

# Retour d'expérience sur l'utilisation d'un jumeau numérique pour l'enseignement de l'automatisme

François Gellot<sup>1</sup>, Stéphane Lecasse<sup>1</sup>, Bernard Riera<sup>1</sup>, Alexandre Philippot<sup>1</sup>  
francois.gellot@univ-reims.fr

<sup>1</sup> CReSTIC (EA3804), Université de Reims Champagne-Ardenne,  
Moulin de la Housse, 51687 Reims, France

**RESUME :** Dans le cadre d'un enseignement progressif de l'automatisme, nous verrons dans cet article qu'il est intéressant d'utiliser un jumeau numérique d'un système réel pour aller vers la programmation de systèmes complexes industriels. Nous utilisons pour cela trois logiciels Siemens communiquant ensemble à savoir NX MCD, pour le jumeau numérique, PLC SIM Advanced pour le simulateur d'automate et TIA Portal pour la partie programmation. Les TP utilisant le jumeau numérique pour réaliser la commande d'un poste de notre cellule flexible ayant été réalisés par deux profils d'élèves ingénieurs en apprentissage de niveau bac+4, un questionnaire leur a été soumis afin de récupérer leur retour et leur perception du jumeau numérique. Globalement les retours sont bons même s'il faut distinguer dans l'analyse du questionnaire les profils automaticiens, des profils plus maintenance.

**Mots clés :** apprentissage, automatisme, jumeau numérique, système réel, retour d'expérience, IEC 61131- 3.

## 1 INTRODUCTION

Dans le cadre de l'apprentissage de la programmation de l'automatisme, au sein des formations de l'EiSINE (Ecole d'ingénieurs en Sciences Industrielles et Numérique), une progression des acquis est mise en place. Après l'apprentissage des bases de la logique combinatoire et séquentielle en L1 (Licence 1) et L2, les bases du grafset (IEC 60848:2013) et du LD (Ladder Diagram), un des 5 langages de programmation des API (Automate Programmable Industriel) de la norme IEC 61131- 3, sont abordés en L3. Dans un premier temps, pour les mettre en pratique, nous utilisons en TP, ce que nous appelons des TP boutons, où chaque binôme dispose d'un API réel dont les cartes d'entrées sont connectées sur une platine avec des boutons poussoirs ou des sélecteurs ; les ordres de l'API s'affichent via des leds sur les cartes de sorties. De cette façon, c'est l'étudiant qui génère l'occurrence d'évènements pour valider la commande qu'il a réalisée dans son API. Dans un deuxième temps, pour aller plus loin dans la formation, des structures complexes de grafset et de cahier des charges sont étudiées et mises en pratiques dans des TP où, nous utilisons un logiciel de simulation de PO (Parties Opératives), Factory IO [1] [2] avec API réel ou simulé. Cette fois-ci, l'occurrence d'évènements est directement générée par le simulateur de PO et les conséquences des ordres envoyés se visualisent sur les actions générées sur le système simulé.

Enfin en 4<sup>ème</sup> année, des langages de programmation API comme le ST et le IL (IEC 61131- 3) sont enseignés ainsi que de la programmation API plus poussée. Pour ne pas perdre de vue le coté professionnalisant de la formation, nous souhaitons aussi mettre l'étudiant face à un système réel à commander. Le coût d'un système réel automatisé est tel qu'il n'est pas possible de mettre à disposition de chaque binôme un système réel. Le système réel (unique) doit donc être partagé par 8 binômes au

maximum au cours d'une séance de TP ce qui réduit fortement le temps de mise à disposition de la PO réelle pour faire les tests. Suite au projet de recherche FFCA (Factories of Future Champagne-Ardenne) du CPER PFEXCEL de la région Grand-Est, nous disposons du jumeau numérique [3] de notre atelier flexible Cellflex [4] (fig 1), ainsi chaque binôme dispose de sa propre machine pour les phases de tests. Cela permet d'initier les étudiants au virtual commissioning [5], processus permettant une évaluation complète du code de l'API, et de son éventuel débogage, avant sa mise en service sur la partie opérative réelle. Cette technique est de plus en plus répandue en industrie avec le développement de l'industrie 4.0.

