

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Interprétation graphique des mesures de performance d'un système

Lefèvre, J.; Sacré, B.

Award date:
1987

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur.

Institut d'informatique

FLORENCE 21/11
1987

**Interprétation graphique des
mesures de performance
d'un système**

J. Lefèvre

B. Sacré

Mémoire présenté en vue de
l'obtention du titre de
licencié et maître en informatique

Promoteur : F. Bodart

Année académique 1986-1987

Remerciements

A Monsieur F. Bodart, notre promoteur de mémoire, nous adressons nos remerciements les plus chaleureux pour l'aide qu'il nous a apportée et pour les encouragements sans cesse renouvelés qu'il nous a prodigués tout au long de l'élaboration de ce mémoire.

Notre vive gratitude également à Messieurs J.M. Leheureux, J.P. Leclercq, M. Clantin pour leurs judicieux et précieux conseils.

Toute notre reconnaissance à Monsieur Sitruk, Directeur du Centre Informatique de la R.A.T.P. de Paris, pour la gentillesse avec laquelle il nous a accueillis et guidés.

Nous remercions aussi Monsieur F. Letellier du Service Organisation de la R.A.T.P. avec qui nous avons eu le plaisir de collaborer.

Enfin, à tous ceux qui, par leur aide aussi minime soit-elle, nous ont permis de mener à bien notre tâche et de concrétiser nos recherches par la rédaction de ce mémoire, du fond du coeur, nous leur disons : "Un très grand merci".

Table des matières

PREAMBULE

PARTIE I - OUTILS D'INTERPRETATION DES MESURES RELATIVES AUX OBJETS DSL

INTRODUCTION	10
Chapitre I. OUTILS D'INTERPRETATION	11
I.1. Mesures globales	12
I.1.1. Objectifs	12
I.1.2. Description	13
I.2. Mesures chronologiques et périodiques	22
I.2.1. Objectifs	22
I.2.2. Description	24
I.3. Calendriers	33
I.3.1. Objectif	33
I.3.2. Description	33
Chapitre II. DESCRIPTION DES BASES DE DONNEES	37
II.1. Spécifications	37
II.1.1. Définition des entités	37
II.1.2. Définition des associations	40
II.1.3. Schéma de la base de données	43
II.2. Statistiques globales	44
II.2.1. Définition des entités.....	44
II.2.2. Définition des associations	44
II.2.3. Schéma de la base de données	48
II.3. Statistiques périodiques	49
II.3.1. Définition des entités	49
II.3.2. Définition des associations	50
II.3.3. Schéma de la base de données	52

II.4.	Statistiques chronologiques	53
II.4.1.	Définition des entités	53
II.4.2.	Définition des associations	54
II.4.3.	Schéma de la base de données	58
PARTIE II - OUTILS DE REPRESENTATION ET D'EVALUATION DU COMPORTEMENT DU SYSTEME		
INTRODUCTION		60
Chapitre III.	CONCEPT DE TRANSACTION	61
III.1.	Introduction	62
III.1.1.	Problème à résoudre	62
III.1.2.	Idée intuitive	65
III.2.	Définition et mesures du comportement	68
III.2.1.	Préalable	68
III.2.2.	Définition	73
III.2.3.	Mesures du comportement	74
III.3.	Algorithme de construction d'une transaction	76
III.3.1.	Préalable	76
III.3.2.	Spécification	78
III.3.3.	Algorithme	78
Chapitre IV	OUTILS DE REPRESENTATION DU COMPORTEMENT DU SYSTEME	88
IV.1.	Outils de représentation du comportement du système	89
IV.1.1.	Objectifs	89
IV.1.2.	Graphe des événements	90
IV.1.3.	Graphe des objets	95
IV.1.4.	Animation	100
IV.2.	Comportement d'un point de synchronisation	113
IV.2.1.	Objectifs	113
IV.2.2.	Description	113
Chapitre V	OUTILS ET METHODE D'EVALUATION DU COMPORTEMENT DU SYSTEME	118
V.1.	Outil et méthode d'évaluation	119
V.1.1.	Objectifs	119
V.1.2.	Description	119
V.1.3.	Méthode d'interprétation	125

V.2.	Comportement d'un type de transaction	127
V.2.1.	Objectif	127
V.2.2.	Description	127
Chapitre VI	DESCRIPTION DES BASES DE DONNEES	132
VI.1.	Comportement du système	133
VI.1.1.	Définition des entités	133
VI.1.2.	Définition des associations	134
VI.1.3.	Schéma de la base de données	136
VI.2.	Comportement d'un point de synchronisation	137
VI.2.1.	Définition des entités	137
VI.2.2.	Définition des associations	139
VI.2.3.	Schéma de la base de données	142
VI.3.	Comportement du type de transaction	143
VI.3.1.	Définition des entités	143
VI.3.2.	Définition des associations	144
VI.3.3.	Schéma de la base de données	146
BIBLIOGRAPHIE		147

PARTIE III - PRODUCTION AUTOMATIQUE DU SCHEMA DE LA DYNAMIQUE

INTRODUCTION	150	
Chapitre VII.	METHODE DE PRODUCTION AUTOMATIQUE DE GRAPHERS HIERARCHISES	154
VII.1.	Spécification du problème	154
VII.1.1.	Définitions	154
VII.1.1.1.	Graphes orientés	154
VII.1.1.2.	Graphes hiérarchisés	155
VII.1.2.	Spécification du problème	157
VII.2.	Proposition de solution	159
VII.2.1.	Méthode des degrés relatifs	159
VII.2.2.	Notre solution	161

