

79/67
UNIVERSITE DE NANCY I
U.E.R. DE SCIENCES MATHÉMATIQUES

CENTRE DE RECHERCHE EN
INFORMATIQUE DE NANCY

SCN ⁷⁹/₈₇ A

UNE APPROCHE CONCEPTUELLE DES
PROBLÈMES DE SYNCHRONISATION



THÈSE

PRÉSENTÉE POUR L'OBTENTION DU
DOCTORAT DE SPÉCIALITÉ EN INFORMATIQUE

SOUTENUE LE 12 OCTOBRE 1979

PAR CHRISTIAN RICHARD

JURY :

PRÉSIDENT C. PAIR
EXAMINATEURS : J.C. DERNIAME
O. FOUCAUT
M. GRIFFITHS
W. LITWIN
C. ROLLAND

UNIVERSITE DE NANCY I
U.E.R. DE SCIENCES MATHÉMATIQUES

CENTRE DE RECHERCHE EN
INFORMATIQUE DE NANCY

UNE APPROCHE CONCEPTUELLE DES
PROBLÈMES DE SYNCHRONISATION



THÈSE

PRÉSENTÉE POUR L'OBTENTION DU
DOCTORAT DE SPÉCIALITÉ EN INFORMATIQUE

SOUTENUE LE 12 OCTOBRE 1979

PAR CHRISTIAN RICHARD

JURY :

PRÉSIDENT C. PAIR
EXAMINATEURS : J.C. DERNIAME
O. FOUCAUT
M. GRIFFITHS
W. LITWIN
C. ROLLAND

AVANT PROPOS

Remercier avec des paroles ou avec des mots a toujours constitué pour moi une démarche inhabituelle. Je serais nettement plus porté à exprimer ma reconnaissance spontanément en posant des gestes, ou par des attitudes, tant il est facile d'être hypocrite et calculateur avec des mots et tant l'omission et l'injustice peuvent aisément s'y dissimuler en opposition à la spontanéité du geste.

Néanmoins, il m'est agréable de reconnaître que ce travail n'aurait jamais pu voir le jour sans tous ceux qui m'ont soutenu, conseillé, critiqué jusqu'à présent, et à qui je voudrais exprimer ma gratitude :

- ma famille pour son soutien de toujours, mes amis pour m'avoir patiemment supporté et soutenu,
- toute l'équipe Remora sans qui ce travail n'aurait jamais été réalisé, tant il est vrai que le travail de recherche est avant tout le fruit d'une équipe, non seulement sur le plan intellectuel, mais aussi humain,
- mes professeurs, à qui je dois ma formation intellectuelle, et qui, tout au long de mes études ont su m'intéresser et me donner le goût de la réflexion,
- les membres du jury qui acceptent aujourd'hui de juger ce travail, après en avoir suivi l'élaboration et m'avoir guidé par leurs conseils,
- tous ceux qui ont participé à la réalisation matérielle de cette thèse.

SOMMAIRE

INTRODUCTION

I

CHAPITRE I : NOTRE POINT DE VUE SUR LES SOLUTIONS ACTUELLES AUX
PROBLEMES DE SYNCHRONISATION

| | | |
|---------|--|-----|
| I-1.- | Hypothèse de raisonnement | I-1 |
| I-2.- | Historique des approches de synchronisation : du mécanisme à la modélisation | I-1 |
| I-2.1.- | Premier stade : les mécanismes | I-1 |
| I-2.2.- | Deuxième stade : la modélisation | I-3 |
| I-2.3.- | Conclusion | I-6 |
| I-3.- | Essai de synthèse des approches existantes | I-6 |
| I-3.1.- | Nos propositions de définition | I-6 |
| I-3.2.- | La notion d'approche opératoire | I-7 |

CHAPITRE II : L'APPROCHE CONCEPTUELLE DE LA SYNCHRONISATION

| | | |
|--------------|---|-------|
| II-1.- | Les éléments de l'approche conceptuelle | II-1 |
| II-1.1.- | Le modèle conceptuel | II-1 |
| II-1.1.1.- | Que doit-il représenter : analyse du réel en termes de catégories | II-1 |
| II-1.1.2.- | Comment représenter : le modèle conceptuel | II-3 |
| II-1.1.2.1.- | Le concept de C-OBJET | II-5 |
| II-1.1.2.2.- | Le concept de C-OPERATION | II-6 |
| II-1.1.2.3.- | Le concept de C-EVENEMENT | II-7 |
| II-1.2.- | Le résultat de la modélisation | II-9 |
| II-1.2.1.- | La structure conceptuelle | II-9 |
| II-1.2.2.- | Exemple | II-11 |
| II-1.2.2.1.- | Enoncé du problème | II-11 |
| II-1.2.2.2.- | Sous-schéma dynamique | II-12 |
| II-1.2.2.3.- | Sous schéma de données | II-13 |
| II-1.2.2.4.- | Lexique | II-14 |
| II-1.3.- | La méthode | II-15 |
| II-1.3.1.- | Présentation de la méthode | II-15 |
| II-1.3.2.- | Exemple d'application de cette méthode | II-15 |
| II-1.4.- | La machine abstraite | II-19 |
| II-1.4.1.- | Définition et fonctionnement de la machine abstraite | II-19 |
| II-1.4.2.- | Représentation conceptuelle du fonctionnement de la machine abstraite | II-19 |
| II-2.- | Le modèle assure la prise en compte de la synchronisation au niveau conceptuel | II-21 |
| II-2.1.- | Le modèle conceptuel permet de satisfaire aux trois conditions de synchronisation | II-22 |
| II-2.2.- | Typologie des relations temporelles entre événements | II-26 |
| II-2.2.1.- | Dépendances chronologiques entre C-EVENEMENTS | II-27 |
| II-2.2.2.- | Dépendances sémantiques entre C-EVENEMENTS | II-36 |
| II-2.2.3.- | Graphe des dépendances chronologiques entre C-EVENEMENTS | II-39 |
| II-2.2.4.- | Conclusion sur les dépendances chronologiques entre C-EVENEMENTS | II-42 |

| | |
|--|-------|
| II-2.3.- Typologie des relations entre C-OPERATIONS situations de synchronisation de base | II-42 |
| II-2.3.1.- Typologie des situations de synchronisation : dépendances entre C-OPERATIONS | II-42 |
| II-2.3.2.- Graphe des dépendances entre C-OPERATIONS | II-49 |
| II-2.3.3.- Conclusion sur les dépendances entre C-OPERATIONS | II-52 |
| II-3.- Conclusion : prise en compte de la synchronisation par les relations de dépendance chronologique entre C-EVENEMENTS | II-53 |
| II-3.1.- La synchronisation est exprimée par les dépendances chronologiques entre C-EVENEMENTS | II-53 |
| II-3.1.1.- Satisfaction de la condition C2 | II-53 |
| II-3.1.2.- Satisfaction de la condition C3 | II-54 |
| II-3.2.- Exemple | II-57 |
| II-3.3.- Conclusion | II-61 |

CHAPITRE III : EVALUATION THEORIQUE DE LA MODELISATION CONCEPTUELLE

| | |
|---|--------|
| III-1.- Rappels sur les réseaux de Petri-Expression des conditions de validation d'une transition au moyen de fonctions booléennes | III-1 |
| III-1.1.- Rappels sur les réseaux de Petri | III-1 |
| III-1.1.1.- Réseaux de Petri | III-1 |
| III-1.1.2.- Principales propriétés définies sur les réseaux de Petri | III-4 |
| III-1.1.3.- Puissance de modélisation des réseaux de Petri : les réseaux de Petri à arcs inhibiteurs | III-5 |
| III-1.2.- Expression des conditions de validation d'une transition au moyen de fonctions booléennes | III-6 |
| III-1.3.- Conclusion | III-8 |
| III-2.- Puissance de modélisation du modèle conceptuel | III-8 |
| III-2.1.- Expression des processus associés aux transitions par le modèle conceptuel | III-8 |
| III-2.2.- Expression de la condition de sensibilisation d'une transition par le modèle conceptuel | III-12 |
| III-2.2.1.- Expression d'un impliquant premier | III-13 |
| III-2.2.2.- Expression de la réunion des impliquants premiers | III-16 |
| III-2.2.3.- Conclusion : expression de la condition de sensibilisation $S(t_i)$ | III-17 |
| III-2.3.- Expression d'un réseau de Petri à arcs inhibiteurs modélisant un système réel par le modèle conceptuel | III-19 |
| III-2.3.1.- Expression des problèmes de conflit entre transitions d'un réseau de Petri à arcs inhibiteurs par le modèle conceptuel | III-19 |
| III-2.3.1.1.- Caractérisation des conflits entre transitions d'un réseau de Petri à arcs inhibiteurs | III-19 |
| III-2.3.1.2.- Expression des situations de conflit entre transitions d'un réseau de Petri à arcs inhibiteurs par le modèle conceptuel | III-20 |

| | | |
|-------------|--|--------|
| III-2.3.2.- | Prise en compte de la notion de marquage d'une place par l'approche conceptuelle | III-23 |
| III-2.3.3.- | Conclusion : pouvoir de modélisation de l'approche conceptuelle | III-23 |
| III-3.- | Contrôles formels autorisés par la modélisation conceptuelle | III-24 |
| III-3.1.- | Sous-schéma dynamique vivant | III-24 |
| III-3.2.- | Détection des bouclages | III-26 |
| III-3.3.- | Détection des trappes | III-27 |

CHAPITRE IV : ANALYSE DE L'APPROCHE CONCEPTUELLE D'UN POINT DE VUE METHODIQUE

| | | |
|----------|--|-------|
| IV-1.- | L'approche conceptuelle : une analyse sémantique | IV-2 |
| IV-1.1.- | L'approche opératoire | IV-2 |
| IV-1.2.- | L'approche conceptuelle | IV-4 |
| IV-1.3.- | Commentaires | IV-5 |
| IV-1.4.- | Conclusions | IV-5 |
| IV-2.- | Simplicité de l'approche conceptuelle | IV-6 |
| IV-2.1.- | Le problème | IV-6 |
| IV-2.2.- | Application de l'approche opératoire | IV-7 |
| IV-2.3.- | Application de l'approche conceptuelle | IV-11 |
| IV-2.4.- | Commentaires | IV-13 |
| IV-2.5.- | Conclusions | IV-14 |
| IV-3.- | La rigueur de la d"marche conceptuelle permet une meilleure maîtrise des problèmes de conception | IV-16 |
| IV-3.1.- | Le problème | IV-16 |
| IV-3.2.- | L'approche opératoire | IV-17 |
| IV-3.3.- | L'approche conceptuelle | IV-19 |
| IV-3.4.- | Commentaires | IV-21 |
| IV-3.5.- | Conclusions | IV-21 |
| IV-4.- | La solution conceptuelle constitue une aide pour le chef de projet | IV-24 |
| IV-5.- | La spécificité des outils pour gérer la représentation conceptuelle | IV-26 |
| IV-6.- | Limitations de l'approche conceptuelle | IV-28 |
| IV-7.- | Conclusion | IV-28 |

CONCLUSION

BIBLIOGRAPHIE

INTRODUCTION

Ce travail est le résultat de la convergence de deux faisceaux d'idées :
- celui de la conception des systèmes d'information (SI) [39], [40], [86],
- et celui de la spécification d'outils pour la gestion de situation concurrentes (le Pilote d'aide à la conception de SI [85] - l'automate de la dynamique du SI [85]).

Le premier nous a conduit à raisonner dans le cadre d'un modèle abstrait en termes de représentation de phénomènes réels (notre modèle conceptuel [39], [40], [86]). Le second nous a naturellement amené à étudier et à tenter de comprendre, afin de pouvoir les utiliser, les méthodes et les outils de description et de gestion de la synchronisation.

Ces deux faisceaux convergent car on peut définir les situations de synchronisation à un niveau abstrait en les considérant comme des phénomènes interconnectés par des liens de causalité [86], [87], et donc utiliser le modèle conceptuel pour décrire des situations de synchronisation. Par ailleurs on peut associer à ce type de modélisation un certain type d'outil spécifique qui remplisse les mêmes fonctions de gestion de la synchronisation de la situation ainsi décrite au niveau conceptuel.

L'approche ainsi développée, que nous qualifions de conceptuelle, est donc équivalente dans ses objectifs aux approches habituelles des problèmes de synchronisation, que nous qualifions d'opératoires : elle permet la modélisation de situations de synchronisation, et peut déboucher sur des outils de gestion de cette modélisation. Néanmoins, elle est différente dans la manière d'aborder les problèmes de synchronisation et dans la façon de la gérer : l'originalité de notre approche réside

- 1) en la perception de la synchronisation en termes de phénomènes interconnectés par des liens de causalité, les interrelations temporelles de ces phénomènes étant la conséquence de ces liens de causalité.
- 2) en la mise en évidence de l'interconnexion entre un modèle de représentation et les outils qui l'exploitent.

Ce travail présente l'approche conceptuelle et en fait une évaluation par rapport aux approches des problèmes de synchronisation existantes. Il met en évidence les caractéristiques et les avantages de cette approche en la comparant, sur des situations identiques, aux approches habituelles de définition de la synchronisation.

Dans un premier chapitre, nous présentons notre point de vue sur les solutions actuelles aux problèmes de synchronisation. Nous tentons, à partir d'une brève analyse historique allant des mécanismes aux outils de modélisation, de :

- 1) dégager les caractéristiques d'un problème de synchronisation en posant des définitions qui servent de base à la suite de ce travail.
- 2) caractériser le type de méthode développée pour résoudre ces problèmes.

Dans un second chapitre, nous développons l'approche conceptuelle des problèmes de synchronisation. Nous montrons qu'elle comporte :

- 1) un modèle d'abstraction du comportement d'un système.
- 2) une méthode pour guider l'élaboration de cette abstraction.
- 3) un outil qui gère cette description abstraite, permettant ainsi le fonctionnement du système à partir de cette seule description abstraite.

Nous montrons enfin comment la modélisation proposée traduit la synchronisation au moyen des relations de causalité entre les différents concepts du modèle.

Dans le troisième chapitre, nous présentons une évaluation théorique de la modélisation conceptuelle, notamment du point de vue de la puissance de modélisation (par comparaison avec les réseaux de Petri à arcs inhibiteurs) et des contrôles formels qu'il autorise.

Enfin, dans le quatrième chapitre, nous analysons l'approche conceptuelle d'un point de vue méthodique : nous essayons d'évaluer les caractéristiques de la méthode conceptuelle d'approche d'un problème de synchronisation par rapport aux caractéristiques des méthodes habituelles d'analyse de ce type de problèmes.

CHAPITRE I

NOTRE POINT DE VUE SUR LES SOLUTIONS

ACTUELLES AUX PROBLÈMES DE SYNCHRONISATION

Ce chapitre présente les conclusions de notre réflexion sur les problèmes de synchronisation menée à partir de l'analyse des solutions qui ont été chronologiquement proposées au problème de la synchronisation. Il a pour but :

- de caractériser la notion de problème de synchronisation et de poser les définitions qui nous serviront de référence dans la suite de ce travail.
- de caractériser les méthodes habituelles d'analyse et de description des situations de synchronisation.

I-1.- HYPOTHESES DE RAISONNEMENT

• Dans toute la suite de ce travail, nous ne nous intéresserons pas à la synchronisation des systèmes physiques, mais à celle des systèmes artificiels, les systèmes qui gèrent des représentations sous forme d'informations c'est-à-dire à la synchronisation des systèmes informatiques. Par là, nous entendons ne pas nous préoccuper des mécanismes assurant la synchronisation physique au sein des calculateurs, mais celle concernant les "actions" gérées par les systèmes informatiques. Celle-ci a pour but de coordonner ces actions afin que le système qui les gère remplisse de manière cohérente les fonctions pour lesquelles il a été conçu.

• Ces systèmes mettent en jeu :

- des processus (qui, dans un certain contexte, peuvent être des programmes).
- des ressources (dont un sous-ensemble constitue les données des processus).
- des commandes d'exécution des processus (qu'elles soient manuelles ou automatisées) dont l'ordonnement définit le comportement du système.

Nous allons préciser ces trois notions à travers un bref historique des problèmes de synchronisation.

I-2.- HISTORIQUE DES APPROCHES DE SYNCHRONISATION : DU MECANISME A LA MODELISATION

I-2.1.- Premier stade : les mécanismes

Le jour où l'on a voulu accroître la rapidité de prise en compte des programmes par la machine, on a remplacé l'opérateur qui les activait depuis un clavier par un automate chargé de lancer leur exécution : ce sont les premières versions des moniteurs d'enchaînement [13] qui ont conduit ultérieurement aux systèmes d'exploitation. On peut constater qu'il s'agissait alors de trouver une solution pratique (un mécanisme) à un problème constaté. Néanmoins ces premiers types de mécanismes, n'autorisant la prise en compte d'un nouveau programme qu'après l'exécution complète du programme précédent ne posaient pas réellement de problème de synchronisation.

Les premiers problèmes de synchronisation ont été rencontrés lors de la conception des systèmes d'exploitation à partir du moment où, pour des raisons d'efficacité, l'on a autorisé la prise en compte d'un programme avant que le programme précédent ont été complètement exécuté. Ils ont donc été perçus en termes d'enchaînement et d'interaction entre les programmes. Dans un premier stade, les solutions apportées à ces problèmes de synchronisation ont donc été des mécanismes conçus pour éviter les erreurs et les incohérences qui pourraient résulter de l'exécution et de l'enchaînement des processus (notamment au niveau du partage des ressources et du parallélisme). On peut considérer que les sémaphores [25], [33], [34], les moniteurs [46], [47], [56], les coroutines [57], [21], les événements de PL/1 [94] et l'attente conditionnelle [93] figurent parmi les plus sophistiquées des solutions aux problèmes de synchronisation posés dans ces termes.

Dans ce type d'approche [25],

- un processus est une suite temporelle d'exécutions d'instructions, correspondant à l'exécution d'un programme. C'est une entité dynamique, créée à un instant donné qui disparaît au bout d'un temps fini, et qui représente une activité que l'on veut considérer comme élémentaire.
- Pour qu'un processus puisse évoluer, il a besoin de procédures et de données de mémoire destinées à les contenir, de l'unité centrale, éventuellement de fichiers et de périphériques. Toutes ces entités seront appelées des ressources. Une ressource est donc une entité nécessaire à l'exécution d'un programme.
- De plus, les divers processus d'un système n'évoluent généralement pas indépendamment les uns des autres. Il existe entre eux des relations qui dépendent de la logique de la tâche à accomplir et qui fixent leur déroulement dans le temps. L'ensemble de ces relations recouvre alors la notion de synchronisation (bien qu'elles ne fassent pas intervenir le temps comme mesure de durée, mais seulement comme moyen d'introduire une relation d'ordre entre des instructions exécutées par un processus).
- En conséquence, le problème de la synchronisation consiste à construire un mécanisme indépendant des vitesses permettant à un processus P_i actif
 - soit d'en bloquer un autre ou de se bloquer lui-même en attendant un signal d'un autre processus.
 - soit d'activer un autre processus P_j en lui transmettant éventuellement de l'information.

On peut ainsi remarquer que ces définitions sont très liées à l'aspect "mécanisme technique" qui caractérise ce type d'approche des problèmes de synchronisation

I-2.2.- Deuxième stade : la modélisation

Dans un deuxième stade, les limitations des mécanismes de résolution de problèmes de synchronisation, principalement leur trop grande dépendance vis-à-vis de solutions techniques d'implémentation nécessitant la description exhaustive de tous les choix physiques concernant la communication et le partage de ressources par exemple, ont conduit à tenter de faire abstraction de ces mécanismes en modélisant directement les situations de synchronisation entre processus (laissant au concepteur la liberté d'associer à cette modélisation l'une ou plusieurs de ces solutions techniques maîtrisées).

Ceci a conduit à une réflexion sur les raisons pour lesquelles il fallait faire appel à un mécanisme du type de ceux évoqués ci-dessus, et l'on a été conduit à s'interroger sur la nature et les causes des problèmes de synchronisation et de la synchronisation elle-même. Ainsi, des définitions de la synchronisation et des problèmes de synchronisation ont été proposées, toutes basées sur cette constatation : "les processus d'un système informatique ne sont pas indépendants, il existe entre eux des interactions temporelles". Donc la synchronisation a été perçue comme l'interaction temporelle des processus.

Dans ce type d'approche

- un processus est un ensemble fini d'opérations dans lequel les opérations sont exécutées de manière telle qu'à un instant donné une et une seule soit exécutée [13]. Une opération est définie comme une règle définissant un ensemble fini de données appelé Sortie à partir d'un ensemble fini de données appelé Entrée. Une fois initialisée, une opération est terminée au bout d'un temps fini. Une opération conduit toujours au même ensemble de Sortie lorsqu'elle est appliquée à un ensemble d'Entrée donné, indépendamment du temps mis pour le produire [13]. De plus, un processus présente deux aspects [26] : il véhicule de l'information et il exécute des actions.
- La notion de ressource est contenue dans celle de processus [26] : une ressource appartient à une classe de ressources utilisée par un certain nombre de processus. Elle correspond à la notion d'Entrée, de Sortie ou de donnée [13] d'une opération (participant à la définition d'un processus).
- Alors, la synchronisation est un terme général pour n'importe quelle contrainte concernant l'ordre dans lequel les opérations seront exécutées [13] et
- le problème de la synchronisation consiste à définir toutes les contraintes régissant l'ordre d'exécution des opérations [13].

On s'aperçoit que ces définitions sont beaucoup plus dégagées des contraintes techniques que celles proposées dans le paragraphe I.2.1.

Enfin, une telle réflexion, conduisant à la perception de la synchronisation en termes d'interaction temporelle des processus a permis d'élaborer une typologie des relations existant entre deux opérations OP_i et OP_j [13] :

- 1) OP_j suit toujours OP_i (chronologie)
- 2) OP_j peut suivre OP_i (conditionnement)
- 3) OP_j est indépendante de OP_i (indépendance)
- 4) OP_j et OP_i sont en exclusion mutuelle (exclusion mutuelle)
- 5) OP_j est prioritaire par rapport à OP_i (priorité)

Cette typologie, traduisant les situations de synchronisation de base entre deux opérations, permet de reconstruire toute situation de synchronisation par composition de ces cinq relations de base [13].

Parallèlement à cette réflexion conduisant à faire abstraction des mécanismes techniques, on a été amené à inventer des moyens permettant de décrire les interactions temporelles entre les processus (c'est-à-dire la synchronisation). De plus, ces moyens de description de la synchronisation devaient être associés à des outils de gestion de la synchronisation tels que ceux évoqués au paragraphe I.2.1., mais qui soient transparents aux concepteurs. C'est-à-dire que, à partir de la description des interactions temporelles entre les processus, définie par le concepteur à l'aide d'un de ces moyens, une solution de gestion de la synchronisation est générée sans aucune autre intervention de ce concepteur. Cette solution est basée sur les outils associés à ce moyen de description de la synchronisation.

Les moyens de description des interactions temporelles peuvent être classés en deux principaux groupes :

1) La description de la synchronisation par un langage

Dans cette approche, les règles définissant les interactions temporelles sont exprimées par le biais d'un langage directement accepté par la machine. L'outil de gestion de la synchronisation ainsi défini est alors sous-jacent au langage. Ainsi par exemple, dans le langage SIMONE [11], [52], les moniteurs de Hoare [48] constitue l'outil de gestion de la synchronisation sous-jacent. Les solutions sont de même nature dans SIMULA [26], [27], MODULA [96], PASCAL PARALLELE [14], ALGOL 68 [3] pour le calcul collatéral, CLU [68] basé sur les types abstraits [67] et actuellement des LANGAGES de HAUT NIVEAU [81], [84], et actuellement le développement des COMMANDES GARDEES [36], des PROCESSUS COMMUNICANTS [48] et des PROCESSUS DISTRI-BUEES [15].

2) La description de la synchronisation par un modèle

Dans cette approche, les règles définissant les interactions temporelles entre les processus sont exprimées dans le formalisme d'un modèle généralement non directement accepté par la machine. Pour que cette description donne naissance à une solution technique, l'élaboration d'algorithmes de traduction d'une situation de synchronisation exprimée dans ce modèle en une expression acceptable par la machine, utilisant les mécanismes de gestion de la synchronisation présentés en I.2.1. s'avère nécessaire. Un des principaux avantages de cette approche de description est que plusieurs mécanismes de gestion de la synchronisation peuvent être associés à une même modélisation (selon le choix de l'algorithme de traduction). Parmi les modèles existants, nous pouvons citer entre autres les RESEAUX DE PETRI [79], [80], [77], [30], [49], [50] et leurs extensions telles que les RESEAUX de PETRI à ARCS INHIBITEURS [1], [2], les RESEAUX de PETRI GENERALISES [43], [54], les RESEAUX de PETRI INTERPRETES [22], [92], les RESEAUX de PETRI TEMPORISES [83], [70] ainsi que les modèles qui en résultent, tels que les RESEAUX de CONTROLE DE PROCESSUS PARALLELES [71], [92], les EXPRESSIONS DE CHEMINS [42], [16], les MOTS DE SYNCHRONISATION [88].

On peut de plus remarquer que la discrimination entre les approches de description basées sur les langages et celles basées sur les modèles repose en fait sur une apparente différence liée à l'aspect utilisation par la machine de ces descriptions. En effet, ou bien la description des interactions temporelles entre les processus est effectuée dans un formalisme directement exploitable par la machine (approche du type "langage"), ou bien une traduction, généralement automatique, en est nécessaire pour la rendre exploitable par une machine (aspect "modèle").

Or cette dichotomie n'est que superficielle et ces deux solutions sont fondamentalement équivalentes. En effet, comme nous l'avons vu ci-dessus, un modèle, pour être exploitable par un calculateur, doit être décrit sous forme d'un langage d'utilisation. Par exemple, les RESEAUX de CONTROLE de PROCESSUS PARALLELES [71], [92] évoqués plus haut doivent être décrits au moyen du langage MAS [71], [97] pour que la machine puisse fournir automatiquement une solution implementée à la situation de synchronisation décrite par le Réseau. D'autre part, un langage de description (ou de simulation) s'appuie toujours sur un modèle qui lui est sous-jacent. Ainsi par exemple, le langage CLU [68] s'appuie sur la notion de TYPE ABSTRAIT [67]. Comme en général le modèle associé au langage n'est pas explicité, la syntaxe peut parfois masquer les vrais concepts qui aident à raisonner.

I-2.3.- Conclusion

Il apparaît donc que la réflexion sur les problèmes de synchronisation et sur la synchronisation elle-même a tout d'abord été menée dans le cadre de l'analyse et la conception des systèmes opératoires, puis en tentant de généraliser les modélisations ainsi mises au point à la prise en compte de la synchronisation dans les systèmes artificiels en général.

Les RESEAUX de PETRI constituent un excellent exemple de modélisation ayant suivi une évolution de ce type. En effet, conçus tout d'abord comme une modélisation de la communication entre automates [79], ils sont maintenant utilisés dans des domaines aussi variés que l'étude des langages formels [44], [45], [77], [24] et l'écriture des compilateurs [5], jusqu'à la conception des systèmes d'information [64], [49], [50] en passant par l'analyse et la conception de hardware [89], [29] ou de software [31], [62], [73].

I-3.- ESSAI DE SYNTHÈSE DES APPROCHES EXISTANTES

Afin de pouvoir s'appuyer sur des notions non ambiguës dans la suite de ce travail, nous allons définir les notions de processus, ressource, synchronisation et problème de synchronisation que nous utiliserons. Ces définitions sont le résultat de l'analyse précédente.

I-3.1.- Nos propositions de définition

Comme toutes les définitions d'un processus présentées ci-dessus contiennent à un degré plus ou moins important l'idée de transformation sur un ensemble de "données", nous définissons un processus par :

Définition d'un processus :

- Un processus p est une transformation τ_p d'un ensemble de données D_p en un ensemble de sorties S_p :

$$p : D_p \xrightarrow{\tau_p} S_p$$

Nous définissons alors la notion de ressource par rapport à la notion de processus comme il est courant de le faire :

Définition des ressources :

- Les éléments des ensembles D_p et S_p , ainsi que les constituants du système abstrait nécessaires à l'exécution de la transformation τ_p sont les ressources liées au processus p (c'est-à-dire les ressources qu'il "utilise" et qu'il "produit").

Définitions : synchronisation - problème de synchronisation

- Le problème de la synchronisation consiste à définir les règles de coopération temporelle entre les processus garantissant le maintien de la cohérence du système abstrait au cours de son évolution dans le temps. Cette définition du problème de la synchronisation est proche de celle proposée par [13].

- Synchroniser l'ensemble des processus coopérants du système revient donc à définir un ordre sur l'exécution de ces processus, conformément à leurs relations d'interdépendance et d'interaction, afin que la cohérence du système abstrait soit garantie. Nous désignerons donc par synchronisation l'ensemble des relations temporelles entre les processus du système abstrait définissant cet ordonnancement (notons que ces relations sont dues soit à la logique même du système, soit au partage de ressources entre ces processus).

On peut d'ailleurs remarquer que dans cette définition, assez voisine de [13], on retrouve l'idée que la synchronisation est liée à une notion d'ordonnement sur l'exécution des processus, idée commune à bon nombre des définitions habituelles de la synchronisation.

Il ressort donc que, pour résoudre un problème de synchronisation, il faut satisfaire les trois conditions de caractérisation suivantes :

- Définition : conditions de la prise en compte de la synchronisation :

Condition 1 : définir précisément l'ensemble des processus du système abstrait, et par conséquent l'ensemble des ressources du système (puisque la notion de ressource est définie par rapport à celle de processus).

Condition 2 : définir les interrelations entre ces processus, c'est-à-dire leur dépendance ou leur indépendance.

Condition 3 : définir leurs règles de coopération temporelle, en fonction des règles définissant la logique du problème envisagé, c'est-à-dire définir un ordre sur l'exécution des processus respectant ces règles, garantissant ainsi le maintien de la cohérence du système abstrait.

Nous nous baserons donc sur ces trois conditions pour analyser la prise en compte de la synchronisation par une approche de modélisation.

I-3.2.- La notion d'approche opératoire

Comme nous l'avons montré, la réflexion sur la synchronisation a tout d'abord été menée dans le cadre de l'analyse et la conception des systèmes opératoires, puis en tentant de généraliser les modélisations ainsi mises au point à la prise en

compte de la synchronisation dans tous les systèmes artificiels en général. C'est pourquoi nous regrouperons dans la suite de ce travail les approches habituelles de la synchronisation sous le vocable d'"approches opératoires".

Comme dans la suite de ce travail, nous ne nous intéresserons qu'aux approches opératoires basées sur la modélisation des situations de synchronisation (voir paragraphe I-2.2.), nous allons nous attacher à qualifier la méthode d'approche des problèmes de synchronisation qui les sous-tend. Pour ce faire nous allons tenter de présenter la méthode type, malgré les difficultés dues à la grande diversité des approches de ce type et des méthodes qui leur sont associées, en essayant de dégager les grandes caractéristiques communes à ces méthodes.

Il nous semble, tout d'abord, que les approches opératoires de modélisation sont de type fonctionnel, c'est-à-dire qu'elles abordent le problème de la modélisation d'un système par l'étude des fonctions dont il doit assurer la réalisation. Chacune de ces fonctions est traduite par un processus, ou un enchaînement de processus, suivant le niveau de détail atteint lors de la définition des fonctions.

Il apparaîtrait que, quel que puisse être le type de modélisation utilisé, la solution aux problèmes de synchronisation proposée par ces approches consiste à définir les situations de synchronisation entre les processus réalisant les fonctions du système abstrait en déterminant quels sont les processus qui contribuent à créer un certain niveau de ressources nécessaire pour déclencher un ou plusieurs autres processus. Pour un état donné d'un processus, cette solution conduit à faire un inventaire aussi exhaustif que possible de toutes les situations qui permettent à un instant t de déclencher un ou plusieurs processus. La méthode d'analyse d'un problème de synchronisation proposée par ces approches repose sur la notion d'état d'un processus (point privilégié d'observation de ce processus) : on cherche pour chacun des processus du système, et pour chaque état de ce processus, à faire l'inventaire de toutes les situations qui permettent de déclencher un ou plusieurs autres processus.

Il nous semble que, dans ces approches, les problèmes de synchronisation sont résolus en partant de la troisième étape (définir les règles de coopération temporelle entre les processus, voir paragraphe I-3.1. ci-dessus) et en essayant de trouver une réponse aux problèmes posés par la seconde étape (définir la dépendance ou l'indépendance entre les processus) et la première étape (définir précisément l'ensemble des processus du système). On tente de trouver ces réponses par un raisonnement sur l'état global des ressources, à l'aide de considérations heuristiques spécifiques d'une classe de problèmes à laquelle se rattache le problème de synchronisation que l'on traite.

On constate par ailleurs que ces méthodes sont assorties d'outils de contrôle visant à s'assurer de l'adéquation des solutions auxquelles elles conduisent. Ces contrôles semblent pouvoir être regroupés en deux grandes catégories :

- les contrôles sur l'exhaustivité de l'inventaire des situations de synchronisation et des processus qui les composent.
- les contrôles sur l'ensemble des relations temporelles exprimant les règles de coopération entre les processus, à savoir les contrôles de cohérence (garantissant la cohérence de ces règles de coopération) et les contrôles de complétude et de fidélité (assurant que la solution fournie décrit bien le système réel que l'on voulait modéliser, c'est-à-dire assurant que sa description est complète).

Dans l'état actuel des travaux relatifs à ce domaine, il semblerait que le seul moyen de s'assurer de l'exhaustivité de l'inventaire des situations de synchronisation et des processus qui les composent, de même que le seul moyen de s'assurer de la complétude et de la fidélité de la description des relations temporelles entre les processus consiste en une simulation de la solution conduisant à effectuer des mesures sur une maquette et en une comparaison de ces mesures avec celles effectuées sur le système réel. Par contre, il existe des méthodes formelles prouvant la cohérence des relations temporelles qui définissent la coopération des processus, telles que l'analyse au moyen d'un schéma de programmes parallèles [53] ou d'un système d'addition de vecteurs [51] ou, dans le cas des réseaux de Petri au moyen d'équations matricielles [72].

En conclusion, alors que les contrôles formels de cohérence des relations temporelles définissant la coopération des processus apparaissent comme très complets, on peut se poser des questions sur la suffisance des contrôles de complétude et de fidélité de ces solutions.

CHAPITRE II

L'APPROCHE CONCEPTUELLE DE LA

SYNCHRONISATION

L'approche conceptuelle comporte :

- un modèle pour conceptualiser des situations dynamiques ou, plus généralement, le fonctionnement d'un système abstrait.
- une méthode qui constitue un guide pour l'élaboration de cette conceptualisation.
- un automate qui décrit les mécanismes de gestion du système abstrait en fonctionnement.

Ce chapitre décrit les différents éléments de l'approche conceptuelle, montre comment la modélisation proposée traduit la synchronisation et comment l'automate présenté en traduit la gestion.

II-1.- LES ELEMENTS DE L'APPROCHE CONCEPTUELLE

II-1.1.- Le modèle conceptuel

II-1.1.1.- Que doit-il représenter : analyse du réel en termes de catégories

Le modèle a été défini à partir d'une analyse des systèmes réels consistant à mettre en évidence les catégories de phénomènes et les associations [40] qui les caractérisent. Cette analyse nous a permis de conclure que :

- 1) Tout phénomène réel à représenter appartient à une des trois catégories :

OBJETS, EVENEMENTS, OPERATIONS

De plus, tout phénomène est caractérisé et décrit par ses propriétés.

- 2) La dynamique du système réel est complètement représentée par trois catégories d'associations [] entre les trois catégories de phénomènes :

MODIFIE (opération , objets)

CONSTATE (événement, objets)

DECLENCHE (événement, opérations)

Il résulte de cette analyse que :

- 1) - un OBJET est un constituant du système réel durable, concret ou abstrait, qui peut être particularisé [7], [17], [55], [65]
exemple : le client "DUGOMMIER", le produit n° 22
- une OPERATION est une action qui peut être exécutée seule, à un moment donné dans le système réel, et qui provoque le changement d'état de un ou plusieurs objets.

exemple : l'opération "analyse de commande" numéro 317 crée l'objet "commande acceptée" n° 117.

- un EVENEMENT est tout ce qui peut arriver à un moment donné. C'est la constatation d'un changement d'état d'un ou de plusieurs objets entraînant l'exécution d'opérations.

exemple : l'événement "arrivée de la commande" n° 316 déclenche l'opération "analyse de commande" n° 135.

Remarquons qu'un changement d'état est différent d'un état. L'état d'un objet peut être durable, alors qu'un changement d'état d'un objet est instantané.

Remarquons également qu'un événement est différent d'un changement d'état : tous les changements d'état d'un objet ne sont pas des événements. Par exemple, beaucoup de modifications peuvent affecter le stock d'un produit, générant ainsi une multitude de changements d'état du stock, mais seuls quelques uns de ces changements d'état sont des événements entraînant l'émission de bons de réapprovisionnement.

2) La dynamique du système réel est entièrement représentée par les trois catégories d'associations entre objets, événements et opérations suivantes :

MODIFIE, CONSTATE, DECLENCHE

- L'association MODIFIE (opération, objets) relie une opération et un ou plusieurs objets. Elle exprime que l'opération modifie les objets.
- L'association CONSTATE (événement, objets) relie un événement et un ou plusieurs objets. Elle exprime que les changements d'état des objets sont des événements.
- L'association DECLENCHE (événement, opérations) relie un événement et une ou plusieurs opérations. Elle exprime qu'un événement déclenche une ou plusieurs opérations.

Exemples :

- l'opération "analyse de commande" n° 317 modifie l'objet "commande acceptée" n° 117 puisqu'elle le fait passer de l'état existant à l'état inexistant.
- l'événement "arrivée de la commande" n° 316 constate le changement d'état de l'objet "commande" n° 316 qui passe de l'état non arrivé à l'état arrivé.
- l'événement "arrivée de la commande" n° 316 déclenche l'opération "analyse de commande" n° 135.

Nous pouvons résumer ce que nous venons de présenter par le schéma suivant, illustrant les trois catégories de phénomènes et leurs associations :

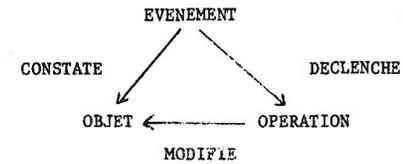


Figure II-1 : les catégories de phénomènes réels et leurs associations

La définition de la dynamique que nous proposons est de type causal : les événements déclenchent l'exécution des opérations dont les effets sont les changements d'état d'objets qui peuvent être, à leur tour, des événements.

Remarques :

- Les phénomènes d'une même catégorie appartiennent à des classes. Dans une même classe, tous les phénomènes sont décrits par la même collection de propriétés.
exemple : la classe "client", la classe "arrivée de commande".
- Les associations entre phénomènes appartiennent aussi à des classes. Une classe d'associations représentant la dynamique est définie entre associations de phénomènes.
exemple : la classe "déclenchement de l'analyse des commandes" est une association entre la classe d'événements "arrivée de commande" et la classe d'opérations "analyse de commande".

II-1.1.2.- Comment représenter : le modèle conceptuel

Ce modèle a été l'objet de plusieurs présentations [40], [85], [86], [87]. Nous ne rappellerons donc que les résultats nécessaires à la compréhension de la suite de ce travail.

Le modèle conceptuel doit satisfaire deux principales exigences :

- être formel
- représenter aisément et de manière homogène les classes de phénomènes et d'associations précédemment définies.

Pour satisfaire ces exigences, nous avons été conduits à :

- choisir un modèle de type relationnel [18], [20]
- inclure le temps dans le modèle
- introduire des types sur les relations afin de représenter les différentes catégories
- rendre le modèle normatif, c'est-à-dire imposer une forme normalisée de relation pour chacun des types de relation (ce qui conduit à la minimalité du modèle).

La correspondance entre la réalité et la représentation conceptuelle réside dans le fait que :

- chaque classe de phénomènes est représentée par une ou plusieurs relations
- chaque propriété d'une classe de phénomènes est représentée par un attribut de relation
- la catégorie d'une classe de phénomènes est représentée par son type de relation.

Ainsi, il y a trois types de relations, décrivant les trois concepts du modèle et que nous appelons C-OBJET, C-EVENEMENT et C-OPERATION. Ces trois types de relations permettent de représenter les différentes classes de phénomènes :

- les classes d'objets sont représentées par le concept de C-OBJET
- les classes d'opérations et d'associations entre classes d'opérations et classes d'objets (classes d'associations MODIFIE) sont représentées par le concept de C-OPERATION
- les classes d'événements et d'associations entre classes d'événements et classes d'objets (classes d'associations CONSTATE) d'une part, entre classes d'événements et classes d'opérations (classes d'associations DECLENCHE) d'autre part, sont représentées par le concept de C-EVENEMENT.

Ceci peut être représenté par la figure suivante :

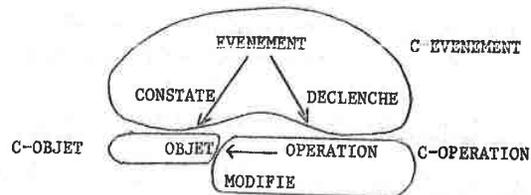


Figure II-1

Figure II-2 : Correspondance entre classes de phénomènes, associations et concepts

Nous nous limiterons à présenter sommairement ces concepts sous leur forme normalisée : (On trouvera une présentation formelle théorique de ces concepts dans [39]).