Dans cet article après avoir décrit succinctement la cellule flexible sur laquelle a été réalisée le jumeau numérique, nous verrons comment s'architecture le dispositif de TP utilisant le jumeau numérique à travers une solution « tout simulé », finalement très proche d'une solution « tout réel ». Une fois la solution simulée validée, les binômes peuvent transférer leur programme sur le système réel pour adapter leur commande aux quelques différences près, avec le simulateur. Dans la troisième partie de cet article, nous verrons le retour d'expérience des étudiants par suite de l'utilisation conjointe du jumeau numérique et du système réel au travers de 2 profils d'étudiants en apprentissage de niveau bac+4. Avant de conclure, nous ferons un bilan de cette utilisation de jumeau numérique pour l'apprentissage de l'automatisme et de ces évolutions futures.

## 2 DESCRIPTION DU SYSTÈME

La cellule flexible (fig 1) aussi appelée CellFlex de l'URCA (Université de Reims Champagne Ardenne) permet aux enseignants chercheurs du CReSTIC (Centre de Recherche en Sciences et Technologies de l'Information et de la Communication) d'effectuer des recherches dans le domaine de l'industrie 4.0.

C'est également une opportunité pour les étudiants de l'EiSINe (Ecole d'ingénieur en Sciences Industrielles et Numérique) ainsi que de l'UTT (Université Technologique de Troyes) en branche AII (Automatique & Informatique Industrielle) de se former en automatisme, en robotique et sur les technologies de l'industrie 4.0 sur un système très proche des systèmes industriels.

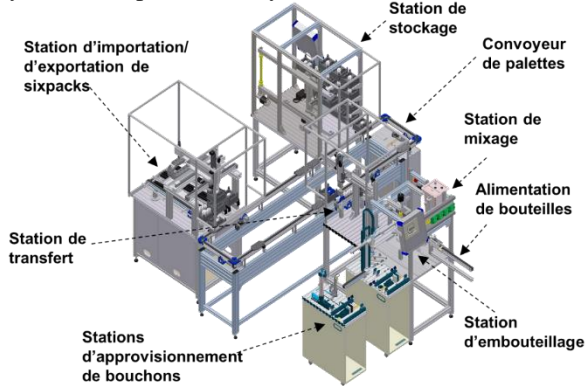


fig 1 : La cellule Cellflex

La CellFlex est composée de 7 stations et d'un AGR (Automated Guided Robot). Les 6 premières stations possèdent chacune leur propre API pour réaliser le contrôle/commande et peuvent fonctionner de façon indépendante ou échanger entre elles sur réseau Profinet pour interagir les unes avec les autres :

- Station de bouchonnage
- Station d'embouteillage/transfert
- Convoyeur central
- Station de stockage
- Station import/export
- Station de mixage
- Station robot de vidange (non représenté)
- AGR (non représenté)

L'atelier permet de réaliser (fig 2) un mélange à partir de 3 solutions, d'en remplir une bouteille et de la bouchonner (dite le « produit »), pour finir par en faire un ensemble de 6 produits assemblés dans un carton (dit le « sixpack »), chaque élément, produit et carton, étant tracé via une puce RFID. Chaque sixpack réalisé peut être stocké ou exporté.

