



HAL
open science

Un panorama de la recherche en conception de la conduite des systèmes de production

Bruno Denis, Jean-Jacques Lesage

► To cite this version:

Bruno Denis, Jean-Jacques Lesage. Un panorama de la recherche en conception de la conduite des systèmes de production. Congrès international de génie industriel de Montréal, Oct 1995, Montréal, Canada. ⟨hal-03745128⟩

HAL Id: hal-03745128

<https://hal.science/hal-03745128v1>

Submitted on 3 Aug 2022

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

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



HAL Authorization

UN PANORAMA DE LA RECHERCHE EN CONCEPTION DE LA CONDUITE DES SYSTÈMES DE PRODUCTION

Bruno Denis et Jean-Jacques Lesage
LURPA — ENS de Cachan
61 avenue du Président Wilson
F-94235 Cachan cedex — France

Résumé

Dans la vie d'un système de production, la conception de sa commande et de sa conduite détermine de manière non négligeable ses performances. Des méthodes et des formalismes sont à la disposition des ingénieristes qui mènent à bien cette phase de conception. Toutefois ces méthodes et ces formalismes sont perfectibles et doivent suivre les nouvelles attentes des utilisateurs. La communauté scientifique française est très active sur ce sujet. Cette publication se propose d'en brosser un tableau, pour en montrer les particularités et en mettant l'accent sur quelques initiatives prometteuses.

1. Introduction

La recherche de productivité est un objectif important de la conception de la conduite des systèmes de production. Elle se traduit par d'importants enjeux techniques et économiques. Si la conception des systèmes continus jouit d'un cadre méthodologique et de modélisation rigoureux depuis de nombreuses années, cet aspect est jusqu'à présent peu développé pour les systèmes discontinus. Il existe encore trop souvent un décalage important entre la complexité des systèmes discontinus et l'artisanat de leur conception. L'objectif de cette publication est de présenter en quelques lignes un panorama représentatif des travaux et des perspectives de recherche spécifiques aux communautés industrielle et universitaire françaises.

Les concepts relatifs à la modélisation et aux méthodes de conception sont rappelés et clarifiés section 2. L'intérêt de la *modélisation pour l'amélioration de la qualité* des systèmes étant souligné par la communauté industrielle [Delcuvelerie, 1995], la section 3 présente les travaux allant dans ce sens, en particulier ceux portant sur l'amélioration des formalismes, et la validation des modèles. Au delà de la production de modèles, le développement d'un système de production intégré doit s'appuyer sur des *méthodes intégrées, multi-modèles* (pour prendre en compte des différents axes du système) [Kiefer et al., 1995], et accompagner les acteurs *tout au long du cycle de vie* de l'automatisme [Lebourgeois, 1995]. La section 4 est consacrée à la présentation des méthodes les plus avancées dans la prise en compte de ces aspects.

2. Modèles et Méthodes

Dans le domaine de la conception des systèmes de production le terme «modèle» est fréquemment utilisé dans deux sens différents. A titre d'exemple la locution «un modèle réseau de Petri» désignera selon le contexte un formalisme de type graphe orienté ou bien le résultat de la

modélisation d'un système particulier¹. Les deux acceptions du terme étant abordées dans cet article, nous proposons de lever toute ambiguïté en désignant toute vue partielle d'un système par *modèle*. *Formalisme* et *langage* seront utilisés lorsque nous considérerons le moyen symbolique d'exprimer un modèle du système.

Une *méthode* se définit comme une démarche raisonnée permettant d'atteindre un but fixé. On distingue les méthodes *mono-modèle* des méthodes *multi-modèles*. Avec son cycle auteur/lecteur, la méthode SADT [Ross, 1977] est une méthode mono-modèle car elle établit un processus menant à la création d'un modèle exprimé dans un langage unique : le formalisme des actigrammes. A l'inverse, la méthode SA-RT [Ward et Mellor, 1985] [Hatley et Pirbhai, 1987] est une méthode multi-modèles car elle donne une cohérence à un ensemble de modèles issus de formalismes différents : les diagrammes de flots de données (DFD), les diagrammes de flots de contrôles (CFD), et les automates finis (CSPEC).

3. Modélisation de la commande des systèmes de production

3.1. Validation de modèles

L'activité de conception de la conduite des systèmes comporte de plus en plus fréquemment une activité de conception de modèles du système. Cette démarche permet d'obtenir des solutions plus rigoureuses, plus évolutives, et d'une plus grande maintenabilité. En contre partie il est légitime de vérifier l'adéquation et la pertinence des modèles élaborés. Deux approches vont être présentées : l'une où les formalismes ont été conçus pour que les modèles élaborés soient validables, et l'autre où un formalisme existant est muni d'outils de validations.

3.1.1. Formalismes conçus pour la validation : l'exemple des langages réactifs

Classiquement on différencie les aspects «réactifs» d'un système de ces aspects «transformationnels» [Harel et Pnueli, 1985]. On qualifie souvent de «réactif» la conduite des systèmes de production compte tenu de l'importance de son comportement dynamique. Après avoir constaté que les outils classiques de conception (plus adaptés aux systèmes transformationnels) ne convenait pas à la modélisation des systèmes réactifs [Berry, 1989], des formalismes spécifiques ont été développés ou adaptés. Parmi eux, se trouvent les langages réactifs, construits sur une base formelle permettant ainsi la validation des modèles.

Leurs points communs sont la modélisation du parallélisme et le déterminisme de leur comportement, alors qu'ils diffèrent par leur nature et par l'hypothèse de synchronisme retenue. Les uns sont impératifs (ou encore procéduraux), alors que les autres sont déclaratifs (ou encore orienté flots de données). Quant à l'hypothèse de synchronisme, elle implique des durées d'actions nulles et retient la possibilité de simultanéité des occurrences d'événements, alors que l'asynchronisme autorise l'apparition d'occurrences d'événements entre le début à la fin d'une action et ne considère jamais comme simultanés deux événements distincts. Le tableau 1 donne une classification sommaire des principaux langages réactifs développés au sein de la communauté scientifique française.

