

Spécification de la synchronisation des jumeaux numériques à l’aide d’automates temporisés et stochastiques

Pascale Marangé¹ and William Derigent¹

Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, Vandoeuvre-lès-Nancy,
54506, France
`prenom.nom@univ-lorraine.fr`

Abstract

Les jumeaux numériques s’imposent aujourd’hui comme une technologie clé pour renforcer la réactivité, la flexibilité et la résilience des systèmes de production. Grâce à leurs capacités de surveillance en temps réel et de prévision, ils s’inscrivent pleinement dans les ambitions portées par l’industrie 5.0. Toutefois, ces apports restent conditionnés à un enjeu fondamental : maintenir en permanence une synchronisation fidèle entre le jumeau numérique et son homologue physique. Or, bien que cette synchronisation constitue le socle de la connexion entre le monde réel et son modèle virtuel, elle demeure largement sous-explorée dans la littérature consacrée aux systèmes de production. Les approches existantes relèvent souvent d’un niveau conceptuel élevé, sans fournir de démarche opérationnelle pour sa mise en œuvre. Cet article propose une première contribution en vue de l’élaboration d’une méthodologie formelle de la synchronisation, en s’appuyant sur une modélisation par automates stochastiques et temporisés. L’objectif est de faire le lien entre les exigences fonctionnelles de synchronisation et les mécanismes concrets à déployer au niveau du jumeau numérique.

1 Introduction

Les jumeaux numériques, nommés également Digital Twins et notés par la suite DT, suscitent un intérêt croissant dans le domaine de l’ingénierie des systèmes de production, tant du point de vue conceptuel que technologique. Plusieurs définitions coexistent dans la littérature, notamment celles proposées par Grieves [6], Kritzinger [8], ou encore par l’ISO 23247 [11]. Dans cet article, nous nous appuyons sur la définition de la norme ISO 23247, selon laquelle un jumeau numérique est ”une représentation numérique ciblée d’un élément manufacturier observable, avec synchronisation entre cet élément et sa représentation numérique”. Cette définition présente un intérêt particulier : elle explicite la nécessité de synchroniser en permanence le modèle virtuel avec son référent physique, soulignant ainsi que la synchronisation n’est pas un aspect accessoire, mais bien une condition essentielle du bon fonctionnement du jumeau numérique.

Concrètement, cette synchronisation implique une mise à jour régulière ou déclenchée par événement du modèle numérique, en fonction de l’évolution de l’élément physique observé. Cette opération nécessite des mécanismes adaptés, que nous désignons ici sous le terme de processus de synchronisation. Ce processus, souvent perçu comme une tâche technique mineure, se révèle en réalité complexe à mettre en œuvre, en raison de plusieurs défis majeurs que nous identifions ci-après :

- Multiplicité des représentations : un jumeau numérique étant conçu en fonction des objectifs de son utilisateur, il n’existe pas une seule représentation possible d’un artefact physique, mais une multitude de modèles, chacun traduisant une vision partielle, orientée par les besoins spécifiques d’usage. Cette diversité engendre des processus de synchronisation hétérogènes, voire incompatibles entre eux.

- Hétérogénéité de granularité : même lorsqu’il s’agit de modéliser un même système au sein d’un seul domaine d’expertise, les modèles adoptés peuvent différer considérablement en termes de niveau de détail. Certains représentent le système de manière globale, d’autres de façon très fine. Il est donc indispensable que les mécanismes de synchronisation prennent en compte ces variations de granularité.
- Limites de la fidélité du modèle : un jumeau numérique reste, par définition, une approximation du réel. Se pose alors la question de la précision attendue dans la correspondance entre le jumeau et son référent physique, et donc du seuil d’erreur acceptable selon les cas d’usage.
- Contraintes techniques et ressources disponibles : la synchronisation repose sur des données issues du monde physique, collectées via des capteurs, transmises par des réseaux et stockées dans des systèmes d’information. La qualité, la fiabilité et la latence de ces infrastructures conditionnent fortement la performance du processus de synchronisation.

Ces constats soulignent la nécessité de développer des approches méthodologiques rigoureuses, capables de garantir une synchronisation efficace, souple et adaptable aux différentes configurations de jumeaux numériques. C’est dans cette perspective que s’inscrit notre contribution.

Dans un premier temps (section 2), nous proposons un état de l’art des travaux existants relatifs à la synchronisation des jumeaux numériques, en identifiant les apports existants et les limites actuelles. La section 3.3 introduit les principes d’une méthodologie de synchronisation destinée aux systèmes de production automatisés, en s’appuyant sur des modèles d’automates temporisés stochastiques pour représenter les entités physiques et numériques. Un cas d’étude, basé sur le système flexible de la plateforme S.Mart de l’AIP Lorrain, est ensuite présenté (section 4). Enfin, la section 5 conclut l’article et propose des pistes de travaux futurs.

