Nancy-Université

Université Henri Poincaré

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

➢ Contact SCD Nancy 1 : theses.sciences@scd.uhp-nancy.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Nancy-Université
Université Henri Poincaré
FACULTE DES SCIENCES & TECHNIQUES

U.F.R. : Sciences et Techniques Mathématiques, Informatique et Automatique
Ecole Doctorale : IAEM Lorraine
Département de Formation Doctorale : Automatique

CENTRE DE RECHERCHE
EN AUTOMATIQUE DE
NANCY

CRAN
CNRS UMR 7039

Thèse

présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy 1

en Sciences, spécialité Automatique,
Traitement du Signal et Génie Informatique

par **Andrés VÉJAR**

Génération de modèles de simulation adaptatifs, pilotée par les trajectoires produits

Soutenue publiquement le 14 novembre 2011

Membres du jury :

Rapporteurs :	Professeur Pierre Castagna	Université de Nantes, IRCCyN
	Professeur Georges Habchi	Université de Savoie, SYMME
Examinateurs :	DR CNRS Jean-Claude Hennet	LSIS, Marseille
	Professeur Gérard Morel	Université Henri Poincaré Nancy 1, CRAN
	MdC HdR Thierry Bastogne	Université Henri Poincaré Nancy 1, CRAN
Directeur de thèse :	Professeur Patrick Charpentier	Université Henri Poincaré Nancy 1, CRAN

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences, B.P. 70239, 54506 Vandœuvre lès Nancy

Table des matières

Introduction générale **ix**

1 Données & Modèle **1**

1.1 Les données source d'observation de l'espace technique 1

1.1.1 Les données, objectivité et fonction cognitive 1

1.1.2 Récolte et usage des données 3

1.1.3 Les données spatio-temporelles 7

1.2 Modélisation de phénomènes physiques 10

1.2.1 Des données au modèle 10

1.2.2 La reconstruction théorique et phénoménologique 13

1.3 Données et Modèles : lien avec notre problématique 15

1.3.1 Le Produit, vecteur d'informations 15

1.3.2 Modélisation adaptative de systèmes manufacturiers par le biais des trajectoires-produit 16

1.4 Conclusions 19

2 Positionnement **21**

2.1 La simulation 22

2.1.1 Niveau de spécificité 22

2.1.2 Statut de la simulation : expérience, outil ou interface ? 23

2.1.3 Intérêt de la simulation 24

2.1.4 Usages de la simulation dans les systèmes manufacturiers 26

2.1.5 Déploiement de la simulation 28

2.1.6 Outils supports pour le développement d'un modèle de simulation 33

2.2 Modélisation avec trajectoires 36

2.2.1 Introduction . . . 36
2.2.2 Utilisation dans le vivant et l'artificiel . . . 36
2.2.3 Utilisation dans le domaine de la production et logistique . . . 39
2.2.4 Extraction de connaissances des trajectoires . . . 40
2.3 Conclusions du chapitre . . . 42

3 Proposition **43**
3.1 Introduction et présentation de la méthodologie proposée . . . 43
3.2 Du produit à la trajectoire dans le contexte des systémes dynamiques à événements discrets . . . 47
3.2.1 Système manufacturier et SDED . . . 47
3.2.2 Le produit comme composition-décomposition des opérations . . . 48
3.3 Éléments structurels du modèle . . . 50
3.3.1 Réseau de transport . . . 50
3.3.2 Noeud buffer-opération . . . 52
3.4 Éléments structurels du modèle : trajectoire produit et paramétrage . . . 52
3.5 Auto-adaptation du modèle . . . 58
3.6 Conclusion : modèle formel du générateur . . . 61

4 Mise en œuvre système générateur **63**
4.1 Introduction . . . 64
4.2 Diagrammes de flux . . . 65
4.3 Structures de données et algorithmes . . . 68
4.4 Conclusions du chapitre . . . 70

5 Expérimentations et validations **73**
5.1 Introduction . . . 73
5.2 Production de données *en silico* . . . 74
5.2.1 Présentation du système d'intérêt . . . 74
5.2.2 Détermination et vérification de la nature des données générées . . . 78
5.3 Scénarios de validation . . . 81
5.4 Résultats et analyse . . . 82
5.4.1 Régime nominal . . . 82
5.4.2 Cas du système perturbé . . . 88

5.5 Conclusion 91

Conclusions et perspectives **93**

Annexes **97**

A Résultats génération par scénarios 98

A.1 Scénario 1 98

A.2 Scénario 2 100

A.3 Scénario 3 107

B Génération de données 109

Liste des publications **115**

Bibliographie **117**

Table des figures

1 Le générateur dans son environnement xi
2 Schéma synthétique des travaux xii

1.1 Stockage et supports 5
1.2 Évolution du traitement des données 5
1.3 Echelles d'observation et spatialisation 9
1.4 Problèmes quantitatifs et abstraction qualitative 11
1.5 Problème de reconstruction 13
1.6 Reconstruction et Qualitative & Quantitative 14
1.7 Épistémologie formelle et appliquée 14
1.8 Produit 16
1.9 Le produit comme vecteur d'informations. 17
1.10 Application localisation 17
1.11 Flux de données de géolocalisation 18

2.1 Simulation par ordinateur : niveaux de spécificité. 23
2.2 Simulation et théorie : Paradigme génératif. 25
2.3 Modélisation de simulations 29
2.4 Processus de simulation en manufacture 32
2.5 Outils Supports 34
2.6 DDDAS 35
2.7 Calcul non conventionnel, Physarum 37
2.8 Le visiteur : Yarbus 38
2.9 Calcul spatial 41

3.1 Cheminement d'applications possibles. 44

3.2 Principe théorique générateur de modèle de simulation 47
3.3 Exemple du graphe orienté d'un produit P 54
3.4 Vitesses composante a 55
3.5 Noeud buffer-opération. File d'attente et serveur. 56
3.6 Temps dans un point $v = 0$ avec congestion 56
3.7 Schéma de contrôle adaptatif direct 59
3.8 Schéma de contrôle adaptatif indirect 59
3.9 Boucle de rétroactions dans le système manufacturier. 61

4.1 Synoptique général du prototype réalisé. 64
4.2 Construction progressive du modèle. 66
4.3 Adaptation. Mise à jour des données du modèle. 67
4.4 Adaptation. Fonction de validité. 68

5.1 Machines d'états pour la simulation «système - réel» 75
5.2 Diagramme de classes UML pour le moteur de simulation. 77
5.3 Temps entre événements global T_{ee}. Histogramme. 79
5.4 Graphe de récurrence du temps entre événements 80
5.5 Système manufacturier, agrégation des données spatiotemporelles 40 machines, z-axis représente le temps. 80
5.6 Scénario 1 : Histogrammes des temps 86
5.7 Scénario 2 : Histogrammes des temps 87
5.8 Scénario «adaptation du modèle». 91

9 Histogrammes temps d'inter-arrivées, Scénario 1 98
10 Histogrammes temps de service, Scénario 1 99
11 Histogrammes temps d'inter-arrivées, Scénario 3 107
12 Histogrammes temps de service, Scénario 3 108

Liste des tableaux

1.1 Épistémologie transcendantale et programmation orientée objets 6

2.1 Classification sur des «statut» de *simulation par ordinateur* selon Varenne 24

4.1 Structures de données m et p 69

5.1 Conditions expérimentales 81
5.2 Cas 1 81
5.3 Cas 2 81
5.4 Temps de calcul de référence scénario 1 82
5.5 Résultats scénario 1, produit 0 83
5.6 Résultats scénario 1, produit 1 83
5.7 Résultats scénario 1, produit 2 84
5.8 Temps de calcul de référence scénario 2 85
5.9 Résultats scénario 2, produit 0 88
5.10 Résultats scénario adaptatif, produit 0 89
5.11 Résultats scénario adaptatif, produit 1 89
5.12 Résultats scénario adaptatif, produit 2 90
5.13 Résultats scénario adaptatif, produit 3 90

14 Paramètres Scénario 1 98
15 Paramètres Scénario 2 100
16 Résultats Scénario 2, produit 0 100
17 Résultats Scénario 2, produit 1 101
18 Résultats Scénario 2, produit 2 101
19 Résultats Scénario 2, produit 3 102

20 Résultats Scénario 2, produit 4 102
21 Résultats Scénario 2, produit 5 103
22 Résultats Scénario 2, produit 6 103
23 Résultats Scénario 2, produit 7 104
24 Résultats Scénario 2, produit 8 104
25 Résultats Scénario 2, produit 9 105
26 Résultats Scénario 2, produit 10 105
27 Résultats Scénario 2, produit 11 106
28 Résultats Scénario 2, produit 12 106

Introduction générale

La modélisation et l'identification de systèmes est un domaine de recherche important de l'automatique. L'idée y est d'obtenir un modèle mathématique d'un système à partir de mesures et d'en régler les paramètres. La structure du modèle nécessite en général une connaissance *a priori* du système à modéliser. L'application d'algorithmes spécifiques d'identification, suite à des essais expérimentaux, permet ensuite le réglage des paramètres du modèle. Quoiqu'il en soit, le modèle obtenu est une forme de simplification du système modélisé et ne lui est donc pas identique. Il possède cependant des propriétés similaires à celui-ci, ce qui en constitue l'intérêt majeur pour l'utilisateur potentiel.

Les travaux de recherche présentés dans cette thèse s'inscrivent dans cette problématique de modélisation et d'identification des systèmes. La différence notoire est liée aux systèmes d'intérêts qui sont les nôtres, puisque nous cherchons à identifier et modéliser des systèmes manufacturiers, avec comme objectif de fournir un modèle de simulation de flux, indépendamment de l'usage qui pourra en être fait. Ce type de système est en général très complexe de par la multitude d'opérations élémentaires qu'il met en œuvre, de par la diversité des produits qui peuvent y être élaborés, de par les décisions qui peuvent y être prises, et ce, à des niveaux hiérarchiques différents,... mais également par l'ensemble des interactions possibles entre tous ces facteurs. Une connaissance *a priori* très marquée sur ces systèmes est donc bien souvent nécessaire pour l'élaboration de modèles logico-mathématique de ceux-ci. Cette phase de modélisation nécessite la plupart du temps des "experts-systèmes" et ce pendant des durées longues voire très longues. Le coût d'obtention de ces modèles s'en ressent fortement.

Une alternative à cette approche est liée à l'utilisation de machines d'apprentissage que celles-ci soient supervisées ou non. De façon générale, l'usage de ces machines ne permet pas d'obtenir une structure du modèle représentative du système étudié : cela en limite donc l'intérêt pour un usage ultérieur où des tests de modification structurelle devraient, par exemple, être réalisés (analyse de type

"what if ?"). Par contre, et contrairement aux modèles construits par des experts, il est possible d'utiliser ces modèles en ligne (au sens où ils restent connectés au système réel, à ses entrées et sorties). Cela présente l'avantage d'avoir un modèle représentatif quelque soit le moment où il est utilisé.

Le défi posé ici peut donc se résumer ainsi : Comment obtenir un modèle de simulation de flux rapidement et à moindre coût, représentatif du système et de son fonctionnement à tout instant, tout en conservant la possibilité d'en modifier la structure et/ou les paramètres afin de pouvoir réaliser des tests et/ou valider des hypothèses de travail ?

Un des éléments clés nous ayant permis de nous lancer dans ce défi, il y a maintenant quelques années, est sans aucun doute lié aux évolutions technologiques en lien avec les STIC [1]. Sans les progrès technologiques considérables réalisés sur les capacités à communiquer, capter, et mémoriser, de différents composants des systèmes, l'idée même de ces travaux n'aurait pu naître. La possibilité d'embarquer des composants dotés de ces capacités sur des produits de toute nature est maintenant une réalité. Ils permettent d'imaginer des produits intelligents [95, 3, 113, 2], actifs [60, 77],... dans leur environnement. Le CRAN [2] a été l'un des laboratoires pionniers sur ces voies de recherche. Les produits peuvent par exemple être dotés d'une instrumentation propre (MEMS, GPS, Tag RFID, etc [3]). Les échanges d'informations entre le produit et son environnement peuvent s'effectuer :

- À certains endroits de synchronisation : emplacements de capteurs R/W - technologies RFID, code-barre, etc.
- En tout lieu : réseaux sans fil de type Wifi, Zigbee, Bluetooth, etc.

Ces technologies, et en particulier les technologies de communication, peuvent contribuer à la localisation d'un produit dans son environnement. Cette information de localisation d'objets est centrale dans les travaux présentés dans cette thèse. Elle est devenue une information cible importante dans de nombreux domaines, comme par exemple en logistique pour la traçabilité des produits, l'inventaire des stocks, l'optimisation d'une flotte de transport, etc. Elle est, encore à ce jour, une information peu utilisée dans le monde manufacturier. La localisation des produits, identifiables et identifiés, dans le temps (données spatio-temporelles) constituera pour nous le type de données d'entrée nécessaires et suffisantes pour la modélisation et l'identification du système dans lequel les produits circulent. Cela classe notre proposition dans les approches de type « Data Driven Modeling ». Dans le cadre de cette thèse, et comme hypothèse forte de travail, nous considérerons qu'il est possible d'ob-

1. STIC : Sciences et Technologies de l'information et de la Communication
2. CRAN : Centre de Recherche en Automatique de Nancy
3. MEMS : microsystème électromécanique, GPS : système de positionnement mondial, Tag RFID : puces électroniques avec un système de radio-identification.

tenir, en tout temps et en tout lieu du système d'intérêt, les positions successives de tous les produits (ou composants) y circulant. C'est le point d'entrée de nos travaux, qui s'affranchissent des problématiques technologiques de localisation, fruit de travaux de recherche d'autres équipes du CRAN [4].

Notre thèse vise donc à affirmer qu'il est possible de construire, de générer, de manière automatique un modèle de simulation de flux dédié aux processus manufacturiers, sur la base d'une collecte temps réel des données de localisation des produits dans le temps. Ce processus génératif est matérialisé par un ensemble d'opérations réunies sous le terme de «générateur» de modèle de simulation. Notre thèses suppose la préexistence du système physique d'intérêt, et ne peut donc bien sûr être utilisée pour la conception d'un nouveau système. La nature des données collectées, et des informations pouvant en être extraites, nous imposera un type de modèle particulier basé sur les réseaux de files d'attente. C'est la seule connaissance *a priori* qui sera introduite dans notre modèle, à savoir sa nature. Cette forme de représentation possède l'avantage d'être par la suite modifiable pour des usages divers du modèle en simulation. Outre le gain de temps attendu par l'automatisation de la construction du modèle, un des autres principaux avantages réside sans aucun doute dans son évolutivité. Toute modification sur le système réel ayant des conséquences sur les trajectoires spatio-temporelles observées pourra en effet être perçue et donc prise en compte par le générateur, d'où un ajustement du modèle. Ces notions simples de trajectoires sont dans les faits relativement riches. En effet, et nous tenterons de le montrer dans nos travaux, elles permettent de mettre à jour ou découvrir la structure du système de production (la disposition des serveurs [5]), mais également les gammes des produits (et un certain nombre de durées pouvant y être rattachées, e.g. temps de service, temps d'attente) et leur nomenclature (composition et décomposition). La Figure 1 positionne le générateur, objet au centre de nos travaux, dans un cadre générique d'usage.

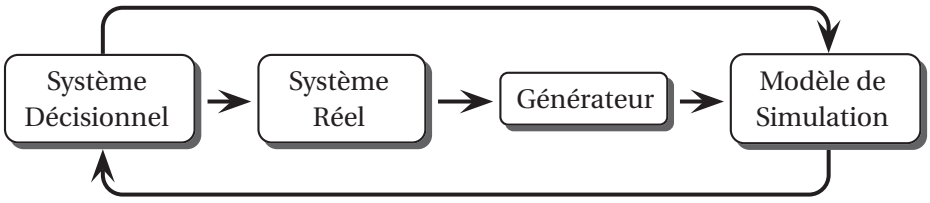


FIGURE 1 – Le générateur dans son environnement

L'approche choisie, et le lien direct avec le monde réel via la collecte de données, est assez proche dans l'idée de celles en usage dans le monde de l'automatique classique. Elle nous permettra, au

4. Projet Système Réseau
5. Le layout d'une usine

même titre que la problématique de modélisation - identification abordée ici, de tisser quelques liens entre deux mondes très proches et cependant très éloignés ne serait ce que par les méthodes et modèles utilisés.

Afin de présenter la suite logique du contenu de ce mémoire, un schéma synthétique est proposé à la Figure 2.

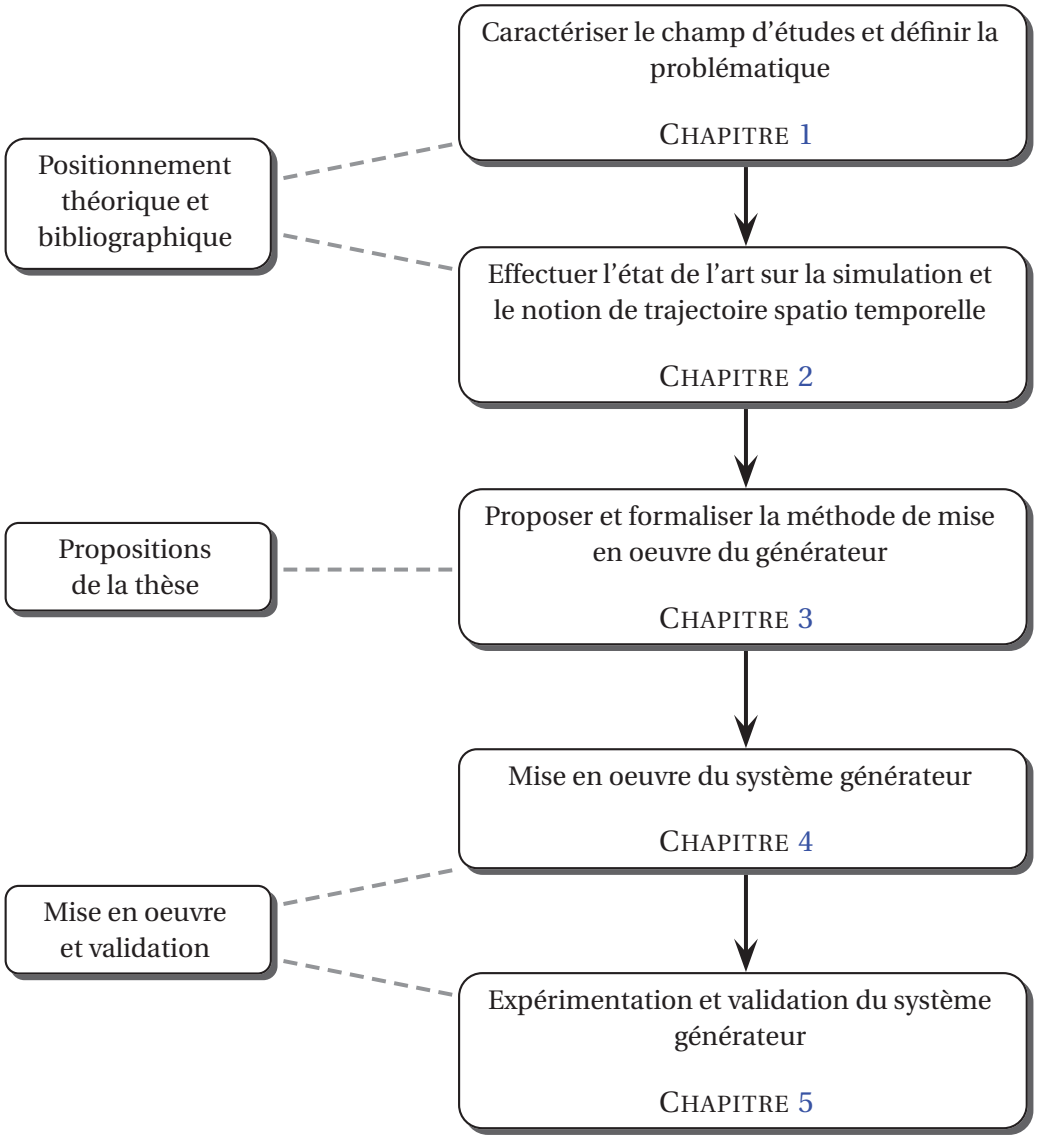


FIGURE 2 – Schéma synthétique des travaux présentés dans ce mémoire.

L'objet du Chapitre 1 est de définir le champ scientifique global dans lequel nos travaux de recherche sont inscrits. Ainsi, et dans un premier temps, un éclairage général sur les notions de données et d'information est réalisé. Cet éclairage se base entre autre sur le sens que l'on peut accorder

à ces données lors de leur usage après récolte. Un premier zoom sur des données de nature spatio-temporelle est ensuite proposé. Le processus théorique de modélisation permettant la transformation de données collectées en une représentation des phénomènes physiques auxquelles elles sont liées est ensuite présenté. Enfin, une première représentation de la problématique est développée afin de préciser la notion de modélisation adaptative de systèmes manufacturiers par le biais des trajectoires-produit dans le contexte de cette thèse.

Le Chapitre 2 s'intéresse, par le biais d'une analyse bibliographique, à deux types d'objets différents très en lien avec nos travaux, à savoir (1) la simulation et (2) les cas d'utilisation de la modélisation par le biais de trajectoires. Le concept de simulation, ses spécificités, son statut et ses intérêts sont ainsi dans un premier temps développés. Une analyse bibliographique des approches méthodologiques de déploiement de la simulation, et des outils méthodes de modélisation y sont également présentés et analysés. La seconde partie de ce chapitre bibliographique se focalise pour sa part sur les notions de trajectoires spatio-temporelles et les domaines applicatifs dans lesquels celles-ci sont utilisées. Les types de connaissance pouvant être extraites des trajectoires font l'objet de la dernière partie de ce chapitre.

Le Chapitre 3 constitue le cœur des propositions de cette thèse. Après une description générale des problèmes posés et de la méthodologie déployée, chacune des étapes de celle-ci est détaillée. Ainsi, et dans un premier temps, une définition formelle du "produit" intégrant son évolution temporelle (la gamme), mais également sa composition (la nomenclature), est proposée. La forme analytique de cette définition peut également être représentée sous la forme d'un graphe orienté. Les éléments structurels du modèle à construire sont ensuite présentés. Les notions de "buffer-opération" (point de la trajectoire du produit de vitesse nulle) et de "réseau de transport" (agrégation des trajectoires produit), éléments centraux du modèle à générer, sont détaillées. Les modalités de calcul des paramètres de ces éléments, par le seul biais des informations issues des flux de localisation, sont ensuite proposées. Enfin un paragraphe spécifique précise les principes d'auto-adaptation, de façon générale, puis dans le cadre particulier de cette thèse.

Le Chapitre 4 présente la mise en œuvre du générateur de façon plus opérationnelle. L'architecture globale du prototype informatique est dans un premier temps proposée. Les diagrammes de flux des modules développés sont ensuite progressivement proposés pour décrire successivement comment le modèle du système est établi puis progressivement mis à jour avec l'arrivée des données de localisation, et comment l'adaptation du modèle aux évolutions « long terme » des informations récoltées est mise en place. Une description plus formelle sous forme algorithmique complète cette

présentation du prototype réalisé.

Le Chapitre 5 présente les expérimentations menées pour valider la proposition effectuée. Dans un premier temps, nous présentons de façon générale la méthode retenue pour la production des données de localisation. Cette production artificielle des données est nécessaire car nous ne disposons pas à l'heure actuelle de données du monde réel. Les scénarios de validation et leur paramétrage sont ensuite présentés. Ils définissent les conditions expérimentales mise en œuvre pour valider les propositions du chapitre 3. Les résultats obtenus sont ensuite proposés et analysés.

Enfin, dans la conclusion générale de cette thèse, nous reprenons l'ensemble des points particuliers abordés dans ce manuscrit, avec en particulier un retour sur les hypothèses initiales fortes de ce travail. Ces différents points nous permettent de mettre en évidence ce qui nous semble être les principaux apports scientifiques de ces travaux. Des perspectives d'amélioration du générateur, mais également de vision plus large et plus long terme, viennent au final conclure l'ensemble de ce travail de thèse.

Chapitre 1

Données & Modèle

> *Where does the mind stop and the rest of the world begin ? . . .*
> *. . . We will advocate an externalism about mind, but one that is in no way grounded in the debatable role of external reference in fixing the contents of our mental states. Rather, we advocate an "active externalism", based on the active role of the environment in driving cognitive processes.* Andy Clark & David J. Chalmers, 1998 [44].

1.1 Les données source d'observation de l'espace technique

Ce premier chapitre est pour nous l'occasion de présenter un éclairage particulier sur la notion de données et d'information, dans un premier temps avec une approche assez générale, puis dans le contexte particulier de nos travaux. Nous présenterons également, d'un point de vue théorique, le processus de modélisation permettant la transformation de données collectées en une représentation des phénomènes physiques auxquelles elles sont liées. Enfin, nous poserons les premières briques indispensables à la compréhension de l'approche choisie dans nos travaux.

1.1.1 Les données, objectivité et fonction cognitive

En tant que construction humaine standardisées, les données sont de plus nombreuses de par le monde. Le sens qui leur est associé dépend des besoins initiaux à la base de leur acquisition. L'attention portée aux données répond à une finalité selon laquelle on les conçoit, les construit, les découvre et les classifie (voir l'application proposée par [47]). Généralement, les données constituent l'interface entre le phénomène observé et les modèles d'explication et/ou de caractérisation de la réalité [127]. Pendant le processus symbiotique d'interaction entre données et modèle, les données

sont, dans la plupart des cas, des entités statiques. Le monde réel est cependant lui en évolution continue.

La recherche de la compréhension objective du système passe par l'obtention de données résultantes d'une forme de permanence du phénomène observé, et par le biais de moyens exogènes au processus cognitif initial du modélisateur. Dans les développements et idées de *l'extended mind* [44] (les théories sensorimotrices [140], la cognition incarnée [184] et l'enactivisme [161]), les données constituent le reflet d'une extension metacorporea des processus cognitifs [82, 8].

Les dimensions normative (ou constitutive, doctrine de la constitution des objectivités) et cognitive (le fait que c'est une conscience qui est constituante) [145] jouent un rôle important dans la description des données :

1. Dimension normative. Les données sont considérées comme des caractéristiques connues *a priori* , et sont utiles pour observer un phénomène.

   La fleuriste : Mes données sont composées par la couleur, et le nom de chaque fleur dans la collection. J'ai envie d'avoir une grande diversité d'espèces mais aussi de couleurs.



Nom *Sorbus mougeotii*	**Nom** à trouver	**Nom** à trouver
Couleur blanc	**Couleur** rouge	**Couleur** bleu
Date 1958	**Date** 1961 ou 1962	**Date** 1957
Lieu Saint Petersbourg	**Lieu** Chez mes parents	**Lieu** Prairies du Lac

Cette approche est bien imparfaite. Pour avoir du sens, les données doivent être liées à la connaissance de l'utilisateur-expert, ou encore l'utilisateur a besoin de croire à l'architecte des données (qui agit comme un dictateur des connaissances, parce que les unités de données du discours sont contraintes par les spécifications de celles-ci). Par exemple, nous croyons, et organisons nos vies avec les dates du calendrier grégorien, mais l'année n'est pas un pointeur absolu et direct des mouvements célestes courants. Le calendrier grégorien a été fixé par la bulle papale

Inter Gravissimas émise par Grégoire XIII le 24 Février 1582. La dimension normative est claire, Methuen [126] étudie l'opposition à cette réforme sur la base des arguments du théologien Jacob Heerbrand et du mathématicien astronome Miachael Maestlin, le mentor du Johannes Kepler.

2. Dimension cognitive dans laquelle il est nécessaire d'isoler les parties de la réalité, et de séparer les données de la réalité observée (la donnée n'est plus le phénomène). Le cas extrême de la carte de ville de la taille de la ville en est un exemple typique :

> *En cet empire, l'Art de la Cartographie fut poussé à une telle Perfection que la Carte d'une seule Province occupait toute une Ville et la Carte de l'Empire toute une Province. Avec le temps, ces Cartes Démesurées cessèrent de donner satisfaction et les Collèges de Cartographes levèrent une Carte de l'Empire, qui avait le Format de l'Empire et qui coïncidait avec lui, point par point. Moins passionnées pour l'Étude de la Cartographie, les Générations Suivantes réfléchirent que cette Carte Dilatée était inutile et, non sans impiété, elles l'abandonnèrent à l'Inclémence du Soleil et des Hivers. Dans les Déserts de l'Ouest, subsistent des Ruines très abîmées de la Carte. Des Animaux et des Mendiants les habitent. Dans tout le Pays, il n'y a plus d'autre trace des Disciplines Géographiques.*
>
> Suarez de Miranda, Viajes de Varones Prudentes, Lib. IV, Cap. XIV, Lerida,1658. [1]

Les données sont toujours liées à un modèle cognitif. Sans l'attention d'un observateur donnant à ces données du sens (autrement dit de l'utilité), celles-ci seront amenées à disparaître.

1.1.2 Récolte et usage des données

Au sens des théories du "vehicle externalism" et de l'"extended mind", le concept d'information peut être assimilé au concept de nature. Les travaux de Rowlands [163, 159, 161, 162, 160] montrent comment le sens d'ubiquité de l'information dans la nature est directement lié à un réseau important de relations mutuelles. L'information est là, dans la nature même de toute chose. La différence entre information et donnée est très importante. L'information participe aux liaisons entre des phénomènes et les objets participant ou subissant ces phénomènes. Les données sont des outils artificiels, des objets techniques de liaison cognitive entre des phénomènes passés et futurs.

1. Borges J.-L., 1994, *De la rigueur de la science* [26].

Dans les systèmes techniques, cette information peut trouver différentes sources et transiter par des médias de plus en plus nombreux et variés, spécialement conçus (ou disposés) à cet effet. Le monde de la production industrielle n'échappe pas à cette règle. Malgré des environnements très nombreux et différents, le contrôle des systèmes industriels passe par une instrumentation (via des ensembles de capteurs et actionneurs : $\{C, A\}$ [2]) de plus en plus poussée. Ces capteurs, et les capacités actuelles de communication, génèrent une impressionnante quantité de données dont le stockage, la structuration, l'extraction d'informations et de connaissances constituent toujours des sujets de recherche importants.

Historiquement de nombreux travaux ont cherché à trouver sous quelle forme conserver les données sur les supports issus des technologies de stockage émergentes. Depuis l'invention de dispositifs de stockage de masse de données, ceux-ci n'ont cessé d'évoluer : apparition des bases de données, de l'algèbre relationnel et du modèle entité-association (Figure 1.1).

La structure de la base de données est, par hypothèse, le reflet partiel des processus modélisés. Elle donne naissance à une forme de dualité monde physique / monde données aussi appelé dialectique digital [112]. La construction de ces modèles nécessite de la part du modélisateur une expertise du processus réel et du système d'information. La représentation des processus réels nécessite plus que le stockage de simples données : il est nécessaire de conserver les structures descriptives des processus ainsi que les comportements associés (Figure 1.2).

Les travaux de recherche s'orientent alors vers la modélisation des processus avec en particulier un *focus* vers la modélisation orientée-objet, approche devenue maintenant quasi universelle. Tous ces efforts aboutissent au final à la naissance du langage de modélisation unifié des objets : UML.

Le langage orienté-objets est lié avec le problème de définition des données (c.f. 1.1.1), et cette liaison est matérialisée, selon J. Petitot [144], par la ressemblance entre la phénoménologie et le cognitivisme. Dans le même travail et avec le même esprit, Petitot propose une analogie entre la phénoménologie transcendantale et la programmation orientée-objets, voir Table 1.1 (les caractères gras ci-dessous sont nôtres) :

> *Si l'on approfondit le parallèle entre épistémologie transcendantale et épistémologie orientée objets, on voit que le "synthétique a priori" qui a tellement épouvanté les philosophes*

2. Sur l'hypothèse de la nature de capteur comme un type d'actionneur, C devient un sous ensemble d'A, $C \subset A$. Du point de vue *cognitif*, modéliser la liaison entre A et C représente l'étude du processus de mesure. Si le processus de mesure a besoin d'un observateur cognitif (ou pas), la mesure devient un problème cybernétique. On désigne donc l'ensemble capteurs et actionneurs comme un complexe $\{C, A\}$, ses éléments étant inter-connectées ou reliées mutuellement (voir par exemple [187] pour une description des protéine-complexes). Cette liaison est considéré pour le moment comme de type boite noire, en attendant une description plus exacte

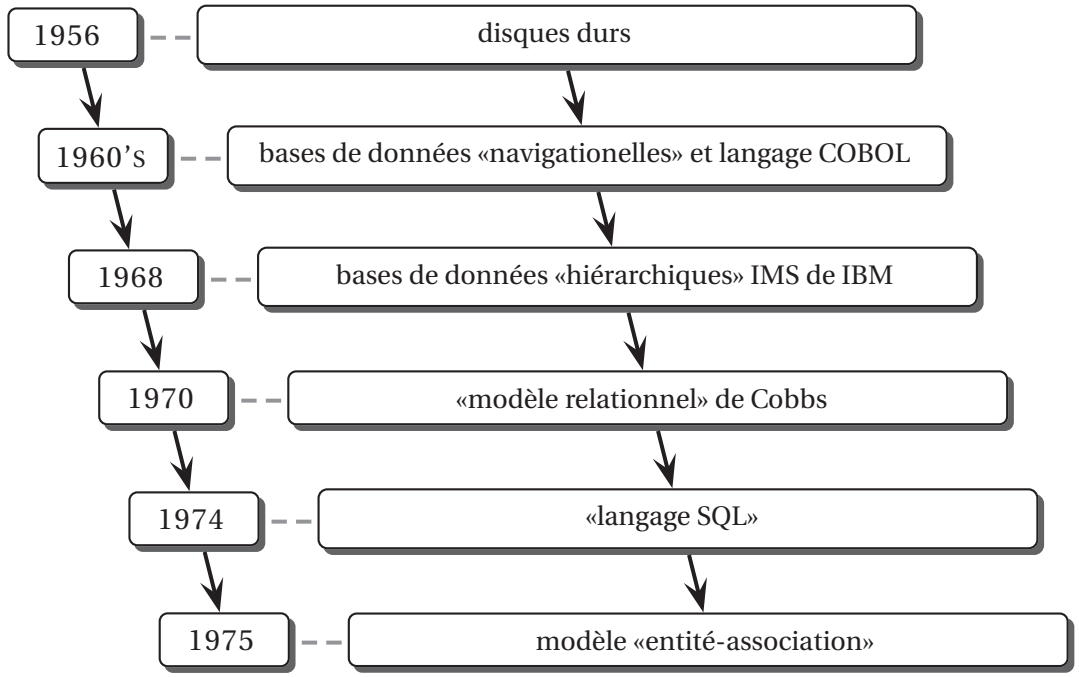


FIGURE 1.1 – Stockage de masse et genèse des supports des données.

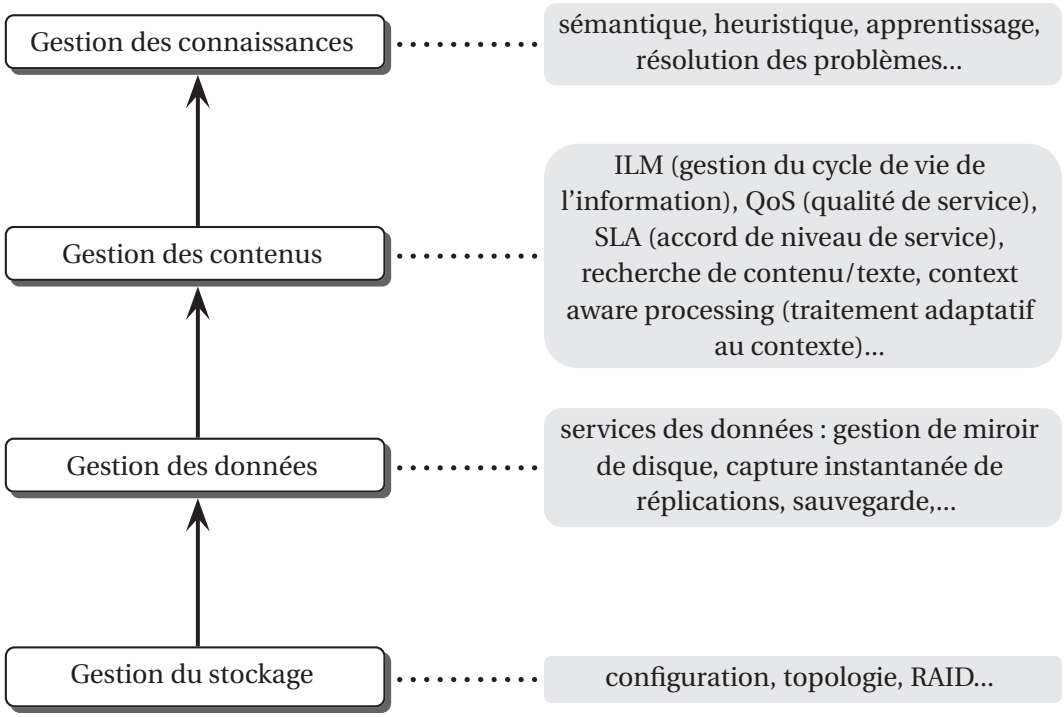


FIGURE 1.2 – Évolution du traitement des données. Adaptée de [11].

logicistes et analytiques correspond essentiellement à une thèse de **modularité et d'encapsulation des objets**. *Ce n'est pas une propriété inhérente de certains énoncés mais une* **stratégie de constitution d'objectivité**.

	Épistémologie transcendantale	**Programmation orientée objets**
Domaine	Constitution d'objectivités scientifiques	Programmation
But	Connaissances scientifiques objectives effectives	Gestion effective de connaissances
	Catégories générales de l'ontologie formelle …applicables à tout type d'objet. Analytique transcendantale et ontologie formelle.	Messages externes …applicables à tout type d'objet. Méthodes universelles.
	Ontologies régionales (Essences) Objets typiques …et règles eidético-constitutives. (Esthétique transcendantale, règles synthétiques *a priori*, vérités eidétiques régionales).	Classes Types d'objets caractérisés par des attributs et des méthodes (opérations spécifiques).
	Anticipation et spécification de ce qui doit survenir synthétiquement a priori à un objet individuel quelconque de la région.	Anticipation et spécification des caractéristiques et des comportements d'un objet individuel quelconque de la classe.
	Objet particulier d'une région (token) = instance de l'essence générique régionale (type).	Objet particulier (token) = instance d'une classe (type).
	Liens de générisation/spécialisation entre ontologies régionales.	Relations d'héritage Surclasses øClasses øSous-classes.
	Découplage de l'Esthétique transcendantale et du synthétique *a priori* relativement à l'Analytique transcendantale.	Encapsulation (modularisation) des data et des programmes (des attributs et des méthodes) constitutifs d'une classe.
	Application des catégories de l'ontologie formelle aux ontologies régionales.	Communication avec l'objet au moyen de messages.
	Schématisme transcendantal. Interprétation des catégories conformément aux caractéristiques d'une Esthétique transcendantale (Par exemple, la catégorie de cause s'interprète sous la forme de conditions initiales en mécanique classique et sous la forme d'instabilité structurelle en théorie des phénomènes critiques).	*Polymorphisme*. Spécificité des réponses de chaque type d'objet aux messages par sélection des méthodes appropriées pour répondre à un message.

TABLE 1.1 – Épistémologie transcendantale et programmation orientée objets (source [144]).

La modularité synthétique, une des caractéristiques de la programmation orientée-objet, est le découplage de :

1. ce qui relève de la mise au format spatio-temporel des données, i.e. de leur "synthèse" géométrique.
2. ce qui relève des opérations catégoriales générales (i.e. appartenant à l'ontologie formelle) applicables à ces objets déjà "intuitivement' constitués.

Cela consiste à découpler les constituants de la programmation orientée-objet entre :

1. *les méthodes spécifiques* : les objets comme liens de structures de données avec des routines sachant manipuler ces structures.
2. *les messages et les méthodes universelles* (non spécifiques) applicables à tout type d'objet prédéfini.

La compréhension des processus réels complexes, comme les processus manufacturiers (objets d'étude particuliers de cette thèse), nécessite l'accès aux données, structures et comportements les représentants. Elle peut également nécessiter de faire évoluer ou muter les données et structures dans le cadre de scénarios définis pour l'étude des réponses à différents stimulus du modèle : on parle alors de *simulation*. C'est pour aider à la construction de ces simulateurs qu'est conçu le premier langage orienté-objets [139].

1.1.3 Les données spatio-temporelles

DANS ce travail de thèse, et grâce à des capteurs-actionneurs, de nouvelles informations provenant de l'environnement, par le biais de flux des données sont à ajouter au système d'information. Cette information est dépendante du processus observé : le processus d'acquisition des données est un processus délicat, il dépend fortement des modèles de référence de conception du système d'information.

Les capteurs peuvent mesurer des grandeurs comme la température, la pression, la vitesse, l'accélération, l'intensité lumineuse, la force, les champs magnétiques, etc. Dans les systèmes manufacturiers, ou les processus de transport et d'assemblage-désassemblage sont les responsables de la plus grande partie de la dynamique du système, il est toujours intéressant d'utiliser la forte dépendance spatiale et temporelle des processus pour modéliser le système. Ce ne sont pas les seuls exemples.

L'écologie par exemple utilise pour la suivi et la caractérisation des mouvements des animaux sauvages (avec des capteurs en radio fréquences ou fondés sur GPS, ou reconstruction de chemins par des images vidéo) ces dimensions spatio-temporelles. Les échelles d'observation vont de quelques millimètres (e.g. observation de fourmis) jusqu'aux kilomètres (observation des baleines, poissons, oiseaux, etc. au niveau planétaire).

Une grande activité scientifique se concentre sur la recherche dans le domaine de l'image médicale et dans la biologie cellulaire, où les échelles d'observation vont du micromètre au centimètre. Au niveau des sociétés humaines, les réseaux de téléphonie mobile ont prouvé leur capacité à suivre des positions de personnes, et la densité spatio-temporelle des utilisateurs est un des paramètres à

considérer pour le positionnement des antennes. A plus grande échelle encore, les mesures du cosmos, montrent les limites du sens accordé classiquement aux notions de temps et d'espace.

Dans le domaine manufacturier, des applications importantes sont développées au niveau de la chaîne logistique, pour l'acquisition d'informations de positionnement des camions, containers, bateaux, et produits d'intérêt (ou dangereux), de manière à gérer cette information et rendre les processus plus performants et réactifs.

Historiquement, l'étude des mouvements d'objets physiques (i.e. balistique, astronomie, navigation) se fait avec des processus de mesure sans sophistication, mais l'étude est fructifère en théorie : le calcul différentiel est né, puis donne naissance aux « systèmes dynamiques », un des champs les plus étudiés des mathématiques.

A des siècles de distance de ces premières observations, la capacité d'observer des processus spatiaux et spatio-temporels est en croissance dans tous les domaines. Ceci est naturellement lié aux évolutions des technologies des capteurs-actionneurs, des systèmes de stockage, des formalismes de représentation, et des algorithmes de détection. Pour les plus petites particules comme les *hadrons*, jusqu'aux plus grandes, comme des ensembles de galaxies ou tout le cosmos, l'homme cherche à étudier les mouvements spatiaux des objets.

Les données spatio-temporelles constituent le résultat d'un processus de mesure : la localisation. La chose mesurée est appelé objet spatial (Z), et peut être défini par une position (x_0), et une structure (W) des objets géométriques ($w_1(x_0,x), w_2(x_0,x),\ldots$) nécessaires pour décrire sa *forme*, en coordonnées relatives par rapport à un indicateur de position.

$$\begin{aligned} Z &= (x_0, W_{x_0}) \\ W_{x_0} &= \{w_1(x_0,x), w_2(x_0,x),\ldots\} \\ & \quad W_{x_0} \subset X, x_0 \in X \end{aligned} \tag{1.1}$$

Les changements dans l'espace-temps de cette structure définissent un objet spatio-temporel ($\mathcal{Z}$) :

$$\mathcal{Z} = \{(t,Z) : Z = Z(t), t \in \mathbb{R}^+\} \tag{1.2}$$

Généralement, et dans le cas particulier de nos travaux, le phénomène d'intérêt peut être observé par le suivi d'un indicateur de position de l'objet, sa trajectoire. On parle d'objet mobile ($x(t)$), et sa trajectoire ($\mathcal{T}$) est caractérisée par une courbe dans l'espace (temps, position).

$$\mathcal{T} = \{(t,x) : x = x(t), t \in \mathbb{R}^+, x \in X\} \tag{1.3}$$

Cette trajectoire peut être approchée ($\hat{\mathcal{T}}$) par des mesures successives de la position de l'objet mobile, la source des données spatio-temporelles.

$$\hat{\mathcal{T}} = \{(t_i, x_i)\}_{i=1}^{n} \tag{1.4}$$

Avec la liaison :

$$\mathcal{T} = \lim_{\substack{\Delta t \to 0 \\ n \to \infty}} \hat{\mathcal{T}} \tag{1.5}$$

La localisation correspond à trouver la position absolue ou relative d'un objet indépendamment de l'échelle de représentation et de la technologie de mesure utilisée. La géolocalisation est la localisation d'un objet sur une carte à l'aide de positions géographiques. Pour cela il est nécessaire de disposer d'une échelle géographique et d'un système d'information compatible avec l'échelle de la représentation.

La géolocalisation et les systèmes d'information géographique sont généralement associés aux technologies *Global Positioning System* (GPS).

Le concept de géolocalisation est utilisé dans le contexte de cette thèse en référence à des domaines d'application grand public où l'information de position des objets est disponible en fonction des besoins (voir Figure 1.3).