VII.3.	Méthode de production automatique de graphes hiérarchisés	165
VII.3.1.	Construction d'une hiérarchie des rangs	165
VII.3.1.1.	Spécification	165
VII.3.1.2.	Réalisation	166
VII.3.2.	Construction d'une hiérarchie serrée	167
VII.3.2.1.	Spécification	167
VII.3.2.2.	Réalisation	167
VII.3.3.	Construction d'une hiérarchie propre	175
VII.3.3.1.	Spécification	175
VII.3.3.2.	Réalisation	175
VII.3.4.	Minimisation du nombre de croisements	177
VII.3.4.1.	Spécification	177
VII.3.4.2.	Existant	177
VII.3.4.3.	Réalisation	179
VII.3.5.	Calcul des coordonnées	187
VII.3.5.1.	Spécification	187
VII.3.5.2.	Réalisation	187
Chapitre VIII.	PRODUCTION AUTOMATIQUE DU SCHEMA DE LA DYNAMIQUE	
VIII.1.	Spécification du problème	195
VIII.1.1.	Modèle de la dynamique	195
VIII.1.2.	Schéma de la dynamique	197
VIII.2.	Production automatique du schéma de la dynamique	203
VIII.2.1.	Méthode de production automatique du schéma de la dynamique	203
VIII.2.2.	Etapas spécifiques au schéma de la dynamique	217
VIII.2.2.1.	Construction d'un graphe orienté	217
VIII.2.2.2.	Suppression des circuits	217
VIII.2.2.3.	Dessin du schéma de la dynamique	222
BIBLIOGRAPHIE		224
CONCLUSION.....		226
ANNEXES		

PREAMBULE

1. DSL-SIM : Un outil d'évaluation du comportement d'un système d'information (1)

Le système d'information d'une organisation peut être défini comme "une construction formée d'ensembles :

- d'informations, représentations -partielles- de faits qui intéressent l'organisation;
- de traitements, procédés d'acquisition, de mémorisation, de recherche, de communication et de transformation des informations et
- de ressources -humaines, techniques et organisationnelles- qui en assurent le fonctionnement."

Le logiciel IDA (Interactive Design Approach) d'aide à la conception d'un système d'information propose "un ensemble d'outils automatisés pour aider efficacement l'analyste à vérifier que le schéma conceptuel qu'il construit est communicable, complet, cohérent, réalisable et conforme aux besoins".

Un de ces outils, DSL-SIM, a pour objectif principal "d'évaluer si une solution conceptuelle est réalisable par rapport aux ressources dont dispose l'organisation : effectifs en personnel, définition des postes et des horaires de travail, nombre de terminaux à mettre en oeuvre, organisation des circuits administratifs et des canaux de communication des informations, etc ...".

Les spécifications de la solution fonctionnelle à évaluer comprend la description du comportement fonctionnel ainsi que la prise en compte des moyens requis. Ces spécifications seront rédigées au moyen du langage DSL (Dynamic Spécification Language) et intégrées dans la base de données par les programmes de mise à jour faisant partie de l'analyseur DSA (Dynamic Spécification Analyser).

L'architecture fonctionnelle de DSL-SIM comprend les trois principaux modules fonctionnels qui sont représentés à la figure 1 :

La génération d'un programme de simulation

L'objectif de ce module est de "dérivé automatiquement un programme de simulation au départ des spécifications enregistrées dans la base de données de spécifications".

Les seuls paramètres à préciser lorsqu'on invoque cet outil servent à déterminer la ou les parties du système d'information que l'on souhaite évaluer et donc simuler.

Le rapport produit par ce programme reprend les éléments de la spécification les plus pertinents pour la simulation (requêtes aux différentes ressources, ...), le domaine de valeur de paramètres,...

(1) Cette présentation de DSL-SIM a été extraite en majeure partie de la thèse de doctorat [Pigneur, 84], nous nous sommes également fortement inspirés de [Bodart, 87] ainsi que du manuel d'utilisation du logiciel [DSL-SIM, 86].