II-1.1.2.1.- Le concept de C-OBJET

Définition : Une relation du type C-OBJET est une relation permanente, c'est-à-dire une relation en troisième forme normale [10] dont tous les constituants sont en dépendance fonctionnelle permanente avec l'identifiant-clé. Rappelons que la dépendance fonctionnelle entre deux constituants A et B d'une relation est permanente si et seulement si A et B ont les mêmes dates de changement de valeur (autrement dit si pour toute occurrence a de A, l'occurrence b de B correspondante a la même durée de vie que a). Cette dépendance fonctionnelle permanente, notée $A \xrightarrow{\cdot} B$, traduit le fait que les propriétés correspondant à ces deux constituants A et B ont le même comportement dynamique [59].

Sémantique : Le C-OBJET représente un aspect dynamique d'une classe de phénomènes réels de la catégorie objet. En d'autres termes, il représente la plus grande collection de propriétés d'un objet ayant un comportement dynamique identique, c'est-à-dire ayant ses propriétés créées, modifiées ou supprimées en même temps : un C-OBJET représente l'évolution au cours du temps d'une classe d'objets du monde réel, vue sous un certain aspect. Un objet du système est alors la réalisation d'une relation du type C-OBJET.

Le C-OBJET représente un état élémentaire du système réel.

Exemple :

Considérons la classe d'objets réels "CLIENT", caractérisée par les propriétés numéro d'INSEE (INSEE), nom (NOM), adresse (ADRESSE), chiffre d'affaires (CHIFFAFF).

La description de cette classe d'objets par l'intermédiaire d'une relation en troisième forme normale serait la suivante :

CLIENT (NCLI, INSEE, NOM, ADRESSE, CHIFFAFF)

La description de cette classe en termes de c-objets conduit aux quatre relations suivantes :

CLIENT (NCLI, INSEE)
 CLIENT-NOM (NCLI, DATE-N, NOM)
 CLIENT-ADR (NCLI, DATE-ADR, ADRESSE)
 CLIENT-CA (NCLI, DATE-CA, CHIFFAFF)

En effet, les quatre propriétés de la classe CLIENT ont un comportement dynamique différent :

- le numéro INSEE est constant durant toute la vie du client
- le nom du client peut changer une fois dans sa vie (si c'est une femme)
- l'adresse et le chiffre d'affaires sont modifiés à des dates différentes.

Chacune de ces quatre relations représente un aspect de la classe CLIENT. Par exemple, les n-uplets de la relation CLIENT-ADR permettent d'obtenir les adresses successives d'un client au cours du temps. Ceci fait apparaître la notion de C-CLASSE [59] : une C-CLASSE est l'ensemble de tous les C-OBJETS décrivant une même classe d'objets réels.

II-1.1.2.2.- Le concept de C-OPERATION

Définition : Une relation de type C-OPERATION est une relation permanente. La normalisation des c-opérations conduit à satisfaire les trois contraintes suivantes :

- (1) C-OP \rightarrow C-OB
- (2) C-OP \rightarrow TYPE-CHANGE
- (3) C-OP \rightarrow TEXT-OP

Remarque : Nous exprimons ces contraintes à l'aide du formalisme des dépendances fonctionnelles [10].

C-OP est l'identifiant d'une c-opération

C-OB est l'identifiant d'un c-objet

TYPE-CHANGE est le type de changement d'état qu'entraîne l'opération (création, suppression ou mise-à-jour)

TEXT-OP désigne le texte de la c-opération

Sémantique : Une C-OPERATION est la représentation d'un aspect d'une classe d'opérations réelles qui modifient chacune de la même façon un et un seul objet appartenant au même C-OBJET et qui correspond à l'application de la même règle de gestion au cours du temps. Une opération est alors une réalisation d'une relation du type C-OPERATION. Plus précisément, les deux premières contraintes expriment que les occurrences (les réalisations) d'une C-OPERATION représentent les opérations qui modifient, de la même façon, l'état des objets correspondant à un unique C-OBJET.

Par abus de langage, nous dirons qu'une C-OPERATION modifie de manière unique l'état d'un et d'un seul C-OBJET.

Exemple :

La C-OPERATION "insertion d'un nouveau client" crée (type de changement d'état : contrainte 2) un C-OBJET "client" supplémentaire (contrainte 1) (plus exactement une occurrence de la C-OPERATION "insertion d'un nouveau client" crée une réalisation supplémentaire du C-OBJET "client").

La troisième contrainte exprime qu'une C-OPERATION représente une règle de gestion du système réel.

En résumé : La C-OPERATION représente une action élémentaire qui provoque un changement d'état élémentaire dans le système réel : elle représente la plus petite transformation du système réel.

II-1.1.2.3.- Le concept de C-EVENEMENT

Définition : Une relation du type C-EVENEMENT est une relation permanente. Par ailleurs la normalisation des c-événements impose les quatre contraintes d'intégrité suivantes :

- (1) C-EV \rightarrow C-OB
- (2) C-EV \rightarrow TYPE-CHANGE
- (3) C-EV \rightarrow P-INIT, P-FIN
- (4) C-EV \rightarrow C-OP

Remarque : Nous exprimons ces contraintes à l'aide des relations de dépendances fonctionnelles (\rightarrow) [19] et des relations de dépendances fonctionnelles multi-valuées ($\rightarrow\rightarrow$) [37].

C-EV est l'identifiant d'un C-EVENEMENT

C-OB est l'identifiant d'un C-OBJET
 C-OP est l'identifiant d'une C-OPERATION
 TYPE-CHANGE est un type de changement d'état d'un C-OBJET
 P-INIT, P-FIN sont les prédicats (initial et final), définissant
 le changement d'état d'un C-OBJET.

Sémantique : Le C-EVENEMENT est la représentation d'une classe d'événements réels correspondant tous à des changements d'état des objets d'un même C-OBJET et déclenchant tous des opérations appartenant à une ou plusieurs C-OPERATIONS. Nous avons défini le C-EVENEMENT par référence au concept de C-OBJET. Le C-EVENEMENT est l'expression du plus petit type de changement d'état remarquable d'un C-OBJET qui déclenche une ou plusieurs C-OPERATIONS. Un événement est alors une réalisation d'une relation du type C-EVENEMENT.

Plus précisément, les deux premières contraintes expriment le fait qu'un C-EVENEMENT représente une classe d'événements qui constatent un seul type de changement d'état d'objets correspondant à un même C-OBJET. Par abus de langage, nous dirons qu'un C-EVENEMENT est la constatation d'un type de changement d'état remarquable d'un et d'un seul C-OBJET.

Exemple :

Le C-EVENEMENT "arrivée de commande" est associé à la création (type de changement d'état : contrainte 2) du C-OBJET "commande" (contrainte 1).

La troisième contrainte exprime que le changement d'état est défini par l'état initial et l'état final de l'objet, décrits par deux prédicats (qui peuvent évoluer dans le temps).

Exemple :

Le C-EVENEMENT "rupture de stock" est défini par les deux prédicats :

P-INIT := stock quelconque (:= VRAI)

P-FIN := stock \leq seuil

La dernière contrainte exprime qu'un événement appartenant à un C-EVENEMENT donné déclenche des opérations appartenant à une ou plusieurs C-OPERATIONS déterminées. Ce déclenchement peut être conditionnel ou/et répétitif. Par abus de langage : un C-EVENEMENT déclenche une ou plusieurs C-OPERATIONS.

Exemples :

Chaque C-EVENEMENT "arrivée de commande" déclenche les C-OPERATIONS "analyse de commande" et "mise-à-jour de l'archivage". (Plus précisément, chaque occurrence du C-EVENEMENT "arrivée de commande" déclenche une réalisation des C-OPERATIONS "analyse de commande" et "mise à jour de l'archivage").

Chaque C-EVENEMENT "rupture de stock" déclenche la C-OPERATION "émission d'un bon de réapprovisionnement" si et seulement si "la trésorerie est suffisante". (c'est-à-dire que chaque occurrence du C-EVENEMENT "rupture de stock" déclenche une réalisation de la C-OPERATION "émission d'un bon de réapprovisionnement" si et seulement si "la trésorerie est suffisante").

En résumé : Un C-EVENEMENT représente un changement d'état élémentaire remarquable du système réel, déclenchant des actions élémentaires : il représente un changement d'état remarquable élémentaire du système réel.

Remarque : Il convient de noter que le déclenchement d'une C-OPERATION par un C-EVENEMENT peut être conditionnel : par exemple, le C-EVENEMENT "rupture de stock" ne déclenche la C-OPERATION "émission d'un bon de réapprovisionnement" que si la condition "la trésorerie est suffisante" est vérifiée.

II-1.2.- Le résultat de la modélisation

II-1.2.1.- La structure conceptuelle

La structure conceptuelle est une collection de relations représentant la dynamique du système réel comme un ensemble de faits qui déclenchent les transformations des objets réels et leurs connexions, augmentée d'un ensemble de contraintes d'intégrité [12], [9] complétant la définition du système, notamment aux frontières de la représentation. Ces contraintes ont principalement pour but de préciser la sémantique des objets, opérations et événements du système réel, et notamment d'éliminer toute sorte de conflit sur l'entrée des objets externes [6]. Par exemple, un objet du type "ligne de commande" ne peut être entré avant l'objet du type "commande" correspondant.

La structure conceptuelle est une représentation d'un système réel indépendante de l'usage et des aspects techniques d'implémentation de ce système [40], [86], [87]. C'est une représentation minimale (au sens de couverture minimale d'une collection de relations [28]) du système réel notamment à cause de l'élémentarité

des concepts [39], [40]. Au modèle relationnel nous avons associé un langage de description qui permet d'exprimer et de décrire la structure conceptuelle, c'est-à-dire la collection des relations ainsi que les contraintes d'intégrité qui s'y rapportent [39], [63].

Elle peut être perçue comme deux sous-structures complémentaires ayant leur sémantique propre, et dont la réunion constitue le schéma conceptuel :

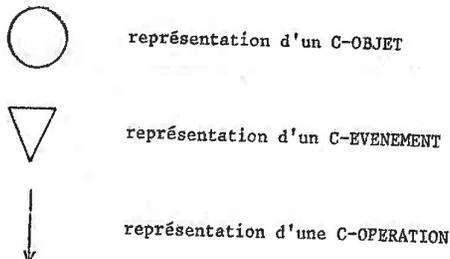
1) le schéma de données :

Il correspond à la structure conceptuelle des données, exprimée par la système de C-CLASSES d'objets. Ce système représente la structure des objets du système réel ainsi que tous leurs états possibles. Il correspond à une vue statique du système réel représenté. Ce sont les occurrences associées à ce schéma qui définissent les états du système réel représenté.

2) le schéma de la dynamique :

Définissant la structure conceptuelle de la dynamique par les interrelations entre C-OBJETS, C-OPERATIONS et C-EVENEMENTS, il représente la dynamique du système réel par le treillis des relations de causalité qu'ont entre elles les catégories de phénomènes qui la composent.

Nous utiliserons dans la suite de ce travail la représentation graphique du schéma de la dynamique pour des raisons de commodité, les conventions graphiques utilisées étant les suivantes :



De plus, la concaténation de ces symboles représentant les concepts du modèle permet de représenter les associations entre ces concepts :

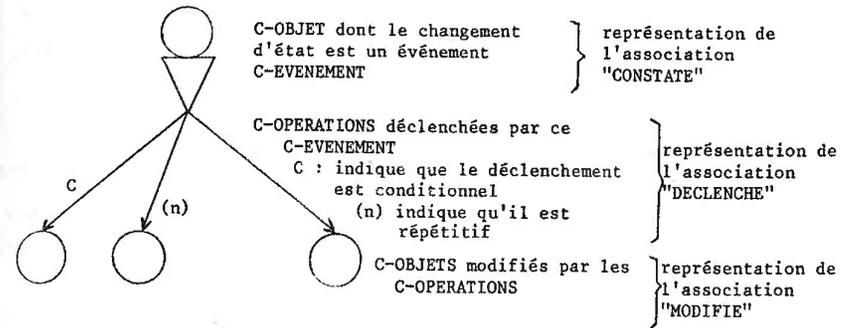


Figure II-2

II-1.2.2.- Exemple

Nous allons présenter un exemple de structure conceptuelle, correspondant à un système de vente par correspondance, en nous réservant de montrer comment aboutir à cette description au paragraphe présentant la méthode associée au modèle conceptuel.

II-1.2.2.1.- Énoncé du problème

- Une entreprise vend 25000 articles "catalogués" à des entreprises industrielles.
- Une commande ne peut pas être enregistrée si le client n'est pas connu de l'entreprise.
- Un numéro d'identification est donné à une commande par le service commercial.
- Une commande enregistrée peut avoir plusieurs lignes de commande.
- Chaque ligne de commande correspond à un produit différent.
- Une ligne de commande peut être :
 - refusée quand le produit commandé n'est pas "catalogué". Elle n'est plus jamais considérée dans cette commande.
 - différée jusqu'à consentement :
 - sur le prix
 - sur la date de livraison.
 - acceptée quand il y a accord sur le prix et la date de livraison.
- Au moment de la livraison, une commande peut être momentanément différée jusqu'à ce que le produit soit réapprovisionné. La facture de la commande est envoyée seulement après livraison complète de toutes les lignes de commande.
- A toute rupture de stock de produit est associée une demande de réapprovisionnement.

II-1.2.2.2.- Sous-schéma dynamique

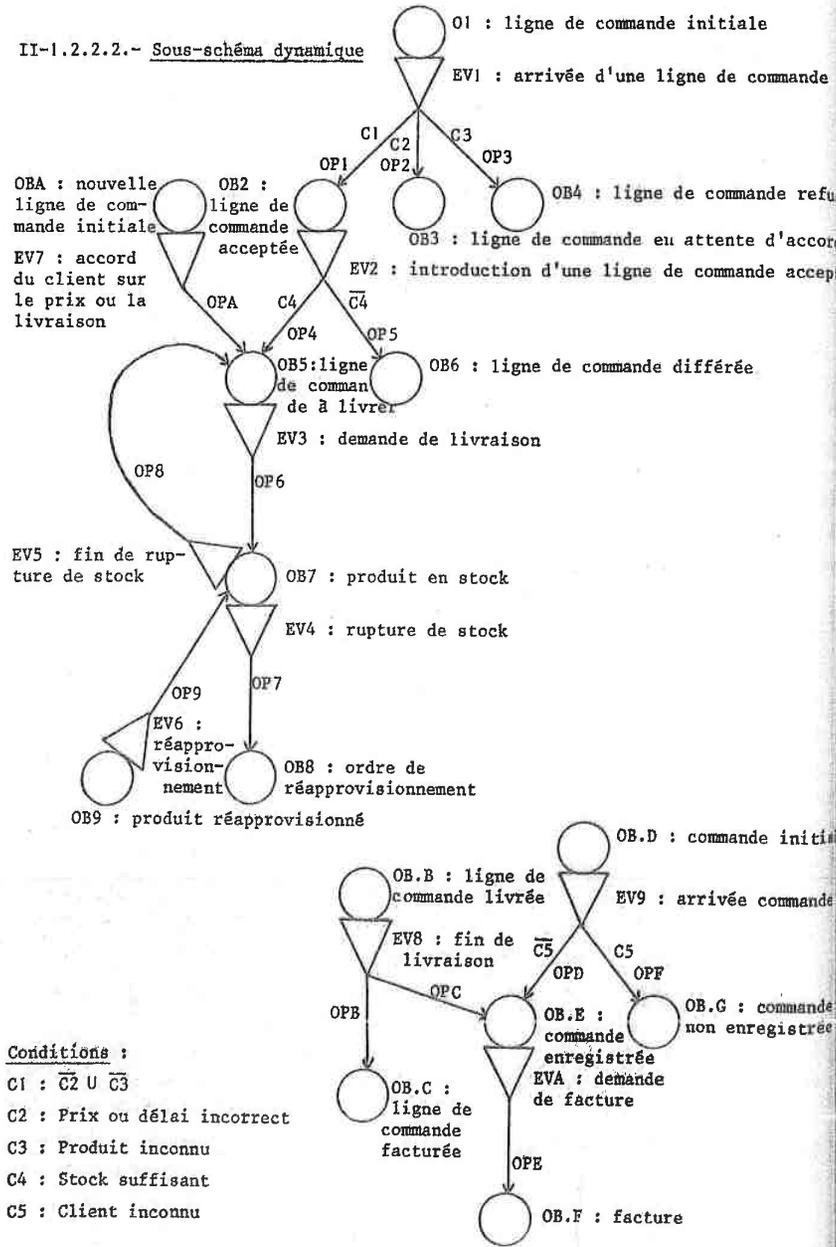


Figure II-3-A

II-1.2.2.3.- Sous-schéma de données

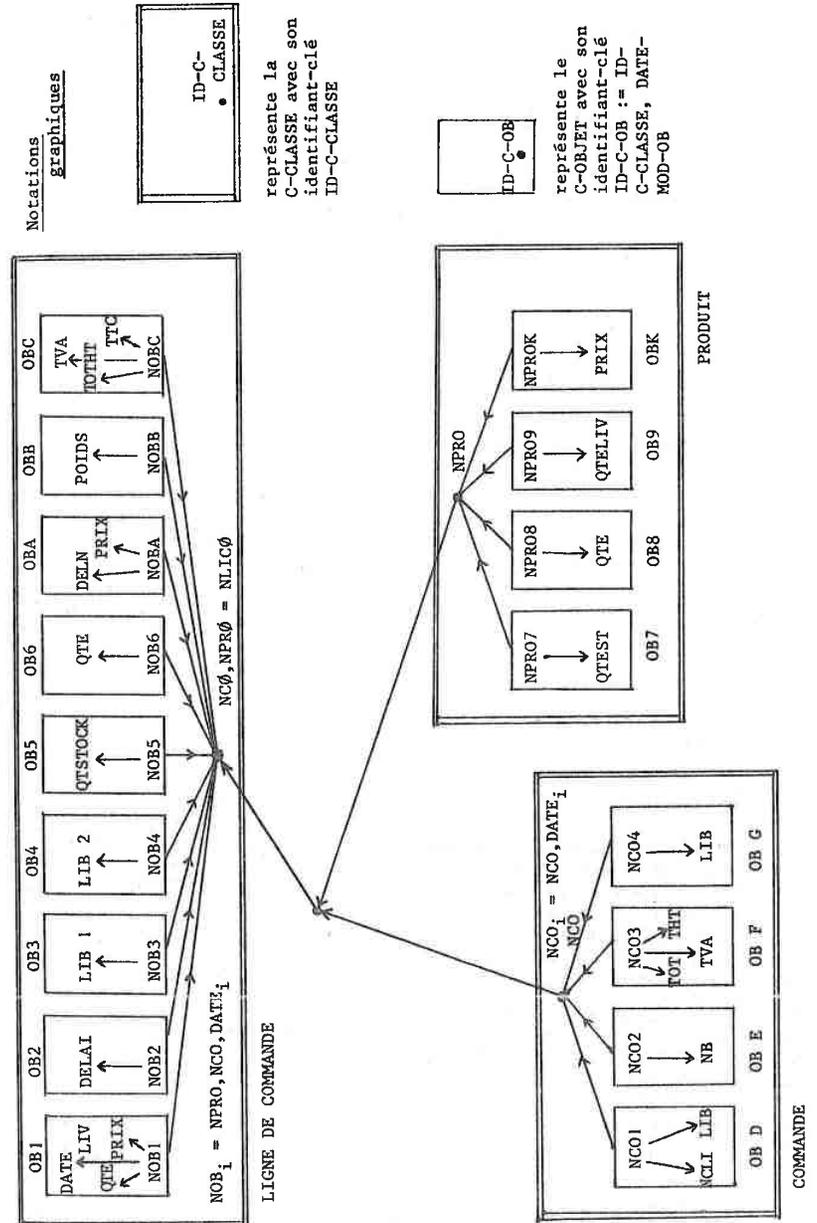


Figure II-3-B

II-1.2.2.4.- LEXIQUE1) C-OBJETS - C-CLASSES

- | | | |
|---|---|------------------------------|
| - OB1 : Ligne de commande initiale | } | C-CLASSE "Ligne de Commande" |
| - OB2 : Ligne de commande acceptée | | |
| - OB3 : Ligne de commande en attente d'accord | | |
| - OB4 : Ligne de commande refusée | | |
| - OB5 : Ligne de commande à livrer | | |
| - OB6 : Ligne de commande différée | | |
| - OBA : Nouvelle ligne de commande initiale | } | C-CLASSE "Produit" |
| - OBB : Ligne de commande livrée | | |
| - OBC : Ligne de commande facturée | | |
| - OB7 : Produit en stock | | |
| - OB8 : Ordre de réapprovisionnement | | |
| - OB9 : Produit réapprovisionné | | |
| - OBK : Prix du produit | } | C-CLASSE "Commande" |
| - OBD : Commande initiale | | |
| - OBE : Commande enregistrée | | |
| - OBF : Facture | | |
| - OBG : Commande non enregistrée | | |

2) C-EVENEMENTS

- EV1 : Arrivée d'une ligne de commande
- EV2 : Introduction d'une ligne de commande acceptée
- EV3 : Demande de livraison
- EV4 : Rupture de stock
- EV5 : Fin de rupture de stock
- EV6 : Réapprovisionnement
- EV7 : Accord sur date ou prix
- EV8 : Fin de livraison
- EV9 : Arrivée d'une commande
- EVA : Demande de facture

3) C-OPERATIONS

- OP1 : Acceptation d'une ligne de commande
- OP2 : Ajournement d'une ligne de commande
- OP3 : Refus d'une ligne de commande
- OP4 : Livraison d'une ligne de commande
- OP5 : Différé d'une ligne de commande
- OP6 : Mise à jour du stock
- OP7 : Emission d'un ordre de réapprovisionnement
- OP8 : Prise en compte des lignes de commande différées
- OP9 : Réapprovisionnement du stock
- OPA : Révision de la ligne de commande en attente d'accord
- OPB : Facturation de la ligne de commande
- OPC : Mise à jour de la commande reçue
- OPD : Enregistrement de la commande
- OPE : Facturation
- OPF : Refus d'enregistrement d'une commande

II-1.3.- La méthodeII-1.3.1.- Présentation de la méthode

• L'approche conceptuelle est de type structuraliste. Elle conduit à définir un système par sa structure. En fait, le système est défini par deux sous-structures (voir paragraphe II-1.2.) :

- le schéma statique, ou schéma de données, image de la structure des constituants du système représenté en termes de C-OBJETS et C-CLASSES.
- le schéma dynamique, image de la structure de fonctionnement du système représenté en termes de C-OBJETS, C-EVENEMENTS, C-OPERATIONS et d'associations entre ces concepts.

• Notre expérience de la conception des bases de données [8], [9] nous conduit à proposer de nous intéresser tout d'abord au schéma statique, puis d'étudier le comportement dynamique des objets (représentés par les C-OBJETS) c'est-à-dire le schéma dynamique. Il est de plus bien évident que la définition de l'un de ces deux schémas permet de vérifier la définition du second et de la corriger éventuellement, selon un processus d'allers et retours, tout en remarquant que si l'on a été systématiques dans la définition du schéma statique, ce processus n'a pas lieu d'exister.

• En conséquence de ces remarques, la méthode d'analyse conceptuelle d'un système consiste à :

a) définir le schéma statique de ce système, c'est-à-dire :

- 1) déterminer quelles sont les classes d'objets du système
- 2) représenter ces classes d'objets au moyen de C-OBJETS regroupés en C-CLASSES.

b) définir le schéma dynamique de ce système, c'est-à-dire :

- 3) déterminer quels sont les changements d'états remarquables des objets (représentés par les C-OBJETS) qui constituent des événements (représentés par les C-EVENEMENTS).
- 4) déterminer quelles sont les actions induites sur le système par ces changements d'états, c'est-à-dire déterminer les opérations (représentées par les C-OPERATIONS) déclenchées par les événements (représentés par les C-EVENEMENTS).

II-1.3.2.- Exemple d'application de cette méthode

Nous présenterons l'application de cette méthode sur une partie de l'exemple présenté au paragraphe II-1.2.2. correspondant au traitement d'une ligne de commande acceptée :

- chaque fois qu'une ligne de commande concernant un produit est acceptée, on livre le produit et on met à jour le stock de ce produit si celui-ci est suffisant pour satisfaire cette commande relative à ce produit, sinon on diffère le traitement de cette ligne de commande.

- lorsqu'une rupture de stock sur un produit a été constaté (après mise à jour correspondant à une livraison), il y a émission d'un bon de réapprovisionnement.

Enfin, le réapprovisionnement du stock d'un produit implique l'essai de prise en compte prioritaire des lignes de commandes mises en attente à cause de l'insuffisance du stock de ce produit.

L'analyse conceptuelle de ce problème à l'aide de la méthode présentée conduit :

1) à déterminer quelles sont les classes d'objets du problème

ainsi, dans le cadre de l'exemple proposé, deux classes d'objets apparaissent :

- la classe d'objets "LIGNE DE COMMANDE"
- la classe d'objets "PRODUIT".

2) à représenter ces classes d'objets au moyen de C-OBJETS

ainsi, du point de vue de la dynamique, la classe d'objets "LIGNE DE COMMANDE" comporte trois aspects différents : elle est tout d'abord à analyser (à traiter), puis soit à livrer, soit en attente. Donc cette classe d'objets sera décrite par trois C-OBJETS :

- le C-OBJET "LIGNE DE COMMANDE ACCEPTEE" (OB2)
- le C-OBJET "LIGNE DE COMMANDE A LIVRER" (OB5)
- le C-OBJET "LIGNE DE COMMANDE DIFFEREE" (OB6)

De même, la classe d'objets "PRODUIT" comporte trois aspects différents, du point de vue de la dynamique : un produit est soit en stock, soit en attente de réapprovisionnement, soit réapprovisionné. Donc cette classe d'objets sera décrite par trois C-OBJETS :

- le C-OBJET "PRODUIT EN STOCK" (OB7)
- le C-OBJET "ORDRE DE REAPPROVISIONNEMENT" (OB8)
- le C-OBJET "PRODUIT REAPPROVISIONNE" (OB9).

3) à déterminer quels sont les changements d'états remarquables des objets (représentés par les C-OBJETS) qui constituent des événements (représentés par des C-EVENEMENTS)

ainsi dans le cadre de notre exemple :

- le passage d'un objet du type OB2 (commande acceptée) de l'état inexistant à l'état existant est un événement représenté par le C-EVENEMENT EV2 (arrivée d'une ligne de commande à traiter).

- le passage d'un objet du type OB5 (commande à livrer) de l'état inexistant à l'état existant est un événement représenté par le C-EVENEMENT EV3 (arrivée d'une ligne de commande à livrer).

- le passage d'un objet du type OB7 (produit en stock) dans l'état où la propriété QTESTOCK est inférieure à un seuil donné est un événement représenté par le C-EVENEMENT EV4 (rupture de stock).

- le passage d'un objet du type OB7 (produit en stock) de l'état où la propriété QTESTOCK est inférieure à un seuil donné à l'état où cette propriété est supérieure à ce même seuil est un événement représenté par le C-EVENEMENT EV5 (suppression de rupture de stock).

- le passage d'un objet du type OB9 (produit réapprovisionné) de l'état inexistant à l'état existant est un événement représenté par le C-EVENEMENT EV6 (arrivée d'un produit réapprovisionné).

4) à déterminer quelles sont les actions induites sur le système par ces changements d'états, c'est-à-dire déterminer les opérations (représentées par les C-OPERATIONS) déclenchées par les événements (représentés par les C-EVENEMENTS)

Dans le cadre de notre exemple :

- un événement du type EV2 (arrivée d'une ligne de commande acceptée) déclenche, si le stock du produit concerné par cette ligne de commande est suffisant (condition C4), une opération de création d'un objet du type OB5 (ligne de commande à livrer). Cette opération sera représentée par la C-OPERATION OP4 de création du C-OBJET OB5 (ligne de commande à livrer). Sinon, si la condition C4 n'est pas satisfaite, un événement du type EV2 déclenche une opération de création d'un objet du type OB6 (ligne de commande différée). Cette opération sera représentée par la C-OPERATION OP5 de création du C-OBJET OB6 (ligne de commande différée).
- un événement du type EV3 (arrivée d'une ligne de commande à livrer) déclenche une opération de mise à jour d'un objet du type OB7 (produit en stock). Cette opération sera représentée par le C-OPERATION OP6 de mise à jour du C-OBJET OB7 (produit en stock).
- un événement du type EV4 (rupture de stock) déclenche une opération de création d'un objet du type OB8 (ordre de réapprovisionnement). Cette opération sera représentée par la C-OPERATION OP7 de création du C-OBJET OB8.
- un événement du type EV5 (suppression de rupture de stock) déclenche une opération de prise en compte des lignes de commandes différées, c'est-à-dire une opération de création d'un ou de plusieurs objets du type OB5 (ligne de

commande à livrer) à partir des objets du type OB6 (ligne de commande différée). Cette opération sera représentée par la C-OPERATION OP8 de création du C-OBJET OB5 (ligne de commande à livrer) intitulée "prise en compte des lignes de commande différées".

- un événement du type EV6 (arrivée d'un produit réapprovisionné) déclenche une opération de mise à jour d'un objet du type OB7 (produit en stock). Cette opération sera représentée par la C-OPERATION OP9 de mise à jour du C-OBJET OB7 (produit en stock).

La partie du sous-schéma dynamique correspondante sera alors :

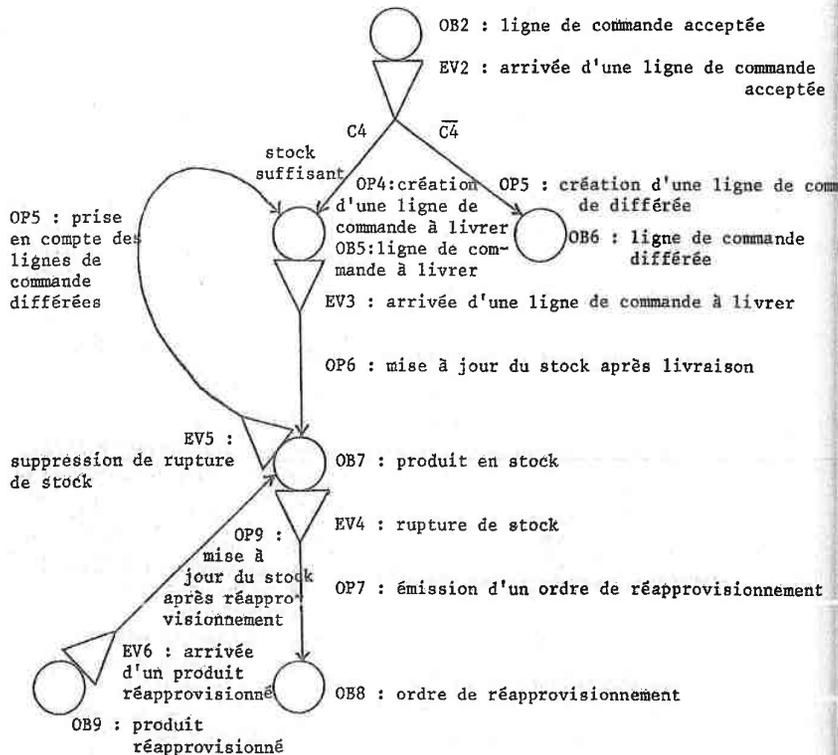


Figure II-4

II-1.4.- La machine abstraite

Comme la description conceptuelle d'un système représente ses règles logiques de fonctionnement [86], [87], nous associons à cette description abstraite un automate pour gérer le fonctionnement de tout système dont on connaît la logique en termes conceptuels. Cet automate, que nous avons appelé machine abstraite exécute des actions selon la logique définie par le schéma conceptuel du système géré.

II-1.4.1.- Définition et fonctionnement de la machine abstraite

La machine abstraite est constituée d'un ensemble de mécanismes opératoires propres à assurer la gestion du fonctionnement d'un système défini par sa structure conceptuelle [86], [63]. Elle comporte donc un ensemble de commandes qui assurent que le fonctionnement du système respecte les règles logiques de fonctionnement décrites dans le schéma conceptuel. Cet ensemble de commandes correspond à l'ensemble des commandes opératoires d'un système en fonctionnement, au sens de la théorie des processeurs discrets [41], et la machine abstraite correspond à la partie opératoire du système, alors que le schéma dynamique traduisant la logique de fonctionnement du système en constitue la partie contrôle, toujours au sens de la théorie des processeurs discrets.

Donc la machine abstraite intègre les mécanismes qui assurent le fonctionnement de tout système décrit en termes de structure conceptuelle [39]. Elle fonctionne selon ses propres règles qui sont mises en oeuvre selon le schéma de fonctionnement du système (la partie opératoire d'un système agit selon la structure de sa partie contrôle [41]). Son fonctionnement est basé sur les trois primitives suivantes :

- reconnaître les événements,
- déclencher l'exécution des opérations,
- mémoriser des représentations.

Le fonctionnement de la machine abstraite, pour un système particulier, peut être schématisé de la manière suivante : la machine reconnaît les événements qui se produisent, détermine les opérations qui doivent être déclenchées (en exploitant la structure conceptuelle correspondant au système), déclenche et contrôle leur exécution, vérifie si les changements d'état provoqués par les opérations sont des événements et, si oui, détermine et déclenche les actions conséquentes.

II-1.4.2.- Représentation conceptuelle du fonctionnement de la machine abstraite

Afin de décrire plus précisément les règles de fonctionnement propres à la machine abstraite, nous utilisons le modèle dynamique et nous présentons le schéma

conceptuel du fonctionnement de la machine abstraite. Ce fonctionnement est décrit au moyen des concepts d'objets, d'opérations et d'événements, bien qu'ils aient une signification particulière que nous allons maintenant exposer.

- les objets de la machine abstraite (a-objets) doivent refléter dans quel état de l'évolution de sa dynamique est le système représenté. Ils doivent également refléter les conséquences de cette évolution de sa dynamique sur le système représenté. Ainsi, un a-objet représentera :

- une occurrence de c-objet dont l'état a été modifié,
- une occurrence de c-événement qui a été constatée,
- une occurrence de c-opération qui a été choisie pour être exécutée (nous appellerons ces c-opérations : c-opération déclenchement),
- une occurrence de c-opération dont l'exécution a été achevée.

Nous pouvons également percevoir un a-objet comme une référence à l'état du système représenté.

- les événements de la machine abstraite (a-événements) décrivent les changements d'état des a-objets qui sont pertinents pour la machine abstraite. Cela signifie que le changement d'état d'un a-objet (par exemple un a-objet qui représente une occurrence du c-objet STOCK) n'est pas toujours un a-événement. Un certain prédicat doit être satisfait pour pouvoir identifier un changement d'état d'un a-objet comme un a-événement.

- les opérations de la machine abstraite (a-opérations) représentent les actions exécutées par la machine abstraite pour pouvoir assurer l'évolution du système représenté. Une a-opération modifie un a-objet (et, dans ce sens, le fonctionnement de la machine abstraite évolue) et elle est déclenchée par un a-événement.

On trouvera une description plus complète des a-objets, a-événements et a-opérations dans [86].

Nous illustrerons le fonctionnement de la machine abstraite au moyen d'une description graphique basée sur les conventions suivantes :

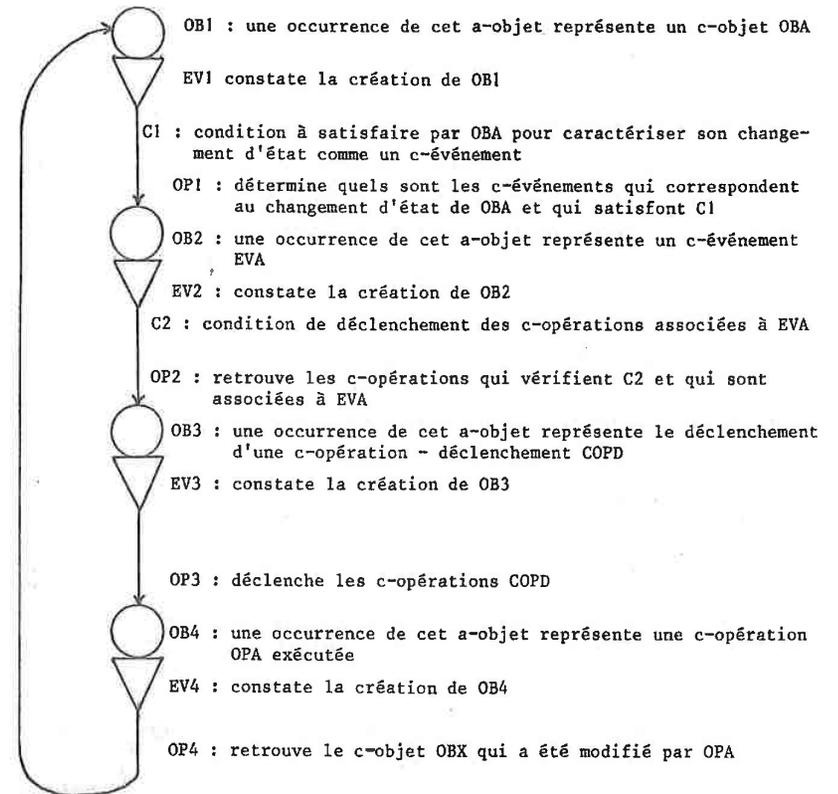
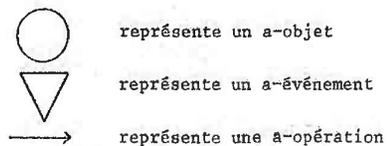


Figure II-5 : Schéma conceptuel du fonctionnement de la machine abstraite

II-2.- LE MODELE ASSURE LA PRISE EN COMPTE DE LA SYNCHRONISATION AU NIVEAU CONCEPTUEL

A l'issue du chapitre I, nous avons établi que pour résoudre un problème de synchronisation les trois conditions suivantes doivent être satisfaites :

- C1 : déterminer l'ensemble des processus du système et, par conséquent, l'ensemble des ressources du système.
- C2 : définir les interrelations entre ces processus, c'est-à-dire leur dépendance ou leur indépendance.

- C3 : définir leurs règles de coopération temporelle en fonction des règles définissant la logique du problème envisagé et le partage de ressources, c'est-à-dire définir un ordre sur l'exécution des processus.

Or, au niveau conceptuel, niveau où nous nous situerons dans la suite de ce travail, les problèmes liés au partage de ressources et conditionnés par des contraintes logiques ou physiques d'accès à celles-ci n'ont pas à être pris en compte. Néanmoins, il existe des cas où ce partage de ressources fait partie intégrante de la logique du problème considéré. Dans ce cas, les règles qui le définissent sont exprimées par les relations dues à la logique du problème, donc il n'existe pas de règles de coopération temporelle dues au partage de ressources à prendre en compte au niveau conceptuel, hormis celles relevant de la logique du problème traité. En conclusion, la synchronisation au niveau conceptuel peut être définie comme l'ensemble des relations temporelles entre les processus du système dûes à la logique du système et les seules ressources à définir à ce niveau sont les constituants du système (et non les ressources "techniques" nécessaires à son fonctionnement. De même les optimisations technologiques telles que le parallélisme ou les limitations techniques entraînant par exemple l'exclusion mutuelle n'ont pas à être prises en compte. Néanmoins, lors de l'implémentation de ce système, pour des raisons économiques ou de performance, ces problèmes auront alors à être pris en compte).

Nous allons donc montrer que le modèle conceptuel que nous proposons permet de satisfaire à ces trois conditions relatives à la prise en compte de la synchronisation, indépendamment de toute optimisation technologique ou d'usage.

II-2.1.- Le modèle conceptuel permet de satisfaire aux trois conditions de synchronisation

Raisonnons d'abord "intuitivement" puis reprenons le raisonnement de manière plus stricte.

• Condition C1 :

Dans une approche conceptuelle de la synchronisation, l'ensemble des processus est défini comme l'ensemble des opérations décrites par les C-OPERATIONS, et l'ensemble des constituants du système est défini comme l'ensemble des objets représentés par les C-OBJETS. Une telle approche correspond à un raisonnement en termes de processus élémentaires, c'est-à-dire en termes de processus déclenchés à la suite d'un changement d'état d'un seul objet et agissant sur un seul objet.

• Condition C2 :

Chaque opération (qui constitue un "processus" au sens de notre approche) décrite par une C-OPERATION est nécessairement déclenché par un événement représenté par un C-EVENEMENT. Donc l'indépendance ou la dépendance entre les événements déclencheurs des opérations exprime la dépendance ou l'indépendance entre ces opérations. Comme un événement est une réalisation d'un C-EVENEMENT, et comme une opération est une réalisation d'une C-OPERATION, la dépendance ou l'indépendance entre C-EVENEMENTS déclencheurs des C-OPERATIONS exprime la dépendance ou l'indépendance entre les C-OPERATIONS.

Nous reviendrons sur ce point dans le paragraphe suivant. Néanmoins illustrons le de manière informelle.