1. On notera que cet usage ambigu du vocabulaire n'est pas une spécificité de la langue française. Le même phénomène existe autour du terme anglo-saxon «model» comme par exemple avec la locution «Petri net model».

Concernant la validation de modèles, et bien que les langages de type déclaratif puissent se prêter aux méthodes de preuves déductives c'est-à-dire à base de démonstrations de théorèmes fondés sur une axiomatique du langage [Halbwachs et al., 1991a], l'ensemble des langages réactifs (impératifs et déclaratifs) utilisent la méthode par énumération d'états. Chaque langage réactif propose un «compilateur» qui construit un automate fini à l'image du comportement logique du programme réactif compilé. Avec cet automate, on cherche à valider deux grandes familles de propriétés : la sûreté (une situation donnée ne peut jamais se produire) et la vivacité (une situation donnée peut toujours se reproduire). Pour cela les travaux issus de la recherche en informatique théorique offrent des techniques et des outils [Sifakis, 1989]. Le lecteur trouvera dans [Elkhatabi et al., 1995] un exemple de validation appliqué à la commande d'un tour à commande numérique, par utilisation conjointe d'ESTEREL et d'un ensemble d'automates.

Tableau 1 : Classification des langage réactifs

<i>Asynchrone</i>	<i>Synchrone [Benveniste et Berry, 1991] [André et Péraldi, 1993b]</i>	
	<i>Déclaratif</i>	<i>Impératif</i>
ELECTRE [Roux et al., 1992]	LUSTRE [Halbwachs et al., 1991b] SIGNAL [le Guernic et al., 1991]	ESTEREL [Boussinot et de Simone, 1991]

3.1.2. Mise en place d'outils de validation : l'exemple du Grafcet

Jusqu'au début des années 90, on opposait les approches conviviales comme le Grafcet [UTE, 1993] aux approches dont la sémantique est plus rigoureuse comme pour les modèles synchrones [Benveniste et Berry, 1991]. Or contrairement aux langages réactifs qui sont encore peu répandus dans le monde industriel (à l'exception des partenariats université-industrie), le Grafcet jouit d'une excellente implantation dans le domaine de la spécification et de la conception des systèmes réactifs. Aujourd'hui des travaux allient le fruit des recherches sur les automates finis et la praticabilité du Grafcet.

Quatre approches s'appuient sur les vérifications de propriétés des automates finis pour la validation de modèles grafcet (figure 1). Trois d'entre elles proposent de traduire les grafcets dans un langage réactif puis de construire les automates finis correspondants aux modèles traduits, ces langages réactifs sont : ESTEREL [André et Péraldi, 1993a], SIGNAL [le Parc, 1993], et ELECTRE [Roux et Rusu, 1994]. Dans la quatrième approche, le comportement exprimé par les grafcets est directement représenté par des automates finis [Roussel, 1994]. Sur les automates finis, il est alors possible de vérifier des propriétés telles que : l'absence de situation de blocage, la possibilité de retour à la situation initiale, la non émission simultanées de deux sorties, la non émission séquentielle de sorties dangereuses...

Il faut noter que ces travaux reposent sur une explicitation exhaustive du comportement (ou d'une partie du comportement) du grafcet étudié. Un accent a par conséquent été porté sur l'automatisation des démarches. Ainsi la plupart d'entre eux proposent ou développent des outils automatiques de traduction de grafcets en langage réactif ou directement en automate fini.

La théorie des automates finis n'est pas le seule à avoir inspirée des techniques de validation de modèles grafcet. La théorie des graphes est utilisée dans [Lesage et Roussel, 1993] pour valider la cohérence des ordres de forçage dans un grafcet, que la syntaxe seule ne peut garantir. La

technique des invariants linéaire de places et de transitions dans un réseau de Petri [Murata, 1989], est utilisée dans [Aygaliac et Denat, 1993] pour vérifier certaines propriétés se limitant à la structure du grafcet (graphe sauf, borné, vivace), l'interprétation qui est incluse dans les réceptivités et les actions ne pouvant pas être validée par cette technique. Deux explorations naissantes ont attirées notre attention. L'une utilise la logique temporelle [Manna et Pnueli, 1992] pour exprimer et démontrer des propriétés de sécurité, de vivacité et également de temps de réponse [Zaytoon et al., 1995]. L'autre s'appuie sur l'algèbre (Max,+) [Gaubert, 1992] pour vérifier a priori des propriétés dynamiques du modèle [GSIP,]