2 Revue de la littérature

Dans le contexte des jumeaux numériques, maintenir une synchronisation effective entre le modèle virtuel et le système physique constitue un préalable indispensable à la validité du jumeau et à la fiabilité de son usage. Une telle synchronisation vise à garantir que toute évolution de l’entité physique – événementielle ou continue – soit correctement reflétée dans sa représentation numérique. En pratique, cela suppose un accès constant à des données actualisées, mais aussi la mise en œuvre de mécanismes capables d’exploiter ces données pour adapter dynamiquement le modèle numérique. Les approches développées à ce jour combinent des techniques issues de plusieurs champs : traitement et exploitation intelligente de données (data mining, apprentissage automatique), synchronisation temporelle [3], modélisation à base de connaissances, ou encore architectures hybrides mêlant modèles et données [1]. Certaines stratégies, comme la recalibration périodique des modèles sur des flux de données récents ([7], [5]), proposent des solutions concrètes pour maintenir la cohérence du jumeau numérique.

Néanmoins, la plupart de ces contributions restent peu formalisées, en particulier lorsqu’il s’agit de répondre à des exigences complexes telles que :

- la détection automatique d’écarts entre le comportement réel et prévu du système physique,
- la propagation de ces écarts dans le modèle virtuel,
- l’adaptation autonome des modèles numériques.

Un point notable concerne la fréquence et le rythme des synchronisations. Ce paramètre, souvent déterminé de façon empirique, impacte directement la réactivité, la précision et la charge des infrastructures de traitement et de communication. L'absence de méthodologies systématiques et d'outils d'aide au choix des modèles de comportement et de leur mode de synchronisation constitue une limite importante dans l'état de l'art actuel. De manière plus générale, la synchronisation représente aujourd'hui un verrou scientifique majeur pour assurer la robustesse, l'évolutivité et la résilience des jumeaux numériques dans des environnements industriels dynamiques. Elle doit permettre une adaptation automatique du jumeau à toute dérive du système réel par rapport à son fonctionnement nominal. Des cadres conceptuels comme l'approche 5M [9] offrent une vision structurée de la synchronisation à plusieurs niveaux (données, états, modèles, ressources), et à différentes échelles (multi-systèmes, multi-niveaux, multi-fidélités). Cependant, ces travaux ne s'intéressent pas aux contraintes concrètes liées aux infrastructures de communication : variabilité topologique, dynamique du trafic, latence, goulots d'étranglement, etc. Plus pragmatiquement, il apparaît indispensable de concevoir des mécanismes de synchronisation tenant compte des spécificités techniques des réseaux de communication et des dispositifs embarqués. Cela implique également de développer des indicateurs de qualité de synchronisation (fraîcheur des données, précision, robustesse) ainsi que des mécanismes de certification de la validité du lien entre monde réel et monde numérique — la blockchain pouvant notamment offrir des pistes prometteuses à cet égard [10]. En définitive, cette revue de la littérature met en lumière le manque de cadres formels pour l'élaboration de processus de synchronisation. C'est à cette problématique que répond notre proposition, en s'inscrivant dans le contexte des systèmes de production automatisés et en amorçant une formalisation progressive de la synchronisation par la modélisation comportementale.

3 Méthodologie de synchronisation

Cette section présente une première proposition méthodologique en vue de la génération formelle et automatique de processus de synchronisation entre un jumeau numérique et son système physique. Cette approche est spécifiquement conçue pour les systèmes de production automatisés. Elle repose sur une succession d'étapes modélisées à l'aide d'automates stochastiques et temporisés. Avant d'en exposer le déroulé, nous précisons le cadre et les hypothèses qui sous-tendent notre démarche.

3.1 Hypothèses

La méthode proposée s'applique dans un cadre industriel bien défini, correspondant à des lignes de production dotées de jumeaux numériques représentant des flux produits (de la source vers la sortie du système, en passant par différentes opérations). Elle repose sur les hypothèses suivantes :

- Le jumeau numérique modélise la dynamique des flux physiques dans une ligne de production, à l'exclusion d'autres types de jumeaux (équipements, maintenance, supervision, etc.).
- Le système est supposé suffisamment instrumenté, c'est-à-dire que les éléments observables sont équipés de capteurs offrant une visibilité adéquate pour alimenter les modèles associés aux DT.
- L'impact du réseau est supposé négligeable, tant en termes de latence que de perte de données, ce qui permet de se focaliser sur les mécanismes de synchronisation eux-mêmes.

- La méthode est conçue pour détecter et corriger des écarts limités entre le jumeau et le système physique. Elle n'est pas adaptée, en l'état, à des désynchronisations profondes ou structurelles.

Malgré ces restrictions, cette approche peut s'appliquer à de nombreux cas industriels réels.

