conditions pour effectuer un t-test sont donc valides sauf pour « 2D_INFO ». Nous utiliserons donc un test de Mann et Whitney pour ce dernier (Tableau 39). Pour les deux résultats, nous obtenons qu'il y a une différence statistiquement significative.

Tableau 39 : Résultats du test de Mann et Whitney et du T-Test pour les temps d'édition entre les différents outils d'édition pour l'XP1. * représente les différences statistiquement significatives.

Mann et Whitney	T-test
INFO 2D vs AbD	INDUS 2D vs AbD
U = 19 P _{value} = 3.67e-08*	T = 6.752 P _{value} = 8.71e-07*

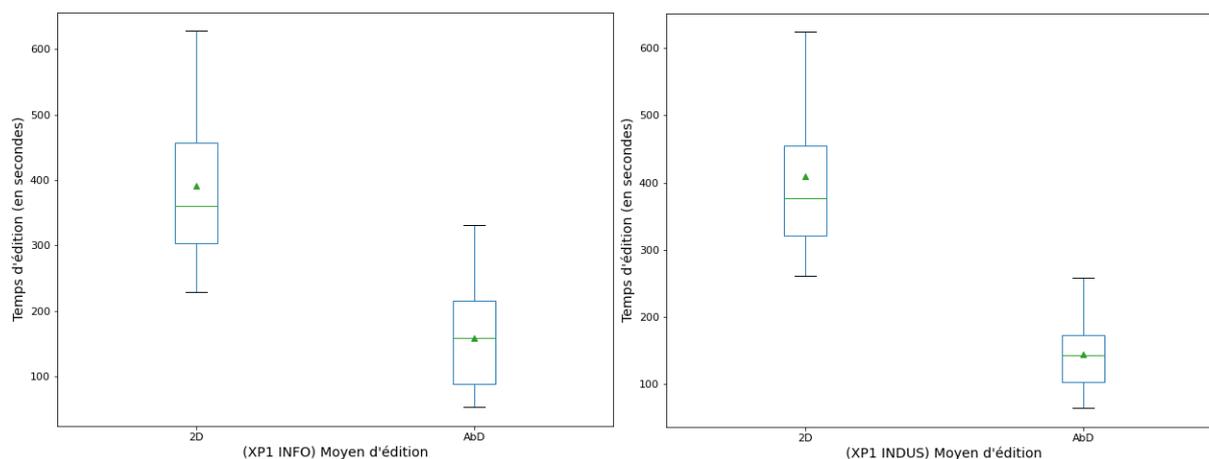


Figure 127 : Résultats de l'XP1. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition AbD.

Discussion :

Les p-values sont inférieures à 0.05 donc il y a bien une différence statistiquement significative au niveau du temps d'édition entre l'édition 2D et l'authoring-by-doing (Figure 127). En ce qui concerne l'XP1, permettant aux participants d'éditionner un scénario procédural simple sans notion

de parallélisme, l'analyse statistique montre que la différence de temps d'édition de procédure en fonction de l'éditeur utilisé est statistiquement significative pour les deux profils, industriels et informatique. Cela montre que l'outil AbD est à recommander pour définir des procédures simples quels que soient les profils. En effet, ce mode AbD permet de définir automatiquement les paramètres des tâches simplement en le faisant. Par ailleurs, cela confirme que les éditeurs rendent accessibles la définition de procédure à des profils industriels.

Pour finir, nous pouvons affirmer que l'édition en authoring-by-doing, qu'elle soit utilisée seule ou combinée à l'édition 2D, permet de réduire le temps d'édition pour les scénarios procéduraux ou parallèles, et cela peu importe le profil du participant. L'hypothèse $H_{4.édition}$ « l'authoring-by-doing réduit le temps mis pour éditer un scénario simple par rapport à l'éditeur 2D » est ainsi validée.

5.5.4.3.3.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre l'édition 2D et l'AbD+2D pour un scénario complexe, le tableau 40 montre que l'homogénéité des variances ainsi que l'hypothèse de normalité sont validées.

Tableau 40 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition entre les différents outils d'édition pour l'XP2. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD + 2D	2D	AbD + 2D
Homogénéité des variances (Test de Levene)	F = 1.078 P _{value} = 0.311		F = 1.023 P _{value} = 0.336	
Hypothèse de normalité (Test de Shapiro)	W = 0.953 P _{value} = 0.684	W = 0.921 P _{value} = 0.294	W = 0.947 P _{value} = 0.716	W = 0.897 P _{value} = 0.356

Les conditions pour effectuer un t-test sont donc valides. Les résultats (Tableau 41) indiquent qu'il y a une différence statistiquement significative uniquement concernant les profils informatiques.

Tableau 41 : Résultats du test du t-test pour les temps d'édition entre les différents outils d'édition de l'XP2. * représente les différences statistiquement significatives.

T-test	
INFO 2D vs AbD+2D	INDUS 2D vs AbD+2D
T = 4.005 P _{value} = 0.0006*	T = 0.365 P _{value} = 0.723

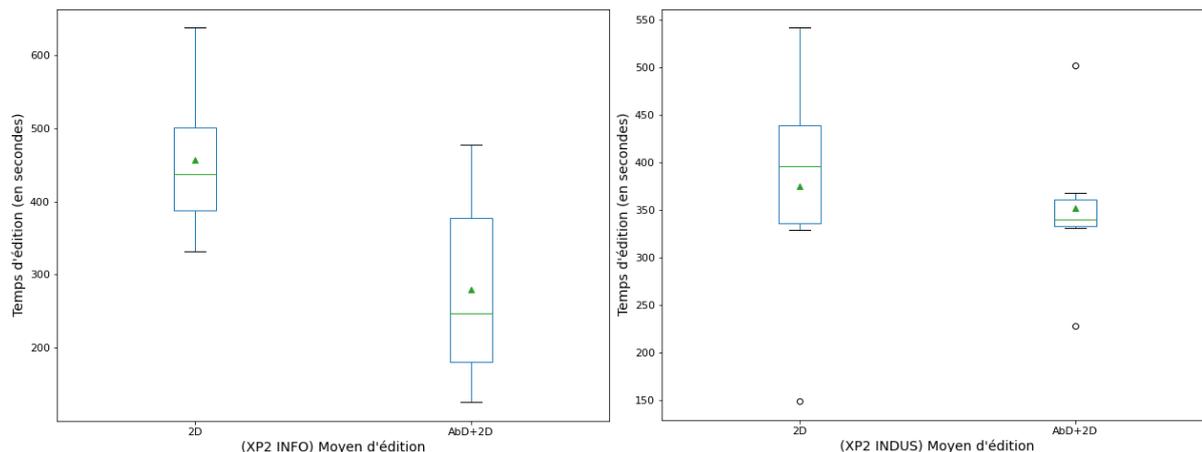


Figure 128 : Résultats de l'XP2. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition AbD+2D.

Discussion :

En ce qui concerne l'XP2, permettant aux participants d'éditer un scénario complexe avec la notion de parallélisme, l'analyse statistique montre que la différence de temps d'édition de procédure en fonction de l'éditeur utilisé est statistiquement significative pour seulement les profils informatiques. Suite à notre discussion sur un scénario simple, nous pouvons supposer que le fait qu'il n'y ait pas de différence significative concernant les profils industriels est dû à l'usage de l'éditeur 2D avec l'AbD. Cependant, nous pouvons noter sur la figure 128 que, même si les moyennes sont statistiquement similaires pour les profils INDUS, elles sont beaucoup plus concentrées sur l'édition 2D combinée avec l'authoring-by-doing qu'avec l'édition 2D. L'hypothèse $H_{5,édition}$ « l'authoring-by-doing réduit le temps mis pour éditer un scénario complexe par rapport à l'éditeur 2D », ne peut donc être validée que pour les profils INFO.

5.5.4.3.4 Le nombre d'action faites

5.5.4.3.4.1 Observations générales

Analysons ensuite le nombre d'action faites. Dans le tableau 42, nous pouvons d'ores et déjà observer des tendances. Les moyennes entre le profil informatique et industriel sont proches sur l'XP1 avec, réciproquement, **15.78** et **16.08** actions en moyenne sur l'édition 2D ainsi que **7.52** et **7.67** pour l'AbD. Elles sont plus éloignées sur l'XP2 avec **22.72** et **19.5** sur l'édition 2D et **25.42** et **35.5** pour l'AbD+2D.

Ensuite, concernant les outils d'édition, les participants ont tendance à faire moins d'actions, sur l'XP1, en utilisant l'authoring-by-doing comparé à l'édition 2D. Tandis que sur l'XP2, la combinaison de l'authoring-by-doing avec l'édition requière d'effectuer plus d'actions que l'édition 2D.

Tableau 42 : Statistique générale sur le nombre d'actions pour l'XP1 et l'XP2 par profil et outil d'édition.

XP1				XP2			
2D		AbD		2D		AbD + 2D	
INFO	INDUS	INFO	INDUS	INFO	INDUS	INFO	INDUS
Min = 12	Min = 12	Min = 6	Min = 6	Min = 13	Min = 16	Min = 11	Min = 21

Moy = 15.78 Max = 21	Moy = 16.08 Max = 21	Moy = 7.52 Max = 19	Moy = 7.67 Max = 13	Moy = 22.72 Max = 37	Moy = 19.5 Max = 21	Moy = 25.42 Max = 35	Moy = 35.5 Max = 52
--------------------------------	--------------------------------	-------------------------------	-------------------------------	--------------------------------	-------------------------------	--------------------------------	-------------------------------

Pour la suite, nous comparerons d'abord les deux profils de l'ensemble des participants dans la section 5.5.4.3.4.2 puis les outils d'édition en 5.5.4.3.4.3.

5.5.4.3.4.2 Comparaison du nombre d'actions faites en fonction du profil des participants

5.5.4.3.4.2.1 Pour un scénario simple (XP1)

Analyses statistiques :

Concernant la comparaison entre les profils industriels et informatiques pour un scénario simple, le tableau 43 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances est ainsi validée. Cependant, concernant l'hypothèse de normalité, seuls les échantillons de l'édition 2D suivent une loi normale. Nous utiliserons donc un test de Mann et Whitney pour les tests concernant l'AbD.

Tableau 43 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP1 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 0.0104 P _{value} = 0.919		F = 0.0006 P _{value} = 0.981	
Hypothèse de normalité (Test de Shapiro)	W = 0.919 P _{value} = 0.063	W = 0.918 P _{value} = 0.270	W = 0.580 P _{value} = 5.56e-07*	W = 0.775 P _{value} = 0.0049*

Ainsi les résultats du t-test pour l'édition 2D et du test de Mann et Whitney pour l'AbD (Tableau 44), nous retournent des p-values supérieures à 0.05. Il n'y a pas de différence statistiquement significative sur le nombre d'actions faites pour réaliser un scénario simple entre un profil INFO et INDUS.

Tableau 44 : Résultats du test de Mann et Whitney et du T-Test pour les moyennes du nombre d'actions de l'XP1 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.

T-test	Mann et Whitney
2D - INFO vs INDUS	AbD - INFO vs INDUS
T = 0.286 P _{value} = 0.776	U = 121.5 P _{value} = 0.274

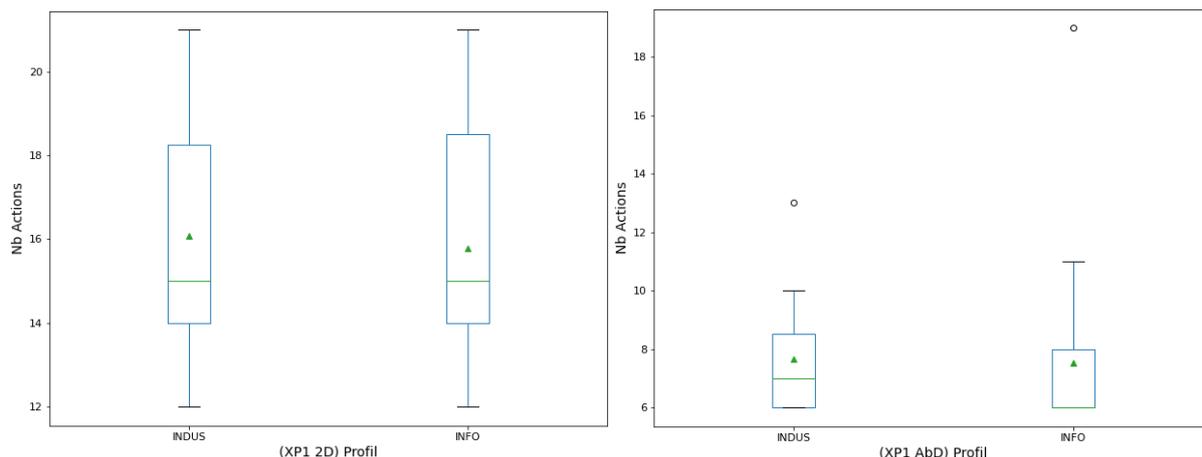


Figure 129 : Résultats de l'XP1. (À gauche) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition 2D. (À droite) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition Abd.

Discussion :

Cette analyse nous montre qu'un expert métier industriel est capable, sans connaissance en développement informatique, de réaliser un scénario procédural simple (sans parallélisme) avec un même nombre d'action que des développeurs informatiques, et cela avec les deux outils d'édition proposés. Les hypothèses $H_{6,édition}$ « le nombre d'actions via l'éditeur 2D est similaire entre un profil industriel et un profil informatique » et $H_{7,édition}$ « le nombre d'actions via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique » sont validées concernant l'édition d'un scénario simple.

5.5.4.3.4.2.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre les profils industriels et informatiques pour un scénario complexe, le tableau 46 donne les résultats du test de Levene et du test de Shapiro. L'homogénéité des variances n'est ainsi pas validée pour l'AbD + 2D. De plus, concernant l'hypothèse de normalité, les échantillons 2D ne suivent pas une loi normale. Les conditions de réalisation d'un t-test n'étant pas valides, nous utiliserons donc un test de Mann et Whitney pour les tests concernant l'XP2.

Tableau 45 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP2 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	2D		AbD + 2D	
	INFO	INDUS	INFO	INDUS
Homogénéité des variances (Test de Levene)	F = 1.822 P _{value} = 0.197		F = 8.844 P _{value} = 0.009*	
Hypothèse de normalité (Test de Shapiro)	W = 0.842 P _{value} = 0.034*	W = 0.667 P _{value} = 0.003*	W = 0.940 P _{value} = 0.504	W = 0.909 P _{value} = 0.432

Nous obtenons donc comme résultat (Tableau 46) qu'il y a une différence statistiquement significative entre les profils INFO et INDUS sur l'éditeur 2D concernant le nombre d'actions pour réaliser un scénario complexe.

Tableau 46 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions de l'XP2 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.

Mann et Whitney	
2D - INFO vs INDUS	AbD+2D - INFO vs INDUS
U = 15	U = 21.5
P _{value} = 0.034*	P _{value} = 0.094

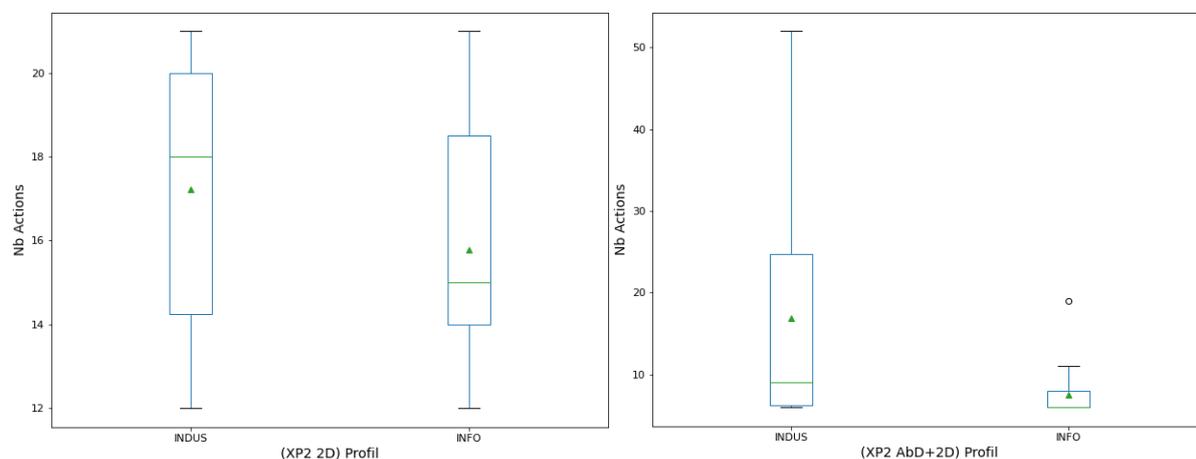


Figure 130 : Résultats de l'XP2. (À gauche) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition 2D. (À droite) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition AbD+2D.

Discussion :

Cette différence significative sur le nombre d'actions entre les profils informatiques et industriels montre que, dans le cas d'un scénario plus complexe, les INDUS font plus d'actions que les INFOS. Nous pouvons de nouveau supposer que cela est dû à l'aisance des profils informatiques car ils sont habitués à ce type de représentation graphique basée sur des nœuds de l'éditeur 2D. Concernant l'AbD + 2D, les deux profils ont donc réalisé un nombre d'actions similaire, ce qui rejoint la validation de l'hypothèse $H_{7,édition}$ « le nombre d'actions via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique » qu'il s'agisse d'un scénario simple ou complexe. Cependant, l'hypothèse $H_{6,édition}$ « le nombre d'actions via l'éditeur 2D est similaire entre un profil industriel et un profil informatique » ne peut donc être validée que partiellement.

5.5.4.3.4.3 Comparaison du nombre d'actions faites en fonction de l'outil d'édition

5.5.4.3.4.3.1 Pour un scénario simple (XP1)

Analyses statistiques :

Concernant la comparaison entre l'édition 2D et l'authoring-by-doing pour un scénario simple, les résultats des tests de Levene et de Shapiro (Tableau 47) montrent que l'homogénéité des variances est validée. Cependant, l'hypothèse de normalité n'est pas valide pour 3 des échantillons. Pour cette analyse, nous utiliserons donc un test de Mann et Whitney.

Tableau 47 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP1 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD	2D	AbD
Homogénéité des variances (Test de Levene)	F = 1.323 P _{value} = 0.256		F = 1.500 P _{value} = 0.233	
Hypothèse de normalité (Test de Shapiro)	W = 0.919 P _{value} = 0.063	W = 0.580 P _{value} = 5.56e-7*	W = 0.918 P _{value} = 0.270	W = 0.775 P _{value} = 0.0049*

Le tableau 48 suivant, nous montre qu'il y a bien une différence statistiquement significative que ce soient pour les profils informatiques et industriels.

Tableau 48 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions entre les différents outils d'édition. * représente les différences statistiquement significatives.