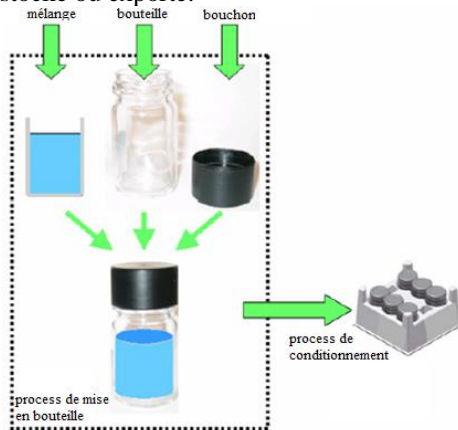


fig 2 : Produits réalisés

La station utilisée pour cet article est la station de bouchonnage (fig 3). Cette station est découpée en plusieurs sous-parties : - la sélection d'un bouchon de couleur - l'éjection de celui-ci en dehors du magasin - le déplacement du bouchon du magasin vers le convoyeur au moyen d'un vérin rotatif - l'activation du convoyeur pour emmener le bouchon en zone de réception - la prise du bouchon pour la dépose sur une bouteille en zone d'embouteillage par un bras mobile.

L'automate de la station des bouchons est constitué uniquement d'une CPU SIEMENS de type ET 200 CPU 1512SP-1 PN auquel est associé sur réseau Profinet, un bloc d'E/S déportées intégrant 16 entrées TOR et 13 sorties TOR. Le cadre du TP étant posé, nous allons maintenant décrire le dispositif simulé de travaux pratiques.

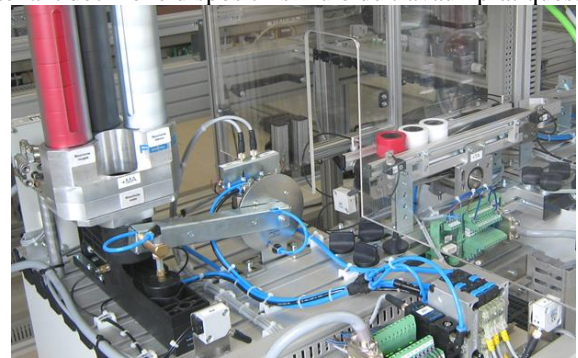


fig 3 : La station des bouchons

### 3 DESCRIPTION DU DISPOSITIF DE SIMULATION

Le système simulé est constitué de trois parties (fig 4) :

- Le jumeau numérique NX MCD
- Le simulateur d'API, PLC SIM Advanced
- Le logiciel de programmation d'API, TIA PORTAL

Ces trois outils logiciels, installés sur chaque poste informatique, sont des softwares de la société SIEMENS pour l'automatisme et la CAO.

#### 3.1 Le jumeau numérique

Le projet de jumeau numérique de notre cellule flexible a vu le jour dans le cadre du CPER Factories of Future Champagne-Ardenne (FFCA). Le CPER FFCA a permis de faire évoluer notre cellule flexible vers l'industrie 4.0, en permettant entre autres d'obtenir les financements pour développer son jumeau numérique. Celui-ci a coûté 42 950€<sup>HT</sup>, soit 35% du budget d'évolution de la cellule au sein du projet FFCA. Le choix de notre jumeau numérique a été guidé par les retours peu nombreux de l'appel d'offres et le type de notre cellule flexible qui est une solution FESTO pour la partie mécanique et SIEMENS pour le contrôle commande. Le choix s'est donc porté vers le logiciel NX MCD. Possédant un ensemble de fichiers CAO au format STP pour 5 des 6 stations du système réel, il a été possible de faire développer le jumeau numérique de 5 stations du système par la société

FEALINX [6], sous-traitante de la société ACE [7] ayant obtenu le marché. Le jumeau est évolutif à condition d'avoir de bonnes notions de mécanique puisque NX MCD est d'abord un logiciel de CAO auquel est associé des fonctionnalités de virtual commissioning. Il est aussi possible de désactiver des capteurs ou des actionneurs sur le système simulé pour générer des pannes afin de travailler sur le diagnostic du système.