3.2 Formalisme de modélisation

Les modèles proposés sont implémentés dans UPPAAL SMC [4], qui étend UPPAAL classique [?] y intégrant des fonctionnalités stochastiques et statistiques. Les automates temporisés stochastiques [4] sont dérivés de la classe des automates temporisés définis par [2] qui partagent des variables et qui sont synchronisés par des événements. Un automate temporisé stochastique est un n -uplet $A = (L, V, E, P, H, T, L_m, l_0, V_0)$, où :

- L est l'ensemble fini des localités ;
- V est un ensemble fini de variables entières ;
- E est l'ensemble fini des événements de synchronisation :
 - $e!$ représente l'envoi d'un événement e
 - $e?$ représente la reception d'un événement e
- P est l'ensemble fini de probabilités :
 - discrètes sur l'ensemble des transitions (une transition permet d'atteindre différentes localités l_i avec une probabilité p_i où $\sum p_i = 1$);
 - continues sur les variables de l'automate (un tirage aléatoire selon une distribution de probabilités permet de déterminer la condition de franchissement d'une transition).
- H est l'ensemble fini des horloges.
- T est l'ensemble fini des transitions $(l, e, g, m, l') \in L \times E \times G \times Maj \times L$ où l et l' sont, respectivement, les localités de départ et d'arrivée, G est l'ensemble des gardes (conditions sur les variables de V) et Maj est l'ensemble des mises à jour des évaluations de variables; e, g et m sont facultatifs (une transition peut ne contenir aucun événement de synchronisation et aucune mise à jour par exemple), mais une transition doit contenir au moins un événement ou une garde ;
- $L_m \subseteq L$ est l'ensemble des localités marquées ;
- $l_0 \in L$ est la localité initiale ;
- $V_0 \in V$ est la valeur initiale des variables de V .

3.3 Démarche de modélisation

La méthodologie proposée s'inscrit dans une chaîne complète de modélisation, de vérification et d'implémentation visant à garantir la validité des mécanismes de synchronisation avant leur intégration dans des environnements industriels.

Dans une première étape, les comportements du système physique et de son jumeau numérique sont formalisés sous la forme d'automates temporisés et stochastiques, puis

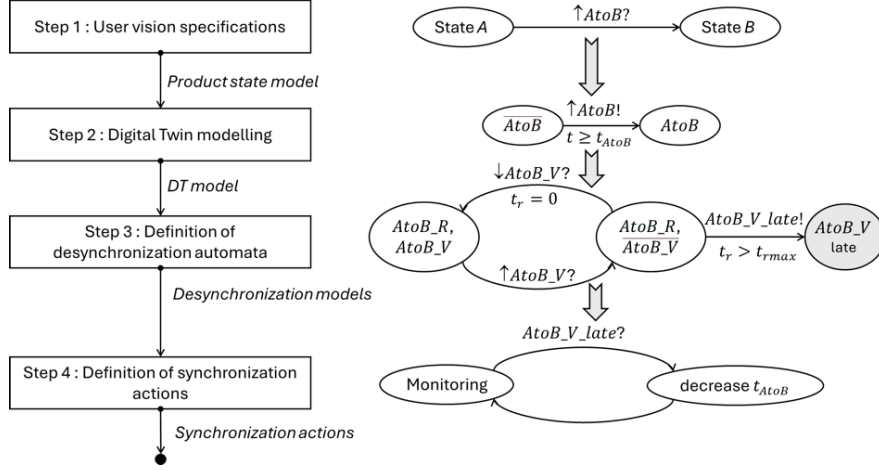


Figure 1: Méthodologie de modélisation du jumeau numérique

implémentés et vérifiés dans l’environnement UPPAAL SMC. Cette phase constitue une étape de validation en amont, destinée à s’assurer que les modèles élaborés permettent effectivement de détecter les désynchronisations temporelles et probabilistes entre le monde réel et le monde virtuel. UPPAAL est ici utilisé pour simuler différents scénarios de fonctionnement, observer la cohérence des transitions et valider la logique de détection des désynchronisation avant tout déploiement.

Dans une seconde étape, les modèles validés sont instanciés dans l’environnement FlexSim, qui permet d’intégrer la logique formelle de synchronisation dans une simulation événementielle 3D représentative d’un système de production réel.

Cet article se concentre exclusivement sur la première étape de cette chaîne. La méthodologie repose sur un workflow en quatre étapes (1), chacune introduisant un modèle structurant la spécification et la surveillance de la synchronisation entre les entités physiques et virtuelles.

• Étape 1 : Modélisation de la vision utilisateur

L’objectif initial est de traduire la manière dont l’utilisateur perçoit le fonctionnement du système. Cette vision est formalisée sous forme d’un modèle d’états produits, qui décrit les transitions successives des produits dans la ligne de production. Ce modèle repose sur un automate dont les places représentent les états possibles du produit, et les transitions traduisent les conditions (principalement des valeurs de capteurs) permettant de passer d’un état à un autre. Dans l’exemple illustratif, l’utilisateur considère deux états, A et B , et observe la transition entre eux via un capteur noté $AtoB$. Le symbole $?$ indique que l’automate attend un événement pour déclencher le franchissement de cette transition et $!$ indique que l’automate émet un événement lors du franchissement de la transition.

• Étape 2 : Modélisation du jumeau numérique

Le jumeau numérique est alors modélisé comme un émulateur du comportement du système réel, capable de fournir les informations nécessaires à l’évaluation du modèle précédent. Le modèle du DT est lui aussi représenté sous forme d’automate, avec des états correspondant à la dynamique des capteurs virtuels, et des transitions simulant leur évolution. Par exemple, le capteur $AtoB$ passe de *faux* à *vrai* après un délai noté t_{AtoB} . Lorsque cette transition est franchie, un message $\uparrow AtoB!$ est envoyé au modèle produit,

qui l'utilise pour actualiser son propre état.