Mann et Whitney	
INFO - 2D vs AbD	INDUS - 2D vs AbD
U = 18.5 P _{value} = 2.32e-08*	U = 1.5 P _{value} = 2.35e-05*

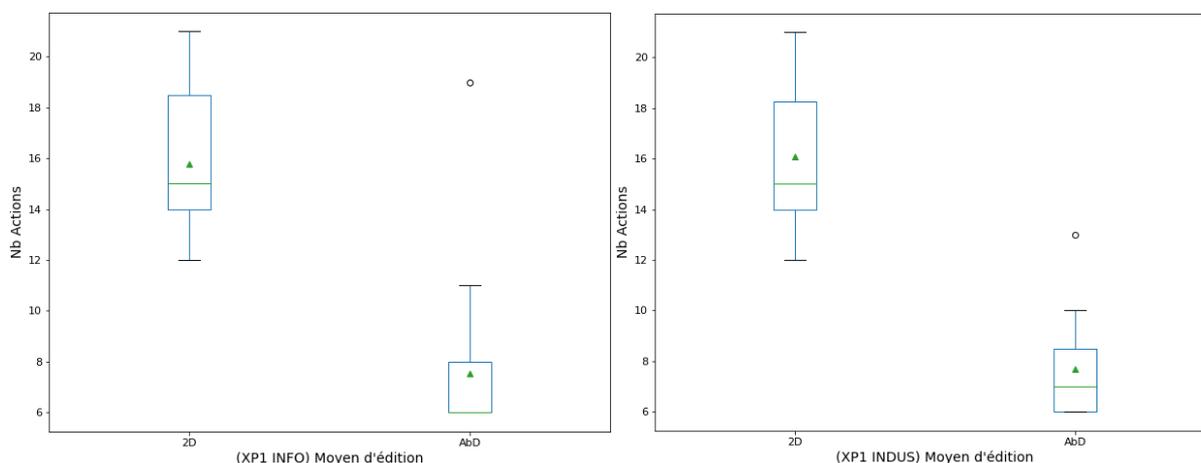


Figure 131 : Résultats de l'XP1. (À gauche) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INFO. (À droite) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INDUS.

Discussion :

Cette différence statistiquement significative nous permet de mettre en avant le fait que l'AbD est plus efficace concernant le nombre d'actions à réaliser pour un scénario simple, quel que soit le profil (INFO ou INDUS). Nous pouvons ainsi valider l'hypothèse H_{8.édition} « l'authoring-by-doing réduit le nombre d'actions faites pour un scénario simple par rapport à l'éditeur 2D ». L'AbD est donc très intéressant pour l'édition de long scénario, pouvant alors être plus fastidieux avec l'éditeur 2D. Cette analyse rejoint les retours d'expérience utilisateur présentés en 5.5.4.4.

5.5.4.3.4.3.2 Pour un scénario complexe (XP2)

Analyses statistiques :

Concernant la comparaison entre l'édition 2D et l'authoring-by-doing pour un scénario complexe avec la notion de parallélisme, les résultats des tests de Levene et de Shapiro (Tableau 49)

montrent que les conditions requises pour effectuer un t-test ne sont pas non plus valides. Pour cette analyse, nous utiliserons donc un test de Mann et Whitney.

Tableau 49 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP2 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.

Échantillons comparés	INFO		INDUS	
	2D	AbD+2D	2D	AbD+2D
Homogénéité des variances (Test de Levene)	F = 1.100 P _{value} = 0.306		F = 32.55 P _{value} = 0.0002*	
Hypothèse de normalité (Test de Shapiro)	W = 0.842 P _{value} = 0.034*	W = 0.940 P _{value} = 0.504	W = 0.667 P _{value} = 0.003*	W = 0.909 P _{value} = 0.432

Ainsi, les résultats du test de Mann et Whitney présentés dans le tableau 50 montrent qu'il y a une différence statistiquement significative du nombre d'actions entre 2D et AbD+2D pour les profils INDUS.

Tableau 50 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions de l'XP2 entre les différents outils d'édition. * représente les différences statistiquement significatives.

Mann et Whitney	
INFO - 2D vs AbD+2D	INDUS - 2D vs AbD+2D
U = 45 P _{value} = 0.103	U = 0.5 P _{value} = 0.003*

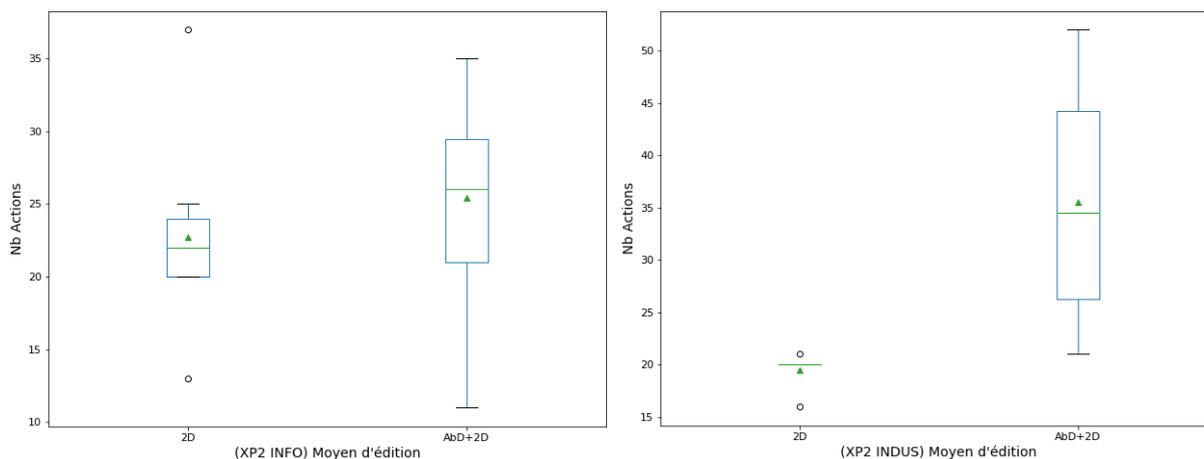


Figure 132 : Résultats de l'XP2. (À gauche) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INFO. (À droite) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INDUS.

Discussion :

Contrairement à l'analyse du scénario simple présentée précédemment, la différence statistique ne concerne, dans le cas d'un scénario complexe, que les profils industriels. Dans la figure 132, nous voyons que le nombre d'actions est très concentré entre 20 et 21 actions, tandis que pour l'AbD + 2D, les résultats sont plus hétérogènes. Cela montre que les profils industriels ont rencontré des difficultés à passer de l'AbD à l'éditeur 2D afin de retravailler le scénario créé en RV par démonstration. De plus, les résultats concernant les profils informatiques sont beaucoup plus homogènes. L'hypothèse $H_{9,édition}$ « l'authoring-by-doing réduit le nombre d'actions faites pour un scénario complexe par rapport à l'éditeur 2D » ne peut pas être validée. Il est alors d'autant plus nécessaire d'améliorer l'interface 2D pour améliorer la compréhension du scénario en cours d'édition.

5.5.4.3.5 Synthèse

Pour conclure sur l'analyse de l'influence du profil des participants ainsi que des outils d'édition utilisés pour la 1^{ère} (scénario simple – XP1) et la 2^{ème} expérience (scénario complexe - XP2), le tableau 51 et le tableau 52 synthétisent les résultats obtenus.

Tout d'abord, concernant les hypothèses H_{2.édition} « les temps d'éditions via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique » et H_{3.édition} « les temps d'éditions via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique », nous n'avons pas trouvé de différences statistiquement significatives entre les différents échantillons (Tableau 51). Ces hypothèses sont ainsi bien validées.

Le résultat très intéressant qui en ressort est que des profils orientés industriels sont capables, dans un temps équivalent à des profils informatiques, d'éditer un scénario d'assemblage et cela, en n'ayant pas de connaissance. Il en va de même pour la comparaison du nombre d'erreurs, d'actions et du niveau de qualité du scénario attendu qui ne sont pas statistiquement différents dans la majorité des cas. Nous notons tout de même qu'il y a une différence statistiquement significative entre les deux profils pour les résultats de l'édition 2D de l'XP1 sur le nombre d'erreur et le niveau de qualité, ainsi que sur le nombre d'actions réalisées avec l'édition 2D durant l'XP2.

Tableau 51 : Synthèse des différences entre les profils informatiques et industriels pour l'édition 2D et l'authoring-by-doing. * représente les différences statistiquement significatives.

	XP1						XP2					
	2D			AbD			2D			AbD+2D		
	INFO	INDUS	P-val	INFO	INDUS	P-val	INFO	INDUS	P-val	INFO	INDUS	P-val
Temps moyen	391s	409s	0.289	158s	145s	0.629	456s	375s	0.156	279s	352s	0.207
Nb d'erreurs moyenne	0.26	1.08	0.0006*	0.35	0.5	0.464	0.64	1.17	0.204	0.33	0.66	0.139
Nb d'actions moyenne	15.78	16.08	0.776	7.52	7.67	0.274	22.72	19.5	0.034*	25.42	35.5	0.094
Niveaux moyen	0.869	0.417	0.041*	0.739	0.75	0.481	1.09	0.5	0.054	1.25	0.833	0.098

Ensuite, concernant les résultats entre les outils d'éditions, nous voulions vérifier les hypothèses H_{4.édition} « l'authoring-by-doing réduit le temps mis pour éditer un scénario simple par rapport à l'éditeur 2D » et H_{5.édition} « l'authoring-by-doing réduit le temps mis pour éditer un scénario complexe par rapport à l'éditeur 2D ». Nous pouvons noter que dans la majorité des cas, il y a une différence statistiquement significative concernant le temps d'édition. En effet, sur l'XP1 et avec les profils informatiques de l'XP2, l'utilisation de l'authoring-by-doing a permis de réduire le temps d'édition de 40% à 65%, soit de moitié par rapport à l'édition 2D seule. Concernant le nombre d'erreur sur le scénario final, il n'y a statistiquement aucun impact suivant l'outil d'édition dans la majorité des cas. Seuls les résultats montrant que les profils industriels ont réalisé moitié moins d'erreur avec l'authoring-by-doing sur l'XP1 sont statistiquement significatifs. Ainsi, l'AbD permet de réduire les erreurs sur les scénarios simples du fait que les paramètres de l'assemblage à effectuer soient automatiquement complétés, évitant ainsi des erreurs. Ensuite, nous pouvons noter que, sur l'XP1, les participants ont dû effectuer moitié moins d'actions pour réaliser un

même scénario en utilisant l'authoring-by-doing. Puis, pour finir, notons que l'outil d'édition utilisé n'a pas d'impact significatif sur le niveau de qualité du scénario final.

Pour conclure, nous constatons que l'authoring-by-doing, comparé à l'édition 2D seule, a permis de réaliser un scénario en moins de temps, moins d'actions, avec moins d'erreurs et pour un niveau de qualité similaire.

Tableau 52 : Synthèse des différences entre l'édition 2D et l'authoring-by-doing pour les profils informatiques et industriels. * représente les différences statistiquement significatives.

	XP1						XP2					
	INFO			INDUS			INFO			INDUS		
	2D	AbD	P-val	2D	AbD	P-val	2D	AbD+2D	P-val	2D	AbD+2D	P-val
Temps moyen	391s	158s	3.67 ^{-08*}	409s	145s	8.71 ^{-07*}	456s	279s	0.0006*	375s	352s	0.723
Nb d'erreurs moyenne	0.26	0.348	0.245	1.08	0.5	0.028*	0.636	0.333	0.153	1.166	0.666	0.326
Nb d'actions moyenne	15.78	7.52	2.32 ^{-08*}	16.08	7.67	2.35 ^{-05*}	22.72	25.42	0.103	19.5	35.5	0.003*
Niveaux moyen	0.869	0.739	0.315	0.417	0.75	0.051	1.09	1.25	0.304	0.5	0.833	0.238

5.5.4.4 Résultats du questionnaire sur l'expérience utilisateur

Dans cette section, nous allons nous concentrer sur les résultats du questionnaire sur l'expérience utilisateur (UEQ). Cette analyse nous permettra ainsi de vérifier nos quatre hypothèses suivantes :

- H_{1.UEQ} : l'expérience utilisateur de l'éditeur 2D est similaire pour les profils industriels et informatiques,
- H_{2.UEQ} : l'expérience utilisateur de l'authoring-by-doing est similaire pour les profils industriels et informatiques,
- H_{3.UEQ} : l'éditeur utilisé a un impact sur l'expérience utilisateur des profils industriels,
- H_{4.UEQ} : l'éditeur utilisé a un impact sur l'expérience utilisateur des profils informatiques,

Pour effectuer cette analyse nous avons utilisé l'outil fourni par les auteurs du questionnaire¹¹. Les résultats sont présentés dans les figure 133 et figure 134. Pour rappel, le questionnaire donne les résultats sur deux axes : la qualité pragmatique et la qualité hédonique. Les résultats vont de -3 à 3, sachant que les valeurs entre -0.8 and 0.8 peuvent être considérées comme neutre.

¹¹ <https://www.ueq-online.org/>

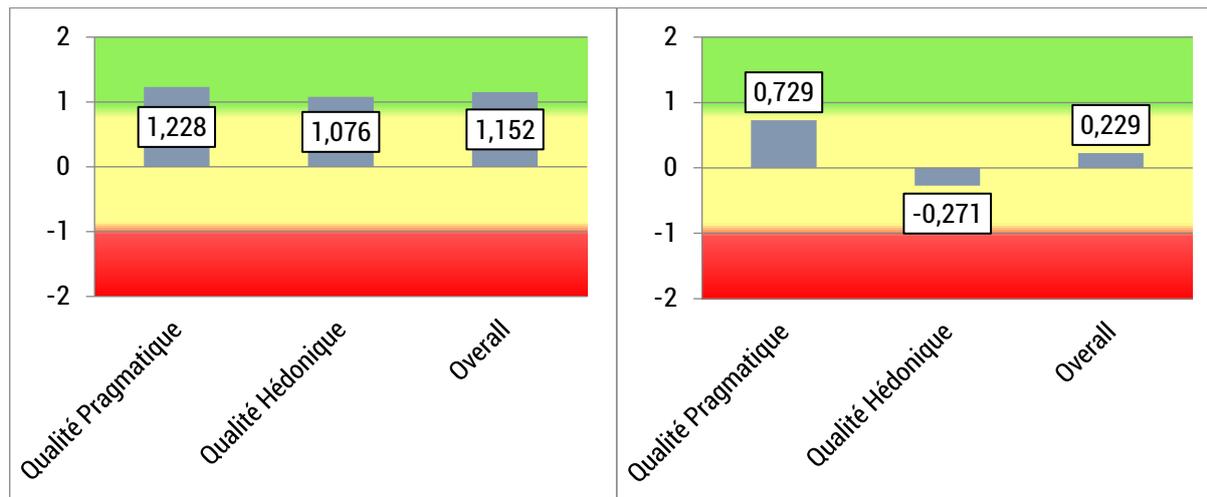


Figure 133 : Résultats de l'outil d'analyse UEQ pour l'édition 2D avec, à gauche, les profils informatiques et à droite, les profils industriels

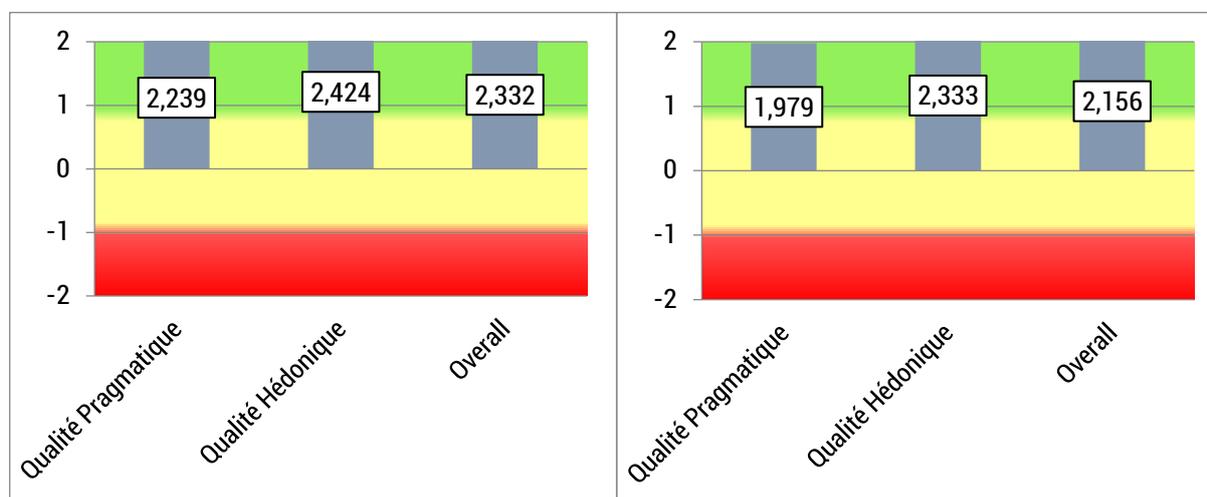


Figure 134 : Résultats de l'outil d'analyse UEQ pour l'authoring-by-doing avec, à gauche, les profils informatiques et à droite, les profils industriels.

Ainsi, nous pouvons observer une différence entre les profils informatiques et industriels sur l'édition 2D tandis que les qualités pragmatique et hédonique sont similaires sur l'authoring-by-doing pour les 2 profils. Nous pouvons également observer une différence notable entre les outils d'édition, que ce soit pour l'édition 2D ou l'authoring-by-doing.

Avant de vérifier si nos observations sur les différences et ressemblances sont statistiquement significatives, nous comparerons les résultats moyens (« overall »). Nous devons, dans un premier temps, tester la distribution normale de nos échantillons. Le test de Shapiro (Tableau 53) retourne une p-value > 0.05 pour l'ensemble des échantillons portant sur l'édition 2D. Les échantillons résultants de l'authoring-by-doing donnent une p-value < 0.05. Ces derniers ne suivant pas une loi normale, les conditions pour effectuer un T-test de Student ne sont donc pas valides. Nous utiliserons donc un test de Mann et Whitney pour les tests concernant ces échantillons.

Tableau 53 : Résultats du test de Shapiro sur les résultats UEQ « overall » pour l'XP1 par profil et outil d'édition. * indique que l'hypothèse de normalité n'est pas validée.

	2D		AbD	
	INFO	INDUS	INFO	INDUS
Hypothèse de normalité (Test de Shapiro)	W = 0.966 P _{value} = 0.607	W = 0.959 P _{value} = 0.765	W = 0.702 P _{value} = 1.48 ^e -05*	W = 0.824 P _{value} = 0.018*

Ainsi, nous comparerons d'abord les résultats moyens « overall » entre les profils industriels et informatiques dans la section 5.5.4.4.1 puis entre l'édition 2D et l'autoring-by-doing dans la section 5.5.4.4.2.

5.5.4.4.1 Comparaison entre les profils industriels et informatiques

Concernant la comparaison entre les profils industriels et informatiques, vérifions d'abord l'homogénéité des variances avec un test de Levene. En comparant INFO et INDUS, les résultats du test (Tableau 54), l'homogénéité des variances est validée.

Tableau 54 : Résultats du test de Levene pour les comparaisons UEQ entre les profils INFO et INDUS.

	2D INFO vs INDUS	AbD INFO vs INDUS
Homogénéité des variances (test de Levene)	F = 2.435 P _{value} = 0.128	F = 0.058 P _{value} = 0.811

Les conditions pour effectuer un T-Test de Student sont donc valides sauf pour « XP1_AbD_INFO » et « XP1_AbD_INDUS ». Nous utiliserons donc un test de Mann et Whitney pour la comparaison sur l'AbD (Tableau 55). Pour les résultats du T-test pour l'édition 2D, nous obtenons ainsi une différence statistiquement significative entre les profils industriels et informatiques avec une p-value de $0.019 < 0.05$. Tandis que, pour les résultats du test de Mann et Whitney, il n'y a pas de différence statistiquement significative pour l'autoring-by-doing entre les deux types de profils de participants.