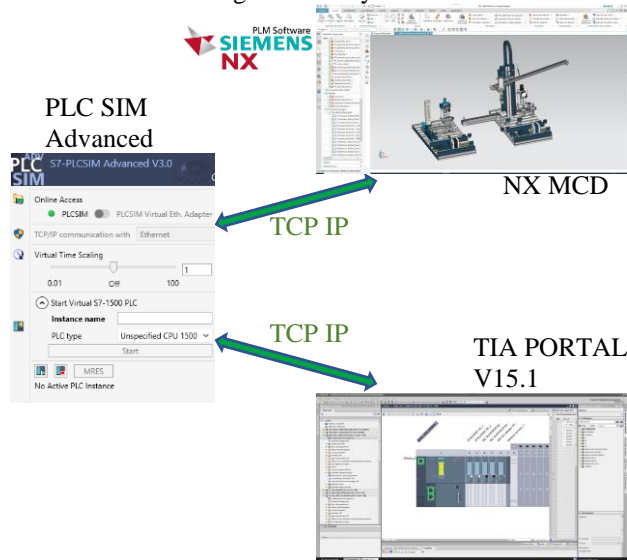


fig 4 : Schéma du dispositif de simulation

Pour communiquer avec le jumeau numérique, NX MCD fournit plusieurs solutions : OPC DA, OPC UA, SHM, Matlab, PLC Sim Advanced, TCP, UDP et Profinet. Une fois le mode de communication choisi, chaque entrée de NX MCD est mise en relation avec la sortie correspondante du système de contrôle/commande à travers le protocole d'échanges et vice versa pour les sorties.

### 3.2 Le simulateur d'automate PLC SIM Advanced

Au même titre que nous avons fait le choix d'une partie opérative simulée, nous avons choisi d'utiliser un automate simulé. Cette fonctionnalité nous est fournie par le simulateur d'automate PLC SIM Advanced de Siemens. L'avantage de cet outil est qu'il est capable de communiquer avec NX MCD de façon très aisée. De plus, un gros point fort est qu'il est multi-automates simulés, et cela permet de travailler sur la communication entre API de façon complètement transparente. Un gros inconvénient de PLC SIM Advanced est qu'il est très « gourmand » en ressource UC, et lorsque l'on passe à 3, 4 ou 5 API simulés auquel s'ajoute le jumeau NX MCD, on se retrouve avec une UC occupée à 100%. Le PC n'arrive alors plus à fournir les ressources nécessaires. Dans notre cas, nous n'aurons qu'un API simulé, puisque la partie de la CellFlex réelle concernée par le TP est commandée par un seul API ; nous n'aurons donc pas de surcharge de la CPU.

### 3.3 L'atelier logiciel TIA PORTAL

Que l'on travaille en mode simulé ou sur le système réel, nous utilisons le même atelier logiciel, à savoir TIA PORTAL V15.1. Nos automates réels étant des S7 1500 de chez Siemens, il nous faut utiliser TIA PORTAL, si l'on veut que le même programme fonctionne sur système réel ou sur jumeau numérique. Sous TIA PORTAL, la seule différence entre automate réel ou simulé, vient du transfert du programme vers l'automate. Si PLC SIM est en exécution, on ne peut que transférer sur l'automate simulé. Dans le cas du transfert vers le système réel, il faut arrêter l'exécution de PLC SIM afin de pouvoir, à travers le réseau, transférer le programme sur l'API réel. Au niveau de notre système réel, nous utilisons un bloc d'E/S déportées sur réseau Profinet relié à notre API. La structuration de l'API déclaré dans TIA Portal n'impacte pas son utilisation sous PLC SIM et reste complètement transparente. De même, les principes de programmation d'API sont complètement identiques.

## 4 RETOURS ETUDIANTS SUR L'UTILISATION DU DISPOSITIF [8]

Depuis 2021, l'utilisation du jumeau numérique de la Cellflex a été mise en place dans le cadre de TP. Les étudiants ont 6 séances de 3h, pour réaliser une commande fonctionnelle pour le système virtuel et réel ainsi que la programmation d'un pupitre opérateur et d'une supervision. Afin d'évaluer le ressenti des étudiants, à l'issue de l'utilisation du jumeau numérique et de la partie opérative réelle pour l'apprentissage de la programmation d'API, un questionnaire en dix points a été réalisé. Il a ensuite été complété par 2 types de profil d'élèves ingénieurs en apprentissage en 2<sup>ème</sup> année (bac+4) au sein de l'EiSINE :

- 14 étudiants GER (Génie Electrique et Robotique) pour lesquels l'automatisme fait partie du cœur de métier
- 17 étudiants MGI (Maintenance et Génie Industriel) pour lesquels l'automatisme n'est pas le cœur de métier.