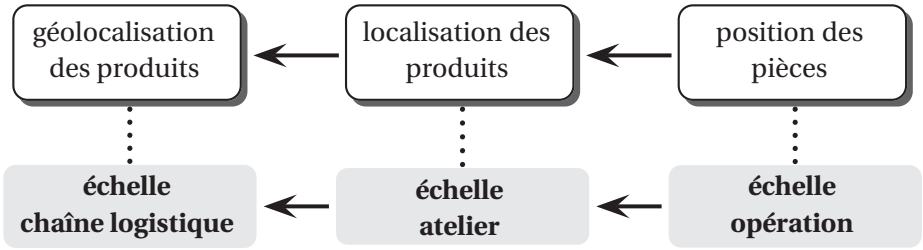


FIGURE 1.3 – Echelles d'observation et vocable de spatialisation associé

Pour notre cible applicative, le système manufacturier, l'échelle de mesure des positions se situe entre quelques dizaines de centimètres jusqu'à quelques dizaines de mètres. Dans la pratique, la majorité des travaux qui abordent la localisation sur des petites et moyennes échelles [149, 51, 25, 103] n'utilise pas le mot géolocalisation mais localisation. C'est le cas de la localisation par RFID [123, 39,

85] ou encore en robotique mobile pour la problématique du SLAM [108, 16] (Simultaneous Localization And Mapping).

Ce premier paragraphe nous a permis de présenter toute l'importance qu'il faut accorder aux données depuis les plus générales, jusqu'aux données spatio-temporelles. C'est en effet sur la base de celles-ci que l'on peut construire des modèles « synthétiques », mais également des théories. Dans le second paragraphe de ce chapitre nous présentons comment ces données peuvent être utilisées dans un processus de modélisation générique.

1.2 Modélisation de phénomènes physiques

> *. . . cette application des mathématiques aux phénomènes naturels est le but de toute science, parce que l'expression de la loi des phénomènes doit toujours être mathématique.*
>
> (Claude Bernard, 1865). This quote was also cited by Denis Noble in 2008.

1.2.1 Des données au modèle

> *To an observer $\mathscr{B}$, an object $\mathscr{A}^\star$ is a model of an object $\mathscr{A}$ to the extent that $\mathscr{B}$ can use $\mathscr{A}^\star$ to answer questions that interest him about $\mathscr{A}$.*
>
> (Minsky, 1995).

CETTE définition d'un modèle par Minsky est l'une des plus générales que l'on peut trouver dans la littérature. Elle dépend de deux éléments fondamentaux que sont *objet* et emphobservateur. Elle montre l'objectif du modèle : étudier l'objet d'intérêt $\mathscr{A}$ à partir d'un autre objet $\mathscr{A}^\star$. Cet objet $\mathscr{A}^\star$ est disponible et modifiable par l'observateur $\mathscr{B}$.

Dans le cadre de cette thèse, l'objet d'intérêt est le système manufacturier, observé par le biais de flux de données. Un des points de départ de nos travaux est basé sur l'hypothèse que l'utilisation des flux rend possible une modélisation de plus haut niveau, comme un modèle de simulation. Quand un attribut d'un objet, une grandeur physique, est disponible sur une certaine durée, on peut parler de disponibilité continue de l'information sur la grandeur.

Dans le cadre des systèmes distribués, les flux d'information issus de différentes grandeurs mesurées sont rendus disponibles pour des algorithmes de synchronisation, de filtrage et d'estimation. Pour un flux continu de données spatio-temporelles (par nature redondantes [152, 70]) utilisé dans but d'étudier un phénomène, le fait de garder toutes ces informations de position (*coté quantitatif*) n'amène aucune valeur ajoutée sur les connaissances du système d'intérêt à court ou à long terme.

Seul un modèle symbolique (*coté qualitatif* [193]) construit sur la base de ces flux de données spatio-temporelles peut permettre de répondre aux questions sur le phénomène lui-même.

La visualisation des portraits de phase des systèmes dynamiques [199, 203], ou la génération de systèmes symboliques qualitatifs [186] sont deux exemples qui utilisent les données pour expliquer un phénomène via un modèle qualitatif. En biologie théorique, la modélisation cherche à établir des catégories générales, et une dynamique qualitative [89], en visant une représentation des clôtures métaboliques et des phénomènes de haut niveau.

La Figure 1.4 montre le processus de modélisation qualitatif adapté du schéma original de Bratko [28, 205]. Elle met en évidence l'obtention des modèles qualitatifs $\star.s$ et quantitatifs $\star.v$, à partir de flux des données. Par rapport à la notation utilisée, le suffixe $.s$ représente le terme «structure» (logique) et le suffixe $.v$ représente le terme «valeur» (au sens mathématique, une détermination d'une variable ou l'évaluation d'une fonction).

Un cycle d'abstraction-concrétisation peut représenter les processus cognitifs mis en œuvre par un observateur. Ce processus simple peut, en fonction de la complexité du système à modéliser, être déployer sous la forme d'un enchaînement de cycles élémentaires. Un cycle élémentaire est représenté par :

- La collecte des données par un modèle quantitatif ($A.v$)
- L'*abstraction* du modèle quantitatif pour l'obtention du modèle qualitatif ($f : A.v \rightarrow A.s$)
- La résolution du problème qualitatif ($w.s : A.s \rightarrow B.s$)
- La *concrétisation* de la solution quantitative avec le modèle qualitatif ($g : B.s \rightarrow B.v$)

Parfois, et selon les utilisations, la concrétisation représentée par la fonction g doit aussi être calée (ou recalée) par des données quantitatives d'entrée. Autrement dit la solution quantitative du

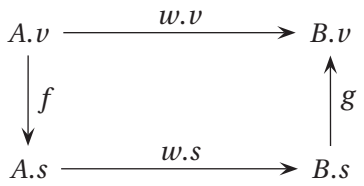


FIGURE 1.4 – Résolution de problèmes quantitatifs par le biais de l'abstraction qualitative. Le problème est représenté par A , avec ses variations $A.v$ par une *modélisation quantitative*, et $A.s$ par une *modélisation qualitative*. La fonction f représente *le processus d'abstraction* et la fonction représente g représente *le processus de concrétisation*. De la même façon, la solution du problème A par la fonction w est représentée par B, avec ses variations de solution quantitative $B.v$ (par la fonction $w.v$) et une solution qualitative $B.s$ (par la fonction $w.s$) .

problème issue de g nécessite non seulement la solution qualitative (le modèle), mais également des données quantitatives issues du problème initial :

$$g : A.v \times B.s \rightarrow B.v \tag{1.6}$$

Dans nos propositions présentées au Chapitre 3, nous nous appuierons sur ce type de processus pour modéliser nos systèmes d'intérêts. Les cycles de processus de concrétisation et abstraction pour l'interaction quantitatif-qualitatif présenté au paragraphe précédent, et Figure 1.4 peuvent être décrits en fonction de deux dynamiques : de haut niveau η et de bas niveau λ.

Un problème qualitatif $a.s \in A.s$ est l'expression de haut niveau d'un problème de bas niveau quantitatif $a.v \in A.v$. De la même façon, une solution qualitative $b.s \in B.s$ est l'expression de bas niveau (généralement $b.s$ est un ensemble de résultats, cet ensemble est paramétriquement défini) d'une solution quantitative de haut niveau $b.v \in B.v$ (la solution sélectionnée).

Une façon générale de représenter des modèles de haut-niveau et bas-niveau dans un schéma *commutatif* a été étudiée par Rueger [165], Nilsson Jacobi [91], et Roth [158].

Ce modèle possède deux dynamiques l'une de haut niveau η_e et l'autre de bas niveau λ_e. Elles sont définies dans l'espace des objets de haut niveau H, et de bas niveau L. Elles sont toutes deux liées à la projection P :

$$\begin{aligned} L &= \lambda_e(L) \\ H &= \eta_e(H) \\ P(L) &= H \end{aligned} \tag{1.7}$$

Ou les dynamiques et la projection configurent le schéma commutatif :

$$P \circ \lambda_e = \eta_e \circ P \tag{1.8}$$

L'ajout d'un indice e sur les dynamiques η_e et λ_e est utilisée pour distinguer ces dynamiques comme issues des données empiriques. Avec un phénomène de haut-niveau H, et des objets de bas-niveau L, le problème de reconstruction consiste à trouver une projection P du bas-niveau vers le haut-niveau $P(L) = H$, (voir Figure 1.5). La projection peut être vue dans certains cas comme une agrégation du microscopique (*e.g.* une collection de données brutes d'une grandeur dans une zone) vers le macroscopique (*e.g.* une moyenne des données brutes dans une zone).

A la Figure 1.6, nous présentons l'unification du schéma de Bratko (Figure 1.4) et le problème

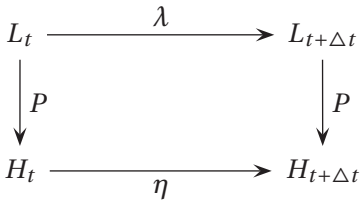


FIGURE 1.5 – Le problème de reconstruction consiste à trouver une projection P entre les données de bas niveau L et les données de haut niveau H, avec $\dim(H) < \dim(L)$.

de reconstruction (Figure 1.5). Le processus d'abstraction f est représenté par la projection P_A, le processus de concrétisation g est représenté par la projection P_B. La dynamique de haut niveau et de bas niveau d'entrée est représenté par η_A et λ_A. Pour la sortie, la dynamique de haut niveau et bas niveau est représentée par η_B et λ_B. Les fonctions de résolution quantitative et qualitative sont représentées par w_v et w_s.

Nous pouvons noter que dans l'unification, chacune des fonctions P_A, P_B, w_v, w_s commutent dans une équation :

$$\begin{aligned} P_A \circ \lambda_A &= \eta_A \circ P_A \\ P_B \circ \lambda_B &= \eta_B \circ P_B \\ w_v \circ \lambda_A &= \eta_B \circ w_v \\ w_s \circ \eta_A &= \lambda_B \circ w_s \end{aligned} \tag{1.9}$$

Dans des problèmes multi-niveaux, la solution proposée devra suivre le diagramme commutatif. Ceci permettra de valider la dynamique d'évolution du modèle obtenu par le générateur par rapport à l'évolution du système réel.

1.2.2 La reconstruction théorique et phénoménologique

SELON la feuille de route du réseau français des systèmes complexes [38], la science des systèmes complexes est dépendante, de façon critique, de la conception de nouveaux protocoles, de nouveaux modèles et nouveaux formalismes pour la *reconstruction des phénomènes* et dynamiques émergents à de multiples échelles. Plus précisément, le futur des systèmes complexes est dépendant de la réussite conjointe de deux objectifs :

1. l'acquisition des données en masse.
2. la reconstruction et la modélisation de ces données.

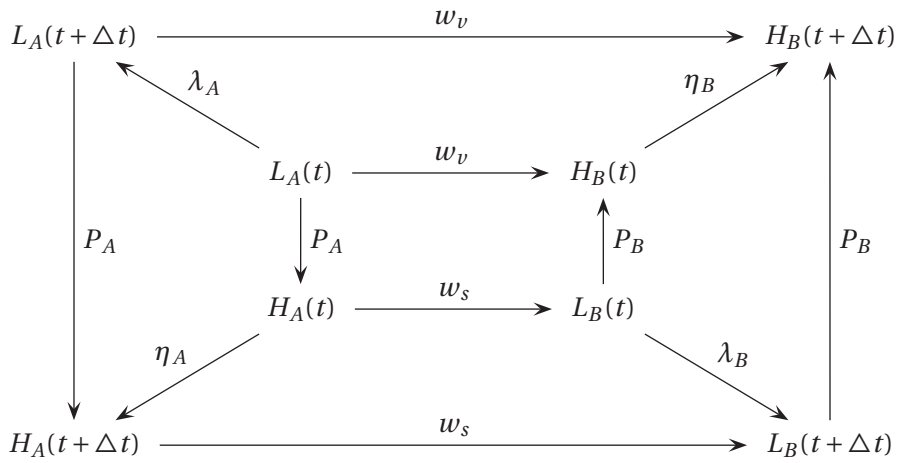


FIGURE 1.6 – Unification du problème reconstruction et la modélisation quantitative / qualitative.

La première question sur cette feuille de route verse sur l'épistémologie formelle, l'expérimentation, et les machines d'apprentissage. Cette question trouve son origine dans la communauté de la physique des hautes énergies, et les développements en biologie à l'époque ⋆-omique (génomique, proteomique, lipidomique, metabolomique) où les bases de données et les échelles d'observation

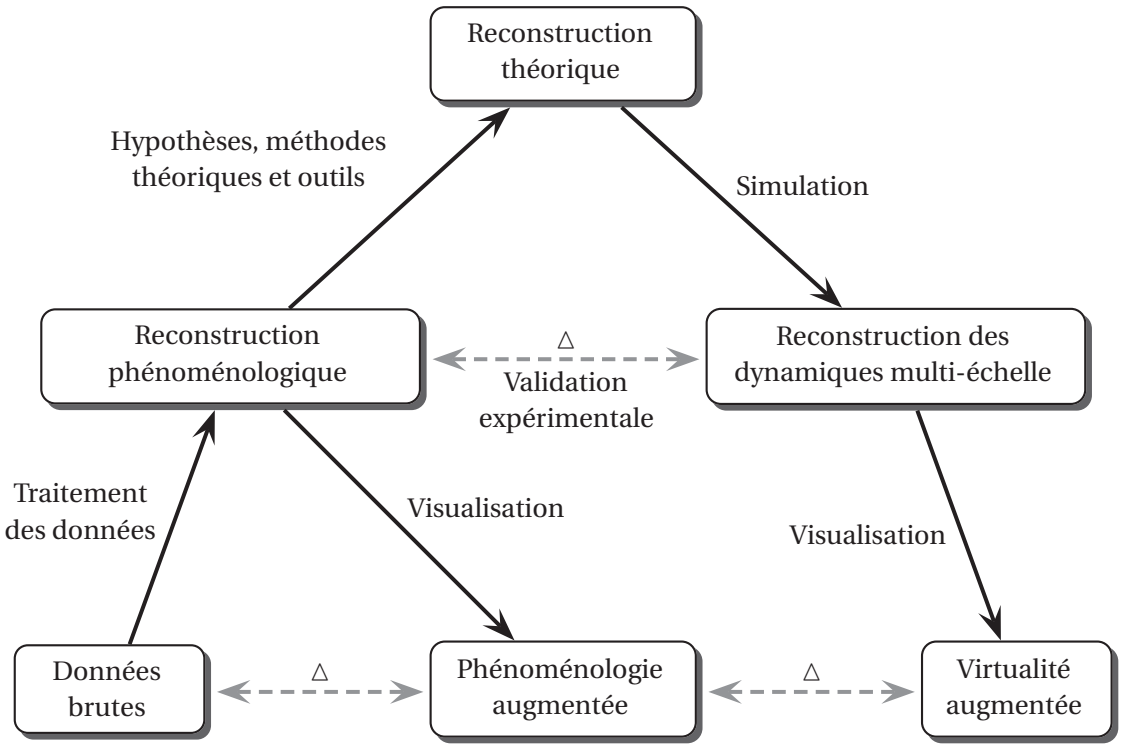


FIGURE 1.7 – Épistémologie formelle et appliquée selon [38]

sont immenses. L'effort joint de ces communautés scientifiques consiste à distribuer l'ensemble de ces données observables du système pour les manipuler. La reconstruction des dynamiques multi-échelles, en considérant les influences vers-le-bas et vers-le haut, pose la question des nouvelles méthodes et outils pour supporter le plan d'expériences et l'interprétation des résultats :

– identifier des entités d'intérêt dans une échelle temporelle et spatiale donnée.
– caractériser les interactions entre entités.
– évaluer et formaliser le comportement du système.

Le parcours proposé par le réseau français de systèmes complexes est présenté dans la Figure 1.7. Ce schéma montre comment des données conduisent à une *reconstruction* (phénoménologique et théorique) afin d'obtenir des modèles pour valider les hypothèses de départ. Le modèle doit être consistant avec les observations (△). A partir des boucles d'inférence on peut avoir différents niveaux de phénoménologies artificiellement reconstruits comme la visualisation après des résultats de reconstruction phénoménologique (*phénoménologie augmentée*) ou dans le cas de la visualisation des données provenant de modèles de simulation (*virtualité augmentée*).

1.3 Données et Modèles : lien avec notre problématique

DANS LE CADRE de nos travaux, il nous a semblé nécessaire de présenter une description du «produit» et de la «modélisation adaptative» au sens où nous l'utiliserons. Cette partie constitue donc une première description des bases de nos travaux. Elle s'appuie naturellement sur les sections précédentes pour en préciser les concepts dans notre contexte particulier d'étude.

1.3.1 Le Produit, vecteur d'informations

AU SENS étymologique le « produit » est le résultat, la conclusion, l'aboutissement…d'une ou plusieurs opérations. Il est mis en œuvre (ou fabriqué) en fonction de sa finalité par un système capable de le produire. Nous conserverons, pour nos travaux, cette définition générale du produit, même s'il en existe une multitude de définitions plus précises et donc plus spécifiques. Ce choix est réalisé afin de pouvoir envisager le produit comme le fruit d'un processus le faisant émerger à partir des composantes du système et de l'environnement.

$$\text{produit} \leftarrow (\text{composantes,environnement})$$

Dans le réseau bipartite présenté Figure 1.8, chaque type d'ensemble fondamental est représenté

par un cercle, et chaque type d'opération par un carré. Ainsi, nous avons trois ensembles fondamentaux «environnement», «objet» et «produit» : (E, O, P). Chaque objet (composant basique) $o \in O$ peut évoluer dans son environnement local $e \in E$ par des opérations $f_o : E \times O \rightarrow O$. Ces évolutions ne conduisent pas à autre ensemble. Le type d'opération $f_{op} : E \times O \times P \rightarrow P$ est une évolution à partir de l'ensemble objet vers l'ensemble produit. Un produit $p \in P$ peut évoluer dans son environnement local $e \in E$ par des opérations $f_p : E \times P \rightarrow P$. La structure du produit est générée par des opérations entre les composants fondamentaux jusqu'aux opérations du produit dans une hiérarchie des opérations.

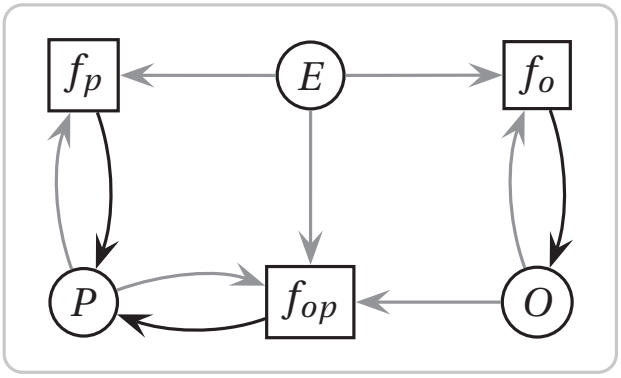


FIGURE 1.8 – Le produit défini par un réseau bipartite d'opérations.

La composition d'un produit est représentée par sa nomenclature (sous-ensembles, composants, matière), et la gamme de fabrication représente le processus d'élaboration du produit. Nous verrons au Chapitre 3 comment ces deux notions interviennent dans notre proposition. Le produit est dans le cadre de nos travaux le vecteur principal des informations collectées, quelque soit la technologie utilisée pour cela (embarquée ou non). C'est ses positions dans le temps et son identifiant qui vont nous permettre de construire notre modèle du système de production le mettant en oeuvre (voir Figure 1.9).

1.3.2 Modélisation adaptative de systèmes manufacturiers par le biais des trajectoires-produit

SELON Maria [117], la modélisation est le processus de production d'un modèle. Un modèle est une représentation des composants, de leur structuration, et du fonctionnement de l'ensemble, du système d'intérêt. La simulation peut être considérée comme les exécutions d'un modèle (avec une procédure systématique, un plan d'exécution). Pour les systèmes manufacturiers, la simulation est utilisée à différentes fins (e.g. : réglage de paramètres, tests de nouveaux mode de fonctionne-

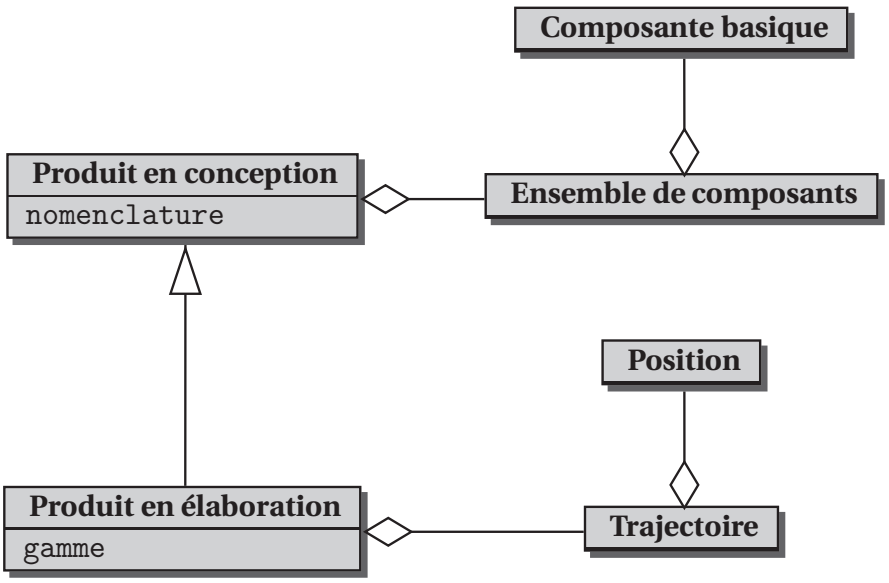


FIGURE 1.9 – Le produit comme vecteur d'informations.

ment,...) et à différents moment de la vie du système (e.g. : conception ou exploitation).

La Figure 1.10 représente notre problématique de travail. Sur la base d'un flux de données de localisation temps réel et continu, un système de construction automatique de modèle de simulation génère un modèle image du système manufacturier d'où sont extraites les données.

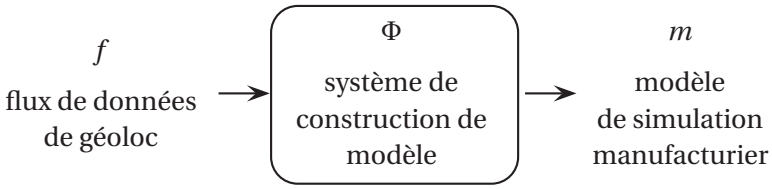


FIGURE 1.10 – Principe de construction du modèle de simulation sur la base d'informations de géolocalisation.

Un des objectifs du générateur est de maintenir à jour le modèle de simulation en cas de changement dans le flux f.

Dans cette thèse, le flux de données de géolocalisation (des produits ou composants des produits sur un système manufacturier en temps-réel) est composé par l'ensemble des données de géolocalisation (voir Figure 1.11), ou chaque donnée est un point de géolocalisation, $p = (id, r, t)$.

– id est l'identifiant donné,

– r est la position, généralement, $r \in \mathbb{R}^2$, et

– t le temps.

Ce choix a été réalisé pour permettre un suivi et la reconstitution de l'histoire du produit (son évolution) et de celles de ses composants.

On peut représenter l'évolution du système par :

$$\begin{aligned}(m_0, p_0) &\rightarrow \Phi_0(m_0, p_0) = (m_1, \Phi_1),\\ (m_1, p_1) &\rightarrow \Phi_1(m_1, p_1) = (m_2, \Phi_2),\\ &\dots\end{aligned} \tag{1.10}$$

Elle peut être généralisée par :

$$\begin{aligned}(m_i, p_i) &\rightarrow \Phi_i(m_i, p_i) = (m_{i+1}, \Phi_{i+1}),\\ i &= 1, 2, 3, \dots\end{aligned} \tag{1.11}$$

Avec m_0 le protomodèle.

La modélisation est récursive. L'état stable du modèle m_∞ est obtenu au temps t solution de l'équation de point fixe suivante :

$$\begin{aligned}(\Phi_\infty, m_\infty) = (\Phi_t, m_t) &= (\Phi_{t+k}, m_{t+k}),\\ k &= 1, 2, 3, \dots\end{aligned} \tag{1.12}$$

Bien sûr, pour défendre cette thèse, nous allons poser quelques hypothèses, mais également quelques définitions très générales. Les principales hypothèses de ce travail sont :

– Il est possible, à chaque instant, et en chaque lieu d'obtenir l'information de localisation des

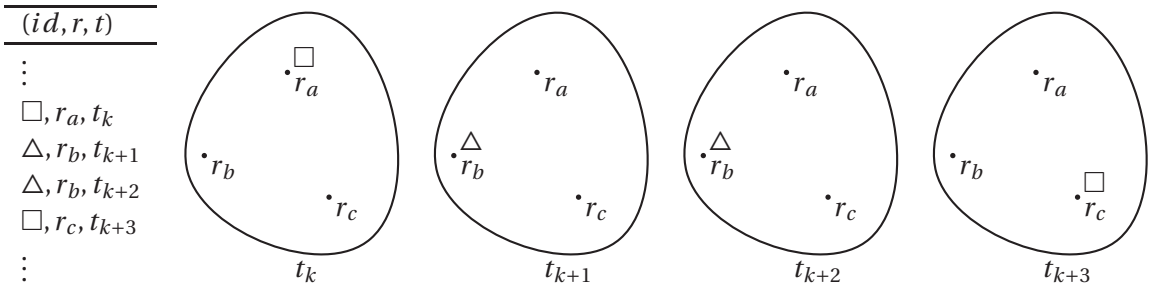


FIGURE 1.11 – Flux de données de géolocalisation. Le tableau (à gauche) représente des données : l'identifiant unique de chaque objet mobile $\{\square, \triangle\}$, la position dans le plan-(x, y), $\{r_a, r_b, r_c\}$ et le temps $\{\dots, t_k, t_{k+1}, t_{k+2}, t_{k+3}, \dots\}$ avec $t_i < t_{i+1}$. A gauche, chaque donnée est une information partielle du système complet. L'observation du mouvement de l'objet $\square$ entre la position r_a et r_b est possible. La naissance et la disparition de l'objet $\triangle$ dans la position r_b l'est également. Une estimation du temps d'attente de $\triangle$ dans r_b est $t_{k+2} - t_{k+1}$. De la même façon, on peut estimer la vitesse de mouvement de $\square$ par $\|r_c - r_a\| / (t_{k+3} - t_k)$.

objets mobiles d'intérêt.

– Chaque objet mobile possède un identifiant (id) qui lui est affecté *a priori*. C'est par exemple le cas pour tout les sous ensembles ou les composantes d'un produit. Cette hypothèse peut être levée pour des systèmes particuliers (systèmes munis d'algorithmes de suivi et de clustering, comme par exemple dans le travail de Beal sur la programmation spatiale [15]).

Le problème de traçabilité des processus d'assemblage et désassemblage, est crucial dans notre application. En effet cette traçabilité nous permettra de composer les trajectoires des composants liées à un produit, et donc au final à celle du produit lui-même.

De même nous considérerons que notre application (le générateur) et son environnement (qui fournit les données) fonctionnent de manière synchronisés. L'ajout de nouvelles caractéristiques en ligne au modèle, via le générateur, permettra une caractérisation « juste » du système d'intérêt ainsi qu'une élimination des coûts de redondance des informations.

1.4 Conclusions

LES DONNÉES spatio-temporelles sont des informations relativement basiques (i.e. position, temps), mais importantes et pertinentes. Elles constituent pour nous le point d'entrée de nos travaux visant à mettre en œuvre une modélisation de haut niveau d'un système technique tel qu'un système manufacturier. Les principaux avantages sont alors :

– De ne pas utiliser d'information de haut niveau sur l'entreprise (issues par exemple de l'ERP) : les modèles générés seront *a priori* peu complexes et donc basés sur des ontologies minimalistes.

– De rester spécifique : le modèle de simulation est le reflet des évidences empiriques particulières au phénomène.

Dans nos travaux, et de par les choix qui ont été réalisés de nous intéresser aux trajectoires des produits, il est nécessaire de modéliser ceux-ci. Les produits pourront être perçus par le biais de leur nomenclature et de leur gamme de fabrication.

L'approche théorique, présentée dans ce premier chapitre, consistant à construire dynamiquement un modèle à partir de données, avec en particulier la prise en compte de l'acquisition de celles-ci, constitue les fondements des propositions présentées au Chapitre 3.

Le Chapitre 2 va nous permettre pour sa part de positionner et de préciser nos travaux par rapport à la modélisation-simulation de façon générale, puis plus précisément par rapport aux travaux

utilisant les notions de trajectoires spatio-temporelles.

Chapitre 2

Positionnement

Au sein de la ménagerie des modèles, la simulation dispose d'une place particulière, relativement ambiguë. On parle en effet de simulation dès lors que l'on "met le modèle en action", mais aussi dès lors que l'on "sauve les phénomènes", selon l'expression de Pierre Duhem (Duhem, 1908)...

... la simulation redevient ambiguë : sur quoi mettre l'accent ? sur le modèle initial ou sur l'image (ou l'animation) donnée à voir à l'issue du calcul ? Le développement de la réalité virtuelle accentue le malaise puisqu'il s'agit alors d'immerger les sens de l'observateur dans une reproduction de la réalité. On retrouve alors le principe d'une description phénoménologique du monde, puisque les sens peuvent être piégés indépendamment de la pertinence du modèle sous-jacent à la simulation. Le scientifique risque alors de devenir esclave de ses sens et d'accepter un modèle sur la seule foi de ce qu'il lui donne à voir.
Guillaume Beslon HDR, 2008 [20].

Are computer simulations genuine experiments or simple arguments in a theoretical discourse ?
Frank Varenne, 2001 [185].

Ce chapitre est divisé en deux parties principales. Dans la première partie nous sommes intéressés, en essayant de prendre un peu de hauteur, à la modélisation et à la simulation, ainsi qu'a leurs processus de mise en oeuvre. Quelles réponses la littérature apporte-t-elle à des questions *a priori* simples et intimement liées : Qu'est la simulation ? Pourquoi est elle utilisée ? Comment est elle déployée, avec quels outils méthodes ?

Dans la deuxième partie nous focaliserons notre recherche bibliographique autour de la notion de trajectoires spatio-temporelles. L'idée est ici de répertorier les domaines applicatifs dans lesquels elles sont utilisées (monde du "vivant" et monde "artificiel"). Nous présenterons ensuite les approches parmi les plus classiques permettant l'extraction des connaissances sur la base des informations liées aux trajectoires spatio-temporelles.

2.1 La simulation

Afin de tenter d'apporter un éclairage original à la simulation dans la communauté des systèmes manufacturiers, nous proposons ici de l'aborder sous les points de vue : niveau de spécificité, statut, finalité, déploiement.

2.1.1 Niveau de spécificité

Dans son travail complet et synthétisant de nombreux travaux antérieurs, Varenne [185] propose des niveaux de spécificité sur les définitions de la *simulation par ordinateur*. Le concept de «simulation» peut, dans un premier temps, être défini au sens large comme l'utilisation des ordinateurs pour modéliser des choses. Nous appellerons cette définition «Simulation Générale» ($\mathscr{S}_G$). Une seconde définition de la simulation complète cette première définition par des caractéristiques nouvelles comme l'utilisation d'éléments discrets de base, ainsi qu'une procédure d'exécution pas à pas. Dans cette définition, l'apparition d'une systématisation du processus permet de la qualifier de «simulation méthodologique» ($\mathscr{S}_G$).

Finalement, un dernier niveau de spécificité, appelé «Simulation Stochastique» ($\mathscr{S}_S$), ajoute, aux modèles précédents, des éléments stochastiques dans un schéma procédural. (*i.e.* Méthodes de Monte-Carlo). La composante stochastique est conduite par la génération de nombres et variables pseudo aléatoires. Un diagramme de classes de ces définitions est présenté à la Figure 2.1. Comme chaque niveau de spécificité hérite des propriétés du niveau supérieur, les liaisons entre chaque définition peuvent en effet être représentées par :

$$\mathscr{S}_G \supset \mathscr{S}_M \supset \mathscr{S}_S \tag{2.1}$$

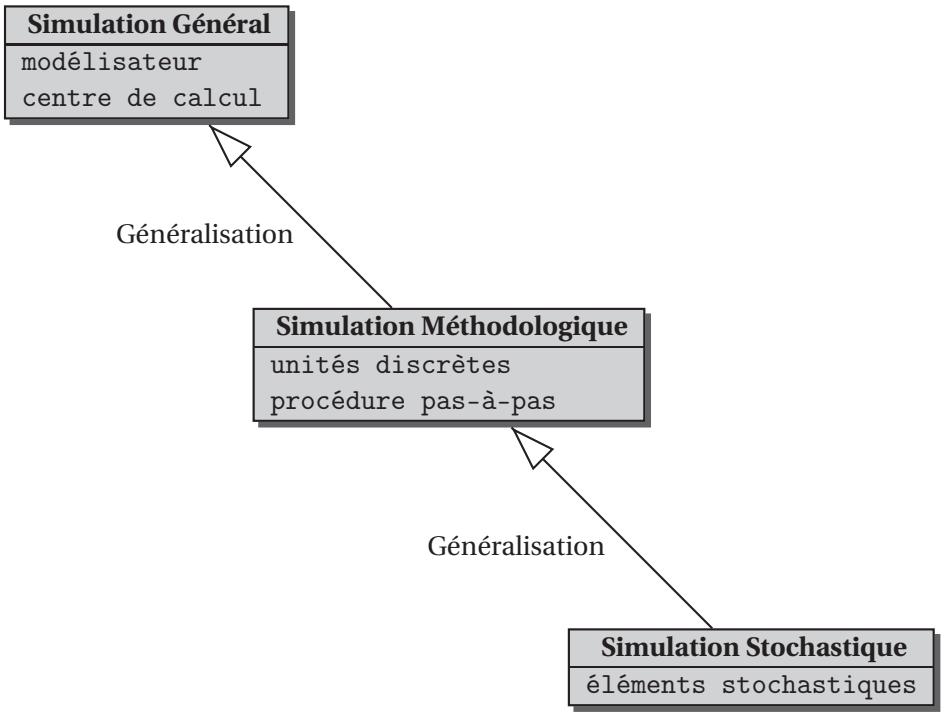


FIGURE 2.1 – Simulation par ordinateur : niveaux de spécificité.

2.1.2 Statut de la simulation : expérience, outil ou interface ?

DANS l'étude [185], Varenne présente trois thèses sur le «statut» de la simulation :

1. Une simulation par ordinateur est *une expérience scientifique*.
2. Une simulation par ordinateur est seulement *un outil*.
3. Une simulation par ordinateur est *un intermédiaire entre théorie et expérience*.

Ces trois thèses sont synthétisées dans la Table 2.1.

La réponse sur le statut de la *simulation par ordinateur* reste liée au modélisateur. Varenne pour sa part, et de manière plus précise, relie le statut de la simulation à la nature (expérience, outil, interface) que les scientifiques attribuent à leurs outils formels. Nous présentons ces différents aspects de la simulation dans les prochains paragraphes.

« Une simulation par ordinateur est une expérience scientifique »
★ **Une véritable expérience** Artificial life (Langton 1989) [104]
★ **Une sorte d'expérience** Une simulation imite la granularité de la nature Voir Von Neumann, Ulam and Kimura (Galison 96) (Dietrich 96)
«Une simulation par ordinateur est seulement un outil »
★ **Un outil pour traiter des expériences réelles** (Legay 97)
★ **Un outil théorique**
⋆ *Une méthode numérique entre autres*, selon les statisticiens (Marshall 54)
⋆ *Un argument conceptuel* (Hartmann 95) (Dennett 95) (Stöckler 00)
⋆ *Une expérience de pensée opaque* (Di Paolo 00)
« Une simulation par ordinateur est un intermédiaire entre théorie et expérience »
★ **Un nouveau moyen de capturer et d'entendre la complexité sans avoir une compréhension** (Wagensberg 85)
★ **Un calcul étape par étape est une expérience *a priori*** (Laskar 89) (Humphreys 90) (Rohrlich 90) (Ekeland 95) (Bedau 98) (Thompson 99) (Parrochia 00)
★ **La simulation est une "zone d'échange" entre les théoriciens et les expérimentateurs** (Galison 96) (Galison 1997)

TABLE 2.1 – Classification sur des «statut» de *simulation par ordinateur* selon Varenne [185]

2.1.3 Intérêt de la simulation

L'UN des intérêts scientifiques majeurs de la simulation reste lié à l'explication et la compréhension des phénomènes auxquels elle permet d'aboutir : c'est le *simuler pour expliquer*.

Ces explications peuvent être l'un des points de départ de théories scientifiques. Dans cette perspective, la boucle épistémologique présentée Figure 2.2 *simulation* ↦ *théorie* ↦ *simulation* peut apparaître comme une réduction du diagramme *d'épistémologie formelle et appliquée* présenté au Chapitre 1, à la Figure 1.7.

Le *paradigme génératif* proposé par Epstein [66, 63, 64] et révisé par Conte [49] s'appuie sur une analyse de la boucle *simulation* ↦ *théorie* ↦ *simulation*. Il représente le lien étroit entre le proces-

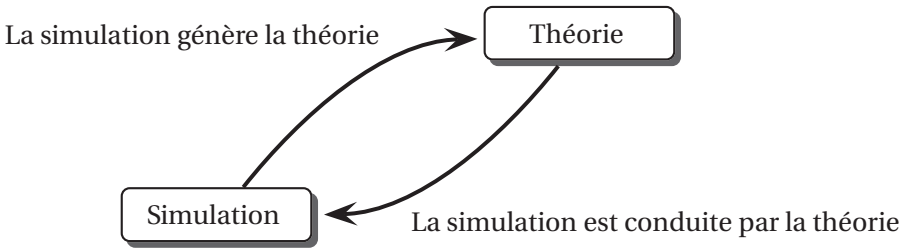


FIGURE 2.2 – Simulation et théorie : Paradigme génératif.

sus de construction du modèle d'un phénomène et l'explication du phénomène en lui-même. Ce paradigme est basé sur l'hypothèse qu'un *processus de génération - par des éléments basiques* (particules, agents, atomes, entre autres) peut représenter *une explication du phénomène*. Une définition de simulation de type $\mathscr{S}_M$ ou plus spécifique (voir équation 2.1) est nécessaire pour le paradigme génératif.

Le but du processus de génération (la simulation) est d'observer la *virtualisation* des caractéristiques clés du phénomène. En résumé, on construit un modèle pour être au plus proche des explications du phénomène représenté.

Dans le cadre de ce paradigme, si le phénomène sous étude est ϕ, l'idée est que sa génération $G\phi$, peut (ou non) être une condition nécessaire, mais (si oui) non suffisante, pour l'explication du phénomène $E\phi$. D'un point de vue formel :

$$\forall \phi \quad (\neg G\phi \rightarrow \neg E\phi) \tag{2.2}$$

Par exemple l'apparition d'un résultat attendu issu d'un modèle de simulation ne garantit pas la validité du modèle.

Le paradigme génératif peut être qualifié de faible ou de fort. Pour la qualification de génératif faible, la génération du phénomène ($G\phi$) est nécessaire mais ne suffit pas pour expliquer le phénomène ($E\phi$) :

$$E\phi \supset G\phi \tag{2.3}$$

Le plus souvent (quand la simulation est utilisée comme un outil), la simulation est un processus "génératif faible" car elle ne peut à elle seule expliquer l'ensemble des phénomènes du système. Il est nécessaire d'avoir un certain degré d'expertise pour pouvoir interpréter les résultats.

On parle de variation générative forte lorsque la génération du phénomène ($G\phi$) est l'explication du phénomène ($E\phi$) :

$$G\phi \equiv E\phi \tag{2.4}$$

Deux exemples de modèles de simulation conçus pour expliquer des phénomènes réels peuvent à ce titre être présentés, le premier sur le développement des plantes (L-systems), le second sur la croissance des cristaux (Automates cellulaires)

- L-Systems (Lindenmayer & Prusinkiewicz [110, 111, 148, 164]) basée sur une grammaire formelle, permettant, par un procédé algorithmique, de modéliser le processus de développement et de prolifération de plantes ou de bactéries.
- Automates cellulaires (Ulam & von Neumann [138, 21, 31, 195]) qui sont à la fois un modèle de système dynamique discret et un modèle de calcul. Le modèle, très simple dans sa définition, permet d'atteindre certains comportements macroscopiques complexes. Il est l'un des modèles standards dans l'étude des systèmes complexes. Les automates cellulaires sont fortement étudiés et sont utilisés pour modéliser de nombreux phénomènes.

L'importance de l'approche générative est liée au modèle explicite du système, qui possède une forme d'universalité le rendant accessible même à des non spécialistes.

Certains théoriciens n'adhèrent pas avec cette vision du paradigme génératif et en particulier sur la connexion entre simulation et théorie. Ils parlent de la dérive de la simulation comme une activité *ad-hoc*, sans méthode claire (et sans une façon unique de concevoir des simulations) :

> *That is, the assumptions necessary to get the simulation 'up and running' are unusual or unrealistic, and the results are contrived and are merely the product of* **the dark manipulative arts of the simulationist**. *They are, in short, a simulation artefact. A reasoned response to this kind of criticism is that the simulation is no more or no less of an artefact than any other kind of mathematical or verbal model. All models are to some extent an artefact [118].*

2.1.4 Usages de la simulation dans les systèmes manufacturiers

UNE DES utilisations les plus reconnues de la simulation de façon générale est celle de la prédiction de l'évolution d'un système. Epstein [65] complète cette raison principale par une liste des raisons pouvant justifier la construction de modèles de simulation (*e.g*, expliquer (très distincte

de prédire), guider la collecte de données , montrer la dynamique de base, suggérer des analogies sur la dynamique , découvrir de nouvelles questions , promouvoir une habitude scientifique de tournure d'esprit , limiter les sorties à résultats plausibles , montrer les incertitudes fondamentales , offrir des options à une crise en temps quasi-réel , démontrer des compromis / suggérer l'efficacité , vérifier la robustesse de la théorie qui prévaut lors de perturbations , mettre en avant les connaissances non compatibles avec les données disponibles, former des opérateurs, sensibiliser le grand public , révéler l'apparence simple (complexe) d'un être complexe (simple)).

Bon nombre de ces questions peuvent s'appliquer aux systèmes manufacturiers et systèmes logistiques. Elles justifient en ce sens l'usage de la simulation. De façon générale, les systèmes manufacturiers sont de plus en plus complexes et regroupent des sous-systèmes hétérogènes avec des relations non triviales entre eux. Ces structures techniques, peuvent être représentées comme un *réseau complexe* [84] ou comme un *système complexe adaptatif* [131]. Ce type de représentation rend très difficile des approches classiques de modélisation par un expert. Dans cette perspective, la modélisation des grands systèmes de production avec les moteurs de simulations et leurs interfaces traditionnelles atteint ses limites : "la simulation n'est pas pour tout le monde" [125]. L'interactivité entre l'outil et l'expert n'est plus suffisante.

Notre travail de thèse cible ce type de systèmes manufacturiers et vise à contourner les problèmes évoqués précédemment.

Depuis quelques années, la simulation de flux est devenue incontournable pour l'évaluation de la dynamique des systèmes manufacturiers [36], [142]. En effet, les modèles de simulation de flux sont utilisés pour dimensionner un système en phase de conception, pour améliorer son fonctionnement en phase de ré-engineering, et anticiper un comportement en phase d'exploitation [129]. Dans le secteur manufacturier, comme dans la plupart des domaines d'application, une simulation est très souvent réalisée par des experts humains.

Malgré tout, il n'en demeure pas moins que les phases de modélisation, puis de maintenance, de ces modèles restent des opérations délicates et chronophages, [54, 130, 55], et ce quel qu'en soit le type d'application visée. Les étapes du processus conduisant à l'élaboration d'un modèle et à l'identification de ses paramètres (récolte des données, analyse comportementale,...) sont encore très souvent réalisées par l'homme. Ces étapes constituent la plus grosse partie du temps dévolu à une simulation [191].

2.1.5 Déploiement de la simulation

Le paragraphe suivant présente, de manière synthétique, un processus rationnel de déploiement de la simulation qui présente l'avantage de répondre en partie aux critiques précédentes.

Selon Sanchez [167, 168], la modélisation pour la simulation est un processus itératif avec des boucles de rétroaction (voir Figure 2.3). Les explications pour chacune de ces boucles peuvent être synthétisées de la façon suivante :

1. **(Système d'intérêt) : (Réalité) ↦ (Modèle descriptif)**.

   Le système d'intérêt fait le lien entre la Réalité et le Modèle descriptif, il faut donc délimiter cette notion de *Système d'intérêt*. Il est en effet important d'identifier les éléments du système, les entrées exogènes et les processus intéressants entre les éléments pour décrire le modèle. Le choix du niveau de modélisation est important. Il est lié à la taille du plus petit élément à prendre en considération.

   Le principe est de décrire le système avec un nombre minimal d'éléments. Ce niveau sert à définir le champ d'application de notre modèle. A la fin de cette étape nous avons un *Modèle descriptif*.

2. **(Modèle descriptif) ↦(Modèle formel) ∧ (Solution analytique) ↦ (Réalité)**.

   Avec le modèle descriptif, nous pouvons essayer de décrire avec rigueur les comportements et les interactions de toutes les parties du système. Ceci peut être fait de différentes façons, même si elles sont souvent de nature mathématiques : le système peut être décrit comme un ensemble d'équations différentielles, ou comme un ensemble de contraintes et objectifs dans une formulation d'optimisation, ou encore par le biais d'une modélisation par des lois de probabilité ou des processus stochastiques. Cette description rigoureuse constitue le *Modèle formel*. Si ce modèle formel accepte une solution analytique, la solution peut être obtenue. Dans le cas où le modèle formel est conçu en conformité avec la réalité, il (et sa solution analytique) autorisent alors une fermeture de la boucle de rétroaction pour réitération afin d'obtenir un modèle plus représentatif.