Tableau 55 : Résultats du test de Mann et Whitney et du t-test pour les résultats UEQ entre les profils INFO et INDUS. * représente les différences statistiquement significatives.

T-test de Student	Mann et Whitney
2D INFO vs INDUS	AbD INFO vs INDUS
T = 2.458 P _{value} = 0.019*	U = 169 P _{value} = 0.285

Concernant l'édition 2D, nous pouvons donc constater que pour l'éditeur 2D, l'expérience utilisateur des profils INDUS est moins bonne que celle des profils INFO comme nous pouvons le voir sur les résultats de la figure 135 (à gauche) et de la figure 136 (à gauche). En effet, nous pensons que les profils informatiques étaient plus familiers avec notre type d'interface 2D tandis que les profils industriels ont pris beaucoup moins de plaisir à utiliser cet outil (Figure 136 – à gauche, Qualité Hédonique). Cela est également lié au manque de visualisation des objets et étapes dans l'interface comme les retours écrits le montrent dans la section 5.5.4.5.

A contrario, les deux profils ont donné un retour similaire pour l'AbD qui est considéré comme « excellent » par l'outil d'analyse UEQ. Enfin, notre hypothèse H_{1.UEQ} « l'expérience utilisateur de l'éditeur 2D est similaire pour les profils industriels et informatiques », ne peut pas être validée.

Par contre, notre hypothèse H_{2.UEQ} « l'expérience utilisateur de l'authoring-by-doing est similaire pour les profils industriels et informatiques » peut être validée.

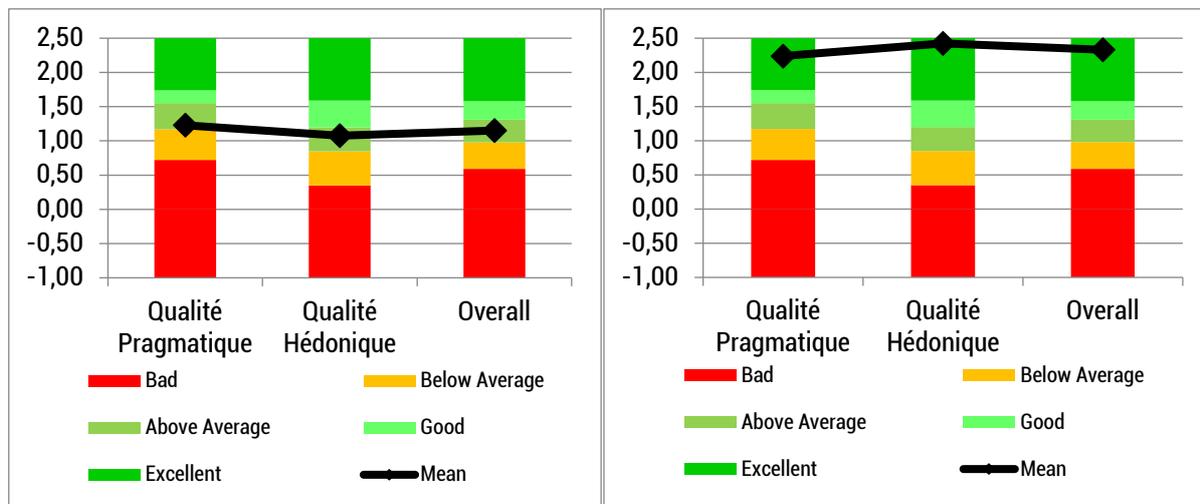


Figure 135 : Benchmark résultant des données sur les profils INFO avec l'outil d'analyse UEQ. À gauche, l'édition 2D et à droite, l'authoring-by-doing.

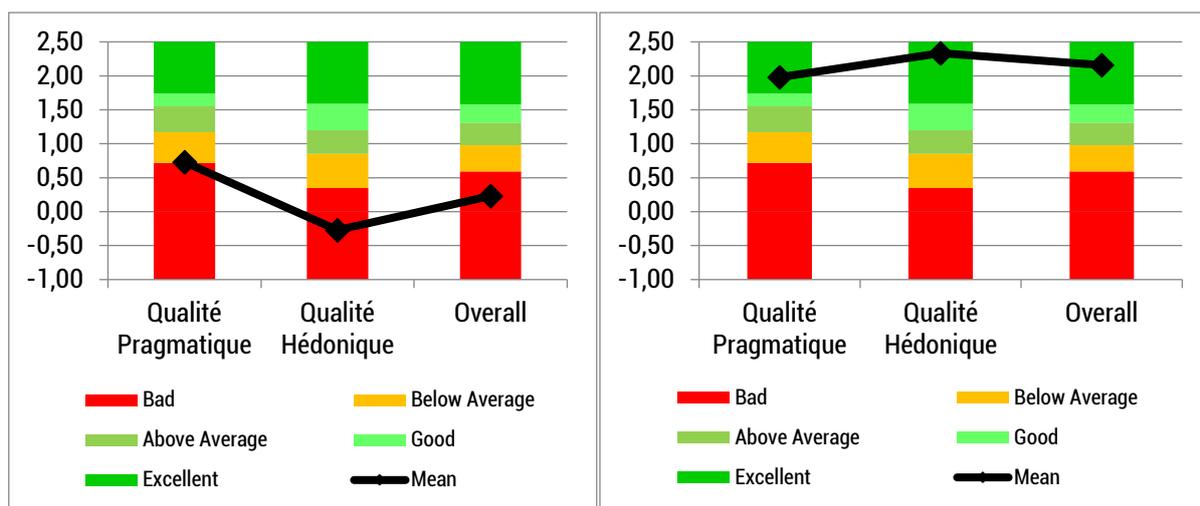


Figure 136 : Benchmark résultant des données sur les profils INDUS avec l'outil d'analyse UEQ. À gauche, l'édition 2D et à droite, l'authoring-by-doing.

Nous constatons donc que pour l'éditeur 2D, l'expérience utilisateur des profils INDUS est moins bonne que celle des profils INFO. Cela montre que l'éditeur 2D reste encore complexe à appréhender pour les profils INDUS qui ne sont pas habitués aux représentations graphiques par rapport aux profils INFOS. Cela renforce les résultats obtenus précédemment dans les sections 5.5.4.3.3.3 et 5.5.4.3.4.3. En effet, les profils INDUS mettent plus de temps et font plus d'erreur que les profils INFO pour obtenir un scénario de qualité et leur ressenti en termes d'expérience utilisateur s'en trouve impactée. La section suivante va permettre d'approfondir l'analyse.

5.5.4.4.2 Comparaison entre les outils d'édition

Concernant la comparaison entre l'édition 2D et l'authoring-by-doing, vérifions tout d'abord l'homogénéité des variances avec un test de Levene. En comparant 2D et AbD, les résultats du test (Tableau 56) montrent que l'homogénéité des variances n'est pas validée.

Tableau 56 : Résultats du test de Levene pour les comparaisons UEQ entre les différents outils d'édition.

	INFO 2D vs AbD	INDUS 2D vs AbD
Homogénéité des variances (test de Levene)	F = 5.662 P _{value} = 0.022*	F = 6.16 P _{value} = 0.021*

Nous utiliserons donc un test de Mann et Whitney pour ce dernier (Tableau 57). Nous pouvons alors confirmer qu'il y a bien une différence statistiquement significative entre l'édition 2D et l'authoring-by-doing. L'authoring-by-doing a parmi une expérience utilisateur plus impactante que l'édition 2D comme nous pouvons le voir entre la figure 137 et figure 138. En effet, celui-ci est beaucoup plus intuitif et ludique. Nous verrons quelques exemples de retour allant en ce sens dans la section suivante. Nos hypothèses H_{3,UEQ} « l'éditeur utilisé a un impact sur l'expérience utilisateur des profils industriels » et « H_{4,UEQ} : l'éditeur utilisé a un impact sur l'expérience utilisateur des profils informatiques » sont donc validées.

Tableau 57 : Résultats du test de Mann et Whitney pour les résultats UEQ entre les outils 2D et AbD. * représente les différences statistiquement significatives.

Mann et Whitney	
INFO 2D vs AbD	INDUS 2D vs AbD
U = 70	U = 15
P _{value} = 1.90e-05*	P _{value} = 0.001*

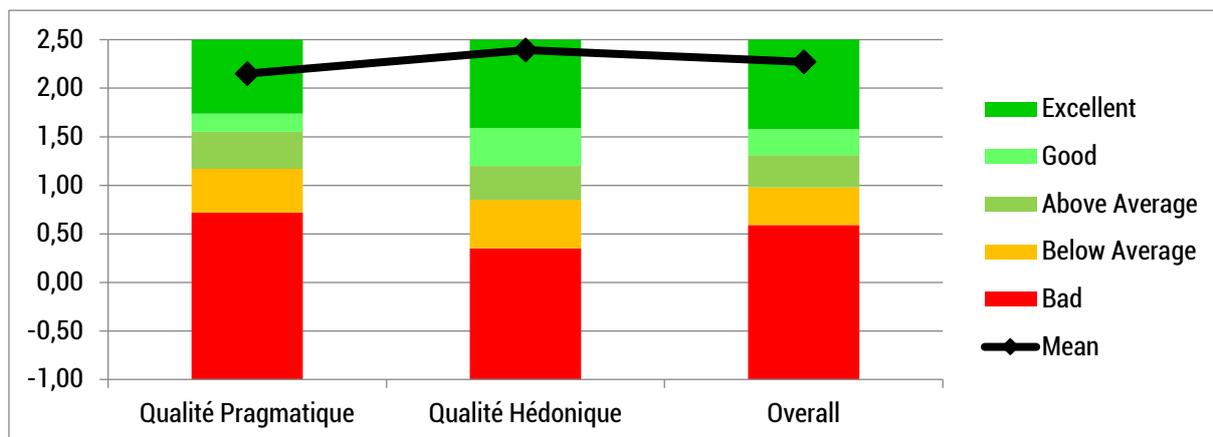


Figure 137 : Benchmark résultant des données sur l'authoring-by-doing avec l'outil d'analyse UEQ

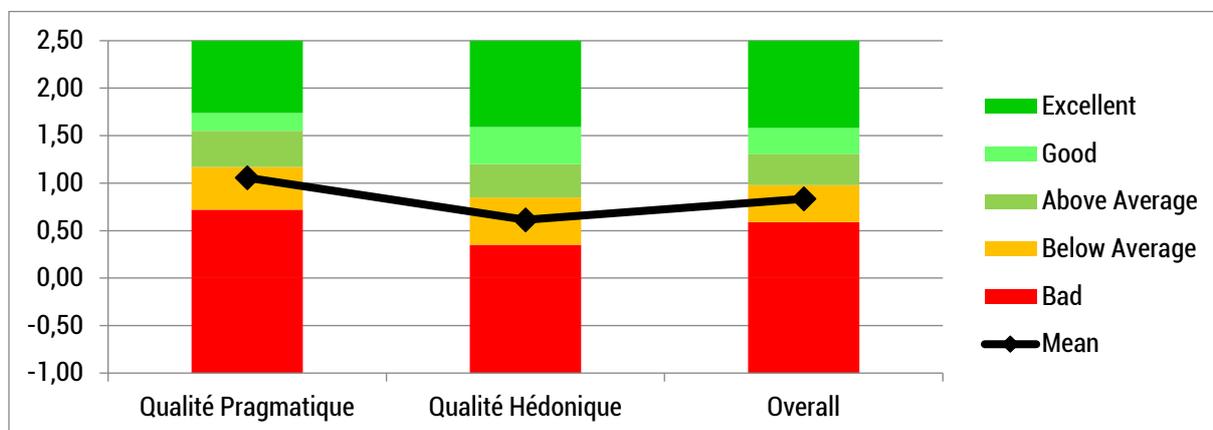


Figure 138 : Benchmark résultant des données sur l'édition 2D avec l'outil d'analyse UEQ

5.5.4.5 Analyse des retours d'expérience

Au terme de cette expérience, chaque participant a répondu à la question concernant le moyen d'édition qu'il a préféré lors du questionnaire final : « Suite à cette expérience, préférez-vous l'édition 2D, l'authoring-by-doing ou la combinaison des deux ? », suivie d'une question ouverte « Pourquoi ce choix ? ». Les résultats représentés dans la figure 139 montrent que la majorité des participants, soit **63%**, a préféré l'authoring-by-doing à l'édition 2D et que **37%** des participants ont préféré la combinaison de l'authoring-by-doing et de l'édition 2D. en outre, aucun participant n'a choisi l'édition 2D seule comme préférence. L'hypothèse $H_{1,REX}$ « Les participants sont favorables à une édition en authoring-by-doing. » est donc validée.

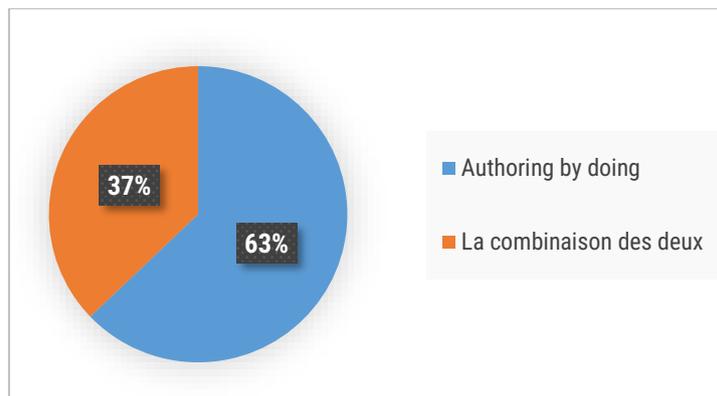


Figure 139 : Répartition des préférences des participants concernant l'outil d'édition préféré.

Lorsque l'on regarde les raisons de leurs choix, les participants ayant choisi l'authoring-by-doing ont expliqué avoir préféré l'outil car il est plus « intuitif », « instinctif », « ludique » ou encore « simple d'utilisation ». Cependant, afin de nuancer ces propos, plusieurs participants ont pointé le fait que l'outil d'édition 2D « serait bien meilleur si nous pouvions voir des visuels des étapes » qui ont été enregistrées lors de l'authoring-by-doing. En effet, le fait de ne pas avoir de retour visuel et en temps réel du scénario d'assemblage au fur et à mesure de sa création a joué sur la compréhension du participant sur le scénario qu'il était en train d'éditer. De même, un des participants explique que l'édition 2D « est plus facile de prise en main [...] mais ne permet pas cette compréhension du scénario tels qu'en 3D ». Il s'agirait alors d'améliorer l'édition 2D en intégrant une visualisation des actions définies durant l'édition d'un scénario.

Concernant les participants ayant choisi la combinaison des deux, ils mettent bien en évidence dans leurs justifications la complémentarité qu'offre l'utilisation des deux outils d'éditeurs. Un des participants indique par exemple « Je préfère la combinaison des deux pour pouvoir créer le scénario avec l'authoring-by-doing puis retravailler en 2D si je veux modifier des détails », ou encore « La combinaison des 2 permet de faire en premier lieu l'expérience en RV, puis modifier par la suite, étape par étape, sur l'édition 2D ». Il est alors plus compréhensible que tous les participants aient préféré soit l'authoring-by-doing, soit la combinaison des deux.

5.6 Conclusion

Pour conclure ce chapitre, nous avons ainsi développé l'aspect édition de scénarios et de procédures métiers du framework INTERVALES. Nous avons tout d'abord décrit les objectifs et besoins auxquels doit répondre un éditeur de scénario. Nous souhaitons alors proposer des outils d'édition permettant à un expert métier n'ayant pas de connaissance en développement informatique d'éditer un scénario.

Après une étude préliminaire dans laquelle nous avons développé un premier outil d'édition de scénario via une interface 2D, nous avons montré au travers d'une expérimentation qu'il y avait un engouement pour la mise en place d'un outil d'édition 2D et qu'il était possible à un expert métier de rédiger un scénario aussi rapidement et avec un même temps de pratique, qu'un développeur informatique. Nous avons cependant noté des limitations et avons alors proposé d'implémenter deux outils d'éditeurs, un éditeur 2D et un éditeur par démonstration « Authoring-by-doing », en se basant sur le framework INTERVALES.

Suite-à cela, nous avons réalisé une expérimentation dans la section 5.5 afin de comparer l'édition de scénarios avec ces outils et sur deux profils de participants (INFO et INDUS). L'objectif de cette expérience était d'étudier et de collecter des résultats concernant l'usage des outils d'édition mis au point, de les comparer mais également de voir si un expert métier est capable de créer et/ou d'éditer un scénario d'assemblage en utilisant ces outils d'édition. Le tableau 58 synthétise l'ensemble de nos hypothèses et si elles sont validées ou non.

Tableau 58 : Synthèse des hypothèses et résultats obtenus.

Hypothèse	Description	Statut
H _{1,édition}	L'éditeur 2D et l'authoring-by-doing permettent d'éditer un scénario	Validée
H _{2,édition}	Les temps d'éditeurs via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique.	Validée
H _{3,édition}	Les temps d'éditeurs via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique.	Validée
H _{4,édition}	L'authoring-by-doing réduit le temps mis pour éditer un scénario simple par rapport à l'éditeur 2D.	Validée
H _{5,édition}	L'authoring-by-doing réduit le temps mis pour éditer un scénario complexe par rapport à l'éditeur 2D.	Partiellement validée (uniquement validée pour les profils INFO)
H _{6,édition}	Le nombre d'actions via l'éditeur 2D sont similaires entre un profil industriel et un profil informatique.	Partiellement validée (uniquement validée pour scénario simple)
H _{7,édition}	Le nombre d'actions via l'authoring-by-doing sont similaires entre un profil industriel et un profil informatique.	Validée
H _{8,édition}	L'authoring-by-doing réduit le nombre d'actions faites pour un scénario simple par rapport à l'éditeur 2D.	Validée
H _{9,édition}	L'authoring-by-doing réduit le nombre d'actions faites pour un scénario complexe par rapport à l'éditeur 2D.	Non validée
H _{10,édition}	Le nombre d'erreurs via l'éditeur 2D est similaire entre un profil industriel et un profil informatique	Partiellement validée (uniquement validée pour scénario complexe)
H _{11,édition}	Le nombre d'erreurs via l'authoring-by-doing est similaire entre un profil industriel et un profil informatique.	Validée
H _{12,édition}	L'authoring-by-doing réduit le nombre d'erreurs commises pour un scénario simple par rapport à l'éditeur 2D.	Partiellement validée (uniquement validée pour les profils INDUS)

Hypothèse	Description	Statut
H _{13.édition}	L'authoring-by-doing réduit le nombre d'erreurs commises pour un scénario complexe par rapport à l'éditeur 2D.	Non validée
H _{1.UEQ}	L'expérience utilisateur de l'éditeur 2D est similaire pour les profils industriels et informatiques.	Non validée
H _{2.UEQ}	L'expérience utilisateur de l'authoring-by-doing est similaire pour les profils industriels et informatiques.	Validée
H _{3.UEQ}	L'éditeur utilisé a un impact sur l'expérience utilisateur des profils industriels.	Validée
H _{4.UEQ}	L'éditeur utilisé a un impact sur l'expérience utilisateur des profils informatiques.	Validée
H _{1.REX}	Les participants sont favorables à une édition en authoring-by-doing.	Validée

Dans un premier temps, ce que nous pouvons conclure de cette expérience est, qu'en effet, ces outils (éditeur 2D, AbD ou Abd+2D) permettent d'éditer un scénario d'assemblage en réalité virtuelle, qu'il soit simple ou complexe. Grâce à ces éditeurs, un expert métier est capable d'éditer un scénario sans connaissance en développement informatique dans un temps similaire à celui d'un profil informatique. De plus, l'approche authoring-by-doing est prometteuse car plus efficace en termes de temps mais aussi en termes d'expérience utilisateur. De plus les participants, profils INFO et INDUS confondus, ont trouvé l'AbD beaucoup plus intuitif mais reconnaissaient l'avantage d'un moyen d'édition par démonstration combinée à une interface 2D.