- **Étape 3 : Détection des désynchronisations**

À ce stade, nous introduisons un automate de désynchronisation conçu pour identifier les écarts entre les valeurs issues des capteurs physiques (C_i^R) et celles simulées par le jumeau numérique (C_i^V). Deux types de désynchronisation sont traités :

- une désynchronisation entre le capteur réel et le capteur virtuel. Dans ce cas, la condition de synchronisation repose sur le respect de l'inégalité suivante (eq. 1) :

$$\forall t, \forall i \in [1, n], \|C_i^R(t) - C_i^V(t)\| \leq \epsilon_{sync} \quad (1)$$

- une désynchronisation dues à une incertitude d'évolution suite à la désactivation d'un capteur. Dans ce cas, il faut attendre l'arrivée du capteur suivant (C_{i+1}^R) pour valider l'évolution du modèle du comportement du jumeau virtuel. Dans le modèle du jumeau virtuel, ça sera modélisé par une probabilité PR d'aller dans un état ou un autre

Les variables ϵ_{sync} et PR sont obtenues par apprentissage sur le système réel.

- **Étape 4 : Définition des actions de synchronisation**

La dernière étape de la méthodologie consiste à définir les actions correctives à entreprendre en réponse aux désynchronisations détectées. Trois types d'actions sont identifiés :

- *Mise à jour de l'état actif du produit* : Si, en raison d'une erreur de mesure ou d'une erreur d'estimation de l'état suivant, l'état estimé du produit diffère de son état réel, le modèle d'état du produit est corrigé pour refléter fidèlement la situation physique. Cela correspond à une mise à jour de localité au sein de l'automate du JN.
- *Ajustement des paramètres du modèle* :
 - * Lorsque l'origine de la désynchronisation réside dans un décalage temporel, les paramètres internes du modèle (par exemple t_{AtoB}) sont modifiés pour réduire l'écart entre les transitions virtuelles et réelles. Ainsi, le message $AtoB_V_late$ peut conduire à diminuer t_{AtoB} .
 - * Lorsque l'origine de désynchronisation est due à une mauvaise estimation de l'évolution suivante, il n'est pas possible de changer la valeur de PR à chaque erreur d'estimation de l'évolution suivante. A chaque fois que le modèle va être resynchronisé, un compte va être incrémenté et quand un seuil sera atteint, la probabilité PR sera recalculée en fonction des bonnes et mauvaises estimations.
- *Révision de la structure du modèle* : Si le comportement du système évolue (usure, réorganisation, etc.), la structure même du modèle du jumeau peut devenir obsolète. Une révision du modèle est alors nécessaire. Toutefois, cette action sort du périmètre de la présente contribution et n'est pas encore supportée par la méthodologie.

Cette décomposition méthodologique pose les bases d'un processus de synchronisation structuré, orienté modèle, et adaptable aux systèmes de production automatisés. Les sections suivantes démontrent son application concrète sur un cas réel.

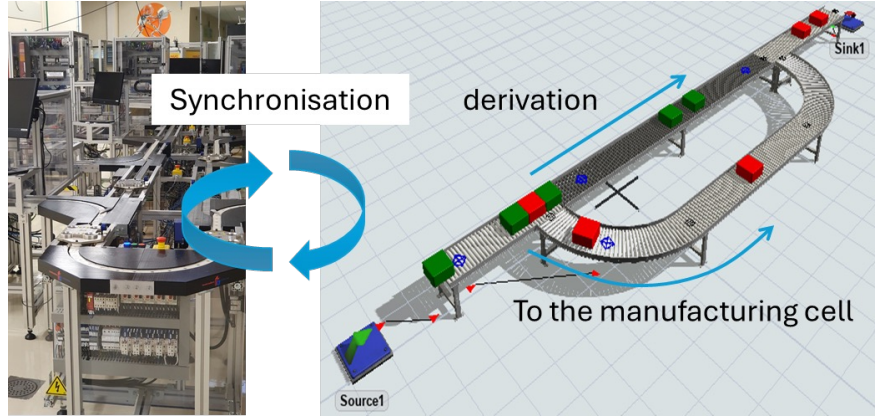


Figure 2: Description du cas d'étude et de ces jumeaux numériques

4 Application

La méthodologie proposée est appliquée à un cas d'étude issu de la plateforme technologique S.Mart de l'Atelier Inter-établissements de Productique Lorrain (AIP Lorrain). L'expérimentation porte sur une portion d'un système de production flexible composé de plusieurs cellules de fabrication interconnectées par des convoyeurs (2). Le système permet aux produits de circuler entre cellules en fonction de leur gamme de fabrication, et de rejoindre soit une cellule de traitement, soit une ligne de dérivation. Les cellules ne sont pas représentées dans le schéma de la 2 pour des raisons de clarté graphique, mais elles sont prises en compte dans l'analyse fonctionnelle.