3. **(Modèle formel) ∧ ¬(Solution analytique) ↦ (Modèle informatique)** .

   Au modèle formel sont généralement ajoutées des caractéristiques importantes représentatives de la réalité, qui au final le rendent intraitable analytiquement ou difficilement calculable. C'est le moment où la simulation par ordinateur entre dans le processus, pour décrire les comporte-

ments du système de façon algorithmique : la sortie du processus est un Modèle informatique, le modèle de simulation.

4. **(Modèle informatique) ↦ (Modèle statistique)**.

   Si le modèle informatique utilise des sources aléatoires comme éléments du modèle, celui-ci devient stochastique, et les sorties du modèle deviennent des variables aléatoires. Le modèle devra être exécuté une quantité suffisante de fois et les données de sortie collectées lors de chaque exécution pour décrire les variables aléatoires de sortie. La caractérisation d'une variable aléatoire, se faisant par le biais de modèles statistiques, les sorties du modèle forment un nouveau modèle statistique.

5. **(Vérification et validation) : (Modèle statistique) ↦ (Réalité)** .

   La boucle de rétroaction entre les sorties du modèle de simulation et la réalité servent au processus de Vérification et de validation. La Vérification est liée à la question "Le modèle informatique fait-il toujours les actions pour lesquelles il a été conçu ? La validation est liée à la question "Le modèle informatique représente-il la réalité de manière acceptable ?

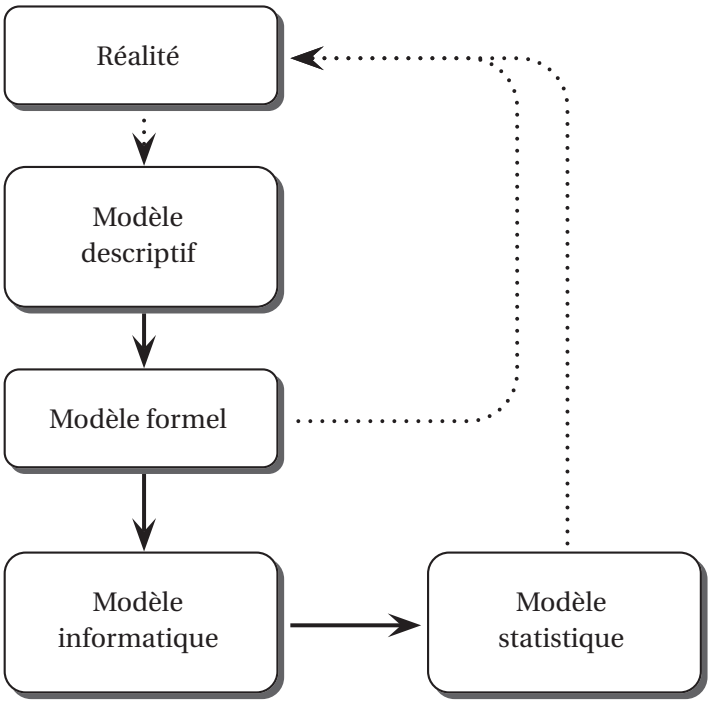


FIGURE 2.3 – Chaîne d'inférence de la modélisation pour la simulation.

Ces boucles sont représentées à la Figure 2.3. La simulation y apparaît comme un processus ra-

tionnel. Une définition de simulation de type $\mathscr{S}_S$ (voir équation 2.1) est nécessaire pour les boucles de modélisation et simulation. En conclusion, on peut retenir que la conception et la construction de modèles de simulation est un travail plus délicat que la construction de modèles analytiques, la chaîne d'inférence étant plus longue.

Dans la littérature des systèmes manufacturiers, différents outils théoriques sont utilisés pour modéliser et simuler les systèmes de production manufacturiers. Nous présentons ici les caractéristiques des outils plus utilisés :

- **Diagramme de Forrester** . Ce type de représentation [93, 69] implique une dynamique continue (système dynamique aux états et à temps continus). Le modèle est lié à un système d'équations différentielles ordinaires (ODE) et peut être résolu analytiquement ou par le biais d'intégrations numériques. Les algorithmes les plus utilisés sont de type adaptatif, comme par exemple LSODA [146] disponible dans la suite ODEPACK [86]. L'approximation continue est construite par le réseau des affectations entre les composantes du modèle. Si la dynamique du système est par essence événementielle (par exemple dans un processus de comptage de l'arrivée de clients à une banque), l'effet des discontinuités (effet échelon) ne sera pas considéré sur ce type de modèles. Une comparaison entre Diagramme de Forrester et Réseau de Petri est présentée dans [92], et les modèles de réseaux de Petri montrent des représentations plus simples et compactes. Une difficulté de la dynamique des systèmes utilisant les diagrammes de Forrester est liée à la validation de ce type de modèles : ils n' utilisent pas une méthodologie formelle, objective et quantitative pour les processus de validation [13]. La dynamique des systèmes de Forrester est également critiquée sur ces capacités à représenter des hiérarchies, et des rétroactions [83].
- **Graphes à liens** . Les graphes à liens [143, 72], ont été conçus pour des modélisations de flux physiques continus sous contraintes de conservation d'énergie. Le modèle reste donc très lié aux ODE.

  Les modèles basés sur les bonds graphs doivent respecter le principe que la multiplication de l'effort par le flux est égal à la puissance. Ce postulat n'est cependant pas vrai pour tous les systèmes, et parfois n'est pas important pour le modèle. Willems [192] fait une analyse de ce type de problème et en montre deux exemples : un flux thermique, et un système mécanique simple où la modélisation avec des bonds graphs donne des résultats incorrects ou incomplets.
- **Automates** . Machine à états finis [41], Petri Nets [137], Discrete Events Specification Systems (DEVS) [201, 202, 183, 147], graphes marqués [136] en général. Un automate [87] est constitué

d'états et de transitions. Un automate fini forme un graphe orienté étiqueté, dont les états sont les sommets et les transitions les arêtes étiquetées. Il est par nature discret, et peut représenter des dynamiques événementielles, discrètes et continues. Une application des machines à états finis probabilistes pour la modélisation des essaims de robots mobiles connectés par un réseau sans fil est présentée dans [194]. Une application de DEVS pour la modélisation et la reconfiguration des simulations est présentée par [130].

– **Algèbre et Calcul de processus** . L'algèbre de processus [18, 19], représente une famille de langages formels (par exemple pi-calculus [128, 169] pour les processus mobiles) permettant de modéliser les systèmes concurrents ou distribués. Network calculus [105], est plus limité en applications que pi-calculus. Sa principale application est l'analyse des performances des réseaux d'ordinateurs, en utilisant des outils des algèbres tropicales comme max-plus ou min-plus. L'algèbre et le calcul de processus sont nécessaires quand la modélisation du système de communication entre processus est fondamentale pour l'évaluation en conception.
– **Modèles d'agents** . Ce type de modélisation se focalise sur la définition et la construction des composants individuels (ou agents) du système en réseau. La dynamique globale se construit par l'interaction entre ces agents (voir par exemple [114, 23]). Une application pour des architectures de logiciel basées sur des agents est présentée par [10]. Le contrôle distribué est un autre type d'applications important [121, 109, 107]. Pour le contrôle des systèmes manufacturiers «holoniques», l'architecture d'agents a été largement analysée [32, 67, 122, 106, 68]. Ces modèles sont également utilisés pour l'ordonnancement dynamique dans les systèmes manufacturiers flexibles [102]. Des applications pour l'audit continu [37], pour la logistique [40], et pour la localisation et le suivi [73, 74] ont été trouvées. En écologie, les modèles d'agents sont utilisés pour la simulation des modèles basés sur l'individu (IBM) [78].
– **Réseaux de files d'attente** . C'est une approche particulière où le processus est représenté par un réseau de files d'attente [188, 22]. Un réseau de files d'attente est une collection de centres de service (serveurs) qui représentent une ressource, et des clients qui représentent des utilisateurs ou des transactions. L'étude des réseaux de files d'attente vise à trouver des solutions analytiques et paramétrables sur des modèles de files d'attente pour l'évaluation des performances des systèmes.

Les méthodologies de mise en œuvre de la simulation de flux n'ont pas changé en substance sur ces vingt dernières années [12, 100, 117, 166, 133, 171]. Dans le cadre de cette thèse, nous ne pouvons pas couvrir et détailler tout le cycle de simulation, mais le lecteur intéressé trouvera une bonne

référence dans les travaux de Habchi [81, 80].

Nous proposons donc ici un schéma de synthèse (Figure 2.4) qui présente les étapes couramment suivies lors de la mise en œuvre d'une simulation : modélisation, scénarios et analyse.

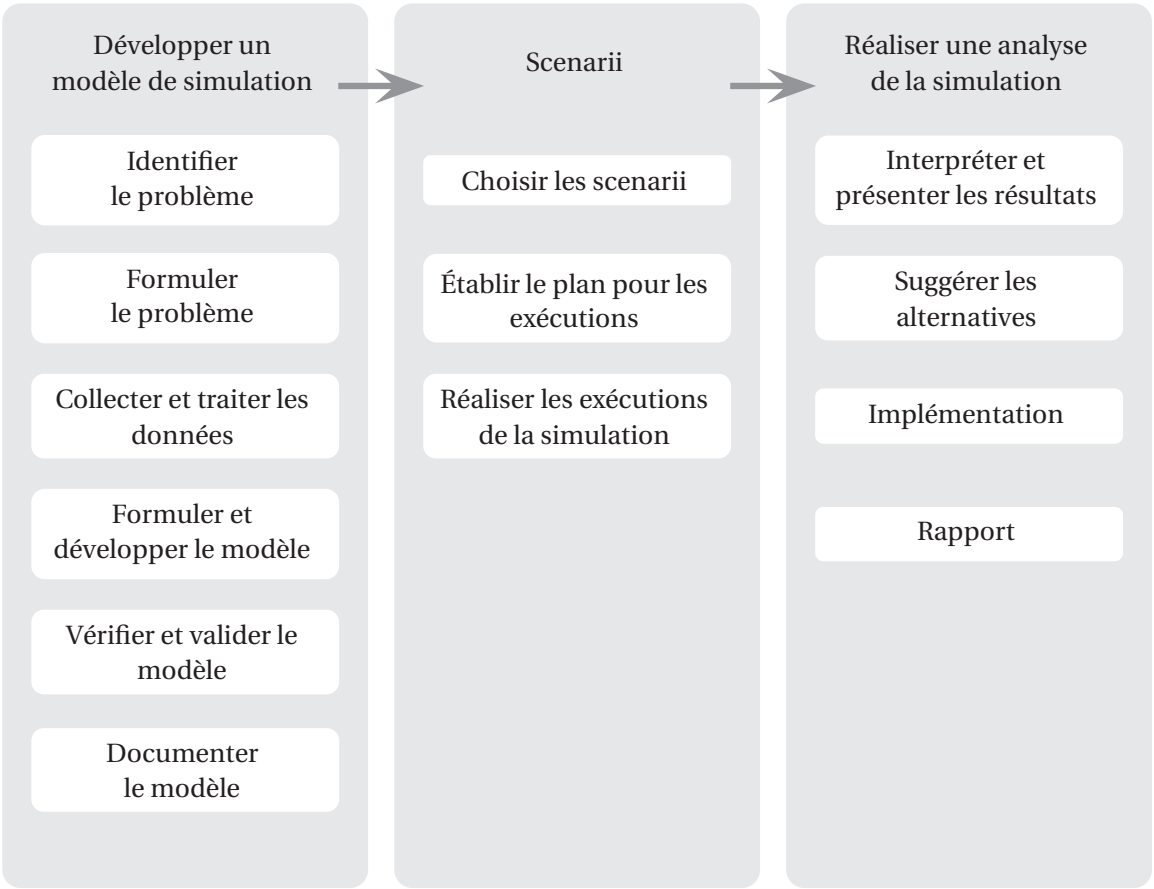


FIGURE 2.4 – Synthèse des processus de simulation utilisés pour les systèmes manufacturiers

Notre travail se focalise sur le support de modélisation en simulation, et en particulier sur la construction et la maintenance d'un modèle de simulation à partir des données spatio-temporelles des trajectoires-produit. Ce type de problématique est crucial et rendu de plus en plus difficile de part la complexité des systèmes actuels. Le paragraphe suivant, sur la base de la Figure 2.4, propose un état de l'art des outils permettant une bonne couverture des phases de développement d'un modèle de simulation, à savoir : collecter et traiter les données du système réel, formuler et élaborer un modèle, et vérifier et valider le modèle.

2.1.6 Outils supports pour le développement d'un modèle de simulation

POUR aider à la mise en oeuvre des modèles de simulation, quelques travaux ont proposé une automatisation de certaines de ses étapes de développement, par exemple avec des templates [79], ou par le biais de la modélisation conceptuelle [204] (dont le Model Driven Engineering), ou encore avec des librairies neutres[177]. D'autres approches s'alimentent directement par des données issues des ERP comme par exemple dans le cas présenté par [101]. [17] introduit quelques challenges existants pour la génération automatique des modèles de simulation dans des systèmes de production.

Notre travail porte plus spécifiquement sur les étapes de collecte et de traitement des données, de formulation et de développement du modèle, et de vérification et validation de celui-ci (Figure 2.4). Notre approche se veut de type Bottom-Up (des données aux modèles). Les contributions trouvées dans la littérature concernant ces différentes étapes présentent les outils supports suivants :

- Simulation générique ou basée sur des templates : des modules génériques et personnalisables (customisables) sont utilisés pour construire le modèle dans un champ d'application particulier [30, 124, 115, 116, 179, 79].
- Acquisition automatique des données : les données sont récupérées en ligne ou à partir des bases de données pour caler (ou recaler) un modèle [154, 153, 155].
- Fusion d'information avec le modèle : la vocation de cette fusion consiste à améliorer la représentativité du modèle de simulation, en « ajoutant », en ligne, des informations issues du système d'intérêt à simuler. L'information ajoutée contribue à l'estimation des paramètres de la simulation [55, 56, 5, 29].
- Outils de simulation conduits par les données : le flux de données conduit la modélisation, il génère la structure du modèle, et permet l'obtention des paramètres du modèle [45, 33, 61, 97, 176, 124, 88].
- Modélisation conceptuelle automatisée : la modélisation est le résultat des généralisations des procèdés. Les concepts sont de haut niveau, et leur application reste liée aux architectes du système. La possibilité d'avoir une modélisation en ligne est alors très difficile. Ce type de modélisation sert plutôt à l'aide à la compréhension du système [58, 204].

L'analyse de ces contributions nous oriente vers un regroupement de celles ci en deux catégories : la première autour de la notion de "template" ou composant générique, la seconde autour de la conduite par les données (Figure 2.5).

Ces deux catégories sont maintenant brièvement présentées :

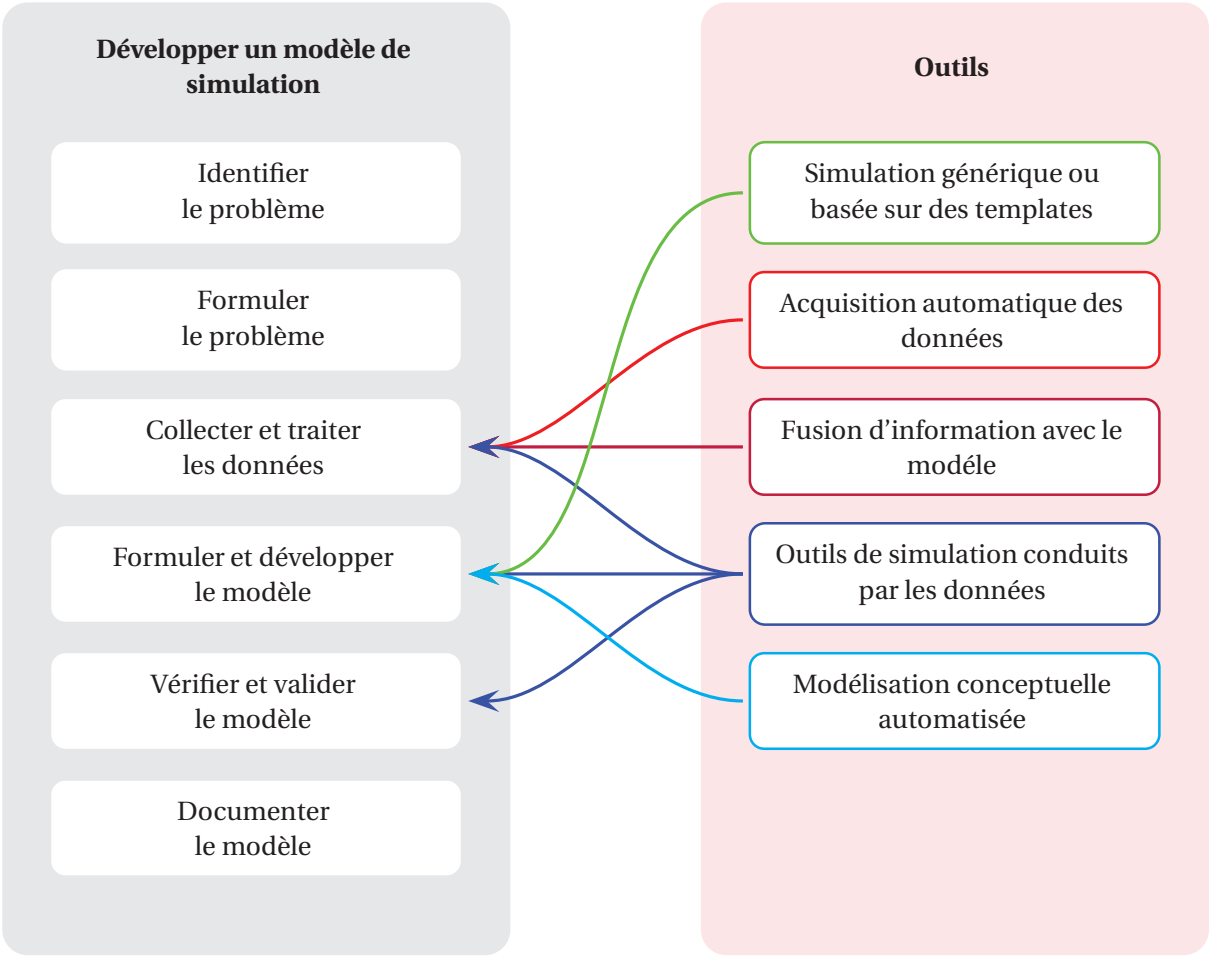


FIGURE 2.5 – Relations entre outils supports et nos trois étapes, l'intérêt du développement d'un modèle de simulation.

Simulation générique ou basée sur des templates Une approche de modélisation « Simulation générique ou basée sur des templates » (Generic or Template based Simulation, GTS) utilise un ensemble d'objets de modélisation pré-construits disponibles et issus de situations de simulation courantes.

Ces objets de modélisation peuvent tout simplement être adaptés à l'application en cours par une commutation à un état « actif » ou « inactif » des paramètres génériques de l'objet[79]. Un avantage important d'une approche de type GTS est lié à la notion de réutilisation des modèles, comme décrit dans [179, 59, 116, 30]. De plus, [177] montre comment l'utilisation des bibliothèques neutres peut élargir le spectre des applications de GTS. Certains auteurs parlent aussi de simulation composable [96, 141], la modélisation pouvant être réduite à une composition de blocs de simulation.

Outils de simulation conduit par les données Dans cette approche de simulation conduite par les données (Data Driven Simulation, DDS), le modèle est généré automatiquement par un flux de données issues de l'instrumentation du système ou issues des bases de l'entreprise. Le modèle est mis à jour lorsque les données sont modifiées. Cette approche diffère des approches traditionnelles de la simulation, où l'expert humain analyse le problème et définit chaque étape de la modélisation généralement en utilisant un outil GTS [33].

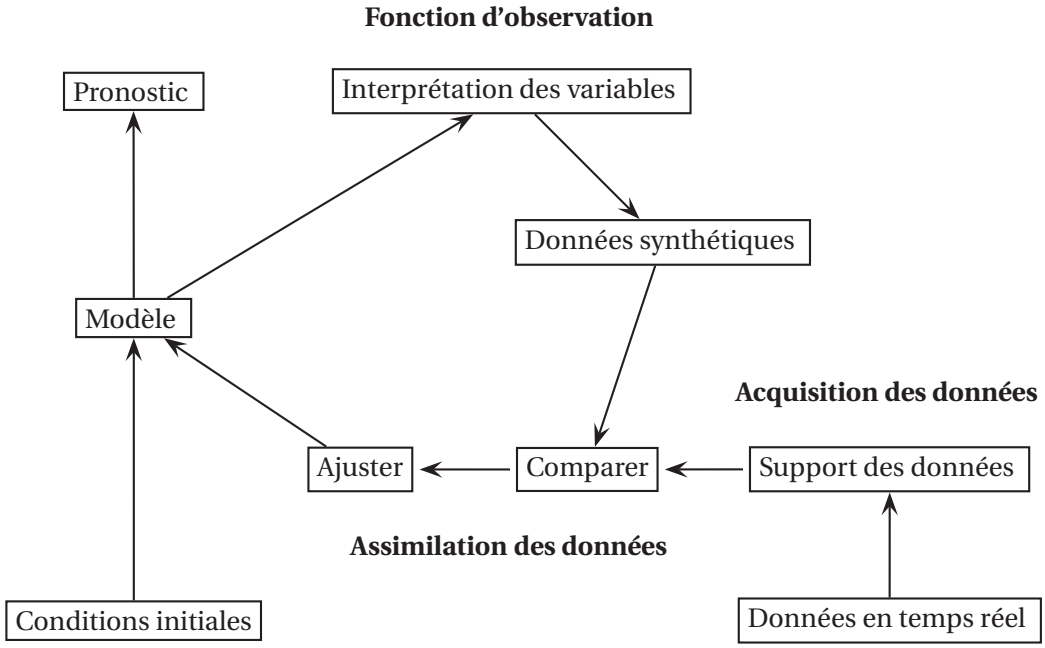


FIGURE 2.6 – Simulation conduite par les données : une application DDDAS. Schéma adapté de [48]

Parmi les applications de modélisation de ce type, on peut citer : la gestion et le contrôle des ressources en eau [176], l'aide à la décision pour la gestion d'un niveau de stock d'un produit [61]. Cependant, la maturité du concept est dû à Darema [52, 53], qui introduit des systèmes DDAS (Dynamic Data Driven Application Systems, voir figure 2.6). Les applications de simulation doivent être capables d'accepter de nouvelles données pendant le temps d'exécution et de modifier les processus de mesure de manière dynamique.

Dans [45], les auteurs présentent la DDS d'un réseau avec blocage des machines pour contrôler les encours de production. Yang [197] utilise la DDS pour construire un modèle d'inventaire, et Kennedy [97] a utilisé DDS pour aider au choix de politiques en sciences sociales.

Les principes du DDS semblent très intéressants car ils permettent l'obtention de modèles de simulation en ligne, adaptatif en structure et en paramètres, par rapport à des flux des données. Ces

modèles sont naturellement typés en fonction du domaine applicatif du système d'intérêt.

2.2 Modélisation avec trajectoires

2.2.1 Introduction

LES DONNÉES spatio-temporelles peuvent être associées à des courbes dans un espace à 3 dimensions (x, y, t), les courbes se nommant alors « trajectoires ». En général, une trajectoire est définie comme une succession temporelle d'états d'un système dynamique. Le concept de trajectoire hybride est un sujet de recherche nouveau et unificateur des théories discrète-continu dans l'automatique moderne. Par exemple dans [182, 27] une étude de la génération de trajectoires optimales, dans le cas d'une navigation autonome basée sur un champs de potentiel, est proposée. L'utilisation des trajectoires pour l'identification des systèmes est commune dans la robotique [6].

Ces notions de trajectoires étant utilisées dans de nombreux domaines scientifiques, nous en présentons maintenant quelques applications et travaux de recherches pouvant nous servir de source d'inspiration.

2.2.2 Utilisation dans le vivant et l'artificiel

Utilisation dans le vivant

Wild animals tracking Un exemple d'utilisation des trajectoires peut être trouvé dans le cas du suivi du comportement des animaux : les tracés d'animaux marins à l'état sauvage sont déterminés par le biais d'un marquage électronique des individus [71]. L'objectif est d'identifier les régions d'intérêt (Regions Of Interest, ROIs) : les habitats et les zones à haut risque, ainsi que des informations sur le comportement de ces animaux. Un des intérêts, pour nous, de ce type de travaux, est lié à la collecte à grande échelle de données pour déterminer le comportement individuel des animaux, sur la base desquels les éthologues déterminent des lois générales. Nous nous inspirerons également de la notion de ROIs dans nos propositions, pour en particulier modéliser l'agencement (ou layout) du système manufacturier considéré. Les informations sur le comportement des produits sera utilisé de son côté pour définir les règles et les paramètres de débit de flux.

Physarum Un exemple de calcul non conventionnel est proposé dans l'étude de Adamatzky [172] sur le champignon Physarum. Sans contrôle central, et avec des règles très simples, ce champignon

réalise des opérations complexes, comme générer un arbre de couverture ou un diagramme de Voronoid. Le Physarum doit optimiser son pattern d'alimentation, en cherchant la route la plus courte entre ses sources d'alimentation , puis doit diffuser les nutriments de manière efficiente. Il résoud les problèmes grâce à la mobilité de son corps.

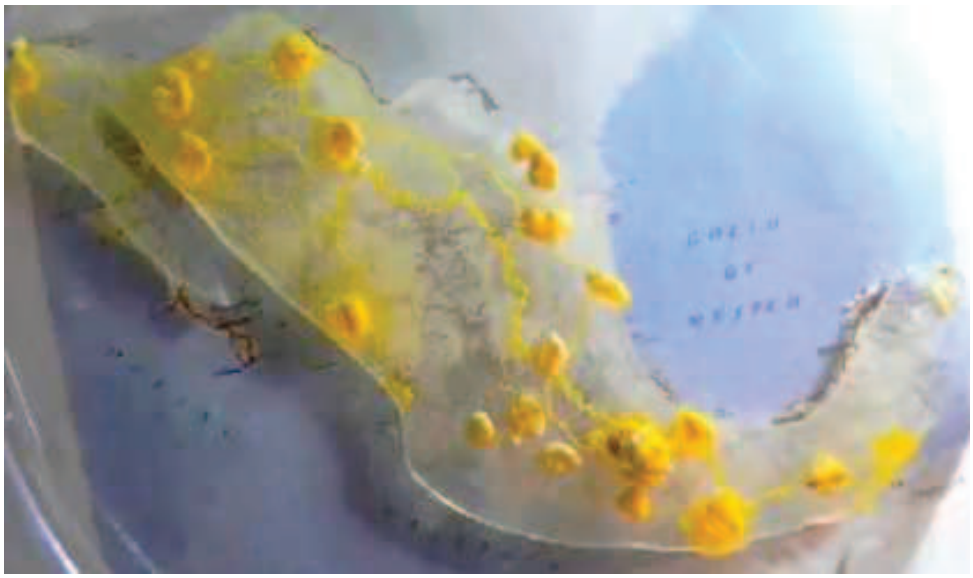


FIGURE 2.7 – Approximation des autoroutes fédérales mexicaines par le Physarum. Dix-neuf grandes villes sont marquées par des flocons d'avoine. Initialement, un morceau de plasmodium est inoculé dans la ville de Mexico.

La Figure 2.7 représente un exemple d'utilisation du Physarum [1].

L'intérêt de ce type de travail par rapport au nôtre réside dans la forme originale de modélisation du réseau de transport, réalisée de façon constructive par l'utilisation d'informations spatiales.

Eye movement Le mouvement des yeux est fortement étudié car il est impliqué dans les processus cognitifs et ses applications pratiques en supervision du comportement humain (par exemple le tracking de l'activité humaine dans des travaux critiques comme ceux des opérateurs de centrales nucléaires ou les pilotes de lignes aériennes commerciales). Les trajectoires typiques des mouvements des yeux sont étudiées dans le travail d'un des pionniers sur ce domaine de recherche (Yarbus [198]). Ils sont présentés à la Figure 2.8.

L'intérêt de cette étude dans le cadre de nos travaux est lié à l'extraction de typologies de comportement (patterns) et de connaissances globales sur la base d'observations locales.

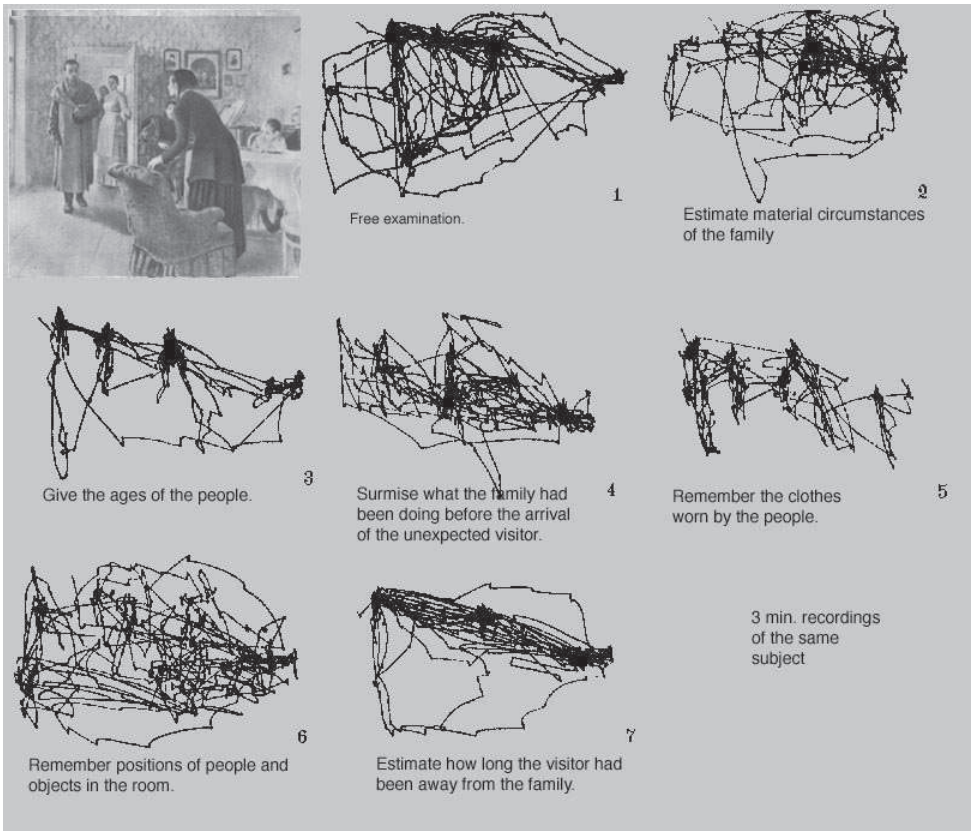


FIGURE 2.8 – Le visiteur : Cette étude de Yarbus (1967) montre comment une tâche donnée à une personne influence le mouvement de ses yeux.

Utilisation dans l'artificiel

Blogsphere Quelques recherches sur la blogsphère sont conduites par les trajectoires. Comme exemple, dans le travail de Roth [34]. La forte dynamique de ce type de système et la possibilité de tracking qu'il offre sont deux caractéristiques que l'on retrouve dans notre objet d'étude (les systèmes manufacturiers).

Transportation Networks Dans les réseaux de transport, le concept de trajectoire joue un rôle crucial. Une application récente est réalisée par Barthélémy et al. [157] sur les trajectoires d'humains, en utilisant l'information donnée par les cartes à puce Oysters.

L'étude de la dynamique de groupes et des topologies sont présentées. Ce type d'étude se base sur des grosses quantités de données issues des trajectoires humaines.

Civil Engineering La localisation intérieure utilisant les balises (tags) [39] est une application des technologies d'identification et de suivi. Par exemple, dans les chantiers de construction, les capteurs

et les balises (tags) sont incorporées dans des matériaux de construction. Des travaux explorent les possibilités d'utilisation de ces balises pour la localisation et le soutien à la construction [35, 178], pour suivre les matériaux et leurs positions en temps réel, ainsi que pour gérer les flux de construction.

Ces études sont les seules que nous ayons trouvées sur des objets *physiques* à des fins d'amélioration dans leur gestion et/ou leur usage.

2.2.3 Utilisation dans le domaine de la production et logistique

D'AUTRES exemples sont issus de domaines plus proches de celui qui est le nôtre. Ils font donc l'objet d'un paragraphe particulier.

Le problème du réseau optimal Ce problème est lié à la conception des réseaux de transport de ressources, il est connu aussi comme le problème de Scott (1969). Dans la conception de ces réseaux, les trajectoires à suivre doivent être planifiées sous contraintes économiques liées au choix d'un chemin en particulier.

Routes des bateaux Les trajectoires des bateaux font l'objet de certains travaux de recherche. L'analyse du réseau peut être faite par le biais de bateaux munis de systèmes automatiques d'identification (Automatic Identificators Systems, AIS). Dans ces études, le réseau global (de taille mondiale) des chemins de bateaux cargo, a été considéré. Ces systèmes servent au suivi en temps réel de la flotte de bateaux et traitent un très gros volume d'information [94].

Robotique L'un des domaines les plus avancés dans l'utilisation de la localisation est celui de la robotique mobile [25, 132]. De nombreuses techniques sont utilisées pour le suivi, la navigation et la localisation des robots. Ces applications utilisent des algorithmes génétiques [16], des agents réactifs [74] et des phéromones artificielles [85], qui font partis des outils les plus populaires de l'intelligence artificielle. Localisation et cartographie simultanée (SLAM) est la dénomination du problème posé quand un robot mobile doit construire une carte d'un environnement inconnu et doit en même temps naviguer dans cet environnement en utilisant cette carte [175, 108]. Ce problème est intéressant, car il gère l'incertitude dans les mesures afin de construire un modèle formel de l'environnement. Pour les systèmes manufacturiers, le modèle de l'environnement est lié au layout du système qui pourrait être construit sur la base de l'incertitude des données issues des capteurs de localisation des produits.

Production and Logistics Les applications dans le domaine de la production et logistique de ces technologies sont de plus en plus nombreuses [150, 99, 40, 98]. La traçabilité des produits, l'inventaire des stocks produits, la géolocalisation d'une flotte de transport ont sont quelques exemples applicatifs parmi les plus classiquement mis en œuvre. A l'heure actuelle le positionnement spatial d'objets physiques se limite à des objets « volumineux » (camions, palettes, engins de manutention...) ou à des personnes en raison des coûts et des limites technologiques existantes.

2.2.4 Extraction de connaissances des trajectoires

DE NOMBREUX et divers domaines applicatifs et/ou scientifiques utilisent des informations de localisation. Des modèles de localisation sont disponibles pour les environnements augmentés [76, 170], des services basés sur la localisation [196, 50], pour la prédiction de la mobilité d'objets en temps-réel [4], et les services pour des objets-mobiles [189]. Bunge (1964) a initié des travaux sur des applications géographiques qui cherchent à mettre en relation les variables spatiales et temporelles d'un territoire donné. Les systèmes d'information géographique (SIG) bénéficient de l'avantage de stocker les relations entre les ROIs et leurs attributs. L'utilisation généralisée des GIS [43, 151, 173, 24] est le meilleur exemple des avantages de l'utilisation des informations spatiales (et spatio-temporelles).

Identification Spatio-temporelle L'identification des systèmes spatio-temporels est l'un des sujets importants dans la communauté de l'identification, parce que lié au contrôle des fluides et à l'analyse des systèmes gouvernés par des équations aux dérivées partielles. On peut citer par exemple, l'identification de systèmes non-linéaires spatio-temporels via des filtres partitionnés [174], ou l'identification de patterns complexes dans des « Coupled Map Lattice » [46]. Nous pouvons trouver une importante source de ce type d'études dans la communauté de mécanique des fluides, mais aussi dans la modélisation en écologie [180]. En économie spatiale, l'identification des modèles peut être faite par analogie à la loi de gravitation de Galileo, car on y parle de modèles de gravitation. Un flux particulier entre deux localisations peut être représenté comme une fonction du flux d'entrée dans la première localisation, du flux de sortie dans le deuxième localisation, et de la distance entre noeuds : $F_{ij} = a_i b_j O_i I_j f(d_{ij})$, ou F_{ij} est le flux, O_i est le flux de sortie, I_j est le flux d'entrée, et d_{ij} est la distance entre noeuds. $f(d)$ peut etre choisi comment une loi de puissance avec décroissance exponentielle, comme $f(d_{ij}) = d_{ij}^{\gamma} \exp(-d_{ij}/k)$, ou a_i et b_j sont des paramètres qui peuvent être estimés de façon itérative à partir des données.

Calcul spatial Le calcul spatial [15] est la description et le contrôle des systèmes multi dispositifs qui sont distribués avec comme objectif de construire une forme d'intelligence par communication-collaboration. Dans ces systèmes, la difficulté de communiquer entre dispositifs dépend de la distance entre eux (voir Figure 2.9. Les challenges à relever ici font partie d'un thème plus large, "l'ingénierie de l'auto-organisation", ou la production de comportements agrégés prévisibles par des éléments qui interagissent localement. Par exemple, les réseaux de capteurs, les « robotic swarms », les cellules durant la morphogénèse, les FPGAs ou réseau logique programmable, les systèmes sans fils ad-hoc, les biofilms, et les systèmes de contrôle distribués, peuvent être considérés comme étant de type « ordinateur spatial » .

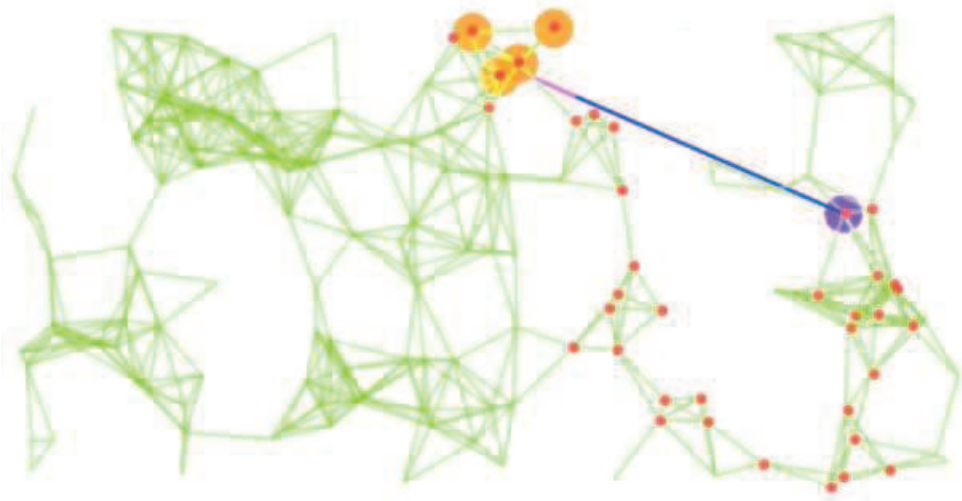


FIGURE 2.9 – Exemple de calcul spatial : suivi d'une cible et envoi de son emplacement à une base mauve.

Applications Des applications de bases de données sont nombreuses. Un survey sur des « data mining temporal », spatial et spatio-temporel [156] montre une grande quantité d'applications, où sont présentées des requêtes spatio-temporelles, du clustering spatial, et du recurrent pattern mining.

Il existe une grande hétérogénéité des domaines d'applications. Quelques cas représentatifs en sont :

- L'apprentissage des régions d'intérêt et le pronostic de mouvement [7].
- Une base de données distribuée spatialement [75].

2.3 Conclusions du chapitre

Ce chapitre à caractère bibliographique nous a permis de faire le point sur deux aspects très en lien avec notre problématique de thèse, à savoir la simulation et la génération de modèles basée sur la notion de trajectoire. Nous n'avons pas trouvé dans la littérature de travaux sur la génération de modèles de simulation, pour les systèmes manufacturiers, basée sur les trajectoires.

Il ressort de ce chapitre que l'une des caractéristiques les plus importantes des données spatio-temporelles est leur capacité à coder les relations structurelles du système à modéliser, comme par exemple, l'agrégation spatiale des données, l'identification des ROIs, et le réseau de transport entre les ROIs [200].

Concernant les processus de modélisation propres aux systèmes manufacturiers, les connaissances acquises sur des problèmes similaires permettent l'identification de templates et de modules génériques, qui peuvent être ajoutés aux logiciels de modélisation et simulation. Le processus de modélisation de type SGT consiste à joindre, connecter et personnaliser ces modules existants. L'évolution des outils de modélisation de type SGT procède donc dans ce cas à une forme d'extension sur la base d'ajouts de templates, eux-mêmes expression des réductions faites à la main sur l'espace des configurations valides d'un sous système particulier.

En plus de l'aide à la modélisation, les progrès des technologies actuelles permettent tives des processus classiques de modélisation et de maintenance en simulation. Au final, et sur la base de la revue de littérature effectuée, il a été montré qu'il existait la possibilité d'utiliser un flux de données de localisation pour générer un modèle de simulation dans un schéma adaptatif. L'existence de situations courantes sur les systèmes d'intérêts permet lors du processus de modélisation de les approcher par des structures génériques (par exemple, un réseau de transport avec des nœuds client/serveur) qui peuvent être découvertes et mises à jour grâce aux sources de données disponibles. Nous présentons dans le chapitre suivant les détails de la construction du modèle de simulation sur la base des ces différentes analyses.

Ces quelques exemples dans le domaine de la production et logistique montrent un intérêt relativement récent de la communauté pour les approches spatiales et spatio-temporelles.

Chapitre 3

Proposition

MAJOR MOKOTO KUSANAGI *You talk about redefining my identity. I want a guarantee that I can still be myself.*

PUPPET MASTER *There isn't one. Why would you wish to ? All things change in a dynamic environment. Your effort to remain what you are is what limits you.*

攻殻機動隊 (Ghost in the Shell). 押井 守 (Mamoru Oshii),1995 (voir [120]).

3.1 Introduction et présentation de la méthodologie proposée

UN SYSTÈME manufacturier est généralement perçu et modélisé comme un système à événements discrets. Les données utilisées et manipulées par ces systèmes sont en général très hétérogènes. Cette hétérogénéité fait partie intégrante du système manufacturier : l'ajout d'une nouvelle source de données doit conduire à une plus exacte représentation du système.

Les systèmes manufacturiers sont des systèmes artificiels, développés et conçus de manière rationnelle. Les modèles de ceux-ci sont réalisés sur la base de templates ou de procédures dont les limites sont donc relativement bien cernées. Ces templates servent lors de la construction des structures des processus à modéliser, et lors de l'estimation des paramètres de ces structures. Dans ce travail nous traitons l'élaboration de modèles de simulation générés et conduits à partir de données dont l'acquisition est effectuée en ligne. La Figure 3.1 synthétise de manière graphique les cheminements possibles permettant d'aboutir à la génération de différentes applications sur la base d'infor-

mations de localisation issues du monde réel ou d'un modèle de simulation. L'obtention d'un modèle de simulation fait partie de ces applications.

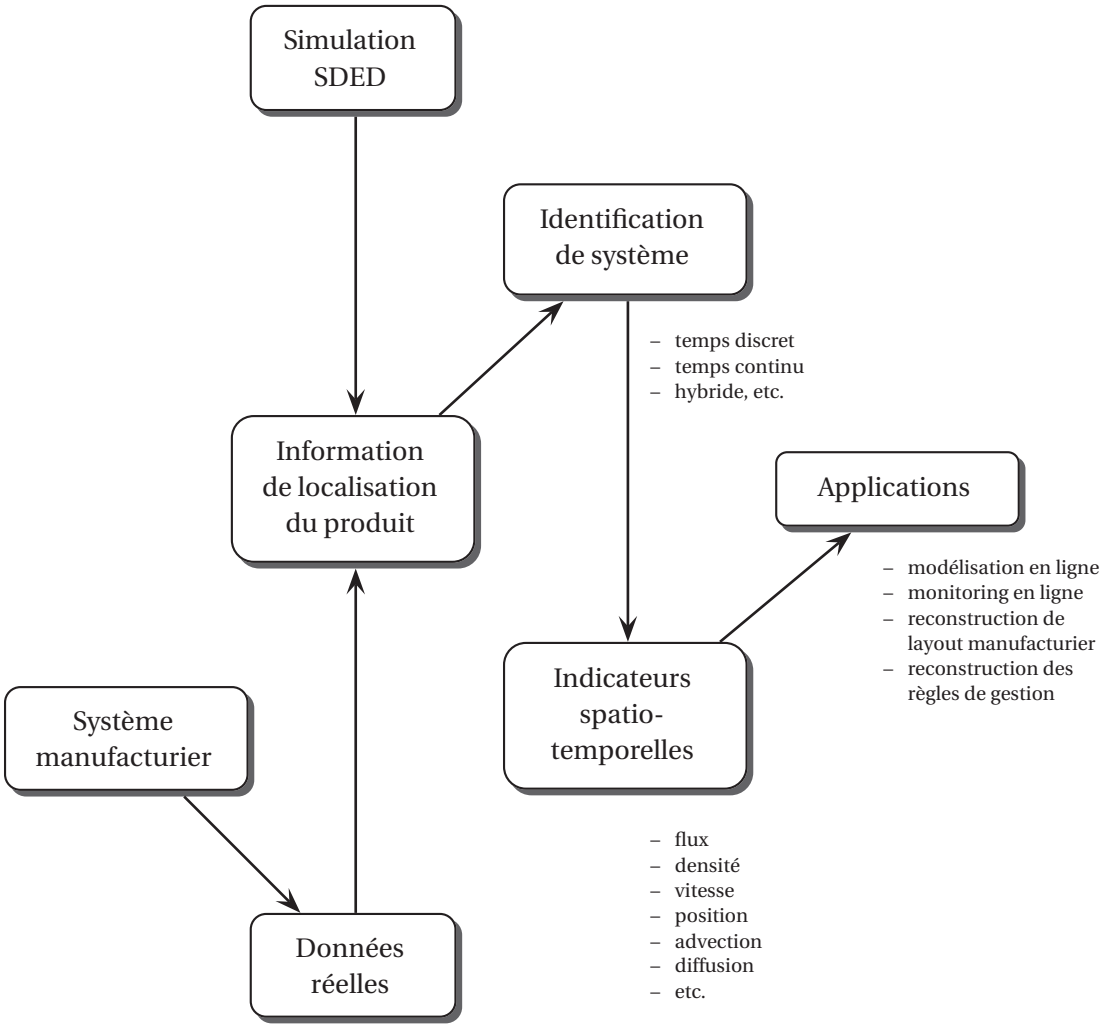


FIGURE 3.1 – Cheminement d'applications possibles.