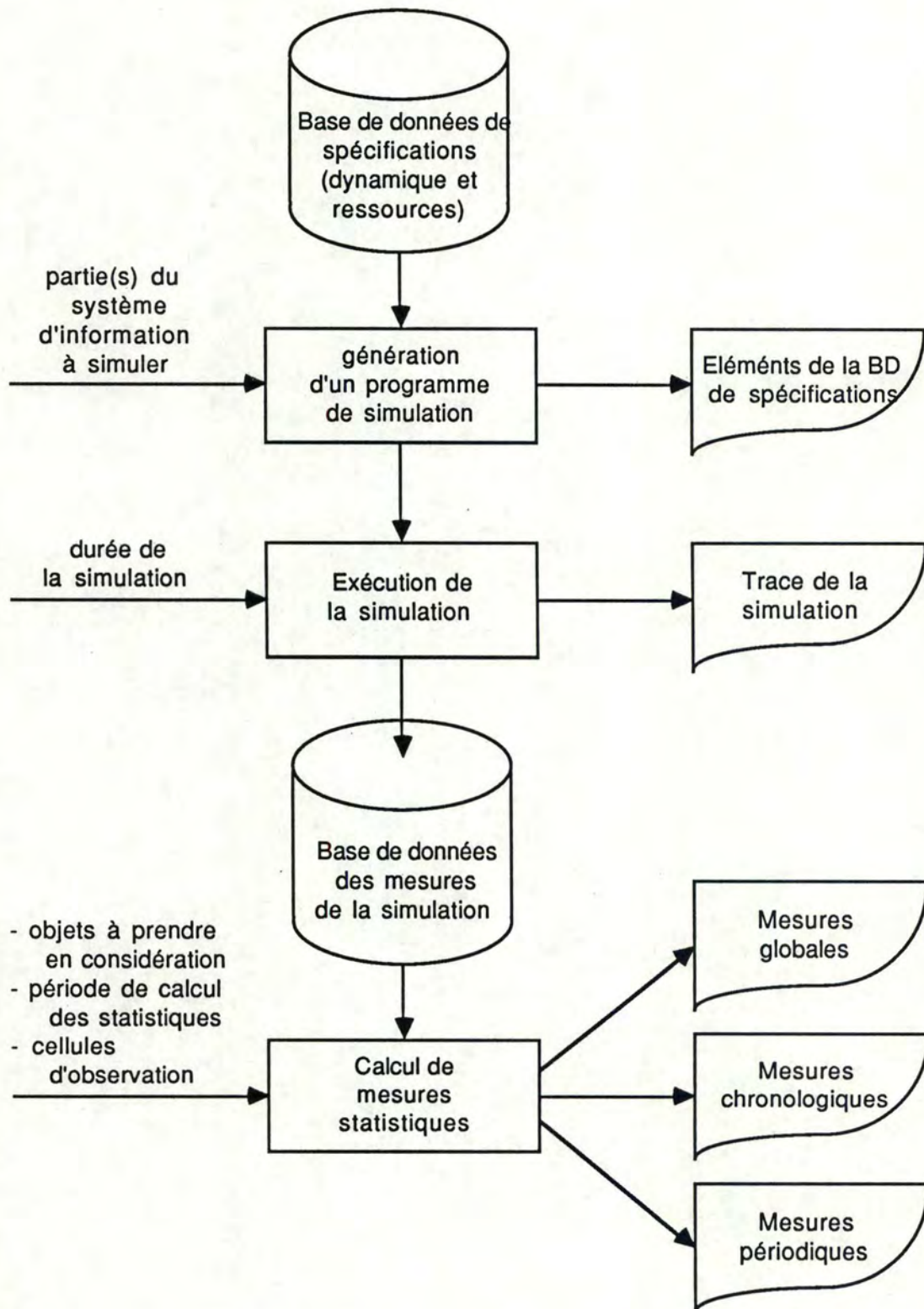


Figure 1. Architecture fonctionnelle de DSL-SIM

Exécution de la simulation

Ce module a pour objectif de "simuler le comportement spécifié sur une période de temps donnée et de collecter des mesures sur le fonctionnement simulé".

La technique adoptée est une simulation à événements discrets "fondée sur la gestion d'un échéancier ou liste d'événements planifiés, ordonnés par date de survenance croissantes. L'exécution de la simulation est alors dynamiquement guidée par cet échéancier dont les événements sont exploités par ordre croissant de leur date de survenance".

"Les paramètres de l'outil d'exécution servent essentiellement à définir la durée de la simulation. Un rapport peut être produit à la demande de l'utilisateur. Il s'agit d'une trace de la simulation qui reprend chronologiquement tous les événements détectés en cours de la simulation lorsqu'ils sont devenus courants". Comme illustré à la figure 2, une ligne de cette trace reprend pour un événement :

- la date de survenance
- le type d'événement
- le type de l'objet concerné ainsi qu'un numéro d'occurrence

```

Execution of simulation
Simulation name : Cas Bancaire
Executed at Jul 10, 1987 15:57:59

Ud09h30m29s   Exec.   Triq. of decision-credit           #28
Ud09h34m20s   Exec.   Term. of preparation-dossier      #20
Ud09h35m21s   Exec.   Term. of entree-demande-pret     #11
Ud09h35m21s   Exec.   Triq. of preparation-dossier     #31
Ud09h35m21s   Exec.   Inco. of preparation-dossier     +31
Ud09h35m21s   Exec.   Triq. of examen-credit          +32
Ud09h35m21s   Exec.   Inco. of examen-credit          +32
Ud09h41m02s   Exec.   Term. of preparation-dossier     +23
Ud09h44m10s   Exec.   Term. of preparation-dossier     #26
Ud09h49m23s   Exec.   Term. of examen-credit          #32
Ud09h49m23s   Exec.   Triq. of decision-credit        #35
Ud09h49m34s   Exec.   Term. of preparation-dossier     #31
Ud09h52m19s   Exec.   Gen. or demande-de-pret         #37
Ud09h52m19s   Exec.   Triq. of entree-demande-pret    #36
Ud09h52m19s   Exec.   Inco. of entree-demande-pret    #38

```

Figure 2 Extrait de la trace de la simulation

Exploitation des résultats statistiques

L'objectif de cet outil est "d'extraire certaines mesures élémentaires de la base de données statistiques et de les traiter pour produire des résultats statistiques. Ces résultats statistiques doivent permettre l'analyse des performances obtenues par la simulation et leur comparaison avec celles attendues par l'utilisateur.

L'approche retenue n'est donc pas de fournir à l'utilisateur des outils d'analyses statistiques (construction d'intervalles de confiance, analyse de séries chronologiques, détection de l'état stationnaire, ...) mais de mettre à sa disposition des outils, restituant le comportement spécifié sous forme de statistiques descriptives.

Il incombera dès lors à l'utilisateur de mener lui-même la stratégie d'analyse des résultats statistiques la plus appropriée sachant qu'il dispose des matériaux de base pour l'aider dans sa démarche".

Ce troisième module permet donc de produire à la demande, au départ de la base de données statistiques des résultats statistiques caractérisant le comportement d'un ou plusieurs type d'objets sur une période donnée.