Les GER et MGI ont suivi la même unité d'enseignement sur cette matière avec le même nombre d'heures.

La 1ère question posée, aborde la difficulté d'utilisation du logiciel NX MCD : Personne n'a trouvé l'utilisation très difficile. Pour 71% des étudiants GER, l'utilisation est relativement facile alors que pour les étudiants de MGI les retours sont plus mitigés ; plus de 50% d'entre eux trouvent l'utilisation difficile à relativement difficile. Cette différence vient certainement du fait que pour les MGI, il y a beaucoup plus de nouveaux concepts à acquérir dans cet enseignement que pour les GER qui côtoient de près l'automatisme durant leur période en entreprise.

La question suivante traite de la qualité des graphismes et de la cinématique dans le logiciel de simulation. Les

réponses sont quasi identiques entre les deux filières ; 74% les trouvent bonnes et 21 % très bonnes. Il y a donc un bon ressenti au niveau de la qualité graphique et de la cinématique fournies par le jumeau.

Les étudiants avaient tous à leur disposition, un cahier des charges avec descriptions de la PO et de ses E/S (Entrées/ Sorties), sans différencier système réel et système simulé. Pour la 3ième question, nous avons voulu savoir si la compréhension du fonctionnement de la PO à travers le jumeau numérique était aisée. Pour les MGI, une très grande majorité a trouvé la compréhension relativement difficile à relativement facile à parts égales. Pour les GER, la compréhension du système à travers le jumeau numérique est relativement facile à facile. On peut se demander si cette différence de perception ne vient pas du fait que les GER sont coutumiers de la notion de cahier des charges de systèmes automatisés ce qui n'est pas le cas des MGI, la différence n'étant pas à attribuer au jumeau numérique.

Les concepts de capteurs et actionneurs d'une PO, sont ensuite abordés dans les deux questions suivantes. Les deux groupes considèrent qu'ils retrouvent bien les notions de capteurs d'une PO à 68% et d'actionneurs d'une PO à 74 %, le quasi-reste étant sans avis.

Le point suivant est de savoir si les étudiants trouvent une différence ou une équivalence entre jumeau numérique et système réel. 45% d'entre eux les trouvent relativement identiques et la même proportion les trouvent pratiquement identiques. Entre GER et MGI, il y a cependant une différence de perception à ce niveau puisque les GER les trouvent quasi-identiques à totalement identiques à 78% alors que 64% des MGI les trouvent moyennement identiques.

La complexité des échanges entre NX MCD et PLC SIM concerne la septième question. 83 % des étudiants de MGI trouvent ce point complexe voir très complexe. A contrario, les étudiants de GER à 57 % trouvent ce point peu complexe. Une telle différence entre les deux groupes peut s'expliquer par le fait que les GER ont suivi l'année précédente un cours sur la communication industrielle alors que les MGI ne suivront ce cours qu'au semestre suivant. Il est à noter que la communication entre ces deux outils, même si elle a été expliquée et montrée lors du début des travaux pratiques, est complètement transparente à l'utilisation.

La question huit demande si pour les étudiants, il y a une différence de perception entre API réel et API simulé. 45 % d'entre eux répondent par l'affirmative, 29% sont sans avis et 26% ne voient pas de différence. Sur cette question, il n'y a pratiquement pas de différence d'appréciation entre les deux formations. Ce sentiment de différence est surprenant puisqu'à l'utilisation, cela n'engendre aucune différence dans la programmation de

l'API. Lorsque l'on est en mode connecté sous TIA Portal, il n'y a là encore aucune différence, de même pour tout ce qui concerne le transfert du programme dans l'API.