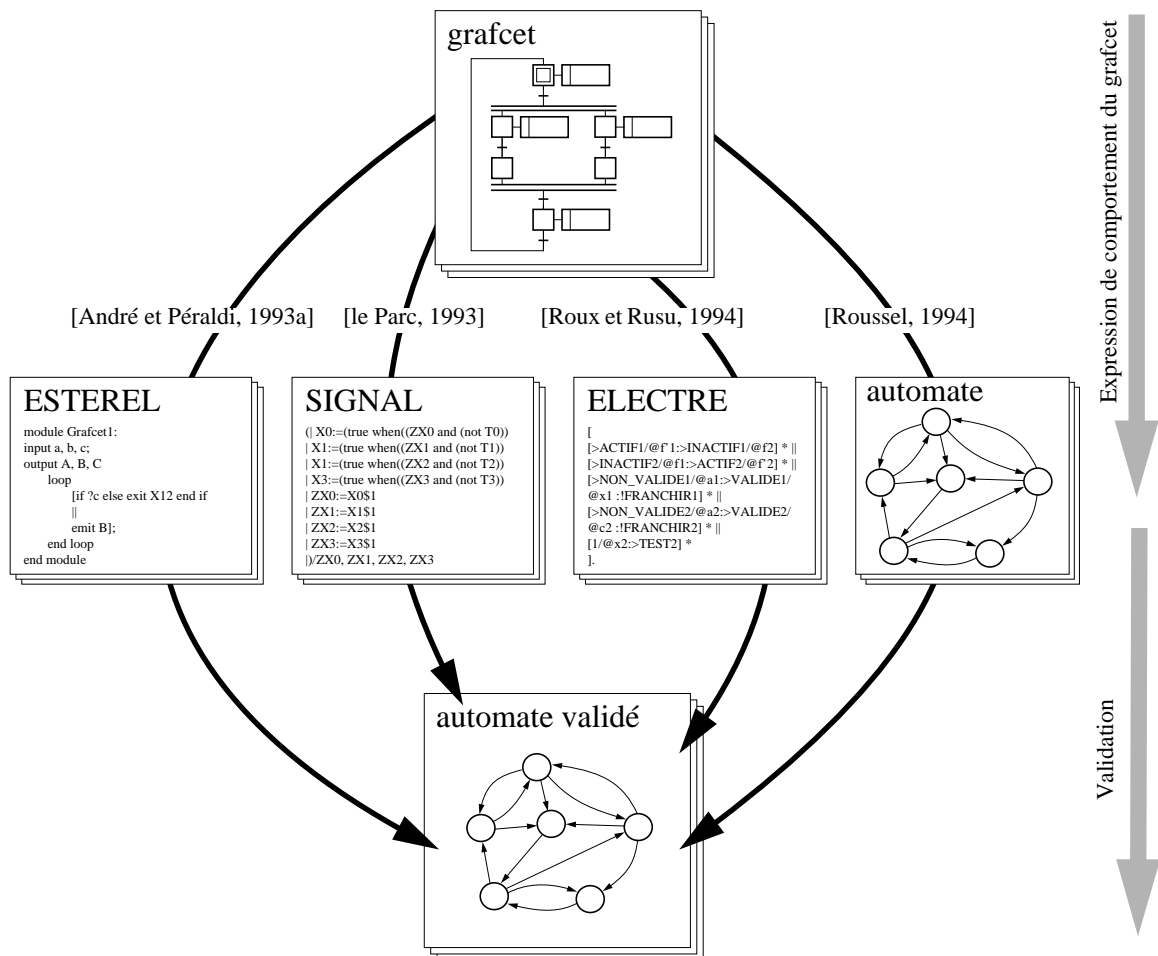


Figure 1 : Quatre stratégie de validation de grafcets par vérification de propriétés à l'aide d'un automate fini

3.2. Etudes sur la sémantique des formalismes

De manière complémentaire à la validation des modèles, les travaux autour de la sémantique des formalismes ont un impact déterminant sur la productivité et la qualité de la conception de la conduite des systèmes. On distingue deux axes de recherche. Le premier contribue à l'enrichissement de la sémantique des formalismes pour qu'ils deviennent plus pertinent à un domaine donné. On augmente ainsi l'efficacité et la productivité de la conception. Le deuxième axe, plus novateur, s'attache à expliciter et formaliser la sémantique des langages. L'objectif

majeur de cette approche est de réduire les interprétations possibles d'un formalisme à une seule interprétation. La qualité de la conception est de ce fait augmentée.

3.2.1. Enrichissement sémantique des formalismes

Depuis de nombreuses années les réseaux de Petri ont fait l'objet de multiples enrichissements. Certains de ces enrichissements font aujourd'hui partie de la culture des automaticiens, par exemple les réseaux de Petri stochastiques [Natkin, 1980], ou encore les réseaux de Petri temporisés [Chretienne, 1984]. Parmi les travaux récents citons les réseaux de Petri contrôlés [Ghabri et Ladet, 1995a] [Ghabri et Ladet, 1995b], les réseaux de Petri continus à vitesses constantes [David et Alla, 1987] et à vitesses variables [Zerhouni, 1992], les réseaux de Petri à marquage flou [Cardoso, 1990], et les réseaux de Petri à objets [Hssain, 1993]. Les réseaux de Petri et leurs extensions étant un sujet en soit, nous laissons le lecteur se reporter à la littérature spécifique au sujet, comme par exemple les actes des conférences dédiées au réseaux de Petri [PNPM, 1995] [ICAPPN, 1996].

Concernant la modélisation fonctionnelle de la conduite des systèmes de production le formalisme SADT est reconnu pour sa puissance d'expression. Cependant son absence complète de sémantique temporelle à conduit à l'apparition de nombreuses extensions telles que SA-RT. Une approche originale est proposée dans [Zaytoon et al., 1993]. Le formalisme SADT est utilisé sans aucun aménagement, puis l'analyste peut construire un «SADT temporel» sur la trame du précédent en précisant par des attributs graphiques le rôle des flots de données et les mécanismes de synchronisation dans les activités (figure 2a). Outre le fait qu'il ouvre la voie à la simulation, le «SADT temporel» peut être considéré comme un formalisme charnière entre l'analyse fonctionnelle et la conception de la partie dynamique de la conduite des systèmes.