4.1 Description du système physique

Chaque cellule est instrumentée à l'aide d'un ensemble identique de capteurs positionnés aux points de contrôle des convoyeurs (3). Les points A à E désignent des localisations critiques pour le suivi des produits au sein du système, notamment dans les zones d'identification, de dérivation et d'entrée/sortie de cellule.

Les modèles présentés dans les étapes suivantes sont décrits sous le logiciel UPAAL¹.

Ils utilisent les variables de la figure 4 qui sont définies de manière globale à tous les modèles, selon la même convention de nommage : la première partie *fn*, *fd* indique s'il s'agit d'un front montant ou descendant, la seconde détaille le nom du capteurs considéré et la dernière *R*, *V* indique s'il s'agit du capteur réel ou de sa représentation virtuelle au sein du jumeau numérique.

4.2 Modélisation de la dynamique produit (étape 1)

La première étape consiste à formaliser le modèle d'état du produit, tel que perçu par l'utilisateur. Ce modèle traduit la logique de circulation du produit au travers des zones clés du système, selon les signaux émis par les capteurs (figure 3). Un produit est détecté à son arrivée devant la cellule lorsque $DSPal == 1$, indiquant son entrée dans une zone d'identification. Il y demeure durant une période estimée à t_1 . Lorsque $DSPal == 0$, il quitte cette zone. Deux cas se présentent alors :

¹<https://uppaal.org>

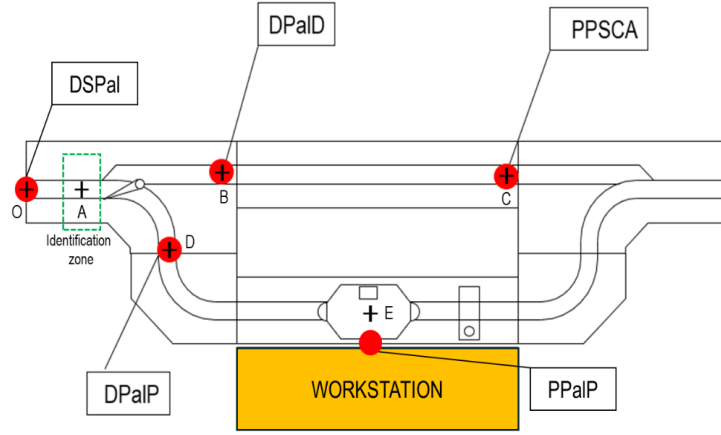


Figure 3: Position des capteurs sur le système manufacturier

```
// Place global declarations here.

broadcast chan fm_DSPal_R, fd_DSPal_R, fm_PPald_R, fd_PPald_R, fm_PPSCA_R, fd_PPSCA_R,
broadcast chan fm_PPalp_R, fd_PPalp_R, fm_DPald_R, fd_DPald_R, fm_DPald_R, fd_DPald_R;
broadcast chan fm_DSPal_V, fd_DSPal_V, fm_PPald_V, fd_PPald_V, fm_PPSCA_V, fd_PPSCA_V,
broadcast chan fm_PPalp_V, fd_PPalp_V, fm_DPald_V, fd_DPald_V, fm_DPald_V, fd_DPald_V;
broadcast chan fm_DSPal_lateV, fd_DSPal_lateV, fm_DSPal_lateR, fd_DSPal_lateR;

int t1, t2, t3, t4, t5, t6, t7, tmax, tvmax;

int pr=80;

clock t_DT_Cell, t_Desynchro_v, t_Desynchro_r;
```

Figure 4: Declaration des variables des modèles

- Si $DPald == 1$, le produit est dirigé vers la dérivation et est considéré hors de la cellule lorsque $PPSCA == 1$.
- Si $DPalp == 1$, il est acheminé vers la cellule de traitement. L'état $PPalp == 1$ signifie qu'il y est en cours de traitement. Une fois le traitement terminé ($PPalp == 0$), le produit est envoyé vers une autre cellule.

Ce modèle reflète les attentes de l'utilisateur en matière de granularité des états suivis, et sert de référence au processus de synchronisation.