6 Conclusion Générale

6.1 Travaux réalisés

Dans le premier chapitre, nous avons posé le contexte de cette thèse. Nous avons pu introduire l'industrie 4.0, définir ce que sont la réalité virtuelle et la réalité augmentée et présenter les exemples de cas d'usages de ces technologies en industrie et présents dans la littérature scientifique. Puis, profitant du cadre industriel de cette thèse, se déroulant en collaboration avec l'entreprise OREKA Ingénierie, une PME répondant depuis plusieurs années à la demande d'applications innovantes en RA/RV pour l'industrie, nous avons présenté une analyse des réalisations et des usages de la RV/RA qui sont faits au sein des entreprises clientes. Ainsi, une analyse des affaires sur 5 ans (2015 à juin 2019) nous a notamment permis de faire ressortir les facteurs communs entre les différents projets et de capitaliser sur le processus de création. En posant ainsi les contextes scientifique et industriel des usages présentés, nous avons identifié 3 grandes catégories de cas d'usages tels que : les outils de conception, de planification ou d'assistance (conception, mise en œuvre, maintenance) ; les outils de formation ; les outils de démonstration. Par ailleurs, nous avons pu démontrer et confirmer que l'arrivée des nouvelles technologies RV/RA dans l'industrie a apporté son lot d'opportunité et de besoins propres parmi ces différents types de cas d'usages. En effet, nous avons montré que la réalité virtuelle comme la réalité augmentée pouvaient toutes deux avoir des besoins spécifiques. Par exemple, la possibilité de travailler en collaboration avec plusieurs agents autonomes ou plusieurs opérateurs (Galambos et al., 2015; Havard, Trigunayat, et al., 2019), dans le cadre de la conception ergonomique d'un poste de travail, l'étude d'implantation d'un atelier de production ou encore l'assistance au montage avec un agent autonome composé d'un bras cobotique. De ce chapitre, nous avons pu faire naître et positionner la problématique de cette thèse.

Dans le second chapitre, nous avons réalisé un état de l'art des travaux de recherche portant sur les modèles d'environnements, de scénarios et des frameworks permettant la création / édition de contenus en RV/RA. Les modèles d'environnements ont été analysés suivant trois catégories : orientés « comportement », orientés « objets synoptiques » et orientés « objets-relations ». Les modèles de scénario peuvent, eux, permettre trois niveaux de liberté en termes de scénarisation : procédural (pas à pas), scénario ouvert avec objectifs (des lignes directrices), et scénario complètement ouvert (simulation). Cet état de l'art a tout d'abord montré qu'il y avait un manque de modèles permettant une création de contenus unifiés pour des usages en RV et RA. La raison principale étant leurs usages spécifiques, principalement orientés conception et formation pour la RV et assistance pour la RA. Il faut cependant noter qu'avec les évolutions des technologies et l'essor des jumeaux numériques, l'usage de la RA pour des activités de revue de conception ou de formation fait l'objet d'un intérêt croissant des communautés scientifiques et industrielles. Cette étude des différents cas d'usages montre qu'ils peuvent utiliser un même moteur de scénario ainsi que des données et connaissances métiers communes. La différence étant qu'en

réalité virtuelle, un scénario sera amené à être plus évolué et ouvert dans le cas d'une simulation, tandis qu'en réalité augmentée, un scénario sera plus souvent amené à être procédural. De plus, nous avons constaté que la majorité des modèles disposant d'un moteur de scénario permettent au minimum la définition d'objectifs de manière procédurale. Ils sont cependant moins nombreux à laisser la possibilité à l'apprenant de réaliser un scénario complexe en autonomie en lui laissant la possibilité de commettre des erreurs. Nous avons ensuite étudié plusieurs langages graphiques pouvant représenter le séquençement de scénario. Nous avons alors pu relever des limitations concernant la représentation graphique. D'une part, nous constatons que même si une représentation légère et claire est un atout a priori, la limitation dans les expressions disponibles peut devenir restrictive lorsqu'il faut définir un scénario complexe. D'autre part, la nécessité d'étudier plusieurs méta-modèles et langages différents pour pouvoir prendre en main l'outil de scénarisation peut freiner la facilité d'adoption du framework. C'est pourquoi, nous nous orientons vers un langage unifié et disposant de plusieurs éléments de construction à la manière de Grafchart, permettant ainsi d'optimiser la représentation d'un scénario complexe. Ainsi, dans ce chapitre, nous avons pu affiner notre problématique et lister les caractéristiques à prendre en compte dans notre conception de modèle. Le framework proposé doit faire abstraction de la technologie utilisée, permette une création commune de contenus que ce soit RV/RA ou PC (Clavier/Souris) et permettre l'intégration d'agents autonomes. Pour cela, nous nous sommes orientés vers une approche « Objet-Relation » pour le modèle d'environnement et un scénario ouvert avec objectif pour le modèle de scénario.

Dans le chapitre 3, nous avons détaillé notre proposition de framework afin de répondre à la fois, aux besoins des usages de l'industrie 4.0 présentés dans le chapitre 1 mais également aux critères identifiés dans l'état de l'art du chapitre 2. Ainsi, nous avons présenté INTERVALES : INTERActive Virtual and Augmented framework for industrial Environments and Scenarios qui intègre les exigences suivantes :

- Décrire les caractéristiques de chaque entité de la scène ;
- Définir les actions réalisables et les paramètres associés à chacune lors de l'interaction avec les entités ;
- Gérer et vérifier les procédures et scénarios pendant l'activité ;
- Enfin, exporter le scénario et l'état de la scène afin d'étendre le framework INTERVALES avec des outils externes.

Le framework proposé est ainsi composé de cinq modules différents : le module « Entity-Feature », pour définir les caractéristiques de chaque élément ; le module d'interaction afin de gérer toutes les interactions faites par l'utilisateur ; le module « Scene state », pour gérer les actions et relations effectuées ; le module « Scenario Authoring », pour définir et éditer des scénarios ; et le module d'import/export, pour enregistrer et charger l'état de la scène à un instant t . De ce fait, ces différents modules rendent possible la création et l'édition de l'environnement et des interactions au sein de celui-ci ainsi que la définition des tâches et des objectifs. Ensuite, nous avons proposé un formalisme sous forme de graphe de nœuds s'inspirant de représentation industrielle pour la représentation de scénario. Notre formalisme utilise le concept de ScenarioObject pouvant être sensibilisé et validé et pouvant être dérivé en différents blocs de

construction. Pour finir, nous avons présenté les développements qui ont été réalisés sur le framework INTERVALES pour son intégration au moteur Unity 3D. Nous avons notamment détaillé les automatismes développés qui ont été mis à disposition afin de simplifier le processus de création d'application en réalité virtuelle et augmentée.

Dans le chapitre 4, nous avons présenté comment INTERVALES pouvait s'intégrer à un processus de conception entre les différents acteurs identifiés : experts métiers, développeurs et utilisateurs finaux. Pour cela, nous avons présenté le workflow qui pouvait être utilisé. Ensuite, plusieurs mises en applications industrielles d'INTERVALES ont été détaillées, qu'elles s'appliquent à la recherche, à des projets industriels ou encore à des besoins dans le domaine de la formation. Cela nous a permis de démontrer qu'il était possible de : mettre en place des environnements virtuels et augmentés, définir les interactions faisables dans ces environnements et produire des scénarios dans différents contextes industriels tout en s'intégrant au processus de développement d'une entreprise. Par conséquent, ce framework INTERVALES, illustré dans différents cas d'usages, montre qu'il permet d'ores et déjà une plus-value en termes de fonctionnalités et d'automatisation de développement. Nous avons également montré qu'il était possible de définir un scénario dans le standard INTERVALES pour un environnement virtuel puis d'utiliser ce scénario pour le transposer en gamme d'assemblage en réalité augmentée.

Cependant, la définition de scénario reste très statique car passant par la formalisation des données en JSON. C'est pourquoi, nous nous sommes concentrés sur une évolution du framework INTERVALES pour proposer des interfaces d'édition de scénarios et de procédures métiers dans notre dernier chapitre. Tout d'abord, un premier prototype d'éditeur de scénario intégré à l'éditeur Unity 3D a été proposé et une étude préliminaire réalisée. Nous avons pu montrer cet engouement pour la mise en place d'un outil d'édition permettant à un expert métier de rédiger un scénario aussi rapidement et avec un même temps de pratique qu'un développeur informatique. Toutefois, cette première version se limitait à un usage direct au sein d'Unity et n'était pas utilisable autrement. Nous avons alors proposé d'implémenter deux outils d'éditeurs basés sur le framework INTERVALES: un éditeur 2D et un éditeur par démonstration « Authoring-by-doing ». Nous avons ensuite réalisé une expérimentation dans le but de comparer ces deux moyens d'éditeurs suivant différents critères de performance, de satisfaction et de comportement. Cette étude a impliqué des participant·e·s de deux profils différents : des profils informatiques et des profils industriels. Les principales contributions de cette expérimentation ont été de montrer que :

- Les profils industriels peuvent créer un scénario simple dans un temps similaire au profil industriel avec chacun des éditeurs ;
- L'éditeur Authoring-by-Doing permet de créer des scénarios simples plus rapidement qu'avec l'éditeur 2D pour chacun des deux profils, notamment parce qu'il réduit la complexité de paramétrage des actions en les complétant automatiquement ;
- L'association de l'éditeur authoring-by-doing avec l'éditeur 2D permet de créer des scénarios complexes pour chacun des deux profils, même si les profils informatiques sont plus rapides que les profils industriels ;

- L'approche authoring-by-doing est plus accessible pour l'expert métier et beaucoup plus « ludique » et « intuitive » que l'éditeur 2D, comme le montrent les retours d'expérience des utilisateurs.

6.2 Perspectives

Suite à ces travaux de recherche et les différentes observations abordées dans ce manuscrit, nous ouvrons plusieurs perspectives pour la suite des travaux :

- **Amélioration de l'édition de scénario avec l'éditeur 2D**

Suite aux résultats de l'expérimentation comparant la création de scénario avec INTERVALES à l'aide de l'éditeur 2D et via l'authoring-by-doing en réalité virtuelle, les travaux sur l'éditeur 2D doivent être poursuivis afin de l'améliorer. En effet, que ce soit suite à une utilisation de l'éditeur 2D uniquement ou combiné avec l'authoring-by-doing en réalité virtuelle, un des retours les plus présents, pour les profils informatiques et experts métiers confondus, portait sur le manque de retour visuel sur le scénario en cours d'édition. Il faudra ajouter notamment, une prévisualisation des entity et des features concernées durant le paramétrage d'une action afin d'assurer à l'expert métier qu'il a bien sélectionné les paramètres souhaités. Il serait également intéressant d'obtenir pour chaque action, une visualisation de l'animation effectuant cette action (Ex : animation d'un assemblage entre deux pièces). De manière plus générale, il serait intéressant de formaliser graphiquement la représentation de l'ObjectiveNode associé à la Task utilisée.

Une autre amélioration possible, remontée par les participants de l'expérimentation, concerne l'automatisation du paramétrage des actions suivant le contexte. En effet, il serait intéressant de filtrer le choix des entity et feature possibles suivant l'action choisie. De plus, un préparamétrage permettrait un gain de temps d'édition. Par exemple, dans le cas d'une RelationSnapOriented, l'éditeur pourrait automatiquement proposer les FeatureSnapOriented des deux entity concernées.

- **Piste d'amélioration du framework INTERVALES**

Actuellement, le framework INTERVALES comprend un modèle d'environnement ainsi qu'un modèle de scénario permettant de les formaliser. Dans la continuité de nos travaux, il faudrait approfondir la notion d'agent en définissant des rôles. Ainsi le même agent pourrait se comporter différemment en fonction du rôle qui lui serait attribué dans le scénario. Par ailleurs le rôle permettrait de limiter les interactions faisables par un agent. En outre, il serait intéressant d'intégrer des notions de suivi de l'apprenant afin d'avoir un suivi inter-sessions de celui-ci et de pouvoir adapter les scénarios en fonction des performances de l'utilisateur sur les précédentes sessions. Enfin différentes perspectives d'évolutions du framework INTERVALES pour renforcer son intégration comme composant des jumeaux numériques et physiques sont à l'étude notamment dans le cadre du projet JENII.

- **Projet JENII (Jumeaux d'Enseignement Numériques, Immersifs et Interactifs pour l'Education et la Formation)**

Dans la suite des travaux concernant le framework INTERVALES, ce dernier va également être intégré au processus de développement et à l'architecture logiciel d'un projet de jumeaux

numériques venant de démarrer pour lequel CESI a confié certaines réalisations à OREKA. Dans un premier temps, le jumeau numérique concerne la mise en place de visites virtuelles de deux plateformes :

- La plateforme Usine Flexible de Production (UFP : Figure 140) installée sur le campus CESI de Rouen qui reproduit un atelier de production modulaire intégrant une ligne de production automatisée, des postes manuels, des AGVs (robots mobiles), robots et cobots, des IIoT, ou encore des interfaces Humain-machine basées sur de la réalité augmentée ou de la réalité virtuelle.
- La plateforme Unité Autonome de Fabrication Additive Métallique (UAFAM : Figure 141) installée sur le campus CESI de Nanterre et dédiée à l'impression 3D des pièces métalliques en procédé de fusion de poudre par laser et basée sur un concept innovant d'usine modulaire et transportable.



Figure 140 : Environnement 3D du jumeau numérique de la plateforme UFP (Rouen)



Figure 141 : Environnement 3D du jumeau numérique de la plateforme UAFAM (Nanterre)

Comme présenté sur la figure 142, les perspectives portent sur l'intégration du framework INTERVALES associé à une architecture de communication basée sur le standard OPC UA pour permettre le développement et l'utilisation de jumeaux numériques et physiques industriels couplés à des interfaces en RA ou RV. Pour cela, nous allons étudier le développement et l'implémentation de connecteurs permettant la communication vers des outils de simulation de flux ou robotique afin d'obtenir des comportements réalistes des agents au sein de l'environnement virtuel, sans avoir à les redévelopper au sein du moteur de rendu 3D. Les besoins portent notamment sur de la co-simulation à travers le jumeau numérique, de la collaboration synchrone entre opérateur en RA et expert en RV ou encore des scénarios combinant système réel et virtuel pour de la conception d'usine.

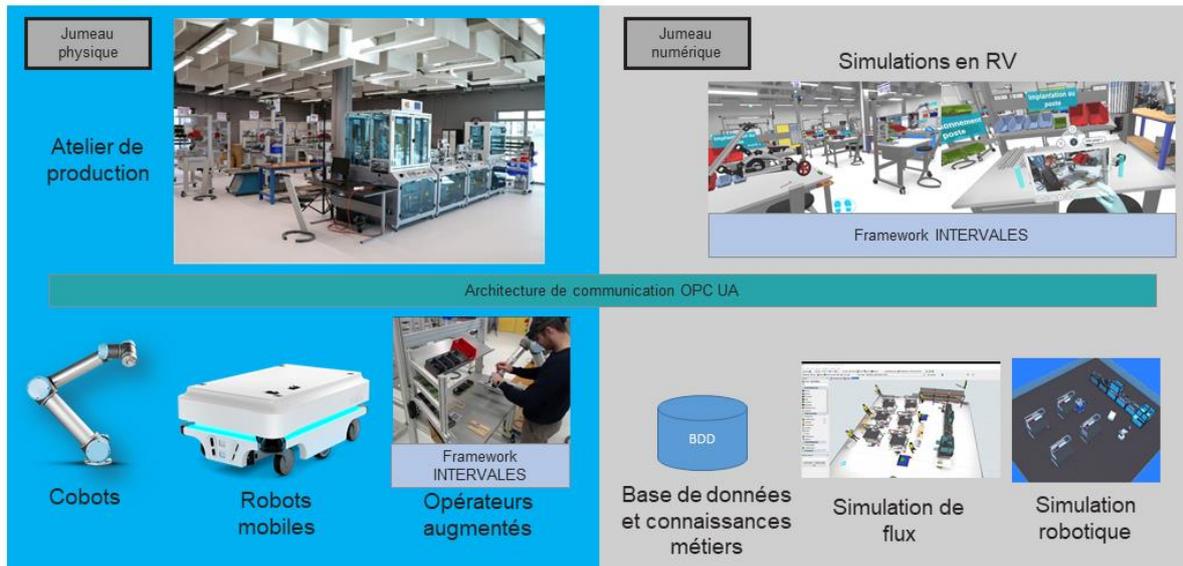


Figure 142 : Architecture et composants des jumeaux numériques et physiques intégrant INTERVALES

7 Publications

ACL (Articles dans des revues internationales avec comité de lecture)

Richard, K., Havard, V., His, J., & Baudry, D. (2021). *INTERVALES: INTERactive Virtual and Augmented framework for industrial Environment and Scenarios*. In *Advanced Engineering Informatics*, 50, 101425. <https://doi.org/10.1016/J.AEI.2021.101425>

C-ACTI (Communications avec actes dans un congrès international)

Badets, A., Havard, V., Richard, K., & Baudry, D. (2020). *Using collaborative VR technology for Lean Manufacturing Training: a case study*, in Presented at the VRIC ConVRgence 2020: 22nd Virtual Reality International Conference, Laval Virtual, Laval, France.

Richard, K., Havard, V., & Baudry, D. (2021). *Authoring-By-Doing: An Event-Based Interaction Module for Virtual Reality Scenario Authoring Framework* (pp. 519–527). https://doi.org/10.1007/978-3-030-87595-4_38

ASCL (Articles dans des revues sans comité de lecture)

Richard, K., Havard, V., Trigunayat, A., & Baudry, D. (2018). *Modèle de données pour la création d'environnement interactif en réalité virtuelle et augmentée*. AFRV 2018: Les Journées de La Réalité Virtuelle.