Si l'on considère l'exemple de système de vente par correspondance présenté au paragraphe II-1.2.2., il apparaît clairement qu'un événement du type EV4 (rupture de stock) ne peut se produire que si un événement du type EV3 (arrivée d'une ligne de commande à livrer) est apparu. En effet, une rupture de stock ne peut être que la conséquence d'une mise à jour d'un stock de produit effectuée à la suite d'une livraison de produit correspondant à une ligne de commande. Ainsi, nous pouvons dire que le C-EVENEMENT EV4 est dépendant du C-EVENEMENT EV3 (ce qui signifie qu'une réalisation de EV4 ne peut avoir lieu qu'après une réalisation correspondante de EV3). Comme la C-OPERATION OP7 (émission d'un ordre de réapprovisionnement) est déclenchée par le C-EVENEMENT EV4 et comme la C-OPERATION OP6 (mise à jour du stock) est déclenchée par le C-EVENEMENT EV3, nous pouvons dire que la C-OPERATION OP7 est dépendante de la C-OPERATION OP6. Ainsi, nous venons d'illustrer le fait que les relations de dépendance ou d'indépendance entre événements (C-EVENEMENTS) exprime la dépendance ou l'indépendance des opérations (C-OPERATIONS) qu'ils déclenchent.

• Condition C3 :

Chaque changement d'état du système est dû à l'exécution d'une opération, entraînant ainsi la modification d'un objet du système. De plus, chaque événement est un changement d'état, et il déclenche une ou plusieurs opérations (les processus selon notre approche). Donc l'ordonnement des c-opérations conformément à leurs règles de dépendance ou d'indépendance représente leurs règles de coopération temporelles, c'est-à-dire qu'il permet de définir un ordre sur leur exécution. Cet ordonnancement est défini par la concaténation de cycles dynamiques au moyen du concept de c-événement (un cycle dynamique est constitué par un c-événement constatant le changement d'état d'un c-objet, par les c-opérations déclenchées par ce c-événement et par les objets modifiés par ces c-opérations).

Intuitivement, nous pouvons dire que l'ordonnancement entre les C-OPERATIONS (représentant les processus du système) conformément à leurs règles de dépendance ou d'indépendance, et en fonction des relations de causalité du modèle (un événement déclenche des opérations qui provoquent des changements d'états qui peuvent à leur tour être des événements) permet de traduire les règles de coopération temporelle entre les processus du système (les opérations selon notre approche). Il se traduit par la concaténation de cycles dynamiques au moyen du concept de C-EVENEMENT.

Nous allons illustrer ce point qui sera développé plus en détail dans la suite de ce travail en nous référant à l'exemple présenté au paragraphe II-1.2.2. :

Considérons la partie du schéma conceptuel traduisant plus particulièrement la gestion du stock des produits. Elle est exprimée par le sous-schéma suivant :

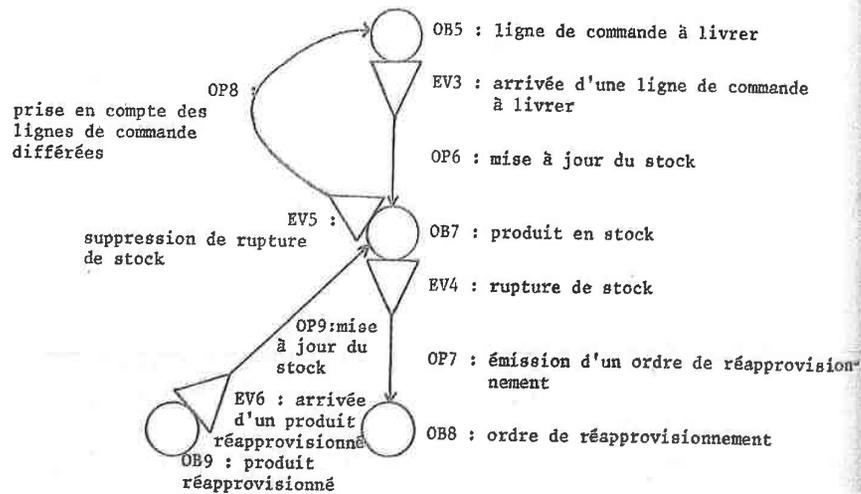


Figure II-6

Sur ce schéma, il apparaît que le C-EVENEMENT EV4 est dépendant du C-EVENEMENT EV3 donc la C-OPERATION OP7 déclenchée par EV4 est dépendante de la C-OPERATION OP6 déclenchée par EV2 (ceci a été justifié lors de l'analyse de la condition C2).

De même, le C-EVENEMENT EV5 est dépendant du C-EVENEMENT EV6 puisqu'une rupture de stock d'un produit ne peut être supprimée que par le réapprovisionnement de ce produit. Donc la C-OPERATION OP8 déclenchée par EV5 est dépendante de la C-OPERATION OP9 déclenchée par EV6.

De plus, l'ordonnancement entre les C-OPERATIONS représentant les règles de coopération temporelle entre les processus (les opérations) se fait par concaténation de cycles dynamiques.

ce sous-schéma conceptuel fait apparaître quatre cycles dynamiques :

- C1 := (EV3, OP6, OB7)
- C2 := (EV4, OP7, OB8)
- C3 := (EV5, OP8, OB5)
- C4 := (EV6, OP9, OB7)

les trois types d'ordonnements possibles entre les processus (les opérations) sont les suivants :

- 1) mise à jour du stock (correspondant à la livraison d'une ligne de commande mais n'entraînant pas de rupture de stock) : OP6.
- 2) mise à jour du stock entraînant une rupture de stock donc entraînant l'émission d'un ordre de réapprovisionnement : OP6 → OP7.
- 3) réapprovisionnement entraînant la prise en compte des lignes de commande différées (essai de satisfaction de ces lignes de commande) : OP9 → OP8.

Or ces trois types d'ordonnements possibles sont parfaitement bien exprimés par la concaténation des quatre cycles dynamiques C1, C2, C3, C4 au moyen du concept de C-EVENEMENT :

- 1) une réalisation du C-EVENEMENT EV3 se produit : elle déclenche une réalisation de OP6 entraînant un changement d'état d'une occurrence de OB7 qui ne constitue pas une rupture de stock. Nous sommes donc en présence d'une occurrence du cycle dynamique C1, traduisant la le premier type d'ordonnement possible.
- 2) une réalisation du C-EVENEMENT EV3 se produit : elle déclenche une réalisation de OP6 entraînant un changement d'état d'une occurrence de OB7 qui constitue une rupture de stock (cet enchaînement événement-opération-objet modifié constitue une occurrence du cycle dynamique C1). Le changement d'état d'une occurrence de OB7 est une réalisation du C-EVENEMENT EV4 : elle déclenche aussitôt une réalisation de OP7 entraînant la création d'une occurrence de OB8 (cet enchaînement événement-opération-objet est une occurrence du cycle dynamique C2 concaténée à l'occurrence de C1 précédente par la réalisation de EV4). Constatons ainsi que la concaténation d'une occurrence de

C1 par l'intermédiaire d'une occurrence de EV4 permet de traduire le deuxième type d'ordonnement possible.

- 3) une réalisation du C-EVENEMENT EV6 se produit : elle déclenche une réalisation de OP9 entraînant un changement d'état d'une occurrence de OB7 qui constitue une suppression de rupture de stock (occurrence du cycle dynamique C3). Ce changement d'état d'une occurrence de OB7 est une réalisation du C-EVENEMENT EV5 : elle déclenche aussitôt une réalisation de OP8 entraînant la création d'une ou plusieurs occurrences de OB5 (occurrence du cycle dynamique C4 concaténée à l'occurrence de C3 précédente par l'intermédiaire d'une réalisation de EV5). Donc la concaténation d'une occurrence du cycle dynamique C4 à une occurrence de C3 par l'intermédiaire d'une réalisation de EV5 permet de traduire le troisième type d'ordonnement possible.

Nous venons donc ainsi d'illustrer le fait que la concaténation des cycles dynamiques au moyen du concept de C-EVENEMENT permet de représenter les règles de coopération temporelle entre les processus du système (les opérations dans notre approche).

Cette analyse rapide sur des exemples permet de constater que la représentation conceptuelle proposée assure la prise en compte des problèmes de synchronisation. Les trois conditions à vérifier pour la réalisation de ce type de problèmes sont satisfaites.

Notre propos est maintenant de définir les dépendances chronologiques entre événements en essayant d'en dégager une typologie. Nous montrerons ensuite comment cette typologie permet de définir trois situations de synchronisation de base proches de celles de BRINCH-HANSEN (: une opération suit toujours une autre opération, elle peut suivre une autre opération, ou elle est indépendante d'une autre opération).

II-2.2.- Typologie des relations temporelles entre événements

Nous allons montrer dans ce paragraphe que l'on peut déduire la synchronisation d'un système décrit sous forme de schéma conceptuel au moyen d'un graphe de dépendances fonctionnelles chronologiques [38]. Parmi toutes ces dépendances fonctionnelles chronologiques, nous isolerons celles qui permettent de retrouver les relations temporelles entre événements puis entre opérations, c'est-à-dire celles qui définissent la synchronisation du système abstrait. Nous définirons ainsi le graphe des dépendances entre événements puis le graphe des dépendances entre opérations.

En fait, l'ensemble des dépendances fonctionnelles chronologiques entre C-EVENEMENTS qui existent dans un schéma conceptuel résulte :

- de l'analyse causale de la structure dynamique (un C-EVENEMENT déclenche une C-OPERATION provoquant un changement d'état d'un C-OBJET constaté par un autre C-EVENEMENT, ce qui implique une dépendance fonctionnelle chronologique du second de ces C-EVENEMENTS par rapport au premier).
- de l'analyse des contraintes d'intégrité pour les C-EVENEMENTS de la frontière (un C-EVENEMENT n'a de sens que si un autre C-EVENEMENT est déjà survenu : par exemple le C-EVENEMENT "arrivée d'une ligne de commande" ne peut survenir que si le C-EVENEMENT "arrivée de la commande" est déjà survenu).

II-2.2.1.- Dépendances chronologiques entre C-EVENEMENTS

Soient EV_i et EV_j deux C-EVENEMENTS tels que :

- EV_i - constate un changement d'état d'un C-OBJET OB_i
- déclenche un ensemble de C-OPERATIONS $EOP^i = \{OP_1^i, \dots, OP_k^i, \dots, OP_n^i\}$
- EV_j constate un changement d'état d'un C-OBJET OB_j

Ceci sera exprimé selon le formalisme des dépendances fonctionnelles de l'algèbre relationnelle de CODD [19], et la notion de dépendance fonctionnelle multi-valuée [37] par :

$$EV_j \rightarrow OB_j$$

$$EV_i \rightarrow OB_i$$

$$EV_i \twoheadrightarrow OP_k^i \text{ (c'est-à-dire : } EOP^i = \{OP_k^i / EV_i \twoheadrightarrow OP_k^i\})$$

De plus, nous désignerons par $CHANGETAT_r(OB_k)$ le $r^{\text{ème}}$ type de changement d'état possible du C-OBJET OB_k , l'ensemble des types de changement d'état possibles d'un C-OBJET pouvant être obtenu par analyse de la collection de relations définissant la structure conceptuelle.

Enfin, PI^k et PF^k désignent respectivement le prédicat initial et le prédicat final associés au C-EVENEMENT EV_k :

$$EV_k \rightarrow PI_k, PF_k$$

et $C(OP_k)$ désigne la condition de déclenchement associée à la C-OPERATION OP_k .

Nous illustrerons notre propos en nous appuyant sur la partie de l'exemple présenté au paragraphe II-1.2.2. concernant le traitement d'une ligne de commande acceptée. Nous présentons ci-dessous la partie du schéma dynamique correspondante, où nous avons numéroté différemment les différents concepts afin d'augmenter la clarté des exemples.

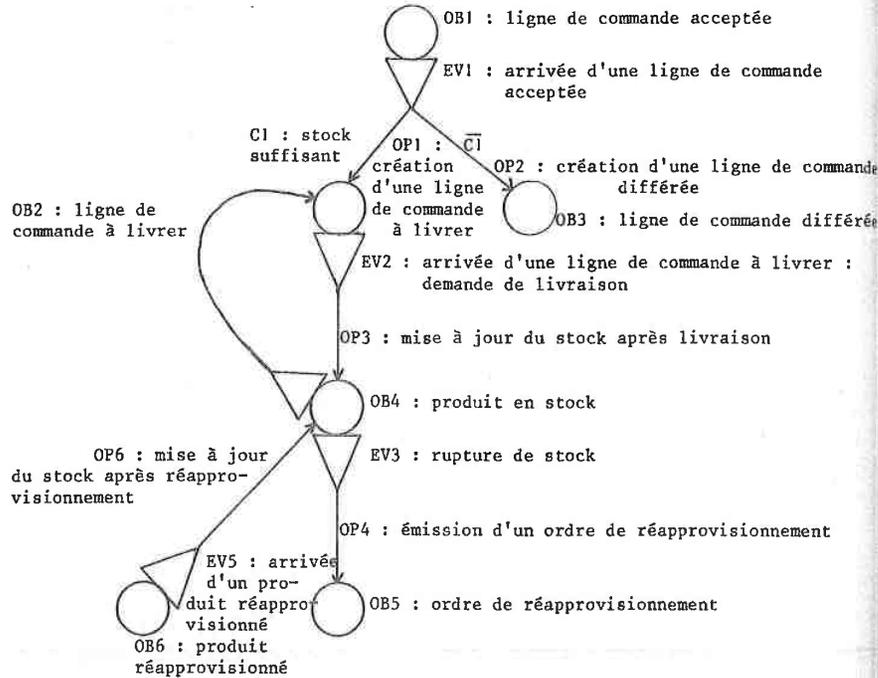


Figure II-7 : traitement d'une ligne de commande acceptée

Définition 1 : Dépendance chronologique permanente directe (DCPD)

• Définition

Nous dirons que EV_j est en dépendance chronologique permanente directe avec EV_i , et nous noterons $EV_i \xrightarrow{P} EV_j$ si et seulement si :

$$(1) (\exists OP_k^i \in EOP^i) (OP_k^i \longrightarrow OB_j)$$

$$(2) (OP_k^i \longrightarrow \text{CHANGETAT}_r(OB_j)) \wedge (EV_j \longrightarrow \text{CHANGETAT}_r(OB_j))$$

$$(3) (PI_j = PF_j = 1) \wedge (C(OP_k^i) = 1)$$

- (1) traduit le fait que OP_k^i soit déclenchée par EV_i et modifie OB_j
- (2) traduit le fait que le changement d'état événementiel constaté par EV_j est induit par OP_k^i
- (3) traduit le fait que ce changement d'état provoqué par OP_k^i est toujours le changement d'état événementiel constaté par EV_j , OP_k^i étant déclenché inconditionnellement par EV_i .

Ainsi, EV_j est en dépendance chronologique permanente directe (DCPD) avec EV_i si et seulement s'il existe une C-OPERATION OP_k^i déclenchée inconditionnellement par EV_i et qui provoque un changement d'état du C-OBJET OB_j , qui est toujours le changement d'état événementiel constaté par le C-EVENEMENT EV_j (un changement d'état événementiel étant défini comme un changement d'état qui constitue un c-événement).

Nous noterons $EV_i \xrightarrow{P} EV_j$ pour traduire le fait que EV_j est en dépendance chronologique permanente directe avec EV_i , afin d'être cohérent, du point de vue des notations avec les notions de dépendances fonctionnelles [10].

• Exemples :

Sur l'exemple présenté en tête de ce paragraphe, il apparaît que

- le C-EVENEMENT EV_4 est en DCPD avec le C-EVENEMENT EV_5 puisque la mise à jour du stock après réapprovisionnement (OP_6) est déclenché inconditionnellement par EV_5 et elle provoque un changement d'état du C-OBJET OB_4 "Produit en stock" qui est toujours l'état événementiel constaté par EV_4 (fin de rupture de stock). Nous noterons alors :

$$EV_5 \xrightarrow{P} EV_4$$

- de même, le C-EVENEMENT EV_2 est en DCPD avec le C-EVENEMENT EV_4 puisque la prise en compte des lignes de commande différées (OP_5) est déclenchée inconditionnellement par EV_4 et elle provoque un changement d'état (création) du C-OBJET OB_2 "ligne de commande à livrer" qui est toujours l'état événementiel constaté par EV_2 (arrivée d'une ligne de commande à livrer). Nous noterons donc :

$$EV_4 \xrightarrow{P} EV_2$$

Il apparaît donc que si EV_j est en DCPD avec EV_i , une réalisation du C-EVENEMENT EV_j suit toujours une réalisation du C-EVENEMENT EV_i (Nous dirons que EV_j suit toujours EV_i) et ainsi, la relation de dépendance chronologique permanente directe peut être considérée comme une relation de dépendance fonctionnelle forte directe entre C-EVENEMENTS, à laquelle elle donne une signification en termes de synchronisation (rappelons qu'une relation de dépendance fonctionnelle $A \rightarrow B$ est dite forte si et seulement si la connaissance d'une réalisation de A détermine la connaissance d'une et une seule réalisation de B [59]).

Définition 2 : dépendance chronologique conditionnelle directe (DCCD)

• Définition

Nous dirons que EV_j est en dépendance chronologique conditionnelle directe avec EV_i , et nous noterons $EV_i \xrightarrow{C} EV_j$ si et seulement si :

- (1) $(\exists OP_k^i \in EOP^i)(OP_k^i \rightarrow OB_j)$
- (2) $(OP_k^i \rightarrow \text{CHANGETAT}_r(OB_j)) \wedge (EV_j \rightarrow \text{CHANGETAT}_r(OB_j))$
- (3) $(C(OP_k^i) \neq 1) \vee ((C(OP_k^i) = 1) \wedge [(PI_j \neq 1) \vee (PF_j \neq 1)])$

- (1) traduit le fait que OP_k^i soit déclenchée par EV_i et modifie OB_j
- (2) traduit le fait que le changement d'état événementiel constaté par EV_j est induit par OP_k^i .
- (3) traduit le fait que -soit OP_k^i est déclenchée conditionnellement par EV_i
- soit OP_k^i est déclenchée inconditionnellement par EV_i et le changement d'état provoqué par OP_k^i n'est pas toujours le changement d'état événementiel constaté par EV_j .

Ainsi, EV_j est en dépendance chronologique conditionnelle directe (DCCD) avec EV_i si et seulement s'il existe une C-OPERATION OP_k^i telle que :

- soit OP_k^i est déclenchée conditionnellement par EV_i et le changement d'état du C-OBJET OB_j provoqué par cette C-OPERATION peut toujours ou non être le changement d'état événementiel constaté par EV_j (il n'importe pas).
- soit OP_k^i est inconditionnellement déclenchée par EV_i et elle provoque un changement d'état du C-OBJET OB_j qui n'est pas toujours le changement d'état événementiel constaté par EV_j .

Nous noterons $EV_i \xrightarrow{C} EV_j$ pour traduire le fait que EV_j est en dépendance chronologique conditionnelle directe avec EV_i pour les raisons de cohérence de notation présentées dans la définition 1.

• Exemples :

Sur l'exemple présenté en tête de ce paragraphe, il apparaît que

- le C-EVENEMENT EV_2 est en DCCD avec le C-EVENEMENT EV_1 puisque la C-OPERATION OP_1 de création d'une ligne de commande à livrer est déclenchée conditionnellement par EV_1 car elle n'a lieu d'être que si le stock correspondant au produit relatif à cette ligne de commande est suffisant (condition C_1). Nous noterons alors :

$$EV_1 \xrightarrow{C} EV_2$$

- de même, le C-EVENEMENT EV_3 est en DCCD avec le C-EVENEMENT EV_2 puisque la C-OPERATION OP_2 de mise à jour du stock, déclenchée inconditionnellement par EV_2 , induit un changement d'état du C-OBJET "PRODUIT" qui n'est pas toujours le changement d'état événementiel constaté par EV_3 (en effet, chaque mise à jour du stock n'entraîne pas forcément une rupture de stock). Nous noterons également :

$$EV_2 \xrightarrow{C} EV_3$$

Il apparaît donc que si EV_j est en DCCD avec EV_i , une réalisation du C-EVENEMENT EV_j peut suivre une réalisation du C-EVENEMENT EV_i (nous dirons que EV_j peut suivre EV_i) et, ainsi, la relation de dépendance chronologique conditionnelle directe peut être considérée comme une relation de dépendance fonctionnelle faible directe entre C-EVENEMENTS à laquelle elle donne une signification en termes de synchronisation (rappelons qu'une relation de dépendance fonctionnelle $A \rightarrow B$ est dite faible si et seulement si une réalisation de A entraîne zéro ou une réalisation de B)

Remarque 1 : Dépendance chronologique directe (DCD)

EV_j est dit en dépendance chronologique directe (DCD) avec EV_i si et seulement si EV_j est soit en dépendance chronologique permanente directe, soit en dépendance chronologique conditionnelle directe avec EV_i .

Nous noterons $EV_i \rightarrow EV_j$ pour traduire le fait que EV_j est en dépendance chronologique directe avec EV_i pour les raisons de cohérence de notation présentées dans la définition 1.

On aurait pu définir directement la dépendance chronologique directe en disant que $EV_i \rightarrow EV_j$ si et seulement s'il existe une C-OPERATION OP_k^i déclenchée par EV_i (conditionnellement ou non) et qui provoque un changement d'état du C-OBJET OB_j constaté par EV_j qui n'est pas forcément toujours un changement d'état événementiel.

Définition 3 : Dépendance chronologique permanente (DCP)• Définition

Nous dirons que EV_j est en dépendance chronologique permanente avec EV_i , et nous noterons $EV_i \xrightarrow{P+} EV_j$ si et seulement si EV_j est en relation avec EV_i par la fermeture transitive stricte de la relation de dépendance chronologique permanente directe :

$$(EV_i \xrightarrow{P+} EV_j) \Leftrightarrow (EV_i \xrightarrow{(-P)+} EV_j)$$

c'est-à-dire $(EV_i \xrightarrow{P+} EV_j) \Leftrightarrow (\exists n > 0) (EV_i \xrightarrow{(-P)}^n EV_j)$

$$\Leftrightarrow (EV_{k_1}, EV_{k_2}, \dots, EV_{k_{n-1}}) (EV_i \xrightarrow{P} EV_{k_1} \xrightarrow{P} EV_{k_2} \xrightarrow{P} \dots \xrightarrow{P} EV_{k_{n-1}} \xrightarrow{P} EV_j)$$

Remarque : Par définition, si EV_j est en DCPD avec EV_i , EV_j est en DCP avec EV_i :

$$(EV_i \xrightarrow{P} EV_j) \Rightarrow (EV_i \xrightarrow{P+} EV_j).$$

De plus, cette relation est évidemment transitive par définition.

Ainsi, EV_j est en dépendance chronologique permanente (DCP) avec EV_i si et seulement s'il existe une suite non vide de C-EVENEMENTS en dépendance chronologique permanente directe conduisant de EV_i à EV_j . Il apparaît ainsi que la relation de dépendance chronologique permanente n'est autre que la fermeture transitive stricte de la relation de dépendance chronologique permanente directe.

C'est pourquoi nous noterons $EV_i \xrightarrow{P+} EV_j$, par analogie avec la notation de la fermeture transitive d'une relation, pour traduire le fait que EV_j est en dépendance chronologique permanente avec EV_i .

• Exemples :

Sur l'exemple présenté en tête de ce paragraphe, il apparaît que :

- le C-EVENEMENT EV_2 est en dépendance chronologique permanente (DCP) avec le C-EVENEMENT EV_5 puisque nous avons vu, lors de l'exposé de la définition 1, que EV_2 est en dépendance chronologique permanente directe avec EV_4 , et EV_4 est en dépendance chronologique permanente directe avec EV_5 . Nous noterons donc :

$$EV_5 \xrightarrow{P+} EV_2$$

De façon formelle :

$$EV_5 \xrightarrow{P+} EV_2 \text{ car } (\exists EV_4) (EV_5 \xrightarrow{P} EV_4 \xrightarrow{P} EV_2)$$

Il apparaît donc que si EV_j est en DCP avec EV_i , une réalisation du C-EVENEMENT EV_j suit toujours une réalisation du C-EVENEMENT EV_i , à la suite de la séquence des réalisations des C-EVENEMENTS en DCPD menant de EV_i à EV_j . Ainsi, la relation de dépendance chronologique permanente apparaît comme une relation de dépendance fonctionnelle forte entre C-EVENEMENTS, à laquelle elle donne une signification en termes de synchronisation (mais non directe puisqu'obtenue par transitivité).

Définition 4 : Dépendance chronologique conditionnelle (DCC)• Définition

Nous dirons que EV_j est en dépendance chronologique conditionnelle avec EV_i , et nous noterons $EV_i \xrightarrow{C+} EV_j$ si et seulement si :

$$(1) \quad (\exists EV_{k_1}, EV_{k_2}, \dots, EV_{k_n}) (EV_i \xrightarrow{P} EV_{k_1} \xrightarrow{P} EV_{k_2} \xrightarrow{P} \dots \xrightarrow{P} EV_{k_n} \xrightarrow{P} EV_j)$$

$$(2) \quad (\exists k_s \in \{i, k_1, \dots, k_n\}) (EV_{k_s} \xrightarrow{C} EV_{k_{s+1}}) [k_{s+1} = j \text{ si } s = n]$$

(1) traduit le fait qu'il existe une suite de dépendances chronologiques directes conduisant de EV_i à EV_j .

(2) exprime qu'au moins une de ces dépendances chronologiques directes est conditionnelle.

Remarque : Par définition, si EV_j est en DCCD avec EV_i , EV_j est en DCC avec EV_i :

$$(EV_i \xrightarrow{C} EV_j) \Rightarrow (EV_i \xrightarrow{C+} EV_j).$$

De plus, cette relation est évidemment transitive.

Ainsi, EV_j est en dépendance chronologique conditionnelle (DCC) avec EV_i si et seulement s'il existe une suite non vide de C-EVENEMENTS en dépendance chronologique directe conduisant de EV_i à EV_j telle qu'au moins une des dépendances chronologiques directes de cette suite soit conditionnelle.

Par abus de langage, nous noterons cette relation $EV_i \xrightarrow{C+} EV_j$ bien qu'il ne s'agisse pas de la fermeture transitive stricte de la relation de dépendance chronologique conditionnelle directe.

• Exemples :

Sur l'exemple présenté en tête de ce paragraphe, il apparaît que :

- le C-EVENEMENT EV3 est en dépendance chronologique conditionnelle avec le C-EVENEMENT EV1 car nous avons vu, lors de l'exposé de la définition 2 que EV2 est en dépendance chronologique conditionnelle directe avec EV1 et EV3 est en dépendance chronologique conditionnelle directe avec EV2. Nous noterons donc :

$$EV1 \xrightarrow{C+} EV3 \text{ car } (\exists EV2)(EV1 \xrightarrow{C} EV2 \xrightarrow{C} EV3)$$

- de même, le C-EVENEMENT EV3 est en dépendance chronologique conditionnelle avec le C-EVENEMENT EV4 car nous avons vu, lors de l'exposé de la définition 1 que EV2 est en dépendance chronologique permanente directe avec EV4 et, lors de l'exposé de la définition 2, nous avons également vu que EV3 est en dépendance chronologique conditionnelle directe avec EV2. Nous noterons donc :

$$EV4 \xrightarrow{C+} EV3 \text{ car } (\exists EV2)(EV4 \xrightarrow{P} EV2 \xrightarrow{C} EV3)$$

Il apparaît donc que si EV_j est en DCC avec EV_i , une réalisation du C-EVENEMENT EV_j peut suivre une réalisation du C-EVENEMENT EV_i dans le cas où la séquence des réalisations des C-EVENEMENTS menant de EV_i à EV_j a été complète. Ainsi, la relation de dépendance chronologique conditionnelle apparaît comme une relation de dépendance fonctionnelle faible entre C-EVENEMENTS à laquelle elle donne une signification en termes de synchronisation (mais non directe puisqu'obtenue par transitivité).

Remarque 2 : Dépendance chronologique (DC)

EV_j est dit en dépendance chronologique (DC) avec EV_i si et seulement si EV_j est soit en dépendance chronologique permanente, soit en dépendance chronologique conditionnelle avec EV_i .

On aurait pu définir directement la dépendance chronologique en disant que EV_j est en dépendance chronologique avec EV_i si et seulement s'il existe une suite non vide de C-EVENEMENTS en dépendance chronologique directe conduisant de EV_i à EV_j . Ainsi, il apparaît que la relation de dépendance chronologique est la fermeture transitive stricte de la relation de dépendance chronologique directe. C'est pourquoi nous noterons $EV_i \xrightarrow{+} EV_j$ pour traduire le fait que EV_j est en dépendance chronologique avec EV_i .

$$\begin{aligned} (EV_i \xrightarrow{+} EV_j) &\Leftrightarrow (\exists n > 0)(EV_i \xrightarrow{+} EV_j)^n \\ &\Leftrightarrow (\exists EV_{k_1}, \dots, EV_{k_{n-1}})(EV_i \xrightarrow{+} EV_{k_1} \xrightarrow{+} EV_{k_2} \xrightarrow{+} \\ &\dots \xrightarrow{+} EV_{k_{n-1}} \xrightarrow{+} EV_j) \end{aligned}$$

Définition 5 : Indépendance chronologique (IC) - Indépendance chronologique mutuelle (ICM)

• Définition

α) Nous dirons que EV_j est en indépendance chronologique avec EV_i , et nous noterons $EV_i \xrightarrow{I} EV_j$, si et seulement si :

$$\begin{aligned} (EV_i \xrightarrow{I} EV_j) &\Leftrightarrow \neg(EV_i \xrightarrow{+} EV_j) \\ &\Leftrightarrow (\neg(EV_i \xrightarrow{P+} EV_j) \wedge \neg(EV_i \xrightarrow{C+} EV_j)) \end{aligned}$$

β) De même, nous dirons que EV_i et EV_j sont en indépendance chronologique mutuelle, et nous noterons $EV_i \xrightarrow{IM} EV_j$ ou $EV_j \xrightarrow{IM} EV_i$ indifféremment, si et seulement si :

$$(EV_i \xrightarrow{IM} EV_j) \Leftrightarrow (EV_j \xrightarrow{IM} EV_i) \Leftrightarrow ((EV_i \xrightarrow{I} EV_j) \wedge (EV_j \xrightarrow{I} EV_i))$$

Ainsi,

α) EV_j est en indépendance chronologique (IC) avec EV_i si et seulement si EV_j n'est ni en dépendance chronologique permanente, ni en dépendance chronologique conditionnelle avec EV_i .

Nous noterons $EV_i \xrightarrow{I} EV_j$ pour traduire le fait que EV_j est en indépendance chronologique pour les raisons de cohérence de notation avec les relations donctionnelles présentées dans la définition 1.

Il convient également de noter que la relation d'indépendance chronologique est en général non symétrique. En effet, si EV_j n'est pas en dépendance chronologique avec EV_i , ceci n'implique pas que EV_i ne soit pas en dépendance chronologique avec EV_j .

β) De plus, si EV_j est en indépendance chronologique avec EV_i , et si EV_i est en indépendance chronologique avec EV_j , nous dirons que EV_i et EV_j sont en indépendance chronologique mutuelle (ICM), ce que nous noterons indifféremment par $EV_i \xrightarrow{IM} EV_j$ ou $EV_j \xrightarrow{IM} EV_i$.

Exemples

Sur l'exemple présenté en tête de ce paragraphe, il apparaît que :

- le C-EVENEMENT EV1 est en indépendance chronologique avec le C-EVENEMENT EV3 puisque EV1 n'est pas en dépendance chronologique avec EV3 (l'arrivée d'une ligne de commande à traiter est indépendante de toute rupture de stock). Nous noterons alors :

$$EV3 \xrightarrow{I} EV1$$

- de même, le C-EVENEMENT EV4 est en indépendance chronologique avec le C-EVENEMENT EV2 puisque EV4 n'est pas en dépendance chronologique avec EV2 (la suppression de rupture de stock est indépendante de l'arrivée d'une ligne de commande à livrer). Nous noterons alors :

$$EV2 \xrightarrow{I} EV4$$

- le C-EVENEMENT EV1 et le C-EVENEMENT EV4 sont en indépendance chronologique mutuelle puisque EV1 est en indépendance chronologique avec EV4 et EV4 est en indépendance chronologique avec EV1 (l'arrivée d'une ligne de commande à traiter et la suppression de rupture de stock sont deux phénomènes mutuellement indépendants). Nous noterons alors :

$$EV4 \xrightarrow{IM} EV1 \text{ ou } EV1 \xrightarrow{IM} EV4$$

- de même, le C-EVENEMENT EV1 et le C-EVENEMENT EV5 sont en indépendance chronologique mutuelle puisque EV1 est en indépendance chronologique avec EV5 et EV5 est en indépendance chronologique avec EV1 (l'arrivée d'une ligne de commande à traiter et l'arrivée d'un produit réapprovisionné sont deux phénomènes mutuellement indépendants). Nous noterons alors :

$$EV1 \xrightarrow{IM} EV5 \text{ ou } EV5 \xrightarrow{IM} EV1$$

II-2.2.2.- Dépendances sémantiques entre C-EVENEMENTS

Le problème représenté sous forme de schéma conceptuel a des frontières. A l'en-droit frontière, les liens de causalité qui peuvent exister avec d'autres parties du problème sont décrits par des contraintes d'intégrité. On pourrait se dispenser de ces contraintes d'intégrité si l'on décrivait tout l'univers qui sert d'environnement au problème.

Ainsi, dans le cadre de l'exemple du traitement d'une commande acceptée, il existe une contrainte d'intégrité précisant que toute occurrence du C-OBJET OB6

"produit réapprovisionné" doit correspondre à une occurrence du C-OBJET OB5 "bon de réapprovisionnement". Ceci traduit le fait qu'une opération extérieure au système a provoqué la création de l'occurrence de OB6, et que cette opération est la conséquence de la création d'une occurrence de OB5. Il convient également de remarquer que cette opération, conduisant à la création de OB6 peut ou non avoir lieu : nous n'avons aucun moyen de contrôle sur elle. Il va de soi que si l'on avait pu décrire l'enchaînement des opérations conduisant de l'émission d'un bon de réapprovisionnement à un produit réapprovisionné, cette contrainte n'aurait pas lieu d'exister.

Les relations de dépendance sémantique entre C-EVENEMENTS sont déduites des contraintes d'intégrité de ce type. Elles expriment un lien de dépendance entre deux C-EVENEMENTS bien que ces deux C-EVENEMENTS n'aient aucun lien du point de vue des dépendances fonctionnelles déduites du sous-schéma dynamique. Ainsi, le C-EVENEMENT "arrivée d'une ligne de commande" ne peut survenir que si le C-EVENEMENT "arrivée de la commande" est lui-même survenu. Ceci traduit un lien sémantique entre ces deux C-EVENEMENTS, bien qu'ils n'aient à proprement parler aucun lien du point de vue de la dynamique causale du système). Ces relations permettent d'assurer la cohérence sémantique lors du fonctionnement du système (par exemple, il serait absurde de prendre en compte un événement du type "arrivée d'une ligne de commande" alors que l'événement "arrivée de la commande" correspondant ne serait pas apparu dans le système). Elles sont donc nécessaires non pas à la prise en compte des problèmes de synchronisation mais au contrôle du fonctionnement du système. Comme elles traduisent des liens entre C-EVENEMENTS qui ne sont pas directement liés à l'aspect dynamique du schéma conceptuel, elles seront obtenues par l'analyse du sous-schéma de données (sous-schéma statique), et plus particulièrement par l'étude des contraintes d'intégrités sur les C-OBJETS associées à ce sous-schéma [12] et par l'étude des dépendances fonctionnelles entre C-OBJETS et C-CLASSES et entre C-CLASSES [39], [59], comme nous allons le montrer ci-dessous.

Définition 6 : Dépendance sémantique (DS)

Soit OB_k le C-OBJET dont le changement d'état est provoqué par une C-OPERATION OP_k^i déclenchée par EV_i , et soit \mathcal{C} l'ensemble des C-CLASSES du schéma statique.

Rappelons que le C-EVENEMENT EV_i (respectivement EV_j) constate un changement d'état événementiel du C-OBJET OB_i (respectivement OB_j).

Nous dirons que EV_j est en dépendance sémantique (DS) avec EV_i , et nous noterons $EV_i \xrightarrow{DS} EV_j$ si et seulement s'il existe une contrainte d'intégrité subordonnant toute occurrence de OB_j à une occurrence de OB_k .

Cette contrainte d'intégrité, si elle existe, traduit une dépendance fonctionnelle entre C-OBJETS, c'est-à-dire qu'elle n'a de sens que si :

- (i) les C-OBJETS concernés sont en dépendance fonctionnelle, mais appartiennent à la même C-CLASSE.
- (ii) les C-OBJETS concernés appartiennent à des C-CLASSES en dépendance fonctionnelle.

Ainsi, EV_j est en dépendance sémantique avec EV_i , et nous noterons $EV_i \xrightarrow{DS} EV_j$ si et seulement si

- (i) $(\exists OP_k^i \in OP)((EV_i \rightarrow OP_k^i) \wedge (OP_k^i \rightarrow OB_k))$
- (ii) $[(\exists C_k \in \mathcal{C})((OB_k, OB_j \in C_k) \wedge (OB_k \rightarrow OB_j))] \vee [(\exists C_k, C_j \in \mathcal{C})((C_k + C_j) \wedge (OB_k \in C_k, OB_j \in C_j) \wedge (C_k \rightarrow C_j))]$

Exemples

Sur l'exemple du système de vente par correspondance présenté au paragraphe II-1.2.2., il apparaît que :

- le C-EVENEMENT EV6 "réapprovisionnement" est en dépendance sémantique avec le C-EVENEMENT EV4 "rupture de stock" car il existe une dépendance fonctionnelle du C-OBJET OB9 (produit réapprovisionné) auquel est associé EV6 vis-à-vis du C-OBJET OB8 (ordre de réapprovisionnement) créé par la C-OPERATION OP7 déclenchée par EV4 (aspect (i) de la définition). Nous noterons :

$$EV4 \xrightarrow{DS} EV6$$

- le C-EVENEMENT EV1 "arrivée d'une ligne de commande" est en dépendance sémantique avec le C-EVENEMENT EV9 "arrivée d'une commande" car il existe une dépendance fonctionnelle (1,n) de la C-CLASSE "ligne de commande" à laquelle appartient le C-OBJET OB1 (ligne de commande initiale) correspondant à EV1 envers la C-CLASSE "commande" à laquelle appartient le C-OBJET OB0 ("commande initiale") correspondant à EV9 (aspect (ii) de la définition). Nous noterons :

$$EV9 \xrightarrow{DS} EV1$$

Définition 7 : Indépendance sémantique (IS) - Indépendance sémantique mutuelle (ISM)

• Définition

- α) Nous dirons que EV_j est en indépendance sémantique (IS) avec EV_i , et nous

noterons $EV_i \xrightarrow{IS} EV_j$ si et seulement si :

- (i) EV_i et EV_j sont en indépendance chronologique mutuelle
- (ii) EV_j n'est pas en dépendance sémantique avec EV_i :

$$(EV_i \xrightarrow{IS} EV_j) \Leftrightarrow ((EV_i \xrightarrow{IM} EV_j) \wedge (\neg EV_i \xrightarrow{DS} EV_j))$$

Notons de plus que la relation d'indépendance sémantique est en général non symétrique. En effet, si EV_j n'est pas en dépendance sémantique avec EV_i , ceci n'implique pas que EV_i ne soit pas en dépendance sémantique avec EV_j .

β) De plus, EV_j est en indépendance sémantique mutuelle (ISM) avec EV_i , et nous noterons indifféremment $EV_i \xrightarrow{ISM} EV_j$ ou $EV_j \xrightarrow{ISM} EV_i$ si et seulement si EV_j est en indépendance sémantique avec EV_i et EV_i est en indépendance sémantique avec EV_j :

$$(EV_i \xrightarrow{ISM} EV_j) \Leftrightarrow (EV_j \xrightarrow{ISM} EV_i) \Leftrightarrow ((EV_i \xrightarrow{IS} EV_j) \wedge (EV_j \xrightarrow{IS} EV_i))$$

• Exemples

Si l'on considère le schéma conceptuel du système de vente par correspondance présenté au paragraphe II-1.2.2., il apparaît que :

- le C-EVENEMENT EV9 est en indépendance sémantique avec le C-EVENEMENT EV1 car il n'y a pas de dépendance sémantique de EV9 envers EV1. En effet, la partie (ii) de la définition de la dépendance sémantique ne s'applique pas dans ce cas, et il n'existe pas de dépendance fonctionnelle de la C-CLASSE "commande" envers la C-CLASSE "ligne de commande", donc la partie (i) ne s'applique pas non plus. Nous noterons donc :

$$EV1 \xrightarrow{IS} EV9$$

- Pour cet exemple, il n'existe pas de C-EVENEMENTS en indépendance sémantique mutuelle.

II-2.2.3.- Graphe des dépendances chronologiques entre C-EVENEMENTS

• Définition

Le graphe des dépendances chronologiques G_c entre C-EVENEMENTS est défini comme étant le graphe des dépendances fonctionnelles [10] entre ces C-EVENEMENTS (car nous avons vu au paragraphe II-2.2.1. que ces relations chronologiques étaient une interprétation en termes de synchronisation des relations de dépendance fonctionnelles fortes et faibles entre C-EVENEMENTS). Pour rendre compte de la typologie de

ces relations, nous labellerons les arcs de ce graphe par le symbole choisi pour préciser la nature des dépendances chronologiques.

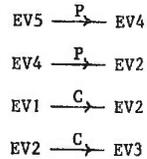
• Utilisation

Le graphe des dépendances chronologiques entre C-EVENEMENTS, traduisant de manière synthétique la dynamique du système, sera plus particulièrement utile pour contrôler la qualité du sous-schéma dynamique (détection des trappes, des circuits, comme nous le montrerons dans le chapitre III de ce travail.