Les systèmes manufacturiers mettent en œuvre des opérations pour lesquelles les dimensions spatiales et temporelles sont essentielles. L'un des meilleurs éléments du système apte à fournir des données spatio-temporelles est sans aucun doute le «composant - produit». Il peut être à la base de la génération d'un modèle utilisant la notion de trajectoire. Ces «composants - produits» constituent les atomes des produits finis. Pour ce faire, une exigence importante est liée à l'instrumentation des systèmes manufacturiers afin d'obtenir la localisation et le suivi des pièces. Nous ne nous sommes pas focalisés dans ce travail sur l'étude liée au challenge de localisation et du suivi, ni à sa mise en œuvre. Nous considérerons ici, comme nous l'avons expliqué dans l'introduction générale, que toutes

les informations de localisation des pièces sont disponibles, et constituent un flux de données de localisation.

L'accès aux flux de données de localisation, doit permettre d'identifier le système dont le modèle (et sa structure) doit appartenir à la classe des modèles des systèmes manufacturiers. Les modèles de type «boîte noire» ne sont pas, à notre sens, éligibles dans ce cas. En effet, ils ne permettent pas un accès et des modifications de leur structure [1], comme par exemple lors de l'utilisation des réseaux de neurones ou de la programmation génétique comme approximateurs. Cette contrainte peut être qualifiée de dure, mais elle est nécessaire. En effet, le modèle que nous prévoyons d'identifier est un modèle de simulation, qui doit être opératoire et prêt pour une utilisation postérieure, par exemple dans des scénarios de type "What If" où doivent pouvoir être testés des changements structurels du modèle. Il est indispensable d'assurer une correspondance entre la structure du modèle généré et la structure du système réel. Une autre raison peut également justifier ce choix. Elle est liée à la nécessaire adaptation du modèle de simulation aux changements de l'environnement, perçus par le biais des flux de données de localisation. Les méthodes d'identification à base de machines d'apprentissage et utilisant des données du système réel ne sont pas utiles si elles ont été conçues pour fonctionner hors ligne (ce qui est généralement le cas).

Les argumentaires précédents nous ont conduit à faire le choix du modèle classique «client-serveur», nommé «buffer-machine» ou «buffer-opération» dans notre contexte particulier. Ce modèle est très simple à comprendre et largement utilisé pour la simulation des systèmes manufacturiers. Ce choix consiste donc à insérer une connaissance *a priori* (le «buffer-opération») comme un élément structurel de base dans le modèle qui sera généré.

De manière graphique, chaque «buffer-opération» sera modélisé comme un nœud sur un réseau de transport. Pour identifier le modèle de simulation à partir du flux de données de localisation nous devons d'abord estimer le réseau de transport (layout), puis l'emplacement précis des nœuds «buffer-opération». Il s'agit d'un problème de séparation, où nous avons besoin de construire des indicateurs spatio-temporels pour classer les données appartenant à la classe "nœud" ou à la classe "voie de transport". La séparation des principaux comportements est un concept clé pour développer le modèle. La génération du layout et des nœuds du système de fabrication permet l'identification de la structure statique du modèle de simulation.

La classification des trajectoires constitue le second point clé de notre proposition. Celle-ci consiste

1. l'accès aux composantes du système : serveurs, machines, gammes, etc.

à regrouper les trajectoires observées et jugées comme similaires, pour au final associer chaque groupe de trajectoires à un type de produit. Les trajectoires sont ensuite fractionnées en routes, représentées graphiquement par un ensemble d'arcs reliant les nœuds, et reliant au final les sources des produits à leurs sorties du système. Deux produits appartiennent à un même "type de produit" s'ils ont un comportement spatio-temporel similaire, que cette trajectoire soit simple (le produit ne possède qu'un seul composant) ou composée (ensemble des trajectoires de chacun des composants du produit). La similarité est liée à la distance spatio-temporelle entre les trajectoires qui doit être assez petite.

Une fois les étapes précédentes effectuées, (layout, routes des produits), et pour prendre en compte les évolutions des flux de données de localisation dans le temps, il est nécessaire de maintenir le modèle de simulation. Ceci constitue la garantie que le modèle est le reflet permanent du système réel et qu'il peut être qualifié d'adaptatif. Comment faire en sorte de distinguer des événements rares par rapport à des changements plus profonds ou structurels ? Comment introduire dans le modèle l'arrivée ou la disparition d'une nouvelle ressource, d'un nouveau produit avec une nouvelle gamme de fabrication ? Comment adapter dynamiquement le modèle à des modifications d'emplacement des ressources ? La mise à jour du modèle est un processus délicat. En effet les interactions entre les différents éléments du modèle (trajectoires, chemin, route produit, ...), obtenus dans les étapes précédentes, sont nombreuses et la modification de l'un de ces éléments impacte forcement les autres. La section 3.5 de ce chapitre présentera les solutions proposées pour répondre à toutes ces questions.

Cette introduction à ce chapitre nous a permis de décrire de façon générale la méthodologie proposée ainsi que les problèmes posés. La Figure 3.2 reprend de manière synthétique l'ensemble des différents points présentés jusqu'à présent de manière littérale. Chaque étape représentée sur ce schéma va faire l'objet d'un paragraphe particulier. Nous nous référerons à ce schéma tout au long de ce chapitre.

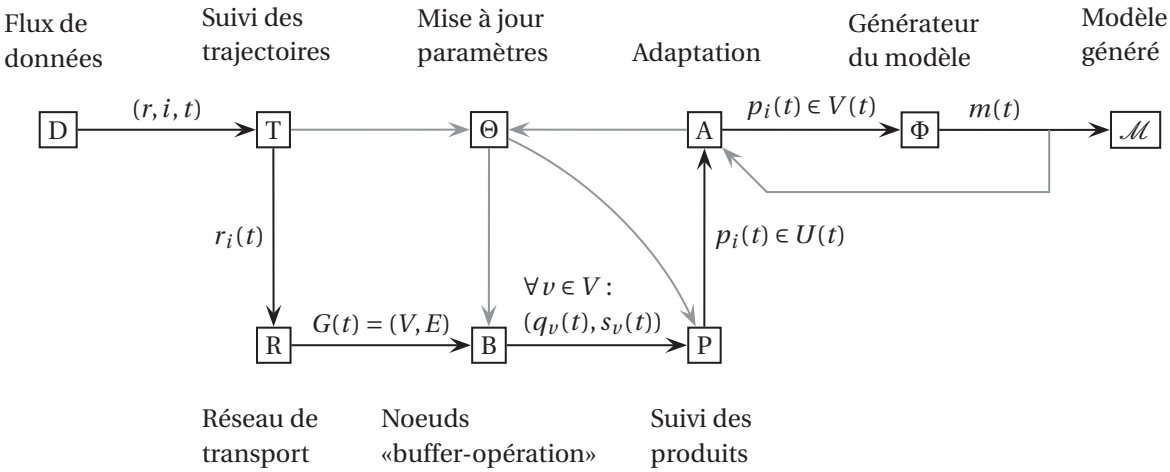


FIGURE 3.2 – Principe théorique de la mise en oeuvre d'un générateur de modèle de simulation conduit par un flux de données de localisation. Les variables et paramètres de ce schéma sont définis dans les paragraphes suivantes.

3.2 Du produit à la trajectoire dans le contexte des systémes dynamiques à événements discrets

3.2.1 Système manufacturier et SDED

LA DYNAMIQUE événementielle des processus manufacturiers est le résultat d'une planification rationnelle qui intègre des *étapes* et des *règles* de production. Les étapes représentent les opérations sur les produits, et les règles représentent les contraintes conduisant les matières primaires au produit fini par un réseau d'opérations. Chaque opération est associée à une ressource de capacité limitée. Une file d'attente ou buffer apparait à l'instant où un ou plusieurs produits sont dans l'attente d'une ressource.

Ce type d'activités dans les systèmes manufacturiers peut être modélisé comme un Système Dynamique à Événements Discrets (SDED ou DEDS en anglais). Les SDED sont des systèmes dynamiques asynchrones où les transitions entre états sont déclenchées par l'occurrence d'événements dans le système.

Il existe différents formalismes pour la modélisation des SDED, dont la définition suivante peut être donnée :

Définition 1 (Systèmes Dynamiques à Évènements Discrets) *Un Système Dynamique à Évènements Discrets (SDED), est une 6-tuple* $(X, X_0, A, B, \phi, \eta)$*, ou* X *est l'ensemble fini d'états,* $X_0 \subseteq X$ *est l'ensemble des états initiaux,* A *est l'ensemble des événements possibles.* B *est l'ensemble des sorties possibles. La*

fonction de transition d'états est $\phi : X \times A \rightarrow X$, *la fonction de transition des sorties est* $\eta : X \times A \rightarrow B$.

La structure générale à tous les formalismes de modélisation des SDED est la suivante :

- Définition de l'espace d'états. Catégorisation d'un ensemble d'entrées A, de sorties B et d'états internes X.
- Définition des règles d'évolution (changement d'états) : Prochain état $\phi : X \times A \rightarrow X$. Sortie $\eta : X \times A \rightarrow B$. Nous pouvons unifier ces règles dans une fonction : $F : X \times A \rightarrow X \times B$ où $F(x,a) \mapsto (\phi(x,a), \eta(x,a))$.
- Définition d'une structure commune à l'espace d'états et aux règles d'évolution (graphe du système ou bien système algébrique, généralement équivalents).

Les principales caractéristiques des SDED sont :

- Parallélisme dans la génération des évènements.
- Synchronisation des événements pour transitions.
- Concurrence dans l'utilisation des ressources.

Nous observons, dans le cadre de cette étude, un système manufacturier par son flux de données de localisation des objets manufacturiers : les «composants - produit». Ces données sont codifiées par la fonction $r_i : \mathbb{R}_+ \rightarrow \mathbb{R}^2,\ i \in \mathbb{N}$ où chaque donnée $r_i(t)$ représente une donnée de localisation à l'instant $t \in \mathbb{R}_+$, liée à une matière première d'identifiant i unique attribué avant son entrée dans le système. En tant que source d'entrée, les données spatio-temporelles $r(t)$ sont considérées comme exogènes au système et ne sont pas modélisées.

L'intérêt pour nous, d'avoir rappelé brièvement ici par le biais des SDED, les différents phénomènes pouvant se produire dans un système manufacturier, est lié aux éléments structurels et génériques à déployer dans le modèle et devant être à même de représenter ces dynamiques et comportements particuliers.

3.2.2 Le produit comme composition-décomposition des opérations

Lors du Chapitre 1 de cette thèse nous avons parlé du produit ou «composant produit» comme élément particulier d'intérêt pour nos travaux. Nous revenons ici sur ce concept de produit en précisant de manière plus formelle les notions de composition et décomposition qui lui sont associées. Dans ce travail nous considérons le «produit» comme étant un (ou des) objet en cours d'élaboration, depuis sa naissance à l'entrée du système, jusqu'à sa disparition à sa sortie. Des objets, composites ou non, peuvent être assemblés : le résultat de l'assemblage est un agglomérat d'objets

nommé produit. Un produit peut être désassemblé : ce processus générant ainsi des objets plus élémentaires.

Un produit type dans un processus batch ou flow-shop est conçu comme le résultat de $n \in \mathbb{N}$ étapes de production matérialisées par des opérations $f_1, f_2, \ldots, f_n$, et dans chaque étape un produit intermédiaire $p_i, i = 1, 2, \ldots, n$ est créé. La matière première, exogène, est désignée par p_0.

Le produit peut être décrit par l'exécution séquentielle des opérations

$$p_1 := f_1(p_0),\ p_2 := f_2(p_1),\ \ldots,\ p_n := f_n(p_{n-1}) \tag{3.1}$$

Cette séquence d'opérations correspond au processus de transformation (ou gamme)du produit.

De manière générale : $p^b := \mathbf{f}_n!(p_0)$, ou $\mathbf{f}_n! = f_n f_{n-1} f_{n-2} \cdots f_2 f_1$, est la composition des n fonctions.

Si nous ajoutons une composante exogène e_i dans chaque étape i nous pouvons redéfinir le produit intermédiaire comme :

$$p_i = \begin{cases} p_0 & \text{if } i = 0 \\ f_i(p_{i-1} \cup e_i) & \text{if } i > 0 \end{cases} \tag{3.2}$$

Pour obtenir un modèle général du produit en considérant des processus de type job-shop nous proposons une définition universelle du produit comme l'aboutissement d'un système entrée-sortie :

Définition 2 (Produit comme système entrée-sortie) *Soit $P_0 \subset \mathscr{P}$ l'ensemble exogène des composants basiques et indivisibles de $\mathscr{P}$. Soit $\mathscr{P} = \{P_i\}$ une famille d'ensembles de produits : composantes basiques, intermédiaires et produit finaux. Elle représente l'ensemble des évolutions possibles des composants basiques dans le temps par le biais des opérations. Soit $\mathscr{F} = \{f_j\}$ une famille des fonctions représentant les opérations telles que :*

1. *Si $f \in \mathscr{F}$, alors il existe des ensembles $P_a, P_b \in \mathscr{P}$ tels que* $\text{domain}(f) \subseteq P_a$ *et* $\text{range}(f) \subseteq P_b$ *(i.e pour chaque opération, les entrées et sorties sont contenues dans l'ensemble $\mathscr{P}$) .*
2. *Si $P_1, P_2, \ldots, P_n \in \mathscr{P}$, alors le produit cartésien $P_1 \times P_2 \times \cdots \times P_n \in \mathscr{P}$ est un sous-ensemble des produits de $\mathscr{P}$ (il existe un élément de $\mathscr{P}$ composé par les éléments du sous ensemble original).*
3. *Chaque ensemble $P \in \mathscr{P}$ contient soit le* $\text{domain}(f)$ *ou le* $\text{range}(f)$*, avec $f \in \mathscr{F}$ ou sinon P est un facteur du* $\text{domain}(f)$*, $f \in \mathscr{F}$ (i.e chaque élément de $\mathscr{P}$ doit être lié à une fonction (opération) sur $\mathscr{F}$, soit comme entrée, soit comme sortie, ou au minimum comme un composant d'entrée).*

Le tuple $p = (\mathscr{P}, P_0, \mathscr{F}, F)$ est un produit avec F une fonction qui pourra être construite à partir des fonctions canoniques $f \in \mathscr{F}$, tels que si $A_1, A_2, \ldots A_k \in P_0$ et $B \in \mathscr{P}$ l'ensemble de sortie :

$$F : A_1 \times A_2 \times \ldots A_k \rightarrow B$$

(i.e il existe une fonction résultante des compositions de fonctions élémentaires qui conduit des éléments basiques jusqu'au produit fini) .

La définition que nous avons proposée du produit montre une très forte dépendance vis a vis des opérations que celui-ci subit. Les notions de composition-décomposition tout comme les opérations de transformation élémentaires font partie intégrante de cette définition. Ces notions rejoignent les notions de nomenclature et de gamme des produits déjà introduites dans le Chapitre 1. Nous verrons que cette définition formelle et littérale du produit proposée ici peut trouver une représentation sous la forme d'un graphe orienté, ceci afin d'en faciliter la lecture et la compréhension. Cet aspect sera introduit dans les paragraphes suivants.

3.3 Éléments structurels du modèle

NOUS présentons dans ce paragraphe la construction de l'ensemble du réseau de transport (R à la Figure 3.2) et la situation (localisation) des files d'attente et serveur (ou buffer-opération, B à la Figure 3.2).

3.3.1 Réseau de transport

LE PREMIER élément structurel du modèle est le réseau de transport. Il est composé par une agrégation des trajectoires dans le temps.

Soit R_I une collection de points des trajectoires dans $\mathbb{R}^2$, dans l'intervalle $I = [t_a, t_b] \subset \mathbb{R}_+$. Les trajectoires sont liées à un phénomène de transport des particules (Equation 3.3), et en particulier à l'équation d'équilibre entre les variations temporelles de densité en un point et les variations spatiales du mouvement de ces particules (voir [42]) :

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \vec{J} = 0 \tag{3.3}$$

Dans le cas où la quantité des composants-produits est conservée dans le système (WIP (work-in-process) constant), il est possible d'écrire une équation de continuité pour $\rho(\vec{x}, t)$ où $\vec{J}$ est la densité

de flux par rapport à la densité de population ρ.

Alors une estimation de la fonction de densité $\rho(r)$, (r représentant la position) permet l'ajout de l'information sur la structure statique du système dans l'intervalle I.

Cette structure est un réseau de transport dans I, modélisé comme un graphe non dirigé. Elle peut être identifiée à partir des connections entre régions de hautes densités. La densité peut être mesurée par des fonctions de type :

$$p(R=r) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{|r-r(i)|}{h}\right) \tag{3.4}$$

Où h est la largeur de bande et $K(a)$ est une fonction de noyau. Par exemple, la fonction noyau proposée par Epanechnikov, est la suivante :

$$K(a) = \begin{cases} k(1-a^2) & \text{si } a^2 < 1 \\ 0 & \text{si non} \end{cases} \tag{3.5}$$

De façon similaire une estimation de la densité spatio-temporelle ρ peut être obtenue par :

$$\rho_{\epsilon,\tau}(\mathbf{r}_s, t_s) = \frac{1}{\pi\tau\epsilon^2} \sum_{\{(\mathbf{r},t)\in\mathscr{S}\,|\,t\in\mathscr{T}\}} \theta(\epsilon - \|\mathbf{r}-\mathbf{r}_s\|)\theta(\tau - |t-t_s|) \tag{3.6}$$

$\mathbf{r}_s \in k\mathbb{Z}^2,\ t_s \in \tau_d\mathbb{Z},\ \mathscr{T} \in t_s + \tau/2[-1,1],\ \mathscr{S} = \{(\mathbf{r}_k, t_k)\}_{k=1}^{N}$.

Avec θ la fonction de Heaviside :

$$\theta(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases} \tag{3.7}$$

Après un traitement des données, en considérant des régions de haute densité et ses connections, l'ensemble des trajectoires est décomposé en segments s_k entre des régions de haute densité :

$$R_I \mapsto \{s_1, s_2, s_3, \ldots\} \tag{3.8}$$

Dans chaque intersection $s_i \cap s_j \neq \varnothing$ un nœud $v \in V$ est défini. Si entre deux noeuds v_i, v_j il existe un segment s, alors nous ajoutons l'arc (v_i, v_j) à l'ensemble des arcs E. Finalement, si un noeud $v_i \in V$ est une extrêmité d'un segment s, mais que l'autre l'autre extrêmité n'existe pas dans V, nous ajoutons cette extrêmité manquante v_m à V et l'arc (v, v_m) à E. Le réseau de transport est alors construit comment une graphe non dirigé $G = (V, E)$. Les noeuds du graphe représentent les positions des

buffer-opération potentiels, et les arcs représentent l'ensemble des trajectoires suivies par l'ensemble des produits.

3.3.2 Noeud buffer-opération

LE SUIVI des trajectoires des objets dans le plan (x, y) dans le temps $(r_i(t))$ nous permet d'en déterminer la vitesse $\dot{r}_i(t)$: soit celle-ci est nulle, soit celle-ci est positive. Le produit est en mouvement ou il est arrêté : cette dernière propriété est le signe d'une attente. L'information $\dot{r}_i(t) = 0$ permet de confirmer la localisation des points particuliers à la base de la construction de notre modèle : les buffers-opération ou noeuds du graphe. Nous verrons au paragraphe suivant que pour chacun de ces points il est possible d'estimer différents paramètres particuliers permettant de les caractériser.

La densité des points ou zones de l'espace où les produits s'arrêtent souvent est bien sûr plus élevée que celles des autres régions.

3.4 Éléments structurels du modèle : trajectoire produit et paramétrage

Cheminement du produit L'élément central est ici le produit. Nous allons nous intéresser aux trajectoires individuelles de chaque produit tel qu'il a été défini précédemment. La trajectoire du produit peut être représentée par un graphe dirigé obtenu par l'union des routes d'intersection non nulle (les points d'intersection représentant des points de composition ou d'assemblage), et de destination commune. Une route est une séquence de noeuds $v_1, v_2, v_3, \ldots v_n$ connectés par des arcs $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$.

Dans la Définition 2, le produit est considéré comme l'ensemble des matières premières, produits intermédiaires et opérations subies, et l'ensemble de leur relations. Cette définition est directement liée à un graphe orienté représentant :

– Les matières premières par les noeuds racines.
– Les opérations (composition, décomposition, transformation) par les noeuds.
– Les transports (ou transitions entre opérations) par les arcs.
– La direction des flux par les arcs orientés (relations de précèdence entre opérations).
Le noeud terminal du graphe est particulier, il représente le produit final obtenu.

Cette représentation, par un graphe, du produit est utilisée pour différentes raisons et en particulier :

- A chaque noeud peut être défini un buffer-opération.
- A chaque type de produit peut être associé un graphe orienté de ce type, le système manufacturier complet pouvant alors être défini comme l'union de tous les graphes des produits.
- La structure des graphes permet de modéliser les relations temporelles (ou/et enchainements) entre les différentes opérations.
- La structure de graphe permet la mise en évidence des liaisons entre espace et temps, une position peut être précisée pour chaque noeud du graphe (serveur).
- L'évolution de la topologie du graphe dans le temps est le reflet observable de l'évolution du système manufacturier d'intérêt.

Enfin, le graphe permet d'organiser, de structurer, toutes les données et informations nécessaires pour la mise en oeuvre d'un simulateur du système.

Exemple Nous allons introduire ici sur la base d'un exemple simple d'un atelier à trois machines M_1, M_2 et M_3, et un type de produit P (Fig. 3.3) cette notion de composant et de produit. Dans cet exemple, le produit est composé de trois composantes a, b, c. L'obtention de P se fait, par l'agglomération progressive des composants initiaux (a,b,c) en produits intermédiaires (p_1,p_2). On peut représenter ce processus comme la séquence d'opérations suivantes :

$$\begin{aligned} a &\mapsto M_1(a) = p_1 \\ (p_1,b) &\mapsto M_2(p_1,b) = p_2 \\ (p_2,c) &\mapsto M_3(p_2,c) = P \end{aligned} \tag{3.9}$$

ou de façon plus synthétique par une composition de fonctions :

$$P = M_3\left(M_2\left(M_1(a),b\right),c\right) \tag{3.10}$$

Nous pouvons représenter les équations (3.9) et (3.10), comme un graphe $G=(V,E)$ (Fig. 3.3) , ou les sommets sont :

$$V = \{a,b,c,M_1,M_2,M_3,P\} \tag{3.11}$$

Les arcs sont :

$$E = \{(a,M_1),(M_1,p_1),(p_1,M_2),(b,M_2),(M_2,p_2),(p_2,M_3),(c,M_3),(M_3,P)\} \tag{3.12}$$

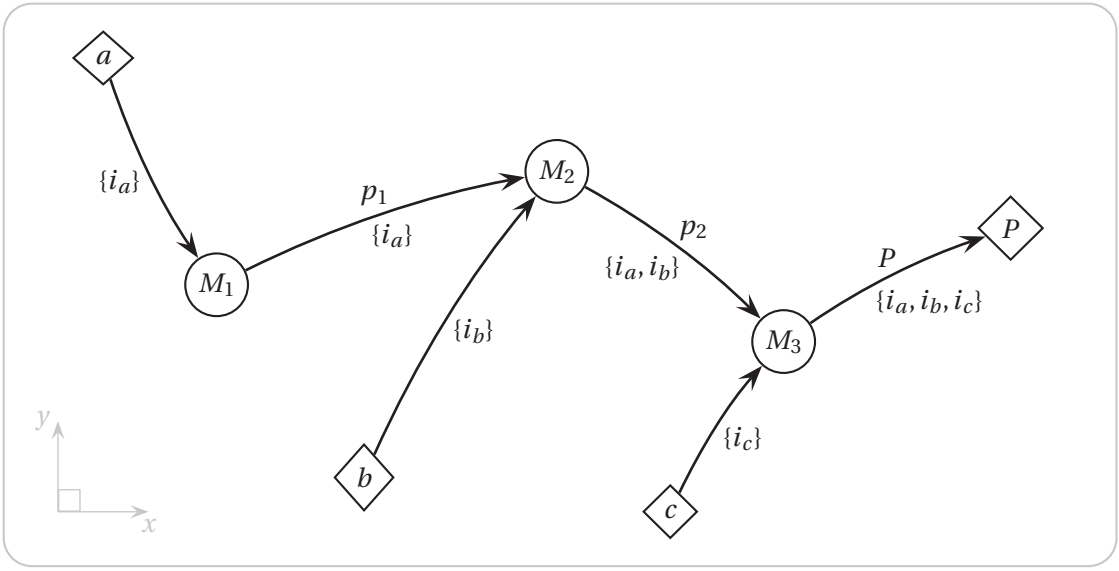


FIGURE 3.3 – Exemple du graphe orienté d'un produit P. Sur ce type de schéma les losanges représentent la naissance ou la disparition des objets et/ou produits (un losange avec une flèche sortante est naturellement un point de naissance, un losange avec une flèche entrante un point de disparition). Les cercles représentent les opérations. Sur chaque flèche sont montrées deux choses : le produit ou composante dans son état, et l'ensemble des identifiants du produit.

Sur la Figure 3.3 sont représentées deux choses différentes :

– la composition (ou nomenclature) du produit P, *(cf paragraphe précédent)*

– la localisation dans le plan (coordonnées cartésiennes x et y) des ressources nécessaires à l'élaboration de P, ainsi que le parcours des différents composants et sous-ensembles de P.

Files d'attente Le suivi de la trajectoire d'un objet dans le plan (x, y) et dans le temps $(r_i(t))$ permet de déterminer les points particuliers où ces objets s'arrêtent $(\dot{r}_i(t) = 0)$. En chacun de ces points il est possible d'obtenir un histogramme représentatif du temps de «service» et du temps d'«attente» par type de produit.

Cependant, avant la disparition d'un produit, aucune information sur sa composition n'est disponible. Les objets (initiaux et intermédiaires) ne fournissent aucune information quant à leur destination finale. C'est à la disparition du produit que sa composition finale est entièrement découverte (de par les mouvements joints de chacun de ses composants, Fig. 3.3). Il est alors possible de retracer l'ensemble des parcours de ses constituants jusqu'à leur naissance, et ainsi de mettre à jour les histogrammes des temps de service d'un constituant destiné à un produit final sur un point $\dot{r}_i(t) = 0$. Durant leur passage dans le système manufacturier, les objets suivent des trajectoires dans le plan (x, y). L'observation de ces trajectoires dans le temps nous permet de déterminer la vitesse $\dot{r}_i(t)$ de

l'objet. A chaque point où $\dot{r}_i(t) = 0$, il est possible d'associer un comportement de type file d'attente. (Fig. 3.4, et Fig. 3.5).

En chacun de ces points deux types d'événements peuvent se produire : l'arrivée ou le départ d'objet(s). Nous allons utiliser ces deux types d'événements pour le paramétrage du buffer-opération.

Nous définissons alors (3.6) :

- e_1 l'instant d'arrivée du premier objet constituant le produit.
- e_2 l'instant d'arrivée du dernier objet constituant le produit.
- s est l'instant commun de sortie de tous les objets constituants le produit.

Sur ces bases nous pouvons alors obtenir :

- O, la durée entre l'arrivée du premier objet et l'arrivée du dernier objet nécessaire à la constitution du produit.
- A, le temps d'attente du produit : durée définie entre le moment où toutes les pièces nécessaires à la constitution du produit sont arrivées et le moment réel de passage.
- T, le temps de service du produit au point $\dot{r}_i(t) = 0$.
- T_{Tot}, le temps total d'arrêt du produit au point $\dot{r}_i(t) = 0$.

On peut lier ces différentes variables par l'équation :

$$T_{Tot} = O + A + T \tag{3.13}$$

Pour calculer A il est nécessaire de connaître le temps de sortie du produit précédent s^{k-1} où $k-1$

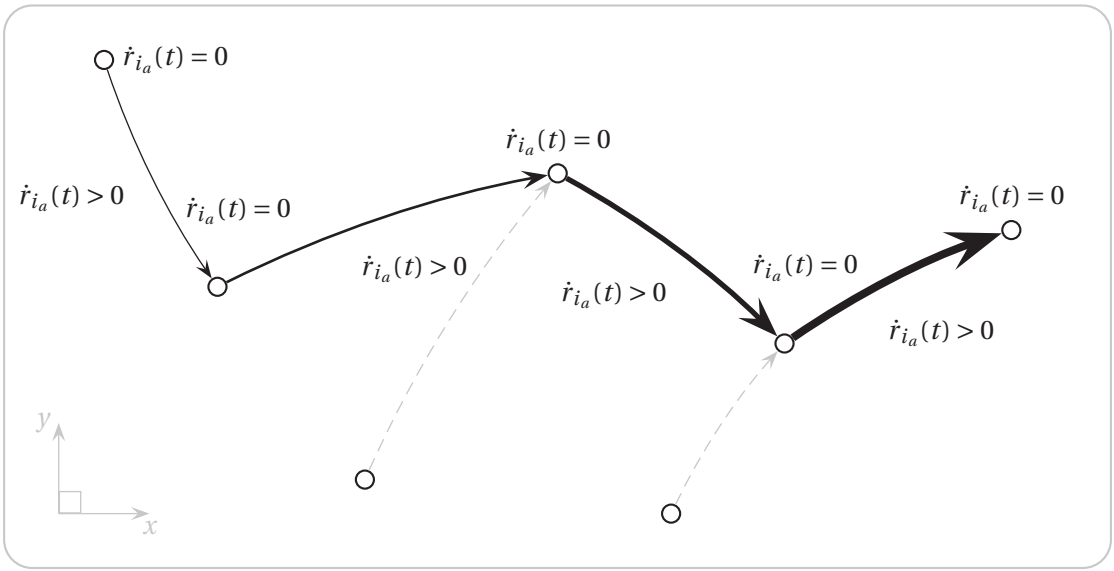


FIGURE 3.4 – Vitesses composante a

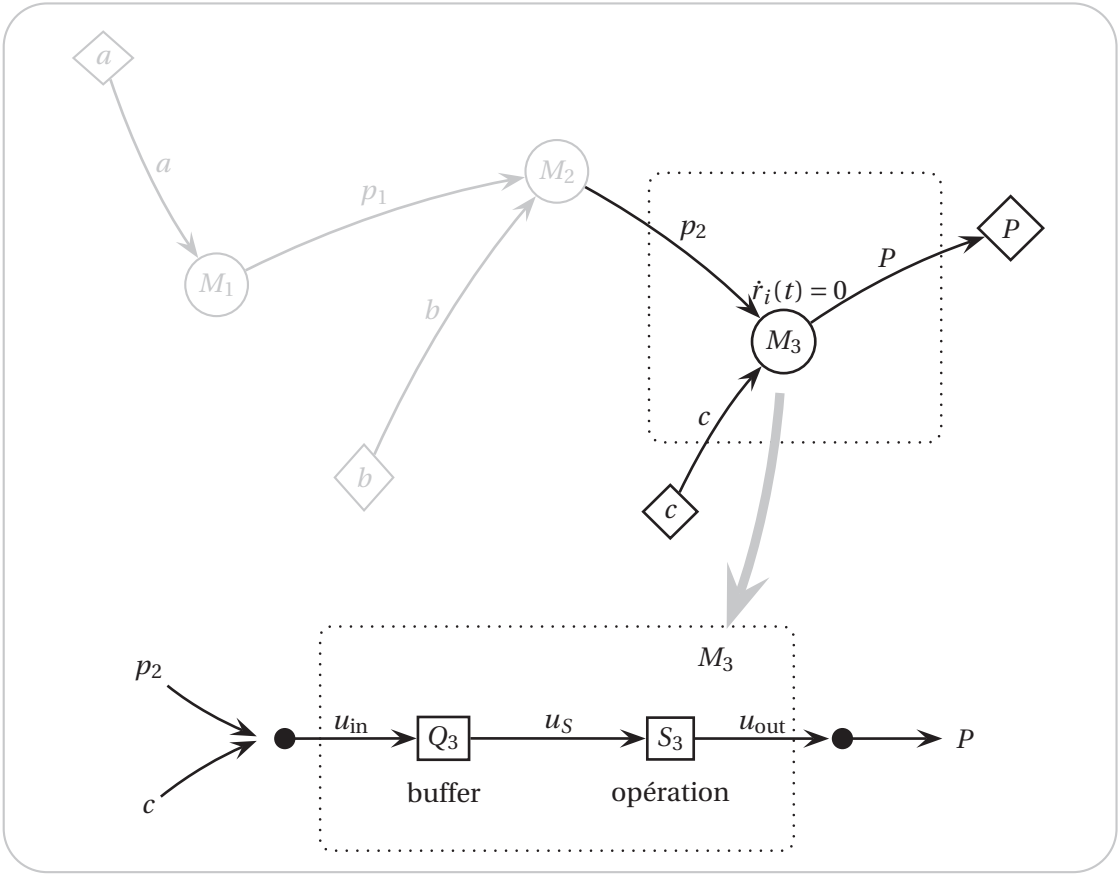


FIGURE 3.5 – Noeud buffer-opération. File d'attente et serveur.

représente le rang de sortie du produit. La règle de calcul de A est : si $e_2^k < s^{k-1}$ alors le temps d'attente est $s^{k-1} - e_2^k$, sinon le temps d'attente est 0. De manière plus formelle et en utilisant la fonction

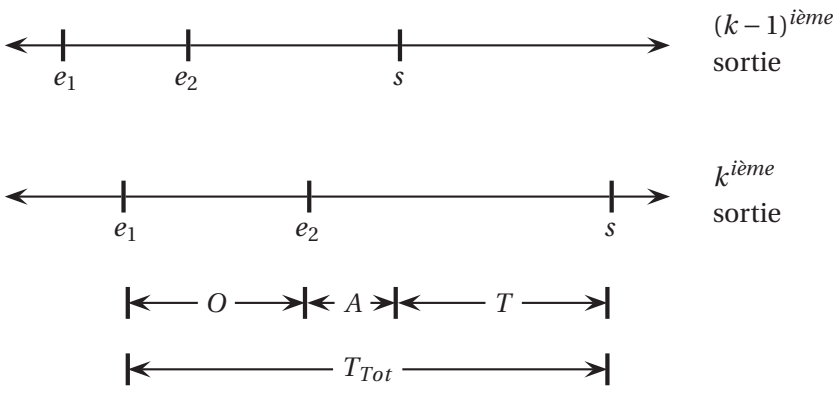


FIGURE 3.6 – Temps dans un point $v = 0$ avec congestion

d'Heaviside :

$$\theta(y) = \begin{cases} 1 & \text{si } y \geq 0 \\ 0 & \text{si } y < 0, \end{cases} \tag{3.14}$$

nous pouvons maintenant définir à chaque $k^{ième}$ sortie d'un produit les variables introduites jusqu'à présent :

$$\begin{aligned} y^k &= e_2^k - s^{k-1} \\ \alpha^k &= \theta(y^k) \\ \beta^k &= 1 - \alpha^k \\ T^k &= \alpha^k(s^k - e_2^k) + \beta^k(s^k - s^{k-1}) \\ A^k &= \beta^k(s^{k-1} - e_2^k) \\ O^k &= e_2^k - e_1^k \end{aligned} \tag{3.15}$$

à la condition où $s^0 = 0$.

Le triplet (T, A, O) possède toute l'information nécessaire à la caractérisation du point $\dot{r}_i(t) = 0$ comme buffer-opération. La proposition réalisée ici s'appuie sur le cas particulier où la file d'attente et le serveur se réduisent à un seul point de l'espace. De façon plus générale, ce point peut être considéré comme un «marqueur» de la zone dans laquelle les produits attendent et subissent une opération. Cette généralisation à la zone ne remet pas en cause toute l'analyse présentée ici, et ne pose pas de difficultés particulières.

La méthode proposée ici d'identification des buffer-opérations peut également générer dans le modèle des buffer-opérations sans lien avec le système réel d'intérêt. En effet certains noeuds du graphe peuvent être obtenus par l'observation de points particuliers de l'espace, où la vitesse des produits est nulle, sans que pour autant cela corresponde à une opération réelle effectuée sur le produit.

Ceci s'explique par le peu de connaissance que nous avons initialement introduit dans le générateur (ontologie minimaliste), il est donc ici impossible de différencier un point de l'espace où les produits s'arrêtent pour y subir une transformation, d'un point de l'espace où les produits s'arrêtent dans un buffer spontané. L'ajout de connaissances sur des dimensions de zones d'attente devant une ressource peut permettre de lever cette différence structurelle entre modèle et réalité. Les noeuds du graphe d'une même zone seraient alors "regroupés" sur un seul noeud.

3.5 Auto-adaptation du modèle

APRÉS avoir présenté nos propositions concernant la construction initiale du modèle, nous allons maintenant nous focaliser sur sa partie adaptation. Cette adaptation du modèle est importante, car elle permet de garantir que celui-ci est le reflet du système et de ces évolutions. Plus qu'une adaptation des paramètres du modèle, il s'agit ici de prendre en compte des évolutions structurelles du système d'intérêt. Ces évolutions peuvent concerner tout autant l'ajout ou la suppression de nouvelles ressources de production, leur indisponibilité temporaire, des modifications de gamme ou de nomenclature des produits. Il s'agira également pour nous de déterminer s'il s'agit réellement d'évolution du système ou d'événements rares. Dans les sections précédentes, nous avons montré comment nous représentons et codifions les informations des trajectoires $R = \{r_i(t), t > 0\}_{i \in I}$ par des graphes $G_R(t) = (V, E)$, et les buffer-opération pour des noeuds $v \in V$. Ces graphes et les buffer-opérations représentent la structure du modèle de simulation. C'est donc cette structure qu'il va nous falloir adapter.

L'évolution conjointe du modèle et du système réel doit être synchronisée. Cette synchronisation est l'un des objectifs de conception du générateur de modèle. Nous allons donc adjoindre au générateur, un système adaptatif. Quoiqu'il en soit l'adaptation sera toujours basée sur la notion de trajectoire des produits : c'est les modifications de ces trajectoires qui permettront l'évolution structurelle et/ou paramétrique du modèle.

Nous présentons donc ici dans un premier temps les principes généraux des systèmes adaptatifs, puis de manière plus détaillée le schéma choisi pour notre générateur.

Les Figures 3.7 et 3.8 présentent deux schémas de contrôle adaptatif : le contrôle adaptatif direct (ou "implicite"), et le contrôle adaptatif indirect ou "explicite" (voir de Wit [57]).

Dans le contrôle adaptatif direct (ou "implicite", Figure 3.7) le but est d'obtenir une loi de contrôle du système. La loi de contrôle du système est fonction des états mesurables du système et de ses performances. Le modèle du système est alimenté par les mêmes entrées que le système. Les écarts mesurés entre sorties du modèle et sorties du système réel permettent, via un algorithme d'adaptation des paramètres, de modifier le paramétrage du contrôleur, avec pour objectif la minimisation des différences observées en sortie.

Dans le contrôle adaptatif indirect (ou "explicit", Figure 3.8) le système (plant) est estimé "explicitement" par une procédure d'estimation en ligne. Généralement cette estimation est un modèle de structure fixe et de paramètres inconnus. L'estimation du système permet de faire la mise à jour des

paramètres de contrôle dans la partie “control design”.

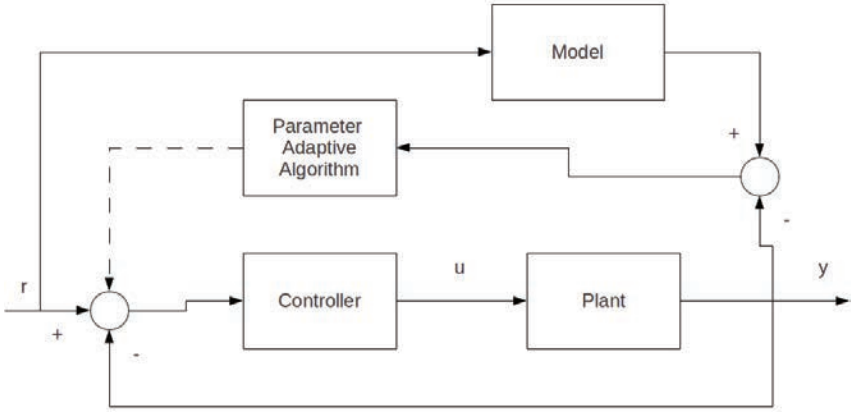


FIGURE 3.7 – Schéma de contrôle adaptatif direct

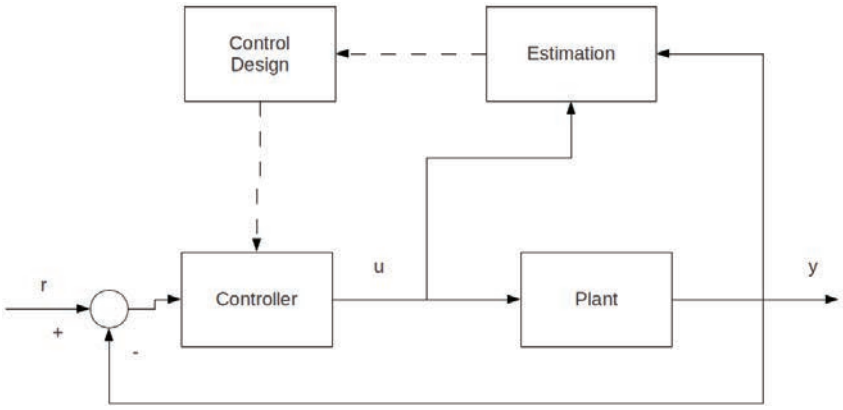


FIGURE 3.8 – Schéma de contrôle adaptatif indirect

Le principe d’adaptation retenu pour notre générateur est proche de celui utilisé dans le contrôle adaptatif indirect. Ainsi, de manière plus formelle :

Soit $R(t_a, b)$ le nouveau comportement :

$$R(t_a, b) = \{\, r_i(t), t \in t_a + [0, b] \,\}_{i \in I} \tag{3.16}$$

Si ce nouveau comportement est lié à un changement structurel, nous devons adapter la structure du modèle. Par exemple la mise à jour du réseau du transport peut être définie par :

$$\begin{aligned} H_R &\leftarrow H(R(t_a, b)) \\ G_R &\leftarrow f(G_R, H_R) \end{aligned} \tag{3.17}$$

Avec H_R la collection de paramètres de G_R.

Une solution commune consiste à comparer la fréquence relative du nouveau phénomène inconnu p_i par rapport à la fréquence des événements sur les éléments structurels déjà définis (ensemble P_j). Ceci revient à adapter le générateur selon un modèle statistique :

$$i(p_i, P_j) = \begin{cases} \emptyset, & p_i < E(p), p \in P_j \\ \tilde{i}, & p_i \sim E(p), p \in P_j \end{cases} \tag{3.18}$$

Une implémentation simple consiste à collecter et garder la trace de tous les éléments, et pour les trajectoires non reconnues les plus récurrentes (de haute et moyenne probabilité sur P_j) de les considérer comme des événements devant modifier le modèle. La définition du seuil de récurrence est l'un des paramètres de l'adaptateur. Ce type de procédure demande aussi une grande capacité de stockage pour l'ensemble des informations collectées.

Une solution plus élaborée est de limiter à une fenêtre de temps $T_w \subset \mathbb{R}_+$ l'observation des récurrences de ces trajectoires. Dans cette solution, un nouveau comportement i est ajouté sur P_j si la fréquence d'apparition du phénomène i est élevée dans T_w. Dans le cas contraire, toute l'information collectée sur i est éliminée. La taille de la fenêtre temporelle est alors un paramètres de l'adaptateur.

Validité des structures du modèle Le modèle adaptatif $M(t)$ est composé par l'union de deux structures : une structure valide par rapport a la fenêtre temporelle de validation $V(T_w, t)$, et une structure non vérifiée qui couvre les changements dynamiques dans une fenêtre temporelle plus petite que la fenêtre temporelle de validation $U(T < T_w, t)$:

$$M(t) = V(T_w, t) \cup U(T < T_w, t) \tag{3.19}$$

Une structure valide est un pattern spatio temporel qui est récurrent dans la fenêtre de validation T_w. Dans la pratique, pour chaque élément i du modèle estimé à partir des données, nous incorporons une fonction de validation $v(i, t)$ qui vérifie si le pattern est utilisé ou pas.

$$t, i(t) \mapsto v(i, t) \tag{3.20}$$

Si le pattern est récurrent, la fonction peut arriver rapidement à sa valeur maximale, mais si le pattern n'est pas récurrent, la fonction de validité tend vers zéro.