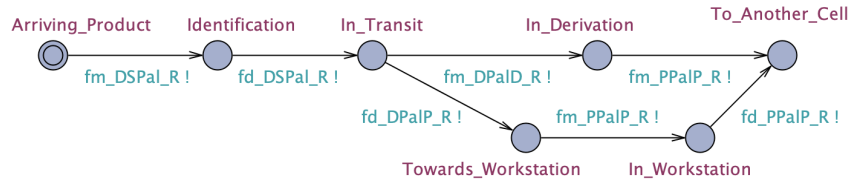


Figure 5: Modèle du système réel

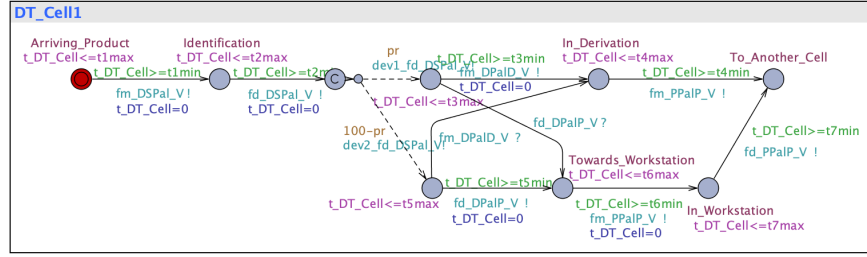


Figure 6: Modèle du jumeau numérique

4.3 Modélisation du jumeau numérique (étape 2)

Le modèle du jumeau numérique est conçu comme un émulateur du système réel, capable de reproduire les évolutions des capteurs nécessaires à la mise à jour du modèle produit. Il est formalisé sous la forme d'un automate intégrant à la fois des transitions temporisées et stochastiques (figure 6).

En fonction du modèle du comportement réel de la figure 3, il y a 2 évolutions possibles :

- déterministe, c'est à dire qu'un seul capteur peut évoluer par la suite. Par exemple, après le front montant de *DPalD_R*, il peut y avoir que le front montant de *DPalP_R*
- indéterministe, c'est à dire que plusieurs capteurs peuvent évoluer par la suite. Par exemple, après le front descendant de *DPalD_R*, il peut y avoir que deux capteurs qui peuvent être activés, les capteurs *DPalD_R* ou *PPalP_R*.

Dans le modèle de la figure 6, les évolutions des capteurs sont modélisées :

- pour les évolutions déterministe, par des transitions déclenchées lorsque des conditions temporelles sont satisfaites. Par exemple, la transition de l'état "Arrivée produit" à "Zone d'identification" ne peut se produire que si $t \geq t1$, avec $t1 = d(OA)/v(OA)$, où $d(OA)$ est la distance entre les points O et A, et $v(OA)$ la vitesse du convoyeur.
- pour les évolutions indéterministe, comme une bifurcation probabiliste : il est redirigé vers la dérivation avec une probabilité pr , et vers la cellule avec une probabilité $100 - pr$. Des messages de type *fm_DSPal* ou *fd_DSPal* indiquent les changements d'état simulés des capteurs.

4.4 Modélisation de l'observateur de désynchronisation (étape 3) et Modélisation des actions de synchronisation (étape 4)

Une fois les modèles établis, la détection des écarts entre le monde réel et sa représentation numérique s'appuie sur des automates de désynchronisation. Deux désynchronisations sont à détecter : (i) désynchronisation pour un même capteur entre le réel et le virtuel et (ii) désynchronisation due à un choix probabiliste.

4.4.1 Désynchronisation temporelle entre réel et virtuel

Afin de caractériser la désynchronisation temporelle entre une entité physique et son jumeau numérique, nous avons développé un modèle d'automate temporisé nommé *Desynchro-vitesse*

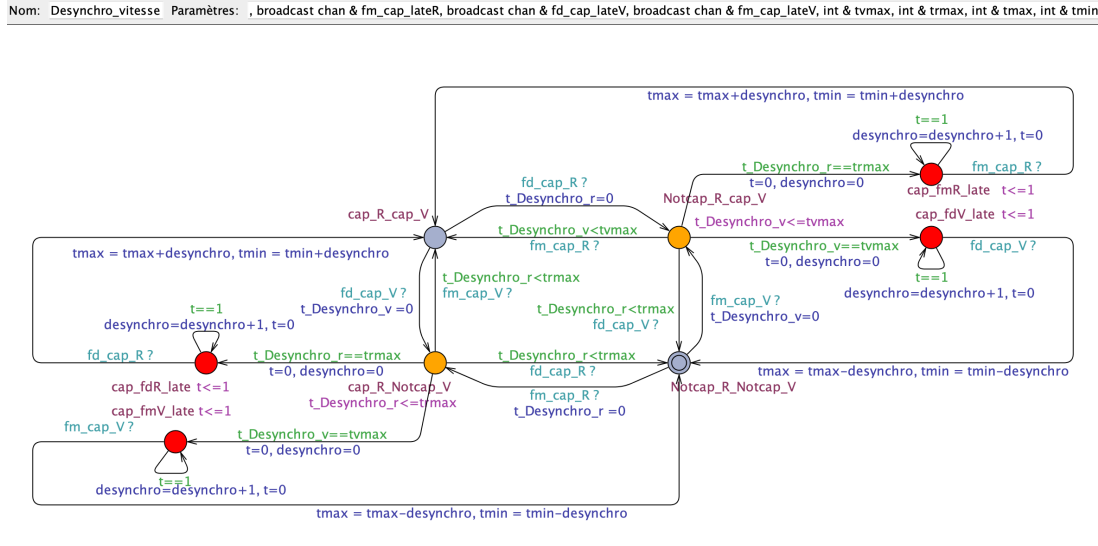


Figure 7: Modèle pour observer la désynchronisation entre réel et le virtuel

(figure 7). Ce modèle permet de formaliser les relations temporelles entre deux flux : un flux réel (R), issu de capteurs physiques ou d'événements terrain, et un flux virtuel (V), émis ou prédit par un modèle numérique. Le comportement de l'automate est régi par plusieurs horloges (notamment $t_Desynchro_r$ et $t_Desynchro_v$) qui mesurent les délais entre les événements réels et virtuels, et un compteur *desynchro* qui enregistre l'accumulation des décalages. Les transitions sont conditionnées par des gardes temporelles ($t_Desynchro_r < trmax$, $t_Desynchro_v == tvmax$) et la réception de signaux via des canaux de synchronisation ($fm_cap_R?$, $fd_cap_V?$,...).