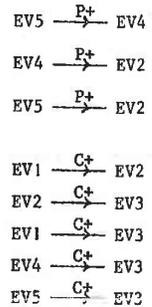
• Exemple

Nous allons présenter le graphe des dépendances chronologiques entre C-EVENEMENTS correspondant au schéma conceptuel du traitement d'une commande acceptée, présenté au début du paragraphe II-2.2.1. :

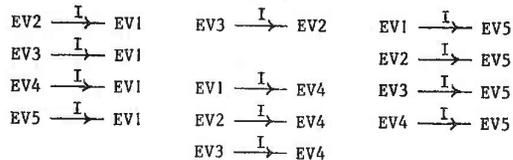
- les dépendances chronologiques directes sont les suivantes :



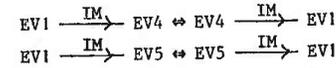
• Ceci entraîne les dépendances chronologiques suivantes :



• D'où les relations d'indépendance chronologiques suivantes :



• et les relations d'indépendances chronologiques mutuelle suivantes :



• Pour des raisons de commodité de lecture, nous regrouperons ces relations dans un tableau à deux dimensions, où le vecteur colonne M_j exprime les relations de EV_j avec les autres C-EVENEMENTS du système : à l'intersection de la ligne i et la colonne j , on trouve la caractérisation de la relation chronologique $EV_i \rightarrow EV_j$.

| | EV1 | EV2 | EV3 | EV4 | EV5 |
|-----|-----|-----|-----|-----|-----|
| EV1 | | C | C+ | IM | IM |
| EV2 | I | | C | I | I |
| EV3 | I | I | | I | I |
| EV4 | IM | P | C+ | | I |
| EV5 | IM | P+ | C+ | P | |

Figure II-8

• Le graphe des dépendances chronologiques est le suivant, compte tenu des conventions graphiques traduisant le label des arcs du graphe :

- \longrightarrow traduit la relation de dépendance chronologique permanente directe (DCPD)
- \dashrightarrow traduit la relation de dépendance chronologique permanente (DCP)
- \longrightarrow traduit la relation de dépendance chronologique conditionnelle directe (DCCD)
- \dashrightarrow traduit la relation de dépendance chronologique conditionnelle (DCC)

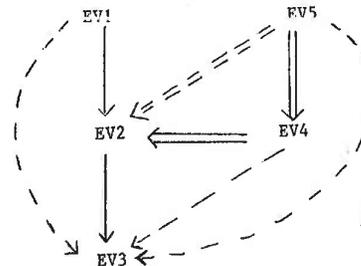


Figure II-9

II-2.2.4.- Conclusion sur les relations temporelles entre C-EVENEMENTS

Nous venons de montrer que la définition conceptuelle d'un système inclut la définition de toutes les relations temporelles entre C-EVENEMENTS. On peut schématiser ces relations sur un graphe. Notre propos est de montrer que cet ensemble de relations (ou le graphe qui les représente) permet de déduire l'ensemble des dépendances temporelles entre processus (c'est-à-dire entre C-OPERATIONS) ainsi que le graphe qui les schématise.

Comme il existe trois types de relations de dépendance chronologique entre C-EVENEMENTS (dépendance permanente, dépendance conditionnelle, indépendance), et comme les C-OPERATIONS sont déclenchées par ces C-EVENEMENTS, il existe trois types de relations de dépendance entre C-OPERATIONS, traduisant ce que nous considérons comme les trois types de situations de synchronisation de base, et que nous allons présenter dans le paragraphe suivant.

II-2.3.- Typologie des relations entre C-OPERATIONS : situations de synchronisation de base

II-2.3.1.- Typologie des situations de synchronisation : dépendances entre C-OPERATIONS

Comme nous l'avons vu au chapitre I de ce travail, l'idée de base concernant la résolution d'un problème de synchronisation consiste à déterminer quelles sont les types de situations qui permettent de lier deux processus (deux C-OPERATIONS dans notre cas), et qui permettent ainsi de définir un ordre sur l'exécution des opérations [13], [25]. Comme toute C-OPERATION est nécessairement déclenchée par un C-EVENEMENT, les relations définissant la dépendance ou l'indépendance des C-OPERATIONS sont conditionnées par les relations de dépendance chronologique entre les C-EVENEMENTS qui les déclenchent.

Ainsi, il y a trois types de situations pouvant lier deux C-OPERATIONS. Ils correspondent aux trois cas de dépendance chronologique entre C-EVENEMENTS (indépendance, dépendance permanente, dépendance chronologique). Ils définissent les trois situations de synchronisation de base de notre modèle.

Notations :

- Soient OP_i , OP_j deux C-OPERATIONS et EV_i , EV_j deux C-EVENEMENTS.
- EV_i déclenche OP_i : $EV_i \rightarrow OP_i$
- EV_j déclenche OP_j : $EV_j \rightarrow OP_j$

• Dans le cas où EV_j est en dépendance chronologique avec EV_i ($EV_i \xrightarrow{+} EV_j$), OP_i provoque un changement d'état du C-OBJET OB_k qui est constaté soit directement par EV_j , dans le cas où $EV_i \rightarrow EV_j$ (EV_j est en dépendance chronologique directe avec EV_i), soit par EV_{k_1} , premier C-EVENEMENT de la suite de dépendances chronologiques directes menant de EV_i à EV_j , dans le cas où $EV_i \xrightarrow{+} EV_j$ (EV_j est en dépendance chronologique non directe avec EV_i) :

$$(EV_i \xrightarrow{+} EV_j) \Rightarrow [(OP_i \rightarrow OB_k) \wedge ((EV_j \rightarrow OB_k) \vee (\exists EV_{k_1} \in EV) ((EV_i \rightarrow EV_{k_1}) \wedge (EV_{k_1} \xrightarrow{+} EV_j) \wedge (EV_{k_1} \rightarrow OB_k)))]$$

Situation de synchronisation I : dépendance permanente

• Définition

Si et seulement si EV_j est en dépendance chronologique permanente avec EV_i , la C-OPERATION OP_j est dite alors en dépendance permanente avec la C-OPERATION OP_i , et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{P+} OP_j$ (condition nécessaire et suffisante)

$$(EV_i \xrightarrow{P+} EV_j) \Leftrightarrow (OP_i \xrightarrow{P+} OP_j)$$

Par définition, cette relation est de plus transitive.

Sémantique

Ceci signifie que si OP_j est en dépendance permanente avec OP_i , le déclenchement par définition toujours suivi de l'exécution dans ce cas de toute réalisation de OP_i sera toujours suivie du déclenchement d'une réalisation de OP_j par une occurrence de EV_j après exécution de la séquence des réalisations des C-OPERATIONS associées aux C-EVENEMENTS de la séquence de dépendances permanentes directes menant de EV_i à EV_j . Cette réalisation de OP_j sera alors exécutée si les éventuelles conditions d'exécution qui lui sont associées sont satisfaites. Ceci correspond à une situation de synchronisation traduisant le fait qu'un processus est toujours pris en compte après un autre, à l'issue d'une séquence d'activations inconditionnelles d'autres processus.

Il convient de plus de remarquer que le déclenchement d'une opération (c'est-à-dire d'une réalisation de C-OPERATION - voir paragraphe II-1.1.2.2.) est différent de son exécution : le déclenchement d'une opération permet son exécution si et seulement si les conditions d'exécution de cette opération sont satisfaites.

Sur l'exemple présenté au paragraphe II-2.2.1., il apparaît que

$$(EV5 \xrightarrow{P+} EV2) \Leftrightarrow (OP6 \xrightarrow{P+} OP3)$$

En effet, l'exécution de toute réalisation de la C-OPERATION OP6 "mise à jour après réapprovisionnement" sera toujours suivie par le déclenchement d'une réalisation de la C-OPERATION OP3, par une occurrence de EV2 "mise à jour du stock après livraison" et son exécution, car aucune condition d'exécution ne lui est associée, et ce après l'exécution de la réalisation de la C-OPERATION OP5 "prise en compte des lignes de commande différées" correspondante.

Cas particulier : dépendance permanente directe

• Définition

Si et seulement si EV_j est en dépendance chronologique permanente directe avec EV_i , la C-OPERATION OP_j est dite en dépendance permanente directe avec la C-OPERATION OP_i , et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{P} OP_j$ (condition nécessaire et suffisante)

$$(EV_i \xrightarrow{P} EV_j) \Leftrightarrow (OP_i \xrightarrow{P} OP_j)$$

• Sémantique

Ceci signifie que si OP_j est en dépendance permanente directe avec OP_i , le déclenchement par définition toujours suivi de l'exécution dans ce cas, de toute réalisation de OP_i sera toujours directement suivie du déclenchement d'une réalisation de OP_j par une occurrence de EV_j . Cette réalisation de OP_j sera alors exécutée si les éventuelles conditions d'exécution qui lui sont associées sont satisfaites. Ceci correspond à une situation de synchronisation traduisant le fait qu'un processus est systématiquement pris en compte immédiatement après la fin d'exécution d'un autre processus donné.

• Exemple

Sur l'exemple présenté au paragraphe II-2.2.1., il apparaît que :

$$(EV5 \xrightarrow{P} EV4) \Leftrightarrow (OP6 \xrightarrow{P} OP5)$$

En effet, l'exécution de toute réalisation de la C-OPERATION OP6 "mise à jour après réapprovisionnement" est toujours directement suivie du déclenchement d'une réalisation de la C-OPERATION OP5 "prise en compte des lignes de commande différées" par une occurrence de EV4.

Cette réalisation de OP5 sera exécutée si les éventuelles conditions d'exécution qui lui sont associées sont satisfaites.

Situation de synchronisation 2 : dépendance conditionnelle

• Définition

Si et seulement si EV_j est en dépendance chronologique conditionnelle avec EV_i , la C-OPERATION OP_j est dite alors en dépendance conditionnelle avec la C-OPERATION OP_i , et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{C+} OP_j$. (Condition nécessaire et suffisante).

$$(EV_i \xrightarrow{C+} EV_j) \Leftrightarrow (OP_i \xrightarrow{C+} OP_j)$$

Par définition, cette relation est de plus transitive.

• Sémantique

Ceci signifie que si OP_j est en dépendance conditionnelle avec OP_i , le déclenchement de toute réalisation de OP_i pourra être suivi par le déclenchement d'une réalisation de OP_j par une occurrence de EV_j si la séquence des réalisations des C-OPERATIONS associées aux C-EVENEMENTS de la séquence de dépendances chronologiques directes non toutes permanentes menant de EV_i à EV_j a été entièrement exécutée et si la réalisation de la dernière C-OPERATION de cette séquence a induit un changement d'état constituant une occurrence de EV_j (déclenchant ainsi la réalisation de OP_j). Cette réalisation de OP_j sera exécutée si les éventuelles conditions d'exécution qui lui sont associées sont satisfaites. Ceci correspond à une situation de synchronisation traduisant le fait qu'un processus puisse être pris en compte à la suite d'un processus donné, après exécution complète d'une chaîne de processus et si le système est dans un certain état (l'événement EV_j existe).

• Exemple

Sur l'exemple du paragraphe II-2.2.1., il apparaît que :

$$(EV1 \xrightarrow{C+} EV3) \Leftrightarrow (OP1 \xrightarrow{C+} OP4)$$

En effet le déclenchement de toute réalisation de la C-OPERATION OP1 "création d'une ligne de commande à livrer" pourra être suivi par le déclenchement d'une réalisation de la C-OPERATION OP4 "émission d'un bon de réapprovisionnement" et par son exécution car il n'y a pas de condition d'exécution associée à cette C-OPERATION si les réalisations des C-OPERATIONS OP1 et OP3 "mise à jour du stock après livraison" ont toutes deux été exécutées : la réalisation de OP1 sera exécutée si la condition CI est vérifiée, celle de OP3 sera déclenchée s'il existe une occurrence de EV2,

c'est-à-dire si la réalisation de OP1 a été exécutée, induisant un changement d'état de OB2 constituant une occurrence de EV2. Enfin, OP4 sera déclenchée par une occurrence de EV3 et exécutée si et seulement si l'exécution de la réalisation de OP3 a induit un changement d'état de OB4 "produit en stock" qui constitue une rupture de stock, donc une occurrence de EV3.

Cas particulier : dépendance conditionnelle directe

• Définition

Si et seulement si EV_j est en dépendance chronologique conditionnelle directe avec EV_i , la C-OPERATION OP_j est dite en dépendance conditionnelle directe avec la C-OPERATION OP_i , et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{C} OP_j$. (Condition nécessaire et suffisante).

$$(EV_i \xrightarrow{C} EV_j) \Leftrightarrow (OP_i \xrightarrow{C} OP_j)$$

• Sémantique

Ceci signifie que, si OP_j est en dépendance conditionnelle directe avec OP_i , le déclenchement de toute réalisation de OP_i pourra être directement suivi par le déclenchement d'une réalisation de OP_j par une occurrence de EV_j si la réalisation de OP_i a été exécutée (les conditions d'exécutions ont été satisfaites) et si le changement d'état que cette réalisation a induit constitue une occurrence de EV_j . Ce déclenchement quand il existe, peut être suivi par l'exécution de cette réalisation de OP_j si les éventuelles conditions d'exécution qui lui sont associées sont satisfaites. Ceci correspond à une situation de synchronisation traduisant le fait qu'un processus puisse être pris en compte immédiatement après un processus donné si celui-ci a conduit le système dans un état donné.

• Exemple

Sur l'exemple du paragraphe II-2.2.1., il apparaît que

$$(EV2 \xrightarrow{C} EV3) \Leftrightarrow (OP3 \xrightarrow{C} OP4)$$

En effet, le déclenchement de toute réalisation de la C-OPERATION OP3 "mise à jour du stock après livraison" est toujours suivi de son exécution. Celle-ci pourra être immédiatement suivie par le déclenchement d'une réalisation de OP4 "émission d'un bon de réapprovisionnement" si le changement d'état de l'occurrence de OB4 "produit en stock" induit par la réalisation de OP3 constitue une occurrence de EV3.

Le déclenchement de cette réalisation de OP4, quand il existe, est toujours suivi par son exécution, car il n'existe pas de condition d'exécution associées à OP4.

Situation de synchronisation 3 : Indépendance

Si EV_j est en indépendance chronologique avec EV_i , la C-OPERATION OP_j est indépendante de la C-OPERATION OP_i , et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{I} OP_j$.

$$(EV_i \xrightarrow{I} OP_j) \Leftrightarrow (OP_i \xrightarrow{I} OP_j)$$

Remarquons que cette condition est suffisante, mais non nécessaire. En effet, une C-OPERATION OP_j déclenchée par EV_j peut être indépendante d'une C-OPERATION OP_i déclenchée par EV_i même si EV_j est en dépendance chronologique avec EV_i . Comme un C-EVENEMENT peut déclencher plusieurs C-OPERATIONS, OP_j n'est pas forcément dépendante de OP_i puisque EV_i peut très bien déclencher une C-OPERATION OP_k provoquant le changement d'état d'un C-OBJET OB_k constaté par EV_j et, dans ce cas, les C-OPERATIONS déclenchées par EV_j sont dépendantes de OP_k et non de OP_j (si celle-ci ne modifie pas le C-OBJET OB_k). Le raisonnement est le même dans le cas d'une dépendance chronologique non directe de EV_j par rapport à EV_i .

Aussi, la C-OPERATION OP_j est indépendante de la C-OPERATION OP_i , et nous noterons $OP_i \xrightarrow{I} OP_j$, si et seulement si OP_j n'est ni en dépendance permanente, ni en dépendance conditionnelle avec OP_i :

$$(OP_i \xrightarrow{I} OP_j) \Leftrightarrow ((OP_j \xrightarrow{P+} OP_i) \wedge \neg(OP_i \xrightarrow{C+} OP_j)) \\ \Leftrightarrow \neg((OP_j \xrightarrow{P+} OP_i) \vee (OP_i \xrightarrow{C+} OP_j))$$

• Sémantique

Ceci signifie que, si la C-OPERATION OP_j est indépendante de la C-OPERATION OP_i , toute réalisation de OP_j peut être déclenchée indépendamment de toute réalisation de OP_i , c'est-à-dire que le déclenchement de toute réalisation de OP_i n'a aucune influence sur le déclenchement de toute réalisation de OP_j .

Ceci correspond à une situation de synchronisation traduisant l'indépendance d'un processus par rapport à un autre.

• Exemples

Sur l'exemple du paragraphe II-2.2.1., il apparaît que :

$$- (EV3 \xrightarrow{I} EV1) \Rightarrow (OP4 \xrightarrow{I} OP1) \wedge (OP4 \xrightarrow{I} OP2)$$

En effet, le déclenchement et l'exécution de toute réalisation de la C-OPERATION OP4 "émission d'un bon de réapprovisionnement" n'a aucune influence sur le déclenchement de la C-OPERATION OP1 "création d'une ligne de commande à livrer", et sur la C-OPERATION OP2 "création d'une ligne de commande différée".

$$- (EV3 \xrightarrow{I} EV5) \Rightarrow (OP4 \xrightarrow{I} OP6)$$

En effet, le déclenchement et l'exécution de toute réalisation de OP4 "émission d'un bon de réapprovisionnement" n'ont du point de vue de la dynamique du problème, aucune influence sur le déclenchement de toute réalisation de OP6 "mise à jour du stock". Néanmoins, il existe une dépendance sémantique $EV3 \xrightarrow{IS} EV5$ traduisant le lien sémantique après réapprovisionnement : "à tout bon de réapprovisionnement doit correspondre un produit réapprovisionné". Mais ceci exprime un contrôle à effectuer lors du fonctionnement du système, afin de garantir sa cohérence, et non un lien causal induisant une chronologie sur EV3 et EV5.

1er cas particulier : Indépendance mutuelle

• Définition

Si EV_i et EV_j sont en indépendance chronologique mutuelle, les C-OPERATIONS OP_i et OP_j sont mutuellement indépendantes, et nous conserverons la même notation que dans le cas des relations chronologiques entre C-EVENEMENTS : $OP_i \xrightarrow{IM} OP_j$, ou indifféremment, $OP_j \xrightarrow{IM} OP_i$

$$(EV_i \xrightarrow{IM} EV_j) \Rightarrow (OP_i \xrightarrow{IM} OP_j)$$

Pour les mêmes raisons que dans le cas de l'indépendance de deux C-OPERATIONS, cette condition est suffisante, mais non nécessaire.

Ainsi, si OP_j est indépendante de OP_i , et si OP_i est indépendante de OP_j , OP_i et OP_j seront dites mutuellement indépendantes. De même, si OP_i et OP_j sont déclenchées par le même C-EVENEMENT EV_k , alors elles sont mutuellement indépendantes (donc indépendantes) car il n'existe aucun lieu de nature causale entre elles (elles n'ont aucune influence l'une sur l'autre). Nous noterons indifféremment $OP_i \xrightarrow{IM} OP_j$ ou $OP_j \xrightarrow{IM} OP_i$.

$$(OP_i \xrightarrow{IM} OP_j) \Leftrightarrow [((OP_i \xrightarrow{I} OP_j) \wedge (OP_j \xrightarrow{I} OP_i)) \vee ((\exists EV_k \in EV)$$

$$(EV_k \rightarrow OP_i \wedge EV_k \rightarrow OP_j))]$$

• Sémantique

Ceci signifie que, si OP_i et OP_j sont en indépendance mutuelle, le déclenchement de toute réalisation de l'une n'a aucune influence sur le déclenchement de toute réalisation de l'autre : elles n'ont aucune influence mutuelle et leurs réalisations respectives peuvent être déclenchées indépendamment l'une de l'autre. Ceci correspond à une situation de synchronisation traduisant l'indépendance mutuelle de deux processus.

• Exemple

Sur l'exemple présenté au paragraphe II-2.2.1., il apparaît que :

$$- (EV1 \xrightarrow{IM} EV5) \Rightarrow (OP1 \xrightarrow{IM} OP6)$$

En effet, l'exécution de toute réalisation de la C-OPERATION OP1 "création d'une ligne de commande à livrer" (respectivement de la C-OPERATION OP6 "mise à jour du stock après réapprovisionnement") n'a aucune influence sur le déclenchement de la C-OPERATION OP6 (respectivement de la C-OPERATION OP1).

II-2.3.2.- Graphe des dépendances entre C-OPERATIONS

Définition

De même que pour le graphe des dépendances chronologiques entre C-EVENEMENTS, le graphe des dépendances entre C-OPERATIONS est défini comme le graphe des dépendances fonctionnelles [10] entre ces C-OPERATIONS (car de même que dans le cas des C-EVENEMENTS, les dépendances entre C-OPERATIONS sont une interprétation des relations de dépendances fonctionnelles fortes et faibles entre C-OPERATIONS). De la même manière que dans le cas des C-EVENEMENTS, nous labellerons les arcs de ce graphe afin de préciser la nature des dépendances entre C-OPERATIONS.

Le graphe des dépendances entre C-OPERATIONS se déduit du graphe des dépendances chronologiques entre C-EVENEMENTS, puisque, sous les hypothèses présentées au paragraphe II-2.3.1., nous avons les relations suivantes :

$$(EV_i \xrightarrow{F+} EV_j) \Leftrightarrow (OP_i \xrightarrow{F+} OP_j)$$

$$(EV_i \xrightarrow{P} EV_j) \Leftrightarrow (OP_i \xrightarrow{P} OP_j)$$

$$(EV_i \xrightarrow{C+} EV_j) \Leftrightarrow (OP_i \xrightarrow{C+} OP_j)$$

$$(EV_i \xrightarrow{C} EV_j) \Leftrightarrow (OP_i \xrightarrow{C} OP_j)$$

$$(EV_i \xrightarrow{I} EV_j) \Rightarrow (OP_i \xrightarrow{I} OP_j)$$

$$(EV_i \xrightarrow{IM} EV_j) \Rightarrow (OP_i \xrightarrow{IM} OP_j)$$

• Exemple :

Le graphe des dépendances entre C-OPERATIONS correspondant au schéma dynamique du traitement d'une ligne de commande acceptée, présenté au paragraphe II-2.2.1. est le suivant :

• Les dépendances directes sont :

$$(EV5 \xrightarrow{P} EV4) \Leftrightarrow (OP6 \xrightarrow{P} OP5)$$

$$(EV4 \xrightarrow{P} EV2) \Leftrightarrow (OP5 \xrightarrow{P} OP3)$$

$$(EV1 \xrightarrow{C} EV2) \Leftrightarrow (OP1 \xrightarrow{C} OP3)$$

$$(EV2 \xrightarrow{C} EV3) \Leftrightarrow (OP3 \xrightarrow{C} OP4)$$

• Ceci entraîne les dépendances suivantes :

$$(EV5 \xrightarrow{P+} EV4) \Leftrightarrow (OP6 \xrightarrow{P+} OP5)$$

$$(EV4 \xrightarrow{P+} EV2) \Leftrightarrow (OP5 \xrightarrow{P+} OP3)$$

$$(EV5 \xrightarrow{P+} EV2) \Leftrightarrow (OP6 \xrightarrow{P+} OP3)$$

$$(EV1 \xrightarrow{C+} EV2) \Leftrightarrow (OP1 \xrightarrow{C+} OP3)$$

$$(EV2 \xrightarrow{C+} EV3) \Leftrightarrow (OP3 \xrightarrow{C+} OP4)$$

$$(EV1 \xrightarrow{C+} EV3) \Leftrightarrow (OP1 \xrightarrow{C+} OP4)$$

$$(EV4 \xrightarrow{C+} EV3) \Leftrightarrow (OP5 \xrightarrow{C+} OP4)$$

$$(EV5 \xrightarrow{C+} EV3) \Leftrightarrow (OP6 \xrightarrow{C+} OP4)$$

• D'où les relations d'indépendances suivantes :

$$\left. \begin{aligned} (EV2 \xrightarrow{I} EV1) &\Rightarrow (OP3 \xrightarrow{I} OP1) \wedge (OP3 \xrightarrow{I} OP2) \\ (EV3 \xrightarrow{I} EV1) &\Rightarrow (OP4 \xrightarrow{I} OP1) \wedge (OP4 \xrightarrow{I} OP2) \\ (EV4 \xrightarrow{I} EV1) &\Rightarrow (OP5 \xrightarrow{I} OP1) \wedge (OP5 \xrightarrow{I} OP2) \\ (EV5 \xrightarrow{I} EV1) &\Rightarrow (OP6 \xrightarrow{I} OP1) \wedge (OP6 \xrightarrow{I} OP2) \end{aligned} \right\} \begin{array}{l} \text{car EV1 déclenche OP1} \\ \text{et OP2} \end{array}$$

$$(EV3 \xrightarrow{I} EV2) \Rightarrow (OP3 \xrightarrow{I} OP4)$$

$$(EV1 \xrightarrow{I} EV4) \Rightarrow (OP1 \xrightarrow{I} OP5) \wedge (OP2 \xrightarrow{I} OP5) \text{ car EV1 déclenche OP1 et OP2}$$

$$(EV2 \xrightarrow{I} EV4) \Rightarrow (OP3 \xrightarrow{I} OP5)$$

$$(EV3 \xrightarrow{I} EV4) \Rightarrow (OP4 \xrightarrow{I} OP5)$$

$$(EV1 \xrightarrow{I} EV5) \Rightarrow (OP1 \xrightarrow{I} OP5) \wedge (OP2 \xrightarrow{I} OP6) \text{ car EV1 déclenche OP1 et OP2}$$

$$(EV2 \xrightarrow{I} EV5) \Rightarrow (OP3 \xrightarrow{I} OP6)$$

$$(EV3 \xrightarrow{I} EV5) \Rightarrow (OP4 \xrightarrow{I} OP6)$$

$$(EV4 \xrightarrow{I} EV5) \Rightarrow (OP5 \xrightarrow{I} OP6)$$

Comme le fait que deux C-EVENEMENTS soient en indépendance constitue une condition suffisante mais non nécessaire à l'indépendance des C-OPERATIONS qu'ils déclenchent, nous avons de plus les relations suivantes :

$$OP2 \xrightarrow{I} OP1$$

$$OP2 \xrightarrow{I} OP3$$

$$OP2 \xrightarrow{I} OP4$$

• et les dépendances mutuelles suivantes :

$$(EV1 \xrightarrow{IM} EV4) \Leftrightarrow (OP1 \xrightarrow{IM} OP5) \wedge (OP2 \xrightarrow{IM} OP5)$$

$$(EV1 \xrightarrow{IM} EV5) \Leftrightarrow (OP1 \xrightarrow{IM} OP6) \wedge (OP2 \xrightarrow{IM} OP6)$$

Pour les mêmes raisons que c-dessus, nous aurons en plus les relations suivantes :

$$OP2 \xrightarrow{IM} OP1$$

$$OP2 \xrightarrow{IM} OP3$$

$$OP2 \xrightarrow{IM} OP4$$

• Soit, en regroupant ces relations dans un tableau, avec les mêmes conventions que dans le cas du graphe de dépendances chronologiques entre C-EVENEMENTS :

| | OP1 | OP2 | OP3 | OP4 | OP5 | OP6 |
|-----|-----|-----|-----|-----|-----|-----|
| OP1 | | IM | C | C+ | IM | IM |
| OP2 | IM | | IM | IM | IM | IM |
| OP3 | I | IM | | C | I | I |
| OP4 | I | IM | I | | I | I |
| OP5 | IM | IM | P | C+ | | I |
| OP6 | IM | IM | P+ | C+ | P | |

Figure II-10

• D'où le graphe des dépendances entre C-OPERATIONS, compte tenu des mêmes conventions graphiques que dans le cas des C-EVENEMENTS :

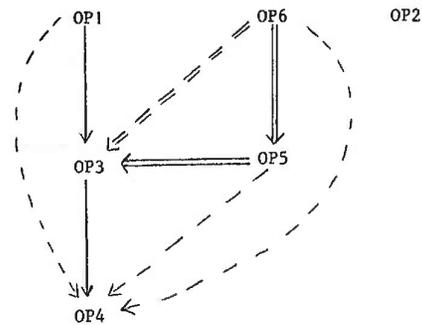


Figure II-11

II-2.3.3.- Conclusion sur les dépendances entre C-OPERATIONS

Nous venons de montrer que la structure conceptuelle contient une expression de la synchronisation. On peut visualiser cette expression par le graphe des dépendances entre C-OPERATIONS (pour retrouver la vision classique de la synchronisation) lui-même déduit du graphe des dépendances chronologiques entre C-EVENEMENT.

C'est en fait la notion de C-EVENEMENT qui permet dans notre modélisation de représenter l'interconnexion entre processus (les C-OPERATIONS), c'est-à-dire leur synchronisation : nous avons ainsi établi une typologie des situations de synchronisation qui permettent de lier deux C-OPERATIONS, typologie dérivée de celle des relations chronologiques entre C-EVENEMENTS. Ceci nous semble un des points caractérisant l'originalité de notre travail : nous avons établi une typologie des situations de synchronisation liant deux "processus" (les C-OPERATIONS) à partir d'une analyse des causes de déclenchement de ces processus (l'arrivée d'un C-EVENEMENT) et des liens existant entre celles-ci (relations chronologiques entre C-EVENEMENTS), et non pas à partir d'une analyse en termes de ressources ou de coopération entre processus basée sur le partage de celles-ci, comme c'est habituellement le cas [13], [25].

Les trois classes de situations entre C-EVENEMENTS conduisent à une certaine représentation des trois situations de synchronisation de base (toutes les autres situations de synchronisation étant des combinaisons de ces situations de base). Cette représentation correspond à la typologie des situations de synchronisation de base couramment admise [13] diminuée des situations correspondant à des contraintes dues aux ressources (exclusion mutuelle, parallélisme) ne relevant pas, de notre point de vue, du niveau conceptuel :

- une opération suit toujours une opération donnée,
- une opération peut suivre une opération donnée,
- deux opérations sont indépendantes.

Notre propos, dans la fin de ce chapitre, est de reprendre la vérification intuitive des conditions C1, C2, C3 de prise en compte de la synchronisation faite au paragraphe II-2.1. en exploitant les définitions introduites.

II-3.- CONCLUSION : PRISE EN COMPTE DE LA SYNCHRONISATION PAR LES RELATIONS DE DEPENDANCE CHRONOLOGIQUE ENTRE C-EVENEMENTS

II-3.1.- La synchronisation est exprimée par les dépendances chronologiques entre C-EVENEMENTS

Rappelons que, pour résoudre un problème de synchronisation au niveau conceptuel, il faut satisfaire les trois conditions suivantes (cf introduction du paragraphe II-2) :

- C1 : déterminer l'ensemble des processus du système, ainsi que ses constituants.
- C2 : définir les interrelations entre ces processus, c'est-à-dire leur dépendance ou leur indépendance.
- C3 : définir les règles de coopération temporelle entre ces processus en fonction des règles définissant la logique du problème envisagé, c'est-à-dire définir un ordre sur l'exécution des processus en fonction de ces règles et de leurs interrelations.

Nous ne reviendrons pas sur la condition C1, dont le développement au paragraphe II-2.1. nous semble suffisant. Rappelons simplement que l'ensemble des processus est l'ensemble des opérations représentées par les C-OPERATIONS, et que l'ensemble des constituants est l'ensemble des objets représentés par les C-OBJETS, ceci correspondant à un raisonnement en termes de processus et de constituants élémentaires.

Nous allons maintenant montrer comment les deux conditions C2 et C3 sont satisfaites au moyen des relations chronologiques entre C-EVENEMENTS induisant les relations de dépendance entre C-OPERATIONS.

II-3.1.1.- Satisfaction de la condition C2

Nous avons abouti à la correspondance suivante entre les relations de dépendance chronologique permanente et conditionnelle et les relations de dépendance entre C-OPERATIONS :

$$\begin{aligned} (EV_i \xrightarrow{P^+} EV_j) &\Leftrightarrow (OP_i \xrightarrow{P^+} OP_j) \\ (EV_i \xrightarrow{P} EV_j) &\Leftrightarrow (OP_i \xrightarrow{P} OP_j) \\ (EV_i \xrightarrow{C^+} EV_j) &\Leftrightarrow (OP_i \xrightarrow{C^+} OP_j) \\ (EV_i \xrightarrow{C} EV_j) &\Leftrightarrow (OP_i \xrightarrow{C} OP_j) \end{aligned}$$

De plus, les règles permettant de définir les relations d'indépendance entre les C-OPERATIONS sont obtenues soit à partir des relations d'indépendance entre C-EVENEMENTS (condition suffisante mais non nécessaire), soit à partir des relations de dépendance entre C-OPERATIONS, elles-mêmes définies à partir des dépendances chronologiques entre C-EVENEMENTS :

$$\begin{aligned} (EV_i \xrightarrow{I} EV_j) &\Leftrightarrow (OP_i \xrightarrow{I} OP_j) \\ (EV_i \xrightarrow{IM} EV_j) &\Leftrightarrow (OP_i \xrightarrow{IM} OP_j) \\ (\neg(OP_i \xrightarrow{P^+} OP_j) \wedge \neg(OP_i \xrightarrow{C^+} OP_j)) &\Leftrightarrow (OP_i \xrightarrow{I} OP_j) \\ ((OP_i \xrightarrow{I} OP_j) \wedge (OP_j \xrightarrow{I} OP_i)) &\Leftrightarrow (OP_i \xrightarrow{IM} OP_j) \Leftrightarrow (OP_j \xrightarrow{IM} OP_i) \\ (OP_i \xrightarrow{IM} OP_j) &\Leftrightarrow [((OP_i \xrightarrow{I} OP_j) \wedge (OP_j \xrightarrow{I} OP_i)) \vee ((\exists EV_k \in EV) \\ &\quad (EV_k \xrightarrow{+} OP_i \wedge EV_j \xrightarrow{+} OP_k))] \end{aligned}$$

Nous venons donc de montrer que les relations de dépendance chronologique entre C-EVENEMENTS permettent de définir l'indépendance ou la dépendance entre les C-OPERATIONS c'est-à-dire qu'elles permettent de satisfaire la condition C2.

II-3.1.2.- Satisfaction de la condition C3

Nous allons montrer que les relations de dépendance chronologique entre C-EVENEMENTS induisant les relations de dépendance entre C-OPERATIONS permettent de satisfaire la condition C3 de prise en compte de la synchronisation, c'est-à-dire qu'elles permettent de définir un ordre sur l'exécution des processus (les C-OPERATIONS dans notre cas) en fonction des règles définissant la logique du système réel et de leurs interrelations (dépendance ou indépendance).

- Le schéma conceptuel est une traduction en termes de C-OBJETS, C-OPERATIONS et C-EVENEMENTS de la sémantique d'un problème donné [86], [87]. Il traduit donc la logique du problème envisagé par les liens de causalité (un C-EVENEMENT déclenche une C-OPERATION qui provoque le changement d'état d'un C-OBJET qui peut à son tour constituer un C-EVENEMENT) exprimés dans le sous-schéma dynamique de ce problème. Or, les relations de dépendance chronologique entre C-EVENEMENTS sont obtenues par analyse de ce sous-schéma dynamique. Elles respectent donc la logique du problème traité (c'est-à-dire qu'un C-EVENEMENT ne sera en relation de dépendance avec un autre C-EVENEMENT que s'il existe bien une subordination des événements réels représentés par le premier C-EVENEMENT aux événements réels représentés par le second C-EVENEMENT).

- De plus, pour tout C-EVENEMENT EV_i donné :

- $(\xrightarrow{P^+})^{-1}(EV_i) = \{EV_j / EV_j \xrightarrow{P^+} EV_i\}$ permet de déterminer tous les C-EVENEMENTS dont l'arrivée doit nécessairement précéder celle de EV_i
- $(\xrightarrow{C^+})^{-1}(EV_i) = \{EV_k / EV_k \xrightarrow{C^+} EV_i\}$ permet de déterminer tous les C-EVENEMENTS dont l'arrivée peut conditionnellement précéder celle de EV_i
- $(\xrightarrow{+})^{-1}(EV_i) = \{EV_k / EV_k \xrightarrow{+} EV_i\}$ permet donc de déterminer tous les C-EVENEMENTS dont l'arrivée peut précéder celle de EV_i

Remarque : par définition des deux relations $\xrightarrow{P^+}$ et $\xrightarrow{C^+}$, nous avons :

$$(\xrightarrow{+})^{-1}(EV_i) = (\xrightarrow{P^+})^{-1}(EV_i) \cup (\xrightarrow{C^+})^{-1}(EV_i) \text{ d'où la remarque précédente sur cette relation.}$$

- De même,

- $(\xrightarrow{P})^{-1}(EV_i) = \{EV_j / EV_j \xrightarrow{P} EV_i\}$ permet de déterminer tous les C-EVENEMENTS dont l'arrivée suit nécessairement celle de EV_i
- $(\xrightarrow{C})^{-1}(EV_i) = \{EV_k / EV_k \xrightarrow{C} EV_i\}$ permet de déterminer tous les C-EVENEMENTS dont l'arrivée peut conditionnellement suivre celle de EV_i
- $(\xrightarrow{+})^{-1}(EV_i) = \{EV_k / EV_k \xrightarrow{+} EV_i\}$ permet de déterminer tous les C-EVENEMENTS dont l'arrivée peut suivre celle de EV_i

Remarque : Pour les mêmes raisons que ci-dessus nous avons :

$$(\xrightarrow{+})(EV_i) = (\xrightarrow{P+})(EV_i) \cup (\xrightarrow{C+})(EV_i) \text{ ce qui justifie la remarque précédente concernant cette relation.}$$

Or si nous effectuons cette analyse pour tout C-EVENEMENT EV_i , et si nous regroupons les résultats sous forme d'un graphe, nous obtenons le graphe des dépendances chronologiques entre C-EVENEMENTS présenté au paragraphe II-2.2.3.

Nous venons donc de montrer que le graphe des dépendances chronologiques entre C-EVENEMENTS permet de définir un ordre sur l'arrivée des C-EVENEMENTS qui respecte la logique du problème envisagé et les interrelations entre ces C-EVENEMENTS.

Or les relations de dépendance chronologique entre C-EVENEMENTS permettent de définir les relations de dépendance entre C-OPERATIONS. Nous pouvons donc effectuer la même analyse que ci-dessus en termes de C-OPERATIONS : pour toute C-OPERATION OP_i :

- $(\xrightarrow{P+})^{-1}(OP_i) = \{OP_j/OP_j \xrightarrow{P+} OP_i\}$ permet de déterminer toutes les C-OPERATIONS dont le déclenchement doit nécessairement précéder celui de OP_i
- $(\xrightarrow{C+})^{-1}(OP_i) = \{OP_k/OP_k \xrightarrow{C+} OP_i\}$ permet de déterminer toutes les C-OPERATIONS dont le déclenchement peut conditionnellement précéder celui de OP_i
- $(\xrightarrow{+})^{-1}(OP_i) = \{OP_\ell/OP_\ell \xrightarrow{+} OP_i\} = (\xrightarrow{P+})^{-1}(OP_i) \cup (\xrightarrow{C+})^{-1}(OP_i)$ permet donc de déterminer toutes les C-OPERATIONS dont le déclenchement peut précéder celui de OP_i .

Remarque : La relation de dépendance $\xrightarrow{+}$ est définie par :

$$(OP_i \xrightarrow{+} OP_j) \Leftrightarrow (OP_i \xrightarrow{P+} OP_j) \vee (OP_i \xrightarrow{C+} OP_j)$$

De même :

- $(\xrightarrow{P+})(OP_i) = \{OP_j/OP_j \xrightarrow{P+} OP_i\}$ permet de déterminer toutes les C-OPERATIONS dont le déclenchement doit nécessairement suivre celui de OP_i .
- $(\xrightarrow{C+})(OP_i) = \{OP_k/OP_k \xrightarrow{C+} OP_i\}$ permet de déterminer toutes les C-OPERATIONS dont le déclenchement peut conditionnellement suivre celui de OP_i .

• $(\xrightarrow{+})(OP_i) = \{OP_\ell/OP_\ell \xrightarrow{+} OP_i\} = (\xrightarrow{P+})(OP_i) \cup (\xrightarrow{C+})(OP_i)$ permet donc de déterminer toutes les C-OPERATIONS dont le déclenchement peut suivre celui de OP_i .

Si nous effectuons cette analyse pour toute C-OPERATION OP_i , et si nous regroupons les résultats sous forme d'un graphe, nous obtenons le graphe des dépendances entre C-OPERATIONS présenté au paragraphe II-2.3.2.

Nous venons donc de montrer que le graphe des dépendances entre C-OPERATIONS permet de déterminer pour toute C-OPERATION OP_i quelles sont les C-OPERATIONS dont le déclenchement doit nécessairement, ou peut conditionnellement précéder ou suivre celui de OP_i . Nous venons ainsi de montrer que ce graphe permet de définir un ordre sur le déclenchement des C-OPERATIONS qui respecte la logique du problème envisagé (puisque les relations de dépendance entre les C-OPERATIONS sont déduites des relations chronologiques entre C-EVENEMENTS et nous avons montré ci-dessus que ces relations respectent la logique du problème traité) et les interrelations entre les C-OPERATIONS puisque traduisant leurs relations de dépendance.

Ainsi, nous venons de montrer que la condition C3 est satisfaite par le modèle conceptuel au moyen des relations de dépendance chronologique entre C-EVENEMENTS et des relations de dépendance entre C-OPERATIONS. Nous avons donc ainsi montré que le modèle conceptuel permet la prise en compte des situations de synchronisation au niveau conceptuel, puisqu'il satisfait aux trois conditions à remplir pour résoudre ces problèmes.

Reprenons l'exemple présenté au paragraphe II-2.2.1. pour illustrer ce propos.