Quand la valeur zéro est obtenue pour, $v(i, t)$, l'élément i du modèle est éliminé (le pattern est oublié).

Au final, la Figure 3.9 présente un schéma de principe général de l'usage possible du générateur et du modèle généré.

Sur ce schéma bloc deux usages du modèle de simulation sont représentées : en ligne et hors ligne. L'usage hors ligne du modèle de simulation permettant l'essai de scénarios de type «what if», alors que l'usage en ligne peut permettre l'adaptation du générateur ou l'optimisation du fonctionnement global du système.

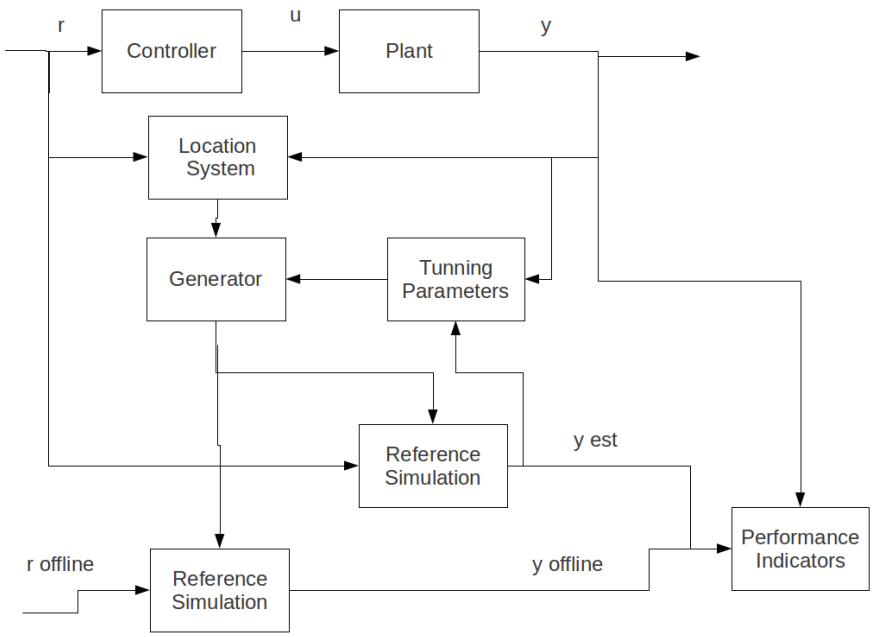


FIGURE 3.9 – Boucle de rétroactions dans le système manufacturier.

3.6 Conclusion : modèle formel du générateur

CE chapitre nous a permis de présenter nos propositions. Sur la base d'une information réduite (les trajectoires produit), nous avons montré comment il était possible de construire un modèle de simulation des flux. La méthode formelle proposée ici utilise une représentation par des graphes des trajectoires des produits. La seule connaissance introduite pour la construction du modèle est liée au choix réalisé de modéliser le système manufacturier sous la forme d'un réseau de file d'attentes. Notre méta-modèle est donc basé sur une ontologie minimaliste. Toutes les autres informations récoltées sont liées au produit et ses mouvements dans le système observé. Le produit (et/ou ses composants) jouent donc un rôle central dans nos propositions. Celles-ci permettent l'obtention

d'un modèle relativement riche, puisqu'il est possible de reconstruire non seulement le layout du système de production et les parcours suivis par les produits, mais également les nomenclatures et les gammes de fabrication suivies. La dynamique d'évolution des produits sur le système peut également être caractérisée (Temps d'attentes, temps de service). La dynamique d'évolution du système est pour sa part la résultante de l'observation et de la prise en compte des changements structurels subis par le système réel par les mécanismes d'adaptation proposés au niveau du générateur. La méthode proposée ne peut naturellement être mise en œuvre que si le système réel pré-existe au modèle de simulation (et pas l'inverse !). Autrement dit la méthode proposée et le modèle obtenu ne peuvent être utilisés qu'en ré-engineering (pour améliorer) ou en exploitation (pour contrôler), mais pas en phase de conception d'un nouveau système. Nous avons proposé dans ce chapitre un schéma général d'usage du générateur et du modèle de simulation généré. L'automatisation de la génération du modèle par le principe proposé réduit l'intervention humaine en général assez lourde dans ces phases d'élaboration du modèle. Le temps de collecte des données devient théoriquement nul, et la maintenance du modèle est assurée par les principes d'auto-adaptation mis en œuvre. Le chapitre 4 va nous permettre de présenter la mise en œuvre effective du générateur.

Chapitre 4

Mise en œuvre système générateur

Telles seraient les phases successives de l'image :

– *elle est le reflet d'une réalité profonde,*

– *elle masque et dénature une réalité profonde,*

– *elle masque l'absence d'une réalité profonde,*

– *elle est sans rapport à quelque réalité que ce soit : elle est son propre simulacre pur.*

Dans le premier cas, l'image est une bonne apparence – la représentation est de l'ordre du sacrement. Dans le second, elle est une mauvaise apparence – de l'ordre du maléfice. Dans le troisième, elle joue à être une apparence – elle est de l'ordre du sortilège. Dans le quatrième, elle n'est plus du tout de l'ordre de l'apparence, mais de la simulation

(L'allégorie de la caverne de Platon dans *Simulacres et simulation*, Jean Baudrillard [14]).

4.1 Introduction

SUR LA BASE des propositions effectuées au chapitre précédent, nous allons dans ce chapitre présenter l'implémentation réalisée. La Figure 4.1 présente un synoptique général du prototype réalisé. La **construction progressive du modèle** est réalisée à chaque nouvel événement lié au flux entrant, à savoir une nouvelle information du type $\{r, i, t\}$. Cette construction progressive est la partie la plus conséquente et importante en terme de développement.

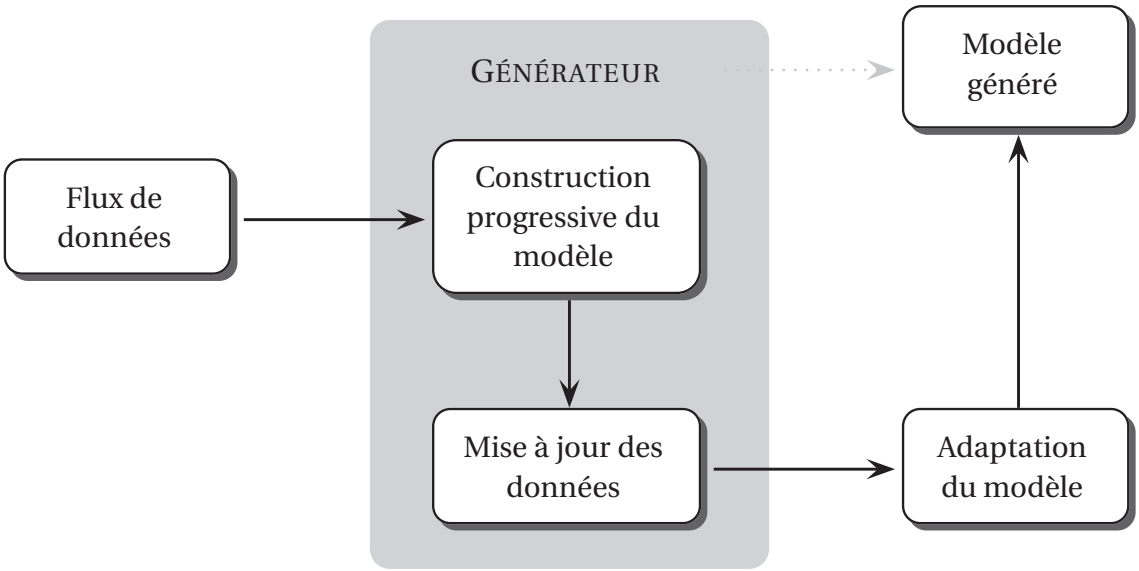


FIGURE 4.1 – Synoptique général du prototype réalisé.

A la disparition du produit du système (nous proposerons une méthode permettant de déceler cette « mort » du produit), il est alors possible de mettre à jour les données de celui-ci et celles du modèle. Il faut en effet attendre la sortie du produit du système pour découvrir si oui ou non ce type de produit existait déjà et donc le créer ou mettre à jour les données le concernant. La **mise à jour des données** produit participe à l'adaptation du modèle. Cette adaptation est complétée par un mécanisme simple visant à observer le côté éphémère ou durable des éléments du modèle (produits, et donc buffer opération et informations liées).

Dans ce chapitre, nous allons dans un premier temps présenter, sous la forme de diagrammes de flux, chacune des composantes apparaissant à la Figure 4.1. Dans un second temps et de manière plus formelle, nous présenterons les algorithmes de chacun des composants logiciels qui ont été codés et les diagrammes de classes UML nécessaires à l'implémentation.

4.2 Diagrammes de flux

LE PREMIER Le premier diagramme de flux présenté Figure 4.2 est le plus conséquent. Il correspond la partie **construction progressive du modèle** présenté Figure 4.1. Le déclenchement de l'exécution de celui-ci est lié à l'arrivée d'une nouvelle donnée dans le flux entrant. Les branches terminales de ce diagramme représentent simplement la mise en attente de l'arrivée de nouvelles données. Au lancement de l'exécution de ce module, les différents ensembles (ou structures d'accueil des données) évoqués et les informations liées sont vides. Elles sont donc dans un premier temps créées puis progressivement mises à jour durant l'exécution itérative de cette fonction. Ainsi, les buffers-opérations, l'exemplaire produit ayant déclenché l'événement, et de manière indirecte sa gamme et sa nomenclature, sont créés et caractérisés par le calcul de l'ensemble des informations spatio-temporelles le concernant (temps d'attente, temps de service, position des serveurs). Ceci permet la construction progressive du graphe « produit » tels que présenté au chapitre 3.

Le second diagramme de flux correspond pour sa part à la **mise à jour des données** du modèle, une fois l'exemplaire produit étant considéré comme définitivement sorti du système (Figure 4.3). Pour notre implémentation, un produit est considéré comme sorti du système si aucun événement lié à ce produit n'apparaît plus au-delà d'une certaine durée (c'est un paramètre de notre générateur). Nous l'avons empiriquement fixé comme étant égal à la moyenne du temps de vie total de l'ensemble des produits. Deux situations peuvent se produire, soit l'exemplaire produit appartient à une classe produit déjà référencée, soit le type de produit est nouveau. La discrimination de ces deux cas s'effectue sur la base d'une mesure de similarité de la distance spatio-temporelle entre les trajectoires des types de produits déjà connus et de l'exemplaire sortant. Dans le cas où le produit est nouveau, un nouveau type est alors créé, et ses caractéristiques enregistrées. Dans le cas contraire, les données récoltées lors de son passage dans le module « construction progressive du modèle » sont fusionnées avec les informations déjà présentes. Elles contribuent donc à l'évolution des paramètres du modèle. La fréquence d'exécution du module «Mise à jour des données du modèle» est plus faible que la fréquence d'apparition des événements dans le flux afin de ne pas ralentir inutilement l'exécution de l'ensemble.

Enfin le troisième et dernier diagramme proposé traite de **l'adaptation du modèle** aux évolutions « long terme » des informations récoltées (Figure 4.4).

A chaque type de produit est associé une « fonction de validité ». Cette fonction reflète quantitativement la récurrence de l'apparition du type de ce produit dans le temps. Quand cette fonction de

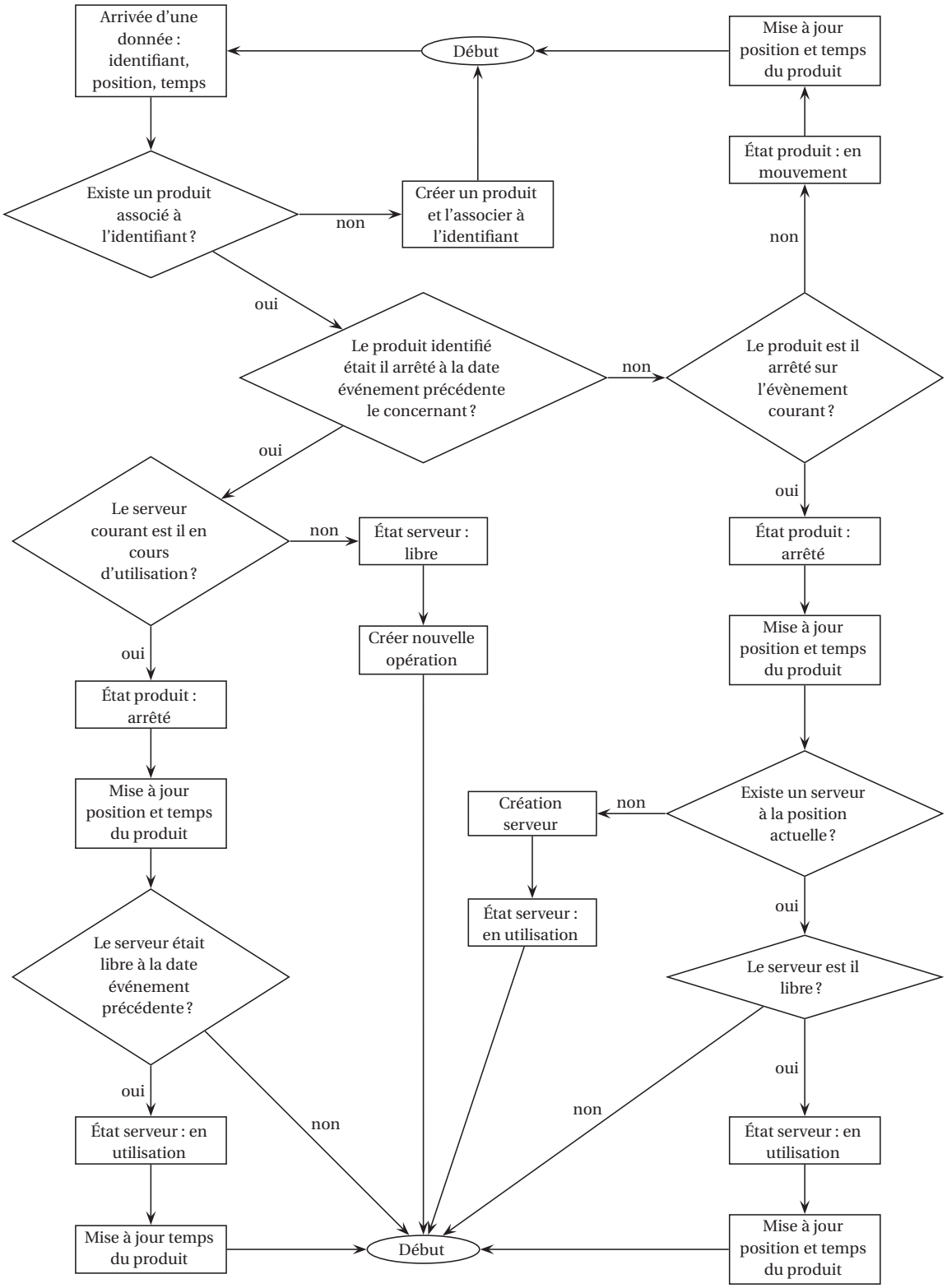


FIGURE 4.2 – Construction progressive du modèle.

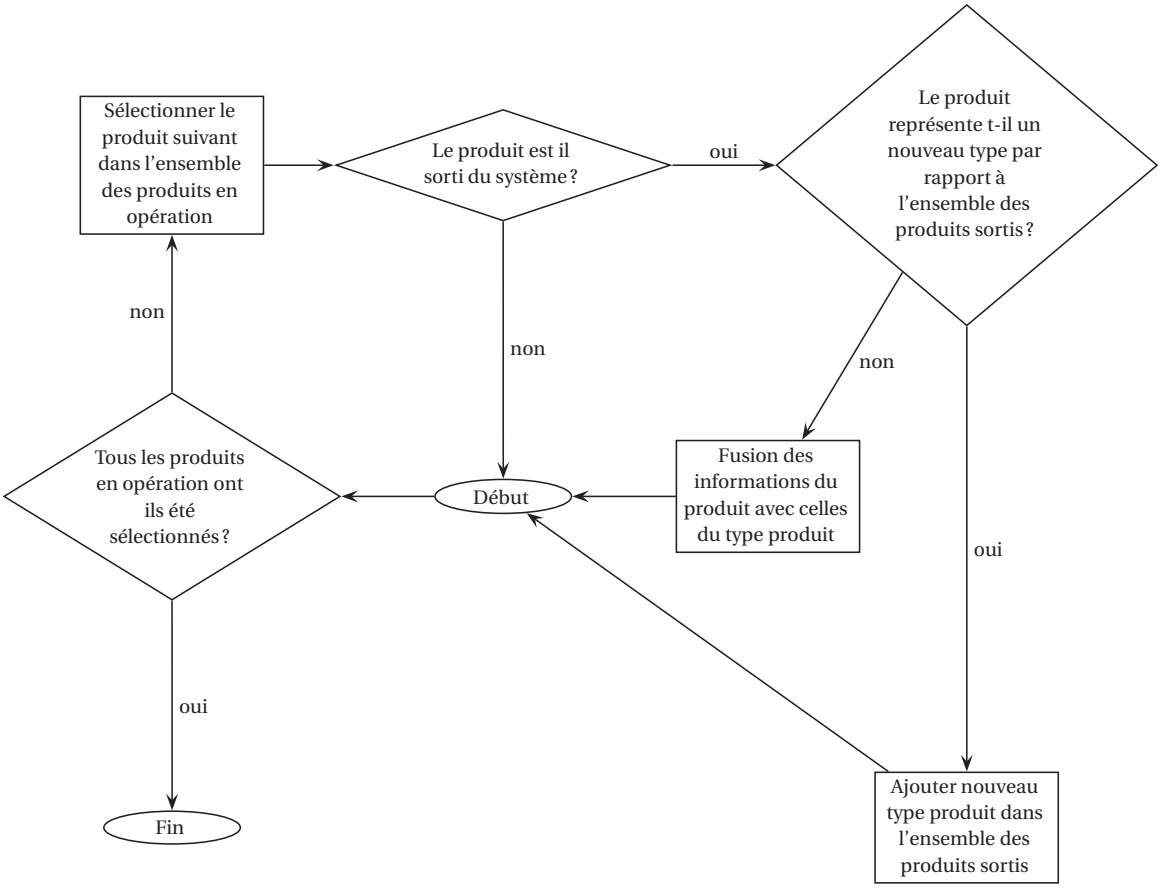


FIGURE 4.3 – Adaptation. Mise à jour des données du modèle.

validité arrive à la valeur zéro, cela signifie que le type de produit n'est plus apparu sur le système depuis longtemps (la fenêtre temporelle est là encore un paramètre du générateur) : le produit est donc amené à disparaître du modèle, tout comme les serveurs uniquement liés à celui-ci. Cette fonction est incrémentée lorsqu'à l'exécution de ce module, on peut constater qu'un exemplaire de ce type de produit était présent sur le système, sinon elle est décrémentée. Tout comme le module précédent, la fréquence d'exécution de ce module est plus faible que la fréquence d'apparition des événements dans le flux, et ce pour les mêmes raisons.

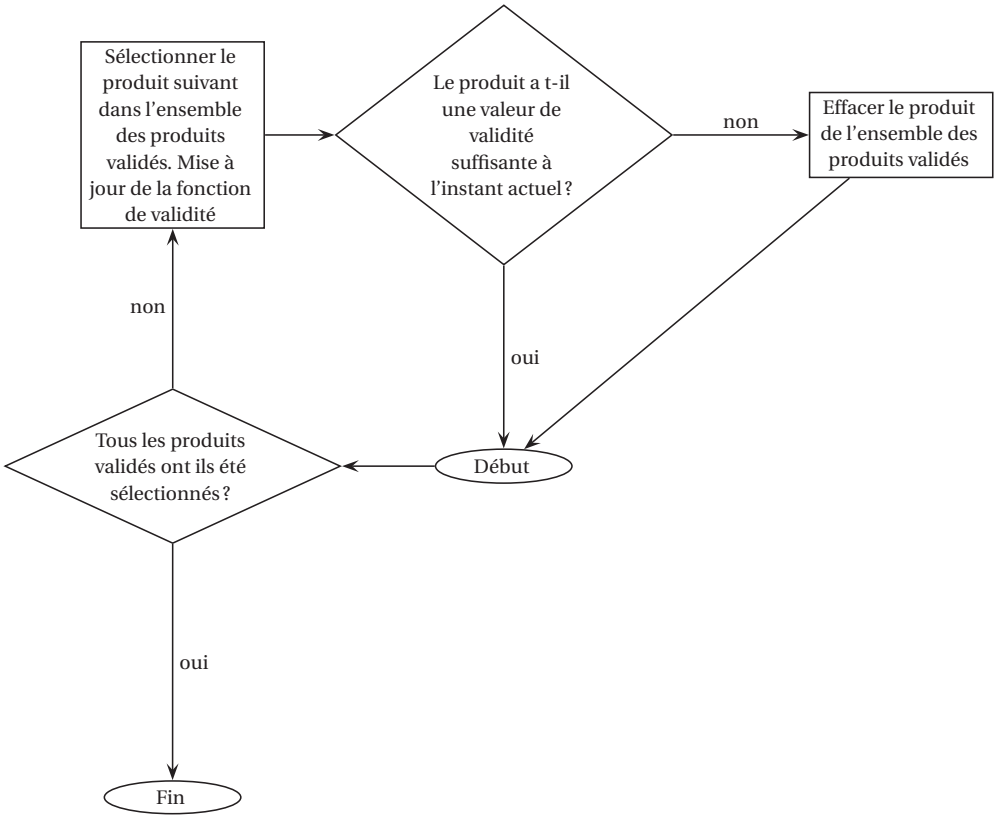


FIGURE 4.4 – Adaptation. Fonction de validité.

4.3 Structures de données et algorithmes

DANS l'algorithme présenté une donnée (i, r, t) est définie par la structure d, où $d.i$ est l'identifiant, $d.r$ la position (x, y) et $d.t$ l'instant. Les autres structures de données utilisées sont m et p (m pour machine et p pour produit). La structure m, est définie dans Table 4.1, et la fonction pour la créer est $\breve{m}(r, t)$. La structure p est définie dans Table 4.1. Elle est créée par la fonction $\breve{p}(i, m)$. En plus des structures d, m, p, il existe deux ensembles globaux, M, et P, vides par défaut.

L'algorithme 1 traduit le diagramme de flux du Figure 4.2 présenté à la section 4.2. Il a pour objectif de construire progressivement le modèle au fur et à mesure de l'arrivée des nouvelles données de localisation.

TABLE 4.1 – Structures de données m et p

var	description	défaut
$m.r$	position (x, y)	none
$m.t$	instant courant	none
$m.s$	binaire pour désigner l'arrêt	0
$m.T$	temps de service	0.0
$m.l$	instant de sortie	0.0
$m.a$	temps d'arrêt	0.0
$p.i$	identifiant	none
$p.m$	structure de type m	none
$p.M$	ensemble ordonné de structures m	$\emptyset$

Algorithme 1 Construction progressive du modèle

for all d **do**
  if $\exists\, p \in P\colon\ p.i = d.i$ **then**
    if $p.m.s = 1$ **then**
      if $p.m.r = d.r$ **then**
        $p.m.a \leftarrow p.m.a + d.t - p.m.t$
        $p.m.t \leftarrow d.t$
      else
        $m \leftarrow m' \in M\colon\ m'.r = p.m.r$
        if $p.m.t - p.m.a < m.l$ **then**
          $p.m.T \leftarrow p.m.t - m.l$
        else
          $p.m.T \leftarrow p.m.a$
        end if
        $m.l \leftarrow p.m.t$
        $p.M \leftarrow p.M \cup \{p.m\}$
        $p.m \leftarrow \breve{m}(d.r, d.t)$
      end if
    else
      if $p.m.r = d.r$ **then**
        $p.m.s \leftarrow 1$
        if $\nexists\, m \in M\colon\ m.r = d.r$ **then**
          $M \leftarrow M \cup \{p.m\}$
        end if
      else
        $p.m.r \leftarrow d.r$
      end if
      $p.m.t \leftarrow d.t$
    end if
  else
    $m \leftarrow \breve{m}(d.r, d.t)$
    $p \leftarrow \breve{p}(d.i, m)$
    $P \leftarrow P \cup \{p\}$
  end if
end for

L'algorithme 2 présente pour sa part la mise à jour des produits (fusion de paramètres, ajout nouveau type produit et correspond au diagramme de Figure 4.3) Le paramètre ϵ est le seuil évoqué à la section 4.2 permettant de discriminer les types de produit par le biais de leurs distances spatio-temporelles. p_{in} représente un produit dans l'ensemble des produits en opération. $T(p_{\text{in}})$ est le temps total des opérations du produit et $T^{\star}$ est la durée de vie moyenne d'un produit dans le système. L'ensemble U est l'ensemble des produits sortis non validés. L'ensemble V correspond à l'ensemble des produits sortis validés.

Algorithme 2 Mise à jour des produits

```
for all p_in do
    if T(p_in) > 2T* then
        if ∃x ∈ U ∪ V : distance(x, p_in) < ε then
            p_in = fusion(x, p_in)
            if x ∈ V then
                V = V ∪ {p_in}
            else
                U = U ∪ {p_in}
            end if
        else
            U = U ∪ {p_in}
        end if
    end if
end for
```

En fin l'algorithme 3, qui reprend le diagramme de flux Figure 4.4, montre le principe d'adaptation du modèle. ω y représente le paramètre "seuil de récurrence" évoqué à la section 4.2.

Algorithme 3 Adaptation

```
for all p ∈ V ∪ U do
    validity(p) = f(validity(p), recurrence(p))
    if validity(p) < ω then
        V = V/{p}
    else
        V = V ∪ {p}
    end if
end for
```

4.4 Conclusions du chapitre

Ce chapitre nous a donc permis de présenter, de manière descriptive puis formelle, le prototype du générateur réalisé. Cette implémentation s'appuie naturellement sur les propositions qui

ont été les nôtres dans le chapitre 3. Le chapitre 5 va maintenant nous servir à valider, non seulement le fonctionnement de ce prototype, mais également et de manière indirecte, l'ensemble de nos propositions.

Chapitre 5

Expérimentations et validations

5.1 Introduction

L'OBJECTIF principal de ce dernier chapitre est de valider les propositions présentées dans ce manuscrit. Pour ce faire et sur la base de deux scénarios différents nous avons mené des expérimentations visant à mesurer la similarité statistique du comportement du système d'intérêt et du modèle issu du générateur. Pour chacun de ces deux scénarios une étude sur le régime nominal et sur le régime perturbé sera conduite.

L'étude du régime nominal nous permettra de vérifier que le layout, les gammes et nomenclatures des produits sont similaires dans le système d'intérêt et dans le modèle généré. Nous y vérifierons également que les distributions de probabilité des temps d'attente et de service sont très proches. L'étude du régime perturbé aura pour sa part vocation à vérifier et valider les mécanismes d'adaptation mis en œuvre.

Les deux cas tests et les conditions expérimentales utilisées seront dans un premier temps présentés dans ce chapitre, avant la partie d'analyse des résultats auxquels ils mènent. Pour rendre ces expérimentations possibles, il nous a fallu générer de manière artificielle des données de localisation. Cette production des données est nécessaire pour mener à bien nos expérimentations car nous ne disposons pas à l'heure actuelle de données issues du monde réel. Cette partie production de données fait donc l'objet du paragraphe suivant.

5.2 Production de données *en silico*

CETTE première partie a donc pour vocation à générer les données nécessaires à la validation de notre générateur. Nous allons dans un premier temps présenter le système à l'origine des données et le paramétrage choisi. Puis dans un second temps nous montrerons que la structure spatio-temporelle des données générées par ce modèle sont conformes à ce qui est communément admis dans la littérature. Précisons également que ce module de simulation a été réalisé avec SimPy (Simulation in Python) [134, 135, 9]. Celui-ci nous génère le flux de localisation des produits. Il fait office d'artefact du système réel.

5.2.1 Présentation du système d'intérêt

LE SYSTÈME d'intérêt choisi représente sur le principe un atelier de fabrication constitué de différentes machines entre lesquelles les produits évoluent, transportés par des AGV's [1]. D'autres systèmes auraient pu également être éligibles. Cependant notre choix s'est porté vers ce type de système "job-shop" car les flux y sont plus diversifiés, ce qui représente un intérêt plus grand pour notre application.

Les différents types de produits sont générés selon l'équation :

$$mach(s,l) = (s-l) \mod M \tag{5.1}$$

mod represente la division euclidienne.

- l : le type de produit, $l = 0,\ldots,L-1$,
- L : le nombre de types de produits uniques,
- M : le nombre de machines ou postes de travail ($M \geq L$),
- s : l'opération, $s = 0,\ldots,S-1$,
- S : le nombre d'opérations dans la gamme.

Cette formulation se base sur la formulation initiale proposée par [181]. La formule ainsi proposée permet de générer des séquences de transformation (les gammes) pour chacun des produits.

Exemple : M=3 machines, S=5 opérations, L=3 types de produits.

1. AGV : Automatic Guided Vehicle

Produits = {[0,1,2,0,1],[2,0,1,2,0],[1,2,0,1,2]} ⇐

$s \rightarrow$ / $l \downarrow$	0	1	2	3	4
0	0	1	2	0	1
1	2	0	1	2	0
2	1	2	0	1	2

Pour les scénarios testés dans cette thèse, nous utiliserons :

- Des temps d'inter-arrivées des produits dans l'atelier fournis par une fonction de distribution exponentielle. Une proportion identique de chaque type de produit est générée.
- Des temps de service de chaque machine également fournis par une fonction de distribution exponentielle.
- Une disposition des machines dans le système réalisée de manière aléatoire en début de simulation, dans un cadre également figé (fonction de distribution uniforme bornée par les dimensions du système).
- Un système de localisation pour obtenir l'information de localisation du produit (i,r,t) à une fréquence fixe.

Les données issues de ce modèle sont naturellement de type (i,r,t). Ces données sont bruitées, selon un principe d'ajout du bruit aux données de localisation $r+\varepsilon$. Le modèle d'erreur choisi obéit à une loi Normale $\hat{r}(t) = r(t) + \varepsilon$, avec $\varepsilon = (\varepsilon_x, \varepsilon_y)$; $\varepsilon_x, \varepsilon_y \sim N(0, \sigma_\varepsilon^2)$.

La Figure 5.1 représente, sous la forme d'une machine d'états, le processus de base mis en oeuvre pour la réalisation d'un produit. Un schéma simplifié est représenté Figure 5.1 (à gauche) et un schéma plus détaillé Figure 5.1 (à droite).

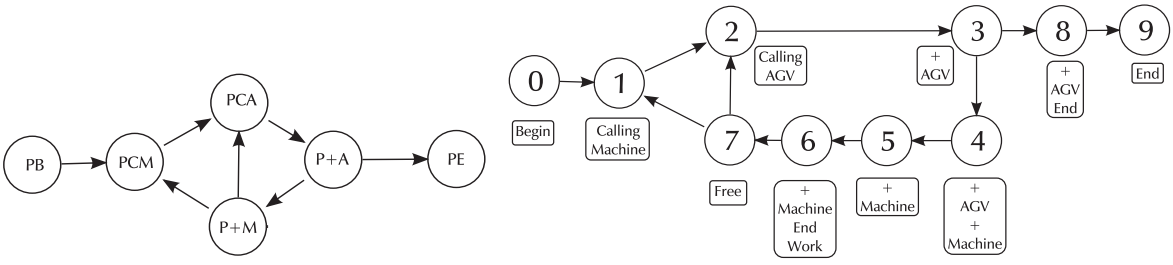


FIGURE 5.1 – Machines d'états pour la simulation «système - réel». Simplifié (gauche), et plus détaillé (droite).

Schéma simplifié Dans le schéma simplifié, le processus est représenté par une machine d'états, avec six états possibles :

1. Arrivée du produit (PB).
2. Le produit appelle une machine (PCM).
3. Le produit appelle un AGV (PCA).
4. Le produit est pris en charge par un AGV (P+A).
5. Le produit est pris en charge par la machine (P+M).
6. Le Produit sort (PE).

Schéma détaillé Ce schéma détaille le schéma précédent. Il est composé de dix états :

1. Arrivée du produit (0)
2. Le produit appelle une machine (1)
3. Le produit appelle un AGV (2)
4. Le produit est pris en charge par un AGV (3)
5. Le couple Produit-AGV est pris en charge par une machine (4)
6. Le produit est associé à la machine, l'AGV est libéré, la machine commence une opération (5)
7. La machine termine l'opération (6)
8. Le produit libère la machine (7)
9. Le produit libère l'AGV (8)
10. Le produit finalisé sort du système (9)

Pour l'implémentation du moteur de simulation apte à générer les données de localisation, nous avons également modélisé les différentes classes nécessaires. Elles sont représentées à la Figure 5.2.

G Classe Globale. Une graine aléatoire est utilisée pour l'initialisation d'un générateur de nombres pseudo-aléatoires. Toutes les variables globales de la simulation sont fixées ici. Par exemple, les AGVs, sont des ressources partagées qui doivent être disponibles pour le transport des produits.

Product Classe Produits. Chaque produit est initialisé avec un Id, et une gamme à parcourir. Chaque événement produit est dirigé par le produit même. Chaque produit garde l'information sur sa position, le temps, l'AGV et la machine courante utilisée.

Machine Classe Machine. Chaque machine est créée avec une position fixe et un Id unique.

World Classe Monde. Cette classe représente le système manufacturier. Elle permet l'ajout et la configuration des machines et produits dans le système et exécute la simulation.

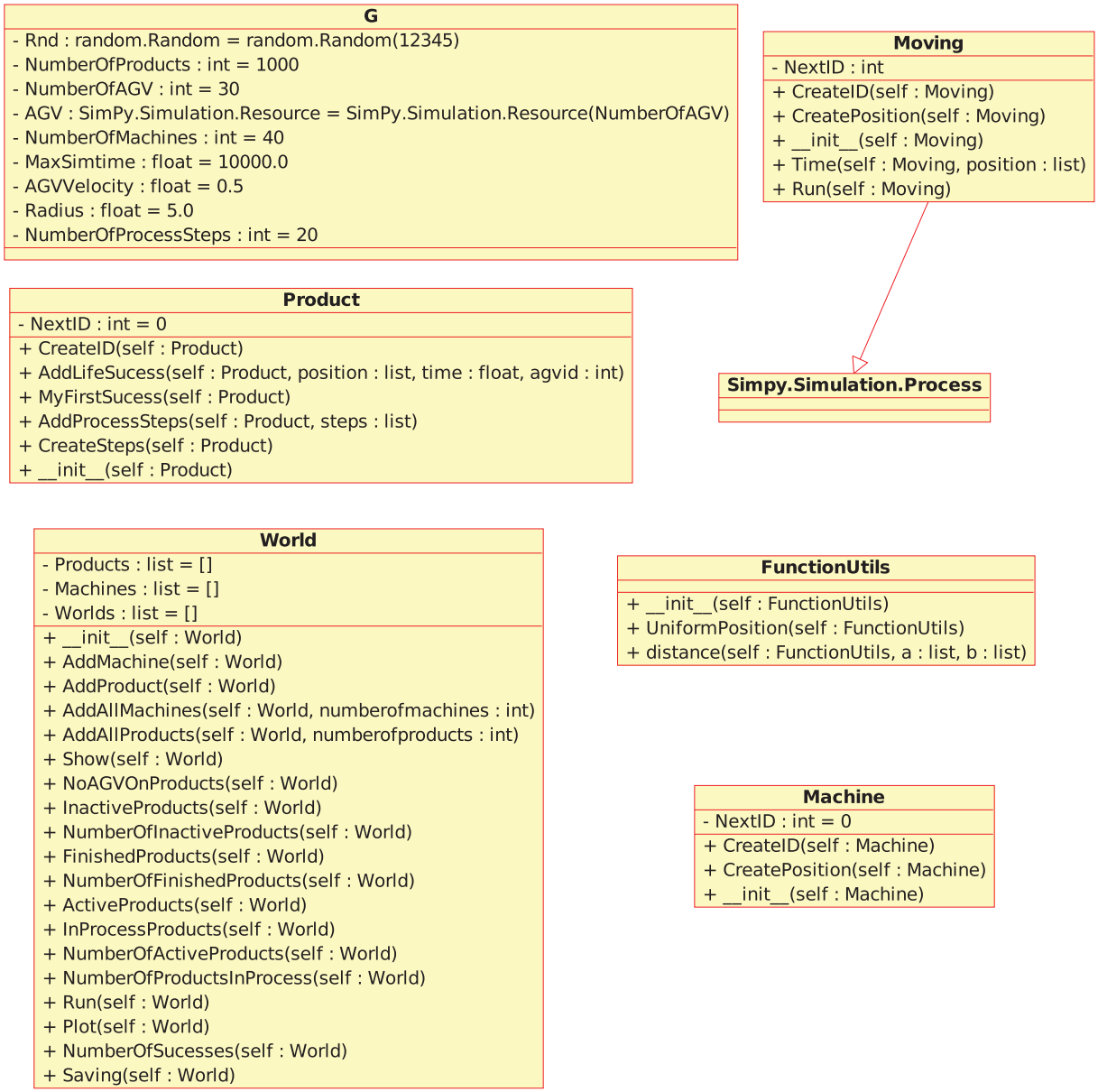


FIGURE 5.2 – Diagramme de classes UML pour le moteur de simulation.

Moving Classe En Mouvement. Cette classe simule les mouvements des AGVs. Chaque AGV de la liste globale des ressources partagées est liée à une instance de cette Classe En Mouvement pour avoir la possibilité d'un paramétrage diffèrent pour chaque AGV, et une gestion des ressources globales.

FunctionUtils Classe Fonctions Outils. Cette classe génère des positions et calcule les distances entre les éléments du système.

5.2.2 Détermination et vérification de la nature des données générées

AFIN de déterminer et de vérifier la nature des données générées par ce modèle (représentant le système réel ou d'intérêt) nous avons étudié les données :

- Temporelles en observant la distribution et l'évolution du temps entre l'ensemble des événements étant générés.
- Spatiales en observant la distribution et l'évolution de l'ensemble des positions des objets.

Pour l'étude des évolutions temporelles, nous définissons l'ensemble des temps entre deux événements consécutifs T_{ee}. $T_{ee} = \triangle_1\{t_i : t_k < t_{k+1}\}_{i=1}^{n} = \{t_{i+1} - t_i\}_{i=1}^{n-1}$

Grâce à un processus de comptage d'occurrences on peut obtenir ($N(t)$, le nombre d'occurrences d'événements jusqu'à l'instant t) qui vérifie l'une des propriétés des événements rares, à savoir que la probabilité qu'il y ait plus d'une occurrence d'événement dans un petit intervalle de temps est négligeable.

$$P\{N(t+\delta) - N(t) > 1\} = o(\delta) \text{ lorsque } \delta \rightarrow 0+ \tag{5.2}$$

Ce processus de comptage global permet de modéliser tout le système comme un processus de Poisson de paramètre λ :

$$P(N_t = k) = \mathrm{e}^{-\lambda t}\frac{(\lambda t)^k}{k!} \tag{5.3}$$

Dans ce cas, le temps entre deux événements du processus de comptage, est une variable aléatoire indépendante, de loi exponentielle également de paramètre λ :

$$P(Tee \leq t) = 1 - \mathrm{e}^{-\lambda t} \tag{5.4}$$

Nous vérifions donc au final que le générateur de données a un comportement global de type loi exponentielle (Figure 5.3). Ce type de comportement est celui qui est couramment admis dans la littérature pour les réseaux de files d'attentes [36].

L'étude des évolutions spatiales va pour sa part nous servir à déterminer le moment à partir duquel le système simulé arrive dans un comportement stable, et si il est possible de vérifier des changements d'états à un niveau de spécificité proche de celui proposé à la figure Figure 5.1 au niveau détaillé. Afin d'obtenir rapidement des évidences empiriques sur ce sujet, nous avons réalisé un graphe de récurrence [62, 90, 119] du temps entre événements du système, pour en identifier les caractéris-

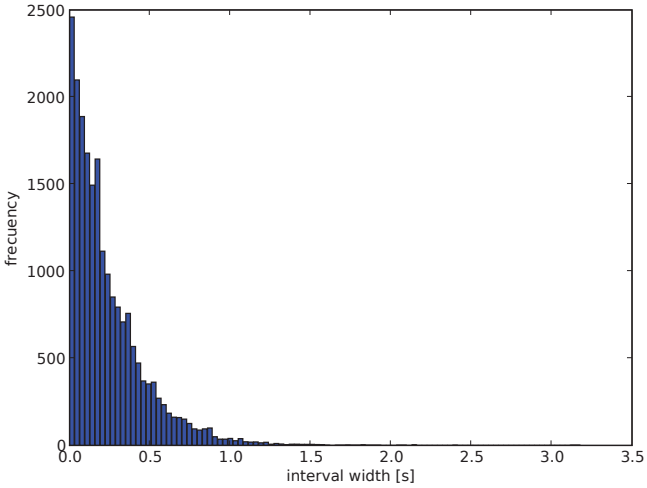


FIGURE 5.3 – Temps entre événements global T_{ee} pour une réalisation du modèle de simulation. Histogramme représentant la loi exponentielle global.

tiques globales. Un graphe de récurrence R_ϵ de la grandeur $x(t)$ est un graphe symétrique par rapport à la diagonale où pour chaque paire d'instants de temps (t_i, t_j) on représente :

$$R(t_i, t_j) = \begin{cases} 1 & \|x(t_i) - x(t_j)\| < \epsilon \\ 0 & \text{sinon} \end{cases} \tag{5.5}$$

Dans la pratique, pour construire le graphe, un point sur le graphe dans la position (t_i, t_j) est ajouté si la distance entre $x(t_i)$ et $x(t_j)$ est assez petite. Sur un graphe de récurrence nous pouvons observer le comportement global du système et vérifier s'il présente une structure déterministe, ou bien stochastique et si cette structure change pendant le temps. A la Figure 5.4 nous présentons le graphe de récurrence pour le temps entre événements global. Sur cette figure, on peut remarquer le changement de phase du système entre le régime transitoire et le régime stable un peu avant l'instant 5000. La structure de ce graphe de récurrence montre une distribution homogène, propre à la nature aléatoire de la durée entre événements. La densité des points dans la zone $t < 5000$ est clairement moins intense que dans la zone $t > 5000$. Ce fait représente un changement de paramètres dans la loi de distribution globale du système.

Afin de compléter l'analyse précédente, uniquement basée sur l'aspect temporel des données, nous avons également analysées ces données sous leur forme spatio-temporelle. La figure Figure 5.5 propose une représentation possible de cette forme spatio-temporelle. Les coordonnées x et y représente l'espace à deux dimensions du système d'intérêt alors que l'axe z y représente le temps. On peut

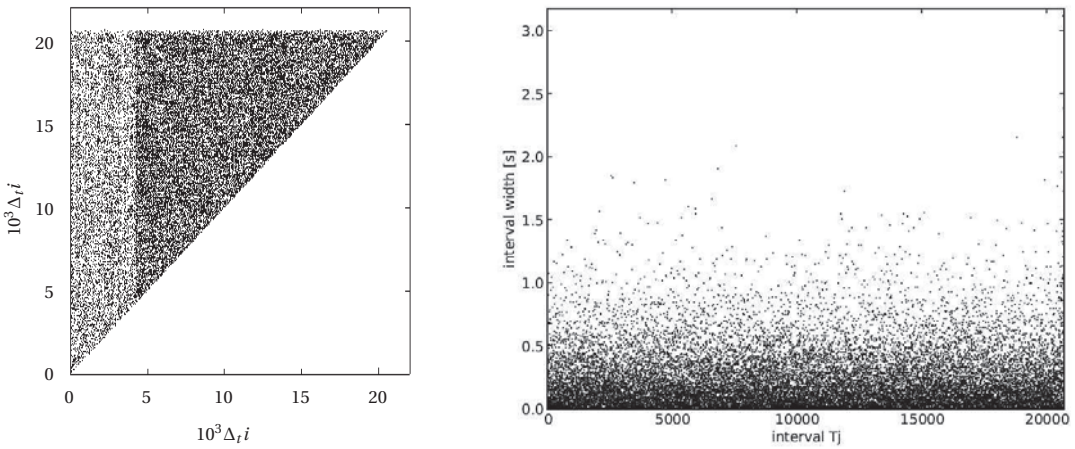


FIGURE 5.4 – Graphe de récurrence du temps entre événements global (gauche) pour une réalisation du modèle de simulation. Temps entre événements global T_{ee}, série temporelle (droite)

remarquer sur cette figure les zones de l'espace dans lesquelles les produits passent le plus de temps (les pics selon l'axe z). Ces zones de densité forte correspondent aux positions des serveurs.

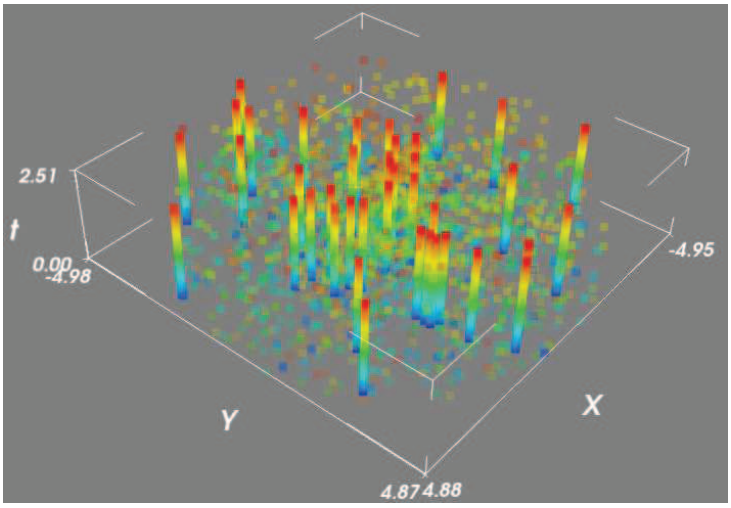


FIGURE 5.5 – Système manufacturier, agrégation des données spatiotemporelles 40 machines, z-axis représente le temps.