8 Table des figures

Figure 1 : Distribution par année des publications et journaux répertoriés en lien avec l'Industrie 4.0 (Danjou et al., 2017b, 2017a).....	13
Figure 2 : Les nouvelles technologies de l'industrie 4.0 selon (Danjou et al., 2017b) (Voir Annexe : Figure 143).....	14
Figure 3 : Réalité virtuelle via casque immersif (à gauche) (Réalité Virtuelle : Industriels, Architectes, Formateurs... Pourquoi Tous Les Pros Vont s'y Mettre, 2016) et via un CAVE (à droite) (Applications Du CAVE, 2017).	15
Figure 4 : Outil collaboratif à distance pour la planification d'agencement d'une usine (Havard, Trigunayat, et al., 2019)	16
Figure 5 : (à gauche) Une session VirCA multi-utilisateurs avec quatre participants. Chaque utilisateur est virtuellement représenté par une tête colorée qui indique la direction du regard et son mouvement. (à droite) Autre cas d'usage, un des utilisateurs utilise un CAVE et la tête rouge symbolise l'opérateur assis à côté du vrai robot (Galambos et al., 2015)	17
Figure 6 : Contrôle de sécurité en réalité virtuelle grâce à la détection de collision (Havard, Jeanne, et al., 2019).....	17
Figure 7 : Formation a une maintenance industrielle sur une chaîne de production en réalité virtuelle (Havard et al., 2017)	18
Figure 8 : Dans cette scène, l'alarme est activée, avertissant l'utilisateur qu'il est entré dans l'espace de travail du robot (Matsas & Vosniakos, 2017)	18
Figure 9 : Continuum entre le monde réel et le monde virtuel (Milgram & Kishimo, 1994)	19
Figure 10 : Mise à l'échelle multidimensionnelle des exemples d'application en réalité augmentée présentés dans (Röltgen & Dumitrescu, 2020)	20
Figure 11 : Une séquence de localisation. (a) Une flèche fixe à l'écran indique la distance de rotation la plus courte jusqu'à la cible. (b) Lorsque l'utilisateur s'oriente sur la cible, une flèche 3D semi-transparente pointe vers la cible. (c) Lorsque l'utilisateur atteint la cible, la flèche 3D commence un fondu progressif jusqu'à une transparence totale. (d) Lorsque la flèche s'est complètement estompée, un bref effet de surbrillance marque l'emplacement précis de la cible. (Henderson & Feiner, 2011)	20
Figure 12 : Gamme de maintenance industrielle en réalité augmentée sur une chaîne de production (Havard et al., 2016, 2017; Richard et al., 2016).....	21
Figure 13 : (à gauche) Le type d'installation utilisé par Funk avec un projecteur et une caméra de profondeur en A. (à droite) La rétroaction projetée peut être enrichie d'informations supplémentaires. (Funk, 2016).....	22

Figure 14 : Répartition des affaires en fonction des types de dispositifs utilisés (à gauche), répartition en nombre, (à droite), répartition en chiffre d'affaires.....	26
Figure 15 : Répartition en % des montants HT entre les affaires (2015-2019).....	26
Figure 16 : Répartition des types d'usage par année (à gauche) et sur la période complète 2015-2019 (à droite).	27
Figure 17 : Répartition niveaux d'interaction (2015-2019).....	27
Figure 18 : Répartition contextes en nombre d'affaire (2015-2019).....	28
Figure 19 : Répartition des critères (2015-2019)	29
Figure 20 : Pourcentage (suivant le montant HT) des 3 principaux critères pour toutes les affaires	29
Figure 21 : Organisation utilisée pour la comparaison des modèles.....	33
Figure 22 : Exemple d'une formalisation machine à états pour le comportement d'un déplacement d'objet (HPTS++ (Lamarche & Donikian, 2002))	35
Figure 23 : Exemple de comportement d'un objet synoptique présenté dans (Willans & Harrison, 2001)	36
Figure 24 : Diagramme de classes du méta-modèle VEHA : machine à états comportementale (Chevaillier et al., 2009).....	37
Figure 25 : Méta-modèle ontologique de DOMAIN-DL (Barot, 2014)	38
Figure 26 : Concepts d'Objet-Relation extrait de #FIVE (Bouville et al., 2017)	38
Figure 27 : Application d'assistance à l'opérateur en réalité augmentée (en haut) et formation en réalité virtuelle (en bas) basées sur le modèle extrait de (Richard et al., 2018)	39
Figure 28 : Fonctionnement des composants principaux de la bibliothèque #SEVEN (Claude et al., 2014)	40
Figure 29 : Exemples de structures de base pour réaliser des ensembles de scénarios complexes avec : les places (en jaune) ; les transitions (en bleus) ; les senseurs (en rouge) et les effecteurs (en vert) (Claude et al., 2014).....	41
Figure 30 : Interface de l'éditeur de réseaux #SEVEN (Claude et al., 2014).....	42
Figure 31 : Architecture de la plateforme HUMANS.....	43
Figure 32 : Relations temporelles d'Allen (Allen, 1983).....	44
Figure 33 : Méta-modèle ontologique de ACTIVITY-DL	44
Figure 34 : Exemple de modèle représenté en ACTIVITY-DL	44
Figure 35 : Méta-modèle ontologique de CAUSALITY-DL	45
Figure 36 : Exemple de modèle représenté en CAUSALITY -DL	45
Figure 37 : Architecture de la plateforme MASCARET	46

Figure 38 : (à gauche) Exemple d'automates définis suivant HAVE ; (à droite) Exemple d'agencement d'actions assignées à différents collectifs.....	47
Figure 39 : Illustrations du modèle de données basé sur les Procédure, Job et Task. Extrait de (Martínez et al., 2014).	49
Figure 40 : Ontologie des opérations sur des systèmes industriels proposée par INOOVAS.....	50
Figure 41 : Processus de création de contenu et d'usage en RA/RV via l'ontologie INOOVAS, extrait de (Havard et al., 2017)	50
Figure 42 : Étape d'édition de procédure en réalité virtuelle basée sur le framework INOOVAS. (a) l'expert sélectionne le composant, (b) l'expert sélectionne le type d'opération, (c) l'expert paramètre l'opération.	51
Figure 43 : Exemple simple d'un automate fini appliqué sur un portillon de métro.....	55
Figure 44 : Représentation graphique des nœuds d'activité.....	56
Figure 45 : Exemple de diagramme d'activité.....	57
Figure 46 : Exemple d'activité HAVE (Chevaillier et al., 2012)	57
Figure 47 : Exemple d'un réseau de Petri simple	58
Figure 48 : Exemple de séquençage en réseau de Petri Sauf	59
Figure 49 : Exemple de grafcet dans l'industrie.....	60
Figure 50 : Exemple en Grafcet de divergence et convergence « OU » à gauche, et d'une divergence / convergence « ET » à droite.	60
Figure 51 : Extrait de syntaxe graphique de Grafchart (Johnsson & Årzén, 1998)	61
Figure 52 : Les principaux éléments constituant la syntaxe de Grafchart (Theorin, 2013)	62
Figure 53 : Une étape Grafchart disposant d'une action.....	62
Figure 54 : Exemple de représentation CTT (Buttussi et al., 2013; Vidani & Chittaro, 2009).....	64
Figure 55 : Exemple de AND/OR Graphs (Johannsmeier & Haddadin, 2017) avec les rectangles bleus étant les OR et les cercles rouges les AND.	65
Figure 56 : Interface WEAVR dans Unity3D	66
Figure 57 : Interface de l'éditeur de scénario d'Interact dans Unity développé par Light & Shadow.	66
Figure 58 : Interface de création de procédure en réalité augmentée avec DiotaEditor	67
Figure 59 : L'architecture INTERVALES : Partie création	74
Figure 60 : Architecture INTERVALES : Partie usage de l'utilisateur final.....	75
Figure 61 : Diagramme de classe du moteur d'Entity-Feature	77

Figure 62 : Création d'une relation depuis une preuve de concept du framework INTERVALES. (A) Sélection de la première feature à connecter sur l'Entity « P280 » (en jaune) ; (B) Sélection de la seconde figure à connecter sur l'entity « P360 » (en jaune), la première feature sélectionnée apparaît en bleu ; (C) Résultat de la relation mise en place.	77
Figure 63 : (a) Déclaration de deux relations sans règles de compatibilité complexes. (b) Déclaration d'une unique relation en utilisant une règle de compatibilité complexe. (c) Déclaration d'une unique relation mais en ne spécifiant aucune propriété.	79
Figure 64 : Diagramme de classe du moteur de relation.	80
Figure 65 : Exemple de RelationSlider entre une Entity « P360 » et « LARD_1 ». (à gauche) la feuille d'instruction ; (à droite) la réalisation de la relation en RV, avec la pièce fantôme en jaune.	80
Figure 66 : Modèle d'orchestration des tâches et objectifs.	81
Figure 67 : Éléments du premier formalisme.	82
Figure 68 : À gauche, exemple de scénario avec seulement les concepts de bases. À droite, le même scénario avec la nécessité d'ajouter un nouvel élément de langage.	82
Figure 69 : Formalisme proposé appliqué au moteur de scénario INTERVALES.	83
Figure 70 : Blocs de construction du formalisme INTERVALES.	83
Figure 71 : Cas d'exemple dans le cas de plusieurs connexions sortantes.	84
Figure 72 : Exemple de scénario en utilisant le formalisme INTERVALES (Vert : Validé, Bleu : Sensibilisé, Rouge : Désensibilisé)	86
Figure 73 : Organisation globale des développements du framework INTERVALES.	87
Figure 74 : Placement d'une FeatureARTracking sur une Entity « P360 » au sein d'Unity 3D.	88
Figure 75 : Diagramme de classe de la librairie INTERVALES_Library.dll.	89
Figure 76 : Diagramme de classe simplifié d'INTERVALES.UnityPackage.	91
Figure 77 : Diagramme de séquence du fonctionnement du Maker RelationSnapOrientedMaker.	92
Figure 78 : Inspecteur permettant la configuration d'une Entity.	93
Figure 79 : Inspecteur permettant la configuration d'une FeatureSnapOriented.	93
Figure 80 : À gauche, le menu rapide pour instancier INTERVALES. À droite, l'instance d'INTERVALES.	94
Figure 81: Workflow d'utilisation du framework INTERVALES en BPMN.	97
Figure 82: (a) L'atelier industriel. (b) Assemblage sur un poste de travail assisté par un bras robot UR10. (c) Vue de l'atelier industriel en RV. (d) Vue CAO du poste de travail cobotique. Toutes les photos présentées ont été prises au laboratoire de LINEACT CESI.	98
Figure 83 : Cartographie des flux d'assemblage et visuel du produit assemblé.	99

Figure 84 : Procédure d'assemblage assistée par un cobot en RV et en RA avec la conception de postes manuels, le prototype numérique et la production d'une formation d'assemblage basée sur INTERVALES et d'un assistant en RA également basé sur INTERVALES	100
Figure 85: Placement des Features sur P360 en se basant sur la feuille d'instructions. (A) : La feuille d'instructions fournie ; (B) : Le modèle 3D de P360 ; (C) : Placement des Features dans Unity ; (D) : Configuration de l'inspecteur de la Feature SnapOriented dans Unity3D. (E) : L'état de la scène INTERVALES en JSON.	102
Figure 86 : Exemple de scénario au format de données INTERVALES en JSON	103
Figure 87 : Positionnement de FeatureARTracking sur l'entity P360 au sein du moteur Unity 3D.	104
Figure 88 : Preuve de concept du guidage à l'assemblage en réalité augmentée.	104
Figure 89 : Environnement virtuel de l'application « Tutoriel OREKA VR ».	106
Figure 90 : Environnement virtuel de la formation aux mesures en physique nucléaire.	106
Figure 91 : Exemple d'actions génériques mis en place dans le projet.	107
Figure 92 : (à gauche) Montage de l'écran de contrôle à réaliser. (au milieu) Placement d'une vis durant l'assemblage de l'équipement. (à droite) Utilisation d'un outil (visseuse) pour l'assemblage.	107
Figure 93 : Interface graphique du premier prototype d'éditeur de scénario INTERVALES	110
Figure 94 : Série d'opérations utilisées pour l'expérience	110
Figure 95: (À gauche) Boîte à moustaches du score SUS en pourcentage selon le type de groupe. (À droite) Boîte à moustaches des temps de réalisation selon le type de groupe.	112
Figure 96 : Résultats fournis par l'outil d'analyse UEQ avec à gauche les développeurs, et à droite, les experts métiers.	112
Figure 97 : Workflow d'événement proposé.	114
Figure 98 : Exemple de fiche d'instructions d'un poste de travail	115
Figure 99 : (A) Actions réalisées en RV par un développeur / expert métier ; (B) Actions effectuées enregistrées ; (C) Action convertie en tâches à faire dans le scénario créé par l'authoring-by-doing.	116
Figure 100 : Interface d'édition de scénario de l'éditeur 2D.	117
Figure 101 : (en haut) L'assemblage en cours de réalisation ; (en bas) Fenêtre montrant le scénario en cours d'édition par « authoring-by-doing ».	117
Figure 102 : Menu radial pour l'authoring-by-doing.	118
Figure 103 : Répartition des participants	119
Figure 104 : Extrait des fiches d'instructions P10 et P20	120

Figure 105 : Extrait de la fiche d'instructions P30	121
Figure 106 : Déroulement de l'expérimentation	122
Figure 107 : Photos prises pendant la réalisation de l'expérience. Avec à gauche, via l'authoring-by-doing et à droite, via l'éditeur 2D.....	122
Figure 108 : Version courte de l'UEQ	126
Figure 109 : Répartition des âges des participants.....	127
Figure 110 : Répartition du niveau de formation des participants	128
Figure 111 : À gauche, répartition du niveau en réalité virtuelle des participants. À droite, répartition du niveau d'aisance manuelle des participants	128
Figure 112 : (À gauche) Boîte à moustaches des temps d'édition selon le passage pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le passage pour l'édition AbD...	130
Figure 113 : (À gauche) Histogramme du nombre d'erreurs selon le passage pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le passage pour l'édition AbD.....	131
Figure 114 : (À gauche) Boîte à moustaches du nombre d'actions selon le passage pour l'édition 2D. (À droite) Boîte à moustaches du nombre d'actions selon le passage pour l'édition AbD.	133
Figure 115 : Répartition du niveau des scénarios édités suivant l'outils d'édition en pourcentage, avec à gauche le 1 ^{er} passage et à droite, le 2 ^{ème}	134
Figure 116 : (À gauche) Histogramme des niveaux d'édition selon le passage pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le passage pour l'édition AbD.	134
Figure 117 : Résultats de l'XP1. (À gauche) Histogramme des niveaux d'édition selon le profil pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le profil pour l'édition AbD.	137
Figure 118 : Résultats de l'XP2. (À gauche) Histogramme des niveaux d'édition selon le profil pour l'édition 2D. (À droite) Histogramme des niveaux d'édition selon le profil pour l'édition AbD+2D.	137
Figure 119 : Résultats de l'XP1. (À gauche) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INFO. (À droite) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INDUS.....	138
Figure 120 : Résultats de l'XP2. (À gauche) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INFO. (À droite) Histogramme des niveaux d'édition selon les outils d'édition pour les profils INDUS.....	139
Figure 121 : Résultats de l'XP1. (À gauche) Histogramme du nombre d'erreurs selon le profil pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le profil pour l'édition AbD. ..	141
Figure 122 : Résultats de l'XP2. (À gauche) Histogramme du nombre d'erreurs selon le profil pour l'édition 2D. (À droite) Histogramme du nombre d'erreurs selon le profil pour l'édition AbD+2D.	142

Figure 123 : Résultats de l'XP1. (À gauche) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INFO. (À droite) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INDUS.....	143
Figure 124 : Résultats de l'XP2. (À gauche) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INFO. (À droite) Histogramme du nombre d'erreurs selon l'outils d'édition pour les profils INDUS.....	144
Figure 125 : Résultats de l'XP1. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition Abd.....	146
Figure 126 : Résultats de l'XP2. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition Abd+2D.	147
Figure 127 : Résultats de l'XP1. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition Abd.....	148
Figure 128 : Résultats de l'XP2. (À gauche) Boîte à moustaches des temps d'édition selon le profil pour l'édition 2D. (À droite) Boîte à moustaches des temps d'édition selon le profil pour l'édition Abd+2D.	150
Figure 129 : Résultats de l'XP1. (À gauche) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition 2D. (À droite) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition Abd.....	152
Figure 130 : Résultats de l'XP2. (À gauche) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition 2D. (À droite) Boîte à moustaches du nombre moyen d'actions selon le profil pour l'édition Abd+2D.	153
Figure 131 : Résultats de l'XP1. (À gauche) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INFO. (À droite) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INDUS.....	154
Figure 132 : Résultats de l'XP2. (À gauche) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INFO. (À droite) Boîte à moustaches du nombre d'actions selon l'outil d'édition pour les profils INDUS.....	155
Figure 133 : Résultats de l'outils d'analyse UEQ pour l'édition 2D avec, à gauche, les profils informatiques et à droite, les profils industriels	158
Figure 134 : Résultats de l'outils d'analyse UEQ pour l'authoring-by-doing avec, à gauche, les profils informatiques et à droite, les profils industriels.	158
Figure 135 : Benchmark résultant des données sur les profils INFO avec l'outil d'analyse UEQ. À gauche, l'édition 2D et à droite, l'authoring-by-doing.	160

Figure 136 : Benchmark résultant des données sur les profils INDUS avec l’outil d’analyse UEQ. À gauche, l’édition 2D et à droite, l’authoring-by-doing.	160
Figure 137 : Benchmark résultant des données sur l’authoring-by-doing avec l’outil d’analyse UEQ	161
Figure 138 : Benchmark résultant des données sur l’édition 2D avec l’outil d’analyse UEQ	161
Figure 139 : Répartition des préférences des participants concernant l’outil d’édition préféré.	162
Figure 140 : Environnement 3D du jumeau numérique de la plateforme UFP (Rouen)	169
Figure 141 : Environnement 3D du jumeau numérique de la plateforme UAFAM (Nanterre).....	169
Figure 142 : Architecture et composants des jumeaux numériques et physiques intégrant INTERVALS	170
Figure 143 : Groupes technologiques et les ressources support (Danjou et al., 2017b)	185

9 Liste des tableaux

Tableau 1 : Synthèse des modèles existants	52
Tableau 2 : Synthèse des représentations graphiques du séquençement et leurs variantes utilisées par les modèles de scénario. « ? » indique que les données n'ont pas pu être vérifiées dû à un souci d'accessibilité.	68
Tableau 3 : Liste des features nécessaires à la réalisation de l'assemblage présenté et leur description associée.	102
Tableau 4 : Synthèse des caractéristiques INTERVALES appliquées aux cas d'usages présentés.	108
Tableau 5 : Tableau résumant les différentes possibilités du niveau global suivant le niveau de respect des étapes et du paramétrage.	124
Tableau 6 : Statistique générale du temps d'éditations (en secondes) pour les deux outils entre XP1-1 et XP1-2.	129
Tableau 7 : Résultats du test de Levene pour XP1-1_2D_Temps vs XP1-2_2D_Temps (a) et XP1-1_AbD_Temps vs XP1-2_AbD_Temps (b).	129
Tableau 8 : Résultats du test de Shapiro pour les temps des deux passages. * indique que l'hypothèse de normalité n'est pas validée.	129
Tableau 9 : Résultats du test de Mann et Whitney pour les temps des deux passages. (a) XP1-1_2D_Temps vs XP1-2_2D_Temps et (b) XP1-1_AbD_Temps vs XP1-2_AbD_Temps. * représente les différences statistiquement significatives.	130
Tableau 10 : Statistique générale du nombre d'erreurs pour les deux outils entre XP1-1 et XP1-2.	130
Tableau 11 : Résultats du test de Levene et du test de Shapiro pour le nombre d'erreurs des deux passages. * indique que l'hypothèse n'est pas validée.	131
Tableau 12 : Résultats du test de Mann et Whitney pour le nombre d'erreurs des deux passages. (a) XP1-1_2D_Erreur vs XP1-2_2D_Erreur et (b) XP1-1_AbD_Erreur vs XP1-2_AbD_Erreur. * représente les différences statistiquement significatives.	131
Tableau 13 : Statistique générale du nombre d'actions pour les deux outils entre XP1-1 et XP1-2.	132
Tableau 14 : Résultats du test de Shapiro pour le nombre d'actions des deux passages. * indique que l'hypothèse n'est pas validée.	132