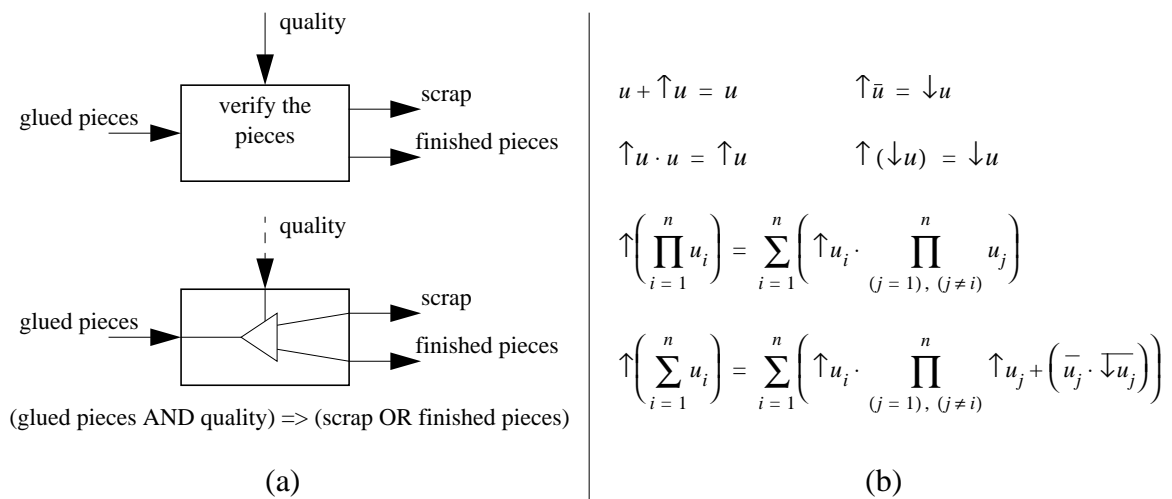


Figure 2 : (a) Exemple d'activité en «SADT temporel» [Zaytoon et al., 1993]
 (b) Exemple de propriétés sur les fronts dans l'algèbre de [Roussel et Lesage, 1993]

La notion de front montant ou descendant d'une variable booléenne est couramment utilisée et en particulier avec des formalismes tel que le Grafset. Dans [Roussel et Lesage, 1993] les auteurs montrent que cette notion de front est plus intuitive que rigoureuse, ce qui interdit toutes manipulations des fronts dans le cadre d'un calcul symbolique d'une expression booléenne. Pour y remédier, une nouvelle algèbre de Boole est proposée avec deux nouveaux opérateurs

unaires : le front montant et le front descendant. Cette algèbre permet d'établir un ensemble de propriétés relatives aux nouveaux opérateurs fronts (figure 2b).

3.2.2. Formalisation des langages

La formalisation des langages n'est pas une finalité en soit, mais il est clair qu'un formalisme rigoureux et consistant permet d'atteindre la cohérence syntaxique des modèles, et une lecture non ambiguë du message porté par le modèle. De plus, tout comme un système de production, un langage peut être amené à évoluer pour accueillir de nouveaux concepts. Ainsi lors de toute évolution et pour ne pas rompre la cohérence initiale du langage il est préférable de travailler sur un modèle, c'est-à-dire une «formalisation du langage» que l'on appelle le *métamodèle*. Si certains langages sont construits dès leur origine sur un formalisme rigoureux (comme par exemple les mathématiques pour les réseaux de Petri), beaucoup restent décrits de manière textuelle, donc soumis à interprétations et à ambiguïtés. Le tableau 2 énumère les premières initiatives de métamodélisation dont les objectifs respectifs sont différents. Ainsi dans [Coad et Yourdon, 1991] il s'agit simplement de clarifier le définition du formalisme, dans [Lhoste et al., 1992] et [Lhoste, 1994] il s'agit de proposer un cadre rigoureux de travail pour l'élaboration d'extensions au Grafcet, dans [Lamperrière, 1994] l'objectif est de construire un éditeur informatique, et dans [Lesage et al., 1992] il s'agit de formaliser l'utilisation intégrée de plusieurs langages. Dans tous les cas, une meilleure maîtrise des formalismes est recherchée afin d'augmenter la qualité des systèmes conçus.

Tableau 2 : Exemples de travaux utilisant le concept de métamodélisation

		<i>Langages du métamodèle</i>		
		<i>Entité/Relation</i>	<i>Formalisme à objets</i>	<i>Algèbre</i>
<i>langages à modéliser</i>	<i>Entité/Relation</i>	[Lesage et al., 1992] [Lamperrière, 1994]		
	<i>Formalisme à objets</i>		[Coad et Yourdon, 1991] ^a	
	<i>Grafcet</i>	[Lhoste, 1994]	[Lhoste et al., 1992]	[Lhoste, 1994]

a. cet exemple issu du génie logiciel, est un des cas particuliers de langage apte à se modéliser lui-même

4. Méthodes de conception

4.1. Méthode d'aide à l'élaboration d'un modèle

Comme nous l'avons précisé dans la section 2, une méthode est une démarche raisonnée pour atteindre un but donné et en particulier la construction d'un modèle. Elle traduit l'expérience et le savoir faire d'un ensemble d'utilisateurs. Il est donc courant de rencontrer des méthodes dites «maison» c'est-à-dire propres à chaque société. De ce fait il existe peu de travaux à caractères généraux sur les méthodes d'élaboration de modèle. Il est tout de même possible de dégager deux grands mécanismes : les méthodes s'appuyant sur un modèle de référence, et les méthodes par construction progressive. Les modèles de référence sont des formalisations de la connaissance pour des classes de problèmes donnés [Faure et al., 1994], comme par exemple le GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts) [ADEPA, 1981] pour l'étude des modes de fonctionnement des automatismes. Ils sont généralement élaborés dans le cadre

de réflexions collectives (exemples : projet ESPRIT n°688 CIMOSA [Williams et al., 1994], projet ESPRIT n°2172 DIAS [Galara et al., 1993]) pour être ensuite utilisés dans les projets. Une méthode s'appuyant sur un modèle de référence consiste en une instanciation puis une spécialisation du modèle. Quant à elles, les méthodes par construction progressive guident le concepteur en lui proposant une démarche de construction pas à pas du modèle. Ces méthodes sont très étroitement liées aux domaines d'emploi, et donc très peu universelles. On peut citer [Froment, 1993] qui propose une méthode d'aide à l'élaboration d'un grafcet.