5.3 Scénarios de validation

CETTE partie a vocation à valider le principe de faisabilité de la génération automatique d'un code de simulation sur la base d'informations de localisation. Le flux de localisation est récupéré par le module générateur de modèle de simulation qui met en œuvre les algorithmes présentés au chapitre précédent. Au final, nous utiliserons un module d'analyse des données pour la vérification des résultats obtenus.

Sur la base du modèle d'intérêt générique présenté au paragraphe précédent, nous précisons maintenant les valeurs des paramètres utilisés lors des deux scénarios tests effectués (voir 5.1).

Paramètres	Valeurs
temps de service sur machines	exponentiel, $1/\mu = 13$
nombre d'AGV's	30
vitesse des AGV's	0.44
dimensions de l'atelier	$(-10,-10),(10,10)$
periode d'échantillonnage pour la localisation	0.5

TABLE 5.1 – Conditions expérimentales

Les tables 5.2 et 5.3 précisent quant à elles les paramètres particuliers à chacun des deux scénarios.

Paramètres	Valeurs
nombre de types de produits	3
nombre de machines	3
nombre d'opérations ou phases	3
quantité de produits (maximale)	500
temps d'inter-arrivées	exponentiel, $1/\lambda = 23$

TABLE 5.2 – Cas 1

Paramètres	Valeurs
nombre de types de produits	13
nombre de machines	20
nombre d'opérations ou phases	8
quantité de produits (maximale)	500
temps d'inter-arrivées	exponentiel $1/\lambda = 23$

TABLE 5.3 – Cas 2

5.4 Résultats et analyse

AFIN de valider le système générateur nous allons présenter les résultats obtenus dans des situations différentes. L'analyse des résultats obtenus en régime nominal sera tout d'abord présentée. Elle sera suivie d'une analyse d'un régime perturbé sera présentée afin de valider les mécanismes d'adaptation mis en place.

5.4.1 Régime nominal

NOUS présentons ici les résultats obtenus pour le scénario 1. Le tableau 5.8 présente les temps de calcul nécessaires à l'obtention de ces résultats[2]. Le temps d'initialisation correspond au temps nécessaire à la construction de la structure du modèle du système d'intérêt (artefact). Le temps de simulation accumulé représente le temps nécessaire à la génération des données nécessaires à l'alimentation du générateur. Le temps de génération accumulé est la durée nécessaire à la génération du modèle par le biais du générateur, et enfin le temps de calcul statistique est le temps nécessaire au calcul des résultats présentés (Table 5.5, Table 5.6 et Table 5.7).

Temps d'initialisation	0.44 [s]
Temps de simulation accumulé	14.39 [s]
Temps de génération accumulé	186.26 [s]
Temps de calcul statistiques de test accumulé	209.16 [s]

TABLE 5.4 – Temps de calcul de référence scénario 1

Les résultats présentés dans chacune de ces tableaux principal (Table 5.5, Table 5.6 et Table 5.7)) sont relativement simples d'interprétation. Ils confirment la modélisation correcte des gammes de chacun des produits et de la localisation des « buffer-opération » (représentées par $A, B, C, \ldots$). La seconde partie des tableaux représente quant à elle les résultats liés aux aspects temporels. Les temps d'inter-arrivées $(1/\lambda)$ et les temps de service $(1/\mu)$, de chaque type de produit sur chaque serveur, obtenus par le générateur, y sont comparés à ceux issus du système d'intérêt. Nous avons préféré présenter ici les temps d'inter-arrivées des produits $(1/\lambda)$ sur les buffer-opérations que les temps d'attente dans le buffer (W), car plus directement liés au paramétrage classique d'un réseau de files d'attente. Ces deux temps sont liés entre eux par l'équation :

$$W = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)} \tag{5.6}$$

2. Valeurs de référence pour un ordinateur netbook Linux 2.6 ; 2 Intel Atom N450 at 1.66Ghz ; 1GB RAM

Gamme	A	B	C
Coordonnée *x* des «buffer-opération»	-2.98	-2.171	2.833
Coordonnée *x* des «buffer-opération» généré	-2.986	-2.166	2.83
Coordonnée *y* des «buffer-opération»	0.001	2.434	-5.919
Coordonnée *y* des «buffer-opération» généré	-0.003	2.44	-5.914
Temps inter-arrivées moyen	23.015	31.382	37.056
Temps inter-arrivées moyen généré	23.015	31.382	37.058
Temps inter-arrivées écart type	22.501	33.659	38.066
Temps inter-arrivées écart type généré	22.501	33.659	38.065
Wilcoxon t	2047.0	1348.0	267.5
Wilcoxon p	0.0	0.0	0.0
Kolmogorov-Smirnov d	0.006	0.006	0.006
Kolmogorov-Smirnov p	1.0	1.0	1.0
Temps de service moyen	11.739	12.195	13.027
Temps de service moyen généré	12.121	12.804	12.521
Temps de service écart type	12.148	11.738	13.28
Temps de service écart type généré	11.801	11.619	13.043
Wilcoxon t	5676.0	5594.5	5119.0
Wilcoxon p	0.257	0.126	0.007
Kolmogorov-Smirnov d	0.082	0.051	0.032
Kolmogorov-Smirnov p	0.64	0.985	1.0

TABLE 5.5 – Résultats scénario 1, produit 0

Gamme	C	A	B
Coordonnée *x* des «buffer-opération»	2.833	-2.98	-2.171
Coordonnée *x* des «buffer-opération» généré	2.811	-2.972	-2.173
Coordonnée *y* des «buffer-opération»	-5.919	0.001	2.434
Coordonnée *y* des «buffer-opération» généré	-5.901	0.007	2.432
Temps inter-arrivées moyen	22.887	29.965	37.964
Temps inter-arrivées moyen généré	47.435	62.03	78.338
Temps inter-arrivées écart type	24.001	27.598	35.492
Temps inter-arrivées écart type généré	42.806	61.568	70.552
Wilcoxon t	655.0	785.0	645.0
Wilcoxon p	0.0	0.001	0.0
Kolmogorov-Smirnov d	0.32	0.275	0.331
Kolmogorov-Smirnov p	0.0	0.001	0.0
Temps de service moyen	11.628	14.345	12.457
Temps de service moyen généré	9.18	12.463	12.303
Temps de service écart type	11.585	15.375	14.029
Temps de service écart type généré	8.562	14.46	14.071
Wilcoxon t	1235.0	1344.0	1438.0
Wilcoxon p	0.238	0.538	0.897
Kolmogorov-Smirnov d	0.126	0.091	0.082
Kolmogorov-Smirnov p	0.367	0.77	0.862

TABLE 5.6 – Résultats scénario 1, produit 1

Si les lois des temps ne suivent pas nécessairement une loi exponentielle, il nous est toujours possible d'obtenir par le générateur le nombre moyen d'éléments dans le buffer Q et ainsi utiliser la loi de Little quand le système est équilibré (régime nominal) :

$$Q = \lambda W \Rightarrow W = \frac{Q}{\lambda} \tag{5.7}$$

Les comparaisons des moyennes et des écarts types montrent que les temps d'arrivées issus de l'artefact et reconstruits par le générateur sont bien identifiés .

Gamme	B	C	A
Coordonnée x des «buffer-opération»	-2.171	2.833	-2.98
Coordonnée x des «buffer-opération» généré	-2.173	2.834	-2.983
Coordonnée y des «buffer-opération»	2.434	-5.919	0.001
Coordonnée y des «buffer-opération» généré	2.435	-5.922	0.008
Temps inter-arrivées moyen	19.804	26.34	30.924
Temps inter-arrivées moyen généré	21.572	28.692	33.685
Temps inter-arrivées écart type	21.124	25.195	29.087
Temps inter-arrivées écart type généré	22.029	27.677	30.345
Wilcoxon t	5736.0	5638.0	5691.0
Wilcoxon p	0.041	0.391	0.445
Kolmogorov-Smirnov d	0.065	0.057	0.056
Kolmogorov-Smirnov p	0.836	0.934	0.937
Temps de service moyen	12.736	12.164	14.055
Temps de service moyen généré	12.176	10.693	14.765
Temps de service écart type	11.802	11.171	13.849
Temps de service écart type généré	11.745	9.943	14.365
Wilcoxon t	7003.0	6571.0	7049.0
Wilcoxon p	0.778	0.337	0.834
Kolmogorov-Smirnov d	0.071	0.076	0.055
Kolmogorov-Smirnov p	0.745	0.674	0.943

TABLE 5.7 – Résultats scénario 1, produit 2

Par contre, si la valeur moyenne des temps de service du générateur est très proche de la moyenne issue du « réel », les différences un peu plus marquées apparaissent dans deux cas (produit 0, opération A) et (produit 2, opération B) au niveau des écarts types. Il nous est difficile à trouver une explication rationnelle à ces différences. La fréquence d'échantillonnage avec laquelle nous travaillons peut être en partie responsable de celles ci.

Il semble donc, sur la base de ces simples comparaisons, possibles de dire qu'il est possible d'obtenir une bonne représentation des temps de service et de temps d'inter-arrivées par le biais du générateur.

Ces premières comparaisons ont été complétées par des tests statistiques. Le premier test utilisé est le test de Wilcoxon [190] (signed rank test). Ce test statistique permet d'effectuer la comparaison de deux populations (il est également dénommé test d'homogénéité), sur la base de la différence entre deux échantillons, à partir de valeurs rangées par ordre croissant en valeur absolue. Dans notre cas particulier, nous comparons bien sûr les valeurs des temps d'inter-arrivées et des temps de service issues du «réel» et à celles obtenues par le biais du générateur. Le résultat de ce test permet de ne pas rejeter (ou rejeter) l'hypothèse que les deux populations sont issues de la même distribution, et de quantifier le risque pris pour cette affirmation. La statistique $t = \min(r_+, r_-)$, ou $r+$ $(r-)$ est l'addition des rangs positifs (négatifs). Cette statistique t est dépendante du nombre de rangs non nulles pour l'approcher d'une loi normale de statistique z si le nombre de rangs n est grand (i.e. $n > 20$). La valeur p est calculée par $p = 2P(Z > |z|)$. Le test de Wilcoxon est un test non paramétrique car aucun modèle *a priori* sur la loi de probabilité des variables n'est nécessaire.

Le second test utilisé est le test non paramétrique de Kolmogorov-Smirnov. Il ne mesure pas les rangs, mais les distances entre les fonctions de répartition (*cdf*[3]). Ce test est basé sur un indicateur qui tend vers zéro quand la distance entre deux distributions est réduit. Si la statistique d est petite ou la valeur p est grande, alors il ne nous est pas possible de rejeter l'hypothèse que les lois de probabilités des deux échantillons sont identiques. Ce test a été utilisé ici en complément du test de Wilcoxon car il n'est pas contraint par le nombre de rangs non nuls entre les échantillons. Les résultats obtenus pas ces deux test nous montrent ici les lois de répartition des données issues de l'artefact et celles issues du générateur sont les mêmes.

A titre d'information complémentaire, nous présentons à la figure 5.6, un exemple d'histogramme des temps d'inter-arrivées et des temps de service d'un produit (le produit 1) sur une machine (la machine A). L'exemple représenté sur cette figure a été choisi comme un exemple "défavorable" afin de pouvoir y distinguer les courbes. En effet, la plupart des autres exemples auraient amené à une superposition quasi parfaite des courbes, rendant ainsi la lecture difficile.

Nous présentons sous une forme en tout point identique à celle du scénario 1, les résultats obtenus pour le scénario 2. Nous limitons cependant la présentation de ces résultats à ceux d'un seul produit, les 12 autres produits n'étant pas représentables sur un même tableau. Ces résultats sont présentés en annexe.

Temps d'initialisation	0.45 [s]
Temps de simulation accumulé	27.53 [s]
Temps de génération accumulé	299.44 [s]
Temps de calcul statistiques de test accumulé	552.58 [s]

TABLE 5.8 – Temps de calcul de référence scénario 2

A la lecture et l'analyse de ces résultats, on peut dire que ceux-ci confirment les résultats obtenus sur le scénario 1. Les moyennes et écart type (artefact, générateur) sont là encore très proches, les lois de distribution sont identiques.

Là encore nous représentons, à titre d'information complémentaire, un exemple d'histogramme des temps d'inter-arrivées et des temps de service (produit 0, machine F) .

3. en anglais, cumulative distribution function

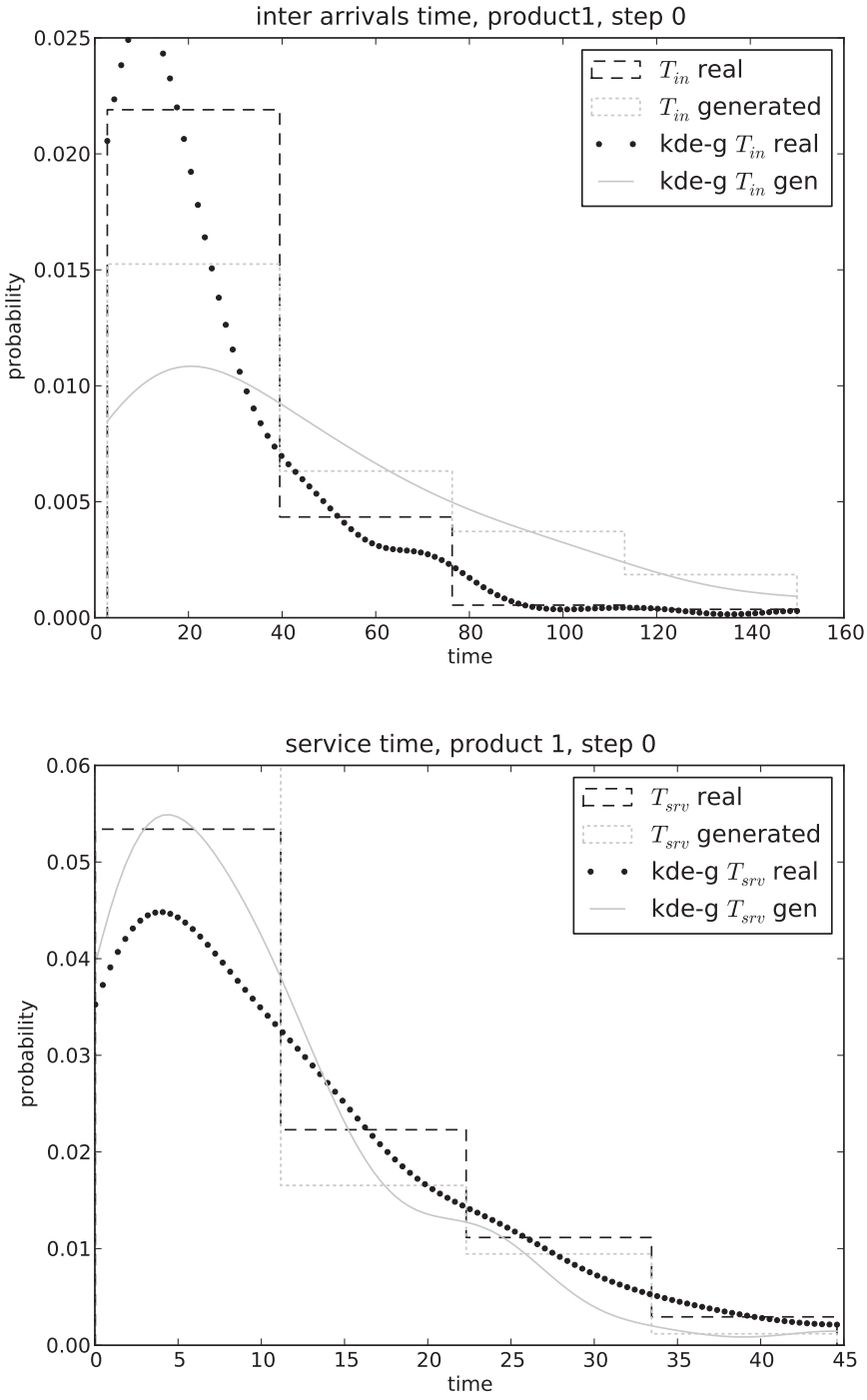


FIGURE 5.6 – Scénario 1 : Histogrammes des temps inter-arrivées et temps de service. Produit 1, machine A

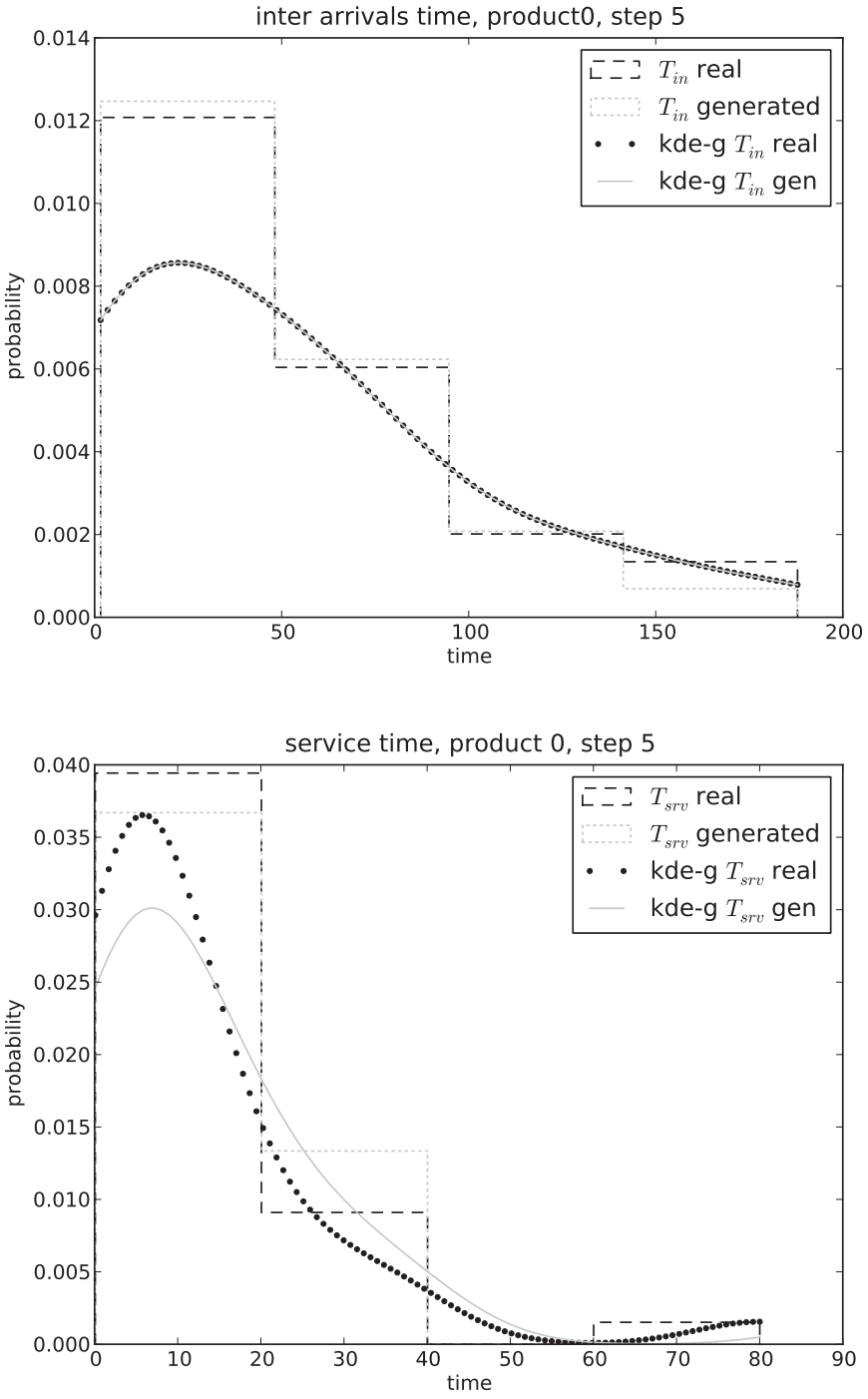


FIGURE 5.7 – Scénario 2 : Histogrammes des temps inter-arrivées et temps de service. Produit 0, machine F

Gamme	A	B	C	D	E	F	G	H
Coordonnée *x* des «buffer-opération»	-2.98	-2.171	2.833	6.53	-5.175	-2.993	1.929	-0.721
Coordonnée *x* des «buffer-opération» généré	-2.979	-2.171	2.832	6.53	-5.175	-2.993	1.929	-0.718
Coordonnée *y* des «buffer-opération»	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677	-6.505
Coordonnée *y* des «buffer-opération» généré	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677	-6.505
Temps inter-arrivées moyen	24.808	41.119	42.906	47.514	49.629	50.898	51.507	51.676
Temps inter-arrivées moyen généré	24.808	41.119	42.906	47.514	49.629	50.898	51.507	51.676
Temps inter-arrivées écart type	21.307	43.859	41.119	33.812	42.621	47.984	50.709	50.454
Temps inter-arrivées écart type généré	21.307	43.859	41.106	33.816	42.621	48.013	50.713	50.454
Wilcoxon t	128.0	26.0	10.0	26.5	2.0	13.5	10.5	1.5
Wilcoxon p	0.0	0.509	0.114	0.672	0.593	0.0	1.0	0.0
Kolmogorov-Smirnov d	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	9.904	9.768	12.634	10.208	11.823	12.078	9.871	10.345
Temps de service moyen généré	9.776	10.414	12.64	10.089	12.021	14.238	9.633	10.952
Temps de service écart type	10.575	9.518	11.678	8.728	12.349	15.393	8.831	11.199
Temps de service écart type généré	11.244	10.633	12.001	8.735	14.653	17.487	8.693	11.093
Wilcoxon t	272.0	225.0	168.5	175.0	268.0	225.0	93.0	205.5
Wilcoxon p	0.879	0.171	0.02	0.059	0.823	0.321	0.001	0.086
Kolmogorov-Smirnov d	0.121	0.121	0.061	0.061	0.091	0.152	0.091	0.061
Kolmogorov-Smirnov p	0.957	0.957	1.0	1.0	0.999	0.811	0.999	1.0

TABLE 5.9 – Résultats scénario 2, produit 0

5.4.2 Cas du système perturbé

AFIN de valider les mécanismes d'adaptation mis en place, plusieurs scénarios étaient là encore éligibles. Nous avons pris ici le parti de présenter un scénario relativement simple sur la base du scénario 1 du paragraphe précédent. A ce scénario initial, un nouveau produit (le produit 3) de gamme A-B-D, est ajouté lors de la vie du système. La Figure 5.8 montre la situation du système avant, pendant, et après l'introduction de ce nouveau produit. Elle montre bien l'apparition progressive d'un nouveau buffer-opération (la machine D), initialement inexistant sur le système. Une vidéo de cette adaptation serait plus représentative de ce type de phénomène. D'autres situations (disparition de machines, modifications d'emplacements, nouveau produits, ...) auraient pu être présentées car également testées. Les phénomènes observés et mesurés confirment tous la bonne adaptation du modèle aux évolutions du système réel.

Les résultats, issus de ce scénario particulier, sont présentés sous la même forme de tableau que les scénarios 1 et 2 précédents (Tableaux 5.10 à 5.13).

Dans son usage hors ligne (tel qu'utilisé dans le cadre de cette thèse), l'algorithme développé s'avère être assez lourd et prend du temps par rapport à la taille du système à identifier. Cependant, son exécution étant événementielle, il peut être facilement parallélisé, voire porté pour une exécution sur des grilles de calcul. Par contre, pour un usage en ligne de ce générateur, le problème du temps de calcul semble moins critique. En effet les temps de calcul du générateur sont liés dans ce cas aux vitesses d'écoulement des produits sur le système.

Produit	0		
Gamme	A	B	C
Coordonnée *x* des «buffer-opération»	-3.954	1.992	-5.83
Coordonnée *x* des «buffer-opération» généré	-3.954	1.992	-5.83
Coordonnée *y* des «buffer-opération»	-2.179	-1.902	0.057
Coordonnée *y* des «buffer-opération» généré	-2.179	-1.902	0.057
Temps inter-arrivées moyen	20.031	39.705	43.473
Temps inter-arrivées moyen généré	20.031	39.705	43.473
Temps inter-arrivées écart type	19.923	49.467	56.511
Temps inter-arrivées écart type généré	19.923	49.467	56.51
Wilcoxon t	1534.0	1332.5	606.5
Wilcoxon p	0.329	0.67	0.398
Kolmogorov-Smirnov d	0.008	0.008	0.008
Kolmogorov-Smirnov p	1.0	1.0	1.0
Temps de service moyen	14.395	11.966	11.716
Temps de service moyen généré	16.4	12.403	13.171
Temps de service écart type	13.72	10.969	12.205
Temps de service écart type généré	14.662	10.997	14.488
Wilcoxon t	3511.5	3662.5	3816.0
Wilcoxon p	0.02	0.057	0.104
Kolmogorov-Smirnov d	0.108	0.054	0.085
Kolmogorov-Smirnov p	0.417	0.99	0.722

TABLE 5.10 – Résultats scénario adaptatif, produit 0

Produit	1		
Gamme	C	A	B
Coordonnée *x* des «buffer-opération»	-5.83	-3.954	1.992
Coordonnée *x* des «buffer-opération» généré	-5.83	-3.954	1.992
Coordonnée *y* des «buffer-opération»	0.057	-2.179	-1.902
Coordonnée *y* des «buffer-opération» généré	0.057	-2.179	-1.902
Temps inter-arrivées moyen	22.134	32.641	52.797
Temps inter-arrivées moyen généré	22.134	32.645	52.797
Temps inter-arrivées écart type	23.023	34.78	54.083
Temps inter-arrivées écart type généré	23.023	34.806	54.083
Wilcoxon t	1650.0	2590.0	658.5
Wilcoxon p	0.13	0.803	0.0
Kolmogorov-Smirnov d	0.008	0.025	0.008
Kolmogorov-Smirnov p	1.0	1.0	1.0
Temps de service moyen	14.294	14.147	11.058
Temps de service moyen généré	14.318	22.283	11.686
Temps de service écart type	13.905	13.91	10.044
Temps de service écart type généré	14.086	32.228	10.404
Wilcoxon t	3094.5	2788.0	2620.5
Wilcoxon p	0.084	0.005	0.002
Kolmogorov-Smirnov d	0.059	0.118	0.067
Kolmogorov-Smirnov p	0.983	0.361	0.944

TABLE 5.11 – Résultats scénario adaptatif, produit 1

Produit	2		
Gamme	B	C	A
Coordonnée *x* des «buffer-opération»	1.992	-5.83	-3.954
Coordonnée *x* des «buffer-opération» généré	1.992	-5.83	-3.954
Coordonnée *y* des «buffer-opération»	-1.902	0.057	-2.179
Coordonnée *y* des «buffer-opération» généré	-1.902	0.057	-2.179
Temps inter-arrivées moyen	16.396	24.686	29.965
Temps inter-arrivées moyen généré	16.396	24.686	29.965
Temps inter-arrivées écart type	15.441	21.779	28.805
Temps inter-arrivées écart type généré	15.464	21.779	28.845
Wilcoxon t	4444.5	2634.5	4716.5
Wilcoxon p	0.525	0.0	0.542
Kolmogorov-Smirnov d	0.025	0.006	0.025
Kolmogorov-Smirnov p	1.0	1.0	1.0
Temps de service moyen	13.16	14.38	13.305
Temps de service moyen généré	13.369	13.635	25.417
Temps de service écart type	13.622	13.936	12.428
Temps de service écart type généré	13.518	13.867	32.614
Wilcoxon t	5543.0	5840.0	3163.5
Wilcoxon p	0.098	0.14	0.0
Kolmogorov-Smirnov d	0.025	0.107	0.233
Kolmogorov-Smirnov p	1.0	0.307	0.0

TABLE 5.12 – Résultats scénario adaptatif, produit 2

Produit	3		
Gamme	A	B	C
Coordonnée *x* des «buffer-opération»	-3.954	1.992	-4.376
Coordonnée *x* des «buffer-opération» généré	-3.954	1.992	-4.376
Coordonnée *y* des «buffer-opération»	-2.179	-1.902	5.379
Coordonnée *y* des «buffer-opération» généré	-2.179	-1.902	5.379
Temps inter-arrivées moyen	21.048	45.864	42.651
Temps inter-arrivées moyen généré	21.048	45.864	42.651
Temps inter-arrivées écart type	21.997	46.571	51.248
Temps inter-arrivées écart type généré	21.997	46.571	51.25
Wilcoxon t	1054.0	410.0	10.5
Wilcoxon p	0.0	0.0	0.0
Kolmogorov-Smirnov d	0.011	0.011	0.011
Kolmogorov-Smirnov p	1.0	1.0	1.0
Temps de service moyen	13.502	12.684	14.409
Temps de service moyen généré	14.719	11.963	15.162
Temps de service écart type	13.354	12.665	14.886
Temps de service écart type généré	13.523	12.438	14.151
Wilcoxon t	1835.0	2108.5	1474.0
Wilcoxon p	0.078	0.819	0.01
Kolmogorov-Smirnov d	0.076	0.065	0.076
Kolmogorov-Smirnov p	0.945	0.987	0.945

TABLE 5.13 – Résultats scénario adaptatif, produit 3

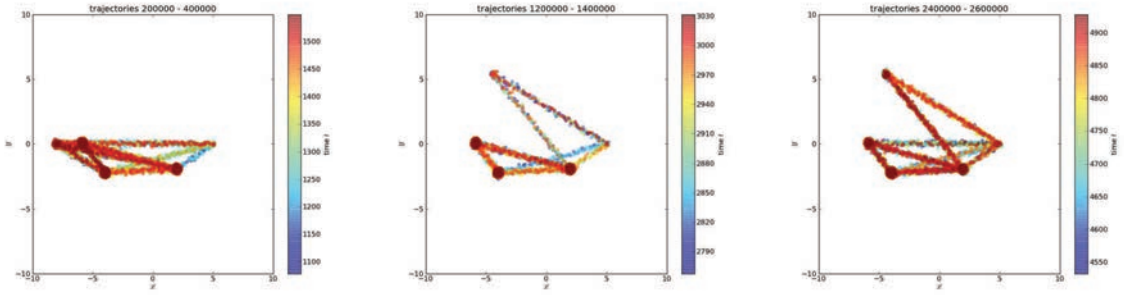


FIGURE 5.8 – Scénario «adaptation du modèle».

5.5 Conclusion

Ce dernier chapitre de thèse nous a permis de valider nos propositions par le biais de l'implémentation du générateur et des résultats qu'il a produit. Les résultats obtenus sur les deux cas tests étudiés ont pu montrer que les distances structurelles entre système d'intérêt et modèle généré étaient minces. Ils nous également permis de montrer que le modèle généré était relativement stable en régime nominal, mais aussi que les mécanismes d'adaptation déployés étaient efficaces.

D'autres tests auraient pu compléter ceux réalisés ici. Les paramètres de réglage du générateur ont ici été fixés empiriquement sans recherche d'une optimisation de ceux-ci. Le réglage de ces paramètres est en effet lié au système d'intérêt et doit donc faire à chaque fois l'objet d'un réglage particulier. La distance temporelle séparant les phénomènes du monde réel et leur prise en compte dans le modèle est également une mesure non effectuée dans ces tests. Elle est là encore fortement corrélée au jeu de données issues du monde réel.

Conclusions et perspectives

Nous affirmions lors de l'introduction de ce manuscrit qu'il était possible de générer automatiquement un modèle de simulation de flux dédié aux entreprises manufacturières, sur la base d'une collecte temps réel des trajectoires produits. Pour ce faire, et suite à une caractérisation du processus théorique de modélisation transformant des données collectées en une représentation des phénomènes physiques auxquels elles sont liées, nous avons précisé la notion de modélisation adaptative des systèmes manufacturiers par le biais des trajectoires-produit. Pour aborder cette problématique dans notre champ d'étude, nous avons dans un second temps mené une analyse bibliographique visant à préciser le concept de simulation et la notion de trajectoire spatio-temporelle, ainsi que ses usages dans des domaines scientifiques divers. Cette notion de trajectoire est en effet un des points clés de notre proposition. Peu de travaux de la littérature de production font explicitement référence à ces notions de trajectoire comme point d'entrée à la modélisation-identification des systèmes d'intérêt. Ceci explique en grande partie le choix de nous intéresser à des domaines scientifiques connexes lors de notre étude bibliographique sur le sujet. Il est ressorti de ces études que les trajectoires spatio-temporelles possédaient des capacités très intéressantes pour coder les relations structurelles du système à modéliser. L'alternative méthodologique choisie pour ces travaux, le Data Driven Simulation ou DDS, élargit les perspectives des processus classiques de modélisation et de maintenance des modèles pour la simulation. Elle ne peut cependant s'envisager que sur des systèmes déjà en fonctionnement, son intérêt étant d'autant plus important que le système à modéliser est complexe. Les propositions réalisées au chapitre 3 s'appuient sur une formalisation mathématique des trajectoires de chacun des exemplaires des produits en circulation dans le système observé. Le regroupement de ces trajectoires, selon un indicateur de similarité basé sur la distance spatio-temporelle les séparant, permet de définir un type de produit et de le caractériser. De manière indirecte, et comme nous l'avons déjà mentionné, cela nous permet de connaître les gammes et nomenclatures des produits, mais également le layout du système de production. Les principes mis en œuvre

pour la partie adaptation du modèle aux évolutions du système observé sont simples et pour l'essentiel liés à des phénomènes d'évaporation des traces laissées par les produits et les buffers-opérations après un certain temps. L'implémentation réalisée et les validations menées nous conduisent au final à affirmer qu'il est possible de générer automatiquement un modèle de simulation de flux dédié aux entreprises manufacturières, sur la base d'une collecte temps réel des trajectoires produits. La faisabilité théorique de notre approche originale est ainsi montrée. Elle nous semble être une alternative nouvelle et intéressante pour la modélisation et la simulation des systèmes complexes et évolutifs.

Bien sûr de nombreuses hypothèses subsistent, et en particulier sur les technologies à la source des flux de localisation des produits. Cependant on peut noter l'émergence actuelle d'une communauté scientifique très active dont l'objet d'étude est lié au positionnement Indoor. Nul doute que des progrès seront réalisés sur ces problématiques dans les prochaines années.

L'introduction de connaissances supplémentaires à celles déjà introduites pour la construction du modèle peut également permettre d'améliorer le réalisme de celui-ci. Cependant, l'ajout de ces connaissances particulières à un type de système particulier rendra la méthode proposée moins générique. La question de la généricité de l'approche pourrait alors se poser. Quoiqu'il en soit, les principales contributions scientifiques des travaux présentés dans ce manuscrit se situent dans :

- La proposition d'une approche originale de génération automatique de modèles de simulation de flux sur la base des données de localisation des produits dans le temps (ou trajectoires spatio-temporelles).
- L'étude bibliographique originale menée au regard du domaine d'étude concerné
- La méthode formelle de construction d'un modèle de simulation de flux, utilisant des graphes pour représenter les trajectoires des produits, et dont le méta-modèle est basé sur une ontologie minimaliste, très peu de connaissances *a priori* sur le système à modéliser étant requises.

On pourrait envisager différents types de perspectives à ces travaux. Le premier d'entre eux concerne les améliorations possibles à nos propositions. Parmi celles-ci, la première d'entre elles concerne la notion d'identifiant associé à chacun des composants constituant le produit final. Cette hypothèse, posée au début de nos travaux, pourrait à notre sens être levée. En effet, l'ensemble des trajectoires de chacun des composants du produit final (le graphe orienté), regroupe des informations spatio-temporelles suffisantes à l'identification finale du produit. La trajectoire peut alors apparaître comme une signature liée à chaque type de produit. Il n'est, dans ce cas, plus nécessaire de lier un identifiant à chacun des composants. Cette amélioration possible ne concerne uniquement que l'implémentation réalisée. La seconde amélioration possible concerne un type de connaissance qu'il est égale-

ment possible d'extraire du flux des données de localisation. L'observation détaillée des séquences d'arrivée et de départ des composants dans une file d'attente peut en effet permettre d'en extraire le comportement et donc la (ou les règles) de priorité avec laquelle la file d'attente est gérée. Ceci permettrait d'accéder à une connaissance sur les décisions locales (priority rules) prises au niveau des serveurs. Cette nature d'information pourrait compléter le modèle issu du générateur dans sa version actuelle. La deuxième famille de perspectives est plus large, et concerne en premier lieu la validation du générateur par rapport à des données réelles.

Le second chantier pourrait concerner le contrôle du système d'intérêt par le biais d'un couplage entre modèle d'évaluation (obtenu par le générateur) et l'optimisation d'une fonction de performance. En effet, le modèle de simulation obtenu par la générateur est de type champs de potentiel. Les metaheuristiques les plus récentes (ACO, APN) utilisent également ce type de modèle. La proximité conceptuelle de la partie évaluation et optimisation peut donc permettre d'accroître l'efficacité du couplage entre elles. Ce couplage constitue certainement une piste très intéressante de recherche très en lien avec nos travaux.

Au final, la problématique abordée dans ces travaux est assez proche des problématiques de modélisation-identification. La notion de trajectoire spatio-temporelle, sur laquelle nous nous sommes fortement appuyés pour nos propositions, peut, de par sa nature continue, constituer un lien (une passerelle) intéressante entre les mondes de l'automatique continue et des SED. Nul doute que c'est là une piste à explorer, nos travaux constituant une première tentative dans ce sens.

Annexes

A Résultats génération par scénarios

A.1 Scénario 1

Paramètres

meanWorkTime	13	processingSteps	3
individualArrivalMean	23	nmach	3
seed	12399	nagvs	300
size	(8, 8)	maxnumproducts	500
limX	(-10.0, 10.0)	epsilon	0.04
limY	(-10.0, 10.0)	endsim	142300.0
agvvel	0.44	limPeriodeLocSys	(0.5, 0.5)
nproductTypes	3	stdNoise	0.005

TABLE 14 – Paramètres Scénario 1

Histogrammes

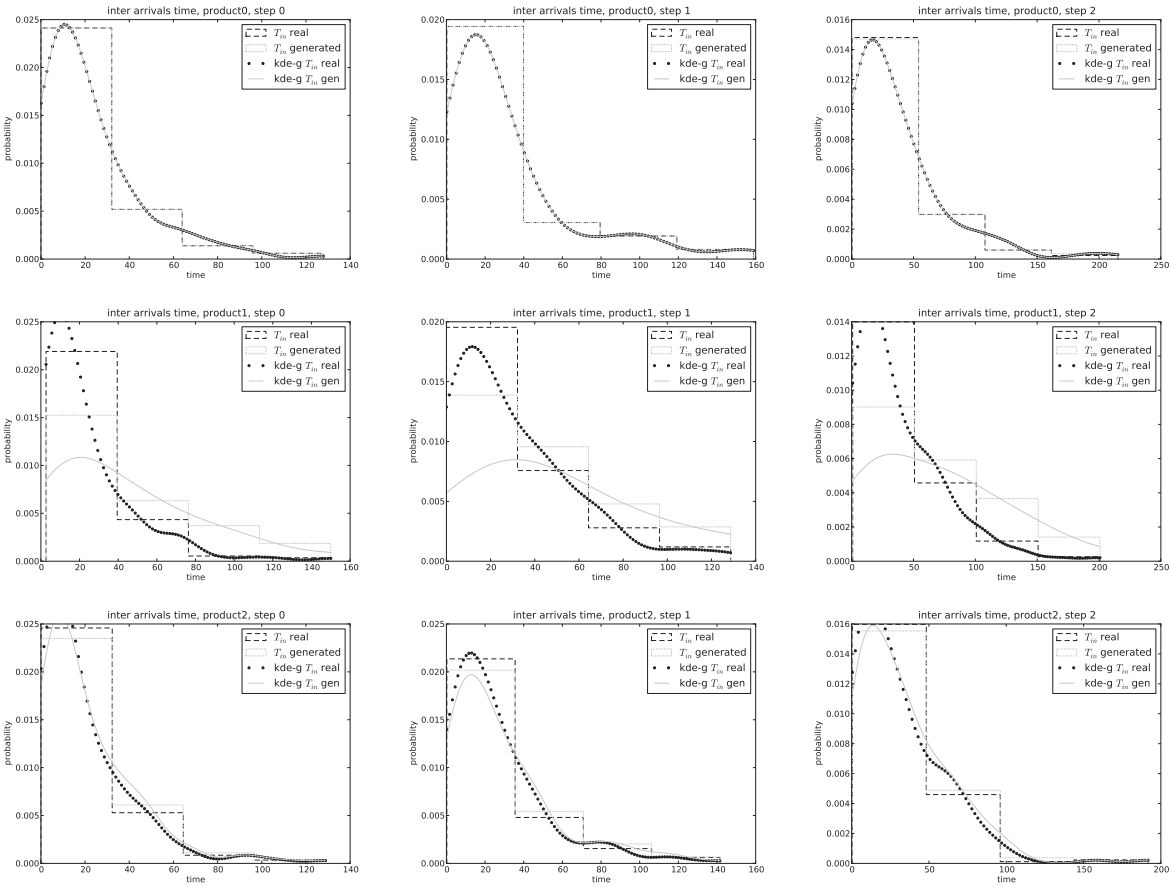


FIGURE 9 – Histogrammes temps d'inter-arrivées, Scénario 1

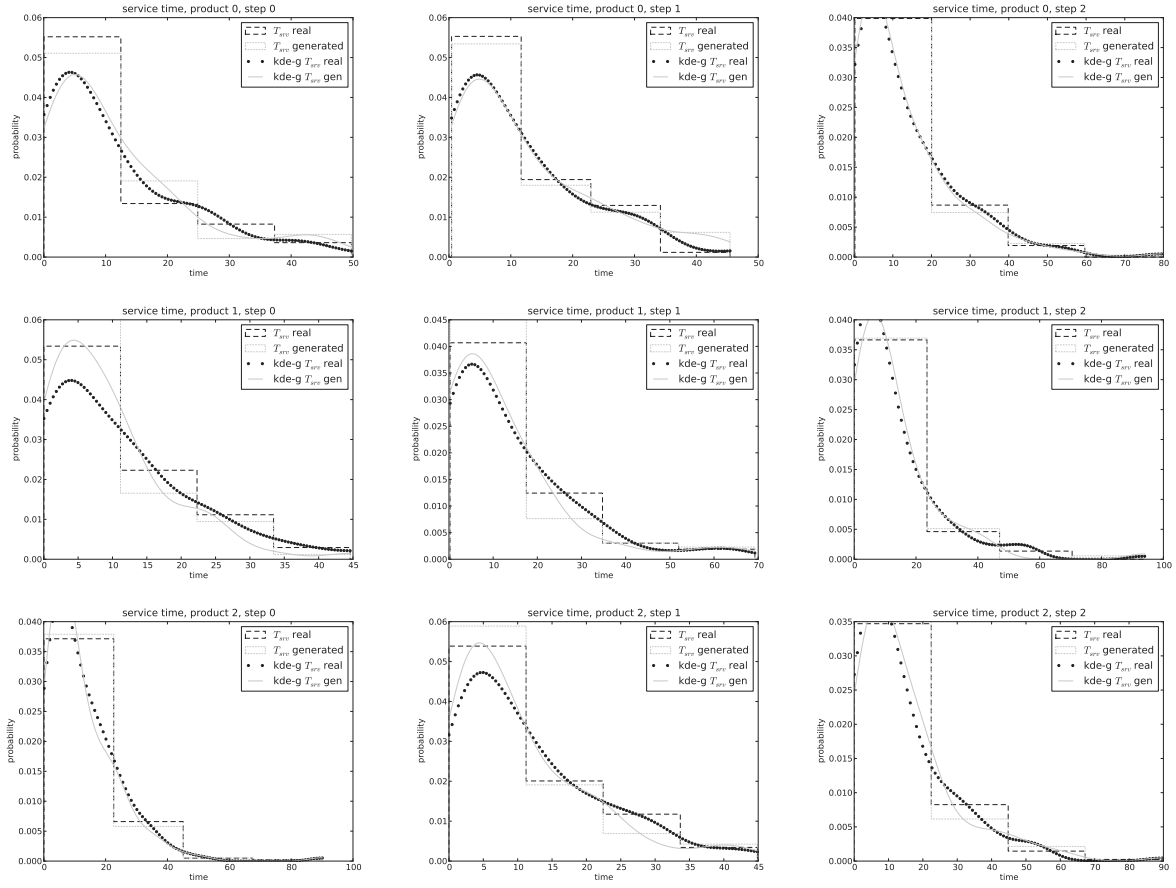


FIGURE 10 – Histogrammes temps de service, Scénario 1

A.2 Scénario 2

Paramètres

meanWorkTime	13	processingSteps	8
individualArrivalMean	23	nmach	20
seed	12399	nagvs	300
size	(8, 8)	maxnumproducts	500
limX	(-10.0, 10.0)	epsilon	0.04
limY	(-10.0, 10.0)	endsim	142300.0
agvvel	0.44	limPeriodeLocSys	(0.5, 0.5)
nproductTypes	13	stdNoise	0.005