Tableau 15 : Résultats du test de Mann et Whitney pour le nombre d'actions des deux passages. (a) XP1-1_2D_Action vs XP1-2_2D_Action et (b) XP1-1_AbD_Action vs XP1-2_AbD_Action. * représente les différences statistiquement significatives.	132
Tableau 16 : Statistique générale du niveau du scénario pour les deux outils entre XP1-1 et XP1-2.	133
Tableau 17 : Résultats du test de Mann et Whitney pour les niveaux des scénarios finaux des deux passages. (a) XP1-1_2D_Niveau vs XP1-2_2D_Niveau et (b) XP1-1_AbD_Niveau vs XP1-2_AbD_Niveau. * représente les différences statistiquement significatives.....	134
Tableau 18 : Synthèse des différences entre le 1 ^{er} et 2 ^{ème} passage pour l'édition 2D et l'authoring-by-doing. * représente les différences statistiquement significatives.	135
Tableau 19 : Statistique générale sur le niveau de qualité pour l'XP1 et l'XP2 par profil et outil d'édition.....	136
Tableau 20 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les profils INFO et INDUS pour l'XP1. * représente les différences statistiquement significatives.	136
Tableau 21 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les profils INFO et INDUS pour l'XP2. * représente les différences statistiquement significatives.	137
Tableau 22 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les différents outils d'édition pour l'XP1. * représente les différences statistiquement significatives.	138
Tableau 23 : Résultats du test de Mann et Whitney pour le niveau de qualité du scénario entre les différents outils d'édition pour l'XP2. * représente les différences statistiquement significatives.	139
Tableau 24 : Statistique générale sur le nombre d'erreurs pour l'XP1 et l'XP2 par profil et outil d'édition.....	140
Tableau 25 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP1 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.	141
Tableau 26 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs pour l'XP1 entre les profils INFO et INDUS pour chaque outil d'édition. * représente les différences statistiquement significatives.....	141
Tableau 27 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP2 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.	142

Tableau 28 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs pour l'XP2 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.	142
Tableau 29 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP1 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.	143
Tableau 30 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs entre les différents outils d'édition pour chaque profil. * représente les différences statistiquement significatives.	143
Tableau 31 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'erreurs de l'XP2 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.	144
Tableau 32 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'erreurs entre les différents outils d'édition pour l'XP2. * représente les différences statistiquement significatives.	144
Tableau 33 : Statistique générale sur le temps d'édition pour l'XP1 et l'XP2 par profil et outil d'édition.	145
Tableau 34 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP1 entre les profils INFO et INDUS par outil d'édition. * indique que l'hypothèse n'est pas validée.	146
Tableau 35 : Résultats du test de Mann et Whitney et du t-test pour les temps d'édition entre les profils INFO et INDUS pour l'XP1.	146
Tableau 36 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP2 entre les profils INFO et INDUS par outil d'édition. * indique que l'hypothèse n'est pas validée.	147
Tableau 37 : Résultats du t-test pour les temps d'édition entre les profils INFO et INDUS pour l'XP2.	147
Tableau 38 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition de l'XP1 entre les différents outils d'édition par profils. * indique que l'hypothèse n'est pas validée.	148
Tableau 39 : Résultats du test de Mann et Whitney et du T-Test pour les temps d'édition entre les différents outils d'édition pour l'XP1. * représente les différences statistiquement significatives.	148
Tableau 40 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du temps d'édition entre les différents outils d'édition pour l'XP2. * indique que l'hypothèse n'est pas validée.	149

Tableau 41 : Résultats du test du t-test pour les temps d'édition entre les différents outils d'édition de l'XP2. * représente les différences statistiquement significatives.....	149
Tableau 42 : Statistique générale sur le nombre d'actions pour l'XP1 et l'XP2 par profil et outil d'édition.....	150
Tableau 43 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP1 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.	151
Tableau 44 : Résultats du test de Mann et Whitney et du T-Test pour les moyennes du nombre d'actions de l'XP1 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.....	151
Tableau 45 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP2 entre les profils INFO et INDUS. * indique que l'hypothèse n'est pas validée.	152
Tableau 46 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions de l'XP2 entre les profils INFO et INDUS. * représente les différences statistiquement significatives.	153
Tableau 47 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP1 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.	154
Tableau 48 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions entre les différents outils d'édition. * représente les différences statistiquement significatives.....	154
Tableau 49 : Résultats du test de Levene et du test de Shapiro pour les comparaisons du nombre d'actions de l'XP2 entre les différents outils d'édition. * indique que l'hypothèse n'est pas validée.	155
Tableau 50 : Résultats du test de Mann et Whitney pour les moyennes du nombre d'actions de l'XP2 entre les différents outils d'édition. * représente les différences statistiquement significatives.....	155
Tableau 51 : Synthèse des différences entre les profils informatiques et industriels pour l'édition 2D et l'authoring-by-doing. * représente les différences statistiquement significatives.	156
Tableau 52 : Synthèse des différences entre l'édition 2D et l'authoring-by-doing pour les profils informatiques et industriels. * représente les différences statistiquement significatives.	157
Tableau 53 : Résultats du test de Shapiro sur les résultats UEQ « overall » pour l'XP1 par profil et outil d'édition. * indique que l'hypothèse de normalité n'est pas validée.	159
Tableau 54 : Résultats du test de Levene pour les comparaisons UEQ entre les profils INFO et INDUS.	159
Tableau 55 : Résultats du test de Mann et Whitney et du t-test pour les résultats UEQ entre les profils INFO et INDUS. * représente les différences statistiquement significatives.	159

Tableau 56 : Résultats du test de Levene pour les comparaisons UEQ entre les différents outils d'édition.....	161
Tableau 57 : Résultats du test de Mann et Whitney pour les résultats UEQ entre les outils 2D et AbD. * représente les différences statistiquement significatives.....	161
Tableau 58 : Synthèse des hypothèses et résultats obtenus.....	163

10 Annexes

10.1 Groupes technologiques et les ressources support de l'industrie 4.0

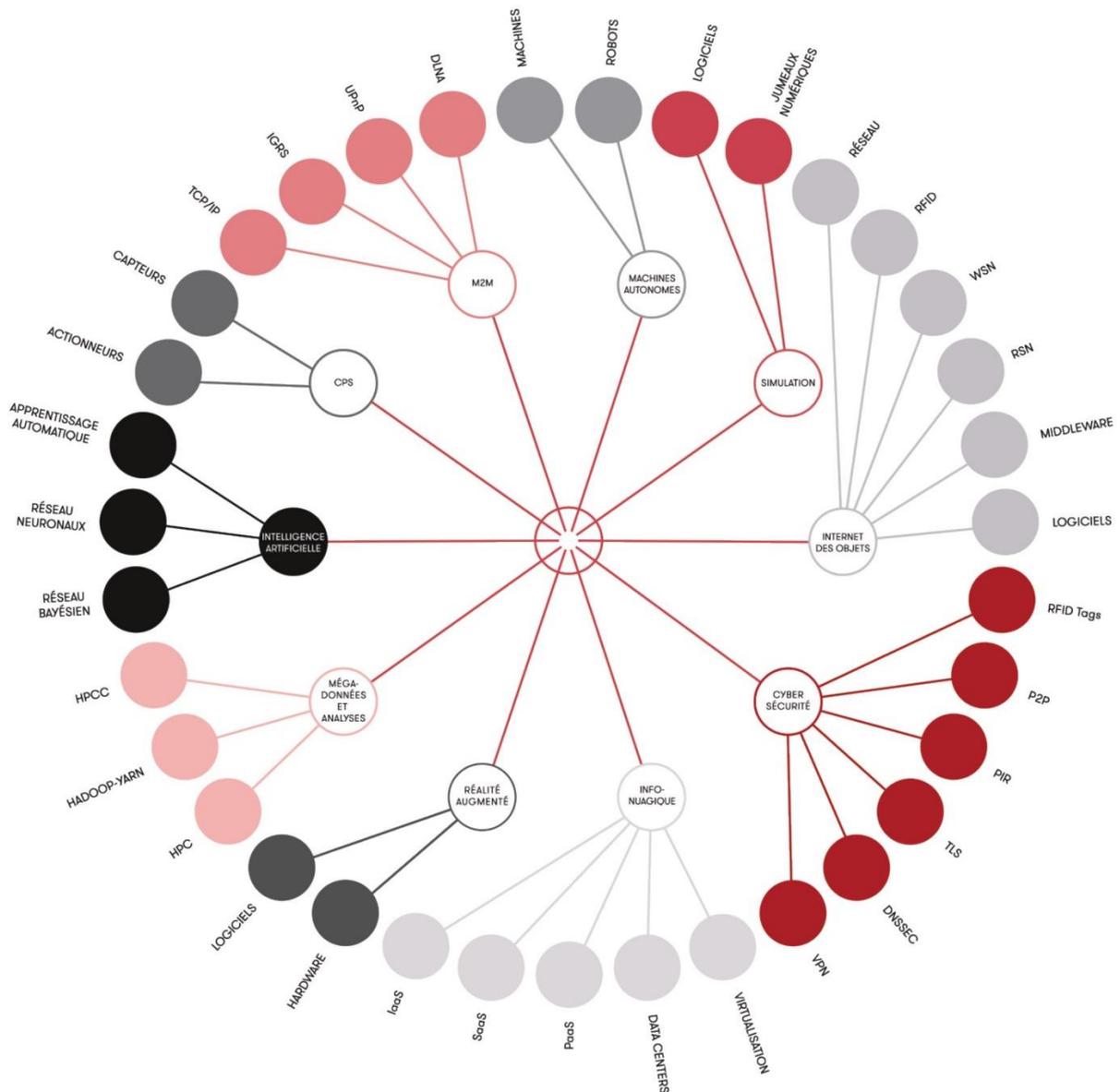


Figure 143 : Groupes technologiques et les ressources support (Danjou et al., 2017b)

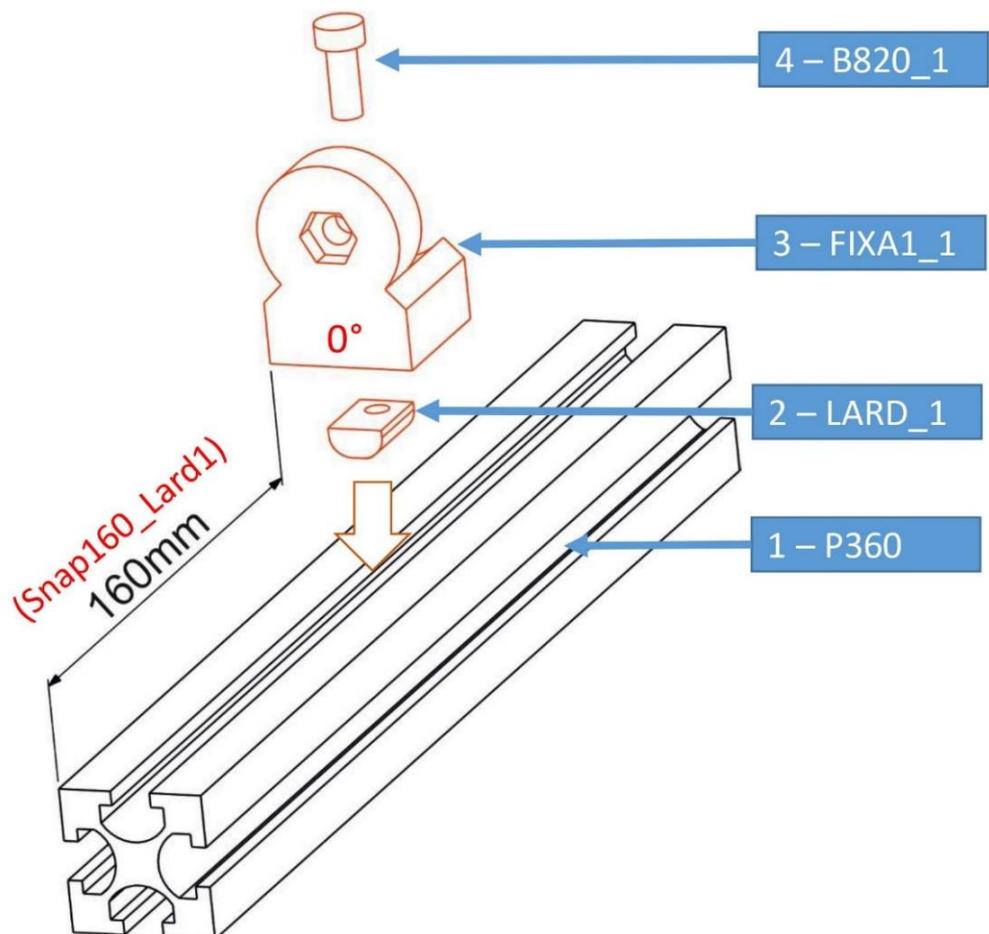
10.2 Fiches d'instructions d'assemblage



Fiche d'instructions visuelles

Modèle :	Poste :
BOOSTER	1
Process :	Opération :
Sous ensemble 1	10

N° d'étape	Article	Description de la tâche	Quantités
1	P360	Prendre un profil de 360 mm	1
2	LARD_1	Placer un lardon à 160 mm (Snap160_Lard1) sur le profil	1
3	FIXA1_1	Placer la fixation d'angle 1 sur le lardon (0°)	1
4	B820_1	Placer le boulon M8 x 20 sur la fixation d'angle 1	1

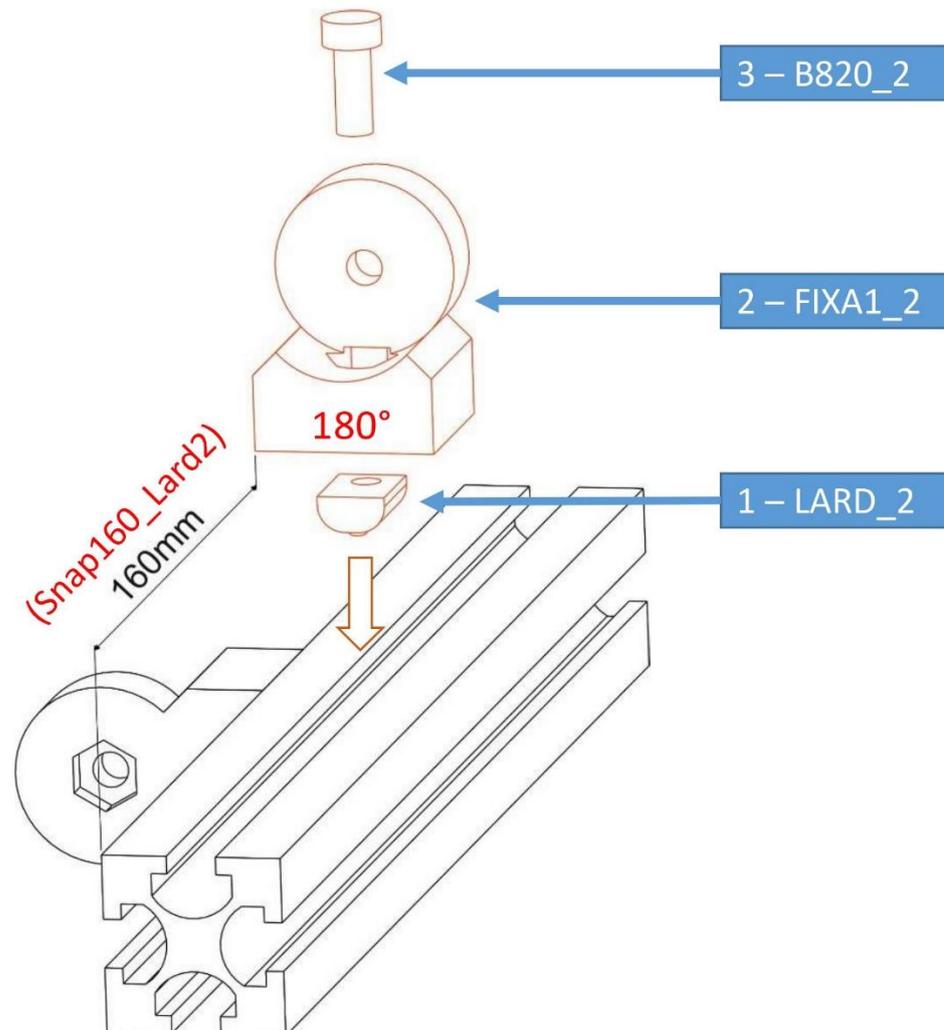




Fiche d'instructions visuelles

Modèle :	Poste :
BOOSTER	1
Process :	Opération :
Sous ensemble 1	20

N° d'étape	Article	Description	Quantités
1	LARD_2	Placer un lardon à 160 mm (Snap160_Lard2) sur le profil	1
2	FIXA1_2	Placer la fixation d'angle 1 sur le lardon (180°)	1
3	B820_2	Placer le boulon M8 x 20 sur la fixation d'angle 1	1

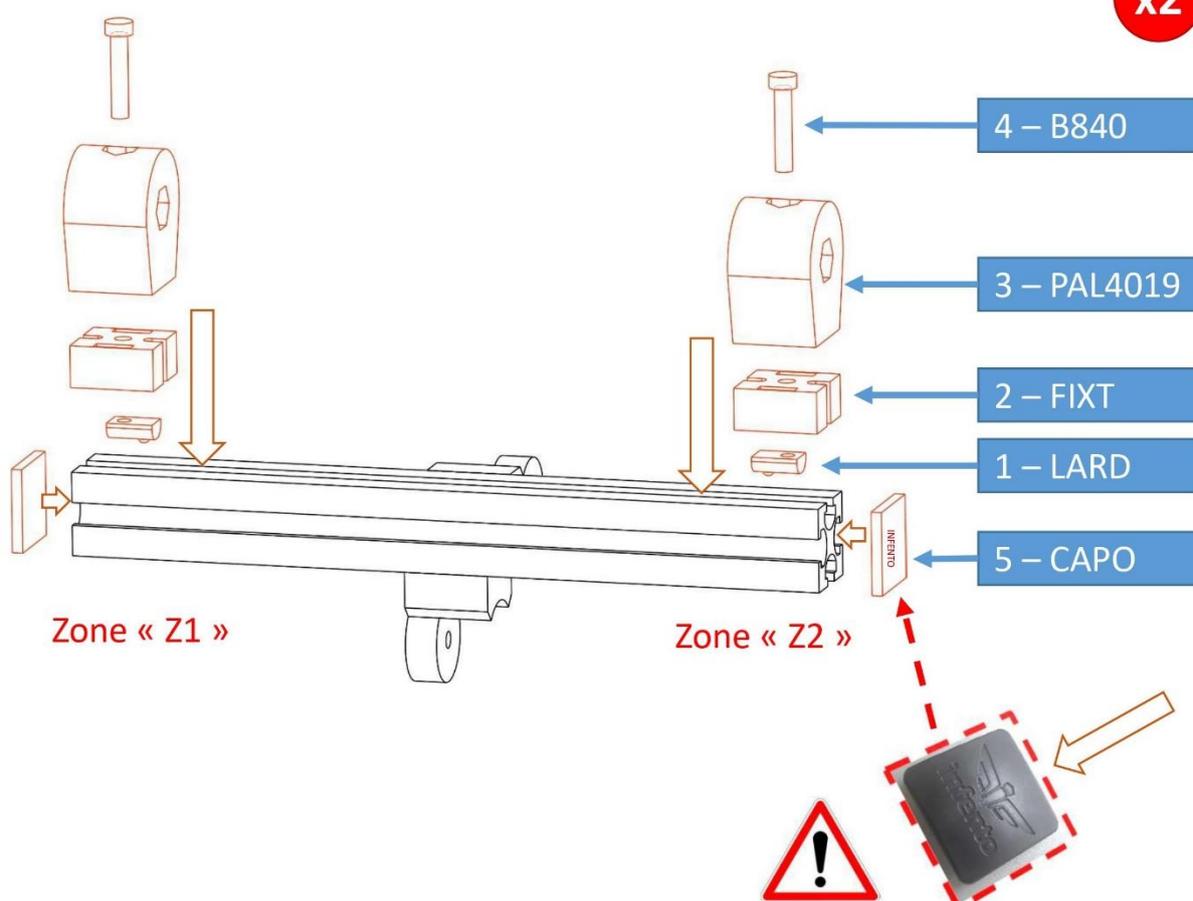




Fiche d'instructions visuelles

Modèle :	Poste :
BOOSTER	1
Process :	Opération :
Sous ensemble 1	30

N° d'étape	Article	Description	Quantités
1	LARD	Placer un lardon sur le profil	x2
2	FIXT	Placer une fixation en T sur le lardon	x2
3	PAL4019	Placer un bloc de palier 40x40x19 sur la fixation en T	x2
4	B840	Placer un boulon M8x40 sur le bloc de palier 40x40x19	x2
5	CAPO	Placer un capot sur une extrémité du profil	x2

x2

10.3 Questionnaire System-Usability-Scale

System Usability Scale

	Pas du tout d'accord				Tout à fait d'accord
1. Je pense que je vais utiliser ce service fréquemment	1	2	3	4	5
2. Je trouve ce service inutilement complexe	1	2	3	4	5
3. Je pense que ce service est facile à utiliser	1	2	3	4	5
4. Je pense que j'aurai besoin de l'aide d'un technicien pour être capable d'utiliser ce service	1	2	3	4	5
5. J'ai trouvé que les différentes fonctions de ce service ont été bien intégrées	1	2	3	4	5
6. Je pense qu'il y a trop d'incohérence dans ce service.	1	2	3	4	5
7. J'imagine que la plupart des gens serait capable d'apprendre à utiliser ce services très rapidement.	1	2	3	4	5
8. J'ai trouvé ce service très lourd à utiliser.	1	2	3	4	5
9. Je me sentais très en confiance en utilisant ce service.	1	2	3	4	5
10. J'ai besoin d'apprendre beaucoup de choses avant de pouvoir utiliser ce service.	1	2	3	4	5

10.4 Questionnaire UEQ

Dites-nous maintenant ce que vous pensez de notre produit (une seule réponse possible par qualité/défaut).