Le modèle comporte plusieurs états intermédiaires permettant de catégoriser les situations selon la synchronisation observée :

- des états bleus ($Notcap_R_Notcap_V$) traduisent l'absence de réception de l'un ou l'autre des événements dans une fenêtre temporelle acceptable,
- des états orange ($t_Desynchro_r == trmax$) marquent une désynchronisation limite, où un événement est sur le point de dépasser le seuil admissible,
- des états rouges (cap_fdV_late , cap_fmR_late) indiquent une désynchronisation excessive, c'est-à-dire un retard avéré du jumeau virtuel par rapport au réel (ou inversement), avec un temps d'attente maximal dépassé ($t \leq 1$).

Chaque transition entre états est déclenchée par un événement (réception d'un signal R ou V) et accompagnée d'actions sur les variables internes ($desynchro++$, $t=0$, $desynchro=0$). Lorsque la désynchronisation devient trop fréquente, le modèle ajuste dynamiquement les bornes temporelles $tmin$ et $tmax$ (ex. $tmax = tmax + desynchro$) afin de réagir au comportement observé et maintenir la robustesse du couplage entre les mondes réel et virtuel.

Ce modèle offre ainsi un cadre formel pour détecter, catégoriser et réagir aux écarts temporels entre les entités physiques et numériques.

4.4.2 Désynchronisation lors d'un choix probabiliste

Afin d'analyser l'impact d'un choix probabiliste sur la désynchronisation entre un système réel et son jumeau numérique, nous avons développé un modèle d'automate nommé *Desynchro-voies* (figure 8). Ce modèle représente une situation où un système peut suivre deux voies alternatives (*dev1* ou *dev2*), choisies en fonction de règles ou d'un comportement probabiliste. Le modèle permet de comparer, à chaque occurrence de bifurcation, le choix effectué dans le monde réel (détecté via les canaux *fm_DPalD_R?*, *fd_DPalP_R?*) et celui réalisé dans le modèle virtuel (émis via *fm_DPalD_V!*, *fd_DPalP_V!*). L'objectif est de quantifier la cohérence entre les deux décisions et d'identifier les cas de désynchronisation comportementale.

Le modèle est structuré en deux branches horizontales, correspondant à l'observation des deux voies :

- Dans la branche inférieure, si le modèle virtuel choisit la voie *dev2* (*dev2_V?*), l'automate vérifie que le système réel a bien suivi la même voie (signal *fm_DPalD_R?*). Si c'est le cas, le compteur *cpt_dev2_bon* est incrémenté. En cas de divergence, c'est *cpt_dev2_mauvais* qui est incrémenté.
- Symétriquement, si le jumeau numérique emprunte la voie *dev1* (*dev1_V?*), le modèle attend la confirmation du monde réel via *fm_DPalD_R?*. Selon que les décisions concordent ou non, il incrémente *cpt_dev1_bon* ou *cpt_dev1_mauvais*.

Ces deux parties permettent ainsi de détecter localement, à chaque instant, si une désynchronisation décisionnelle s'est produite. L'automate fonctionne ici comme un observateur, sans influencer les décisions mais en suivant les effets.

La composante verticale du modèle correspond à la synchronisation des prises de décision probabiliste dans le jumeau numérique. Lorsque le nombre total de décisions divergentes (*cpt_dev1_mauvais* + *cpt_dev2_mauvais*) atteint un seuil défini (ici 30), l'automate effectue une mise à jour de l'indicateur probabiliste *pr*, calculé comme le rapport entre les décisions correctes observées sur la voie 1 et le total des décisions correctes ($pr = \frac{cpt_dev1_bon}{cpt_dev1_bon + cpt_dev2_bon}$). Cette valeur permet d'estimer la préférence ou le biais du jumeau numérique vers l'une des deux voies, comparée aux décisions réelles.

Enfin, un état de réinitialisation défini comme un état committed permet de remettre à zéro tous les compteurs après chaque calcul de *pr*, de façon à isoler les phases d'observation et de traitement. Le système peut ainsi fonctionner de manière cyclique, enchaînant des périodes de collecte de données comportementales et des phases d'analyse statistique.

Ce modèle offre une base pour détecter et quantifier les désynchronisations induites par des divergences de choix entre un système réel et son jumeau numérique.

Cette étude de cas permet d'illustrer l'applicabilité de la démarche sur un système réel, et de valider l'enchaînement des étapes de modélisation, de surveillance et de correction de la synchronisation. L'exemple présenté constitue une première expérimentation concrète, à partir de laquelle pourront être testées des extensions de la méthodologie vers des environnements plus complexes.