Au niveau de l'avant dernière question, on se pose la question d'un gain de temps ou non à utiliser un jumeau numérique pour réaliser la commande du système. Pour les GER, à 75 %, cela procure un gain de temps pour la programmation des API, ce n'est vrai qu'à 53 % pour les MGI. Globalement, moins de 10% des étudiants trouvent que c'est une perte de temps d'utiliser un jumeau numérique. Les étudiants qui ne trouvent pas un gain de temps, n'ont pas eu à se partager le même système réel, n'ont pas connu la problématique de disponibilité d'un système réel pour faire des tests.

Pour conclure sur le questionnaire, il est demandé s'il est plus difficile de programmer une commande API pour le jumeau numérique par rapport au système réel. Les étudiants à 42% ont trouvé une difficulté identique entre les deux systèmes. 50% des GER ont trouvé que programmer le jumeau numérique est plus facile que faire la commande réelle. Cela peut s'expliquer par le réglage des capteurs qui est « parfait » sur le jumeau numérique, ce qui n'est pas le cas des systèmes réels, le reste des GER étant sans avis. A contrario, les MGI estiment pour 24 % d'entre eux qu'il est plus difficile de programmer une commande pour système simulé que pour système réel. On retrouve dans cette différence de perception entre GER et MGI, l'analyse de la question 1 dans laquelle il est pressenti que pour les MGI, moins habitués aux concepts de l'automatisme, l'utilisation du jumeau numérique ajoute une difficulté supplémentaire aux concepts à intégrer, ce qui n'est pas le cas pour les étudiants de GER plus rompus à l'exercice de conception d'une commande automatisée.

Le questionnaire et le retour d'expérience sont mis en ligne à l'adresse web indiquée en [8].

## 5 RETOUR ENSEIGNANT SUR L'UTILISATION DU DISPOSITIF

L'utilisation du jumeau numérique apporte un réel plus lors des TP sur l'atelier Cellflex pour réaliser la commande de la station de bouchonnage. En effet, lorsque le jumeau n'était pas présent, la mise à disposition du système réel pour un binôme pouvait occasionner quelques tensions durant l'attente pour faire ses tests en ligne. De plus, aujourd'hui, quand les groupes passent sur le système réel, ils ont déjà une très bonne connaissance du système et de son fonctionnel, et arrivent très rapidement à corriger les 2 ou 3 bugs, dus au passage entre système simulé vers réel. Il faut noter aussi que le temps pour réaliser une commande correcte sur l'API a diminué et laisse plus de temps aux étudiants pour réaliser la programmation du pupitre opérateur ainsi que la mise en place d'une supervision.

Tout n'est pas parfait dans l'utilisation d'un jumeau numérique. Il existe malgré tout quelques différences entre le système réel et son jumeau numérique dont il faut avoir conscience :

- Au niveau des fichiers CAO à notre disposition, le système ne possède qu'une colonne d'alimentation en bouchons alors que le système réel en possède trois, afin de permettre une diversité de produits. Cette différence aurait pu être gommée en revoyant la partie CAO, pour ajouter les 2 colonnes manquantes et le système d'approvisionnement en bouchons.
- Sur le jumeau, il n'est pas possible de simuler du liquide ou de gérer l'air comprimé. Cela peut par exemple poser des problèmes pour gérer correctement une ventouse pour l'aspiration ou l'expiration.
- Sur le système réel, les bouchons sont approvisionnés par l'opérateur. Sur le simulateur, l'approvisionnement se fait soit via une commande de l'automate qu'il faut écrire, soit via un bouton sur un pupitre opérateur, simulé ou non.
- Le comportement du jumeau numérique dépend d'un moteur physique dont on ne connaît pas les performances et qui provoque parfois des situations qu'on ne rencontrerait pas sur le système réel
- Au niveau du comportement, il faut relativiser la notion de jumeau. En effet, d'un point de vue du temps, on n'a aucune garantie sur la similitude entre le comportement du système simulé et celui du système réel. Le temps de déplacement par exemple d'une pièce entre les deux extrémités d'une bande transporteuse ne peut être garanti comme équivalent entre les deux systèmes. Le temps peut dériver sur le système réel du fait de l'usure ou d'un frottement supérieur sur la bande transporteuse, de même, au niveau du jumeau, on dispose d'une échelle de temps effective qui peut dériver très fortement en fonction du taux d'occupation de l'UC du PC. Le comportement équivalent recherché entre jumeau et système réel se situe au niveau de l'ordre d'occurrences des événements pour être pris en compte par le système de contrôle/commande. De cette façon, le même programme API commande aussi bien le système réel que son jumeau, pour obtenir les mêmes réponses au cahier des charges.