"Les résultats statistiques qui apparaissent dans les rapports produits se présentent principalement sous la forme de :

- nombres de changements d'état, caractéristiques du comportement des objets considérés (exemple : nombre de déclenchements)
- durées de certains états caractéristiques par lesquels évoluent les objets considérés (exemple : durée d'attente d'un processus)
- volume ou nombre d'objets dans certains états caractéristiques à un moment donné (exemple : nombre de processus en attente à l'instant t ou capacité disponible d'une ressource à l'instant t)

Les résultats statistiques concernant des durées et des volumes mettent systématiquement en évidence les valeurs minimales, moyennes, maximales et écart-type de la distribution de probabilité correspondante.

Enfin, tout résultat statistique est donné pour une période de référence qui est fixée en précisant sa date de début et sa date de fin. Cela signifie qu'elle peut ne pas nécessairement correspondre à la durée simulée. Cela permet dès lors de ne s'intéresser qu'à certaines périodes "critiques" de la période simulée et, par exemple, de ne pas prendre en compte certaines mesures correspondant à la "période de chauffe" du modèle - avant que le système n'ait atteint son état d'équilibre."

Les paramètres de cet outil servent à déterminer le contenu des rapports à produire à partir des mesures élémentaires enregistrées dans la base de données statistiques. Ils permettent notamment de mentionner les types d'objets pour lesquels on veut obtenir des résultats statistiques ainsi que les périodes de référence à l'intérieur de la période simulée pour lesquelles on souhaite produire des résultats statistiques. "Ceux-ci peuvent être présentés sous forme de valeurs :

- "globales" pour une période de référence (exemple : du 5^{ème} au 20^{ème} jour de la simulation)
- "périodiques" par période se répétant dans une période plus large (exemple : heure par heure d'une journée moyenne reconstruite à partir de la 5^{ème} à la 20^{ème} journée de la simulation)
- "chronologiques" pour des périodes fines à l'intérieur d'une période de référence plus large (exemple : pour toutes les heures du 5^{ème} au 20^{ème} jour)".

Tous ces résultats sont présentés dans la version actuelle de DSL-SIM (release 2.0R0) sous la forme de tableaux de chiffres dont un exemple est repris à la figure 3.

Global Statistics : Cas Bancaire

PROCESS acceptation-du-pref

REPORTED BETWEEN 10 AND 50

CALENDAR related to acceptation-du-pref

SPRMS FROM 12h TO 13h, FROM 17h TO 18h OR 0 TO 5 DAY PER WEEK

Triggering Number	136
Inception Number	136
Termination Number	136

	Min	Mean	Max	Std Dev
Active Time	2m	2m	2m	0s
Waiting Time	0s	18m17s	52m7s	14m0s
Total Time by INCR/SPMS	0s	52m26s	3h29m41s	20m53s
Idle Time by PROCESS		52m26s		
Elapsed Time		1h17m45s		
Number of PROC in Wt State	0.00	4.02	23.00	5.77
Number of PROC in ACTV State	0.00	0.53	1.00	0.50

Required Resources :

	Capacity		Shareability	
	Max	Used	Max	Used
imprimante-service-expedition	1.00	0.66		0.66

Flow And	UDP And i	Tri	Number of				wait Time	intr Time	idle Time
			Incd	Intr	kesm	Term		by Incr	by Incd/Intr
11h	12h	30	0	0	0	0			
12h	13h	28	58	0	0	58	9m56s		9m30s
13h	14h	17	0	0	0	0			
15h	16h	20	0	0	0	0			
16h	17h	33	0	0	0	0			
17h	18h	2	78	0	0	78	24m30s		1h24m25s

Figure 3. résultats statistiques produits par DSL-SIM

2. L'interprétation graphique de la performance d'un système d'information

La forme de présentation tabulaire des résultats de la simulation pose des difficultés d'interprétation étant donné leur aspect peu convivial. Il est donc nécessaire de disposer d'un environnement plus ergonomique qui permet non seulement une présentation graphique de ces résultats, mais qui présente également un environnement d'exploitation multifenêtres offrant ainsi au concepteur une capacité maximale de navigation dont l'intérêt se révélera prédominant pour les outils que nous proposons.

Nous avons donc procédé à la spécification de différentes formes de représentation attachée à la simulation : une présentation classique en forme de camemberts et de diagrammes à bâtons; mais aussi une présentation sous la forme d'enchaînements qui s'apparentent au dessin d'un graphe.

Ces formes de représentation ont été choisies sur base d'un dialogue avec des utilisateurs de DSL-SIM à la RATP où nous avons fait notre stage et avec les membres de l'équipe IDA.

Ce choix exclusivement empirique aurait pu s'appuyer sur une étude concernant les profils cognitifs d'utilisateurs. Toutefois, tant pour des raisons de formation que de temps, une telle approche était exclue pour nous. En effet, l'étude de profils cognitifs demande des connaissances psychologiques et le travail de plusieurs personnes pendant plusieurs mois.

Les nombreuses discussions avec les utilisateurs du logiciel ont quand même permis de mettre en évidence qu'il fallait des formes de représentation dépouillées, significatives et faciles à comprendre.

Lors de la présentation de ces diagrammes, nous nous intéresserons presque exclusivement à leur contenu informationnel. Mais il ne faut pas oublier qu'ils doivent s'inscrire dans un environnement multifenêtres destiné à recréer interactivement les processus d'interprétation manuelle des rapports produits par DSL-SIM. Il s'agit de donner différentes capacités de libre circulation parmi les différents diagrammes proposés. Cette possibilité de navigation est fondamentale dans le cadre d'une interprétation des résultats de la simulation car le processus d'interprétation est essentiellement un processus non structuré. La possibilité pour le concepteur de pouvoir accéder à tout moment à n'importe quelle information est donc capitale.