4.2. Méthodes multi-modèles

Lorsque l'étendue et la complexité d'un système de production augmentent, certaines méthodes proposent l'utilisation conjointe de plusieurs modèles pour assurer le développement du système. En effet, quelque soit le formalisme utilisé, un modèle ne représente qu'un aspect d'un système et à un stade donné de son développement (figure 3a). L'objectif des méthodes multi-modèles est par conséquent d'assister l'ingénieur soit sur plusieurs aspects complémentaires du systèmes (figure 3b), soit sur plusieurs étapes successives de son développement (figure 3d), ou soit sur les deux à la fois (figure 3c).

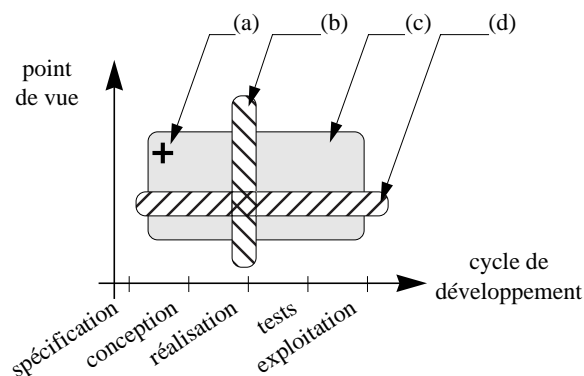


Figure 3 : Différents taux de couverture des méthodes

Cependant les méthodes couvrant à la fois le cycle de développement d'un système et ses différents point de vue sont peu courantes : la méthode *universelle* n'existe pas. Seules les méthodes s'appliquant à une classe de systèmes très restreints peuvent proposer une large couverture. C'est par exemple le cas de la méthode MERISE [Tardieu et al., 1993] qui ne s'applique qu'au domaine des systèmes d'informations d'entreprise, mais qui traite à la fois le point de vue des données et des traitements et les aspects allant du conceptuel à la réalisation. D'autres exemples, dans le domaine du génie logiciel sont les méthodes à objets qui permettent également de couvrir le développement des systèmes de la spécification au codage, mais dont l'emploi se limite aux systèmes réduits à un simple programme informatique. Parmi les méthodes plus proches de la conception des systèmes de production, on peut citer la méthode SA-RT [Ward et Mellor, 1985] [Hatley et Pirbhai, 1987] qui s'applique à des systèmes dits *embarqués* dont l'ingénieur doit concevoir à la fois l'aspect logiciel et l'aspect matériel. Si cette fois les systèmes visés sont plus complexes et variés, en contre partie la méthode SA-RT est peu détaillée sur les phases les plus en aval de la conception du système, en particulier la conception de son architecture matérielle aussi que sa réalisation. Une des contreparties de la généralité d'une

méthode (capacité à traiter des systèmes variés) est la quasi impossibilité de couvrir la totalité du développement des systèmes cibles sous tout leurs aspects. De multiples travaux ont pour objectifs l'élaboration de méthodes multi-modèles spécifiques au domaine du génie automatique. Elles sont basées sur un postulat : aucun formalisme ne peut à lui seul couvrir le développement complet d'un système de production moderne. A l'instar des systèmes de production conçus (systèmes dont les activités sont intégrées entre elles et aux activités de l'entreprise [Williams et al., 1994]), les méthodes multi-modèles sont des «systèmes» intégrant des modèles pour proposer une représentation cohérente d'un système de production selon plusieurs points de vue et au cours de plusieurs phases de son développement. Les méthodes suivantes sont des exemples représentatifs :

- ACSY (Analyse de Commande des SYstèmes) [Chapurlat et al., 1993] intègre le formalisme des diagrammes de flots, le langage textuel, et le Grafcet, pour couvrir l'ensemble du cycle de vie des systèmes à événements discrets sous les aspects fonctionnels, dynamiques et traitements de données ;
- CASPAIM (Conception Assistée de Système de Production Automatisé dans l'Industrie Manufacturière) [Farah et al., 1995] intègre des réseaux de Petri, des statecharts et la méthode HOOD pour couvrir en particulier les phases d'analyse, de conception et d'implantation logicielles du système de contrôle-commande d'un système flexible de production manufacturière ;
- GRAI-GIM (GRAI Integrated Method) [Roboam et al., 1989] intègre les méthodes GRAI, IDEF0 et MERISE pour prendre en compte les aspects décisionnels, informationnels, opératifs et physique des systèmes de production ;
- SPEX [Tixador, 1989] [Panetto, 1991] intègre des grafquets, des schémas à relais, des logigrammes, et du langage C dans des boîtes fonctionnelles regroupées en diagrammes fonctionnels pour la spécification, le prototypage et la simulation de la partie commande des systèmes ;
- l'extension de l'analyse fonctionnelle proposée dans [Zaytoon et al., 1994], intègre la SADT et les réseaux de Petri pour l'étude de la sécurité opérationnelle des systèmes automatisés de production ;
- [Denis et al., 1995] proposent une méthode intégrant des diagrammes de flots, des diagrammes d'interconnexion d'équipements et des réseaux de Petri pour concevoir et simuler des architectures de commande de systèmes complexes de production ;
- [Sayat et Ladet, 1993] proposent une méthode intégrant des grafquets et des réseaux de Petri pour s'intéresser au niveau coordination et au niveau commande des systèmes de production.

4.3. Formalisation de l'intégration

Bien que spécifique, chaque méthode doit pouvoir bénéficier de bases théoriques communes. C'est pour cette raison que des travaux se portent sur les mécanismes constitutifs des méthodes et en particulier le mécanisme d'intégration de plusieurs formalismes en une méthode spécifique. Dans le cadre du projet PTA (Poste de Travail adapté aux besoins de l'automaticien) [Frachet, 1990] deux notions d'intégration ont été mises en évidence : l'intégration forte par le partage de données au sein d'un Atelier Intégré pour le Génie Automatique (AIGA), et l'inté-

gration faible par communication au moyen de standards d'échange entre AIGA différents. L'intégration repose sur un *modèle de données générique* BASE-PTA (Base Application et Standard d'Échange) [AFNOR, 1992] formalisant les invariants des systèmes automatisés de production. La mise en place d'un atelier intégré se fait par spécialisation du modèle BASE-PTA comme par exemple dans [Soriano et al., 1993]. Dans la mouvance de ces travaux, le projet CRISTAL (Conception et Réalisation Intégrées des SysTèmes Automatisés Logiques) [Maghoul et al., 1993] [Lhoste, 1994] à également retenu le principe de l'intégration par les données mais s'appuie sur des *modèles particuliers des systèmes d'information* d'outils existant pour remonter vers une base formelle identifiant des «objets d'automatisation», véritables invariants caractérisant les métiers de l'automatisation.