5 Conclusion et perspectives

Cet article a introduit une méthodologie formelle de conception de processus de synchronisation entre jumeaux numériques et systèmes physiques, fondée sur l'utilisation d'automates temporisés et stochastiques. L'ensemble des étapes de la démarche a été présenté de manière

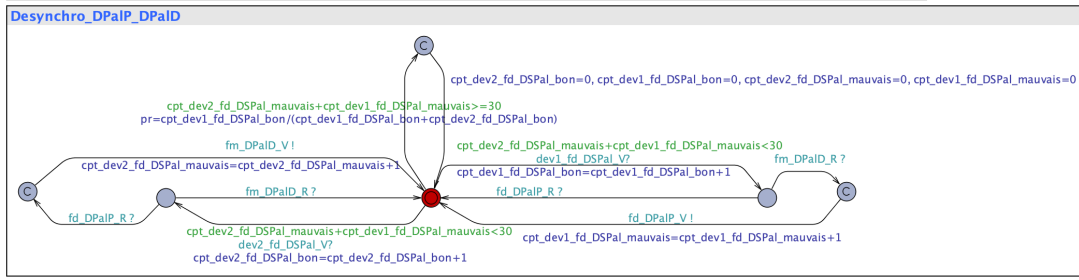


Figure 8: Modèle pour observer la désynchronisation d'un choix probabiliste

structurée, mettant en évidence les liens logiques entre les modèles successivement élaborés : vision utilisateur, modélisation du jumeau, détection de désynchronisations et actions correctives. L'étude de cas appliquée à un système de production flexible issu de la plateforme S.Mart a permis d'illustrer la mise en œuvre concrète de la méthode et d'en valider la cohérence conceptuelle. Elle constitue une première expérimentation utile pour identifier les forces et les limites du formalisme proposé. Les travaux futurs viseront à renforcer la validité opérationnelle de la méthode par son implémentation dans un environnement logiciel dédié aux jumeaux numériques, tel que FlexSim. Cette étape sera déterminante pour évaluer la faisabilité pratique, la robustesse des mécanismes de synchronisation proposés et leur intégration dans des environnements industriels réels. Au-delà de cette mise en œuvre, ce travail met en évidence la complexité intrinsèque des processus de synchronisation lorsqu'ils doivent être à la fois formels, automatisés et adaptatifs. Il ouvre ainsi la voie à de nombreux travaux complémentaires, notamment :

- l'extension du formalisme à d'autres types de systèmes ou d'usages (maintenance, logistique, pilotage énergétique) ;
- la prise en compte explicite des contraintes réseaux (latence, pertes, sécurité) dans les modèles ;
- l'exploration de mécanismes d'apprentissage adaptatif pour affiner les paramètres de synchronisation en contexte dynamique.

Cette première contribution pose les bases d'une approche rigoureuse de la synchronisation dans les systèmes cyber-physiques, et appelle à la constitution de passerelles plus larges entre ingénierie des modèles, simulation distribuée et pilotage intelligent.

References

- [1] Wael Alghamdi and Emad Albassam. Synchronization patterns for digital twin systems. *Journal of Applied Data Sciences*, 5(3):1026–1037, 2024.
- [2] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.
- [3] Jiayuan Chen, Changyan Yi, Samuel D Okegbile, Jun Cai, and Xuemin Shen. Networking architecture and key supporting technologies for human digital twin in personalized healthcare: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 26(1):706–746, 2023.
- [4] Alexandre David, Kim G Larsen, Axel Legay, Marius Mikućionis, and Danny Bøgsted Poulsen. Uppaal smc tutorial. *International journal on software tools for technology transfer*, 17(4):397–415, 2015.

- [5] Mustafa Ersan and Erdal Irmak. Development and integration of a digital twin model for a real hydroelectric power plant. *Sensors*, 24(13):4174, 2024.
- [6] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.
- [7] Reiner Jedermann, Kunal Singh, Walter Lang, and Pramod Mahajan. Digital twin concepts for linking live sensor data with real-time models. *Journal of Sensors and Sensor Systems*, 12(1):111–121, 2023.
- [8] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihm. Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline*, 51(11):1016–1022, 2018.
- [9] Weiran Liu, Jiangfeng Cheng, Zhiwen Wen, Xiaofu Zou, Zhaozong Wang, Hongting Liu, and Fei Tao. A 5m synchronization mechanism for digital twin shop-floor. *Chinese Journal of Mechanical Engineering*, 36(1):136, 2023.
- [10] Sabah Suhail, Rasheed Hussain, Raja Jurdak, and Choong Seon Hong. Trustworthy digital twins in the industrial internet of things with blockchain. *IEEE Internet Computing*, 26(3):58–67, 2021.
- [11] Bernhard Wallner, Benedikt Zwölfer, Thomas Trautner, and Friedrich Bleicher. Digital twin development and operation of a flexible manufacturing cell using iso 23247. *Procedia CIRP*, 120:1149–1154, 2023.