Les représentations sous la forme d'enchaînements nous ont conduits à une recherche à deux niveaux. Tout d'abord, d'un point de vue conceptuel, nous avons enrichi le modèle de la dynamique du concept de transaction. Ensuite, au niveau de la construction de ces graphiques, nous nous sommes livrés à une étude approfondie de ce qui existe dans la littérature pour nous rendre compte qu'il n'existe pas de solution générale au problème posé, il n'existe que des éléments de solutions. Nous avons donc développé nos propres algorithmes de construction qui constituent une part importante de notre mémoire.

La première partie de notre mémoire décrit des outils graphiques d'interprétation des mesures relatives aux objets DSL qui interviennent dans la modélisation du comportement fonctionnel du système d'information ou dans la spécification des ressources requises par les traitements.

Le chapitre 1 de cette partie est consacré à la description des fonctionnalités de ces diagrammes qui ont été classés suivant qu'ils représentent des mesures statistiques (globales, périodiques, chronologiques) ou des calendriers.

Le chapitre 2 décrit les bases de données qui mémorisent les informations manipulées par les graphiques décrits dans le chapitre précédent.

La deuxième partie présente des outils de représentation et d'évaluation du comportement du système ayant comme source principale d'information les mesures contenues dans le rapport produit par la simulation.

La nécessité d'une mesure globale de la performance du système, pour évaluer le comportement de ce dernier, nous a amenés à définir le concept de transaction qui complète le modèle de la dynamique. Cette nouvelle notion fera l'objet d'une définition précise dans le chapitre 3 où l'on donne également l'algorithme qui permet sa construction en puisant les informations dans le rapport produit par la simulation.

Le chapitre 4 présente des outils de représentation graphique du comportement du système. Différents aspects de ce comportement y sont abordés, à savoir la réaction du système d'information à un événement déterminé et le comportement d'un point de synchronisation.

Le chapitre 5 présente une démarche générale d'évaluation du comportement du système d'information ayant le schéma de la dynamique comme support méthodologique.

Lors de cette analyse, on peut être conduit à examiner plus finement certains aspects du comportement. A cet effet, nous proposons également un diagramme qui montre l'évolution de la performance du système d'information à différents moments.

Le chapitre 6 décrit les bases de données qui mémorisent les informations extraites de la trace de la simulation et manipulées par les outils décrits dans les deux chapitres précédents.

La troisième partie de ce mémoire est consacrée à la construction des diagrammes qui se présentent sous la forme d'enchaînements.

Le chapitre 7 décrit une méthode originale de production automatique de graphes hiérarchisés. Les étapes qui composent cette méthode sont tout d'abord présentées intuitivement. Nous expliquons ensuite en détail l'algorithme sous-jacente.

Ces algorithmes permettent la construction de certaines formes de représentation décrites au chapitre 4 mais ne permettent pas tels quels la construction du schéma de la dynamique. Nous avons donc adapté au chapitre 8 la méthode de production automatique de graphes hiérarchisés à la production du schéma de la dynamique.

Le mémoire se termine par une annexe qui reprend les tables qui permettent la représentation interne du schéma de la dynamique ainsi qu'une illustration complète de la méthode de production automatique de ce diagramme.

Compte tenu de l'ampleur du travail accompli et du temps nécessaire à la maîtrise de l'outil de programmation d'interface homme machine MS-WINDOWS et en accord avec notre promoteur, nous n'avons pas pu aborder le problème de la programmation.

3. Définitions de base

L'objectif de ce paragraphe est de fixer la terminologie employée par la suite dans la description des outils d'interprétation. Il ne s'agit pas pour autant de rappeler les concepts sous-jacents aux modèles de la dynamique et des ressources sur lesquels se fonde l'outil d'interprétation (1), ni même de présenter les mesures statistiques produites par DSL-SIM (2), mais simplement de donner quelques éléments clés de l'interprétation.

Dans un premier point, nous illustrerons le cycle de vie d'un processus, nous définirons ensuite la notion de ressource critique, et pour terminer, nous préciserons différentes formes d'événements contraignants.

1. Cycle de vie d'un processus

Dans l'état actuel de développement de l'outil de simulation, un processus peut se trouver dans quatre états :

- l'état déclenché : le processus est créé mais en attente d'exécution.

Si le processus est en attente d'exécution par défaut de capacité ou de point d'entrée d'une ressource qu'il requiert, nous dirons que ce processus est en attente avant activation.

Si le processus est en attente d'exécution par défaut de calendrier, nous dirons que ce processus est en repos avant activation.

- l'état actif : le processus dispose de toutes les ressources nécessaires à son fonctionnement, et est en cours d'exécution (3).

- l'état interrompu : l'exécution du processus est suspendue et ce dernier est à nouveau en attente d'exécution

Si le processus interrompu est en attente d'exécution par défaut de capacité ou de point d'entrée d'une ressource qu'il requiert, nous dirons que ce processus est en attente avant réactivation.

Si le processus interrompu est en attente d'exécution par défaut de calendrier, nous dirons que ce processus est en repos avant réactivation.

- l'état terminé : le processus cesse d'exister après avoir atteint son terme

Par conséquent, un processus est en attente lorsque la disponibilité d'au moins une ressource qu'il requiert est insuffisante par manque de capacité ou par manque de point d'entrée. De même un processus est en repos lorsque le moment où la ressource est requise correspond à une période du calendrier d'indisponibilité de la ressource.