II-3.2.- Exemple

Rappelons le tableau regroupant les relations de dépendance chronologique entre les C-EVENEMENTS correspondant à ce problème et le graphe des dépendances chronologiques associé :

| | EV1 | EV2 | EV3 | EV4 | EV5 |
|-----|-----|-----|-----|-----|-----|
| EV1 | | C | C+ | IM | IM |
| EV2 | I | | C | I | I |
| EV3 | I | I | | I | I |
| EV4 | IM | P | C+ | | I |
| EV5 | IM | P+ | C+ | P | |

Figure II-12 A

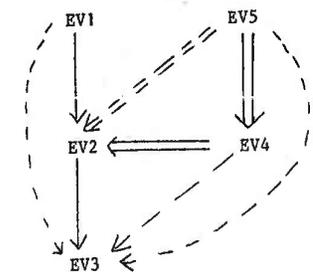


Figure II-12 B

Ils permettent de déduire le tableau regroupant les dépendances entre les C-OPERATIONS, ainsi que le graphe associé :

| | OP1 | OP2 | OP3 | OP4 | OP5 | OP6 |
|-----|-----|-----|-----|-----|-----|-----|
| OP1 | | | | | | |
| OP2 | IM | | | | | |
| OP3 | I | IM | | | | |
| OP4 | I | IM | I | | | |
| OP5 | IM | IM | P | C+ | | |
| OP6 | IM | IM | P+ | C+ | P | |

Figure II-13 A

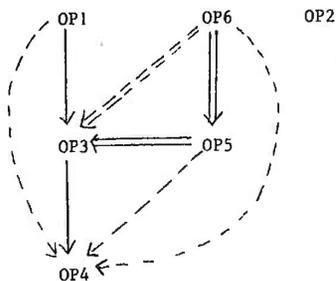


Figure II-13 B

Le cheminement dans ce graphe, c'est-à-dire la détermination des C-OPERATIONS à déclencher à la suite de l'exécution d'une C-OPERATION OP_i donnée se fait :

- d'après l'arrivée du C-EVENEMENT EV_j déterminé par le changement d'état d'un C-OBJET produit par cette C-OPERATION OP_i .
- d'après les conditions d'exécution relatives à l'état du système (de l'ensemble des C-OBJETS) associées aux C-OPERATIONS en dépendance directe avec OP_i et qui sont déclenchées par EV_j .

Nous allons illustrer cela à partir d'une situation complexe du point de vue de la synchronisation extraite du sous-schéma dynamique du problème considéré, et qui traduit plus particulièrement la gestion du stock des produits. Elle est exprimée par le sous-schéma suivant que nous avons renuméroté, conformément à l'exemple présenté en II-2.2.1. :

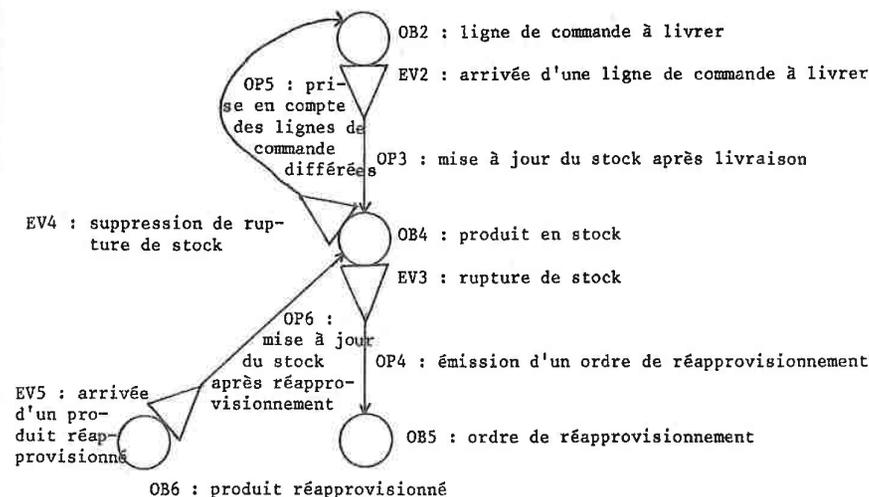


Figure II-14

Les trois types d'ordonnancement possibles entre les processus (les C-OPERATIONS) sont les suivants :

- 1) mise à jour du stock après livraison (correspondant à la livraison d'une ligne de commande, mais n'entraînant pas de rupture de stock) : OP3
- 2) mise à jour du stock après livraison entraînant une rupture de stock donc l'émission d'un bon de réapprovisionnement : OP3 - OP4.
- 3) réapprovisionnement entraînant la prise en compte des lignes de commande différées (essai de satisfaction de ces lignes de commande).

Montrons comment ces situations et leur éventuel enchaînement (dans le cas de (3) et (1) ou (3) et (2)) sont contenues dans le graphe des dépendances entre C-OPERATIONS.

Les dépendances entre les C-OPERATIONS intervenant dans cette situation de synchronisation sont exprimées par le sous-graphe suivant, extrait du graphe de dépendances correspondant à l'exemple de traitement d'une ligne de commande acceptée (paragraphe II-2.2.1.) :

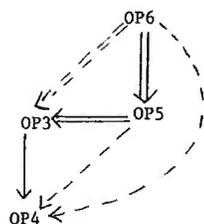


Figure II-15

Sur ce sous-graphe de dépendances, il apparaît que :

- OP3 peut être activée seule (situation (1))
- OP4 peut être activée à la suite de OP3 (dépendance conditionnelle - situation (2))
- OP5 est toujours activée à la suite de OP6 (dépendance permanente - situation (3)).

Et ainsi le graphe des dépendances entre C-OPERATIONS rend bien compte des situations de synchronisation de base constituant le problème global de synchronisation. De plus, il exprime leurs différents enchaînements. En effet, nous pouvons avoir les séquences d'activations suivantes :

- $OP6 \xrightarrow{P} OP5 \xrightarrow{P} OP3$ (dépendance permanente de OP5 par rapport à OP6 et dépendance permanente de OP3 par rapport à OP6, traduisant l'enchaînement de la situation (3) et de la situation (1)).
- $OP6 \xrightarrow{P} OP5 \xrightarrow{P} OP3 \xrightarrow{C} OP4$ (dépendances permanentes de OP5 par rapport à OP6, de OP3 par rapport à OP5 et dépendance conditionnelle de OP4 par rapport à OP3, traduisant l'enchaînement de la situation (3) et de la situation (2)).

La détermination de la situation de synchronisation dans laquelle le système se trouve est effectuée par la reconnaissance de l'arrivée des événements déclencheurs de ces opérations par la machine abstraite (voir paragraphe II-1.4.).

En conclusion, il apparaît bien sur cet exemple que le raisonnement sur le graphe des dépendances entre C-OPERATIONS est équivalent au raisonnement sur le schéma conceptuel, mais il présente l'avantage de bien mettre en évidence les situations de synchronisation (d'enchaînement) entre les C-OPERATIONS.

II-3.3.- Conclusion

- Nous venons de montrer que le modèle conceptuel permet la prise en compte des situations de synchronisation, au moyen des relations de dépendance chronologiques entre C-EVENEMENTS induisant les relations de dépendance entre C-OPERATIONS. La détermination de la situation de synchronisation dans laquelle se trouve le système abstrait étant assurée par la reconnaissance de l'arrivée des événements par la machine abstraite, qui constitue ainsi l'outil de gestion de la synchronisation associé à cette modélisation. C'est un outil spécifique, dont le fonctionnement est basé sur la reconnaissance des événements, ceci supposant bien sûr un langage permettant l'exploitation de la structure conceptuelle définie en termes de ce langage par la machine abstraite [63], [86], pour la gestion de la synchronisation au moyen de la détermination des dépendances fonctionnelles entre C-EVENEMENTS et C-OPERATIONS.

- Nous allons, dans la suite de ce travail, essayer de déterminer la puissance de modélisation de notre approche, ainsi que les contrôles qu'elle permet, au moyen d'une comparaison formelle avec les approches habituelles, notamment les réseaux de Petri [80], [77], [30] à arcs inhibiteurs [1], [2] dont la puissance de modélisation est connue [44].

CHAPITRE III

EVALUATION THÉORIQUE DE LA

MODÉLISATION CONCEPTUELLE

Dans ce chapitre, nous allons tenter de situer le modèle conceptuel par rapport aux modèles de représentation de processus synchronisés existants, notamment du point de vue de la puissance de modélisation et des contrôles formels qu'il autorise. Nous utilisons le modèle des réseaux de Petri comme modèle de référence, et nous montrons que le modèle conceptuel a la même puissance de modélisation qu'un réseau de Petri inhibiteur (appliqué à la modélisation d'un système réel). Nous montrons en outre que le graphe des dépendances chronologiques entre C-EVENEMENTS est utile pour effectuer des contrôles formels du type de ceux qui ont été associés aux réseaux de Petri.

III-1.- RAPPELS SUR LES RESEAUX DE PETRI - EXPRESSION DES CONDITIONS DE VALIDATION D'UNE TRANSITION AU MOYEN DE FONCTIONS BOOLEENNES

III-1.1.- Rappels sur les réseaux de Petri

On trouvera une présentation globale de ce type de réseau dans [77], [92], [30], [78] ainsi qu'une étude théorique de ces réseaux dans [1], [43], [90] par exemple. Nous nous contenterons de rappeler ici les notions indispensables à la compréhension de la suite de ce chapitre.

III-1.1.1.- Réseaux de Petri

• Définitions : [77], [90], [92].

• Un réseau de Petri est un triplet $\mathcal{P} = (P, T, A)$

P est un ensemble de places non vide : $P \neq \emptyset$

T est un ensemble de transitions : $T \neq \emptyset$, $P \cap T = \emptyset$

A est une relation qui correspond à un ensemble d'arcs où chaque arc lie soit une place à une transition, soit une transition à une place.

\mathcal{P} est donc un graphe bi-alterné, c'est-à-dire un graphe comprenant deux types de sommets alternés : les places et les transitions.

• Nous utiliserons de plus les notations suivantes :

$$(\forall t \in T) ({}^*t = \{p \in P/A(p,t)\} \text{ et } t^* = \{p \in P/A(t,p)\})$$

$$(\forall p \in P) ({}^*p = \{t \in T/A(t,p)\} \text{ et } p^* = \{t \in T/A(p,t)\})$$

Nous appellerons *t (respectivement t^*) l'ensemble des places amont (respectivement aval) de la transition t , et *p (respectivement p^*) l'ensemble des transitions amont (respectivement aval) de la place p .

• Un marquage M du réseau \mathcal{P} est une application de P dans $\mathbb{N} : P \xrightarrow{M} \mathbb{N}$ qui à toute place p_i du réseau associe un entier naturel $M(p_i)$.

• Une transition t est dite sensibilisée si à chacune de ses places amont est associée par M un entier naturel non nul :

$$(\forall p_i \in {}^*t) (M(p_i) \neq 0)$$

Si l'on interprète le marquage d'un réseau comme une distribution de "jetons" sur l'ensemble des places du réseau (à chaque place p_i on associe $M(p_i)$ jetons), une transition est dite sensibilisée si toutes ses places amont possèdent au moins un jeton.

• Une transition t_i sensibilisée peut être tirée, un jeton est alors enlevé à chacune de ses places amont et un jeton est ajouté à une de ses places aval. Il n'y a donc pas conservation des jetons, car en général $\text{Card}({}^*t_i) \neq \text{Card}(t_i)$. La mise à feu d'une transition t se traduit donc par :

$$(\forall p_j \in {}^*t_i) (M(p_j) = M(p_j) - 1)$$

$$(\forall p_j \in t_i) (M(p_j) = M(p_j) + 1)$$

Soit M_i un marquage sensibilisant t_i et soit M_j le marquage résultant du tir de t_i . Nous écrivons alors $M_i \xrightarrow{t_i} M_j$.

Soit σ une séquence finie de transitions $t_i, t_{i+1}, \dots, t_{i+k}$ appartenant à T . On dit que σ est une séquence de tir tirable à partir de M_i si et seulement s'il existe des marquages $M_{i+1}, M_{i+2}, \dots, M_{i+k+1}$ de \mathcal{P} tels que :

$$M_i \xrightarrow{t_i} M_{i+1} \xrightarrow{t_{i+1}} M_{i+2} \dots M_{i+k} \xrightarrow{t_{i+k}} M_{i+k+1}$$

ce qui sera noté

$$M_i \xrightarrow{\sigma} M_{i+k+1}$$

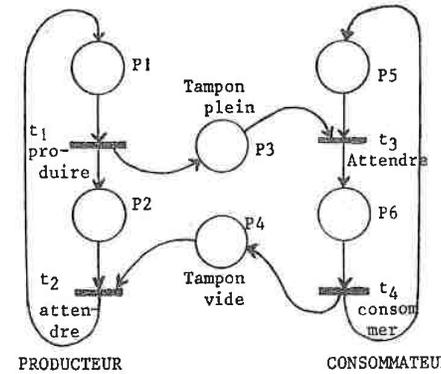
• La classe des marquages conséquents \vec{M}_O est l'ensemble des marquages accessibles depuis le marquage M_O par une séquence de tir :

$$\vec{M}_O = \{M_i / \exists \sigma \text{ et } M_O \xrightarrow{\sigma} M_i\}$$

• Utilisation :

• Généralement, à chaque transition est associée une action (ou opération, ou processus). Le tir de la transition symbolise l'exécution de l'action. Les places amont de cette transition représentent les conditions à remplir pour que l'action qui lui est associée soit exécutée.

• Exemple d'utilisation : producteur/consommateur avec tampon unique



Convention graphique :

- Les places sont représentées par des cercles et les transitions par des barres.

Figure III-1

• Notion de conflit entre deux transitions [77], [92]

- Deux transitions t_i et t_j sont dites en conflit lorsque le tir de l'une de ces transitions désensibilise l'autre, ce qui est illustré sur le schéma ci-dessous :

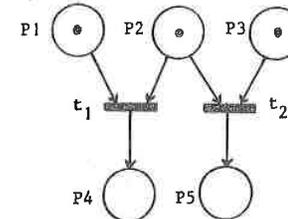


Figure III-2-A

Supposons que les deux transitions soient sensibilisées par l'existence d'un jeton dans chacune des places P1, P2, P3. Le tirage de l'une de ces deux transitions désensibilise l'autre. En effet, supposons que l'on tire la transition t_1 . Ceci se traduira par la suppression du jeton de P1 et du jeton de P2 et par l'arrivée d'un jeton dans P4. Mais la transition t_2 ne pourra plus être tirée : elle n'est plus sensibilisée puisque la place P2 qui est une place amont de t_2 ne possède plus de jeton, comme il apparaît sur la figure ci-dessous. (Le problème est parfaitement symétrique si l'on choisit de tirer t_2).

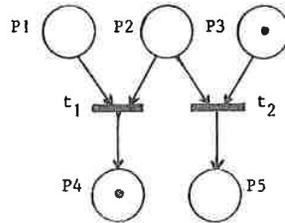


Figure III-2-B

Il faut donc déterminer une stratégie de choix de prise en compte des transitions en cas de conflit, et préciser quelles sont les transitions à tirer dans ce cas.

- Nous pouvons de plus remarquer que deux transitions t_i et t_j ne peuvent être en conflit que si elles ont au moins une place amont en commun, c'est-à-dire si :

$${}^*t_i \cap {}^*t_j \neq \emptyset$$

III-1.1.2.- Principales propriétés définies sur les réseaux de Petri

Soit \mathcal{P} un réseau de Petri et M_0 son marquage initial donné, \vec{M}_0 étant la classe des marquages conséquents correspondante.

• Réseau sauf pour un marquage initial donné [77], [92]

• \mathcal{P} est sauf pour un marquage initial donné M_0 si et seulement si tout marquage appartenant à \vec{M}_0 est tel qu'il y ait au plus un jeton dans chaque place de \mathcal{P} .

• Dans le cas où les opérateurs associés aux transitions du réseau ne sont pas ré-entrants, il est nécessaire de vérifier qu'aucun opérateur ne sera sollicité à nouveau avant d'avoir terminé sa tâche. Si le réseau de Petri est sauf pour son marquage initial, aucune transition ne pourra à la fois être en train d'être tirée et être tirable. Quand aucun opérateur n'est associé à plus d'une transition, le fonctionnement correct est donc garanti. (Dans le cas où un opérateur est associé à plusieurs transitions, il faut bien sûr vérifier que ces transitions ne soient pas tirables simultanément).

• **Exemple** : Le réseau présenté au paragraphe III-1.1.1. et représentant un système producteur-consommateur avec tampon unique est sauf pour le marquage initial constitué d'un jeton dans les places P1 et P5.

• Réseau vivant pour un marquage initial donné [61], [77], [92]

• Si, pour toute transition t de \mathcal{P} et pour tout marquage M_i de \vec{M}_0 il existe une séquence de tir qui tire t à partir de M_i , alors \mathcal{P} est dit vivant pour M_0 (rappelons qu'une séquence de tir est finie par définition).

• Le fait qu'un réseau de Petri soit vivant pour un marquage initial M_0 donné a deux conséquences importantes :

- d'une part cela entraîne l'absence de blocage (marquage accessible à partir duquel aucune transition n'est tirable).
- d'autre part, il est certain qu'aucune partie du réseau ne deviendra inaccessible après une certaine séquence de tir.

• **Exemple** : le réseau présenté au paragraphe III-1.1.1. et représentant un système producteur-consommateur avec tampon unique est vivant pour le marquage initial qui le rend sauf.

• Réseau borné pour un marquage initial donné [77], [90], [92]

• Si pour tout marquage M_i de \vec{M}_0 , le nombre de jetons de chaque place de \mathcal{P} est inférieur ou égal à un entier naturel k , alors \mathcal{P} est dit k -borné pour le marquage initial M_0 . (Il est alors évident qu'un réseau sauf est un réseau 1-borné).

• Cette propriété prend toute son importance au moment de l'implémentation d'un système conçu et modélisé au moyen d'un réseau de Petri, car comme la capacité de tout composant hardware donné est borné, ce réseau de Petri doit lui aussi être borné (c'est-à-dire qu'il existe k de \mathbb{N} tel que le réseau soit k -borné).

• **Exemple** : comme le réseau présenté au paragraphe III-1.1.1. est sauf pour le marquage présenté ci-dessus, il est évidemment 1-borné pour ce même marquage.

III-1.1.3.- Puissance de modélisation des réseaux de Petri : les réseaux de Petri à arcs inhibiteurs

- Kosaraju a montré [58] que, dans le cas général, un branchement conditionné à la valeur zéro d'une variable n'était pas représentable par un réseau de Petri seul, Agerwala a alors proposé [1] d'ajouter aux réseaux de Petri des arcs "inhibiteurs". Si (p_i, t_j) est un tel arc, il est nécessaire que la place p_i soit vide pour que la transition t_j soit sensibilisée. Les arcs inhibiteurs sont représentés par des flèches labellées d'un zéro.

- De plus, Agerwala a démontré [1] que les réseaux de Petri à arcs inhibiteurs ont la puissance de modélisation d'une machine de Turing.

- Enfin, comme un arc inhibiteur est toujours dirigé d'une place vers une transition, il est nécessaire que dans un réseau de Petri à arcs inhibiteurs les actions soient nécessairement associées aux transitions du réseau, et les conditions à remplir pour exécuter ces actions soient associées aux places de ce réseau [77], [92].

III-1.2.- Expression des conditions de sensibilisation d'une transition au moyen de fonctions booléennes

Lorsqu'on modélise un système au moyen d'un réseau de Petri à arcs inhibiteurs, on associe à chaque transition de ce réseau un processus du système, et les places amont de cette transition expriment alors les conditions de déclenchement de ce processus.

Montrons d'abord que les conditions de sensibilisation d'une transition (c'est-à-dire les conditions d'activation du processus associé à cette transition) qui sont exprimées par les places du réseau, peuvent se ramener à une fonction booléenne sur les transitions du réseau (c'est-à-dire que les conditions d'activation d'un processus seront définies à partir de la terminaison d'autres processus), et par la même sur les états des constituants du système, traduisant la terminaison des processus associés à ces transitions.

Soient t_i une transition du réseau, et $t_i = \{p_1^i, \dots, p_j^i, \dots, p_{n_i}^i\}$ l'ensemble des places amont de t_i .

Pour que t_i soit sensibilisée, donc tirable, il faut que, pour toutes les places p_j^i de t_i :

- si l'arc (p_j^i, t_i) est inhibiteur, le marquage de p_j^i soit nul : $M(p_j^i) = 0$
- si l'arc (p_j^i, t_i) n'est pas inhibiteur, le marquage de p_j^i soit non nul : $M(p_j^i) > 0$.

C'est-à-dire que, pour que la transition t_i soit tirable (c'est-à-dire pour que le processus P_{t_i} qui lui est associé soit activable), il faut que la fonction booléenne suivante, que nous noterons $S(t_i)$ puisqu'elle traduit les conditions de sensibilisation de t_i prenne la valeur 1 (ou VRAI) :

$$S(t_i) = \bigwedge_{i=1}^{n_i} f_j^i \text{ avec } f_j^i = \begin{cases} f_{p_j^i} & \text{si l'arc } (p_j^i, t_i) \text{ n'est pas inhibiteur} \\ \overline{f_{p_j^i}} & \text{si l'arc } (p_j^i, t_i) \text{ est inhibiteur} \end{cases}$$

où $f_{p_j^i}$ désigne la fonction booléenne traduisant le fait que le marquage $M(p_j^i)$ de p_j^i soit non nul, et $\overline{f_{p_j^i}}$ désigne la négation de cette fonction.

De même, le marquage $M(p_j^i)$ de p_j^i est non nul si et seulement si au moins une des transitions amont de p_j^i a été tirée.

Soit $p_j^i = \{t_1^{ij}, \dots, t_k^{ij}, \dots, t_m^{ij}\}$ l'ensemble des transitions amont de la place p_j^i . Pour que le marquage $M(p_j^i)$ de p_j^i soit non nul, il faut que la fonction booléenne $f_{p_j^i}$ suivante prenne la valeur 1 (ou VRAI) :

$$f_{p_j^i}(t_1^{ij}, \dots, t_m^{ij}) = \bigvee_{k=1}^m \tau(t_k^{ij}) \text{ où } \tau(t_k^{ij}) \text{ prend la valeur 1 (VRAI) si et seulement si } t_k^{ij} \text{ a été tirée.}$$

Nous venons ainsi de montrer que les conditions de sensibilisation d'une transition t_i peuvent être exprimées par une fonction booléenne $S(t_i)$ sur des transitions t_k^{ij} du réseau.

Or si l'on raisonne en termes de système modélisé par un réseau de Petri à arcs inhibiteurs, à chaque transition t_i de ce réseau est associé un processus agissant sur des constituants du système (réseau de Petri interprété). Donc le fait qu'une transition t_i ait été tirée, entraînant l'exécution du processus associé se traduit par une combinaison d'états des constituants du système résultant de l'action de ce processus sur le système. Ainsi, la fonction $\tau(t_k^{ij})$ prenant la valeur VRAI si et seulement si t_k^{ij} a été tirée est en fait une fonction booléenne sur les états des constituants du système, caractérisant la terminaison du processus associé à la transition t_k^{ij} :

$$\tau(t_k^{ij}) = b_k^{ij}(e_1^k, e_2^k, \dots, e_{p_k}^k) \text{ où } b_k^{ij} \text{ est la fonction booléenne traduisant la terminaison du processus } P_{t_i} \text{ associé à } t_i$$

e_s^k est un état d'un constituant élémentaire du système susceptible d'être la conséquence de l'action de P_{t_i} sur ce constituant élémentaire. C'est une variable booléenne prenant la valeur VRAI si l'état est atteint.

Ainsi, comme une fonction booléenne de fonctions booléennes est une fonction booléenne, il apparaît bien que la fonction $S(t_i)$ traduisant les conditions de sensibilisation de t_i peut se ramener à une fonction booléenne F sur les états des constituants élémentaires du système modélisé par ce réseau de Petri inhibiteur :

$$S(t_i) = F(e_1^i, \dots, e_j^i, \dots, e_{q_i}^i)$$

• Forme canonique des fonctions $S(t_i)$

Toute fonction booléenne peut se réduire à une forme canonique disjonctive [60], [69], [82] : une fonction booléenne $S(t_i)$ traduisant les conditions de sensibilisation d'une transition t_i (c'est-à-dire les conditions d'activation du processus P_{t_i} associé) s'écrira donc :

$$S(t_i) = F(e_1^i, \dots, e_j^i, \dots, e_{q_i}^i) = \bigvee_{j=1}^{q_i} m_j^i$$

- où • m_j^i est un impliquant premier : $m_j^i = a_1^{ij} \wedge a_2^{ij} \wedge \dots \wedge a_{p_{ij}}^{ij}$
- et a_k^{ij} est un atome : $a_k^{ij} = \begin{cases} e_s^i \\ \neg e_s^i \end{cases}$
 - e_s^i étant un état d'un constituant élémentaire du système.

III-1.3.- Conclusion

Pour montrer que toute situation de synchronisation, et tout système réel modélisé sous forme d'un réseau de Petri à arcs inhibiteurs est exprimable par le modèle conceptuel, il faut montrer que :

- tout processus associé à une transition est exprimé par le modèle conceptuel,
- toute condition de sensibilisation $S(t_i)$ est exprimée par le modèle conceptuel,
- toutes les relations entre les transitions et les places (ramenées par le biais des fonctions $S(t_i)$ à des relations entre transitions) et en particulier les situations de conflit entre transitions sont exprimées (et résolues) par le modèle conceptuel.

III-2.- PUISSANCE DE MODELISATION DU MODELE CONCEPTUEL

Dans ce paragraphe, nous allons nous attacher à montrer que les trois points ci-dessus sont vérifiés. Dans ce cas, nous pourrions en conclure que le modèle conceptuel offre la même puissance de modélisation que les réseaux de Petri à arcs inhibiteurs pour ce qui est de la modélisation des systèmes (car il est d'autres domaines d'application des réseaux de Petri, notamment l'étude de langage formels [45] ou la compilation [5] par exemple).

III-2.1.- Expression des processus associés aux transitions par le modèle conceptuel

Soit P_λ un processus associé à une transition t_i d'un réseau de Petri inhibiteur.

Rappelons que, au chapitre I, nous avons défini un processus P_λ comme une transformation τ_λ d'un ensemble de données D_λ en un ensemble de sorties S_λ

$$P_\lambda : D_\lambda \xrightarrow{\tau_\lambda} S_\lambda.$$

Décomposons l'ensemble de données D_λ en un ensemble de constituants d'entrée élémentaires ED_λ par décomposition de chaque élément de D_λ en constituants élémentaires du système, et décomposons de la même manière S_λ en un ensemble de sorties élémentaires ES_λ . La transformation τ_λ se décompose en un ensemble de transformations élémentaires τ_k^λ assurant le passage de l'ensemble des données élémentaires ED_λ à l'ensemble des sorties élémentaires ES_λ . Chacune de ces transformations élémentaires τ_k^λ agit sur un constituant du système et un seul : elles font passer d'un état d'un constituant donné à un état d'un autre constituant, et leur ordonnancement s'effectue par le biais de ces états.

Dans notre modélisation, un constituant élémentaire du système est défini par un objet, représenté par un C-OBJET. Les ensembles ED_λ et ES_λ seront donc représentés respectivement par les ensembles de C-OBJETS I_λ et F_λ , les états des occurrences des C-OBJETS de ces ensembles traduisant les états des constituants du système définis par les objets qu'ils représentent. Montrons que la transformation τ_λ est représentée par l'ensemble C_λ des cycles dynamiques conduisant des C-OBJETS de I_λ à ceux de F_λ , conformément aux règles qui la définissent. Un processus P_λ sera alors représenté par le triplet

$$P_\lambda = (I_\lambda, C_\lambda, F_\lambda)$$

• Rappelons qu'un cycle dynamique C_j^λ est défini par :

$$C_j^\lambda = (EV_j, OP_j, OB_j)$$

- où - EV_j est le C-EVENEMENT déclencheur du cycle,
- OP_j est l'ensemble des C-OPERATIONS OP_k^i déclenchées par le C-EVENEMENT $EV_j : OP_j = \{OP_k^j / EV_j \longrightarrow OP_k^j\}$
 - OB_j est l'ensemble des C-OBJETS modifiés par les C-OPERATIONS déclenchées par $EV_j : OB_j = \{OB_\ell^j / (\exists OP_k^j \in OP_j)(OP_k^j \longrightarrow OB_\ell^j)\}$.

Remarque : Par définition de la C-OPERATION, OP_j et OB_j ont même cardinal.

- De plus, un cycle dynamique C_j^λ est dit élémentaire si et seulement s'il ne comporte qu'une C-OPERATION et une seule (donc, par définition de la C-OPERATION, un et un seul C-OBJET modifié). Dans ce cas la C-OPERATION OP_k^j sera notée de la même

manière que l'ensemble OP_j . Il en ira de même pour OB_j^1 et OB_j . Par définition, il est clair que tout cycle dynamique peut se décomposer en un ensemble de cycles dynamiques élémentaires.

• Pour montrer qu'un processus P_λ est représenté par notre modèle, il nous faut donc définir un ensemble C_λ de cycles dynamiques assurant la transformation de l'ensemble des C-OBJETS de I_λ en l'ensemble des C-OBJETS de F_λ .

• Soit EVI_λ l'ensemble des C-EVENEMENTS constatant le passage d'un C-OBJET de I_λ dans un état tel qu'il constitue une donnée du processus P_λ (cet état est l'image de l'état dans lequel se trouve un constituant élémentaire du système lorsqu'il constitue un élément de ED_λ , c'est-à-dire une donnée du processus P_λ) :

$$EVI_\lambda = \{EV_i / (EV_i \rightarrow OB_k) \wedge (OB_k \in I_\lambda) \wedge (EV_i \rightarrow CHG_r(OB_k)) \wedge (CHG_r(OB_k) \in EI_\lambda)\}$$

où - $CHG_r(OB_k)$ désigne le $r^{\text{ème}}$ changement d'état de OB_k
 - EI_λ désigne l'ensemble des changements d'états des C-OBJETS de I_λ tels que ceux-ci passent dans un état où ils constituent une donnée de P_λ .

• De même, soit EVF_λ l'ensemble des C-EVENEMENTS déclenchant une C-OPERATION provoquant un changement d'état d'un C-OBJET OB_k de F_λ tel que OB_k passe dans un état où il constitue une sortie de P_λ :

$$EVF_\lambda = \{EV_j / (EV_j \rightarrow OP_k) \wedge (OP_k \rightarrow OB_k) \wedge (OB_k \in F_\lambda) \wedge (OP_k \rightarrow CHG_r(OB_k)) \wedge (CHG_r(OB_k) \in EF_\lambda)\}$$

où - $CHG_r(OB_k)$ désigne le $r^{\text{ème}}$ changement d'état de OB_k
 - EF_λ désigne l'ensemble des changements d'états des C-OBJETS de F_λ tels que ceux-ci passent dans un état où ils constituent une sortie de P_λ .

• Alors, C_λ est l'ensemble des cycles dynamiques C_j^λ dont le C-EVENEMENT déclencheur EV_j est tel que :

- EV_j constate un changement d'état d'un C-OBJET conduisant celui-ci dans un état qui est l'image de l'état d'un constituant élémentaire du système provoquant l'application d'une des transformations élémentaires τ_k^λ dont l'ensemble des C-OPERATIONS déclenchées par EV_j constitue l'image.

- Il existe au moins un C-OBJET OB_s de OB_j tel qu'une des C-OPERATIONS de OP_j déclenchée par EV_j induise un changement d'état de OB_s conduisant celui-ci dans un état qui est l'image de l'état du constituant élémentaire du système sur lequel porte τ_k^λ après application de celle-ci.

Nous avons ainsi défini chacun des éléments du triplet représentant le processus P_λ , prouvant ainsi que tout processus était représentable par le modèle conceptuel.

De plus, nous pouvons effectuer des contrôles sur la validité d'une représentation d'un processus puisque les relations de dépendance chronologique entre les C-EVENEMENTS déclencheurs des cycles dynamiques décrivant un même processus ne sont pas quelconques, comme nous allons le montrer.

- Relations chronologiques entre C-EVENEMENTS décrivant un même processus :

1) Les C-EVENEMENTS de EVI_λ sont mutuellement indépendants entre eux. Il en va de même pour les C-EVENEMENTS de EVF_λ :

$$-(\forall EV_{\alpha_1} \in EVI_\lambda)(\forall EV_{\alpha_2} \in EVI_\lambda)((EV_{\alpha_1} \neq EV_{\alpha_2}) \Rightarrow (EV_{\alpha_1} \xrightarrow{IM} EV_{\alpha_2}))$$

la justification en est évidente car :

- $(\exists EV_1 \in EVI_\lambda)(\exists EV_2 \in EVI_\lambda)(EV_1 \neq EV_2 \text{ et } EV_1 \xrightarrow{+} EV_2) \Rightarrow (EV_2 \in \overline{EVI}_\lambda)$ est toujours faux
 - il en va de même si $EV_2 \xrightarrow{+} EV_1$

$$-(\forall EV_{\omega_1} \in EVF_\lambda)(\forall EV_{\omega_2} \in EVF_\lambda)((EV_{\omega_1} \neq EV_{\omega_2}) \Rightarrow (EV_{\omega_1} \xrightarrow{IM} EV_{\omega_2}))$$

la justification en est tout aussi évidente que la précédente :

- $(\exists EV_1 \in EVF_\lambda)(\exists EV_2 \in EVF_\lambda)(EV_1 \neq EV_2 \text{ et } EV_2 \xrightarrow{+} EV_1) \Rightarrow (EV_2 \in \overline{EVF}_\lambda)$ est toujours faux
 - il en va de même si $EV_1 \xrightarrow{+} EV_2$.

2) • Par définition de C_λ , tous les C-EVENEMENTS n'appartenant pas à EVI_λ sont en dépendance chronologique avec au moins un C-EVENEMENT de EVI_λ : puisque l'ensemble C_λ des cycles dynamiques représentant le processus P_λ permet de passer de I_λ à F_λ , il existe donc au moins une chaîne de C-OPERATIONS liant, conditionnellement ou non, tout C-EVENEMENT n'appartenant pas à EVI_λ avec un C-EVENEMENT de EVI_λ :

$$(\forall EV_i \in \overline{EVI}_\lambda)(\exists EV_\alpha \in EVI_\lambda)(EV_\alpha \xrightarrow{+} EV_i)$$

• De même, par définition de C_λ , pour tout C-EVENEMENT n'appartenant pas à EVF_λ il existe au moins un C-EVENEMENT appartenant à EVF_λ et n'appartenant pas à EVI_λ (dans le cas général où $EVI_\lambda \cap EVF_\lambda \neq \emptyset$) qui soit en dépendance chronologique avec ce C-EVENEMENT n'appartenant pas à EVF_λ . En effet, puisque l'ensemble C_λ des cycles dynamiques représentant le processus P_λ permet de passer de I_λ à F_λ , il

existe donc au moins une chaîne de C-OPERATIONS liant, conditionnellement ou non, tout C-EVENEMENT n'appartenant pas à EVF_λ avec un C-EVENEMENT de EVF_λ n'appartenant pas à EVI_λ :

$$(\forall EV_i \in \overline{EVF_\lambda})(\exists EV_\omega \in \bigcup_{EVF_\lambda} (EVF_\lambda \cap EVI_\lambda))(EV_i \xrightarrow{+} EV_\omega)$$

Nous avons donc, dans ce paragraphe, montré que tout processus P_λ associé à une transition t_i était exprimé par le modèle conceptuel et que les C-EVENEMENTS permettant sa description étaient en relations chronologiques non quelconques. Il nous faut maintenant montrer que toute condition de sensibilisation $S(t_i)$ d'une transition t_i d'un réseau de Petri à arcs inhibiteurs est exprimée par notre modèle.

III-2.2.- Expression de la condition de sensibilisation d'une transition par le modèle conceptuel

Rappelons que la condition de sensibilisation $S(t_i)$ d'une transition d'un réseau de Petri à arcs inhibiteurs peut se ramener, dans le cas de la modélisation d'un système, à une fonction booléenne sur l'état des constituants élémentaires de ce système. Elle s'exprime sous forme canonique disjonctive :

$$S(t_i) = \bigvee_{j=1}^{q_i} m_j^i$$

où : - m_j^i est un impliquant premier : $m_j^i = a_1^{ij} \wedge a_2^{ij} \wedge \dots \wedge a_{p_{ij}}^{ij} = \bigwedge_{k=1}^{p_{ij}} a_k^{ij}$

- a_k^{ij} est un atome : $a_k^{ij} = \begin{cases} e_s^i \\ \bar{e}_s^i \end{cases}$

- et e_s^i est une variable booléenne traduisant la présence d'un état d'un constituant élémentaire du système.

Donc
$$S(t_i) = \bigvee_{j=1}^{q_i} \left(\bigwedge_{k=1}^{p_{ij}} a_k^{ij} \right) \quad \text{avec} \quad a_k^{ij} = \begin{cases} e_s^i \\ \bar{e}_s^i \end{cases}$$

Dans notre modèle conceptuel, le changement d'état d'un constituant élémentaire défini par un objet représenté par un C-OBJET est constaté par un événement représenté par un C-EVENEMENT. Donc, le passage d'un constituant élémentaire du système dans un état représenté par e_s^i contribuant à la définition de la condition de sensibilisation $S(t_i)$ sera constaté par un événement du type EV_e^i associé à l'objet de type OB_k dont e_s^i représente un état événementiel.

Pour montrer qu'une condition de sensibilisation $S(t_i)$ est représentée par le modèle conceptuel, il nous faut montrer :

- 1) que tout impliquant premier m_j^i est représenté,
- 2) que la réunion de ces impliquants premiers est représentée.

III-2.2.1.- Expression d'un impliquant premier

Rappelons qu'un impliquant premier m_j^i s'écrit : $m_j^i = \bigwedge_{k=1}^{p_{ij}} a_k^{ij}$ avec $a_k^{ij} = \begin{cases} e_s^i \\ \bar{e}_s^i \end{cases}$

Tout changement d'état entraînant le passage d'un constituant élémentaire du système dans un état représenté par e_s^i est constaté par un événement qui constitue une réalisation du C-EVENEMENT EV_e^i . Il en est de même pour l'absence de cet état représenté par \bar{e}_s^i , qui constitue un état du constituant élémentaire du système complémentaire de celui représenté par e_s^i , et qui, par conséquent, sera représenté par \bar{e}_s^i . Le changement d'état entraînant le constituant élémentaire dans l'état représenté par \bar{e}_s^i sera donc constaté par un C-EVENEMENT du type $EV_{\bar{e}}^i$.

Soit $EV_a_k^{ij}$ le C-EVENEMENT défini par :

$$EV_a_k^{ij} = \begin{cases} EV_e^i & \text{ssi } a_k^{ij} = e_s^i \\ EV_{\bar{e}}^i & \text{ssi } a_k^{ij} = \bar{e}_s^i \end{cases}$$

Il existe généralement un ordre d'arrivée entre certains des C-EVENEMENTS $EV_a_k^{ij}$ associés aux atomes a_k^{ij} de m_j^i . Cet ordre se traduit par un ensemble de relations de dépendance chronologique entre ces C-EVENEMENTS.

Soit $S(EV_a_k^{ij})$ l'ensemble des C-EVENEMENTS "successeurs" de $EV_a_k^{ij}$, c'est-à-dire l'ensemble des C-EVENEMENTS qui sont en dépendance chronologique avec $EV_a_k^{ij}$ et qui sont associés à un atome de m_j^i :

$$S(EV_a_k^{ij}) = \{EV_a_r^{ij} / (EV_a_r^{ij} \in (\xrightarrow{+})(EV_a_k^{ij}))\} \\ = \{EV_a_r^{ij} / (EV_a_k^{ij} \xrightarrow{+} EV_a_r^{ij})\}$$

De même, soit $P(EV_a_k^{ij})$ l'ensemble des C-EVENEMENTS "prédécesseurs" de $EV_a_k^{ij}$, c'est-à-dire l'ensemble des C-EVENEMENTS associés à un atome de m_j^i avec lesquels $EV_a_k^{ij}$ est en dépendance chronologique :

$$P(EV_a_k^{ij}) = \{EV_a_r^{ij} / (EV_a_r^{ij} \in (\xrightarrow{-})^{-1}(EV_a_k^{ij}))\} \\ = \{EV_a_r^{ij} / (EV_a_r^{ij} \xrightarrow{-} EV_a_k^{ij})\}$$

Nous définissons l'ensemble EVI_j^i des C-EVENEMENTS initiaux de m_j^i et l'ensemble EVT_j^i des C-EVENEMENTS terminaux de m_j^i par :

- EVI_j^i est l'ensemble des C-EVENEMENTS associés à un atome de m_j^i et n'ayant pas de prédécesseurs :

$$EVI_j^i = \{EVA_k^{ij} / P(EVA_k^{ij}) = \emptyset\}$$

- EVT_j^i est l'ensemble des C-EVENEMENTS associés à un atome de m_j^i et n'ayant pas de successeurs :

$$EVT_j^i = \{EVA_k^{ij} / S(EVA_k^{ij}) = \emptyset\}$$

Remarque :

Comme généralement il n'y a pas dépendance chronologique entre tous les C-EVENEMENTS EVA_k^{ij} associés aux atomes a_k^{ij} de m_j^i , nous aurons

$$EVI_j^i \cap EVT_j^i \neq \emptyset$$

Montrons que le produit (opération 'ET') des atomes a_k^{ij} définissant m_j^i s'exprime dans notre modèle :

$$\text{Soient } EVI_j^i = \{EVA_{\alpha_1}^{ij}, EVA_{\alpha_2}^{ij}, \dots, EVA_{\alpha_p}^{ij}\}$$

$$EVT_j^i = \{EVA_{\omega_1}^{ij}, EVA_{\omega_2}^{ij}, \dots, EVA_{\omega_q}^{ij}\}$$

Alors m_j^i est représenté par l'ensemble des cycles dynamiques menant des C-EVENEMENTS de EVI_j^i aux C-EVENEMENTS de EVT_j^i et qui participent à la traduction d'au moins un des processus associés aux transitions amont t_k^{ij} des places amont p_j^i de la transition t_i dont on cherche à traduire la condition de sensibilisation $S(t_i)$. La présence des C-EVENEMENTS de EVT_j^i est alors constatée par un C-EVENEMENT EVM_j^i (traduisant m_j^i) assurant qu'un C-OBJET OBm_j^i modifié par une C-OPERATION déclenchée par chacun des C-EVENEMENTS de EVT_j^i a atteint l'état correspondant à la présence de tous les C-EVENEMENTS de EVT_j^i :

Soit C_i l'ensemble des cycles dynamiques traduisant les processus $p_{\lambda_1}, \dots, p_{\lambda_t}$ associés aux transitions t_k^{ij} amont des places p_j^i amont de la transition t_i .