TABLE 15 – Paramètres Scénario 2

Tableaux de test statistiques

Gamme	A	B	C	D	E	F	G	H
Coordonnée *x* des «buffer-opération»	-2.98	-2.171	2.833	6.53	-5.175	-2.993	1.929	-0.721
Coordonnée *x* des «buffer-opération» généré	-2.98	-2.171	2.833	6.53	-5.175	-2.993	1.929	-0.721
Coordonnée *y* des «buffer-opération»	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677	-6.505
Coordonnée *y* des «buffer-opération» généré	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677	-6.505
Temps inter-arrivées moyen	24.808	41.119	42.906	47.514	49.629	50.898	51.507	51.676
Temps inter-arrivées moyen généré	24.808	41.119	42.906	47.514	49.629	50.898	51.507	51.676
Temps inter-arrivées écart type	21.307	43.859	41.119	33.812	42.621	47.984	50.709	50.454
Temps inter-arrivées écart type généré	21.307	43.859	41.106	33.816	42.621	48.013	50.713	50.454
Wilcoxon t	128.0	26.0	10.0	26.5	2.0	13.5	10.5	1.5
Wilcoxon p	0.0	0.509	0.114	0.672	0.593	0.0	1.0	0.0
Kolmogorov-Smirnov d	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	9.904	9.768	12.634	10.208	11.823	12.078	9.871	10.345
Temps de service moyen généré	9.776	10.414	12.64	10.089	12.021	14.238	9.633	10.952
Temps de service écart type	10.575	9.518	11.678	8.728	12.349	15.393	8.831	11.199
Temps de service écart type généré	11.244	10.633	12.001	8.735	14.653	17.487	8.693	11.093
Wilcoxon t	272.0	225.0	168.5	175.0	268.0	225.0	93.0	205.5
Wilcoxon p	0.879	0.171	0.02	0.059	0.823	0.321	0.001	0.086
Kolmogorov-Smirnov d	0.121	0.121	0.061	0.061	0.091	0.152	0.091	0.061
Kolmogorov-Smirnov p	0.957	0.957	1.0	1.0	0.999	0.811	0.999	1.0

TABLE 16 – Résultats Scénario 2, produit 0

Gamme	T	A	B	C	D	E	F	G
Coordonnée *x* des «buffer-opération»	0.591	-2.98	-2.171	2.833	6.53	-5.175	-2.993	1.929
Coordonnée *x* des «buffer-opération» généré	0.591	-2.98	-2.171	2.833	6.53	-5.175	-2.993	1.929
Coordonnée *y* des «buffer-opération»	-6.946	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677
Coordonnée *y* des «buffer-opération» généré	-6.946	0.001	2.434	-5.919	-0.402	3.428	0.159	-3.677
Temps inter-arrivées moyen	17.015	30.981	36.063	39.088	41.166	44.282	45.91	45.623
Temps inter-arrivées moyen généré	17.015	30.981	36.063	39.088	41.166	44.282	45.91	45.623
Temps inter-arrivées écart type	18.176	33.278	40.633	40.644	48.163	48.045	44.429	47.866
Temps inter-arrivées écart type généré	18.176	33.278	40.633	40.644	48.163	48.045	44.426	47.871
Wilcoxon t	102.0	148.0	31.5	20.5	29.0	13.0	13.5	17.0
Wilcoxon p	0.0	0.0	0.639	0.0	0.09	0.741	0.85	0.0
Kolmogorov-Smirnov d	0.022	0.022	0.022	0.022	0.022	0.022	0.022	0.022
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.772	12.73	14.126	12.553	14.666	15.336	11.825	9.888
Temps de service moyen généré	12.713	12.555	13.084	12.471	14.596	15.481	12.806	10.154
Temps de service écart type	14.552	20.343	13.369	11.213	15.446	12.449	11.332	8.602
Temps de service écart type généré	14.656	19.965	11.773	11.223	15.474	12.87	12.567	9.07
Wilcoxon t	495.0	438.5	441.0	367.5	310.0	529.0	536.0	369.5
Wilcoxon p	0.649	0.119	0.126	0.02	0.004	0.9	0.923	0.035
Kolmogorov-Smirnov d	0.065	0.174	0.109	0.043	0.065	0.109	0.087	0.087
Kolmogorov-Smirnov p	1.0	0.452	0.935	1.0	1.0	0.935	0.993	0.993

TABLE 17 – Résultats Scénario 2, produit 1

Gamme	S	T	A	B	C	D	E	F
Coordonnée *x* des «buffer-opération»	5.881	0.591	-2.98	-2.171	2.833	6.53	-5.175	-2.993
Coordonnée *x* des «buffer-opération» généré	5.881	0.591	-2.98	-2.171	2.833	6.53	-5.175	-2.993
Coordonnée *y* des «buffer-opération»	-0.664	-6.946	0.001	2.434	-5.919	-0.402	3.428	0.159
Coordonnée *y* des «buffer-opération» généré	-0.664	-6.946	0.001	2.434	-5.919	-0.402	3.428	0.159
Temps inter-arrivées moyen	22.588	35.451	50.604	59.311	60.923	66.32	66.327	66.801
Temps inter-arrivées moyen généré	22.588	35.451	50.604	59.311	60.923	66.32	66.327	66.801
Temps inter-arrivées écart type	26.801	49.115	83.267	112.301	115.192	115.596	128.372	129.221
Temps inter-a rrivées écart type généré	26.801	49.115	83.267	112.301	115.195	115.594	128.372	129.221
Wilcoxon t	148.0	83.0	140.5	7.0	17.5	2.5	1.0	4.0
Wilcoxon p	0.825	0.102	0.182	0.0	0.0	0.0	0.655	0.0
Kolmogorov-Smirnov d	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	14.634	12.447	15.222	10.29	14.224	11.389	13.405	14.463
Temps de service moyen généré	14.623	12.78	15.567	11.296	14.092	11.314	14.275	16.069
Temps de service écart type	12.113	10.344	14.629	12.594	13.601	9.047	13.052	18.72
Temps de service écart type généré	12.122	10.209	15.334	11.973	13.314	9.055	12.994	20.563
Wilcoxon t	264.0	313.5	303.5	259.0	269.5	240.0	295.0	323.0
Wilcoxon p	0.366	0.961	0.451	0.245	0.165	0.064	0.743	0.875
Kolmogorov-Smirnov d	0.028	0.083	0.139	0.139	0.056	0.056	0.139	0.222
Kolmogorov-Smirnov p	1.0	0.999	0.851	0.851	1.0	1.0	0.851	0.297

TABLE 18 – Résultats Scénario 2, produit 2

Gamme	R	S	T	A	B	C	D	E
Coordonnée *x* des «buffer-opération»	-5.06	5.881	0.591	-2.98	-2.171	2.833	6.53	-5.175
Coordonnée *x* des «buffer-opération» généré	-5.06	5.881	0.591	-2.98	-2.171	2.833	6.53	-5.175
Coordonnée *y* des «buffer-opération»	-6.809	-0.664	-6.946	0.001	2.434	-5.919	-0.402	3.428
Coordonnée *y* des «buffer-opération» généré	-6.809	-0.664	-6.946	0.001	2.434	-5.919	-0.402	3.428
Temps inter-arrivées moyen	20.372	30.908	42.515	54.223	59.938	60.904	65.926	66.312
Temps inter-arrivées moyen généré	20.372	30.908	42.515	54.223	59.938	60.904	65.926	66.312
Temps inter-arrivées écart type	20.108	39.043	48.842	50.116	72.035	71.633	88.838	81.935
Temps inter-arrivées écart type généré	20.108	39.043	48.842	50.116	72.035	71.633	88.838	81.935
Wilcoxon t	119.0	171.5	193.5	65.0	10.0	10.0	37.0	1.5
Wilcoxon p	0.0	0.778	0.0	0.0	0.114	0.0	0.698	0.0
Kolmogorov-Smirnov d	0.024	0.024	0.024	0.024	0.024	0.024	0.024	0.024
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.107	11.982	14.191	12.651	12.995	9.134	13.831	15.07
Temps de service moyen généré	12.217	11.944	14.575	13.663	12.338	9.08	13.771	17.353
Temps de service écart type	11.435	10.689	10.169	10.428	13.452	11.964	12.396	15.005
Temps de service écart type généré	11.597	10.686	10.575	11.521	13.904	11.372	12.328	16.01
Wilcoxon t	381.5	265.0	289.0	391.5	423.0	279.0	351.0	342.0
Wilcoxon p	0.2	0.012	0.015	0.257	0.722	0.011	0.091	0.171
Kolmogorov-Smirnov d	0.071	0.048	0.071	0.071	0.143	0.119	0.024	0.167
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	0.752	0.91	1.0	0.564

TABLE 19 – Résultats Scénario 2, produit 3

Gamme	Q	R	S	T	A	B	C	D
Coordonnée *x* des «buffer-opération»	6.5	-5.06	5.881	0.591	-2.98	-2.171	2.833	6.53
Coordonnée *x* des «buffer-opération» généré	6.5	-5.06	5.881	0.591	-2.98	-2.171	2.833	6.53
Coordonnée *y* des «buffer-opération»	-4.509	-6.809	-0.664	-6.946	0.001	2.434	-5.919	-0.402
Coordonnée *y* des «buffer-opération» généré	-4.509	-6.809	-0.664	-6.946	0.001	2.434	-5.919	-0.402
Temps inter-arrivées moyen	29.734	44.523	57.363	78.578	86.632	98.206	96.751	98.189
Temps inter-arrivées moyen généré	29.734	44.523	57.363	78.578	86.632	98.206	96.751	98.189
Temps inter-arrivées écart type	30.552	42.127	58.738	85.197	129.637	158.6	152.073	155.796
Temps inter-arrivées écart type généré	30.552	42.127	58.738	85.197	129.637	158.6	152.073	155.796
Wilcoxon t	104.5	108.0	137.0	48.5	26.5	0.0	0.0	0.0
Wilcoxon p	0.393	0.342	0.0	0.166	0.0	0.18	0.317	0.317
Kolmogorov-Smirnov d	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	11.75	13.139	12.696	9.763	9.25	8.662	14.265	12.94
Temps de service moyen généré	11.736	12.83	12.696	9.581	9.357	10.56	14.017	12.802
Temps de service écart type	11.566	13.42	9.325	9.349	8.204	8.276	15.021	10.52
Temps de service écart type généré	11.531	13.221	9.325	9.289	8.115	10.301	15.279	10.439
Wilcoxon t	174.5	183.5	210.5	197.0	216.0	177.0	190.0	67.0
Wilcoxon p	0.322	0.285	0.78	0.658	0.974	0.381	0.365	0.001
Kolmogorov-Smirnov d	0.034	0.103	0.034	0.103	0.138	0.103	0.069	0.069
Kolmogorov-Smirnov p	1.0	0.996	1.0	0.996	0.927	0.996	1.0	1.0

TABLE 20 – Résultats Scénario 2, produit 4

Gamme	P	Q	R	S	T	A	B	C
Coordonnée *x* des «buffer-opération»	-0.466	6.5	-5.06	5.881	0.591	-2.98	-2.171	2.833
Coordonnée *x* des «buffer-opération» généré	-0.466	6.5	-5.06	5.881	0.591	-2.98	-2.171	2.833
Coordonnée *y* des «buffer-opération»	-4.766	-4.509	-6.809	-0.664	-6.946	0.001	2.434	-5.919
Coordonnée *y* des «buffer-opération» généré	-4.766	-4.509	-6.809	-0.664	-6.946	0.001	2.434	-5.919
Temps inter-arrivées moyen	20.803	29.869	43.795	49.569	61.994	66.905	66.123	66.478
Temps inter-arrivées moyen généré	20.803	29.869	43.795	49.569	61.994	66.905	66.123	66.478
Temps inter-arrivées écart type	23.712	30.504	39.638	57.331	82.484	127.889	99.888	108.946
Temps inter-arrivées écart type généré	23.712	30.504	39.638	57.331	82.484	127.889	99.89	108.946
Wilcoxon t	238.5	144.5	161.0	146.0	21.5	6.0	2.5	4.0
Wilcoxon p	0.0	0.0	0.0	0.0	0.0	0.07	0.0	0.0
Kolmogorov-Smirnov d	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	14.183	14.07	13.645	13.957	11.729	17.361	13.264	15.657
Temps de service moyen généré	14.219	14.06	13.865	13.957	11.613	16.441	14.694	15.968
Temps de service écart type	16.54	13.054	12.422	15.407	8.83	19.55	11.391	12.62
Temps de service écart type généré	16.569	13.046	12.574	15.407	8.811	19.479	11.435	12.169
Wilcoxon t	349.0	378.0	378.5	315.5	403.0	321.0	352.0	401.0
Wilcoxon p	0.366	0.473	0.481	0.147	0.857	0.336	0.436	0.819
Kolmogorov-Smirnov d	0.05	0.025	0.1	0.025	0.075	0.1	0.15	0.125
Kolmogorov-Smirnov p	1.0	1.0	0.983	1.0	1.0	0.983	0.724	0.893

TABLE 21 – Résultats Scénario 2, produit 5

Gamme	O	P	Q	R	S	T	A	B
Coordonnée *x* des «buffer-opération»	0.979	-0.466	6.5	-5.06	5.881	0.591	-2.98	-2.171
Coordonnée *x* des «buffer-opération» généré	0.979	-0.466	6.5	-5.06	5.881	0.591	-2.98	-2.171
Coordonnée *y* des «buffer-opération»	1.796	-4.766	-4.509	-6.809	-0.664	-6.946	0.001	2.434
Coordonnée *y* des «buffer-opération» généré	1.796	-4.766	-4.509	-6.809	-0.664	-6.946	0.001	2.434
Temps inter-arrivées moyen	25.481	51.464	65.293	76.143	84.983	89.545	95.201	87.739
Temps inter-arrivées moyen généré	25.481	51.464	65.293	76.143	84.983	89.545	95.201	87.739
Temps inter-arrivées écart type	28.608	55.51	63.435	52.984	71.968	99.421	125.086	121.526
Temps inter-arrivées écart type généré	28.608	55.51	63.435	52.984	71.968	99.421	125.086	121.526
Wilcoxon t	106.5	116.0	80.5	35.5	52.5	5.0	4.0	0.0
Wilcoxon p	0.065	0.258	0.0	0.498	0.0	0.0	0.715	0.0
Kolmogorov-Smirnov d	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.0
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	15.129	12.23	17.209	11.477	11.071	10.245	12.94	10.483
Temps de service moyen généré	16.083	11.51	17.209	10.601	11.071	10.729	11.752	11.883
Temps de service écart type	18.302	12.055	13.589	11.136	9.702	7.33	10.074	8.568
Temps de service écart type généré	19.462	12.263	13.589	11.163	9.702	7.052	8.142	10.431
Wilcoxon t	231.0	190.5	260.5	224.0	242.5	216.0	200.0	269.0
Wilcoxon p	0.208	0.049	0.544	0.165	0.308	0.249	0.07	0.837
Kolmogorov-Smirnov d	0.091	0.091	0.03	0.182	0.03	0.152	0.152	0.121
Kolmogorov-Smirnov p	0.999	0.999	1.0	0.601	1.0	0.811	0.811	0.957

TABLE 22 – Résultats Scénario 2, produit 6

Gamme	N	O	P	Q	R	S	T	A
Coordonnée *x* des «buffer-opération»	-1.232	0.979	-0.466	6.5	-5.06	5.881	0.591	-2.98
Coordonnée *x* des «buffer-opération» généré	-1.232	0.979	-0.466	6.5	-5.06	5.881	0.591	-2.98
Coordonnée *y* des «buffer-opération»	-4.65	1.796	-4.766	-4.509	-6.809	-0.664	-6.946	0.001
Coordonnée *y* des «buffer-opération» généré	-4.65	1.796	-4.766	-4.509	-6.809	-0.664	-6.946	0.001
Temps inter-arrivées moyen	17.448	26.852	44.08	53.981	58.69	65.447	67.872	70.316
Temps inter-arrivées moyen généré	17.448	26.852	44.08	53.981	58.69	65.447	67.872	70.316
Temps inter-arrivées écart type	16.548	22.857	39.23	46.284	61.879	79.16	101.564	122.496
Temps inter-arrivées écart type généré	16.548	22.857	39.23	46.284	61.879	79.16	101.564	122.496
Wilcoxon t	127.5	292.5	232.0	170.0	44.5	27.0	9.5	20.5
Wilcoxon p	0.121	0.409	0.0	0.392	0.648	0.0	0.674	0.0
Kolmogorov-Smirnov d	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	11.317	15.245	14.415	13.665	16.133	13.03	15.652	11.074
Temps de service moyen généré	11.016	14.148	14.292	13.665	16.675	13.03	14.592	12.321
Temps de service écart type	12.6	15.417	14.982	13.883	16.821	13.535	16.987	9.526
Temps de service écart type généré	12.147	15.991	15.11	13.883	16.253	13.535	16.212	9.731
Wilcoxon t	507.5	463.0	492.5	348.5	550.0	368.0	516.0	468.0
Wilcoxon p	0.215	0.081	0.156	0.005	0.781	0.008	0.46	0.091
Kolmogorov-Smirnov d	0.083	0.125	0.063	0.021	0.104	0.021	0.104	0.125
Kolmogorov-Smirnov p	0.995	0.822	1.0	1.0	0.946	1.0	0.946	0.822

TABLE 23 – Résultats Scénario 2, produit 7

Gamme	M	N	O	P	Q	R	S	T
Coordonnée *x* des «buffer-opération»	0.449	-1.232	0.979	-0.466	6.5	-5.06	5.881	0.591
Coordonnée *x* des «buffer-opération» généré	0.449	-1.232	0.979	-0.466	6.5	-5.06	5.881	0.591
Coordonnée *y* des «buffer-opération»	1.94	-4.65	1.796	-4.766	-4.509	-6.809	-0.664	-6.946
Coordonnée *y* des «buffer-opération» généré	1.94	-4.65	1.796	-4.766	-4.509	-6.809	-0.664	-6.946
Temps inter-arrivées moyen	25.081	42.67	56.946	71.925	87.992	96.252	103.485	101.738
Temps inter-arrivées moyen généré	25.081	42.67	56.946	71.925	87.992	96.252	103.485	101.738
Temps inter-arrivées écart type	20.163	38.004	43.892	80.824	104.426	111.615	130.978	163.689
Temps inter-arrivées écart type généré	20.163	38.004	43.892	80.824	104.426	111.615	130.978	163.689
Wilcoxon t	102.0	125.0	152.5	36.5	35.5	7.0	49.0	7.0
Wilcoxon p	0.0	0.896	0.487	0.199	0.0	0.0	0.0	0.0
Kolmogorov-Smirnov d	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.018	16.781	13.054	10.943	15.898	13.079	14.575	12.27
Temps de service moyen généré	11.227	16.887	13.57	11.136	15.009	11.532	14.737	13.355
Temps de service écart type	9.743	15.691	10.603	10.988	11.843	11.157	12.515	8.122
Temps de service écart type généré	9.586	14.966	11.638	10.764	12.24	9.179	12.373	9.194
Wilcoxon t	229.0	264.0	259.0	253.5	274.0	293.0	246.0	244.0
Wilcoxon p	0.195	1.0	0.926	0.267	0.505	0.939	0.209	0.36
Kolmogorov-Smirnov d	0.206	0.147	0.147	0.088	0.118	0.147	0.059	0.147
Kolmogorov-Smirnov p	0.422	0.825	0.825	0.999	0.962	0.825	1.0	0.825

TABLE 24 – Résultats Scénario 2, produit 8

Gamme	L	M	N	O	P	Q	R	S
Coordonnée *x* des «buffer-opération»	-2.991	0.449	-1.232	0.979	-0.466	6.5	-5.06	5.881
Coordonnée *x* des «buffer-opération» généré	-2.991	0.449	-1.232	0.979	-0.466	6.5	-5.06	5.881
Coordonnée *y* des «buffer-opération»	5.36	1.94	-4.65	1.796	-4.766	-4.509	-6.809	-0.664
Coordonnée *y* des «buffer-opération» généré	5.36	1.94	-4.65	1.796	-4.766	-4.509	-6.809	-0.664
Temps inter-arrivées moyen	19.803	28.369	41.322	51.984	65.924	77.02	79.517	78.125
Temps inter-arrivées moyen généré	19.803	28.369	41.322	51.984	65.924	77.02	79.517	78.125
Temps inter-arrivées écart type	20.483	25.035	38.598	45.853	62.481	86.094	107.962	124.666
Temps inter-arrivées écart type généré	20.483	25.035	38.598	45.853	62.481	86.094	107.962	124.666
Wilcoxon t	135.0	197.0	278.0	187.0	45.5	13.0	2.5	95.0
Wilcoxon p	0.0	0.503	0.89	0.899	0.0	0.0	0.0	0.0
Kolmogorov-Smirnov d	0.024	0.024	0.024	0.024	0.024	0.024	0.024	0.024
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.642	11.748	12.486	11.643	13.866	12.736	13.66	11.888
Temps de service moyen généré	12.699	12.592	10.239	12.067	14.927	12.582	14.68	11.818
Temps de service écart type	15.042	10.965	15.33	14.37	12.752	10.826	10.888	12.055
Temps de service écart type généré	15.914	10.885	10.577	14.039	13.76	10.365	12.796	12.115
Wilcoxon t	455.5	269.5	421.0	346.0	372.0	385.5	447.0	375.0
Wilcoxon p	0.701	0.008	0.325	0.049	0.32	0.138	0.754	0.237
Kolmogorov-Smirnov d	0.07	0.14	0.093	0.093	0.116	0.07	0.163	0.093
Kolmogorov-Smirnov p	1.0	0.765	0.989	0.989	0.917	1.0	0.58	0.989

TABLE 25 – Résultats Scénario 2, produit 9

Gamme	K	L	M	N	O	P	Q	R
Coordonnée *x* des «buffer-opération»	2.043	-2.991	0.449	-1.232	0.979	-0.466	6.5	-5.06
Coordonnée *x* des «buffer-opération» généré	2.043	-2.991	0.449	-1.232	0.979	-0.466	6.5	-5.06
Coordonnée *y* des «buffer-opération»	-4.561	5.36	1.94	-4.65	1.796	-4.766	-4.509	-6.809
Coordonnée *y* des «buffer-opération» généré	-4.561	5.36	1.94	-4.65	1.796	-4.766	-4.509	-6.809
Temps inter-arrivées moyen	28.774	44.008	54.329	76.234	92.386	105.274	115.566	114.542
Temps inter-arrivées moyen généré	28.774	44.008	54.329	76.234	92.386	105.274	115.566	114.542
Temps inter-arrivées écart type	32.381	38.984	53.601	64.869	85.477	121.331	134.002	144.048
Temps inter-arrivées écart type généré	32.381	38.984	53.601	64.869	85.477	121.331	134.002	144.048
Wilcoxon t	113.0	66.0	126.0	67.5	13.5	31.0	30.5	20.5
Wilcoxon p	0.775	0.0	0.947	0.0	0.85	0.282	0.0	0.0
Kolmogorov-Smirnov d	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.785	13.992	17.718	13.967	10.981	14.009	14.585	13.503
Temps de service moyen généré	12.731	14.648	19.16	13.536	10.944	14.429	14.308	15.795
Temps de service écart type	13.073	11.527	17.974	13.231	10.842	12.053	11.509	14.505
Temps de service écart type généré	13.139	11.636	17.881	12.544	9.662	13.151	11.408	17.154
Wilcoxon t	193.5	151.0	140.0	179.0	181.5	197.0	200.0	207.0
Wilcoxon p	0.415	0.236	0.094	0.405	0.263	0.476	0.705	0.82
Kolmogorov-Smirnov d	0.069	0.138	0.172	0.138	0.138	0.138	0.103	0.172
Kolmogorov-Smirnov p	1.0	0.927	0.741	0.927	0.927	0.927	0.996	0.741

TABLE 26 – Résultats Scénario 2, produit 10

Gamme	J	K	L	M	N	O	P	Q
Coordonnée *x* des «buffer-opération»	5.428	2.043	-2.991	0.449	-1.232	0.979	-0.466	6.5
Coordonnée *x* des «buffer-opération» généré	5.428	2.043	-2.991	0.449	-1.232	0.979	-0.466	6.5
Coordonnée *y* des «buffer-opération»	6.809	-4.561	5.36	1.94	-4.65	1.796	-4.766	-4.509
Coordonnée *y* des «buffer-opération» généré	6.809	-4.561	5.36	1.94	-4.65	1.796	-4.766	-4.509
Temps inter-arrivées moyen	20.081	29.826	39.041	45.342	63.094	71.36	74.56	71.203
Temps inter-arrivées moyen généré	20.081	29.826	39.041	45.342	63.094	71.36	74.56	71.203
Temps inter-arrivées écart type	21.066	33.674	34.365	41.955	56.518	72.744	76.918	73.887
Temps inter-arrivées écart type généré	21.066	33.674	34.365	41.955	56.518	72.744	76.918	73.887
Wilcoxon t	150.0	49.5	157.0	96.5	36.0	68.0	49.0	42.0
Wilcoxon p	0.304	0.0	0.0	0.0	0.351	0.0	0.0	0.0
Kolmogorov-Smirnov d	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	12.385	10.984	12.157	15.685	13.572	15.564	11.907	15.655
Temps de service moyen généré	12.364	10.925	11.6	15.303	14.618	15.531	11.972	17.082
Temps de service écart type	11.788	8.303	9.65	15.438	12.29	14.327	10.683	14.585
Temps de service écart type généré	11.788	8.368	9.548	15.347	12.271	14.874	9.596	15.667
Wilcoxon t	363.0	229.5	380.0	343.0	283.0	325.0	372.0	241.0
Wilcoxon p	0.517	0.01	0.794	0.313	0.204	0.364	0.655	0.038
Kolmogorov-Smirnov d	0.026	0.077	0.128	0.103	0.103	0.103	0.154	0.103
Kolmogorov-Smirnov p	1.0	1.0	0.884	0.981	0.981	0.981	0.708	0.981

TABLE 27 – Résultats Scénario 2, produit 11

Gamme	I	J	K	L	M	N	O	P
Coordonnée *x* des «buffer-opération»	-4.24	5.428	2.043	-2.991	0.449	-1.232	0.979	-0.466
Coordonnée *x* des «buffer-opération» généré	-4.24	5.428	2.043	-2.991	0.449	-1.232	0.979	-0.466
Coordonnée *y* des «buffer-opération»	-6.256	6.809	-4.561	5.36	1.94	-4.65	1.796	-4.766
Coordonnée *y* des «buffer-opération» généré	-6.256	6.809	-4.561	5.36	1.94	-4.65	1.796	-4.766
Temps inter-arrivées moyen	17.064	16.694	23.728	31.206	35.582	47.859	55.996	51.675
Temps inter-arrivées moyen généré	17.064	16.694	23.728	31.206	35.582	47.859	55.996	51.675
Temps inter-arrivées écart type	16.993	16.049	23.018	31.384	44.976	75.785	93.623	72.936
Temps inter-arrivées écart type généré	16.993	16.049	23.018	31.384	44.976	75.785	93.623	72.936
Wilcoxon t	92.0	320.5	85.0	296.0	369.5	45.5	42.5	145.0
Wilcoxon p	0.715	0.0	0.0	0.0	0.0	0.0	0.185	0.0
Kolmogorov-Smirnov d	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
Kolmogorov-Smirnov p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Temps de service moyen	15.395	13.803	14.513	10.849	12.986	13.398	13.139	14.785
Temps de service moyen généré	15.773	13.795	14.593	11.264	12.429	15.058	13.077	14.305
Temps de service écart type	14.285	15.346	14.203	13.719	12.045	11.527	9.337	14.405
Temps de service écart type généré	14.151	15.351	14.221	13.217	10.794	14.402	8.913	12.864
Wilcoxon t	517.0	485.5	522.0	397.0	535.5	551.0	486.0	529.0
Wilcoxon p	0.263	0.211	0.29	0.017	0.596	0.704	0.217	0.545
Kolmogorov-Smirnov d	0.125	0.021	0.042	0.125	0.104	0.083	0.104	0.083
Kolmogorov-Smirnov p	0.822	1.0	1.0	0.822	0.946	0.995	0.946	0.995

TABLE 28 – Résultats Scénario 2, produit 12

A.3 Scénario 3

Histogrammes

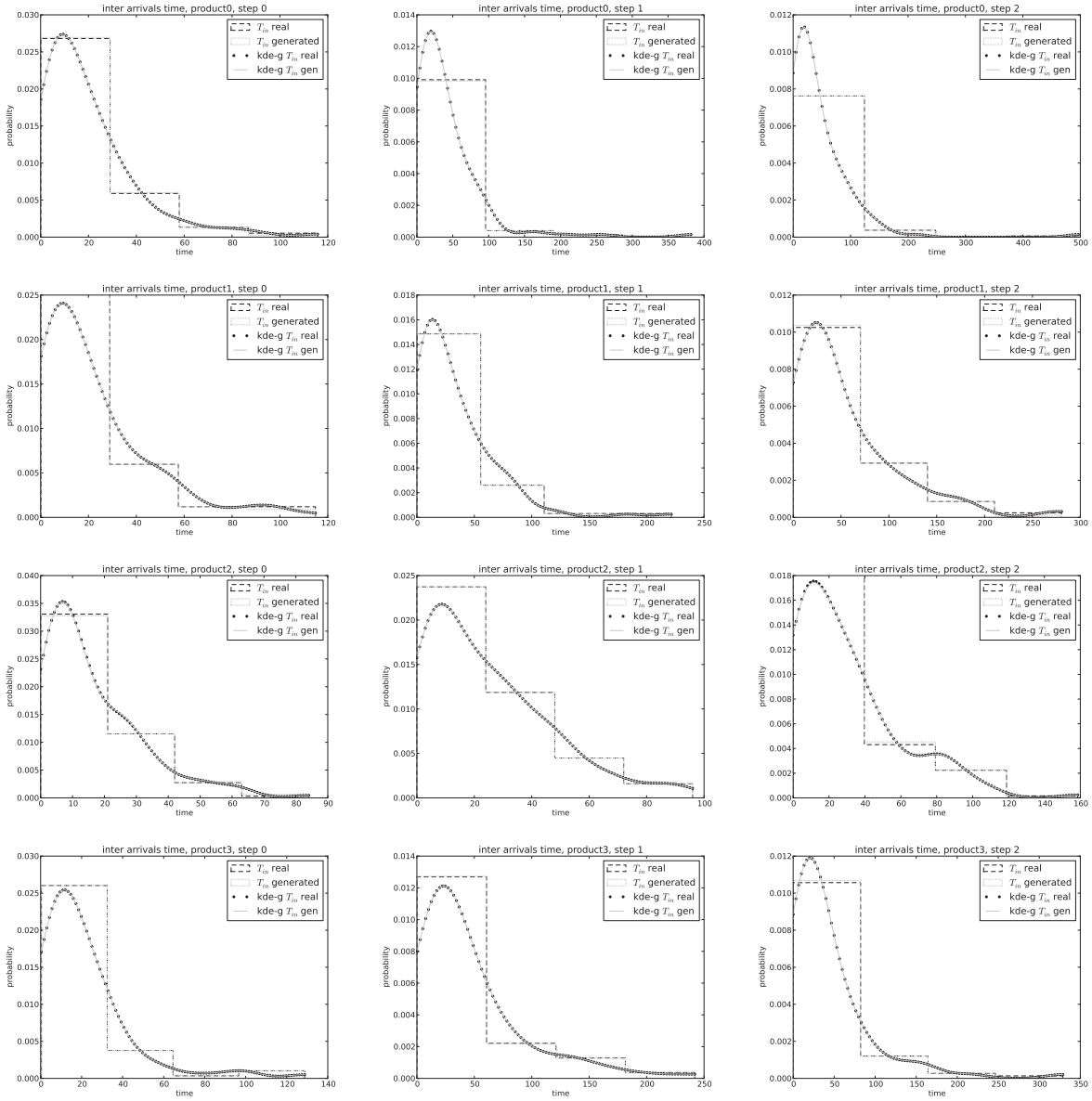


FIGURE 11 – Histogrammes temps d'inter-arrivées, Scénario 3

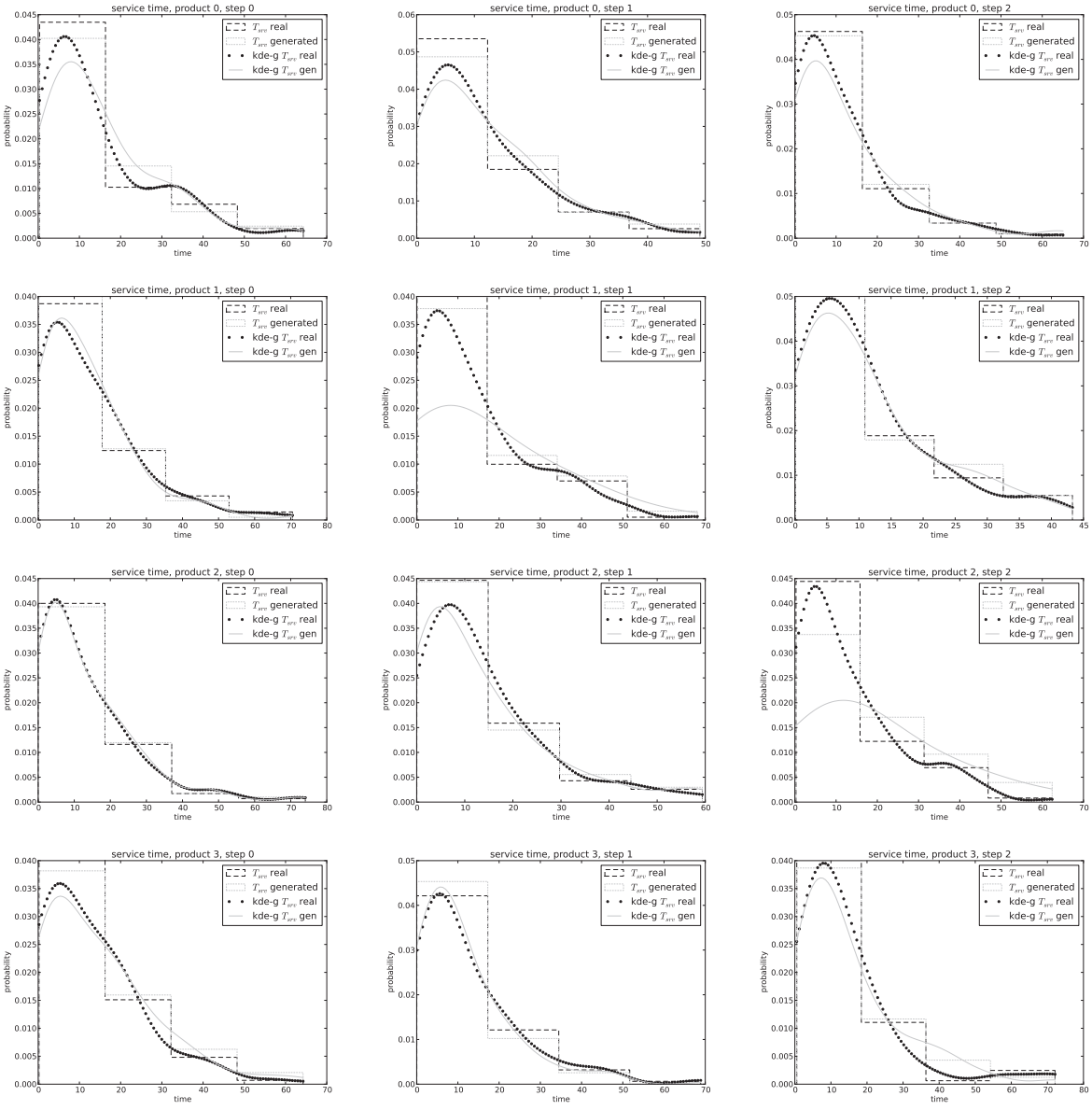


FIGURE 12 – Histogrammes temps de service, Scénario 3

B Génération de données

```
# -*- coding: utf-8 -*-
''' FMS simulation with SimPy. Process Part.
    Andres Vejar 2008. '''
import SimPy.Simulation as Sim
from manu import Point, Print, Scheme, LocSys
from resources import Machine, Agv
from stream import savePickle
import random

class Sources(Sim.Process,Scheme)
    ''' Plant products, Source/Arrivals type process.'''
    def __init__(self):
        Scheme.__init__(self)
        Sim.Process.__init__(self)
        self.ProdTypes = []
        self.products = []
        self.machines = []
        self.agvs = []
        self.plot = True

    def makeseed(self):
        random.seed(self.seed)

    def productTypes(self):
        S = self.processingSteps # Num process steps
        M = self.nmach # Num Machines
        ##
        # TypeP = [[0,1,2],[2,0,1],[1,2,0],[0,1,3]]
        for l in range(self.nproductTypes):
            TypeP = map(lambda s,m: (s-l)%m, range(S), S*[M])
            self.ProdTypes.append(TypeP)

    def machineCreation(self):
        # creating a ResourceSon objects
        maxlenlabel = 4
        (Lx,Ux) = self.limX
        (Ly,Uy) = self.limY
        (Lx,Ux) = (Lx+3,Ux-3)
        (Ly,Uy) = (Ly+3,Uy-3)

        mwt = self.meanWorkTime
        for k in range(self.nmach+1):
            px = random.uniform(Lx,Ux)
            py = random.uniform(Ly,Uy)
            pos = (px,py)

            if k == self.nmach:
                pos = (Ux-2,(Uy+Ly)/2.0)
                m = Machine(name='exit',code=-1,position=pos,\
                    meanWorkTime=0.0)
            else:
                machname = 'm-'+str(k).zfill(maxlenlabel)
                m = Machine(name=machname,code=k,position=pos,meanWorkTime=mwt)
            self.machines.append(m)

    def agvCreation(self):
        # creating a ResourceSon objects
        # creating a Agv object
        maxlenlabel = 4
        (Lx,Ux) = self.limX
        (Ly,Uy) = self.limY
        for k in range(self.nagvs):
            px = random.uniform(Lx,Ux)
            py = random.uniform(Ly,Uy)
            pos = (px,py)
            agvname = 'agv-'+str(k).zfill(maxlenlabel)
            a = Agv(name=agvname,code=k,position=pos,velocity=self.agvvel)

            # put the agv in the plant
            self.agvs.append(a)

```

```
72      def changingMsg(self,toconsole,tofile):
73          self.toconsole = toconsole
74          self.tofile = tofile
75
76      def pem(self):
77          maxlenlabel = 4
78          (Lx,Ux) = self.limX
79          (Ly,Uy) = self.limY
80
81          # px = random.uniform(Lx,Ux)
82          # py = random.uniform(Ly,Uy)
83
84          px = Lx+2
85          py = (Uy+Ly)/2.0
86          pos = (px,py)
87
88          (LP,UP) = self.limPeriodeLocSys
89
90          k = 0
91          while k < self.maxnumproducts:
92
93              typeP = random.randint(0,self.nproductTypes-1)
94              t = random.expovariate(1/float(self.arrivalmean))
95              yield Sim.hold,self,t
96
97              pname = 'p-'+str(k).zfill(maxlenlabel)
98              lstm = [m for m in self.ProdTypes[typeP] ]
99
100             # adding exit -1
101             lstm.append(-1)
102             per = random.uniform(LP,UP)
103             p = Product(name=pname,position=pos,code=k, \
104                 machinelist=lstm, begin=Sim.now(), periode=per,\
105                 msgfile=self.msgfile, simnoagvs=self.simnoagvs,\
106                 typeP=typeP)
107             p.tofile = self.tofile
108             p.toconsole = self.toconsole
109             self.products.append(p)
110             Sim.activate(p,p.pem(machines=self.machines,\
111                                  agvs=self.agvs))
112             k = k+1
113
114     def plotP(self):
115         if self.plot == True:
116             self.runplot(self.products,self.machines)
117
118     def runDemo(self,toconsole,tofile,endsim):
119         self.makeseed()
120         self.productTypes()
121         self.machineCreation()
122         self.agvCreation()
123         Sim.initialize()
124         Sim.activate(self,self.pem())
125         self.changingMsg(toconsole,tofile)
126         Sim.simulate(until=endsim)
127         self.plotP()
128
129     def savePositions(self,file):
130         lTraj=[]
131         for p in self.products:
132             s = p.trajectory
133             lTraj.append(s)
134         savePickle(file,lTraj)
135
136     def saveSimDat(self,file):
137         from stream import Pdata,Mdata,Sdata
138         s = Sdata(self.ProdTypes)
139         for p in self.products:
140             p.arrivetomach.insert(0,p.tinicio)
141             tstart = p.arrivetomach
142             pdat = Pdata(p.Tproc,p.typeP,tstart)
143             s.prod.append(pdat)
144         for m in self.machines:
145             mdat = Mdata(m.code,m.position)
```

```
146             s.mach.append(mdat)
147         savePickle(file,s)
148
149
150 class Product(Sim.Process,Point,Print,LocSys):
151
152     def __init__(self,name,position,code,machinelist,begin,periode,\
153             msgfile,simnoagvs,typeP):
154         # initializing fathers
155         Sim.Process.__init__(self,name)
156         Point.__init__(self,position,code)
157         Print.__init__(self,filename=msgfile,toconsole=False,tofile=False)
158         LocSys.__init__(self,periode)
159
160         # adding values of new variables
161         self.typeP = typeP
162         self.Type = 'Product'
163         self.processlist = machinelist
164         self.begin = begin
165         self.simnoagvs = simnoagvs
166         self.trajectory.append((self.position,self.begin))
167         self.Tproc = []
168         self.tinicio = None
169         self.arrivetomach =[]
170
171     def pem(self,machines,agvs):
172
173         # the real thing
174         self.tinicio = Sim.now()
175         # pointer to all machines
176         self.machines = machines
177
178         # pointer to all agvs
179         self.agvs = agvs
180         #print self.processlist
181         for machinecode in self.processlist:
182             # calling machine
183             # pointer to machine
184             machine = self.getmachine(machinecode)
185
186             # calling agv
187
188
189             # initializing location system
190             tinicio = Sim.now()
191
192
193             # waiting for free agv
194             yield Sim.waituntil,self,self.agvfree
195             # pointer to nearest free agv
196             agv = self.agvnear()
197
198             # verification  of free agv
199             while agv==[]:
200                 yield Sim.waituntil,self,self.agvfree
201                 agv = self.agvnear()
202
203
204             # requesting free agv
205             yield Sim.request,self, agv
206
207             self.eventPrint(Sim.now(),'detect free agv',self,agv.name)
208             self.requestPrint(Sim.now(),self,agv)
209
210
211             self.eventPrint(Sim.now(),'agv travel to product',self,agv.name)
212
213
214             # waiting for agv
215             timetoarrive = self.distance(agv.position)/agv.velocity
216
217             if self.simnoagvs == True:
218                 timetoarrive = 0
219
```

```
220             yield Sim.hold, self, timetoarrive
221             agv.position = self.position
222
223             #verificar
224             self.trajectory.append((self.position,Sim.now()))
225
226             # updating location system
227             tfin = Sim.now()
228
229             timedream = tfin - tinicio
230             tpaso = tinicio
231             while timedream >= self.periode:
232                 self.trajectory.append((self.position,tpaso))
233                 tpaso = tpaso + self.periode
234                 timedream =  timedream - self.periode
235
236
237             # begin trip to machine
238             self.eventPrint(Sim.now(),'agv travel to machine',self,agv.name)
239             # adding experiment
240             entermach = machine.position
241
242             timetrip = self.distance(entermach)/agv.velocity
243
244             if self.simnoagvs == True:
245                 timetrip = 0
246
247             # location system
248             while timetrip >self.periode:
249                 to,per,pos = Sim.now(),self.periode,entermach
250                 yield Sim.hold, self, self.periode
251
252                 self.position = self.npto(to+per,to, to+timetrip,pos)
253                 agv.position = self.position
254                 self.trajectory.append((self.position,Sim.now()))
255                 timetrip = timetrip - self.periode
256
257             #synchronization
258             if self.distance(entermach) !=0:
259                 yield Sim.hold, self, self.periode
260                 self.position = entermach
261                 agv.position = self.position
262                 self.trajectory.append((self.position,Sim.now()))
263
264             self.eventPrint(Sim.now(),'finish agv travel',self,agv.name)
265
266             # release agv
267             yield Sim.release,self, agv
268             self.releasePrint(Sim.now(),self,agv)
269             self.eventPrint(Sim.now(),'machine work begin',self,machine.name)
270
271             # initializing location system
272             tinicio2 = Sim.now()
273             self.arrivetomach.append(tinicio2)
274             yield Sim.request, self, machine
275             self.requestPrint(Sim.now(),self,machine)
276
277
278             if machine.name != 'exit':
279                 processingT = random.expovariate(1./machine.meanT)
280                 #modif
281                 #print processingT, self.code
282                 yield Sim.hold, self,processingT
283                 self.eventPrint(Sim.now(),'machine work end  ',self,machine.name)
284             else:
285                 processingT = 0.0
286                 yield Sim.hold, self, processingT
287                 self.eventPrint(Sim.now(),'product exit',self,machine.name)
288             self.Tproc.append(processingT)
289
290             yield Sim.release,self, machine
291             self.releasePrint(Sim.now(),self,machine)
292
293             # updating location system
```

```
294             tfin2 = Sim.now()
295
296             timedream2 = tfin2 - tinicio2
297
298             tpaso2 = tinicio2
299             while timedream2 >= self.periode:
300                 self.trajectory.append((self.position,tpaso2))
301                 tpaso2 = tpaso2 + self.periode
302                 timedream2 =  timedream2 - self.periode
303
304             # end and die
305
306     def getmachine(self,code):
307         ml = [m for m in self.machines if m.code==code]
308         return ml.pop()
309
310     def agvfree(self):
311         '''agvfree Method of Product class.
312         Return True if exist a free agv.'''
313         # agv_instance.waitQ: waiting process
314         # agv_instance.activeQ: active process
315
316         al = [ a for a in self.agvs if len(a.waitQ)==0 and len(a.activeQ)==0]
317
318         #for print
319         t = Sim.now()
320         agvnames = [a.name for a in al]
321         name = 'agvfree query'
322         #self.eventPrint(t,name,self,agvnames)
323
324         return len(al)>0
325
326     def agvnear(self):
327         # return  the nearest free agv
328         al = [ (self.distance(a.position),a) for a in self.agvs\
329             if len(a.waitQ)==0 and len(a.activeQ)==0]
330         al.sort()
331
332         #for print
333         t = Sim.now()
334
335         agvdisname = [(str(round(a[0],2)),a[1].name) for a in al]
336         name = 'agvnear query'
337         #self.eventPrint(t,name,self,agvdisname)
338         if len(al)>0:
339             return al[0][1]
340         else:
341             return []
```

Liste des publications

Chapitres de livres

P. Palominos, V. Parada, G. Gatica and A. Véjar, «Marriage in Honeybee Optimization to Scheduling Problems», in J. R. Montoya-Torres, A. A. Juan, L. Huaccho, J. Faulin, G. Rodriguez-Verjan, editors, *Hybrid Algorithms for Service, Computing and Manufacturing Systems : Routing, Scheduling and Availability Solutions*. IGI Global, 2011.