(1) Le lecteur trouvera une définition précise de ces modèles dans [Bodart, 87]

(2) Les résultats statistiques sont définis dans [DSL-SIM, 86]

(3) Une des hypothèses à la base du modèle de la simulation est l'allocation globale des ressources aux processus

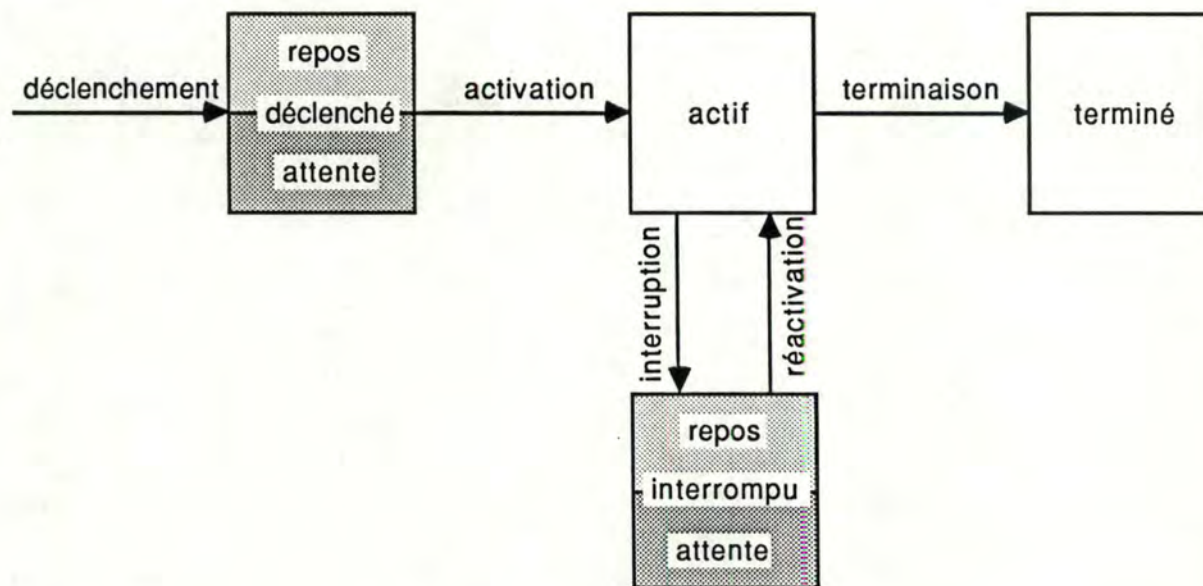


Figure 4 cycle de vie d'un processus

2. Ressource critique

Une ressource est dite critique lorsqu'elle est la cause d'attente et/ou d'interruptions intempestives ou anormalement longues pour les processus qui la requiert.

Ce mauvais comportement peut être attribué à :

- une trop forte utilisation ou une trop faible capacité disponible
- un manque de points d'entrée
- un calendrier de disponibilité trop contraignant

3. Événement contraignant

Un événement contribuant à une synchronisation est dit contraignant s'il a participé en dernier lieu à la réalisation.

Un type d'événement contribuant à un point de synchronisation est dit contraignant s'il est le plus contraignant en fréquence et/ou en durée.

Un type d'événement est le plus contraignant en fréquence si la majorité des événements contraignants sont de ce type-là.

Un type d'événement est le plus contraignant en durée si le total des durées de participation des événements de ce type est inférieur à tous les autres.

- PARTIE I -

Outils d'interprétation des mesures
relatives aux objets DSL

INTRODUCTION

Cette première partie présente un certain nombre de diagrammes ayant pour objet d'aider le concepteur à interpréter les résultats de la simulation.

Ces formes de représentation sont le résultat d'un dialogue avec les utilisateurs et les membres de l'équipe IDA. Au terme de nos discussions, nous avons choisi les formes de représentation les plus simples, les plus dépouillées et les plus faciles à comprendre.

Comme nous le verrons, la possibilité pour le concepteur de pouvoir accéder à tout moment à n'importe quelle information et de pouvoir comparer sur un même écran des données relatives à deux graphiques différents est particulièrement intéressante pour les informations traitées ici.

De plus, comme nous l'avons déjà souligné dans le préambule, cette possibilité de navigation est fondamentale car le processus d'interprétation est essentiellement un processus non structuré.

Dans le chapitre I de cette partie, nous aborderons les différentes formes de représentation des mesures statistiques. Nous décrirons ensuite dans le chapitre II les structures de données mémorisant les informations représentées par les diagrammes décrits dans le premier chapitre.

- CHAPITRE I-

Outils d'interprétation

INTRODUCTION

Ce chapitre présente des diagrammes ayant pour objectif d'aider le concepteur dans sa tâche d'interprétation des résultats produits par l'outil de simulation.

Nous distinguons essentiellement trois formes de représentation : les camemberts pour les mesures globales, les diagrammes à bâtons pour les mesures chronologiques et périodiques et enfin des segments de droites pour la juxtaposition des calendriers.

Sous cette classification, dans le point objectif, nous examinerons systématiquement le rôle joué par ces mesures dans le processus d'interprétation ainsi qu'une justification des formes de représentation choisies.

Nous procéderons ensuite à la description complète des différents outils en détaillant les informations présentées par chaque outil et les fonctionnalités qu'ils doivent offrir à l'utilisateur.

Pour éviter une redondance au niveau de la description des outils concernant les mesures statistiques, nous avons adopté la découpe suivante :

1. Informations
 - 1.1. Partie commentaire (commune à tous les outils)
 - 1.2. Partie semi-graphique
 - 1.2.1. Type de processus
 - 1.2.2. Ressource
 - 1.2.3. Point de synchronisation (commune à tous les outils)
2. Fonctionnalités

Les informations présentées par les outils proposés dans ce chapitre seront reprises sous la forme d'un tableau en guise de conclusion de chapitre.