Soit $\Gamma(EVI_j^i)$ l'ensemble des C-EVENEMENTS déclencheurs d'un cycle dynamique contribuant à la définition de EVT_j^i à partir de EVI_j^i :

$$\Gamma(EVI_j^i) = \{EV_k / (\exists EVA_{\alpha_r}^{ij} \in EVI_j^i) (\exists EVA_{\omega_s}^{ij} \in EVT_j^i) (\exists C_k^\lambda \in C_i) ((EVA_{\alpha_r}^{ij} \xrightarrow{+} EV_k) \wedge \wedge (EV_k \xrightarrow{+} EVA_{\omega_s}^{ij})) \wedge (EV_k \in C_k^\lambda)\}$$

et soit $C(m_j^i)$ l'ensemble des cycles dynamiques contribuant à la définition de m_j^i :

$$C(m_j^i) = \{C_k^\lambda / (C_k^\lambda \in C_i) \wedge (EV_k \in \Gamma(EVI_j^i))\}$$

Alors, la représentation de m_j^i peut être schématisée par :

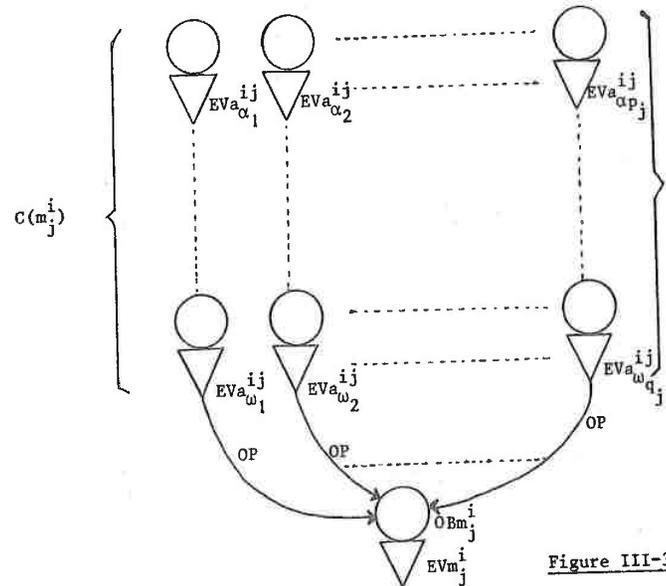


Figure III-3

- avec :
- OBm_j^i (.....,PROPCPTE,.....)
 - EVM_j^i (.....,PF,.....)
 - OP (.....,OBm_j^i,.....,TEXTE,.....)
- et :
- $PF := (PROPCPTE = q)$
 - $TEXTE := (.....; PROPCPTE := PROPCPTE + 1 ;$)

En effet, pour s'assurer de la présence de tous les C-EVENEMENTS définissant m_j^i , il suffit de s'assurer de l'arrivée de tous ceux de EVT_j^i puisque chaque C-EVENEMENT de cet ensemble est soit chronologiquement dépendant d'au moins un C-EVENEMENT de EVI_i soit à la fois élément de EVT_j^i et de EVI_j^i . Pour ce faire, chaque C-EVENEMENT de EVI_i déclenche inconditionnellement la C-OPERATION OP, qui modifie le C-OBJET OBm_j^i de manière tel que, lorsque le changement d'état événementiel de OBm_j^i constaté par EVM_j^i survient, l'on soit assuré de la présence de tous les C-EVENEMENTS de EVT_j^i (par exemple, OP peut modifier la valeur d'une propriété de OBm_j^i , par exemple en l'augmentant d'une unité, à chacun de ses déclenchements, et le prédicat final de EVM_j^i teste la valeur de cette propriété correspondant à l'arrivée de tous les éléments de EVT_j^i). Le C-EVENEMENT EVM_j^i traduit alors l'impliquant premier m_j^i : lorsqu'une occurrence de EVM_j^i survient, cela signifie que m_j^i prend la valeur 1 (ou 'VRAI').

Pour avoir représenté la condition de sensibilisation $S(t_i)$ d'une transition t_i , il nous faut encore représenter la réunion de ces impliquants premiers m_j^i :

III-2.2.2.- Expression de la réunion des impliquants premiers

Pour que la réunion des m_j^i soit vraie, il suffit que l'un des impliquants premiers m_j^i soit vrai (prenne la valeur '1'). Aussi, pour représenter $S(t_i)$ définie comme la réunion des m_j^i ($S(t_i) = \bigvee_{j=1}^{p_i} m_j^i$) il suffit que chaque C-EVENEMENT EVM_j^i (voir paragraphe III-2.2.1. ci-dessus) traduisant le fait que m_j^i prend la valeur 1 (VRAI) déclenche une C-OPERATION OP modifiant un C-OBJET $OBS(t_i)$. Le changement d'état de ce C-OBJET provoqué par la C-OPERATION OP est alors constaté par le C-EVENEMENT $EVS(t_i)$ traduisant alors la condition de sensibilisation $S(t_i)$: une occurrence de $EVS(t_i)$ traduit le fait que $S(t_i)$ prend la valeur 1 ('VRAI'). Ceci peut se représenter de la manière suivante :

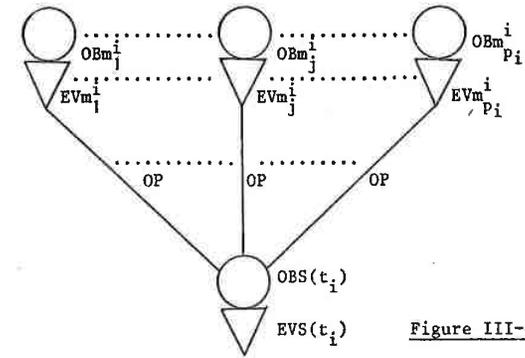


Figure III-4

- avec : - $OBS(t_i)(\dots, PROPCPTE, \dots)$
- $EVS(t_i)(\dots, OBS(t_i), \dots, PF, \dots)$
- $OP(\dots, OBS(t_i), \dots, TEXTE, \dots)$
- et : - $PF := (PROPCPTE \ 0)$
- $TEXTE := (\dots; PROPCPTE := PROPCPTE + 1; \dots)$

III-2.2.3.- Conclusion : expression de la condition de sensibilisation $S(t_i)$

Nous venons de montrer qu'une condition de sensibilisation $S(t_i)$ d'une transition t_i d'un réseau de Petri à arcs inhibiteurs est, tout au moins lorsque ce réseau est utilisé pour modéliser un système réel, exprimée par le modèle conceptuel.

En effet, chaque condition de sensibilisation $S(t_i)$ s'écrit sous forme canonique comme une réunion d'impliquants premiers, chacun de ceux-ci étant un produit d'atomes. Or nous avons montré que tout impliquant premier était représenté par notre modèle, de même que leur réunion. Donc toute condition de sensibilisation $S(t_i)$ est exprimée par notre modèle, et l'expression de cette condition peut être résumée par la figure ci-dessous, où OPR représente la C-OPERATION remettant à zéro la propriété PROPCPTE, permettant ainsi la prise en compte d'une nouvelle sensibilisation de t_i .

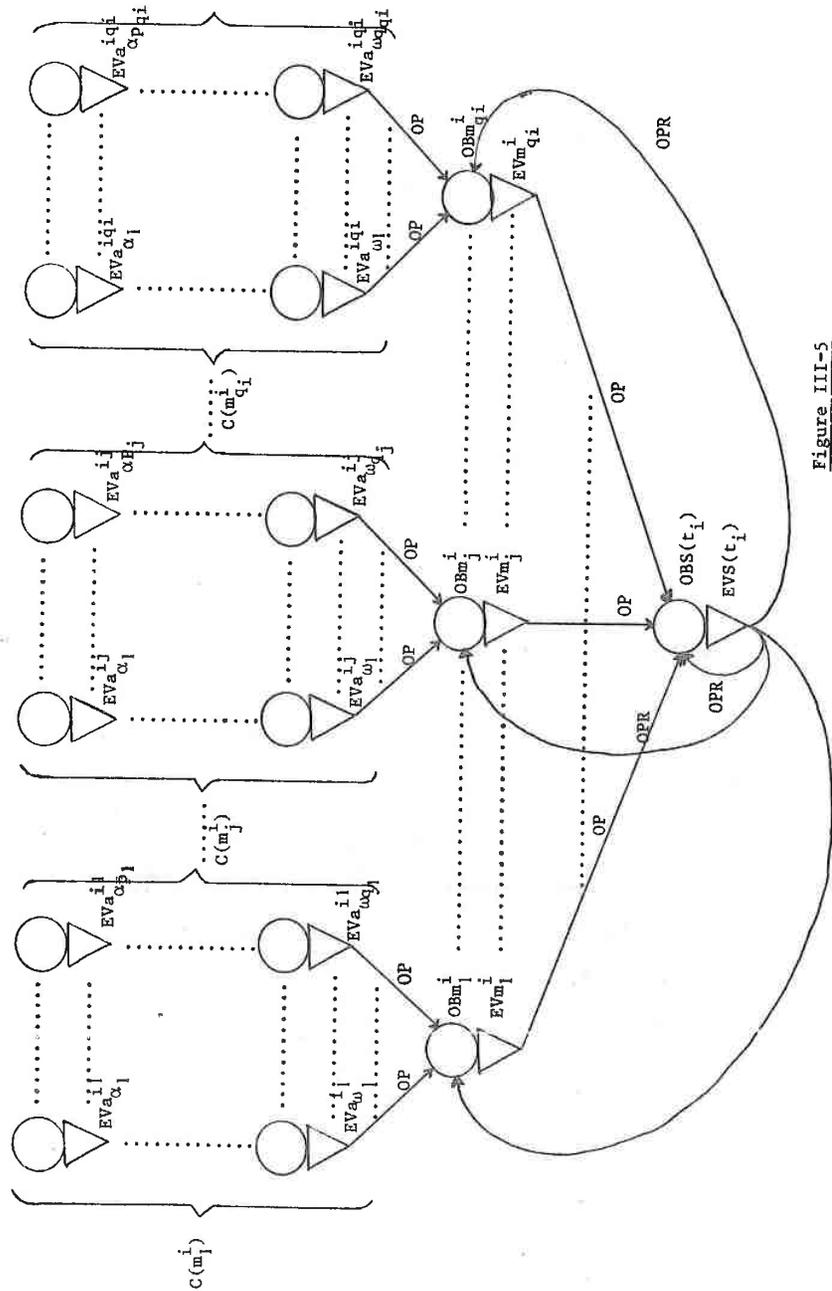


Figure III-5

III-2.3.- Expression d'un réseau de Petri à arcs inhibiteurs modélisant un système réel par le modèle conceptuel

La troisième condition à vérifier, pour montrer que tout réseau de Petri à arcs inhibiteurs modélisant un système réel est exprimé par le modèle conceptuel, est que les liens existant entre les places et les transitions d'un tel réseau sont exprimés par le modèle conceptuel. Or, comme les conditions de sensibilisation $S(t_i)$ permettent de relier une transition à un ensemble de transitions amont des places de t_i (voir paragraphe III-1.2.), nous pouvons ainsi définir un ordonnancement entre les transitions du réseau (c'est-à-dire entre les processus associés à ces transitions) qui traduise les liens existant entre les places et les transitions de ce réseau. Ceci est valable sous réserve que les notions de conflit entre deux transitions et de marquage d'une place soient effectivement exprimées par le modèle conceptuel. Examinons ces problèmes.

III-2.3.1.- Expression des problèmes de conflit entre transitions d'un réseau de Petri à arcs inhibiteurs par le modèle conceptuel

III-2.3.1.1.- Caractérisation des conflits entre transitions d'un réseau de Petri à arcs inhibiteurs

Rappelons que, dans un réseau de Petri classique (non inhibiteur) deux transitions t_i et t_j sont dites en conflit lorsque le tir de l'une de ces transitions désensibilise l'autre. De plus, ceci ne peut avoir lieu que si ces deux transitions ont au moins une place amont en commun, c'est-à-dire si

$${}^*t_i \cap {}^*t_j \neq \emptyset$$

Le problème est plus complexe dans le cas d'un réseau de Petri à arcs inhibiteurs. Bien que la définition d'une situation de conflit entre deux transitions soit la même que dans le cas d'un réseau de Petri classique ("deux transitions t_i et t_j sont en conflit lorsque le tir de l'une désensibilise l'autre"), la caractérisation des situations de conflit est plus complexe. En effet, soit p_k une place à la fois amont de t_i et de t_j toutes deux sensibilisées. Si l'un au moins des deux arcs (p_k, t_i) ou (p_k, t_j) est inhibiteur, il n'y aura pas de conflit : supposons que (p_k, t_i) soit inhibiteur, alors ou bien p_k a un marquage non nul et t_j est tirable jusqu'à ce qu'il devienne nul et t_i est ensuite tirable, ou bien le marquage de p_k est nul et t_i est alors tirable, le problème étant symétrique si (p_k, t_j) est l'arc inhibiteur ; si les deux arcs sont inhibiteurs, t_i et t_j seront tirables tant que le marquage de p_k sera nul.

Soit μ la fonction définie par :

$$\mu : A \times P \times T \rightarrow \{0, 1\}$$

$$(p_k, t_i) \longmapsto \mu(p_k, t_i) = \begin{cases} 1 & \text{si } (p_k, t_i) \text{ est un arc inhibiteur} \\ 0 & \text{sinon} \end{cases}$$

où A est l'ensemble des arcs du réseau (voir paragraphe III-1.1.1.)

La fonction μ caractérise alors la nature des arcs menant d'une place à une transition (les seuls qui peuvent être inhibiteurs).

Alors deux transitions t_i et t_j seront en conflit si et seulement si :

$$({}^i t_i \cap {}^j t_j \neq \emptyset) \wedge (\exists p_k \in {}^i t_i \cap {}^j t_j) (\mu(p_k, t_i) + \mu(p_k, t_j) = 0)$$

Le deuxième terme de cette condition traduisant le fait qu'aucun des deux arcs (p_k, t_i) et (p_k, t_j) ne doit être inhibiteur.

Soit, en généralisant :

- n transitions $t_{i_1}, t_{i_2}, \dots, t_{i_n}$ d'un réseau de Petri à arcs inhibiteurs seront en conflit si :

$$\left(\bigcap_{r=1}^n {}^i t_{i_r} \neq \emptyset \right) \wedge \left(\exists p_k \in \bigcap_{r=1}^n {}^i t_{i_r} \right) \left(\sum_{r=1}^n \mu(p_k, t_{i_r}) \leq n-2 \right)$$

Dans ce cas, supposons que $\sum_{r=1}^n \mu(p_k, t_{i_r}) = m$. Alors, seules les $(n-m)$ transitions t_{i_r} telles que $\mu(p_k, t_{i_r}) = 0$ seront en conflit puisque les m transitions t_{i_r} telles que $\mu(p_k, t_{i_r}) = 1$ seront toutes tirables tant que le marquage de p_k sera égal à zéro.

Nous avons de plus vu, au paragraphe III-1.1.1., que lorsque des transitions sont en conflit, il faut déterminer une stratégie de choix de prise en compte de celles-ci, et préciser quelles sont les transitions à tirer dans ce cas.

III-2.3.1.2.- Expression des situations de conflit entre transitions d'un réseau de Petri à arcs inhibiteurs par le modèle conceptuel

- Soient n transitions $t_{i_1}, \dots, t_{i_r}, \dots, t_{i_n}$ en situation de conflit. Comme nous l'avons exposé ci-dessus, les $n-m$ transitions t_{i_r} telles que $\mu(p_k, t_{i_r}) = 0$ sont en réelle situation de conflit, les m transitions t_{i_r} telles que $\mu(p_k, t_{i_r}) = 1$ étant tirables dès que leur condition de sensibilisation $S(t_{i_r})$ est satisfaite (VRAIE).

- Soit μ_L la fonction booléenne associée à μ et définie par :

$$\begin{aligned} \mu_L(p_k, t_i) &= V \text{ ssi } \mu(p_k, t_i) = 1 \\ \mu_L(p_k, t_i) &= F \text{ ssi } \mu(p_k, t_i) = 0 \end{aligned}$$

- Soient : $C = \bigcup_{i=1}^n C(t_{i_r})$ avec $C(t_{i_r}) = \bigcup_{j=1}^{q_{i_r}} C(m_j^{i_r})$

C est la réunion des cycles dynamiques contribuant à la définition de la condition de sensibilisation $S(t_{i_r})$ de chacune des transitions en conflit

$$EVI = \bigcup_{r=1}^n EVI_{i_r} \text{ avec } EVI_{i_r} = \bigcup_{j=1}^{q_{i_r}} EVI_j^{i_r}$$

EVI est la réunion des ensembles de C-EVENEMENTS initiaux de chaque impliquant premier de toutes les conditions de sensibilisation $S(t_{i_r})$ des transitions en conflit.

- Soit de plus :

$$\mu_L(t_{i_r}) = \bigwedge_{p_k \in \bigcap_{r=1}^n {}^i t_{i_r}} \mu(p_k, t_{i_r})$$

Alors, la situation de conflit liant les n transitions t_{i_1}, \dots, t_{i_n} sera représentée par :

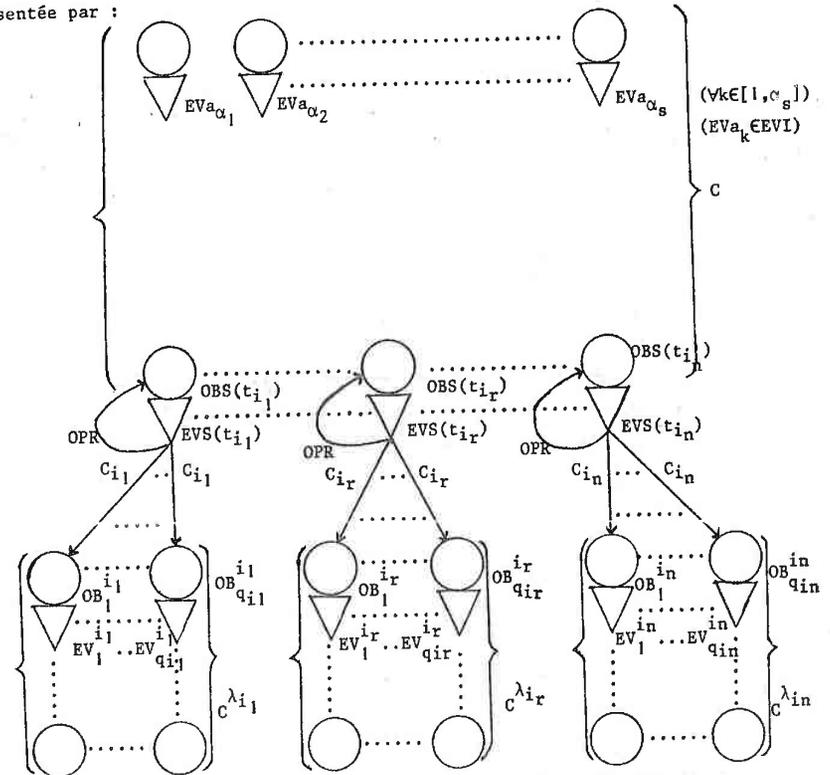


Figure III-6

avec : $C_{i_r} = C_{i_r}^i \vee \mu_L(t_{i_r})$

En effet, à partir de l'ensemble EVI des C-EVENEMENTS initiaux de chaque impliquant premier de toutes les conditions de sensibilisation $S(t_{i_r})$ des transitions en conflit, nous définirons les C-EVENEMENTS $EVS(t_{i_r})$ traduisant ces conditions de sensibilisation au moyen des enchaînements des cycles dynamiques de l'ensemble C.

La stratégie de choix des transitions t_{i_r} à prendre en compte lors d'une situation de conflit entre $t_{i_1}, \dots, t_{i_r}, \dots, t_{i_n}$ sera traduite par l'ensemble des conditions $C_{i_1}^1, \dots, C_{i_r}^1, \dots, C_{i_n}^1$ sur le déclenchement des processus $P_{\lambda_{i_1}}, \dots, P_{\lambda_{i_r}}, \dots, P_{\lambda_{i_n}}$.

De plus, pour assurer que les processus associés aux transitions t_{i_r} telles que $\mu(p_k, t_{i_r}) = 1$ soient toujours déclenchés lorsque leur condition de sensibilisation $S(t_{i_r})$ représentée par le C-EVENEMENT $EVS(t_{i_r})$ est satisfaite, la condition effective de déclenchement d'un processus $P_{\lambda_{i_r}}$ associé à une transition t_{i_r} sera :

$$C_{i_r} = C_{i_r}^1 \vee \mu_L(t_{i_r})$$

Elle sera associée au déclenchement de chacune des C-OPERATIONS conduisant chaque C-OBJET OB_j^{ij} de $I_{\lambda_{i_r}}$ (voir paragraphe III-2.1.) dans un état tel qu'il constitue une donnée du processus $P_{\lambda_{i_r}}$ associé à t_{i_r} , c'est-à-dire dans l'état événementiel constaté par le C-EVENEMENT $EV_j^{i_r}$ de $EVI_{\lambda_{i_r}}$ qui lui est associé (voir paragraphe III-2.1.).

Ainsi, nous venons de montrer que le modèle conceptuel permettait de représenter les situations de conflit entre transitions d'un réseau de Petri à arcs inhibiteurs, et qu'en plus il permettait d'exprimer la stratégie de choix des transitions à tirer, dans ce cas.

Remarque : Si l'on n'avait considéré que les transitions t_{i_r} telles que $(\forall p_k \in \bigcap_{r=1}^n t_{i_r})(\mu(p_k, t_{i_r}) = 0)$ c'est-à-dire telles que tous les arcs (p_k, t_{i_r}) reliant toutes les places amont de l'intersection des ensembles t_{i_r} à la transition t_{i_r} ne soient pas des arcs inhibiteurs, la démonstration aurait été semblable à celle présentée ci-dessus, mais les conditions de déclenchement C_{i_r} se réduiraient aux seules conditions $C_{i_r}^1$ traduisant la stratégie de choix des transitions en cas de conflit.

Pour avoir montré que tout réseau de Petri à arcs inhibiteurs modélisant un système réel est exprimé par le modèle conceptuel, il nous reste à montrer que celui-ci rend compte de la notion de marquage d'une place.

III-2.3.2.- Prise en compte de la notion de marquage d'une place par l'approche conceptuelle

La notion de marquage d'un réseau de Petri est une notion liée non plus à l'aspect description d'un système, mais à l'aspect fonctionnement du système, c'est-à-dire à l'exploitation de la description sous forme de réseau de Petri en vue de la gestion du fonctionnement de ce système. Aussi, cet aspect lié au fonctionnement se traduit par la gestion du schéma conceptuel au moyen de la machine abstraite présentée au paragraphe II-1.4. dans notre approche :

Dire qu'une place p_k d'un réseau de Petri à arcs inhibiteurs possède un certain marquage $M(p_k)$ revient à dire que :

- soit il y a eu $M(p_k)$ tirs parmi les transitions de p_k
- soit $M(p_k)$ constitue le marquage initial de p_k .

- Dans le premier cas, la réalisation des $M(p_k)$ tirs parmi les transitions de p_k va se traduire dans l'approche conceptuelle par $M(p_k)$ occurrences des C-EVENEMENTS traduisant les conditions de sensibilisation tirées. Ces $M(p_k)$ déclenchements parmi les processus associés aux transitions de p_k permettent alors d'aboutir à $M(p_k)$ occurrences des C-EVENEMENTS traduisant les conditions de sensibilisation des transitions dont p_k constitue une place amont. Ces $M(p_k)$ occurrences sont mémorisées par la machine abstraite (voir paragraphe III-1.4.) qui pourra ainsi déclencher $M(p_k)$ fois les C-OPERATIONS traduisant les processus associés à ces transitions à chaque fois que les conditions de sensibilisation sont satisfaites (chaque fois que les occurrences des autres C-EVENEMENTS nécessaires à la satisfaction d'un impliquant premier des transitions dont p_k constitue une place amont sont présentes). Ceci montre bien que l'exploitation de la traduction d'un réseau de Petri en termes de schéma conceptuel par la machine abstraite définie en II-1.4. rend compte de la notion de marquage de ce réseau.

- Dans le second cas, ce marquage traduit $M(p_k)$ déclenchements d'opérations externes au système réel, ayant amené celui-ci dans l'état traduit par le marquage M . Ceci sera exprimé dans notre modèle par $M(p_k)$ occurrences des C-OBJETS "externes" traduisant p_k dans les conditions de sensibilisation des transitions de p_k . Les conséquences de l'arrivée de ces $M(p_k)$ occurrences, conduisant à $M(p_k)$ occurrences des C-EVENEMENTS associés à ces C-OBJETS "externes" et leur exploitation par la machine abstraite sont les mêmes que dans le cas exposé ci-dessus.

III-2.2.3.- Conclusion : pouvoir de modélisation de l'approche conceptuelle

• Nous venons de montrer que le modèle conceptuel permet de représenter tout réseau de Petri à arcs inhibiteurs utilisé modélisant un système réel. (Réseau de Petri

interprété). De plus, l'exploitation de cette traduction en termes de modèle conceptuel par la machine abstraite définie en II-1.4. permet de rendre compte de la notion de marquage du réseau.

• Aussi, dans le cas de l'analyse et la modélisation d'un système réel, le modèle conceptuel a la même puissance de modélisation qu'un réseau de Petri à arcs inhibiteurs, à savoir la puissance de la Machine de Turing. Donc l'approche conceptuelle permet la modélisation de tout système réel, donc de tout système informatique.

Il reste néanmoins à remarquer qu'une modélisation conceptuelle d'un problème de synchronisation conduit à une expression conceptuelle plus simple que celle résultant de la traduction de cette modélisation en termes de réseaux de Petri.

III-3.- CONTROLES FORMELS AUTORISES PAR LA MODELISATION CONCEPTUELLE

Dans ce paragraphe, nous allons nous attacher à montrer qu'en plus des contrôles de cohérence et de complétude [85], [91] qui lui sont propres, le modèle conceptuel permet de définir des propriétés analogues à celles définies sur les réseaux de Petri sur le schéma conceptuel auquel il conduit, ou, plus précisément, sur le sous-schéma dynamique de ce schéma conceptuel. Ces propriétés seront définies par analyse du graphe des dépendances chronologiques entre C-EVENEMENTS associés à ce schéma conceptuel.

III-3.1.- Sous schéma dynamique vivant

• Définition 1

- Par analogie avec la définition de la vivacité d'un réseau de Petri (voir paragraphe III-1.1.2.), nous dirons qu'un sous-schéma dynamique est potentiellement vivant si et seulement si tout C-EVENEMENT soit est en dépendance chronologique avec au moins un C-EVENEMENT externe (c'est-à-dire n'ayant pas de prédécesseur dans le graphe des dépendances chronologiques : son demi-degré intérieur est nul), soit est lui-même un C-EVENEMENT externe et lié à un objet qui n'est modifié par aucune C-OPERATION du sous-schéma dynamique :

$$(\forall EV_i \in E) [((\exists EV_k \in E) ((d_i(EV_k) = 0 \wedge (\exists OP_\ell \in OP) (OP_\ell \rightarrow OB_k) \wedge (EV_k \xrightarrow{P^+} EV_i))) \vee ((d_i(EV_i) = 0) \wedge (\exists OP_k \in OP) (OP_k \rightarrow OB_i))]$$

- où :
- OP désigne l'ensemble des C-OPERATIONS de ce schéma
 - E désigne l'ensemble des C-EVENEMENTS de ce schéma
 - $d_i(EV_k)$ désigne le demi-degré intérieur de EV_k
 - OB_k et OB_i désignent respectivement les C-OBJETS associés à EV_k et EV_i

• Définition 2

- α) Nous dirons qu'un sous-schéma dynamique est certainement vivant si et seulement si tout C-EVENEMENT soit est en dépendance chronologique permanente avec un C-EVENEMENT externe, soit est lui-même un C-EVENEMENT externe :

$$(\forall EV_i \in E) [((\exists EV_k \in E) ((d_i(EV_k) = 0) \wedge (\exists OP_\ell \in OP) (OP_\ell \rightarrow OB_k) \wedge (EV_k \xrightarrow{P^+} EV_i))) \vee ((d_i(EV_i) = 0) \wedge (\exists OP_k \in OP) (OP_k \rightarrow OB_i))]$$

- β) Nous dirons qu'un sous-schéma dynamique est conditionnellement vivant si et seulement si tout C-EVENEMENT soit est en dépendance chronologique avec un C-EVENEMENT soit est en dépendance chronologique avec un C-EVENEMENT externe, soit est lui-même un C-EVENEMENT externe, et l'une au moins de ces dépendances chronologiques est conditionnelle :

- 1) $(\forall EV_i \in E) [((\exists EV_k \in E) ((d_i(EV_k) = 0) \wedge (\exists OP_\ell \in OP) (OP_\ell \rightarrow OB_k) \wedge (EV_k \xrightarrow{C^+} EV_i))) \vee ((d_i(EV_i) = 0) \wedge (\exists OP_k \in OP) (OP_k \rightarrow OB_i))]$
- 2) $(\exists EV_j \in E) (\exists EV_k \in E) ((d_i(EV_k) = 0) \wedge (\exists OP_\ell \in OP) (OP_\ell \rightarrow OB_k) \wedge (EV_k \xrightarrow{C^+} EV_j))$

• Intérêt de cette notion

La notion de vivacité certaine (respectivement conditionnelle) a comme dans le cas des réseaux de Petri, deux conséquences importantes :

• d'une part cela entraîne l'absence certaine (respectivement conditionnelle) de blocage puisque tout C-EVENEMENT est certainement (respectivement conditionnellement) atteint à partir d'au moins un C-EVENEMENT externe, et ainsi les C-OPERATIONS qu'il déclenche pourront être activées.

• d'autre part, il est certain qu'aucune partie du sous-schéma dynamique ne deviendra inaccessible à partir des C-EVENEMENTS externes après une certaine séquence d'arrivée d'occurrences de C-EVENEMENTS.

En conclusion, la notion de vivacité d'un sous-schéma dynamique permet d'assurer la cohérence de ce sous-schéma du point de vue de la dynamique du système. Elle peut être établie par l'analyse des composantes connexes du graphe des dépendances chronologiques entre C-EVENEMENTS associé.

Exemple : Le sous-schéma dynamique du problème simplifié de gestion des stocks présenté au paragraphe II-1.3.3. et dont le graphe des dépendances chronologiques entre C-EVENEMENTS a été présenté au paragraphe III-2.2.3. est conditionnellement vivant puisqu'il satisfait à la définition 2-β.

III-3.2.- Détection des bouclages

Dire que le système modélisé sous forme d'un schéma conceptuel est sujet à des bouclages dans son fonctionnement revient à dire qu'il existe un circuit dans le graphe des dépendances chronologiques entre C-EVENEMENTS associé au sous-schéma dynamique de ce schéma conceptuel. Ceci est à rapprocher des méthodes d'analyse matricielle des réseaux de Petri [90] garantissant l'absence de bouclages.

De plus nous pouvons remarquer qu'un sous-schéma dynamique comporte un bouclage certain (le graphe des dépendances chronologiques associé présente certainement un circuit) s'il existe un C-EVENEMENT qui est en dépendance chronologique permanente avec lui-même :

$$\begin{aligned} & (\exists EV_i \in E)(EV_i \xrightarrow{P^+} EV_i) \\ \Leftrightarrow & (\exists EV_i \in E)(EV_i \in (-\xrightarrow{P^+})(EV_i)) \end{aligned}$$

De même, un sous-schéma dynamique comporte un bouclage conditionnel (le graphe des dépendances chronologiques associé présente conditionnellement un circuit) s'il existe un C-EVENEMENT qui est en dépendance chronologique conditionnelle avec lui-même :

$$\begin{aligned} & (\exists EV_i \in E)(EV_i \xrightarrow{C^+} EV_i) \\ \Leftrightarrow & (\exists EV_i \in E)(EV_i \in (-\xrightarrow{C^+})(EV_i)) \end{aligned}$$

En conséquence, nous définirons un sous-schéma dynamique sans bouclage par :

• Définition

- α) Un sous-schéma dynamique est dit certainement sans bouclage si et seulement s'il n'existe pas de C-EVENEMENT EV_i tel que EV_i soit en dépendance chronologique (conditionnelle ou permanente) avec lui-même :

$$\begin{aligned} & (\nexists EV_i \in E)((EV_i \xrightarrow{P^+} EV_i) \vee (EV_i \xrightarrow{C^+} EV_i)) \\ \Leftrightarrow & (\nexists EV_i \in E)(EV_i \xrightarrow{+} EV_i) \end{aligned}$$

- β) Un sous-schéma dynamique est dit conditionnellement sans bouclage si et seulement s'il n'existe pas de C-EVENEMENT EV_i tel que EV_i soit en dépendance chronologique permanente avec lui-même (néanmoins, il peut y avoir un circuit conditionnel) :

$$(\nexists EV_i \in EV)(EV_i \xrightarrow{P^+} EV_i)$$

Les bouclages concernant un sous-schéma dynamique sont donc liés à l'existence de circuits permanents ou conditionnels (selon la nature des dépendances chronologiques liées à ces circuits) dans le graphe des dépendances chronologiques entre C-EVENEMENTS associé à ce sous-schéma dynamique. Ceux-ci pourront être détectés par des algorithmes classiques de recherche de circuits dans un graphe [75].

L'intérêt de la détection de ces circuits au niveau conceptuel, est de pouvoir prévenir un éventuel bouclage non souhaité au niveau de la génération des occurrences de C-EVENEMENTS déclenchant des réalisations de C-OPERATIONS provoquant la modification ou la création d'occurrences de C-OBJETS :

- la détection d'un bouclage certain au niveau des types d'événements conduit à interroger le concepteur du système modélisé pour déterminer si ce bouclage sans fin au niveau de la génération des occurrences est voulu,
- la détection d'un bouclage conditionnel à ce niveau conduit soit à poser la même question que ci-dessus, soit à demander si les conditions associées au déclenchement des C-OPERATIONS liées aux C-EVENEMENTS du circuit correspondant à ce bouclage sont telles que celui-ci prenne fin en un temps fini.

La détection des bouclages constitue donc un contrôle de la cohérence du sous-schéma dynamique.

III-3.3.- Détection des trappes

Dire que le système modélisé sous forme de schéma conceptuel recèle des trappes revient à dire que dans le graphe des dépendances chronologiques entre C-EVENEMENTS associé à ce schéma, il existe un C-EVENEMENT n'ayant pas de successeur et avec lequel aucun autre C-EVENEMENT n'est en dépendance sémantique :

Définition

- un sous-schéma dynamique est dit sans trappe si et seulement s'il n'existe pas de C-EVENEMENT EV_i tel que EV_i n'ait pas de successeur dans le graphe des dépendances chronologiques associé à ce sous-schéma et tel qu'aucun autre C-EVENEMENT ne soit en dépendance sémantique avec EV_i :

$$(\nexists EV_i \in E)[(d_e(EV_i) = 0) \wedge (\exists EV_k \in E)(EV_i \xrightarrow{DS} EV_k)]$$

où $d_e(EV_i)$ désigne le demi-degré extérieur de EV_i

L'intérêt de la détection des trappes au niveau conceptuel est de pouvoir mettre en évidence les C-EVENEMENTS constituant des trappes et d'interroger le concepteur du système pour déterminer s'il s'agit d'une sortie normale du système ou d'une

anomalie. Ceci permet d'éviter, au niveau des occurrences une accumulation d'occurrences du C-EVENEMENT et des C-OPERATIONS et C-OBJETS qui lui sont associés à partir desquels le fonctionnement du système ne pourrait plus se poursuivre (c'est-à-dire que les modifications de ces C-OBJETS ne constituent pas des événements du système).

III-3.4.- Conclusion sur les contrôles permis par le modèle conceptuel

Nous venons ainsi de montrer que, outre les contrôles de cohérence et de complétude [91] liés à sa nature relationnelle, le modèle conceptuel permettait de définir des propriétés analogues à celles des réseaux de Petri (sauf si elles portent sur le comptage d'occurrences) sur le sous-schéma dynamique auquel il conduit. Toutes ces propriétés sont définies par l'analyse des relations chronologiques entre les C-EVENEMENTS de ce sous-schéma, et par l'exploration du graphe de ces dépendances. Nous tenons également à faire remarquer que la liste des propriétés définies sur un sous-schéma dynamique au moyen des relations chronologiques entre C-EVENEMENTS présentée ici n'est pas exhaustive, notre propos ayant été de montrer que ces relations permettent la définition de telles propriétés, dont l'étude exhaustive est encore à mener à bien.

III-4.- CONCLUSION

Nous venons donc, dans ce chapitre, d'établir que le modèle conceptuel que nous proposons permet de représenter tout réseau de Petri à arcs inhibiteurs modélisant un système réel ou abstrait. De plus, l'exploitation de cette représentation par la machine abstraite associée permet de rendre compte de la notion de marquage, traduisant le fonctionnement du système associé à ce réseau.

Ainsi, l'approche conceptuelle possède la même puissance de modélisation des systèmes réels que les réseaux de Petri à arcs inhibiteurs (la puissance de la machine de Turing) et permet donc la représentation de tout système réel ou abstrait (tel que nous l'avons déjà défini au chapitre I).

De plus, les relations de dépendance chronologique entre C-EVENEMENTS permettent de définir sur le sous-schéma dynamique issu de cette modélisation des propriétés formelles analogues à celles définies sur les réseaux de Petri.

Notre objectif, dans ce chapitre, est de montrer :

- comment utiliser le modèle de la dynamique dans une approche de définition des problèmes de synchronisation.
- quels sont les avantages et les limites de cette approche par rapport aux solutions existantes que nous qualifions d'approches opératoires, comme nous l'avons exposé au chapitre I.

Nous avons retenu un certain nombre de points de comparaison à propos desquels nous discuterons des deux types d'approches sur des exemples typiques de problèmes de synchronisation.

Par ailleurs, pour éviter une trop grande diversité des formalismes, nous utiliserons systématiquement la modélisation des réseaux de Petri [30], [79], [80], [92]. Notre choix est justifié

- par la qualité de ce modèle reconnu comme un modèle d'abstraction des problèmes de synchronisation se situant parmi les modèles qui présentent la plus grande indépendance vis-à-vis des solutions techniques d'implémentation. Cette indépendance est due principalement à la définition de ces réseaux comme des outils mathématiques, en termes de graphe bi-alterné [54], [90].
- par sa puissance de modélisation, puisque les réseaux de Petri à arcs inhibiteurs [1] possèdent la puissance de modélisation de la machine de Turing [1], [78].
- par sa notoriété qui explique la multiplicité de ses utilisations : les réseaux de Petri semblent être un des modèles les plus utilisés parmi ceux possédant un haut niveau d'abstraction et d'indépendance vis-à-vis des solutions techniques. En effet, leur champ d'application est très vaste et va de l'étude de langages formels [23], [24], [44], [45] et l'écriture de compilateurs [5] jusqu'à la conception des systèmes d'information [49], [50], [64] en passant par l'analyse et la conception de hardware [29], [73] ou de software [31], [43], [62].

Remarquons enfin que des notions telles que celles de moniteur [46], [47] ou de semaphore [33], [34] nous semblent être d'excellents moyens pour contrôler et gérer le fonctionnement de systèmes physiques constituant des solutions à des problèmes de synchronisation, mais ils ne nous semblent pas représenter des outils de modélisation de ces problèmes eux-mêmes (chapitre I). C'est pourquoi nous ne baserons pas notre comparaison sur ce type de notions.

IV-1.- L'APPROCHE CONCEPTUELLE : UNE ANALYSE SEMANTIQUE

Pour cette évaluation comparative, nous nous appuyons sur l'exemple d'un système producteur-consommateur avec tampon unique [25].

IV-1.1.- L'approche opératoire

Comme nous l'avons présenté au paragraphe I-3.2., il est difficile d'associer une méthode d'analyse aux modélisations de type opératoire. Devant chaque problème nouveau, le concepteur est sensé trouver le raisonnement qui le conduit de manière heuristique à la solution. Les réseaux de Petri n'échappent pas à cette constatation.