En amont des projets PTA et CRISTAL qui se concentrent sur l'intégration d'outils en ateliers logiciels, les principes de l'intégration des formalismes et des méthodes font également l'objet de travaux. La métamodélisation, déjà évoquée pour la formalisation des langages, permet également de réunir et donc d'intégrer la sémantique d'une méthode et de ses langages au sein d'un unique modèle [Denis et al., 1993]. Cela revient à faire un modèle des concepts contenus dans une méthode et les langages la constituant. Si jusqu'à présent, seul le formalisme des données a été utilisé pour élaborer les métamodèles (une fois de plus, il s'agit d'une intégration par les données), d'autre langage plus riche sont à l'étude.

5. Conclusions

Cet article montre à quel point les recherches en conception des systèmes de production sont actives et prolifiques. Depuis le milieu des années 70 et plus sensiblement à partir du début des années 80, des formalismes propres au génie automatique ont vus le jour (ex : Grafcet) et des langages issus de disciplines connexes comme le génie logiciel ont été adaptés (ex : analyse fonctionnelle). L'expérience acquise lors de l'utilisation de ces formalismes c'est traduite par la mise en place de nombreuses méthodes apparaissant au milieu des années 80 (ex : CASPAIM). Tout en continuant à voir apparaître de nouvelles méthodes, le début des années 90 voient l'essor de la validation et de la rationalisation du travail des ingénieristes. Après s'être pourvu des «outils» de base (langages et méthodes) le monde de la conception des systèmes de production est entré de plein pied dans une ère de productivité par le biais de la validation de ses modèles, de la création de méthodes adéquates aux problèmes spécifiques, et de la formalisation de ses langages et ses méthodes.

6. Références

- [ADEPA, 1981] ADEPA. *Guide d'Étude des Modes de Marches et d'Arrêts*. ADEPA, 1981.
- [AFNOR, 1992] AFNOR. Représentation des systèmes de contrôle et de commande des systèmes automatisés – Contribution à l'intégration des outils XAO du génie automatique : modèle conceptuel Base-PTA. Norme expérimentale Pr Z 68-901, AFNOR, 1992.
- [André et Péraldi, 1993a] C. André et M.-A. Péraldi. GRAFCET and synchronous languages. *Automatique Productique Informatique Industrielle*, vol. 27, n°1, pp. 95–105, 1993.
- [André et Péraldi, 1993b] C. André et M.-A. Péraldi. Synchronous programming: introduction and application to industrial process control. In *proceedings of computers in design, manufac-*

- turing, and production*, 7th annual european computer conference, pp. 461–470, Paris-Evry, France. 24–27 Mai 1993.
- [Aygaliac et Denat, 1993] P. Aygaliac et J.-P. Denat. Validation of functional GRAFCET models and performance evaluation of the associated system using Petri nets. *Automatique Productive Informatique Industrielle*, vol. 27, n°1, pp. 81–93, 1993.
- [Benveniste et Berry, 1991] A. Benveniste et G. Berry. The synchronous approach to reactive and real-time systems. *Proceeding of the IEEE*, vol. 79, pp. 1270–1282, Sept. 1991.
- [Berry, 1989] G. Berry. Real time programming: special purpose or general purpose languages. Rapport de recherche RR 1065, INRIA, Août 1989.
- [Boussinot et de Simone, 1991] F. Boussinot et R. de Simone. The Esterel Language. *Proceeding of the IEEE*, vol. 79, pp. 1293–1304, Sept. 1991.
- [Cardoso, 1990] J. Cardoso. *Sur les réseaux de Petri à marquage flou*. Thèse de doctorat, Université Paul Sabatier, Toulouse, 23 Oct. 1990.
- [Chapurlat et al., 1993] V. Chapurlat, T. Giaccone, G. Monneret, F. Pereyrol, et D. Simottel. A structured model for specification of discrete events systems: the ACSY model. *Automatique Productive Informatique Industrielle*, vol. 27, n°1, pp. 65–80, 1993.
- [Chretienne, 1984] P. Chretienne. *Les Réseaux de Petri temporisés*. Thèse d'état, Université de Paris VI, Juin 1984.
- [Coad et Yourdon, 1991] P. Coad et E. Yourdon. *Object Oriented Design*. Prentice Hall, Juin 1991.
- [David et Alla, 1987] R. David et H. Alla. Continuous Petri Nets. In *8th European Workshop on Application and Theory of Petri Nets*, pp. 275–294, Zaragoza, Spain. Juin 1987.
- [Delcuvellerie, 1995] J.-L. Delcuvellerie. Méthode de conception d'architecture de contrôle commande : l'expérience d'un intégrateur. In *Journées d'études SAPID*, Paris. 30–31 Mai 1995.
- [Denis et al., 1993] B. Denis, J.-J. Lesage, et G. Timon. Toward a theory of integrated modelling. *Journal of Design Sciences and Technology*, vol. 2, pp. 87–96, Oct. 1993.
- [Denis et al., 1995] B. Denis, J.-J. Lesage, et G. Timon. Formalisation de la conception d'architecture de la conduite des systèmes de production. In *2ème Conférence Internationale sur l'Automatisation Industrielle*, vol. I, pp. 257–264, Nancy, France. 7–9 Juin 1995.
- [Elkhatabi et al., 1995] S. Elkhatabi, D. Corbeel, et J.-C. Gentina. Surveillance par filtre de comportement. *Automatique Productive Informatique Industrielle*, vol. 29, n°3, pp. 307–330, 1995.
- [Farah et al., 1995] A. Farah, E. Craye, et J. Bourey. Une méthode hiérarchique orientée objet pour l'implantation répartie du système de contrôle/commande d'un SFPM. In *2ème Conférence Internationale sur l'Automatisation Industrielle*, vol. II, pp. 795–800, Nancy, France. 7–9 Juin 1995.
- [Faure et al., 1994] J.-M. Faure, F. Couffin, S. Lamperrière, et A. Bassand. An instantiation methodology for conceptual data model design in automation engineering. In *Proceeding of Workshop on Integrated Manufacturing Systems Engineering ISME'94*, Grenoble, France. 12–14 Déc. 1994.