I.1. MESURES GLOBALES

I.1.1. Objectifs

Lors d'une première analyse du comportement d'un système, il est intéressant de disposer de mesures rendant compte du comportement global des processus d'un type donné, des points de synchronisation et des ressources du système considéré pour identifier rapidement ceux qui présentent un comportement pathologique et qui nécessitent une analyse plus approfondie faisant apparaître les causes de leur mauvais comportement.

Pour mener à bien cette sélection, nous proposons des outils semi-graphiques qui permettent à l'utilisateur, d'une part, d'identifier visuellement et donc rapidement un objet présentant un mauvais comportement, et qui, d'autre part, visualisent les premières informations permettant à l'utilisateur de confirmer ou d'infirmer sa première impression.

Le support graphique utilisé par ces outils est le camembert. Cette représentation permet de mettre en évidence des données dont la valeur se distingue des autres valeurs présentées, ce qui convient particulièrement bien pour les mesures manipulées ici.

Ainsi, par exemple, l'utilisateur peut facilement identifier une proportion d'attente importante par rapport à la durée totale d'un processus ou encore remarquer une ressource dont le taux d'utilisation moyen est particulièrement élevé (cf. figure I.1).



figure 1.1

Une proportion d'attente ou un taux d'utilisation important n'est pas nécessairement synonyme de problème. Une proportion d'attente peut être importante sans que la durée d'attente soit élevée; de même, un taux d'utilisation important ne signifie pas que la capacité résiduelle est faible. Il faut donc être très prudent dans l'interprétation d'un tel graphique et tenir compte des valeurs exactes sur lesquelles les pourcentages sont calculés. Par conséquent, il est nécessaire de donner à l'utilisateur, en plus des pourcentages, la valeur de la mesure représentée.

Certaines informations se prêtent mal à une représentation graphique; par exemple, le nombre de déclenchements, d'activations et de terminaisons de processus d'un même type. Lorsque ces mesures constituent une information importante pour l'interprétation, ce qui est le cas ici, elles sont adjointes sous la forme de texte au graphique proprement dit.

Pour suggérer un comportement anormal, il n'est pas nécessaire d'entrer dans des distinctions subtiles comme, par exemple, différencier l'attente avant activation de l'attente avant réactivation. Par conséquent, certaines mesures sont représentées de manière cumulées. Si l'utilisateur préfère des valeurs non cumulées, il peut toujours les consulter dans le listing produit par le programme d'exploitation des résultats statistiques.

Outre cet aspect interprétation, comme les outils présentés ici donnent une vue agréable et synthétique des aspects clés du comportement du modèle simulé, ils constituent, avec l'outil de juxtaposition d'information sur le schéma de la dynamique (cf VI.1.2.), un moyen de communication privilégié.

1.1.2. Description

Nous allons décrire trois outils d'interprétation des mesures globales qui se distinguent selon le type d'objet considéré (processus, ressource, point de synchronisation).

La description de ces outils comprend : le détail des informations affichées ainsi qu'une présentation des fonctionnalités qu'ils offrent à l'utilisateur.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent soit, sous la forme d'un texte dans une zone de commentaire, soit sous une forme semi-graphique dans une seconde partie.

1.1. Partie commentaire

Tous les outils d'interprétation des mesures globales ont une partie commentaire qui contient des informations relatives

- | | |
|----------------------------|--|
| -à la simulation | (le titre, la date d'exécution) |
| -aux mesures statistiques | (le type de statistique, la période durant laquelle les statistiques globales ont été calculées) |
| -au type d'objet considéré | (nom du type de processus, de la ressource ou du point de synchronisation) |

-au calendrier

(s'il s'agit d'un type de processus, le calendrier représente la période d'activité; dans le cas d'une ressource, il représente son calendrier global de disponibilité.

Lorsque la ressource possède des calendriers particuliers à certains processus et qu'un type de processus a été sélectionné, seul le calendrier particulier au type de processus est repris. Si, par contre, aucun type de processus n'a été sélectionné, tous les calendriers de la ressource sont affichés).

1.2. Partie semi-graphique

Les informations présentées dans cette partie diffèrent selon le type d'objet considéré.

1.2.1. Type de processus**Informations**

Comme illustré à la figure 1.2, cet outil [1] présente sous la forme de deux camemberts

-la proportion des durées moyennes

- d' attente
- de repos
- d' activité

des processus d'un type donné par rapport à la durée de vie moyenne de ces processus.

-la proportion des volumes moyens des processus d'un type donné

- en attente
- en activité

par rapport au nombre moyen de processus du type considéré qui existent simultanément.

Sont repris également, sous la forme d'un texte, les compteurs d'événements représentant les nombres

- de déclenchements
- d' activations
- d' interruptions
- de terminaisons

des processus du type considéré.

Utilités

Le camembert représentant la répartition de la durée de vie moyenne d'un type de processus peut faire apparaître

- une durée d'attente trop importante qui peut être synonyme d'un manque de capacité ou de points d'entrée ou
- une durée de repos trop grande qui peut être due à un calendrier de disponibilité trop contraignant,

pour au moins une des ressources requises par les processus du type considéré.

Il faut faire attention à l'interprétation de ce graphique car une proportion d'attente et/ou de repos importants ne signifient pas nécessairement des temps d'attente et/ou de repos élevés.

De même, si la durée de vie du processus est très grande, une petite proportion d'attente ou de repos représente des durées non négligeables. L'utilisateur veillera donc à rapprocher les pourcentages des durées exactes présentées conjointement au graphique.

Une proportion d'attente ou de repos importante signifie, dans tous les cas, un temps d'inactivité qu'il s'agit de réduire au maximum. Si la durée de vie des processus est grande, cette diminution sera d'autant plus facile à effectuer.