Ainsi qu'il a été présenté au chapitre I lors d'un essai de synthèse des méthodes associées aux approches opératoires, on peut constater que ce type d'analyse commence en général par la détermination des fonctions à assurer par le système, et de leurs interrelations entre ces fonctions. En ce sens, les approches opératoires constituent des approches "fonctionnelles". Ensuite on s'interroge sur la traduction de ces fonctions en termes de processus, et sur la traduction de leurs interrelations en termes de synchronisation entre ces processus, ce qui conduit à définir plus précisément les processus du système, en cherchant leurs états remarquables et en exprimant le passage d'un état à un autre. Néanmoins, il ne nous semble pas exister de méthode réellement systématique d'analyse.

Ainsi, dans le cas du système producteur-consommateur, cette démarche nous conduit à

- déterminer deux fonctions : "Produire" et "Consommer",
- définir les interrelations entre ces fonctions :
 - la fonction "Produire" peut intervenir lorsque la fonction "Consommer" a vidé le tampon des ressources qu'il contenait.
 - la fonction "Consommer" peut intervenir lorsque la fonction "Produire" a garni le tampon des ressources à consommer.
- traduire chacune de ces deux fonctions par un processus. Ainsi le système producteur-consommateur avec tampon unique sera constitué par :
 - deux processus (Producteur et Consommateur), et deux états remarquables de chacun de ces deux processus (attente et activité) apparaissent à la suite de l'analyse de ce problème, conformément à la méthode d'analyse opératoire présentée au chapitre I.

- Le processus "Consommateur" atteint l'état d'"attente" dès qu'il n'y a plus de ressources à consommer dans le tampon. Il atteint l'état de "consommation" dès qu'il y a des ressources à consommer dans le tampon.
- Le processus "Producteur" atteint l'état d'"attente" dès que le tampon contient des ressources. Il atteint l'état de "production" dès que les ressources contenues dans le tampon ont été consommées.

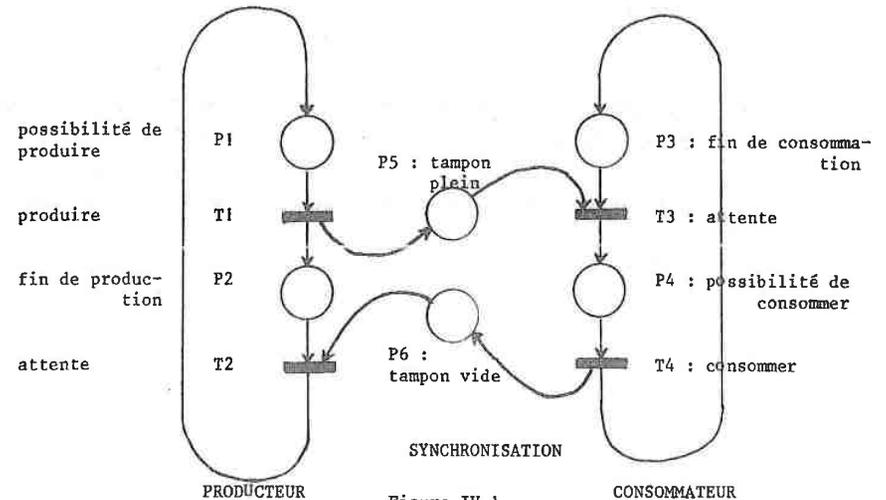
- il apparaît donc, après analyse des états, que le tampon qui contient les ressources produites et qui doivent être consommées est la ressource partagée entre ces deux processus. Il conditionne la synchronisation de ces processus :

- Ainsi, ces deux processus coopèrent au moyen de messages traduisant l'état du tampon (vide ou plein). De plus, ces messages valident les conditions nécessaires au déclenchement des actions de production ou de consommation. Ces messages et ces conditions sont exprimées par les places du réseau de Petri associé à ce système. Ceci correspond à la condition 3 de prise en compte de la synchronisation, et à la traduction des interrelations entre les fonctions traduites par ces processus.

- De plus, ces deux processus sont dépendants l'un de l'autre (réponse à la condition 2 : définir les interrelations entre les processus).

- Enfin, il n'y a que ces deux processus dans le système producteur-consommateur (réponse à la condition 1 : définir précisément l'ensemble des processus du système).

Donc la solution en termes de réseau de Petri au problème de la modélisation d'un système Producteur-Consommateur avec tampon unique est donc la suivante :



IV-1.2.- L'approche conceptuelle

Au contraire, on peut définir une méthode d'analyse associée au modèle de la dynamique, ainsi que nous l'avons montré au paragraphe II-1.3. Rappelons que cette méthode analyse un système à partir de sa structure (approche structurelle) que l'on détermine en trois étapes :

- détermination des objets du système, et représentation de ces objets en termes de types (C-OBJETS).
- détermination des changements d'états de ces objets qui constituent les événements du système et représentation de ces événements en termes de types (C-EVENEMENTS).
- détermination des opérations qui sont les conséquences de ces changements d'état événementiels et représentation de ces opérations en termes de types (C-OPERATIONS).

Elle conduit donc sur cet exemple à déterminer

1) les objets du système, représentés par des C-OBJETS et à n'en trouver qu'un seul type : les objets du type "PRODUIT" (produits et consommés par ce système) représentés par le C-OBJET "PRODUIT".

2) les changements d'état remarquables de cet unique C-OBJET qui constituent des C-EVENEMENTS :

- le passage de la propriété "QTE-STOCK" du C-OBJET "PRODUIT" à la valeur zéro, constituant le C-EVENEMENT EV1.
- le passage de la propriété "QTE-STOCK" au niveau N, constituant le C-EVENEMENT EV2.

3) les conséquences de chaque C-EVENEMENT :

- le C-EVENEMENT EV1 déclenche la C-OPERATION OP1 "PRODUCTION" qui fournit une quantité N de ce produit.
- le C-EVENEMENT EV2 déclenche la C-OPERATION OP2 "CONSOMMATION" qui consomme une quantité N de ce produit.

Ainsi, le sous-schéma dynamique du système Producteur-Consommateur avec tampon unique est donc :

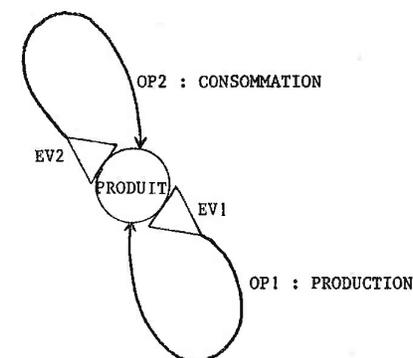


Figure IV-2

IV-1.3.- Commentaires

A la suite de l'examen des deux modélisations ci-dessus d'un même problème, nous pouvons faire deux remarques immédiates

Remarque 1 : La description conceptuelle de ce système met en évidence la sémantique de celui-ci, qui est finalement assez simple :

"Ce système a un seul type de ressource qu'il consomme dès que le niveau de ressource a atteint un certain seuil N et il produit une quantité N de cette ressource dès que le niveau a atteint la valeur zéro".

Remarque 2 : La modélisation au moyen des réseaux de Petri oblige à définir le problème au moyen des places et des transitions auxquelles n'est associée aucune sémantique, contrairement aux concepts de notre approche, car ils ne représentent aucun composant du système réel. En effet, selon les cas ils peuvent représenter différentes choses : une place, par exemple, peut soit représenter une condition de synchronisation, un état du système, ... Ce fait contribue à rendre la description d'un problème par un réseau de Petri (qui donc est un réseau de Petri interprété) moins directement compréhensible que la description conceptuelle du même problème, de par le fait qu'un même concept du réseau (la place, par exemple) puisse avoir plusieurs sémantiques différentes associées en fonction de l'interprétation liée à ce réseau.

IV-1.4.- Conclusions

Bien qu'ils affirment leur indépendance vis-à-vis des solutions techniques, les modèles du type opératoire ont été initialement conçus pour l'analyse et la modélisation de systèmes technologiques en fonctionnement. De notre point de vue, il en

résulte qu'ils permettent la modélisation de mécanismes de synchronisation qui constituent une solution à un problème posé par une situation de synchronisation donnée, et non pas la modélisation de cette situation elle-même. De plus, ils obligent à définir un problème de synchronisation au moyen des concepts sur lesquels ils reposent, et ceux-ci ne représentent jamais des constituants du système réel. Ainsi la définition de la coopération entre les processus est posée, à un degré plus ou moins important, en fonction des solutions "techniques" de synchronisation disponibles (par exemple envoi de "messages" par l'intermédiaire de places dans un réseau de Petri, comme dans le cas du système producteur-consommateur).

On peut faire le parallèle avec les approches développées dans le domaine des bases de données. Initialement, les données ont été définies en fonction des méthodes d'accès disponibles. Aujourd'hui néanmoins, personne ne conteste l'intérêt d'une description sémantique [4], y compris pour déterminer les méthodes d'accès les plus adéquates à sa gestion.

Au contraire, par la nature même du processus de recherche qui a conduit à la définition du modèle dynamique (partir de l'analyse des constituants des systèmes réels), l'approche conceptuelle conduit à une analyse sémantique des problèmes de synchronisation. En effet, basée sur une analyse causale de la réalité, elle permet de décrire la sémantique d'un problème, c'est-à-dire qu'elle permet une description de la logique de ce problème exprimant la situation de synchronisation elle-même. Ainsi, elle autorise une description des opérations réelles participant à la définition de cette situation de synchronisation indépendante de toute idée de solution technique. Donc, l'approche conceptuelle apparaît comme indépendante de toute contrainte technologique ou relative à l'usage du système modélisé, d'ailleurs les outils associés à cette modélisation (la machine abstraite) ont été définis après le modèle, et non avant.

IV-2.- SIMPLICITE DE L'APPROCHE CONCEPTUELLE

L'exemple précédent met en évidence la facilité de la méthode et la simplicité de la description conceptuelle. Prenons, pour illustrer la simplicité de l'approche conceptuelle, un autre exemple classique de synchronisation : le problème des philosophes affamés.

IV-2.1.- Le problème

On trouvera une définition complète du problème dans [35]. Rappelons les grandes lignes de ce problème :

- Plusieurs philosophes sont réunis autour d'une table pour philosopher et, accessoirement, manger des spaghettis.
- Il y a autant de fourchettes que de philosophes.
- Chaque philosophe a besoin de deux fourchettes pour manger ses spaghettis, aussi ne peut-il manger que si la fourchette située immédiatement à sa droite et celle située immédiatement à sa gauche sont toutes deux disponibles.
- A chaque instant, un philosophe est donc dans l'un des trois états suivants :
 - il mange,
 - il veut manger, mais les deux fourchettes dont il a besoin ne sont pas toutes les deux disponibles, aussi attend-il qu'elles soient toutes deux disponibles,
 - il pense.

IV-2.2.- Application de la méthode opératoire

L'application de l'analyse opératoire à ce problème conduit à considérer chaque philosophe comme une fonction à assurer par le système : il doit permettre le "fonctionnement" des philosophes. Les interrelations entre ces fonctions sont assurées par les fourchettes situées à la droite et à la gauche de chaque philosophe : les fonctions traduisant les philosophes évoluent (traduisent l'état de manger, penser ou attendre) conditionnellement aux états de ces fourchettes.

Ainsi, chacune de ces fonctions sera traduite par un processus "philosophe", processus cyclique ayant deux états d'activité : manger et penser. De plus, ce processus cyclique a un état implicite correspondant à l'attente de manger. Chacun de ces états d'activité sera représenté par une transition, conformément à la pratique courante lors de l'analyse d'un système par un réseau de Petri. Les conditions de passage d'un de ces états d'activité à un autre, traduisant les interrelations entre les fonctions exprimant le comportement des philosophes, seront représentées chacune par une place du réseau. Ces conditions sont :

- pour passer de l'état de penser à l'état de manger :
 - 1) Le philosophe a fini de penser.
 - 2) Les deux fourchettes susceptibles d'être utilisées par ce philosophe sont disponibles.
- pour passer de l'état de manger à celui de penser :
 - 3) Le philosophe a fini de manger.

De plus, les conditions sur les fourchettes doivent être mises à jour, c'est-à-dire que, quand un philosophe commence à manger, il doit signaler que les fourchettes qu'il utilise ne sont plus disponibles, et, enfin, il doit signaler que ces fourchettes sont disponibles dès qu'il a fini de manger.

Enfin l'état d'attente est implicitement géré par le mécanisme inhérent aux réseaux de Petri : ce mécanisme ne permet le tir d'une transition que si toutes les places "amont" ont été marquées, c'est-à-dire si et seulement si les conditions associées à ces places ont été satisfaites (voir paragraphe III-1.1.).

Ainsi, le processus "Philosophe numéro i " sera représenté par le réseau de Petri suivant :

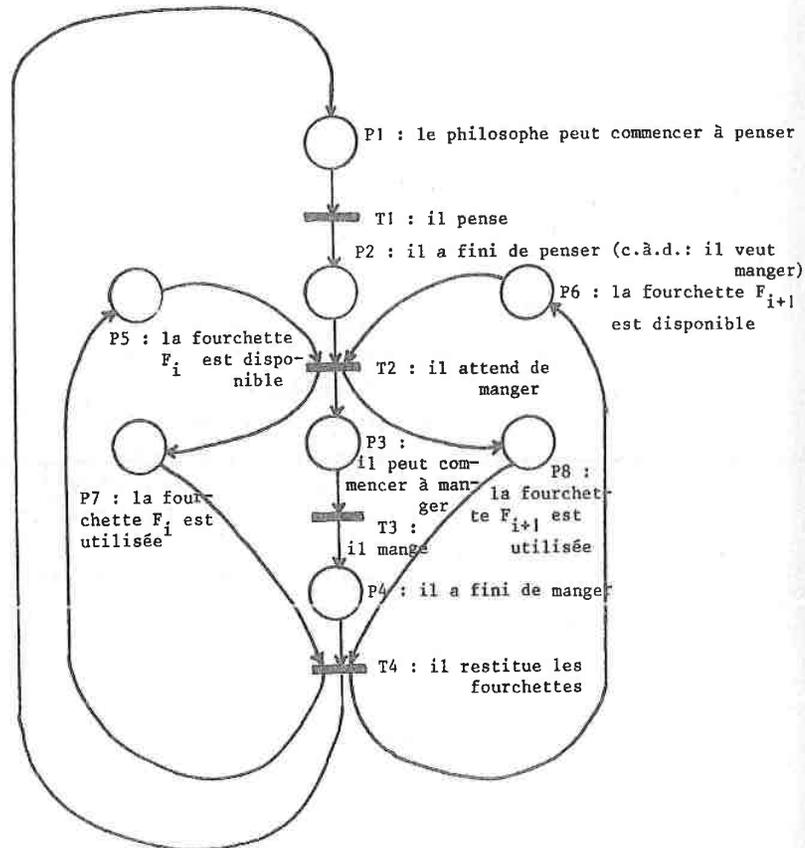


Figure IV-3

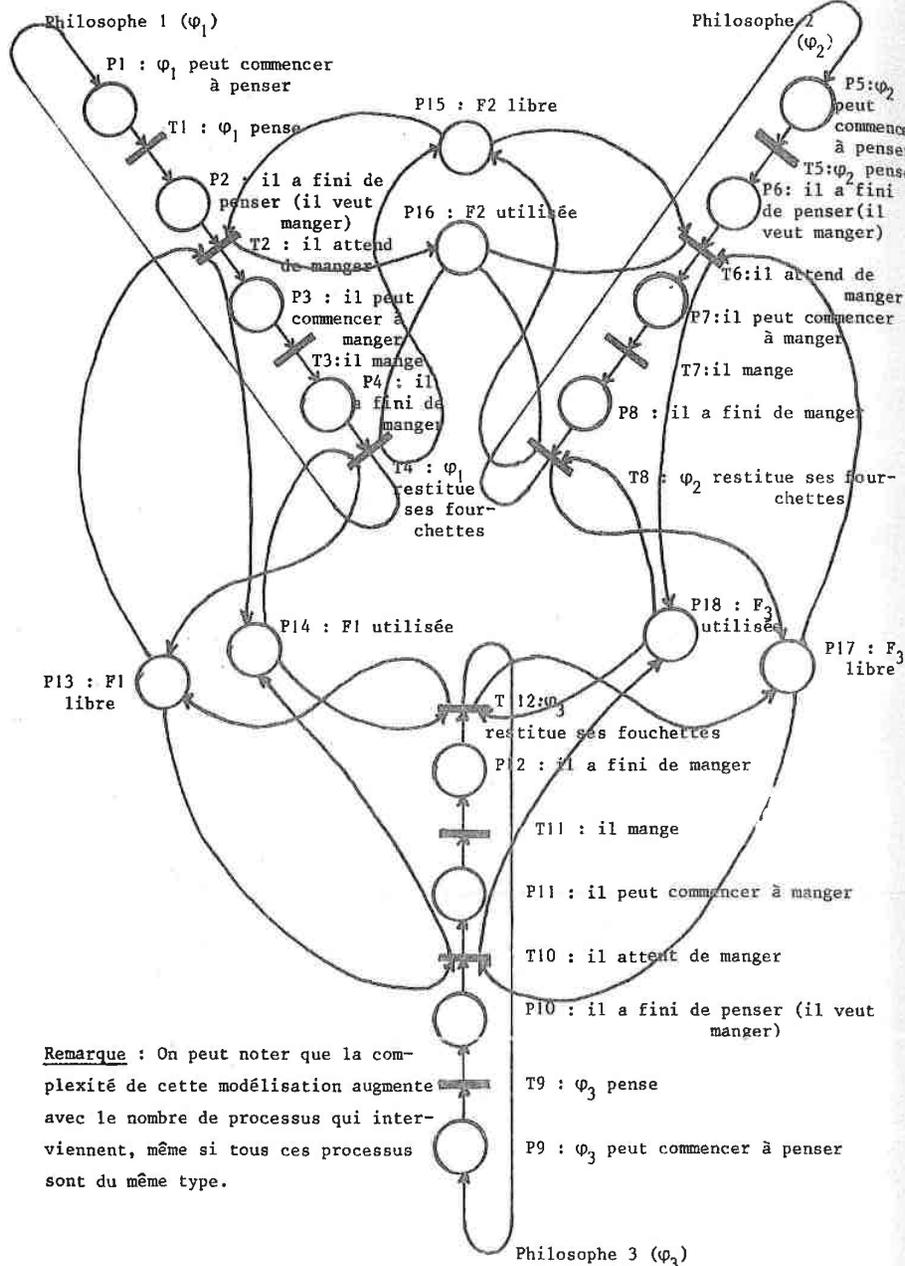
Ceci décrit uniquement le fonctionnement d'un seul processus "Philosophe". Les problèmes de synchronisation apparaissent lorsque le système comporte plusieurs philosophes : une fourchette peut alors être utilisée puis libérée par deux philosophes, exprimant ainsi les interrelations entre les fonctions exprimant le comportement d'un philosophe réel :

1) ainsi, l'ordonnement entre les exécutions des différents processus "philosophe n° i " du système se fera selon l'ordre de libération des fourchettes (traduit par le marquage des places associées), correspondant à la condition 3 de prise en compte de la synchronisation.

2) De plus, comme chaque processus "philosophe n° i " est dépendant des deux processus "philosophe n° $i-1$ " et "philosophe n° $i+1$ " voisins, tous les processus du système sont dépendants entre eux par transitivité de la relation de dépendance, ceci correspondant à la condition 2 de prise en compte de la synchronisation.

3) Enfin, en conséquence de ce que nous venons d'exposer, il nous faut définir autant de processus "Philosophe n° i " qu'il n'y a de philosophes dans notre système, afin de définir explicitement la communication entre ceux-ci (la libération ou la saisie des fourchettes associées), ceci correspondant à la condition 1 de prise en compte de la synchronisation.

Enfin, par exemple, la coopération entre trois processus du type "philosophe n° i " correspondant au problème de trois philosophes affamés sera représenté par le réseau de Petri suivant :



Remarque : On peut noter que la complexité de cette modélisation augmente avec le nombre de processus qui interviennent, même si tous ces processus sont du même type.

Figure IV-4

IV-2.3.- Application de l'approche conceptuelle

La méthode conceptuelle conduit à déterminer :

1) les objets du système, qui sont au nombre de trois et à les décrire par trois relations de type C-OBJET

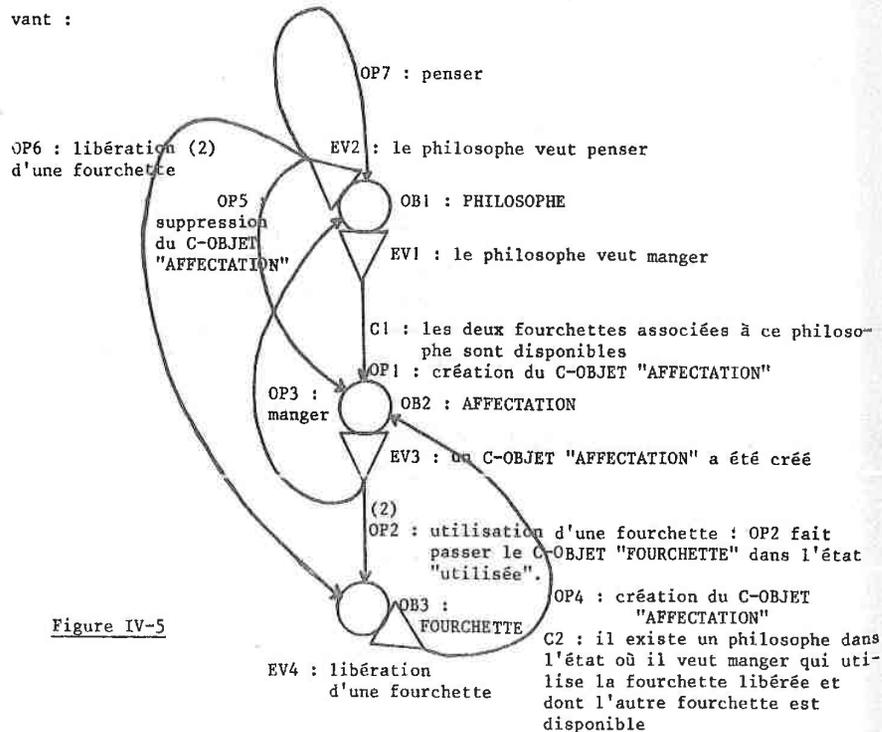
- PHILOSOPHE (NOPHI, ETATPHI, NOF1, NOF2)
où NOPHI identifie chaque philosophe
ETATPHI exprime le fait qu'un philosophe est dans l'état où il veut manger ou dans l'état où il veut penser,
NOF1, NOF2 identifient les deux fourchettes qui lui sont nécessaires pour pouvoir manger.
- FOURCHETTE (NOF, ETATF)
où NOF identifie la fourchette
ETATF exprime l'utilisation ou la disponibilité d'une fourchette.
- AFFECTATION (NOPHI, NOF1, NOF2)
cette relation décrit l'association entre un philosophe et les deux fourchettes dont il a besoin pour pouvoir manger.

2) les changements d'état remarquables des C-OBJETS constituant des C-EVENEMENTS, ainsi que les conséquences de chaque C-EVENEMENT :

- EV1 : le passage du C-OBJET 'PHILOSOPHE' dans l'état où il veut manger est un C-EVENEMENT EV1 qui déclenche la C-OPERATION OP1 de création d'un C-OBJET 'AFFECTATION' traduisant l'attribution à ce philosophe des deux fourchettes qui lui sont nécessaires pour manger, si elles sont disponibles (condition C1), c'est-à-dire que ce philosophe peut alors manger.
- EV2 : le passage du C-OBJET 'PHILOSOPHE' dans l'état où il veut penser est un C-EVENEMENT EV2 qui déclenche les C-OPERATIONS OP5 de suppression d'un C-OBJET 'AFFECTATION', OP6 de libération des fourchettes utilisées et OP7 qui fait passer le philosophe dans l'état où il veut manger. OP5 et OP6 correspondent à la restitution des ressources et OP7 correspond à "penser" puisque le philosophe passe dans l'état où il veut manger après avoir pensé.

- EV3 : la création du C-OBJET 'AFFECTATION' est un C-EVENEMENT EV3 qui déclenche deux fois la C-OPERATION OP2 qui fait passer un objet du type 'FOURCHETTE' dans l'état "utilisée", et il déclenche la C-OPERATION OP3 qui fait passer le C-OBJET 'PHILOSOPHE' dans l'état où il veut penser (ce qui est équivalent à "manger" puisqu'il veut penser après avoir mangé). Ces deux C-OPERATIONS représentent le fait que le Philosophe mange.
- EV4 : le passage du C-OBJET 'FOURCHETTE' dans l'état "disponible" est un C-EVENEMENT EV4 qui déclenche la C-OPERATION OP4 de création d'un C-OBJET 'AFFECTATION' si et seulement s'il existe un philosophe qui veut utiliser cette fourchette pour manger et dont l'autre fourchette est disponible (condition C2). Cette C-OPERATION OP4 correspond donc à la recherche d'un philosophe que la fourchette qui vient d'être libérée permettrait de satisfaire.

Le sous-schéma dynamique du problème des philosophes affamés est donc le suivant :



IV-2.4.- Commentaires

- L'exemple des philosophes affamés fait apparaître la relative lourdeur de la description d'un problème de synchronisation complexe en termes de réseaux de Petri. Cette lourdeur nous semble attachée à la nécessité d'effectuer l'inventaire de toutes les situations dans lesquelles la libération des deux fourchettes utilisables par un philosophe lui permet de manger. Ceci conduit à une description qui prend en compte le nombre de philosophes, et qui ainsi devient de plus en plus complexe, en fonction de l'augmentation du nombre de philosophes : l'exemple des trois philosophes affamés contient six transitions (T2, T4, T6, T8, T10, T12) et six places (P13, P14, P15, P16, P17, P18) de synchronisation qui expriment, pour chacun des trois philosophes le fait qu'il peut manger dès que les deux fourchettes qui lui sont nécessaires sont disponibles.

De plus, dans une analyse de type opératoire (illustrée ici par les réseaux de Petri), l'absence d'un concept équivalent à celui d'objet conduit à raisonner exclusivement sur les processus et leurs états. Dans ce cas, il nous faut décrire les règles de comportement de chaque philosophe au moyen de quatre places (P1, P2, P3, P4) et de quatre transitions (T1, T2, T3, T4) ainsi que nous l'avons exprimé sur le réseau de la figure IV-3. De plus, il nous faut déterminer, pour chaque transition, si elle est indépendante ou non des deux philosophes adjacents, cette dépendance étant exprimée par les places (P5, P6, P7, P8) traduisant l'état des fourchettes que ce philosophe peut utiliser.

- Dans l'approche conceptuelle, la description est plus simple, pour deux principales raisons :

- la description conceptuelle est indépendante du nombre de philosophes, contrairement aux descriptions en termes opératoires, puisque le raisonnement conceptuel est basé sur la notion de types : type d'objets (C-OBJET), type d'événements (C-EVENEMENT) et type d'opérations (C-OPERATION). Dans cette approche, un philosophe particulier est une occurrence du C-OBJET "PHILOSOPHE".

- le concept d'événement causal (représenté par un C-EVENEMENT) permet de décrire une fois et une seule une situation de déclenchement d'opérations (représentées par les C-OPERATIONS) même si plusieurs séquences différentes d'opérations conduisent à cette situation par des chemins différents. Ainsi, la C-OPERATION OP3 "manger" est déclenchée par le C-EVENEMENT EV3 constatant un changement d'état du C-OBJET "AFFECTATION" qui peut être provoqué soit par OP1, soit par OP4, alors que la description des conditions permettant à un philosophe de manger nécessite quatre places (P2, P5, P6, P3) et une transition (T2) dans une approche opératoire telle que les réseaux de Petri.

Il convient de remarquer que le sous-schéma dynamique du problème des philosophes affamés permet bien d'assurer la synchronisation des processus philosophes du système :

- lorsqu'un philosophe veut manger (EV1), ou bien les deux fourchettes dont il a besoin sont disponibles (C1) et il peut commencer à manger (OP3) après qu'on lui ait affecté ces fourchettes (OP1) qui deviennent indisponibles (OP2), ou bien ces deux fourchettes ne sont pas disponibles et il attend donc pour manger que ses deux fourchettes se libèrent (EV4), ce qui conduit à les lui affecter (OP4) et ainsi il peut manger.
- lorsque ce philosophe veut penser (EV2), il libère les deux fourchettes qu'il utilisait (OP6), il signale qu'elles ne lui sont plus attribuées (OP5) et il commence à penser (OP7).

Ceci décrit la sémantique du problème des philosophes affamés, mais ne permet pas d'assurer que tout philosophe pourra manger à un moment donné. En effet, l'analyse conceptuelle d'un système est la traduction des règles gouvernant ce système, et, dans le cas du problème qui nous préoccupe, les règles qui permettraient d'assurer l'absence de famine ne sont pas contenues dans la sémantique de ce problème, et donc ne peuvent pas apparaître dans le schéma conceptuel. (Notons d'ailleurs que nombre de modélisations de ce problème [25] présentent la même caractéristique).

IV-2.5.- Conclusions

Les exemples du système producteur-consommateur et du problème des philosophes affamés mettent en évidence à la fois la facilité de modélisation et la simplicité de la description d'un problème de synchronisation par l'approche conceptuelle par rapport à la lourdeur de l'analyse et de la description issues d'une approche opératoire.

En fait, plus le problème est complexe, moins la solution fournie par une approche opératoire est facile à déterminer, et plus la description de ce problème en termes opératoires devient lourde et difficile à comprendre. Ces difficultés de modélisation par une approche opératoire sont de notre point de vue dues à :

1) la nature purement abstraite de certains concepts (place ou transition dans la cas des réseaux de Petri) ayant certes une puissance de modélisation très importante (puissance de la machine de Turing dans le cas des réseaux de Petri à arcs inhibiteurs par exemple), mais dépourvus d'une sémantique propre qui puisse faciliter le raisonnement du concepteur et le conduire à une solution simple.

2) la nécessité de faire l'inventaire de toutes les situations de synchronisation à l'aide de concepts non clairement définis : processus et état d'un processus.

3) l'obligation de raisonner de manière privilégiée par rapport aux processus. Ceci est une conséquence de l'aspect "fonctionnel" des approches opératoires, consistant à analyser un système à partir des fonctions qu'il doit assurer, ces fonctions étant ensuite réalisées par des processus. En particulier ce type d'analyse conduit à privilégier les processus par rapport aux ressources, celles-ci étant perçues uniquement par leurs liens avec ces processus, comme nous l'avons vu au chapitre I. Dans le fond, cela montre l'insuffisance des concepts sur lesquels reposent ces types de raisonnement et en particulier cela met en évidence l'absence d'un concept équivalent à celui d'objet dans l'approche conceptuelle.

Au contraire, l'approche conceptuelle permet une analyse plus facile et conduit à une définition simple d'un problème de synchronisation sous forme de schéma conceptuel. Les avantages de cette approche, du point de vue de la simplicité de la description et de la facilité d'analyse, résultent de la simplicité et de l'évidence sémantique des concepts, ainsi que des relations de causalité entre ces concepts (le changement d'état d'un C-OBJET est un C-EVENEMENT qui déclenche des C-OPERATIONS provoquant chacune le changement d'état d'un C-OBJET qui peut constituer un C-EVENEMENT) :

- la simplicité parce que le modèle conceptuel est basé sur seulement trois concepts [40], [87].
- l'évidence sémantique parce que la méthode de définition des concepts de ce modèle, basée sur l'idée de traduction des phénomènes réels en termes de types (voir chapitre II, paragraphe II-1.1.) permet leur rattachement aux éléments d'un système réel [39], [40].
- la causalité parce que la définition des concepts basée sur la traduction des phénomènes réels permet de définir des associations entre ces concepts qui expriment les interrelations causales entre les phénomènes réels [39], [40], [87] :
 - association CONSTATE liant un C-EVENEMENT et un C-OBJET
 - association DECLENCHE liant un C-EVENEMENT et une ou plusieurs C-OPERATIONS
 - association MODIFIE liant une C-OPERATION et un C-OBJET.

La simplicité, l'évidence sémantique et la causalité facilitent l'analyse du concepteur et lui permettent une plus grande maîtrise de la conception. La causalité

et l'élémentarité, liée à la définition des concepts dans une forme normalisée [39], [40] assurant l'élimination de toute redondance au niveau du schéma conceptuel, constituent également des facteurs de simplicité du résultat de la modélisation représentant le résultat de la démarche conceptuelle. En effet, elles permettent de réduire le nombre de situations à décrire, comme nous l'avons remarqué sur le problème des philosophes affamés (cas de la C-OPERATION OP3 qui peut être déclenchée soit après OP1, soit après OP4).

En effet, dans la structure conceptuelle, le nombre de situations de synchronisation décrites est plus grand que le nombre de C-EVENEMENTS du schéma conceptuel qui les décrit car le même état élémentaire peut contribuer à la définition de plusieurs états remarquables du système qui auraient été considérés comme des primitives de description de ces situations de synchronisation par une approche opératoire. De plus, comme la structure conceptuelle décrit les changements d'état élémentaires du système et les passages élémentaires d'un état à un autre sans référence au temps, elle permet de reconstruire plusieurs situations de synchronisation temporelles par différentes combinaisons de cycles dynamiques interconnectés par le biais des C-EVENEMENTS, sans que le concepteur ait à en faire un inventaire exhaustif, comme nous l'avons montré au chapitre II, paragraphes II-2.1. et II-3. (utilisation des relations de dépendance chronologique entre C-EVENEMENTS et des relations de causalité entre les concepts du modèle). Au contraire, dans une approche opératoire, celui-ci doit raisonner en tenant compte du temps et il doit décrire toutes les séquences temporelles d'enchaînement de processus correspondant à ces situations de synchronisation.

IV-3.- LA RIGUEUR DE LA DEMARCHE CONCEPTUELLE PERMET UNE MEILLEURE MAITRISE DES PROBLEMES DE CONCEPTION

Nous nous appuyons, dans ce paragraphe, sur l'exemple du domaine des organisations, concernant le traitement d'une ligne de commande acceptée, tel qu'il a été présenté au paragraphe III-2.2.1.

IV-3.1.- Le problème

Rappelons l'énoncé de ce problème, qui constitue une partie du système de vente par correspondance présenté au paragraphe II-1.2.

Le traitement d'une ligne de commande acceptée se déroule comme suit :

- lorsqu'une ligne de commande concernant un produit a été acceptée, on livre le produit et on met à jour le stock de ce produit si celui-ci est suffisant pour en satisfaire la commande, sinon on diffère le traitement de cette ligne de commande.

- lorsqu'une rupture de stock sur un produit a été constatée, il y a émission d'un ordre de réapprovisionnement.
- le réapprovisionnement du stock d'un produit implique l'essai de prise en compte des lignes de commande différées à cause de l'insuffisance du stock de ce produit.

IV-3.2.- L'approche opératoire

Dans une analyse opératoire de ce problème apparaissent cinq fonctions principales :

- analyse d'une ligne de commande (F1)
- gestion des produits, constituée par trois fonctions :
 - mise à jour du stock (F2)
 - émission de demande de réapprovisionnement (F3)
 - réapprovisionnement (F4)
- reprise des lignes de commande en attente après réapprovisionnement (F5).

Chacune de ces fonctions F_i sera représentée par un processus \mathcal{P}_i .

- les interactions entre ces fonctions, conditionnant la synchronisation entre les processus les représentant, sont les suivantes :
 - lorsqu'une ligne de commande a été déterminée par F1 comme pouvant être livrée, la fonction F2 doit être activée.
 - lorsqu'une mise à jour du stock par F2 entraîne une rupture de stock, la fonction F4 doit être activée.
 - lorsqu'un réapprovisionnement par la fonction F4 a eu lieu, la fonction F5 doit être activée.

Une telle analyse conduit à la représentation de ce problème par le réseau de Petri suivant :

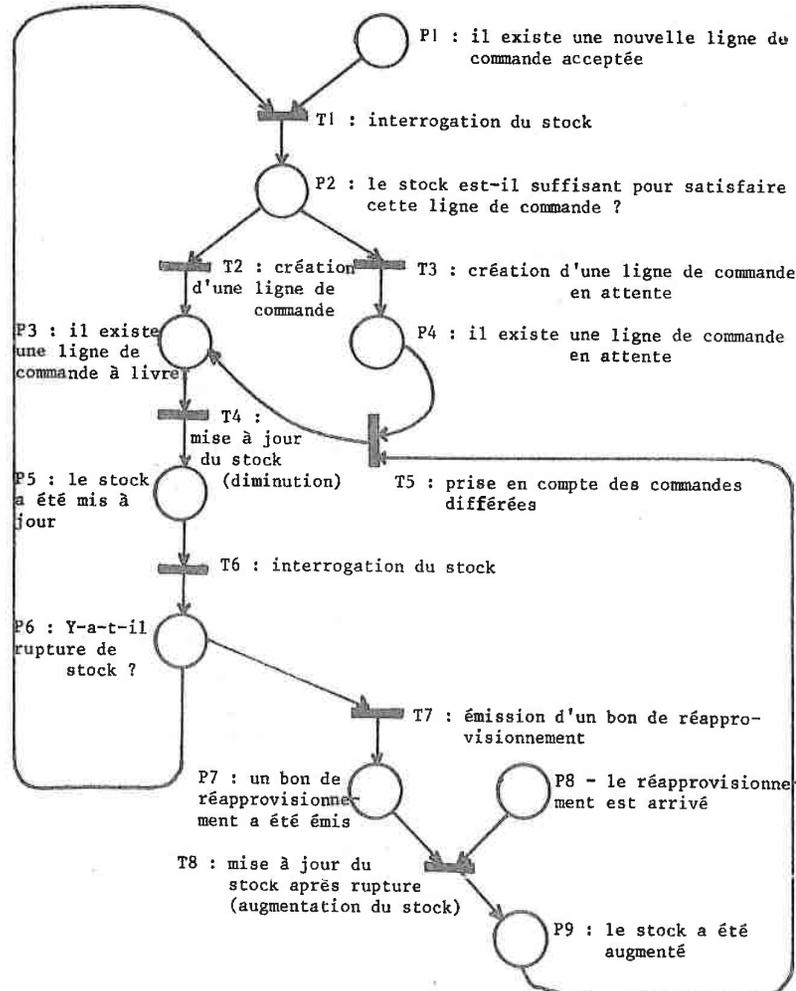


Figure IV-6

\mathcal{P}_1 : (T1,T2,T3) et (P1,P2,P3,P4)

\mathcal{P}_2 : (T4) et (P3,P5)

\mathcal{P}_3 : (T6,T7) et (P5,P6,P7)

\mathcal{P}_4 : (T8) et (P7,P8,P9)

\mathcal{P}_5 : (T5) et (P9,P4)

Communication entre ces processus qui représentent les fonctions, assurant leur synchronisation en traduisant les interactions de ces fonctions.

$\mathcal{P}_1 \rightarrow \mathcal{P}_2$: (P3)

$\mathcal{P}_2 \rightarrow \mathcal{P}_3$: (P5)

$\mathcal{P}_3 \rightarrow \mathcal{P}_4$: (P7)

$\mathcal{P}_4 \rightarrow \mathcal{P}_5$: (P9)

$\mathcal{P}_5 \rightarrow \mathcal{P}_2$: (P3)

$\mathcal{P}_2 \rightarrow \mathcal{P}_1$: (P6)

IV-3.3.- L'approche conceptuelle

Rappelons les résultats de l'analyse auxquels nous étions parvenus :

1) détermination des objets représentés par des C-OBJETS :

- OB1 : "ligne de commande à traiter"
- OB2 : "ligne de commande à livrer"
- OB3 : "ligne de commande différée"
- OB4 : "produit en stock"
- OB5 : "ordre de réapprovisionnement"
- OB6 : "produit réapprovisionné"

2) détermination des événements représentés par les C-EVENEMENTS, à partir de l'ensemble des C-OBJETS puisqu'un C-EVENEMENT est un changement d'état remarquable d'un C-OBJET (c'est-à-dire qu'un événement est un changement d'état remarquable d'un objet) :

- EV1 : "arrivée d'une ligne de commande à traiter" lié à OB1
- EV2 : "arrivée d'une ligne de commande à livrer" lié à OB2
- EV3 : "rupture de stock" lié à OB4
- EV4 : "suppression de rupture de stock" lié à OB4
- EV5 : "arrivée d'un produit réapprovisionné" lié à OB6

3) définition des opérations (représentées par des C-OPERATIONS) déclenchées par des événements (représentés par des C-EVENEMENTS) :

- OP1 : "création de OB2" déclenchée par EV1 si le stock est suffisant (condition C1)

- OP2 : "création de OB3" déclenchée par EV1 si le stock n'est pas suffisant ($\bar{C}1$)
- OP3 : "mise à jour de OB4" déclenchée par EV2
- OP4 : "création de OB5" déclenchée par EV3
- OP5 : "création de OB2" déclenchée par EV4
- OP6 : "mise à jour de OB4" déclenchée par EV5

Cette analyse conceptuelle conduit au sous-schéma dynamique présenté au paragraphe II-2.2.1., que nous rappelons ci-dessous :

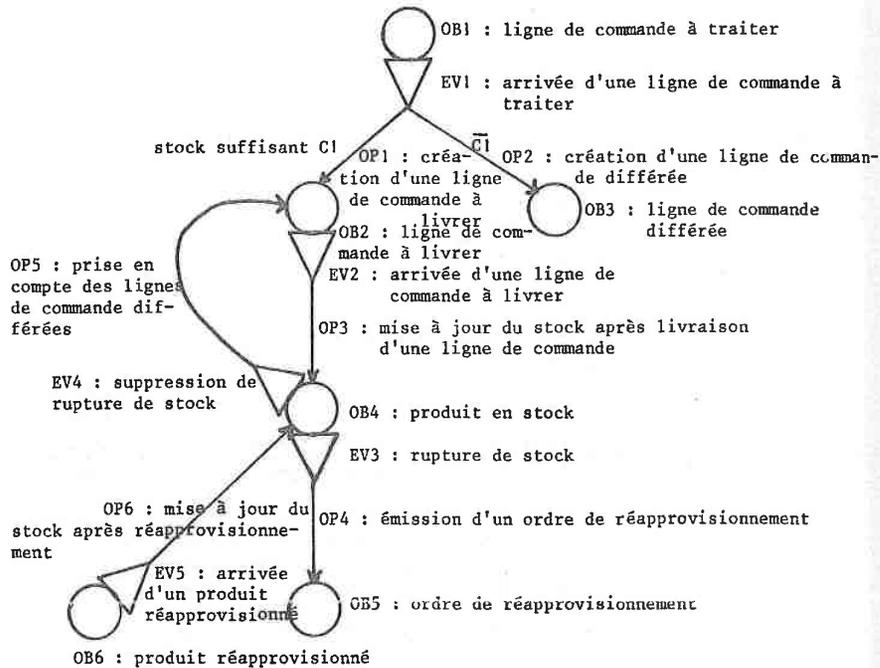


Figure IV-7

IV-3.4.- Commentaires

Cet exemple met en évidence la difficulté à effectuer une analyse d'un problème en termes de fonctions : le découpage du problème en fonctions à assurer par le système semble plus souvent relever de "l'art du programmeur" que d'une approche systématisable de ces problèmes. Cet aspect heuristique du découpage d'un problème en fonctions nous semble expliquer l'apparente difficulté à démarrer une analyse de ce type. De plus, il est difficile d'être sûr que l'analyse du problème que l'on a effectuée soit correcte et que la représentation que l'on en fait ne contienne pas d'erreurs. En effet, si l'on considère le réseau de Petri de la figure IV-6, l'omission d'une place telle que P1, exprimant une condition de prise en compte d'une ligne de commande conduit à une solution erronée puisque tentant de prendre en compte des lignes de commandes acceptées même s'il n'en existe plus, d'où les nombreux contrôles associés à ces approches.