- [Frachet, 1990] J.-P. Frachet. Le concept PTA : première mise en oeuvre. In *Automation 1990*, vol. 1, pp. 33–48, Paris la Défense, France. 16–18 Mai 1990.
- [Froment, 1993] B. Froment. Structural design with GRAFCET: towards a global method. *Automatique Productive Informatique Industrielle*, vol. 27, n°1, pp. 11–23, 1993.
- [Galara et al., 1993] D. Galara, J. Favennec, B. Jung, et G. Morel. Distributed Intelligent Actuators and Sensors. In *proceedings of computers in design, manufacturing, and production*, 7th annual european computer conference, pp. 79–85, Paris-Evry, France. 24–27 Mai 1993.
- [Gaubert, 1992] S. Gaubert. Théorie des systèmes linéaires dans les diodes. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris, 1992.
- [Ghabri et Ladet, 1995a] M.-K. Ghabri et P. Ladet. Les réseaux de Petri contrôlés et leurs applications — 1ère partie. *Automatique Productive Informatique Industrielle*, vol. 29, n°2, pp. 203–223, 1995.
- [Ghabri et Ladet, 1995b] M.-K. Ghabri et P. Ladet. Les réseaux de Petri contrôlés et leurs applications — 2ème partie. *Automatique Productive Informatique Industrielle*, vol. 29, n°2, pp. 225–242, 1995.
- [GSIP,] GSIP. Equipe Génie des Systèmes Intégrés de Production du Groupe génie de la production du CRAN.
- [Halbwachs et al., 1991a] N. Halbwachs, P. Caspi, P. Raymond, et D. Pilaud. Programmation et vérification des systèmes réactifs : le langage LUSTRE. *Technique et Science Informatiques*, vol. 10, n°2, pp. 139–157, 1991.
- [Halbwachs et al., 1991b] N. Halbwachs, P. Caspi, P. Raymond, et D. Pilaud. The synchronous data flow programming language LUSTRE. *Proceeding of the IEEE*, vol. 79, pp. 1305–1320, Sept. 1991.
- [Harel et Pnueli, 1985] D. Harel et A. Pnueli. *Proceeding of NATO Advanced Study Institute on Logics and Models for Verification and Specification of Concurrent Systems*, vol. 13 of *NATO Advanced Study Institute Series F.*, chapitre On the development of reactive systems: logic and models of concurrent systems, pp. 477–498. Springer-Verlag, New York, 1985.
- [Hatley et Pirbhai, 1987] D. J. Hatley et I. A. Pirbhai. *Strategies for real-time system specification*. Dorset House, 1987.
- [Hssain, 1993] A. A. Hssain. *Conduite hiérarchisée intégrée des ateliers manufacturiers flexibles : une approche mixte objets/réseaux de Petri*. Thèse de doctorat, INPG, 21 Déc. 1993.
- [ICAPPN, 1996] ICAPPN. *17th International Conference on Application and Theory of Petri Nets*, Osaka, Japan. 24–28 Juin 1996.
- [Kiefer et al., 1995] F. Kiefer, A. Michez, C. L. Guirriec, et J. M. Gazzo. Les enjeux de la conception des systèmes intégrés de production. In *Congrès international de génie industriel de Montréal*, Québec, Canada. 18–20 Oct. 1995.
- [Lamperrière, 1994] S. Lamperrière. *Formalisation d'un outils de modélisation entité-relation à l'aide d'un métamodèle*. Rapport de recherche, dea de production automatisée, ENS de Caen, 1994.