	1	2	3	4	5	6	7		
Agaçant	<input type="radio"/>	Agréable	1						
Incompréhensible	<input type="radio"/>	Compréhensible	2						
Moderne	<input type="radio"/>	Sans fantaisie	3						
Appropriation simple	<input type="radio"/>	Appropriation compliquée	4						
Apporte de la valeur	<input type="radio"/>	Peu de valeur ajoutée	5						
Ennuyeux	<input type="radio"/>	Captivant	6						
Inintéressant	<input type="radio"/>	Intéressant	7						
Imprévisible	<input type="radio"/>	Prévisible	8						
Rapide	<input type="radio"/>	Lent	9						
Original	<input type="radio"/>	Conventionnel	10						
Rigide	<input type="radio"/>	Facilitant	11						
Bien	<input type="radio"/>	Médiocre	12						
Compliqué	<input type="radio"/>	Simple	13						
Repoussant	<input type="radio"/>	Attractif	14						
Habituel	<input type="radio"/>	Avant-gardiste	15						
Désagréable	<input type="radio"/>	Agréable	16						
Sécurisant	<input type="radio"/>	Insécurisant	17						
Stimulant	<input type="radio"/>	Soporifique	18						
Répond aux attentes	<input type="radio"/>	Ne répond pas aux attentes	19						
Inefficace	<input type="radio"/>	Efficace	20						
Clair	<input type="radio"/>	Déroutant	21						
Non pragmatique	<input type="radio"/>	Pragmatique	22						
Sobre	<input type="radio"/>	Surchargé	23						
Attrayant	<input type="radio"/>	Rébarbatif	24						
Sympathique	<input type="radio"/>	Inamical	25						
Conservateur	<input type="radio"/>	Innovant	26						

10.5 Questionnaire final

INTERVALES Scenario Editor - Retour utilisateur final

Merci de votre participation à ce sondage.

Ce sondage permet d'obtenir votre retour d'expérience. Il est composé de trois questionnaires et ne prendra que 5 minutes.

Merci de cliquer sur le bouton ci-dessous pour démarrer.

***Obligatoire**

Information sur
l'utilisateur

Cette partie permet de vous connaître pour l'entretien qui suivra ce questionnaire.

1. Prénom et première lettre du nom *

2. Groupe *

Une seule réponse possible.

- Groupe "INFO"
- Groupe "INDUS"

3. Suite à cette expérience, préférez-vous l'édition 2D, l'authoring by doing ou la combinaison des deux ? *

Une seule réponse possible.

- Edition 2D
- Authoring by doing
- La combinaison des deux

4. Pourquoi ce choix ? *

5. Avez-vous des retours / avis d'ajout de fonctionnalité ?

Ce contenu n'est ni rédigé, ni cautionné par Google.

Google Forms

11 Bibliographie

- Aehnelt, M., & Bader, S. (2018). Providing and adapting information assistance for smart assembly stations. *Lecture Notes in Networks and Systems*, 15(January), 540–562. https://doi.org/10.1007/978-3-319-56994-9_37
- Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Readings in Qualitative Reasoning About Physical Systems*, 361–372. <https://doi.org/10.1016/B978-1-4832-1447-4.50033-X>
- Amokrane, K. (2010). Suivi de l'apprenant en environnement virtuel pour la formation à la prévention des risques sur des sites SEVESO. *Http://Www.Theses.Fr*. <http://www.theses.fr/2010COMP1874>
- Applications du CAVE*. (2017). <https://www.realite-virtuelle.com/cave-realite-virtuelle-vr>
- Artaud, A. (1938). *Le Théâtre et son double*. Gallimard.
- Arzén, K.-E. (1991). Sequential function charts for knowledge-based, real-time applications. *Annual Review in Automatic Programming*, 16, 91–96. [https://doi.org/10.1016/0066-4138\(91\)90015-4](https://doi.org/10.1016/0066-4138(91)90015-4)
- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355–385. <https://doi.org/10.1162/pres.1997.6.4.355>
- Badawi, M., & Donikian, S. (2004). *Autonomous agents interacting with their virtual environment through synoptic objects*. <https://www.researchgate.net/publication/228919558>
- Bader, S., & Aehnelt, M. (2014). Tracking assembly processes and providing assistance in smart factories. *ICAART 2014 - Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, 1, 161–168. <https://doi.org/10.5220/0004822701610168>
- Badets, A., Havard, V., Richard, K., & Baudry, D. (2020). *Using collaborative VR technology for Lean Manufacturing Training: a case study*.
- Barot, C. (2014). *Scénarisation d'environnements virtuels. Vers un équilibre entre contrôle, cohérence et adaptabilité* [Université de Technologie de Compiègne]. <https://tel.archives-ouvertes.fr/tel-00980537v2>
- Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., & Stein, L. A. (2004). OWL Web Ontology Language Reference. In *W3C Recommendation* (Vol. 2008, Issue February). <http://www.w3.org/TR/owl-ref/>
- Berg, L. P., & Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality*, 21(1). <https://doi.org/10.1007/s10055-016-0293-9>
- Borsci, S., Lawson, G., & Broome, S. (2015). Empirical evidence, evaluation criteria and challenges for the effectiveness of virtual and mixed reality tools for training operators of car service maintenance. In *Computers in Industry* (Vol. 67, pp. 17–26). <https://doi.org/10.1016/j.compind.2014.12.002>
- Bouville, R., Gouranton, V., Boggini, T., Nouviale, F., & Arnaldi, B. (2017). #FIVE: High-level components for developing collaborative and interactive virtual environments. *2015 IEEE 8th*

- Workshop on Software Engineering and Architectures for Realtime Interactive Systems, SEARIS 2015*, 33–40. <https://doi.org/10.1109/SEARIS.2015.7854099>
- Brom, C., & Abonyi, A. (2006). Petri nets for game plot. *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*, 3, 6–13. <https://pdfs.semanticscholar.org/72bb/b73733f075d42ab8cbb55eff1ec1245ff42e.pdf>
- Brom, C., Šisler, V., & Holan, T. (2007). Story manager in “Europe 2045” uses petri nets. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4871 LNCS, 38–50.
- Buche, C., Querrec, R., de Loor, P., & Chevaillier, P. (2011). *Mascaret: Pedagogical multi-agents system for virtual environment for training*. (Vol. 2, Issue 4). <https://hal.archives-ouvertes.fr/hal-00610857>
- Burkhardt, J.-M., Lourdeaux, D., Couix, S., & Rouillé, M. (2009). La modélisation de l'activité humaine finalisée. *Le Traité de La Réalité Virtuelle : L'humain Virtuel (Vol. 5)*, 213–245.
- Buttussi, F., Pellis, T., Vidani, C., Pausler, A., Carchietti, D., & Chittaro, E. (2013). Evaluation of a 3D serious game for advanced life support retraining. *International Journal of Medical Informatics*, 82(9), 798–809. <https://doi.org/10.1016/j.ijmedinf.2013.05.007>
- Carpentier, K., Lourdeaux, D., & Mouttapa-Thouvenin, I. (2013). Dynamic selection of learning situations in virtual environment. *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, 2, 101–110. <https://doi.org/10.5220/0004247901010110>
- Casarin, J., Pacqueraud, N., & Bechmann, D. (2018). UMI3D : A Unity3D Toolbox to Support CSCW Systems Properties in Generic 3D User Interfaces. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW), 1–20. <https://doi.org/10.1145/3274298>
- Caudell, T. P., & Mizell, D. W. (1992). *Augmented reality: an application of heads-up display technology to manual manufacturing processes*. 659–669 vol.2. <https://doi.org/10.1109/hicss.1992.183317>
- Chevaillier, P., Querrec, R., & Septseault, C. (2009). VEHA, un métamodèle d'environnement virtuel informé et structuré. *Techniques et Sciences Informatiques*, 28(6–7), 715–740. <https://doi.org/10.3166/tsi.28.715-740>
- Chevaillier, P., Trinh, T. H., Barange, M., De Loor, P., Devillers, F., Soler, J., & Querrec, R. (2012). Semantic modeling of virtual environments using MASCARET. *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems, SEARIS 2012*, 1–8. <https://doi.org/10.1109/SEARIS.2012.6231174>
- Claude, G. (2016). Séquencement d'actions en environnement virtuel collaboratif [Rennes, INSA]. In [Http://Www.Theses.Fr](http://www.Theses.Fr). <https://www.theses.fr/2016ISAR0010>
- Claude, G., Gouranton, V., Bouville Berthelot, R., Arnaldi, B., Claude, G., Gouranton, V., Bouville Berthelot, R., & Arnaldi, B. (2014). *a Sensor Effector Based Scenarios Model for Driving Collaborative Virtual Environment*. <https://hal.archives-ouvertes.fr/hal-01086237>
- Cremer, J., Kearney, J., & Papis, Y. (1995). HCSM: a framework for behavior and scenario control in virtual environments. *ACM Transactions on Modeling and Computer Simulation*, 5(3), 242–267. <https://doi.org/10.1145/217853.217857>
- Damiani, L., Demartini, M., Guizzi, G., Revetria, R., & Tonelli, F. (2018). Augmented and virtual reality

- applications in industrial systems: A qualitative review towards the industry 4.0 era. *IFAC-PapersOnLine*, 51(11), 624–630. <https://doi.org/10.1016/j.ifacol.2018.08.388>
- Danjou, C., Rivest, L., & Pellerin, R. (2017a). *Douze positionnements stratégiques pour l' Industrie 4.0 : entre processus , produit et service , de la surveillance à l' autonomie .* 3–5. [http://espace2.etsmtl.ca/id/eprint/16253/1/Rivest L 2017 16253.pdf](http://espace2.etsmtl.ca/id/eprint/16253/1/Rivest_L_2017_16253.pdf)
- Danjou, C., Rivest, L., & Pellerin, R. (2017b). *Industrie 4.0 : des pistes pour aborder l'ère du numérique et de la connectivité.* 27. https://cefrio.qc.ca/media/1268/pme_20-le-passage-au-numerique.pdf
- David, R., & Alla, H. (1992). *Du Grafset aux réseaux de Petri.*
- De Mello, L. S. H., & Sanderson, A. C. (1990). AND/OR Graph Representation of Assembly Plans. *IEEE Transactions on Robotics and Automation*, 6(2), 188–199. <https://doi.org/10.1109/70.54734>
- de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers and Industrial Engineering*, 139, 106159. <https://doi.org/10.1016/j.cie.2019.106159>
- Devillers, F., & Donikian, S. (2003). A scenario language to orchestrate virtual world evolution. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2003.*
- Dopico, M., Gomez, A., De la Fuente, D., García, N., Rosillo, R., & Puche, J. (2016). A vision of industry 4.0 from an artificial intelligence point of view. *Proceedings of the 2016 International Conference on Artificial Intelligence, ICAI 2016 - WORLDCOMP 2016*, 407–413.
- Drath, R., & Horch, A. (2014). Industrie 4.0: Hit or hype? [Industry Forum]. *IEEE Industrial Electronics Magazine*, 8(2), 56–58. <https://doi.org/10.1109/MIE.2014.2312079>
- Echterbille, P. (2014). *Évolution de la suite logicielle FORMID pour la conception, l'exécution et le suivi de situations d'apprentissage à distance.* <https://dumas.ccsd.cnrs.fr/dumas-01196781>
- Erkoyuncu, J. A., del Amo, I. F., Dalle Mura, M., Roy, R., & Dini, G. (2017). Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality. *CIRP Annals - Manufacturing Technology*, 66(1), 465–468. <https://doi.org/10.1016/j.cirp.2017.04.006>
- Ernst, F., & Frische, P. (2015). Industry 4.0 / Industrial Internet of Things - Related Technologies and Requirements for a Successful Digital Transformation: An Investigation of Manufacturing Businesses Worldwide. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2698137>
- Feng, Z., González, V. A., Mutch, C., Amor, R., Rahouti, A., Baghouz, A., Li, N., & Cabrera-Guerrero, G. (2020). Towards a customizable immersive virtual reality serious game for earthquake emergency training. *Advanced Engineering Informatics*, 46, 101134. <https://doi.org/10.1016/j.aei.2020.101134>
- Friedrich, W., Jahn, D., & Schmidt, L. (2002). ARVIKA-Augmented Reality for Development, Production and Service. *ISMAR, 2002*, 3–4.
- Fuchs, P., Moreau, G., Burkhardt, J.-M., & Coquillart, S. (2006). *Le traité de la réalité virtuelle Volume 2-L'interfaçage: l'immersion et l'interaction en environnement virtuel. Traité de La Réalité Virtuelle.*
- Funk, M. (2016). Augmented reality at the workplace: a context-aware assistive system using in-

- situ projection. *PHD Thesis*, 250. <https://doi.org/10.18419/opus-8997>
- Gajšek, B., Stradovnik, S., & Hace, A. (2020). Sustainable Move towards Flexible, Robotic, Human-Involving Workplace. *Sustainability*, 12(16), 6590. <https://doi.org/10.3390/su12166590>
- Galambos, P., Csapó, Á., Zentay, P., Fülöp, I. M., Haidegger, T., Baranyi, P., & Rudas, I. J. (2015). Design, programming and orchestration of heterogeneous manufacturing systems through VR-powered remote collaboration. *Robotics and Computer-Integrated Manufacturing*, 33, 68–77. <https://doi.org/10.1016/j.rcim.2014.08.012>
- Geng, J., Song, X., Pan, Y., Tang, J., Liu, Y., Zhao, D., & Ma, Y. (2020). A systematic design method of adaptive augmented reality work instruction for complex industrial operations. *Computers in Industry*, 119, 103229. <https://doi.org/10.1016/j.compind.2020.103229>
- Gerbaud, S. (2008). *Contribution à la formation en réalité virtuelle: Scénarios collaboratifs et intégration d'humains virtuels collaborant avec des utilisateurs réels*. 167.
- Gerbaud, S., & Arnaldi, B. (2009). Humains virtuels et collaboration dans un environnement virtuel de formation. *Revue Des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques*, 28(6–7), 741–766. <https://doi.org/10.3166/TSI.28.741-766>
- Gerbaud, S., Mollet, N., & Arnaldi, B. (2007). Virtual environments for training: From individual learning to collaboration with humanoids. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4469 LNCS, 116–127. https://doi.org/10.1007/978-3-540-73011-8_14
- Gong, Y., Su, C., & Wei, Y. (2019). An Interactive Task Modeling Method for the Development of Augmented Reality Maintenance Guiding System. *Proceedings - 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2019, 1*, 136–140. <https://doi.org/10.1109/IHMSC.2019.00039>
- Gordon, A., Van Lent, M., Van Velsen, M., Carpenter, P., & Jhala, A. (2004). Branching storylines in virtual reality environments for leadership development. *Proceedings of the National Conference on Artificial Intelligence*, 844–851.
- Gorecky, D., Khamis, M., & Mura, K. (2017). Introduction and establishment of virtual training in the factory of the future. *International Journal of Computer Integrated Manufacturing*, 30(1), 182–190. <https://doi.org/10.1080/0951192X.2015.1067918>
- Havard, V. (2018). *Développement de méthodes et outils basés sur la réalité augmentée et virtuelle pour l'assistance ou l'apprentissage d'opérations dans un contexte industriel To cite this version: HAL Id: tel-01713362 Présentée et soutenue par*. 204. <https://tel.archives-ouvertes.fr/tel-01713362v2>
- Havard, V., Baudry, D., Jeanne, B., Louis, A., & Savatier, X. (2021). A use case study comparing augmented reality (AR) and electronic document-based maintenance instructions considering tasks complexity and operator competency level. *Virtual Reality 2021 25:4*, 25(4), 999–1014. <https://doi.org/10.1007/S10055-020-00493-Z>
- Havard, V., Baudry, D., Louis, A., & Mazari, B. (2015). Augmented reality maintenance demonstrator and associated modelling. *2015 IEEE Virtual Reality Conference, VR 2015 - Proceedings*, 329–330. <https://doi.org/10.1109/VR.2015.7223429>
- Havard, V., Baudry, D., Savatier, X., Jeanne, B., Louis, A., & Mazari, B. (2016). Augmented industrial maintenance (AIM): A case study for evaluating and comparing with paper and video media supports. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial*