Revues internationales avec comité de lecture

A. Véjar and P. Charpentier, *Generation of an Adaptive Simulation Driven by Product Trajectories*. (to appear in) Journal of Intelligent Manufacturing, S.I : Intelligent & Distributed Production Control. 2011.

Conférences internationales avec comité de lecture

M. Zanin and A. Véjar , *The emergence of transportation networks*. European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3 : Emergence and Networks. (2009). University of Warwick, UK.

A. Véjar and I. Gómez-Castaño, *Reparation as an emergent process in metabolic networks*, European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3 : Emergence and Networks. (2009). University of Warwick, UK.

P. Palominos, A. Véjar and F. Toledo. *Marriage in Honey Bees Optimization Algorithm for Flow-Shop Problems : Changing the Space of Mating-Flight*, 20th International Conference on Production Research (ICPR'20), 2009. Shanghai, China.

A. Véjar and P. Charpentier, *Simulation code generation using product location information*. 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), 2009. Moscow, Russia.

A. Véjar, F. Chaxel, J.-Y. Bron and P. Charpentier, *Génération automatique de modèle de simulation par les données de localisation des produits*. 6ème Conférence Internationale Conception & Production Intégrées (CPI'09), 2009. Fès, Morocco.

Conférences nationaux avec comité de lecture

A. Véjar and P. Charpentier, *Générateur de code de simulation à partir de l'information de localisation du produit*. 3èmes Journées Doctorales / Journées Nationales du GdR MACS (JD-JN-MACS), 2009. Angers, France.

A. Véjar and P. Charpentier, *Information de localisation du produit pour la génération de code de simulation*. 8ème Congrès International de Génie Industriel (CIGI'09), 2009. Bagnères de Bigorre, Tarbes, France.

A. Véjar and P. Charpentier, *Construction d'un modèle de simulation par le biais de données de localisation des produits*. 10ème Conférence de la Société Française de Recherche Opérationelle et d'Aide à la Décision (ROADEF'09), 2009. Nancy, France.

Bibliographie

[1] A. Adamatzky, G.J. Martinez, S.V. Chapa-Vergara, R. Asomoza-Palacio, and C.R. Stephens. Approximating mexican highways with slime mould. *Natural Computing*, pages 1–20, 2010.

[2] E Aitenbichler, F Lyardet, G Austaller, J Kangasharju, and M Muhlhauser. Engineering intuitive and self-explanatory smart products. In *Proceedings of the 22nd Annual ACM Symposium on Applied Computing*, pages 1632–1637. ACM New York, NY, USA, ACM Press, 2007.

[3] Erwin Aitenbichler, Sebastian Ries, Julian SSchröder-Bernhardi, Georg Turban, Stephan Borgert, Dirk Bradler, Michael Hartle, and Gina Häussge. Smart Products : Integration Challenges. In *AmI-Blocks{'}08 : Second European Workshop on Smart Products : Building Blocks of Ambient Intelligence*, pages 39–40, 2008.

[4] Mohammed ALRahmawy and Andy Wellings. A model for real time mobility based on the RTSJ. In *JTRES '07 : Proceedings of the 5th international workshop on Java technologies for real-time and embedded systems*, pages 155–164, New York, NY, USA, 2007. ACM Press.

[5] Sten F. Andler, Lars Niklasson, Bjorn Olsson, Anne Persson, Leo J. de Vin, Benkt Wangler, Tom Ziemke, and Tomas Planstedt. Information fusion from databases, sensors and simulations : A collaborative research program. In *Proceedings of the 29th Annual IEEE/NASA on Software Engineering Workshop*, pages 234–244, Washington, DC, USA, 2005. IEEE Computer Society.

[6] B. Armstrong. On finding excitation trajectories for identification experiments involving systems with nonlinear dynamics. *The International Journal of Robotic Research*, 8(6) :28–48, 1989.

[7] Daniel Ashbrook and Thad Starner. Learning Significant Locations and Predicting User Movement with GPS. *ISWC 2002*, 2002.

[8] M. Auvray. Théories de la perception : Le monde comme mémoire externe. In P. Fuchs, editor, *Le Traité de la Réalité Virtuelle III. L'Homme et l'Environnement Virtuel*, volume 1, pages 209–228. Paris : Les Presses de l'Ecole des Mines, 2006.

[9] Alex Bahouth, Steven Crites, Norman Matloff, and Todd Williamson. Revisiting the Issue of Performance Enhancement of Discrete Event Simulation Software. In *40th Annual Simulation Symposium (ANSS{'}07)*, volume 0, pages 114–122, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[10] Javier Bajo, Vicente Julián, Juan Manuel Corchado, Carlos Carrascosa, Yanira de Paz, Vicente Botti, and Juan Francisco de Paz. An execution time planner for the artis agent architecture. *Eng. Appl. Artif. Intell.*, 21 :769–784, August 2008.

[11] C. Bandulet. The storage evolution : From blocks, files and objects to objects storage systems. Storage Networking Industry Association, 2007.

[12] Jerry Banks and John S Carson II. Applying the simulation process. In *WSC {'}88 : Proceedings of the 20th conference on Winter simulation*, pages 52–55, New York, NY, USA, 1988. ACM.

[13] Y. Barlas and S. Carpenter. Philosophical roots of model validation : Two paradigms. *System Dynamics Review*, 6(2) :148–166, 1990.

[14] Jean. Baudrillard. *Simulacres et simulation.* Galilee, [Paris] :, 1981.

[15] Jacob Beal. Dynamically defined processes for spatial computers. *Self-Adaptive and Self-Organizing Systems Workshops, IEEE International Conference on*, 0 :206–211, 2010.

[16] Momotaz Begum, George K I Mann, and Raymond G Gosine. Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots. *Appl. Soft Comput.*, 8(1) :150–165, 2008.

[17] Sören Bergmann and Steffen Strassburger. Challenges for the automatic generation of simulation models for production systems. In *2010 Summer Simulation Multiconference*, SummerSim '10, pages 545–549, San Diego, CA, USA, 2010. Society for Computer Simulation International.

[18] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and control*, 60(1-3) :109–137, 1984.

[19] J.A. Bergstra, A. Ponse, and S.A. Smolka. *Handbook of process algebra.* Elsevier Science, 2001.

[20] Guillaume Beslon. *Apprivoiser la vie ; modélisation individu-centrée de systèmes biologiques complexes.* Habilitation à diriger des recherches, INSA-Lyon, November 2008.

[21] W.A. Beyer, P.H. Sellers, and M.S. Waterman. Stanislaw m. ulam's contributions to theoretical theory. *letters in mathematical physics*, 10(2) :231–242, 1985.

[22] G. Bolch. *Queueing networks and Markov chains : modeling and performance evaluation with computer science applications.* Wiley-Blackwell, 2006.

[23] Eric Bonabeau. Agent-based modeling : Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99 :7280–7287, May 2002.

[24] Philippe Bonnifait, Maged Jabbour, and Gérald Dherbomez. Real-Time Implementation of a GIS-Based Localization System for Intelligent Vehicles. *EURASIP Journal on Embedded Systems*, 2007, 2007.

[25] G A Borges and M J Aidon. Design of Real-Time Dynamic Localization System for Mobile Robots. In Michel Devy and Lerasle Frédéric, editors, *SIRS 2001 : proceedings of the 9th international symposium on intelligent robotic systems*, pages 425–434. LAAS-CNRS, 2001.

[26] J.-L. Borges. *Histoire universelle de l'infamie / Histoire de l'éternité*, chapter De la rigueur de la science, page 107. collection 10/18. Union générale d'éditions, 1951.

[27] A. Boumaza and B. Scherrer. Optimal control subsumes harmonic control. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2841–2846. IEEE, 2007.

[28] I. Bratko. Qualitative modelling. *working paper*, pages 8–18, 2005.

[29] Marcus Brohede and Sten F. Andler. A distributed whiteboard infrastructure for information fusion simulations. In *Proceedings of the 2008 32nd Annual IEEE Software Engineering Workshop*, SEW '08, pages 134–140, Washington, DC, USA, 2008. IEEE Computer Society.

[30] N Brown and S Powers. Simulation in a Box : (A Generic Reusable Maintenance Model). In *Proceeedings of the 2000 Winter Simulation Conference*, pages 1050–1056, 2000.

[31] A.W. Burks. *Essays on cellular automata*. University of Illinois Press, 1970.

[32] S. Bussmann. An agent-oriented architecture for holonic manufacturing control. In *Proc. of 1st Int. Workshop on Intelligent Manufacturing Systems*, pages 1–12, 1998. EPFL, Lausanne, Switzerland.

[33] Bing Cao, Richard Farr, Mike Byrne, and James Tannock. Data-driven Simulation of the Extended Enterprise. *18th International Conference on Production Research*, 2004.

[34] D. Cardon, G. Fouetillou, and C. Roth. Two paths of glory structural positions and trajectories of websites within their topical territory. *AAAI ICWSM'11 Intl Conf Weblogs and Social Media, Barcelona, July 2011.*, 2011.

[35] Francois Caron, Saiedeh Navabzadeh Razavi, Jongchul Song, Philippe Vanheeghe, Emmanuel Duflos, Carlos Caldas, and Carl Haas. Locating sensor nodes on construction projects. *Auton Robot*, 2007.

[36] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Events Systems*. Kluwer Academic Publishers, Dordrecht, 1999.

[37] Timon Du Vincent S. Lai Charles Ling-yu Chou. Continuous auditing with a multi-agent system. *Decision Support Systems* , pages 2274 – 2292, 2007.

[38] David Chavalarias, Paul Bourgine, Edith Perrier, Fréderic Amblard, François Arlabosse, Pierre Auger, Jean-Bernard Baillon, Olivier Barreteau, Pierre Baudot, Elisabeth Bouchaud, Soufian Ben Amor, Hugues Berry, Cyrille Bertelle, Marc Berthod, Guillaume Beslon, Giulio Biroli, Daniel Bonamy, Daniele Bourcier, Nicolas Brodu, Marc Bui, Yves Burnod, Bertrand Chapron, Catherine Christophe, Bruno Clément, Jean-Louis Coatrieux, Jean-Philippe Cointet, Valérie Dagrain, Katia Dauchot, Olivier Dauchot, François Daviaud, Silvia De Monte, Guillaume Deffuant, Pierre Degond, Jean-Paul Delahaye, René Doursat, Francesco D'Ovidio, Marc Dubois, Berengère Dubruelle, Marie Dutreix, Robert Faivre, Emmanuel Farge, Patrick Flandrin, Sara Franceschelli, Cédric Gaucherel, Jean-Pierre Gaudin, Michael Ghil, Jean-Louis Giavitto, Francesco Ginelli, Vincent Ginot, François Houllier, Bernard Hubert, Pablo Jensen, Ludovic Jullien, Zoi Kapoula, Daniel Krob, François Ladieu, Gabriel Lang, Chrsitophe Lavelle, André Le Bivic, Jean-Pierre Leca, Christophe Lecerf, Pierre Legrain, Denis L'hôte, Maud Loireau, Jean-Francois Mangin, Olivier Monga, Michel Morvan, Jean-Pierre Muller, Ioan Negrutiu, Nadine Peyreiras, Denise Pumain, Ovidiu Radulescu, Jean Sallantin, Eric Sanchis, Daniel Schertzer, Marc Schoenauer, Michèle Sebag, Eric Simonet, Adrien Six, Fabien Tarissan, and Patrick Vincent. French Roadmap for complex Systems 2008-2009. This second issue of the French Complex Systems Roadmap by the French National Network for Complex SYstems (http ://RNSC.fr) and the Paris Ile-de-France Complex Systems Institute (http ://iscpif.fr) is the outcome of the "Entretiens de Cargèse 2008", an interdisciplinary brainstorming session organized over one week in 2008, jointly by RNSC, ISC-PIF and IXXI. It capitalizes on the first roadmap and gathers contributions of more than 70 scientists from major French institutions., 03 2009.

[39] Lun-Chi Chen, Ruey-Kai Sheu, Hui-Chieh Lu, Win-Tsung Lo, and Yen-Ping Chu. Object Finding System Based on RFID Technology. In Heng Tao Shen, Jinbao Li, Minglu Li, Jun Ni, and Wei Wang, editors, *APWeb Workshops : XRA, IWSN, MEGA, and ICSE, Harbin, China, January 16-18, 2006, Proceedings*, volume 3842, pages 383–396. Springer, 2006.

[40] Harry K. H. Chow, K. L. Choy, and W. B. Lee. A dynamic logistics process knowledge-based system - An RFID multi-agent approach. *Know.-Based Syst.*, 20(4) :357–372, 2007.

[41] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Trans. Softw. Eng.*, 4 :178–187, May 1978.

[42] Debashish Chowdhury, Andreas Schadschneider, and Katsuhiro Nishinari. Physics of transport and traffic phenomena in biology : from molecular motors and cells to organisms. *Physics of Life Reviews*, 2(4) :318–352, Dec 2005.

[43] R. L. Church. Geographical information systems and location science. *Computers and Operations Research*, 29(6) :541–562, 2002.

[44] Andy Clark and David J. Chalmers. The extended mind. *Analysis*, 58(1) :7–19, 1998.

[45] G M Clark and C R Cash. Data-driven simulation of networks with manufacturing blocking. In *Proceedings of the 1993 Winter Simulation Conference*, 1993.

[46] D. Coca and S. A. Billings. Identification of coupled map lattice models of complex spatio-temporal patterns. *Physics Letters A*, 287(1-2) :65–73, Aug 2001.

[47] Víctor Codocedo and Hernán Astudillo. No mining, no meaning : relating documents across repositories with ontology-driven information extraction. In *Proceeding of the eighth ACM symposium on Document engineering*, DocEng '08, pages 110–118, New York, NY, USA, 2008. ACM.

[48] J.L. Coen, JD Beezley, LS Bennethum, CC Douglas, M. Kim, R. Kremens, J. Mandel, G. Qin, and A. Vodacek. A wildland fire dynamic data-driven application system. In *11th Symposium on Integrated Observing and Assimilation Systems for the Atmosphere, Oceans, and Land Surface*, 2007.

[49] Rosaria Conte. From simulation to theory (and backward). In Flaminio Squazzoni, editor, *Epistemological Aspects of Computer Simulation in the Social Sciences*, pages 29–47. Springer-Verlag, Berlin, Heidelberg, 2009.

[50] Antonio Coronato, Giuseppe De Pietro, and Massimo Esposito. A Semantic Location Service for Pervasive Grids. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006 : OTM 2006 Workshops*, volume 4278, pages 1274–1284. Springer, 2006.

[51] M. Crease, P. Gray, and J. Cargill. Using Location Information in an Undergraduate Computing Science Laboratory Support System. In *Proc of Location Modelling for Ubiquitous Computing (workshop, Ubicomp)*, 2001.

[52] F Darema. Dynamic Data Driven Application Systems (Symbiotic Measurement & Simulation Systems), "A new paradigm for application simulations and a new paradigm for measurement systems". NSF sponsored Workshop, 2000.

[53] F Darema. Dynamic Data Driven Applications Systems : A New Paradigm for Application Simulations and Measurements. In *Proceeding of the 2004 International Conference on Computational Science*, pages 662–669, 2004.

[54] L. J. De Vin, A. H. C. Ng, and J. Oscarsson. Simulation-Based Decision Support for Manufacturing System Life Cycle Management. *Journal of Advanced Manufacturing Systems*, 3 :115–128, 2004.

[55] L J De Vin, A H C Ng, J Oscarsson, and S F Andler. Information Fusion for Simulation Based Decision Support in Manufacturing. *FAIM 2005 Special Issue of Robotics and Computer Integrated Manufacture*, 22 :429–436, 2006.

[56] Leo J. De Vin, Magnus Holm, and Amos H. C. Ng. The information fusion jdl-u model as a reference model for virtual manufacturing. *Robot. Comput.-Integr. Manuf.*, 26 :629–638, December 2010.

[57] C.A.C. de Wit. *Adaptive control for partially known systems : theory and applications*, volume 7. Elsevier Science Ltd, 1988.

[58] Durk-Jouke der Zee and Jack G A J der Vorst. Guiding Principles for Conceptual Model Creation in Manufacturing Simulation. In S G Henderson, B Biller, M.-H. Hsieh, J Shortle, J D Tew, and R R Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*. IEEE, 2007.

[59] A Diaz-Calderon, C Paredis, and Khosla P. Architectures and Languages for Model Building and Reuse : Organization and Selection of Reconfigurable Models. In *Proceeedings of the 2000 Winter Simulation Conference*, pages 386–393, 2000.

[60] Dragos Dobre and Eddy Bajic. Active product modeling for chemical security management based on smart object concept. In *7 Conférence Internationale de Modélisation, Optimisation et Simulation des Systèmes, MOSIM 08*, Paris, France, 2008.

[61] G D'Souza and A Allaway. A data-driven modelling approach to product level decision support. *The Journal of Product and Brand Management*, 6(2) :130, 1997.

[62] J.-P. Eckmann, S. Oliffson Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *EPL (Europhysics Letters)*, 4(9) :973, 1987.

[63] Joshua M. Epstein. Agent-based computational models and generative social science. *Complex.*, 4 :41–60, May 1999.

[64] Joshua M. Epstein. *Generative Social Science : Studies in Agent-Based Computational Modeling (Princeton Studies in Complexity).* Princeton University Press, Princeton, NJ, USA, 2007.

[65] Joshua M. Epstein. Why model ? *Journal of Artificial Societies and Social Simulation*, 11(4) :12, 2008.

[66] Joshua M. Epstein and Robert Axtell. *Growing artificial societies : social science from the bottom up.* The Brookings Institution, Washington, DC, USA, 1996.

[67] K Fischer. Agent-based design of holonic manufacturing systems. *Robotics and autonomous Systems*, 27(1-2) :3–13, 1999.

[68] K Fischer, M Schillo, and J Siekmann. Holonic multiagent systems : A foundation for the organisation of multiagent systems. In Marik, V and McFarlane, D and Valckenaers, P, editor, *HOLONIC AND MULTI-AGENT SYSTEMS FOR MANUFACTURING*, volume 2744 of *LECTURE NOTES IN ARTIFICIAL INTELLIGENCE*, pages 71–80, 2003. 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems, PRAGUE, CZECH REPUBLIC, SEP 01-03, 2003.

[69] J.W. Forrester. *Principles of systems.* MIT Press Cambridge, MA, 1980.

[70] Carlo Gaetan and Xavier Guyon. *Modélisation et statistique spatiales.* Springer, 2008.

[71] P. Gaspar, J.-Y. Georges, A. Lenoble, S. Ferraroli, S. Fossette, and Y. Le Maho. Marine animal behaviour : neglecting ocean currents can lead us up the wrong track. *Proc. R. Soc. B.*, 273(1602) :2697–2702, 11 2006.

[72] Peter Gawthrop and Lorcan Smith. *Metamodelling : for bond graphs and dynamic systems.* Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1996.

[73] F Gechter, V Chevrier, and F Charpillet. Un mod{è}le de r{é}solution de probl{è}mes utilisant des agents r{é}actifs pour la localisation et le suivi. *Revue d{'}intelligence artificielle*, 18(5-6) :743–777, 2004.

[74] Franck Gechter, Vincent Chevrier, and François Charpillet. A reactive agent-based problem-solving model : Application to localization and tracking. *ACM Trans. Auton. Adapt. Syst.*, 1 :189–222, December 2006.

[75] M. Gorawski and R. Malczok. Distributed spatial data warehouse indexed with virtual memory aggregation tree. In *Proceedings of the Second Workshop on Spatio-Temporal Database Management, (STDBM'04)*, Toronto, Canada, 2004.

[76] Joachim Goßmann and Marcus Specht. Location Models for Augmented Environments. *Personal Ubiquitous Comput.*, 6(5-6) :334–340, 2002.

[77] David Gouyon and Michael David. Implementing the concept of product-driven control using wireless sensor networks : some experiments and issues. In *17th IFAC World Congress*, Seoul, Korea, 2008.

[78] Volker Grimm and Steven F Railsback. *Individual-based Modeling and Ecology (Princeton Series in Theoretical and Computational Biology)*. Princeton University Press, Princeton, July 2005.

[79] Ashu Guru and Paul Savory. A Template-Based Conceptual Modeling Infrastructure for Simulation of Physical Security Systems. In R G Ingalls, M D Rossetti, J S Smith, and B A Peters, editors, *WSC'04 : Proceedings of the 36th conference on Winter simulation*, pages 866–873, 2004.

[80] Georges Habchi. Modelling and simulation of complex production systems [mosim'99]. *Simul. Pr. Theory*, 8(5) :281–282, 2000.

[81] Georges Habchi. Modelling and simulation : Analysis, design and optimisation of industrial systems [mosim'01]. *Simulation Modelling Practice and Theory*, 11(1) :1–3, 2003.

[82] V. Havelange, C. Lenay, and J. Stewart. Les représentations : mémoire externe et objets techniques. *Intellectica*, 35 :115–131, 2003.

[83] F.G. Hayden. The inadequacy of forrester system dynamics computer programs for institutional principles of hierarchy, feedback, and openness. *CBA Faculty Publications*, page 14, 2006.

[84] Dirk Helbing, Dieter Armbruster, Alexander S. Mikhailov, and Erjen Lefeber. Information and material flows in complex networks. *Physica A : Statistical Mechanics and its Applications*, 363(1) :xi – xvi, 2006. Information and Material Flows in Complex Networks, Information and Material Flows in Complex Networks.

[85] Herianto, Toshiki Sakakibara, and Daisuke Kurabayashi. Artificial Pheromone System Using RFID for Navigation of Autonomous Robots. *Journal of Bionic Engineering*, 4(4) :245–253, 2007.

[86] A. C. Hindmarsh. ODEPACK, A Systematized Collection of ODE Solvers , R. S. Stepleman et al. (eds.), North-Holland, Amsterdam, (vol. 1 of), pp. 55-64. *IMACS Transactions on Scientific Computation*, 1 :55–64, 1983.

[87] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to automata theory, languages, and computation*, volume 3. Addison-wesley Reading, MA, 1979.

[88] Yilin Huang and Alexander Verbraeck. A dynamic data-driven approach for rail transport system simulation. In *Winter Simulation Conference*, WSC '09, pages 2553–2562. Winter Simulation Conference, 2009.

[89] I. C. Baianu. Organismic supercategories and qualitative dynamics of systems. *Bulletin of Mathematical Biophysics*, 33 :349–365, 1971.

[90] Joseph S. Iwanski and Elizabeth Bradley. Recurrence plots of experimental data : To embed or not to embed ? *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 8(4) :861–871, 1998.

[91] Martin Nilsson Jacobi. Hierarchical organization in smooth dynamical systems. *Artificial Life*, 11(4) :493–512, 2005.

[92] E. Jiménez, L. Recalde, and M. Silva. Forrester diagrams and continuous Petri nets : A comparative view. In *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, volume 2, pages 85–94. IEEE, 2001.

[93] Forrester J.W. Industrial dynamics. *Journal of the Operational Research Society*, 48(10) :1037–1041, 1997.

[94] Pablo Kaluza, Andrea Kölzsch, Michael T. Gastner, and Bernd Blasius. The complex network of global cargo ship movements. *Journal of The Royal Society Interface*, 7(48) :1093–1103, 2010.

[95] M. Karkkainen, J. Holmstrom, K Framling, and K. Artto. Intelligent products - a step towards a more effective project delivery chain. *Computers in Industry*, 50 :141–151, 2003.

[96] S Kasputis and H C Ng. Model Composability : Formulating a Research Thrust : Composable Simulations. In *Proceedings of the 2000 Winter Simulation Conference*, pages 1577–1584, 2000.

[97] C Kennedy and G Theodoropoulos. Towards Intelligent Data-Driven Simulation for Policy Decision Support in the Social Sciences. Technical report, University of Birmingham, School of Computer Science, 2005.

[98] Jindae Kim, Kaizhi Tang, Soundar Kumara, Shang-Tae Yee, and Jeffrey Tew. Value analysis of location-enabled radio-frequency identification information on delivery chain performance. *International Journal of Production Economics*, 112(1) :403–415, Mar 2008.

[99] Moon-Chan Kim, Chang Ouk Kim, Seong Rok Hong, and Ick-Hyun Kwon. Forward-backward analysis of rfid-enabled supply chain using fuzzy cognitive map and genetic algorithm. *Expert Systems with Applications*, 35(3) :1166 – 1176, 2008.

[100] Robert Kittell and Ann Dunkin. Applying the simulation process : simulation study of HDA parts degreaser. In *WSC {'}91 : Proceedings of the 23rd conference on Winter simulation*, pages 816–821, Washington, DC, USA, 1991. IEEE Computer Society.

[101] S. Koh, A. Gunasekaran, and S. Saad. Parts verification for multi-level-dependent demand manufacturing systems : a recognition and classification structure. *The International Journal of Advanced Manufacturing Technology*, 31 :305–315, 2006. 10.1007/s00170-005-0184-9.

[102] Khalid Kouiss, Henri Pierreval, and Nasser Mebarki. Using multi-agent architecture in fms for dynamic scheduling. *Journal of Intelligent Manufacturing*, 8 :41–47, 1997. 10.1023/A :1018540317470.

[103] R V Kulkarni, G K Venayagamoorthy, and M X Cheng. Bio-inspired node localization in wireless sensor networks. In *Systems Man and Cybernetics 2009 SMC 2009 IEEE International Conference on*, pages 205–210, 2009.

[104] C.G. Langton, editor. *Artificial Life : Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE'87)*. Addison-Wesley, Redwood City, CA, 1989.

[105] J.Y. Le Boudec and P. Thiran. *Network calculus*. Citeseer, 2001.

[106] Paulo Leitão, Raymond Boissier, Francisco Casais, and Francisco Restivo. Integration of Automation Resources in Holonic Manufacturing Applications. *Holonic and Multi-Agent Systems for Manufacturing*, pages 1085–1086, 2003.

[107] Paulo Leitão. Agent-based distributed manufacturing control : A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7) :979 – 991, 2009. Distributed Control of Production Systems.

[108] J. J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3) :376–382, 1991.

[109] R.M. Lima, R.M. Sousa, and P.J. Martins. Distributed production planning and control agent-based system. *International journal of production research*, 44(18) :3693–3709, 2006.

[110] A. Lindenmayer. Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology*, 30(3) :455–484, 1971.

[111] A. Lindenmayer. Adding continuous components to l-systems. *L Systems*, pages 53–68, 1974.

[112] Peter Lunenfeld. *The Digital Dialectic : New Essays on New Media*. MIT Press, Cambridge, Mass, 1998.

[113] Fernando Lyardet and Erwin Aitenbichler. Smart Products : Building Blocks of Ambient Intelligence. In *Constructing Ambient Intelligence : AmI 2007 Workshops ; Darmstadt, Germany, November 2007 ; Revised Papers*, volume 11 of *CCIS*, pages 156–157. Springer Verlag, Heidelberg, 2007.

[114] Charles M. Macal and Michael J. North. Tutorial on agent-based modeling and simulation. In *WSC '05 : Proceedings of the 37th conference on Winter simulation*, pages 2–15. Winter Simulation Conference, 2005.

[115] G T Mackulak and J K Cochran. The Generic Specific Modeling Approach : An Application of Artificial Intelligence to Simulation. In *IIE Integrated Systems Conference & Society of Integrated Manufacturing Conference Proceedings*, pages 82–87, 1990.

[116] G T Mackulak, F P Lawrence, and T Colvin. Effective Simulation Model Reuse : A case study for AMHS modeling. In *Proceedings of the 1998 Winter Simulation Conference*, pages 979–984, 1998.

[117] Anu Maria. Introduction to modeling and simulation. In *Proceedings of the 1997 Winter Simulation Conference*, pages 7–13, Washington, DC, USA, 1997. IEEE Computer Society.

[118] J. P. Marney and Heather F.E. Tarbert. Why do simulation ? towards a working epistemology for practitioners of the dark arts. *Journal of Artificial Societies and Social Simulation*, 3 :4, 2000.

[119] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths. Recurrence Plots for the Analysis of Complex Systems. *Physics Reports*, 438(5–6) :237–329, 2007.

[120] Angus McBlane. Just a ghost in a shell ? In J. Steiff and T.D. Tamplin, editors, *Anime and Philosophy*, volume 47. Open Court Pub Co, 2010.

[121] Chris McCormick, Pierre-Yves Laligand, Huang Lee, and Hamid Aghajan. Distributed agent control with self-localizing wireless image sensor networks. In *Proceedings of Cognitive Systems and Interactive Sensors (COGIS 2006)*, March 2006.

[122] D. McFarlane and S. Bussmann. Holonic manufacturing control : Rationales, developments and open issues. In S.M. Deen, editor, *Agent-Based Manufacturing - Advances in the Holonic Approach*, pages 303–326. Springer-Verlag, 2003.

[123] Duncan Mcfarlane, Sanjay Sarma, Jin L Chirn, C Y Wong, and Kevin Ashton. Auto ID systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4) :365–376, June 2003.

[124] C. McLean, A. Jones, T. Lee, and F. Riddick. An architecture for a generic data-driven machine shop simulator. *Winter Simulation Conference*, 2 :1108–1116, 2002.

[125] Charles Mclean and Swee Leong. The role of simulation in strategic manufacturing. In *In Proceedings of the 2001 International working Conference on Strategic Manufacturing*, 2001.

[126] C. Methuen. Time human or time divine : theological aspects in opposing the gregorian calendar reform. *Reformation and Renaissance Review*, 3(1) :36–50, 2001.

[127] D. Michel and D. Dubois. La représentation cognitive : quelques modèles récents. *L'année psychologique*, 76(2) :541–562, 1976.

[128] R. Milner. *Communicating and mobile systems : the pi-calculus*, volume 13. Cambridge university press, 1999.

[129] Samieh Mirdamadi, Franck Fontanili, and Lionel Dupont. Discrete Event Simulation-Based Real-Time Shop Floor Control. In Ivan Zelinka, Zuzana Oplatková, and Alessandra Orsoni, editors, *21st European Conference on Modelling and Simulation ECMS 2007*, pages 572–577, 2007.

[130] Saurabh Mittal, Eddie Mak, and James J. Nutaro. DEVS-Based Dynamic Model Reconfiguration and Simulation Control in the Enhanced DoDAF Design Process. *The Journal of Defense Modeling and Simulation : Applications, Methodology, Technology*, 3(4) :239–267, 2006.

[131] László Monostori and Balázs Csanád Csáji. Complex adaptive systems (cas) approach to production systems and organisations. In Mamoru Mitsuishi, Kanji Ueda, and Fumihiko Kimura, editors, *Manufacturing Systems and Technologies for the New Frontier*, pages 19–24. Springer London, 2008.

[132] Antonio Paulo Moreira, Armando Sousa, and Paulo Costa. Vision Based Real-Time Localization of Multiple Mobile Robots. In *3rd Int. Conf. on Field and Service Robotics*, pages 103–106, jun 2001.

[133] Michael F Morris. Simulation as a Process. *SIGSIM Simul. Dig.*, 4(1), 1972.

[134] K. Muller. Advanced systems simulation capabilities in SimPy. *Europython 2004*, 2004.

[135] K. Muller and T. Vignaux. SimPy : Simulating Systems in Python. *ONLamp.com Python Devcenter*, 2003.

[136] T. Murata. Circuit theoretic analysis and synthesis of marked graphs. *Circuits and Systems, IEEE Transactions on*, 24(7) :400–405, 1977.

[137] T. Murata. Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, 77(4) :541–580, 1989.

[138] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.

[139] Kristen Nygaard and Ole-Johan Dahl. The development of the simula languages. In Richard L. Wexelblat, editor, *History of programming languages I*, pages 439–480. ACM, New York, NY, USA, 1981.

[140] J. K. O'Regan and A. Noë. A sensorimotor account of vision and visual consciousness. *The Behavioral and brain sciences*, 24(5), 2001.

[141] E H Page and J M Opper. Observation on the Complexity of the Composable Simulation. In *Proceedings of the 1999 Winter Simulation Conference*, pages 553–560, 1999.

[142] K.-J. Park and Y.-H. Lee. An On-line Simulation Approach to Search Efficient Values of Decision Variables in Stochastic Systems. *Int J Adv Manuf Technol*, 2005.

[143] H.M. Paynter. *Analysis and design of engineering systems*. MIT press, 1961.

[144] J. Petitot. Phénoménologie naturalisée et morphodynamique : la fonction cognitive du synthétique a priori. *Intellectica*, 17 :79–126, 1993.

[145] J. Petitot. Objectivité faible et philosophie transcendentale. *Journée d"etude sur la philosophie de Bernard d'Espagnat*, 35 :115–131, 1996.

[146] Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *Siam J. Sci. Stat. Comput*, 4(1) :136–148, 1983.

[147] E. Posse, J.S. Bolduc, and H. Vangheluwe. Generation of DEVS Modelling and Simulation Environments. In *Proceedings of the 2003 Summer Computer Simulation Conference*, 2003.

[148] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer, 1991.

[149] J. Pugh and A. Martinoli. Relative localization and communication module for small-scale multi-robot systems. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 188–193. IEEE, 2006.

[150] Robin G. Qiu. Rfid-enabled automation in support of factory integration. *Robot. Comput.-Integr. Manuf.*, 23(6) :677–683, 2007.

[151] Cesar A. Quiroga. Performance measures and data requirements for congestion management systems. *Transportation Research Part C*, 8 :287–306, 2000.

[152] Brian D. Ripley. *Spatial Statistics*. John Wiley & Sons , 1981.

[153] N Robertson and T Perera. Automated data collection for simulation ? *Simulation Practice and Theory*, 9(6-8) :349 – 364, 2002.

[154] Neil H Robertson and Terrence Perera. General manufacturing applications : feasibility for automatic data collection. In *Proceedings of the 33nd conference on Winter simulation*, pages 984–990, 2001.

[155] N.H. Robertson and T. Perera. Feasibility for automatic data collection. *Winter Simulation Conference*, 1 :984–990, 2001.

[156] John F Roddick, Kathleen Hornsby, and Myra Spiliopoulou. An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. In *TSDM {'}00 : Proceedings of the First International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining-Revised Papers*, pages 147–164, London, UK, 2001. Springer-Verlag.

[157] C. Roth, S.M. Kang, M. Batty, and M. Barthélemy. Structure of urban movements : polycentric activity and entangled hierarchical flows. *PloS one*, 6(1) :e15923, 2011.

[158] Camille Roth. Co-evolution in epistemic networks – reconstructing social complex systems. *Structure and Dynamics : eJournal of Anthropological and Related Sciences*, 1(3) :article 2, 2006.

[159] M Rowlands. The body in mind. *Understanding Cognitive Processes*, 1999.

[160] M Rowlands. *The nature of consciousness*. Cambridge Univ Pr, 2001.

[161] M. Rowlands. Enactivism and the extended mind. *Topoi*, 28(1) :53–62, 2009.

[162] M Rowlands. Extended cognition and the mark of the cognitive. *Philosophical Psychology*, 22(1) :1–19, 2009.

[163] Mark Rowlands. Understanding the "active" in "enactive" . *Phenomenology and the Cognitive Sciences*, 6(4) :427–443, September 2007.

[164] G. Rozenberg and A. Salomaa. *The mathematical theory of L systems*, volume 90. Academic Press, 1980.

[165] A. Rueger. Robust supervenience and emergence. *Philosophy of science*, pages 466–489, 2000.

[166] John Ryan and Cathal Heavey. Process modeling for simulation. *Computers in Industry*, 2006.

[167] Paul Sanchez. As simple as possible, but no simpler : A gentle introduction to simulation modeling. *Winter Simulation Conference*, 0 :2–10, 2006.

[168] P.J. Sánchez. Fundamentals of simulation modeling. In *Proceedings of the 39th conference on Winter simulation : 40 years ! The best is yet to come*, pages 54–62. IEEE Press, 2007.

[169] D. Sangiorgi and D. Walker. *The pi-calculus : a Theory of Mobile Processes*. Cambridge Univ Pr, 2003.

[170] Ichiro Satoh. A location model for smart environments. *Pervasive Mob. Comput.*, 3(2) :158–179, 2007.

[171] Robert E Shannon. Simulation modeling and methodology. In *WSC {'}76 : Proceedings of the 76 Bicentennial conference on Winter simulation*, pages 9–15. Winter Simulation Conference, 1976.

[172] Tomohiro Shirakawa, Andrew Adamatzky, Yukio-Pegio Gunji, and Yoshihiro Miyake. On simultaneous construction of voronoi diagram and delaunay triangulation by *physarum polycephalum*. *I. J. Bifurcation and Chaos*, 19(9) :3109–3117, 2009.

[173] Marcelo Gurgel Simas de Oliveira and Paulo Cezar Martins Ribeiro. Production and analysis of coordination plans using a geographic information system. *Transportation Research Part C*, 9 :53–68, 2001.

[174] A. Sitz, J. Kurths, and H. U. Voss. Identification of nonlinear spatiotemporal systems via partitioned filtering. *Phys. Rev. E*, 68(1) :016202, Jul 2003.

[175] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, pages 167–193, 1990.

[176] D P Solomatine. Data-driven modelling : paradigm, methods, experiences. In *Proc. 5th International Conference on Hydroinformatics*, 2002.

[177] Young Jun Son, Albert T Jones, and Richard A Wysk. Automatic Generation of Simulation Models From Neutral Libraries : An Example. In J A Joines, R R Barton, K Kang, and P A Fishwick, editors, *WSC'00 : Proceedings of the 32th conference on Winter simulation*, pages 1558–1567, 2000.

[178] Jongchul Song, Carl T. Haas, and Carlos H. Caldas. A proximity-based method for locating ID tagged objects. *Advanced Engineering Informatics*, 21(4) :367–376, 2007.

[179] M J Steele, M M G Rabadi, and G Cates. Generic Simulation Models of Reusable Launch Vehicles. In *Proceedings of the 2002 Winter Simulation Conference*, pages 747–753, 2002.

[180] L. Stone and S. Ezrati. Chaos, cycles and spatiotemporal dynamics in plant ecology. *J. Ecol.*, pages 279–291, 1996.

[181] Frédéric Thiesse and Elgar Fleisch. On the value of location information to lot scheduling in complex manufacturing processes. *International Journal of Production Economics*, 112(2) :532–547, 2008.

[182] Cesar Torres-Huitzil, Bernard Girau, Amine Boumaza, and Bruno Scherrer. Embedded harmonic control for trajectory planning in large environments. In *International Conference on ReConFigurable Computing and FPGAs - ReConFig 08*, Cancun Mexique, 2008.

[183] H.L.M. Vangheluwe. Devs as a common denominator for multi-formalism hybrid systems modelling. In *Computer-Aided Control System Design, 2000. CACSD 2000. IEEE International Symposium on*, pages 129–134. IEEE, 2000.

[184] Francisco J. Varela, Evan T. Thompson, and Eleanor Rosch. *The Embodied Mind : Cognitive Science and Human Experience*. The MIT Press, new edition edition, 1992.

[185] Franck Varenne. What does a computer simulation prove ? In Society for Computer Simulation (SCS), editor, *Simulation in Industry - Proc. of the 13th European Simulation Symposium*, pages 549–554, 2001.

[186] T. Vicha and M. Dohnal. Qualitative identification of chaotic systems behaviours. *Chaos, Solitons & Fractals*, 38(1) :70–78, Oct 2008.

[187] Victor Spirin and Leonid A. Mirny. Protein complexes and fuctional modules in molecular networks. *PNAS*, 100(21) :12123–12128, 2003.

[188] J. Walrand. *An introduction to queueing networks*, volume 21. Prentice Hall Englewood Cliffs, NJ, 1988.

[189] Tao Wan, Karine Zeitouni, and Xiaofeng Meng. An OLAP system for network-constrained moving objects. In *SAC '07 : Proceedings of the 2007 ACM symposium on Applied computing*, pages 13–18, New York, NY, USA, 2007. ACM Press.

[190] F. Wilcoxon and A. C. Co. Individual Comparisons by Ranking Methods. *Breakthroughs in Statistics*, 1997.

[191] T R Willemain. Model Formulation : What Experts Think About and When. *Operations Research*, 43(6) :916–932, 1995.

[192] Jan C. Willems. The behavioral approach to open and interconnected systems. *Control Systems Magazine*, pages 46–99, 2007.

[193] B. Williams and J. de Kleer. Qualitative reasoning about physical systems : a return to roots. *Artificial Intelligence*, 51 :1–9, 1991.

[194] Alan F T Winfield, Wenguo Liu, Julien Nembrini, and Alcherio Martinoli. Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2(2-4) :241–266, 2008.

[195] S. Wolfram. *Theory and applications of cellular automata*. World Scientific, 1986.

[196] Zhengdao Xu and Arno Jacobsen. Adaptive location constraint processing. In *SIGMOD {'}07 : Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 581–592, New York, NY, USA, 2007. ACM.

[197] Minghui Yang. Using data driven simulation to build inventory model. *Proceedings of the 2008 Winter Simulation Conference*, pages 2595–2599, 2008.

[198] A.L. Yarbus. Eye movements during perception of complex objects. *Eye movements and vision*, 7 :171–196, 1967.

[199] Kenneth Man-Kam Yip. *KAM : A System for Intelligently Guiding Numerical Experimentation by Computer*. Artificial Intelligence. MIT Press, 1991.

[200] M. Zanin and A. Véjar. The emergence of transportation networks. In *European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3 : Emergence and Networks European Conference on Complex Systems, ECCS09. Satellite Meeting EmergeNET3 : Emergence and Networks*, Warwick UK, 09 2009.

[201] B.P. Zeigler. Devs representation of dynamical systems : Event-based intelligent control. *Proceedings of the IEEE*, 77(1) :72–80, 1989.

[202] B.P. Zeigler, Y. Moon, D. Kim, and G. Ball. The devs environment for high-performance modeling and simulation. *Computational Science & Engineering, IEEE*, 4(3) :61–71, 1997.

[203] Feng Zhao, Chris Bailey-Kellogg, Xingang Huang, and Iván Ordóñez. Intelligent simulation tools for mining large scientific data sets. *New Generation Computing*, 17 :333–347, 1999.

[204] Ming Zhou, Qun Zhang, and Zhimin Chen. What Can Be Done to Automate Conceptual Simulation Modeling ? In L F Perrone, F P Wieland, J Liu, B G Lawson, D M Nicol, and R M Fujimoto, editors, *WSC {'}06 : Proceedings of the 38th conference on Winter simulation*, pages 809–814, 2006.

[205] D. Šuc, D. Vladušič, and I. Bratko. Qualitatively faithful quantitative prediction. *Artificial Intelligence Journal*, 158 :189–214, 2004.

Résumé

Cette thèse propose une méthode originale pour la génération automatique d'un code de simulation pour les systèmes à événements discrets. Cette méthode utilise l'information de localisation des produits lors du fonctionnement du système. Ce flux composé par des tuples (product id, location, time) constitue le point d'entrée pour l'algorithme proposé de génération d'un modèle de simulation de type réseau de files d'attente. Ce type d'approche permet, outre un gain important de temps pour la conception initiale du modèle, une maintenance et reconfiguration " on line " du modèle. La thèse est composée de 5 chapitres. Le premier chapitre pose la problématique et fixe le cadre théorique de la thèse. Le second chapitre est une revue de la littérature sur la simulation en général et sur les travaux utilisant la notion de trajectoires à des fins de modélisation. Le troisième chapitre sert à mettre en avant la proposition au cœur de cette thèse. Le quatrième chapitre décrit le générateur développé. Le cinquième et dernier chapitre présente les travaux d'expérimentation et de validation du générateur.

Mots clés

localisation, trajectoires, simulation, modélisation adaptative

Abstract

This thesis work proposes a novel method for the automatic generation of simulation code of discrete event systems. This method uses the location information of the products during operation of the system. This flow, of tuples (product id, location, time) is the input of the proposed algorithm that generates a queueing-network simulation model. This type of approach can achieve a significant time gain for the design, maintenance and on-line reconfiguration of the model. The thesis is composed on five chapters. The first chapter poses the problem and fixes the theoretical framework of the thesis. The second chapter is a literature review of the broad sense of simulation and present works using the notion of trajectories for modeling. The third chapter is to highlight the proposal of the heart of this thesis. The fourth chapter describes the generator developed. The fifth and final chapter presents the experiments and the validation of the generator.

Keywords

location, trajectories, flow simulation, adaptive modeling

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences, B.P. 70239, 54506 Vandœuvre lès Nancy