- [le Guernic et al., 1991] P. le Guernic, T. Gautier, M. le Borgne, et C. le Maire. Programming real-time applications with signal. *Proceeding of the IEEE*, vol. 79, pp. 1321–1336, Sept. 1991.
- [le Parc, 1993] P. le Parc. *Apport de la méthodologie synchrone pour la définition et l'utilisation du langage GRAFCET*. Thèse de doctorat, Université de Rennes I, Jan. 1993.
- [Lebourgeois, 1995] F. Lebourgeois. La sûreté de conduite des installations chimiques : de la conception à l'exploitation. In *2ème Conférence Internationale sur l'Automatisation Industrielle*, vol. I, pp. XXVII–XXXVI, Nancy, France. 7–9 Juin 1995.
- [Lesage et al., 1992] J.-J. Lesage, B. Denis, et G. Timon. Une approche formelle de la modélisation intégrée. In *Actes du Symposium International "La conception en 2000 et au-delà"*, pp. 343–353, Strasbourg, France. 24–27 Nov. 1992.
- [Lesage et Roussel, 1993] J.-J. Lesage et J.-M. Roussel. Hierarchical approach to GRAFCET using forcing order. *Automatique Productique Informatique Industrielle*, vol. 27, n°1, pp. 25–38, 1993.
- [Lhoste, 1994] P. Lhoste. *Contribution au génie automatique : concepts, modèles, méthodes et outils*. Habilitation à diriger des recherches, Université de Nancy I, 10 Fév. 1994.
- [Lhoste et al., 1992] P. Lhoste, H. Panetto, et M. Roesch. Grafcet : de la syntaxe à la sémantique. In *GRAFCET'92*, pp. 13–25, Paris, France. 25-26 Mars 1992.
- [Maghoul et al., 1993] B. Maghoul, P. Lhoste, et O. Douchin. CRISTAL : Conception et Réalisation Intégrées des SysTèmes Automatisés Logiques. Rapport intermédiaire, ANVAR, CRAN/EURILOR, 1993.
- [Manna et Pnueli, 1992] Z. Manna et A. Pnueli. *The Temporal logic of reactive and concurrent systems: specification*. Springer-Verlag, 1992.
- [Murata, 1989] P. Murata. Petri nets: properties, analysis and applications. *Proceeding of IEEE*, vol. 47, n°4, pp. 541–580, 1989.
- [Natkin, 1980] S. Natkin. *Les réseaux de Petri Stokastiques*. Thèse de docteur ingénieur cnam, CNAM Paris, Juin 1980.
- [Panetto, 1991] H. Panetto. *Une contribution au génie automatique : le prototypage des machines et systèmes automatisés de production*. Thèse de doctorat, Université de Nancy I, 1991.
- [PNPM, 1995] PNPM. *6th International Workshop on Petri nets and Performance Models*, Durham, North Carolina, USA. 3–6 Oct. 1995.
- [Roboam et al., 1989] M. Roboam, M. Zanettin, et L. Pun. GRAI-IDEF0-Merise (GIM): integrated methodology to analyse and design manufacturing systems. *Computer integrated manufacturing systems*, vol. 2, n°2, pp. 82–98, 1989.
- [Ross, 1977] D. T. Ross. Structured Analysis (SA): a language for communicating ideas. *IEEE Transactions on Software Engineering*, vol. SE-3, pp. 16–34, Jan. 1977.
- [Roussel, 1994] J.-M. Roussel. *Analyse de grafcet par génération de logique de l'automate équivalent*. Thèse de doctorat, Ecole Normale Supérieure de Cachan, 16 Déc. 1994.

- [Roussel et Lesage, 1993] J.-M. Roussel et J.-J. Lesage. Une algèbre de Boole pour l'approche événementielle des systèmes logiques. *Automatique Productive Informatique Industrielle*, vol. 27, n°5, pp. 541–560, 1993.
- [Roux et al., 1992] O. Roux, D. Creusot, F. Cassez, et J.-P. Elloy. Le langage asynchrone ELECTRE. *Technique et Science Informatiques*, vol. 11, n°5, pp. 35–66, 1992.
- [Roux et Rusu, 1994] O. Roux et V. Rusu. Du GRAFCET au langage réactif ELECTRE. *Automatique Productive Informatique Industrielle*, vol. 28, n°2, pp. 131–157, 1994.
- [Sayat et Ladet, 1993] B. Sayat et P. Ladet. Control specification of a production system using GRAFCET and Petri nets. *Automatique Productive Informatique Industrielle*, vol. 27, n°1, pp. 53–64, 1993.
- [Sifakis, 1989] J. Sifakis. *Automatic verification methods for finite state systems*, vol. 407 of *Lecture notes in computer science*. Springer-Verlag, 1989.
- [Soriano et al., 1993] T. Soriano, E. Getan, F. Louni, et J.-P. Frachet. GAMA project: toward an integrated maintenance environment for AMSs. In *International Conference on Industrial Engineering and Production Management*, vol. I, pp. 497–507, Mons, Belgium. 2–4 Juin 1993.
- [Tardieu et al., 1993] H. Tardieu, A. Rochfeld, et R. Colletti. *La Méthode MERISE : principes et outils*, vol. Tome 1. Éditions d'Organisation, 1993.
- [Tixador, 1989] J.-M. Tixador. *Une contribution au génie automatique : la spécification exécutable des machines et systèmes automatisés de production*. Thèse de doctorat, Université de Nancy I, 1989.
- [UTE, 1993] UTE. Diagramme fonctionnel GRAFCET. Norme C 03-191, Union Technique de l'Électricité, 1993.
- [Ward et Mellor, 1985] P. T. Ward et S. J. Mellor. *Structured development for real-time systems*, vol. 1–3 of *Yourdon computing series*. Prentice-Hall, 1985.
- [Williams et al., 1994] T. J. Williams, P. Bernus, J. Brosvic, D. Chen, G. Doumeingts, L. Nemes, J. L. Nevins, B. Vallespir, J. Vlietstra, et D. Zoetekouw. Architecture for integrating manufacturing activities and entreprises. *Computers in industry*, vol. 24, pp. 111–139, Sept. 1994.
- [Zaytoon et al., 1995] J. Zaytoon, P. de Loor, et G. Villerman-Lecollier. Du RTM/RTTL pour l'analyse et la vérification des systèmes commandés par grafcet. In *Ouvrage collectif sur le grafcet à paraître*. Hermès, Paris, 1995.
- [Zaytoon et al., 1993] J. Zaytoon, E. Niel, A. Mille, et A. Jutard. A temporal SADT for automated manufacturing systems. In *International Conference on Industrial Engineering and Production Management*, vol. I, pp. 154–164, Mons, Belgium. 2–4 Juin 1993.
- [Zaytoon et al., 1994] J. Zaytoon, G. Villerman-Lecollier, et P. de Loor. Associating discrete part manufacturing systems semantics to SADT by means of Petri nets. In *IFAC Conference on integrated systems engineering*, pp. 293–298, Baden-Baden, Germany. Sept. 1994.
- [Zerhouni, 1992] S. N. Zerhouni. *Application des réseaux de Petri continus à l'analyse dynamique des systèmes de production*. Thèse de doctorat, INPG, Grenoble, Jan. 1992.