- Intelligence and Lecture Notes in Bioinformatics*), 9768, 302–320. https://doi.org/10.1007/978-3-319-40621-3_22
- Havard, V., Jeanne, B., Lacomblez, M., & Baudry, D. (2019). Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations. *Production & Manufacturing Research*, 7(1), 472–489. <https://doi.org/10.1080/21693277.2019.1660283>
- Havard, V., Jeanne, B., Savatier, X., & Baudry, D. (2017). Inoovas - Industrial Ontology for Operation in Virtual and Augmented Scene: the architecture. *International Conference on Control Decision and Information Technology*. <https://doi.org/10.1109/CoDIT.2017.8102608>
- Havard, V., Trigunayat, A., Richard, K., & Baudry, D. (2019). Collaborative virtual reality decision tool for planning industrial shop floor layouts. *Procedia CIRP*, 81, 1295–1300. <https://doi.org/10.1016/j.procir.2019.04.016>
- Hayes, B., & Scassellati, B. (2016). Autonomously constructing hierarchical task networks for planning and human-robot collaboration. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June, 5469–5476. <https://doi.org/10.1109/ICRA.2016.7487760>
- Henderson, S., & Feiner, S. (2011). Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Transactions on Visualization and Computer Graphics*, 17(10), 1355–1368. <https://doi.org/10.1109/TVCG.2010.245>
- Hermann, M., Pentek, T., Otto, B., & Pentek, T. *. (2016). Design Principles for Industrie 4.0 Scenarios: A Literature Review Competence Center Corporate Data Quality (CC CDQ) View project Industrial Data Space View project Design Principles for Industrie 4.0 Scenarios: A Literature Review. *49th Hawaii International Conference on System Sciences (HICSS)*, September 2016, 16. <https://doi.org/10.13140/RG.2.2.29269.22248>
- Hervás, R., Bravo, J., & Fontecha, J. (2010). A context model based on ontological languages: A proposal for information visualization. *Journal of Universal Computer Science*, 16(12), 1539–1555. <https://doi.org/10.3217/jucs-016-12-1539>
- Hervás, R., Bravo, J., Fontecha, J., & Villarreal, V. (2013). Achieving Adaptive Augmented Reality through Ontological Context-Awareness applied to AAL Scenarios. *J. UCS*, 19(9), 1334–1349. <http://www.esi.uclm.es/www/Jesus.Fontecha/web/index.php/research-topics>
- Hill, R. W., Gratch, J., Marsella, S., Swartout, W., Rickel, J., & Traum, D. (2003). *Virtual Humans in the Mission Rehearsal Exercise System*. *The Eleventh International Automated Negotiating Agent Competition (ANAC) View project Embodied Virtual Agent View project Virtual Humans in the Mission Rehearsal Exercise System*. <https://www.researchgate.net/publication/220633128>
- ISO/IEC. (2014). *Mixed and augmented reality (MAR) reference model*. 69. <https://goo.gl/ZNdqfJ>
- Johannsmeier, L., & Haddadin, S. (2017). A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes. *IEEE Robotics and Automation Letters*, 2(1), 41–48. <https://doi.org/10.1109/LRA.2016.2535907>
- Johnsson, C., & Årzén, K.-E. (1996). *Batch recipe structuring using high-level Grafchart*. <https://www.researchgate.net/publication/228595228>
- Johnsson, C., & Årzén, K.-E. (1998). Grafchart and its Relations to Grafcet and Petri Nets. *IFAC Proceedings Volumes*, 31(15), 95–100. [https://doi.org/10.1016/s1474-6670\(17\)40535-0](https://doi.org/10.1016/s1474-6670(17)40535-0)
- Johnsson, C., & Årzén, K.-E. (1999). Grafchart and grafcet: A comparison between two graphical

- languages aimed for sequential control applications. *IFAC Proceedings Volumes*, 32(2), 19–24. [https://doi.org/10.1016/S1474-6670\(17\)56006-1](https://doi.org/10.1016/S1474-6670(17)56006-1)
- Kallmann, M., & Thalmann, D. (1999). Direct 3D interaction with Smart Objects. *ACM Symposium on Virtual Reality Software and Technology, Proceedings, VRST*, 124–130. <https://doi.org/10.1145/323663.323683>
- Keller, M., Rosenberg, M., Brettel, M., & Friederichsen, N. (2014). How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(1), 37–44. <http://waset>.
- Lacoche, J., Duval, T., Arnaldi, B., Maisel, E., & Royan, J. (2017). Providing plasticity and redistribution for 3D user interfaces using the D3PART model. *Journal on Multimodal User Interfaces*, 11(2), 197–210. <https://doi.org/10.1007/s12193-017-0239-x>
- Lamarche, F., & Donikian, S. (2002). Automatic orchestration of behaviours through the management of resources and priority levels. In *Proceedings of the International Conference on Autonomous Agents* (Issue 3). <https://doi.org/10.1145/545121.545124>
- Lamberti, F., Manuri, F., Sanna, A., Paravati, G., Pezzolla, P., & Montuschi, P. (2014). Challenges, Opportunities and Future Trends of Emerging Techniques for Augmented Reality-based Maintenance. *IEEE Transactions on Emerging Topics in Computing*, 2(4), 411–421. <https://doi.org/10.1109/TETC.2014.2368833>
- Lanquepin, V., Lourdeaux, D., Barot, C., Carpentier, K., Lhomme, M., & Amokrane, K. (2013). HUMANS: A HUMAN models based artificial environments software platform. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2466816.2466826>
- Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5298 LNCS, 63–76. https://doi.org/10.1007/978-3-540-89350-9_6
- Lécuyer, F., Gouranton, V., Reuzeau, A., Gagne, R., & Arnaldi, B. (2020). Action Sequencing in VR, a No-Code Approach. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12230 LNCS* (pp. 57–76). https://doi.org/10.1007/978-3-662-61983-4_4
- Lee, J. Y., & Rhee, G. (2008). Context-aware 3D visualization and collaboration services for ubiquitous cars using augmented reality. *The International Journal of Advanced Manufacturing Technology*, 37(5–6), 431–442.
- Lejeune, A. (2004). IMS Learning Design. *Distances et Savoirs*, 2(4), 409–450. <https://doi.org/10.3166/ds.2.409-450>
- Lejeune, A. (2011). Une approche auteur pour la scénarisation et le suivi de situations d'apprentissage. *Atelier Méthodes de Conception EIAH, Conférence Environnements Informatiques Pour l'Apprentissage Humain, 2011*, 1–6.
- Létondor, E. (2016). *Extensions de la suite logicielle FORMID pour assister la conception et le suivi de situations pédagogiques exploitant des simulations ou des objets tangibles*. <https://dumas.ccsd.cnrs.fr/dumas-01631003>
- Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, 34(7), 577–590.

- <https://doi.org/10.1080/10447318.2018.1455307>
- Lugrin, J. L., & Cavazza, M. (2006). AI-based world behaviour for emergent narratives. *International Conference on Advances in Computer Entertainment Technology 2006*. <https://doi.org/10.1145/1178823.1178853>
- Lutfi, M., & Valerdi, R. (2020). Virtual Reality in Model Based Systems Engineering: A Review Paper. *Communications in Computer and Information Science*, 1294, 197–205. https://doi.org/10.1007/978-3-030-60703-6_25
- Malik, A. A., Masood, T., & Bilberg, A. (2020). Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace. *International Journal of Computer Integrated Manufacturing*, 33(1), 22–37. <https://doi.org/10.1080/0951192X.2019.1690685>
- Marion, N. (2010). Modélisation de scénarios pédagogiques pour les environnements de réalité virtuelle d'apprentissage humain. [Http://Www.Theses.Fr](http://www.theses.fr). <http://www.theses.fr/2010BRES2009>
- Martínez, H., Laukkanen, S., & Mattila, J. (2014). A New Flexible Augmented Reality Platform for Development of Maintenance and Educational Applications. *International Journal of Virtual Worlds and Human Computer Interaction*. <https://doi.org/10.11159/vwhci.2014.003>
- Masood, T., & Egger, J. (2020). Adopting augmented reality in the age of industrial digitalisation. *Computers in Industry*, 115, 103112. <https://doi.org/10.1016/j.compind.2019.07.002>
- Mateas, M., & Stern, A. (2002). A Behavior Language for Story-Based Believable Agents. *IEEE Intelligent Systems*, 17(4), 39–47. <https://doi.org/10.1109/MIS.2002.1024751>
- Matsas, E., & Vosniakos, G. C. (2017). Design of a virtual reality training system for human–robot collaboration in manufacturing tasks. *International Journal on Interactive Design and Manufacturing*, 11(2), 139–153. <https://doi.org/10.1007/s12008-015-0259-2>
- Mealy, G. H. (1955). A Method for Synthesizing Sequential Circuits. *Bell System Technical Journal*, 34(5), 1045–1079. <https://doi.org/10.1002/j.1538-7305.1955.tb03788.x>
- Merchant, Z., Goetz, E. T., Cifuentes, L., Keeney-Kennicutt, W., & Davis, T. J. (2014). Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. <https://doi.org/10.1016/j.compedu.2013.07.033>
- Milgram, P., & Kishimo, F. (1994). A taxonomy of mixed reality. *IEICE Transactions on Information and Systems*, 77(12), 1321–1329.
- Moalla, M. (1985). Réseaux de Petri interprétés et Grafocet. *Technique et Science Informatique*, 4, 17–30.
- Mohseni-Kabir, A., Rich, C., Chernova, S., Sidner, C. L., & Miller, D. (2015). Interactive Hierarchical Task Learning from a Single Demonstration. *ACM/IEEE International Conference on Human-Robot Interaction, 2015-March*, 205–212. <https://doi.org/10.1145/2696454.2696474>
- Mollet, N., & Arnaldi, B. (2006). Storytelling in virtual reality for training. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3942 LNCS, 334–347. https://doi.org/10.1007/11736639_45
- Mollet, N., Gerbaud, S., & Arnaldi, B. (2007). STORM: a Generic Interaction and Behavioral Model for 3D Objects and Humanoids in a Virtual Environment. *IPT-EGVE the 13th Eurographics Symposium on Virtual Environments*, 95–100. <https://hal.inria.fr/inria-00475525>

- Moore, E. F. (1956). Gedanken-experiments on sequential machines. *Automata Studies*, 34, 129--153.
- Mott, B. W., & Lester, J. C. (2006). U-DIRECTOR: A decision-theoretic narrative planning architecture for storytelling environments. *Proceedings of the International Conference on Autonomous Agents, 2006*, 977–984. <https://doi.org/10.1145/1160633.1160808>
- Nebeling, M., & Speicher, M. (2018). The Trouble with Augmented Reality/Virtual Reality Authoring Tools. *Adjunct Proceedings - 2018 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2018*, 333–337. <https://doi.org/10.1109/ISMAR-Adjunct.2018.00098>
- OGC. (2015). *ARML 2.0 SWG*. <http://www.opengeospatial.org/projects/groups/arml2.0swg>
- Ottogalli, K., Rosquete, D., Amundarain, A., Aguinaga, I., & Borro, D. (2019). Flexible framework to model industry 4.0 processes for virtual simulators. *Applied Sciences (Switzerland)*, 9(23), 4983. <https://doi.org/10.3390/app9234983>
- Park, K. B., Choi, S. H., Kim, M., & Lee, J. Y. (2020). Deep learning-based mobile augmented reality for task assistance using 3D spatial mapping and snapshot-based RGB-D data. *Computers and Industrial Engineering*, 146, 106585. <https://doi.org/10.1016/j.cie.2020.106585>
- Paternò, F., & Santoro, C. (2003). A unified method for designing interactive systems adaptable to mobile and stationary platforms. *Interacting with Computers*, 15(3 SPEC.), 349–366. [https://doi.org/10.1016/S0953-5438\(03\)00013-4](https://doi.org/10.1016/S0953-5438(03)00013-4)
- Patle, D. S., Manca, D., Nazir, S., & Sharma, S. (2019). Operator training simulators in virtual reality environment for process operators: a review. *Virtual Reality*, 23(3), 293–311. <https://doi.org/10.1007/s10055-018-0354-3>
- Pérez, L., Diez, E., Usamentiaga, R., & García, D. F. (2019). Industrial robot control and operator training using virtual reality interfaces. *Computers in Industry*, 109, 114–120. <https://doi.org/10.1016/j.compind.2019.05.001>
- Petri, C. A. (1962). *Kommunikation mit automaten*. <http://edoc.sub.uni-hamburg.de/informatik/volltexte/2011/160/>
- Petri, C. A. (1966). *Communication with automata*.
- Querrec, R., Vallejo, P., & Buche, C. (2013). MASCARET: creating virtual learning environments from system modelling. *The Engineering Reality of Virtual Reality 2013*, 8649, 864904. <https://doi.org/10.1117/12.2008186>
- Raisamo, R., Rakkolainen, I., Majaranta, P., Salminen, K., Rantala, J., & Farooq, A. (2019). Human augmentation: Past, present and future. *International Journal of Human Computer Studies*, 131, 131–143. <https://doi.org/10.1016/j.ijhcs.2019.05.008>
- Rauch, E., Linder, C., & Dallasega, P. (2020). Anthropocentric perspective of production before and within Industry 4.0. *Computers and Industrial Engineering*, 139, 105644. <https://doi.org/10.1016/j.cie.2019.01.018>
- Réalité virtuelle : Industriels, architectes, formateurs... pourquoi tous les pros vont s'y mettre.* (2016). <https://www.usine-digitale.fr/editorial/realite-virtuelle-industriels-architectes-formateurs-pourquoi-tous-les-pros-vont-s-y-mettre.N387743>
- Richard, K., Havard, V., Jeanne, B., & Baudry, D. (2016). Modélisation de maintenance en réalité augmentée. *CONFERE'16*.

- Richard, K., Havard, V., Trigunayat, A., & Baudry, D. (2018). Modèle de données pour la création d'environnement interactif en réalité virtuelle et augmentée. *AFRV 2018: Les Journées de La Réalité Virtuelle*.
- Rickel, J., Marsella, S., Gratch, J., Hill, R., Traum, D., & Swartout, W. (2002). Toward a New Generation of Virtual Humans for Interactive Experiences. *IEEE Intelligent Systems*, 17(4), 32–38. <https://doi.org/10.1109/MIS.2002.1024750>
- Riedl, M. O., & Young, R. M. (2010). Narrative planning: Balancing plot and character. *Artificial Intelligence*, 39, 217–268. <http://repository.lib.ncsu.edu/ir/handle/1840.16/3940>
- Riedl, M., Saretto, C. J., & Young, R. M. (2003). Managing Interaction between Users and Agents in a Multi-agent Storytelling Environment. *Proceedings of the International Conference on Autonomous Agents*, 2, 741–748. <https://doi.org/10.1145/860690.860694>
- Röltgen, D., & Dumitrescu, R. (2020). Classification of industrial Augmented Reality use cases. *Procedia CIRP*, 91, 93–100. <https://doi.org/10.1016/j.procir.2020.01.137>
- Ruble, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision*, 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- Rüßmann, Michael Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. *The Boston Consulting Group*. https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries
- Sanfilippo, E., & Terkaj, W. (2019). Editorial: Formal Ontologies meet Industry. *Procedia Manufacturing*, 28, 174–176. <https://doi.org/10.1016/j.promfg.2018.12.028>
- Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4), 40. <https://doi.org/10.9781/ijimai.2017.445>
- Silva, A., Raimundo, G., & Paiva, A. (2003). Tell me that bit again... bringing interactivity to a virtual storyteller. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2897, 146–154. https://doi.org/10.1007/978-3-540-40014-1_19
- Stone, R. (2001). Virtual reality for interactive training: An industrial practitioner's viewpoint. *International Journal of Human Computer Studies*, 55(4), 699–711. <https://doi.org/10.1006/ijhc.2001.0497>
- Su, C.-H., Cheng, T.-W., Su, C.-H., & Cheng, T.-W. (2019). A Sustainability Innovation Experiential Learning Model for Virtual Reality Chemistry Laboratory: An Empirical Study with PLS-SEM and IPMA. *Sustainability*, 11(4), 1027. <https://doi.org/10.3390/su11041027>
- Su, C.-J., Liu, P.-T., Lin, Y.-C., & others. (2012). Automatic generation of Augmented Reality enabled pedagogical system using Object-Oriented Analysis and Design in Process Modeling. *Asia Pacific Industrial Engineering and Management Systems Conference*.
- Sugiyama, K., Tagawa, S., & Toda, M. (1981). Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2), 109–125. <https://doi.org/10.1109/TSMC.1981.4308636>

- Swartout, W., Gratch, J., Hill, R. W., Hovy, E., Marsella, S., Rickel, J., & Traum, D. (2006). Toward virtual humans. *AI Magazine*, 27(2), 96–108.
- Swartout, W., Gratch, J., Hill, R. W., Traum, D., Swartout, W., Gratch, J., Hill, R., Hovy, E., Lindheim, R., Marsella, S., Rickel, J., & Traum, D. (2004). *Simulation Meets Hollywood SEE PROFILE SIMULATION MEETS HOLLYWOOD: Integrating Graphics, Sound, Story and Character for Immersive Simulation*. https://doi.org/10.1007/1-4020-3051-7_13
- Terrier, R., Gouranton, V., Bach, C., Pallamin, N., Arnaldi, B., & Scenario-Based Vr, B. A. (2020). *Scenario-based VR Framework for Product Design Framework for Product Design Scenario-based VR Framework for Product Design*. 1–8. <https://hal.archives-ouvertes.fr/hal-02502648>
- Theorin, A. (2013). *Adapting Grafchart for Industrial Automation*. https://pdfs.semanticscholar.org/56df/9df4c1f137655acd1331bb4d194e908789fe.pdf?_ga=2.129810309.84394454.1564647733-584875602.1564647733
- Traum, D., Rickel, J., Gratch, J., & Marsella, S. (2003). Negotiation over Tasks in Hybrid Human-Agent Teams for Simulation-Based Training. *Proceedings of the International Conference on Autonomous Agents*, 2, 441–448. <https://doi.org/10.1145/860575.860646>
- Vanneste, P., Huang, Y., Park, J. Y., Cornillie, F., Decloedt, B., & den Noortgate, W. Van. (2020). Cognitive support for assembly operations by means of augmented reality: an exploratory study. *International Journal of Human-Computer Studies*, 102480. <https://doi.org/10.1016/j.ijhcs.2020.102480>
- Vidani, A. C., & Chittaro, L. (2009). Using a task modeling formalism in the design of serious games for emergency medical procedures. *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications, VS-GAMES 2009*, 95–102. <https://doi.org/10.1109/VS-GAMES.2009.24>
- Wang, J., Liu, S., Liu, J., & Du, Z. (2017). An assembly process model based on object-oriented hierarchical time Petri Nets. *AIP Conference Proceedings*, 1834(April). <https://doi.org/10.1063/1.4981588>
- Wang, Z., Bai, X., Zhang, S., Billingham, M., He, W., Wang, Y., Han, D., Chen, G., & Li, J. (2021). The role of user-centered AR instruction in improving novice spatial cognition in a high-precision procedural task. *Advanced Engineering Informatics*, 47, 101250. <https://doi.org/10.1016/j.aei.2021.101250>
- Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6), 80. <https://doi.org/10.2307/3001968>
- Willans, J. S., & Harrison, M. D. (2001). Prototyping Pre-implementation Designs of Virtual Environment Behaviour. In *LNCS* (Vol. 2254).
- Zabell, S. L. (2008). On student's 1908 article "the probable error of a mean." *Journal of the American Statistical Association*, 103(481), 1–7. <https://doi.org/10.1198/016214508000000030>
- Zhu, J., Ong, S. K., & Nee, A. Y. C. (2013). An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology*, 66(9–12), 1699–1714.