Au contraire, la méthode sous-tendant l'approche conceptuelle (voir paragraphe II-1.3.) constitue un guide de savoir faire pour l'analyse et la description en termes de modèle conceptuel d'un problème : on détermine les objets (à partir des constituants du système réel), puis leurs changements d'état événementiels, et les actions qui en sont la conséquence. Parallèlement, les contrôles formels présentés au chapitre III, ainsi qu'un inventaire systématique des opérations que l'on peut effectuer sur les objets permettent de s'assurer que l'on a pas oublié d'événements ou d'opérations, voire même d'objets.

IV-3.5.- Conclusions

A la lumière des exemples présentés ci-dessus, la démarche d'analyse conceptuelle nous semble plus rigoureuse que les démarches de conception opératoires, pour trois raisons principales : la précision et l'élémentarité des concepts du modèle ainsi que l'aspect structurel de la méthode d'analyse.

Comme nous l'avons montré au paragraphe II-1.3., l'approche conceptuelle est de type structuraliste, c'est-à-dire qu'elle conduit à définir un système par sa structure, et en fait par deux sous-structures :

- le schéma statique ou image de la structure des constituants du système représenté,
- le schéma dynamique ou image de la structure de fonctionnement de ce système.

Notre expérience de la conception des bases de données [8], [9] nous a conduits à proposer de définir en premier lieu le schéma statique, puis d'étudier l'évolution des objets du schéma statique, c'est-à-dire de définir le schéma dynamique.

Ainsi, dans le cas de l'analyse conceptuelle d'un système posant des problèmes de synchronisation, on commence par rechercher de manière systématique quels sont les constituants de ce système, qui seront définis comme des objets représentés par des C-OBJETS (schéma statique). Ensuite, une fois ces objets déterminés, en s'appuyant sur leur définition, on recense quels sont les changements d'état de ces objets qui constituent des événements du système (représentés par des C-EVENEMENTS), c'est-à-dire qui entraînent la création, la modification ou la suppression d'objets de ce système. Ces actions de création, modification ou suppression des objets constituent les opérations (représentées par des C-OPERATIONS) du système (schéma dynamique). Donc, à partir d'une analyse structuraliste du système (c'est-à-dire à partir d'une analyse des constituants de ce système), l'élaboration du schéma conceptuel qui le modélise suit une méthode rigoureuse assurant que le schéma conceptuel traduit correctement la réalité du système.

- D'autre part, les approches opératoires, elles, sont de type fonctionnel, ce qui revient à dire que ces approches abordent les problèmes par l'étude des fonctions déterminées à partir de leurs énoncés. Ainsi, dans le cas de l'analyse opératoire d'un système posant des problèmes de synchronisation, on commence par établir l'inventaire de toutes les fonctions que doit assurer le système. C'est la raison pour laquelle la notion de processus est prépondérante dans ces approches : un processus (ou l'enchaînement de plusieurs processus) constitue l'expression de chacune de ces fonctions (voir paragraphe I-3.2.). La difficulté d'une approche fonctionnelle résulte du fait que l'on n'est jamais certain d'avoir décrit toutes les fonctions que le système doit assurer, et surtout toutes les interactions entre ces fonctions. Cet aspect de la méthode d'analyse relevant plus de "l'art du programmeur" que d'une méthode rigoureuse. Ici, nous retrouvons la difficulté de faire l'inventaire de toutes les situations de synchronisation entre les processus réalisant ces fonctions ainsi que de leurs dépendances réciproques, difficulté que nous avons déjà signalée au paragraphe précédent et que nous avons reliée à l'absence de méthode rigoureuse aidant à l'élaboration d'une modélisation qui puisse être complète.

- Il semblerait que l'on retrouve ici les différences existant entre les méthodes de programmation déductive [74], [76] de type structuraliste, et les méthodes de programmation structurée [32], [95] de type fonctionnel, où la programmation structurée s'appuie sur une méthode rigoureuse basée sur l'analyse des résultats souhaités (analyse structurelle) alors que certaines méthodes de programmation structurée relèvent de "l'art du programmeur" quant à la structuration des programmes

De plus, la précision des concepts, liée à leur définition précise sous forme normalisée constitue à notre avis un facteur de rigueur de l'approche conceptuelle. De même, l'élémentarité de ces concepts, liée à leur définition sous forme normalisée assurant l'élimination de toute redondance au niveau du schéma conceptuel conduit à la minimalité de la description [39], [40], au sens de couverture minimale d'un ensemble de relations [28]. Le schéma conceptuel représente donc le plus petit noyau de C-OBJETS, C-EVENEMENTS, C-OPERATIONS nécessaires à la description d'un système réel. Ainsi, sur l'exemple présenté dans ce paragraphe, le C-OBJET OB4 "produit en stock" possède deux changements d'états événementiels constatés par EV3 (rupture de stock) et EV4 (suppression de rupture de stock) déclenchant respectivement l'émission d'un bon de réapprovisionnement (C-OPERATION OP4) et la prise en compte des commandes différées (OP5). Une telle description en termes opératoires, représentée selon le formalisme des réseaux de Petri nécessite trois transitions (T6, T7, T5) et cinq places (P6, P7, P9, P4, P3). Dans ce sens, l'élémentarité des concepts constitue un facteur de rigueur puisqu'elle conduit à une représentation minimale.

Enfin, de par la méthode rigoureuse d'analyse qui la sous-tend, de par la simplicité et la facilité de description qu'elle permet, et de par les qualités de la représentation à laquelle elle conduit, l'approche conceptuelle facilite la maîtrise de la conception et permet aux concepteurs de contrôler à tout instant leur travail. Ainsi, par exemple, si l'on avait omis le C-EVENEMENT EV4 (fin de rupture de stock) permettant la prise en compte des lignes de commandes différées, dans le schéma conceptuel traduisant le problème simplifié de gestion des stocks, présenté au paragraphe IV-3.3. ci-dessus (figure IV-7), ceci apparaîtrait nettement (on aurait défini un C-OBJET 'ligne de commande différée' qui ne serait jamais pris en compte par une C-OPERATION et qui ne constituerait pas une "sortie" du système modélisé). Par contre, une telle omission serait moins directement perceptible sur le réseau de Petri de la figure IV-6, correspondant à ce même problème.

On peut également remarquer que les contrôles formels sur la qualité d'un schéma conceptuel, présentés au chapitre III, accroissent encore la rigueur de l'approche. En effet, bien que durant toute l'étape de conception la définition précise et élémentaire des concepts permette une plus grande rigueur, cela n'empêche pas des contrôles complémentaires sur la qualité de la modélisation obtenue. Néanmoins ces contrôles complètent les qualités des concepts plutôt que de pallier à leur insuffisance, comme c'est le cas dans certaines approches opératoires reposant sur des concepts parfois trop vagues ou redondants. On peut d'ailleurs remarquer que c'est dans des approches structurales de ce type, liées à l'étude des bases de données

qu'on est allé le plus loin dans les démonstrations sur la non-redondance notamment (processus de normalisation d'un ensemble de relations aboutissant à une collection de relations en troisième forme normale [19], [37], ou notion de couverture minimale d'un ensemble de relations [28], par exemple). En effet, s'il existe des méthodes formelles pour prouver la cohérence d'une solution dans les approches opératoires (par exemple l'analyse de réseaux de Petri par des systèmes d'équations matricielles [72], [90], ou l'analyse au moyen de schéma de programmes parallèles [53] ou au moyen d'un système d'additions de vecteurs [51]), elles ne permettent pas la détection de la redondance des descriptions fournies.

IV-4.- LA SOLUTION CONCEPTUELLE CONSTITUE UNE AIDE POUR LE CHEF DE PROJET

Dans le domaine de l'informatique des organisations, le chef d'un projet informatique doit expliquer la solution qu'il a développée aux utilisateurs du système informatique qui ne sont généralement pas des informaticiens. Ceci constitue une tâche plus difficile qu'il ne paraît au premier abord : on ne peut pas utiliser le vocabulaire "barbare" d'un spécialiste, mais néanmoins il faut expliquer cette solution avec précision. Il nous apparaît que la solution proposée peut constituer une aide non négligeable, notamment par la simplicité et l'évidence sémantique des concepts, puisque la méthode de définition de ces concepts, basée sur l'idée de traduction des phénomènes réels en termes de types permet leur rattachement aux éléments du système réel [40], [86], et qui peuvent donc être exposés en termes clairs.

De plus, elle permet notamment :

- 1) la compréhension plus aisée d'un problème, puisque la description conceptuelle de celui-ci en termes de types constitue, de par les qualités qui lui sont propres, une vue synoptique minimale de tous les aspects de ce problème. Par là même, elle constitue une aide à la compréhension de ce problème (voir par exemple la description du système producteur/consommateur, au paragraphe IV-1.1. sous forme de schéma conceptuel et de réseau de Petri).
- 2) l'obtention plus aisée d'un consensus avec les utilisateurs du futur système automatisé. En effet, quel que soit le domaine auquel se rattache le système que l'on veut créer, qu'il soit technique (SGBD, SGSI [86], voire système d'exploitation) ou bien réel (système d'information, application de gestion), il est bon d'avoir une description sur laquelle discuter avec les spécialistes du domaine et les futurs utilisateurs, et qui ne nécessite pas de leur part l'apprentissage d'un modèle de représentation com-

plexe. A partir de cette description, il doit être aisé d'obtenir un consensus de l'ensemble des personnes concernées par la description du système (concepteurs, utilisateurs, spécialistes du domaine) sur l'adéquation de celui-ci. La représentation conceptuelle, de par ses qualités de simplicité, de minimalité et surtout d'évidence sémantique nous semble particulièrement adaptée à ce propos. Par exemple, il nous paraît plus facile d'argumenter sur l'adéquation de la description conceptuelle du problème simplifié de gestion des stocks (paragraphe IV-3.3., figure IV-7) que sur la description de ce même problème en termes de réseau de Petri (paragraphe IV-3.2., figure IV-6), et l'obtention d'un consensus semble plus aisée dans le cas de l'approche conceptuelle de par l'évidence sémantique de ce type de description.

- 3) Enfin, la remise en cause de la description conceptuelle d'un problème est aisée, et sa modification est facile si le consensus précédemment évoqué n'a pu être obtenu. En effet, une telle description en termes de types d'objets, de types d'événements et de types d'opérations, de par les liens de causalité qui existent entre ces concepts, permet de prévoir toutes les conséquences d'une modification qui l'affecte, et donc de s'assurer que celle-ci n'introduit pas d'incohérence dans cette description.

Il est bien évident que ces caractéristiques de l'approche conceptuelle augmentent encore la fiabilité des solutions auxquelles elle conduit.

Parallèlement, le chef de projet a en tête une vision précise, globale, synthétique et minimale du problème décrit et il peut se référer à cette image en permanence, notamment lorsqu'il s'agit de répondre sur le champ à une proposition de modification, accroissant ainsi la capacité de maintenance et de flexibilité du système proposé.

De plus, comme nous l'avons signalé au chapitre I, les problèmes de synchronisation ont été initialement abordés dans le cadre des systèmes artificiels (les systèmes informatiques). Ils ont donc été étudiés par des spécialistes du domaine (les informaticiens) et pour des spécialistes de ce domaine à qui l'on peut demander l'acquisition d'un certain type de vocabulaire et l'apprentissage de modèles contraignants, tels que les modèles mathématiques. Mais ceci n'est pas forcément bon : il nous semble que l'on maîtrise moins bien un problème et son éventuelle correction avec par exemple un réseau de Petri en tête plutôt qu'avec le schéma conceptuel de ce problème présent à l'esprit, comme on peut s'en rendre compte sur les différents exemples présentés tout au long de ce chapitre. Ainsi, il nous apparaît que ce que notre domaine nous a imposé ne constitue pas une contrainte forcément mauvaise pour le milieu des informaticiens.

IV-5.- LA SPECIFICITE DES OUTILS POUR GERER LA REPRESENTATION CONCEPTUELLE

Comme nous l'avons montré au paragraphe II-1.4., un outil est dépendant du type de représentation qu'il doit gérer, aussi avons-nous développé la notion de machine abstraite pour gérer la représentation conceptuelle d'un système synchronisé. Comme nous l'avons montré dans ce paragraphe, la machine abstraite [63] intègre les mécanismes qui assurent le fonctionnement de tout système décrit en termes de structure conceptuelle. Elle fonctionne selon ses propres règles. Elles seront mises en oeuvre selon le schéma de fonctionnement du système réel à automatiser. Ses primitives sont : reconnaître les événements - déclencher l'exécution des opérations - mémoriser des représentations.

Le fonctionnement de la machine abstraite pour un système particulier peut être schématisé de la façon suivante : la machine reconnaît les événements qui se produisent, détermine les opérations qui doivent être déclenchées (en exploitant la structure conceptuelle correspondant au système), déclenche et contrôle leur exécution, vérifie si les changements d'état provoqués par les opérations sont des événements, et si oui, détermine et déclenche des actions conséquentes.

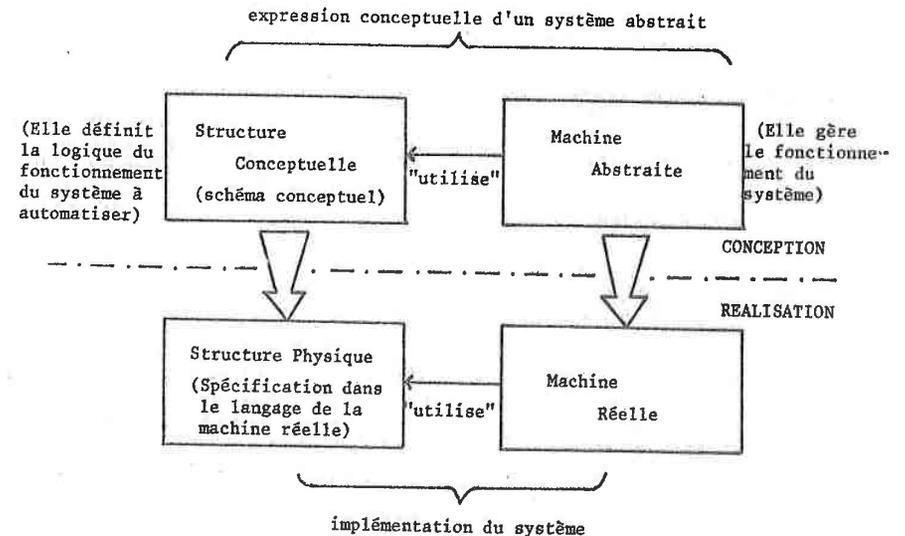
Il apparaît donc, après ce rappel du fonctionnement de la machine abstraite, présenté au paragraphe II-1.4. que le mécanisme de synchronisation exploité par la machine abstraite est la gestion de l'événement qui permet, comme nous l'avons vu au chapitre II, de retrouver par le biais des dépendances chronologiques entre C-EVENEMENTS toutes les relations temporelles entre C-OPERATIONS (nos processus) définissant la synchronisation du système modélisé.

Enfin, l'approche conceptuelle de conception d'un système informatique induit le processus de spécification d'un outil [63], [86] suivant, qui consiste à procéder en deux étapes qui s'enchaînent :

- la première étape, ou étape conceptuelle, conduit :
 - à définir la logique de fonctionnement du système à automatiser sous la forme d'une description par un schéma conceptuel. Ce schéma conceptuel (ou structure conceptuelle) correspond à la spécification formelle (ou conceptuelle) du logiciel.
 - à exprimer cette structure dans le langage de description associé à une machine abstraite définie une fois pour toutes [86]. Cette machine comporte les mécanismes opératoires propres à assurer la gestion du fonctionnement d'un système défini par sa structure conceptuelle de fonctionnement [87].

- la deuxième étape correspond :
 - à l'implémentation de la machine abstraite sur une machine réelle
 - au passage de la spécification conceptuelle (dans le langage de la machine abstraite) à la spécification physique dans le langage de la machine concrète.

Cette démarche peut donc se résumer par le schéma suivant :



De même, cette approche de réalisation d'un système permet une très grande portabilité puisque la description conceptuelle d'un problème et la machine abstraite sont définies indépendamment de toute considération technique (voir paragraphe II-1.4.), seule la description physique et la machine réelle sont dépendants du matériel sur lequel est implémenté ce système. Aussi, lorsque celui-ci doit être implémenté sur un autre type de matériel, il suffit de modifier la description physique et la machine réelle pour que ce système soit utilisable, ce qui assure la portabilité de ce système. Pour les mêmes raisons, si l'on change une partie des caractéristiques du matériel servant à l'implémentation du système, il suffit de modifier la machine réelle associée et éventuellement la description physique du système réel pour en assurer le bon fonctionnement, ce qui en rend la maintenance aisée.

IV-6.- LIMITATIONS DE L'APPROCHE CONCEPTUELLE

L'approche conceptuelle comporte, à notre avis, trois principales limitations :

- De par l'aspect structuraliste de cette approche de modélisation, et de par l'aspect causal du modèle sur lequel elle s'appuie, l'approche conceptuelle est limitée à la prise en compte des problèmes déterministes.

- La sémantique associée aux concepts du modèle est parfois gênante, puisqu'orientée vers la description des systèmes réels (perçus par leurs constituants), alors que certains modèles utilisés dans les approches opératoires sont définis mathématiquement (tels les réseaux de Petri) et ne possède donc aucune sémantique associée a priori. Ceci conduit peut être à une plus grande lourdeur de description des systèmes réels (voir paragraphes IV-1. à IV-4.) mais permet un champ d'application beaucoup plus étendu (les réseaux de Petri sont par exemple utilisés pour la définition et l'étude de langage formels [23], [24], [45], alors que l'approche conceptuelle s'avère très peu adaptée à ce genre d'étude, de par la sémantique associée aux concepts qui en sont le fondement).

- Enfin, le raisonnement en termes de types de constituants (C-OBJETS), de changements d'état (C-EVENEMENT), de transformations (C-OPERATIONS) élémentaires qui caractérise l'approche conceptuelle conduit à des problèmes d'efficacité au niveau d'une éventuelle implémentation (en termes d'occurrences) de la description conceptuelle du système réel à laquelle il conduit. En effet, un tel raisonnement conduit à un grand nombre de C-EVENEMENTS, C-OPERATIONS et C-OBJETS dont il faut implémenter les occurrences, et des stratégies d'implémentation sont à l'étude [63], [85].

IV-7.- CONCLUSION

L'approche conceptuelle, bien qu'équivalente aux approches opératoires dans sa finalité, est néanmoins différente par le raisonnement auquel elle conduit le concepteur, par la solution aux problèmes de synchronisation qu'elle lui permet de définir et par les outils que l'on est tenté de lui associer.

L'analyse que nous venons d'en faire a mis en lumière ses avantages et ses limitations : elle paraît mieux adaptée au processus de conception-réalisation d'un système automatisant un problème réel, la sémantique associée à ses concepts de base autorisant une description permettant une bonne compréhension du problème à traiter, et qui, associée à une machine abstraite, permet une implémentation aisée (paragraphe IV-2.6.). D'un autre côté, cette sémantique associée aux concepts de base du modèle conduit à une généralité moins grande du modèle, et à une polyvalence

moins importante de celui-ci car la sémantique qui lui est associée est parfois gênante pour l'analyse de certains types de problèmes (étude de langages, ou de systèmes automatiques ou logiques par exemple).

Il reste néanmoins à constater les avantages de cette approche dans les domaines variés de l'informatique où l'on est conduit à gérer des représentations de systèmes dynamiques.

Il reste de plus de nombreuses questions en suspens, telles que

- une étude formelle approfondie des propriétés du modèle liées à la minimalité
- une étude formelle des règles de passage d'une solution issue d'une analyse structuraliste à une solution issue d'une analyse fonctionnelle (et réciproquement).

CONCLUSION

En nous basant sur le modèle conceptuel proposé par l'équipe Remora, nous avons présenté une nouvelle approche des problèmes de synchronisation, qui, bien qu'équivalente dans sa finalité aux approches habituelles, en diffère néanmoins par la manière d'aborder les problèmes et par la façon de gérer la synchronisation.

En effet, il nous semble que l'analyse de la synchronisation s'est arrêtée à la compréhension de la synchronisation comme l'interaction des processus dans le temps : on se place à l'instant t et on cherche toutes les conditions de dépendance entre les processus susceptibles d'être déclenchés à cet instant. Or, il nous est apparu que ces dépendances temporelles entre les processus ne sont qu'une conséquence d'un certain état de fait.

Ainsi, l'idée fondamentale de notre réflexion sur la synchronisation est de remplacer le raisonnement en terme d'interrelations temporelles entre processus par un raisonnement de type causal, consistant à déterminer quelles sont les causes et les conséquences de l'exécution des processus, la conséquence de l'exécution de l'un pouvant être la cause de l'exécution de l'autre. Ainsi, les relations temporelles entre les processus, définissant habituellement la synchronisation apparaissent comme des conséquences et non pas comme les causes premières de celle-ci. Une des caractéristiques fondamentales de notre approche est que nous avons introduit une nouvelle manière de poser les problèmes de synchronisation en termes de causalité, ce qui constitue l'originalité de l'approche conceptuelle des problèmes de synchronisation. Nous avons montré, au chapitre II, que le modèle causal élaboré dans le cadre de la représentation des SI permet de représenter les problèmes de synchronisation au moyen des liens de causalité entre les concepts de ce modèle permettant la définition des interrelations entre les processus, avec des qualités que nous avons analysé aux chapitres III et IV.

De plus, notre réflexion sur la synchronisation et sa modélisation nous a permis de montrer que le modèle conceptuel constitue un modèle de représentation d'un système réel en fonctionnement puisque les règles le définissant sont contenues dans le schéma conceptuel qui le représente [86], [87]. Comme, pour nous, un outil est très fortement lié au type de modélisation qu'il exploite, nous avons défini un outil spécifique pour la gestion du fonctionnement (donc de la synchronisation) d'un système décrit en termes conceptuels : la machine abstraite que nous avons présentée au chapitre II et dont l'intérêt, notamment le fait qu'elle débouche sur un mécanisme de gestion très simple de la synchronisation (à savoir la gestion des événements) a été évaluée aux chapitres III et IV.

Nous pensons enfin que l'approche conceptuelle telle qu'elle a été présentée dans ce travail permet la conception des systèmes répartis, au sens de [68]. Notre prochain objectif de travail sera de montrer que le modèle conceptuel permet de décrire un système réparti et de montrer qu'associé à la machine abstraite gérant la synchronisation du système au niveau conceptuel il permet de rendre compte de son fonctionnement, ouvrant ainsi la voie au processus technique d'implémentation.

BIBLIOGRAPHIE

- [1] AGERWALA T.
"A complete model for representing the coordination of asynchronous processes"
Computer research report 32, John Hopkins University, Baltimore (Md), 1974.
- [2] AGERWALA T., FLYNN M.
"Comments on capabilities, limitations and correctness of Petri Nets"
Proceedings of the 1^{rst} annual Symposium on Computer Architecture
Lipovski G.J. and Szygenda S.A. editors, A.C.M., New-York, 1973.
- [3] Groupe ALGOL de l'AFCET
"Définition du langage algorithmique ALGOL 68"
Editions Hermann, Paris, 1972.
- [4] ANSI/X3/SPARC
"Report on Data Base Management Systems"
Interim report, 1975.
- [5] BAER J.L., ELLIS C.S.
"Model, design and evaluation of a compiler for parallel processing environment"
I.E.E.E. transactions on software engineering, Vol. SE-3, Nb. 6, 1977.
- [6] BANON D.
"Projet Remora : un outil pour la gestion des Systèmes d'Information"
Thèse de 3ème cycle, Nancy, 1979.
- [7] BENCI G., BODART F., BOGAERT H., CABANES A.
"Concepts for the design of a conceptual schema"
in Modelling in Data Base Management systems, proceedings of IFIP Working Conference TC-2, 1976.
- [8] BENCI G., ROLLAND C.
"Cours de conception des Bases de Données"
Séminaire au Centro de Calculo de Sabadel, Madrid, publié dans la revue Techniques de l'Ingénieur, 1976.
- [9] BENCI G., ROLLAND C.
"La conception des Bases de Données de demain"
S.C.M. Publication, 1979.
- [10] BERNSTEIN P.A.
"A sophisticated introduction to Data Base design"
Proceedings of the 4th V.L.D.B. Conference, Berlin, 1978.

- [11] BEZIVIN J. et al.
"SIMONE : manuel de programmation"
IRIA-SFER publications, Rocquencourt, 1976.
- [12] BRACCHI G., FURTADO A., PELAGATTI
"Constraints specification in evolutionnary Data Base design"
Proceedings of IFIP Working Group TC-8, Oxford, 1979.
- [13] BRINCH-HANSEN P.
"Operating Systems principles"
Prentice Hall series in automatic computation, 1973.
- [14] BRINCH-HANSEN P.
"The programming language Concurrent PASCAL"
I.E.E.E. transactions on software engineering, Vol. SE-1, Nb. 2,
1977.
- [15] BRINCH-HANSEN P.
"Distributed Processes : a concurrent programming concept"
C.A.C.M. Vol. 21, Nb. 11, 1978.
- [16] CAMPBELL R.H., HABERMANN A.N.
"The specification of process synchronization by path expressions"
Lecture notes in Computer Science, Vol. 16, Springer Verlag, Berlin,
1974.
- [17] CHEN P.P.S.
"The entity-relationship model towards an unified view of data"
A.C.M. - T.O.D.B.S. Vol. 1, Nb. 1, 1976.
- [18] CODD E.F.
"A relational model of data for large shared Data Banks"
C.A.C.M., Vol. 13, Nb. 6, 1970.
- [19] CODD E.F.
"Normalized data structures : a brief tutorial"
A.C.M. SIGFIDET Workshop on data description, access and control,
San Diego, California, 1971.
- [20] CODD E.F.
"Recent investigations in relational Data Base systems"
Proceedings of IFIP Congress 74, North-Holland Publications,
Amsterdam, 1974.
- [21] CONWAY M.E.
"Design of a separable Transition-Diagram Compiler"
C.A.C.M., Vol. 6, pp 396-408, 1963.

- [22] COURVOISIER M.
"Etude des systèmes logiques de commande asynchrones à évolutions
simultanées"
Thèse de doctorat ès Sciences, Toulouse, 1974.
- [23] CRESPI-REGHIZZI S., MANDRIOLI D.
"Petri Nets and commutative grammars"
Internal report n° 74-5, laboratorio di calcolatori, Istituto di
elettrotecnica ed elettronica di Politecnico di Milano, Italie, 197
- [24] CRESPI-REGHIZZI S., MANDRIOLI D.
"Petri Nets and Szilard languages"
Information and Control, Vol. 33, Nb. 2, 1977.
- [25] CROCUS
"Systemes d'exploitation des ordinateurs"
Editions Dunod, Paris, 1975.
- [26] DAHL O.J., NYGAARD K.
"SIMULA, an Algol based simulation language"
C.A.C.M., Vol. 9, Nb. 2, 1966.
- [27] DAHL O.J. et al.
"The SIMULA 67 common base language"
Publication S 22, Norwegian computing center, Oslo, 1970.
- [28] DELOBEL C.
"Contribution théorique à la conception et à la définition d'un
Système d'Information appliqué à la gestion"
Thèse de doctorat ès Sciences, Grenoble, 1973.
- [29] DENNIS J.B.
"Modular asynchronous control structures for a high performance
processor"
Record of the project MAC conference on concurrent systems and
parallel computation A.C.M., New-York, 1970.
- [30] DENNIS J.B. (editor)
"Record of the project MAC conference on concurrent systems and
parallel computation"
A.C.M., New-York, 1970.
- [31] DENNIS J.B.
"Concurrency in software systems"
in Advanced course in software engineering, Bauer F.L. editor,
Springer-Verlag, Berlin, 1973.

- [32] DERNIAME J.C.
"CIVA : un système de programmation modulaire"
Thèse de doctorat ès Sciences, Nancy, 1974.
- [33] DIJKSTRA E.W.
"Cooperating sequential processes"
in Programming languages, Genuys F. editor, Academic Press, 1967.
- [34] DIJKSTRA E.W.
"Structure of T.H.E. multiprogramming system"
C.A.C.M., Vol. 11, Nb. 5, 1968.
- [35] DIJKSTRA E.W.
"Hierarchical ordering of sequential processes"
Acta Informatica, Vol. 1, Nb. 2, 1971.
- [36] DIJKSTRA E.W.
"Guarded commands, non determinacy and formal derivation of programs"
C.A.C.M., Vol. 18, Nb. 8, 1975.
- [37] FAGIN R.
"Multivaluated dependencies and a new normal form for relational Data Bases"
A.C.M. - T.O.D.B.S., Vol. 2, Nb. 3, 1977.
- [38] FLORY A.
"Un modèle et une méthode pour la conception d'une Base de Données"
Thèse de doctorat ès Sciences, Lyon, 1977.
- [39] FOUCAUT O.
"Propositions pour la conception des systèmes d'information"
Thèse de doctorat ès Sciences, Nancy, à paraître.
- [40] FOUCAUT O., ROLLAND C.
"Concepts for the design of an Information System Conceptual Schema and its utilization in the Remora project"
Proceedings of the 4th V.L.D.B. Conference, Berlin, 1978.
- [41] GLUSHKOV V.M., LETICHEVSKII A.A.
"Theory of algorithms and discrete processors"
Advances in information systems science, Vol. 1, ch. 1
- [42] HABERMANN A.N.
"Path expressions"
Carnegie Mellon University, 1975.

- [43] HACK M.
"Decision problems for Petri Nets and Vector Addition Systems"
Computation structures group, Memo. 95, Project MAC, M.I.T., Cambridge (Mass), 1974.
- [44] HACK M.
"Petri Net languages"
Computation structures group, Memo. 124, project MAC, M.I.T., Cambridge (Mass), 1975.
- [45] HACK M., PETERSON J.L.
"Petri Nets and languages"
Proceedings of M.I.T. Conférence on Petri Nets and related methods, M.I.T., Cambridge (Mass), 1975.
- [46] HOARE C.A.R.
"Towards a theory of parallel programming"
Operating Systems Techniques, Hoare editor, 1972.
- [47] HOARE C.A.R.
"Monitors, an Operating Systems structuring concept"
C.A.C.M., Vol. 17, Nb. 10, 1974.
- [48] HOARE C.A.R.
"Communicating Sequential Processes"
C.A.C.M., Vol. 21, Nb. 8, 1978.
- [49] HOLT A.W., COMMONER F.
"Events and Conditions"
in Record of the project MAC Conference on concurrent systems and parallel computation, A.C.M., New-York, 1970.
- [50] HOLT A.W., SAINT H., SHAPIRO R.M., WARSHALL S.
"Final report of the Information System theory project"
Technical report RADC-TR-68-305, Rome air development center, Griffiss A.F.B., New-York, 1968.
- [51] KASAMI T., TOKURA N., PETERSON W.
"Vector Addition Systems and synchronization problems of concurrent processes"
Draft manuscript, 1974.
- [52] KAUBISCH W.H., PERROT R.H., HOARE C.A.R.
"Quasiparallel programming"
in Software practice and experience, Vol. 16, Wiley and Sons, 1976.

- [53] KELLER R.M.
"Parallel Program Schemata and maximal parallelism"
"I/. Fundamental results"
Journal of A.C.M., Vol. 20, Nb. 3, 1973
"II/. Construction of closures"
Journal of A.C.M., Vol. 20, Nb. 4, 1973.
- [54] KELLER R.M.
"Generalized Petri Nets as models for system verification"
Technical report 202, Department of Electrical Engineering, Princeton University, 1975.
- [55] KENT W.
"Describing information (not data reality)"
Technical report TRO 3012.
- [56] KESSELS J.L.W.
"An alternative to event queues for synchronization in monitors"
C.A.C.M., Vol. 20, Nb. 7, 1977.
- [57] KNUTH D.E.
"The Art of computer programming"
Vol. 1, Addison Wesley editors, 1968.
- [58] KOSARAJU S.R.
"Limitations of Dykstra's semaphore primitives and Petri Nets"
Operating Systems review, Vol. 7, Nb. 4, 1973.
- [59] KRIEGUER M.
"Projet Remora : modèle de représentation et méthode de construction des composantes statiques d'un S.I."
Thèse de 3ème cycle, Nancy, 1978.
- [60] KUNTZMANN J.
"Algèbre de Boole"
2ème édition, DUNOD éditeur, Paris, 1968.
- [61] LAUTENBACH K.
"Liveness in Petri Nets"
G.M.D. internal report, ISF-75-02-1, Bonn, 1975.
- [62] LAUTENBACH K., SCHMID H.A.
"Use of Petri Nets for proving correctness of concurrent processes systems"
Proceedings of IFIP Congress 74, North Holland Publications, Amsterdam, 1974.

- [63] LEIFERT S.
"Langage et outils pour la définition et la gestion de Systèmes d'Informations"
Thèse de 3ème cycle, Nancy, à paraître.
- [64] LEONARD M., LUONG B.T.
"Approche des Systèmes d'Information intégrant les données et les traitements"
Proceedings of INFORSID Working Conference on "La dynamique dans les systèmes d'information, Cergy-Pontoise, 1978.
- [65] LINDGREEN P.
"Basic operations on information as a basis for Data Base design"
Proceedings of IFIP Working Group TC-2, North Holland Publication, Amsterdam, 1974.
- [66] LISKOV B., ZILLES S.
"Programming with Abstract Data Types"
Proceedings of Symposium on very high level languages, SIGPLAN notices, Vol. 9, Nb. 4, 1974.
- [67] LISKOV B. et al.
"Abstraction mechanisms in CLU"
C.A.C.M., Vol. 20, Nb. 8, 1977.
- [68] LITWIN W.
"Distributed Data Bases : a way of thinking about"
Proceedings of IRIA Semmar on distributed data sharing systems, Aix-en-Provence, 1979.
- [69] MAC CLUSKEY E.J.
"Algebraic minimization and the design of two terminal contact networks"
M.I.T. Thesis, M.I.T., Cambridge (Mass), 1956.
- [70] MOALLA M., PULOU J., SIFAKIS J.
"Réseaux de Petri synchronisés"
Rapport de recherche ENSIMAG n° 80, Grenoble, 1977.
- [71] MOALLA M., SAUCIER G., SIFAKIS J., ZACHARIADES M.
"A design tool for the multilevel description and simulation of systems of interconnected modules"
13th design automation conference ; Palo Alto, 1976.
- [72] MURATA T., CHURCH R.W.
"Analysis of marked graphs and Petri Nets by matrix equations"
Research report MDC 1-1-8, department of information engineering, University of Illinois, 1975.

- [73] NOE J.D.
"A Petri Net model of the CDC 6600"
Proceedings of the A.C.M. SIGOPS Workshop on system performance
evaluation, A.C.M., New-York, 1971.
- [74] PAIR C.
"Reflexions sur la programmation"
I.N.P.L., 1975.
- [75] PAIR C., DERNIAME J.C.
"Etude des problèmes de cheminement dans les graphes finis"
Dunod éditeur, Paris, 1971.
- [76] PAIR C., MAROLDT J.
"Introduction à une méthode de programmation déductive"
I.N.P.L., 1975.
- [77] PETERSON J.L.
"Petri Nets"
Computing Surveys, Vol. 9, Nb. 3, 1977.
- [78] PETERSON J.L., BREDT T.H.
"A comparison of models of parallel computation"
Proceedings of IFIP Congress 74, North Holland Publication,
Amsterdam, 1974.
- [79] PETRI C.A.
"Kommunikation mit Automaten"
Schriften des Reinisch-Westfalischen Institutes für Instrumentelle
Mathematik an den Universität Bonn, Heft 2, Bonn, 1962.
(traduction : GREENE C.F., supplément 1 to technical report RADC-
TR-65-337, Vol. 1, Rome air development center, Griffiss A.F.B.,
New-York, 1965).
- [80] PETRI C.A.
"Concepts of Net theory"
Proceedings of Symposium and summer school on mathematical founda-
tions of Computer Science, High Tatras, Sept 3-8 1973, Mathematics
Institute of Slovak Academy of Science, 1973.
- [81] PULOU J.
"Un outil pour la spécification de la synchronisation dans les lan-
gages de haut niveau"
R.A.I.R.O. informatique, Vol. 12, Nb. 4, 1978.
- [82] QUINE W.V.O.
"A way to symplify truth functions"
American mathematical monthly, Vol. 22, pp. 521-531, 1955.

- [83] RAMCHANDANI C.
"Analysis of asynchronous concurrent systems by Petri Nets"
Project MAC, technical report MAC-TR-120, M.I.T., Cambridge (Mass)
1974.
- [84] RAYNAL M.
"Une expression de la synchronisation pour les types abstraits"
R.A.I.R.O. informatique, Vol. 12, Nb. 4, 1978.
- [85] ROLLAND C., FOUCAUT O., RICHARD C., THIERY O.
"Information System design and Computer Aided design"
Proceedings of EURO-IFIP Congress, London, 1979.
- [86] ROLLAND C., LEIFERT S., RICHARD C.
"Tools for Information System dynamics management"
Proceedings of the 5th V.L.D.B. Conference, Rio de Janeiro, 1979.
- [87] ROLLAND C., RICHARD C.
"Concepts for a Distributed Computing System design"
Proceedings of IRIA Seminar on distributed data sharing systems,
Aix-en-Provence, 1979.
- [88] ROUCAIROL G.P.
"Mots de synchronisation"
R.A.I.R.O., informatique, Vol. 12, Nb. 4, 1978.
- [89] SHAPIRO R.M., SAINT H.
"A new approach to optimization of sequencing decisions"
Annual review of automatic programming, Vol. 6, Nb. 5, 1970.
- [90] SIFAKIS J.
"Structural properties of Petri Nets"
Rapport de recherche ENSIMAG n° 102, Grenoble, 1977.
- [91] THIERY O.
"L'aide à la conception dans le projet Remora"
Thèse de 3^{ème} cycle, Nancy, 1976.
- [92] VALETTE R., COURVOISIER M.
"Recherche d'un modèle adapté aux systèmes de commande de processus
à évolutions parallèles"
R.A.I.R.O. automatique/Systems analysis and control, Vol. 11, Nb. 1,
1977.
- [93] VAUCHER J.
"A Wait Until algorithm for general purpose simulation language"
Proceedings of Winter simulation conference, San Francisco, 1973.

- [94] VEILLON F., CAGNAT J.F.
"Cours de programmation en langage PL/1"
Vol. 3, Collection U, Armand Colin éditeur, Paris, 1972.
- [95] WIRTH N.
"Systematic programming : an introduction"
Prentice Hall series in automatic computation, 1973.
- [96] WIRTH N.
"MODULA : a programming language for modular multiprogramming"
Software practice and experience, Vol. 17, Wiley and Sons editors,
1977.
- [97] ZACHARIADES M.
"MAS : réalisation d'un langage d'aide à la description et à la conception des systèmes logiques"
Thèse de 3ème cycle, Grenoble, 1977.



NOM DE L'ETUDIANT : RICHARD *Christian*

NATURE DE LA THESE : 3ème cycle : INFORMATIQUE

VU, APPROUVE

et PERMIS D'IMPRIMER

NANCY LE 01 OCT 1979 8175

LE PRESIDENT DE L'UNIVERSITE DE NANCY I


M. BOULANGE