## 6 CONCLUSION

Dans le cadre de l'enseignement de l'automatisme, l'utilisation d'un jumeau numérique d'une PO réelle est un plus pour se rapprocher des conditions réelles de programmation des systèmes de contrôle/ commande industriels. En effet, par l'utilisation du jumeau numérique l'étudiant a à sa disposition quasiment l'équivalent du système réel pour réaliser sa commande d'automatisme.

La solution numérique du système, permet de reproduire le système autant de fois que nécessaire et diminue donc le coût tout en augmentant la disponibilité du système pour les phases de test et de validation lors de l'apprentissage de l'automatisme. Il faut par-contre faire attention, à ce que la complexité du jumeau numérique ne prenne pas le pas sur les objectifs d'enseignement.

Dans cet article, à travers le retour du questionnaire étudiant, on peut voir que le travail sur un jumeau numérique ne pose pas globalement de problème particulier, en revanche son ressenti n'est pas le même selon que l'utilisateur est coutumier des automatismes (étudiants GER) ou beaucoup moins (étudiants MGI).

## Bibliographie

- [1] B. Riera, R. Pichard, A. Philippot, R. Saddem, F. Gellot, D. Annebicque, F. Emprin, " HOME I/O et FACTORY I/O : 2 logiciels innovants de simulation de PO pour la formation à l'automatique ", *Actes du 12<sup>ème</sup> Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CESTIS 2017), Le Mans (France), Mai 2017.*
- [2] A. Philippot, S. Lecasse, B. Riera, F. Gellot, " Développement d'un connecteur logiciel pour l'apprentissage de l'automatisme ", *Actes du 14<sup>ème</sup> Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CESTIS-EEA'2020), Valenciennes (France), Juin 2021.*
- [3] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, D. Devine, " Digital Twin: Origin to Future ", *Applied System Innovation 2021*, 4, 36.  
<https://doi.org/10.3390/asi4020036>
- [4] Site de l'atelier Cellflex 4.0 :  
<https://www.univ-reims.fr/meserp/cellflex-4.0/cellflex-4.0.9503.27026.html>
- [5] A. Heidari, O. Salamon, « Virtual Commissioning of an Existing Manufacturing Cell at Volvo Car Corporation Using DELMIA V6 », Master's Thesis, CHALMERS University of Technology, Goteborg, Sweden 2012, Report No. EX023/2012
- [6] FEALINX <http://www.fealinx.com/>
- [7] ACE (AUTOMATISMES DU CENTRE EST) <https://www.acefrance.com/fr/>
- [8] Accès au questionnaire et aux retours : <https://www.univ-reims.fr/meserp/projets/cectsis-2023/questionnaire-et-ret-etudiants-sur-l-utilisation-d-un-jumeau-numerique,26695,43717.html>

## REMERCIEMENTS

Ce travail a pu être développé grâce au projet FFCA (Factories of Future Champagne-Ardenne). Les auteurs tiennent à remercier la région Grand-Est dans le cadre du projet FFCA (CPER PFEXCEL).

Les travaux présentés ont été réalisés dans le cadre du projet DeMETeRE (Déploiement de Micro-Environnements Territoriaux pour la Réussite Etudiante), lauréat du PIA4 DemoES (Démonstrateur de l'Enseignement Supérieur) et financé au titre du plan France 2030 (ANR-21-DMES-0011).