Le camembert représentant les volumes moyens de processus d'un même type dans différents états peut faire apparaître, de manière similaire au graphique précédent et de façon complémentaire, un manque de capacité ou de points d'entrées pour au moins une des ressources requises par les processus du type considéré.

Toutefois, comme on ne dispose pas du nombre de processus en repos, on ne peut pas déceler avec cet outil un problème de calendrier de disponibilité.

Tout comme pour le camembert précédent, l'utilisateur veillera à rapprocher les pourcentages des volumes exacts.

Le nombre de déclenchements, d'activations et de terminaisons des processus du type considéré permet une analyse partielle de la propagation de la charge. Une différence sensible entre ces trois mesures étant déjà l'annonce d'une mauvaise propagation et vraisemblablement synonyme de temps d'attente important.

Une analyse complète de la charge peut être effectuée en analysant ces mêmes mesures mais sur le schéma de la dynamique (cf VI.1).

Le nombre d'interruptions permet, au même titre que la durée d'attente et de repos, d'estimer le caractère contraignant des ressources requises par les processus du type considéré.

Exemple

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Globales concernant les Processus Examen du prêt établies durant la période 1d à 5d
- Actifs de 8h30m à 12h30m , de 13h30m à 17h30m du 0 au 5^{ème} jour de la semaine

156 déclenchements
 156 activations
 32 interruptions
 160 terminaisons

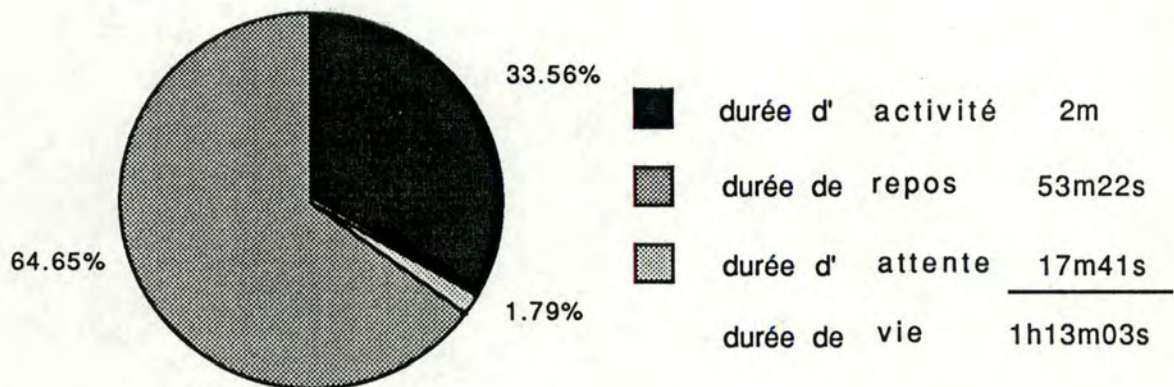
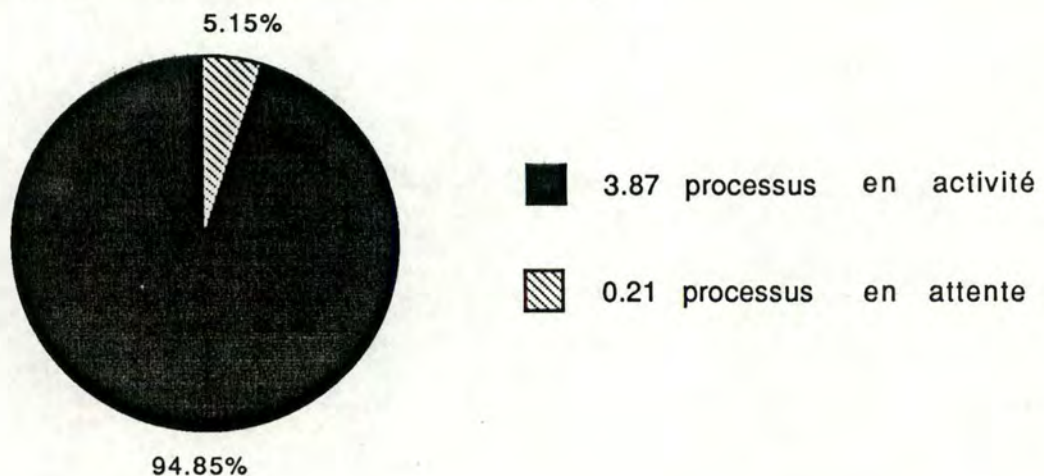
RÉPARTITION DE LA DURÉE DE VIE MOYENNERÉPARTITION DU NOMBRE MOYEN DE PROCESSUS

figure I.2 statistiques globales : processus

1.2.2. Ressource

Informations

Comme illustré à la figure I.3, cet outil [2] présente sous la forme de deux camemberts

-le pourcentage de la capacité maximum théorique d'une ressource qui est

-utilisé en moyenne
-disponible en moyenne

-le pourcentage du nombre maximum de points d'entrée d'une ressource qui est

-utilisé en moyenne
-disponible en moyenne

Nous avons repris également le nombre moyen de processus (de tout type) en attente devant la ressource.

Utilités

Les informations présentées par cet outil permettent de vérifier aisément le caractère contraignant d'une ressource par manque de capacité ou de points d'entrée.

Si l'on constate, d'une part, une file d'attente importante devant la ressource et, d'autre part,

-une utilisation élevée de la ressource (ou une capacité résiduelle faible) ou

-un grand nombre de points d'entrée utilisés (ou un petit nombre de points d'entrée disponibles),

on peut raisonnablement en déduire que la ressource est contraignante.

Les informations présentées par l'outil permettent également de vérifier si une ressource n'est pas sous-utilisée . Il est cependant conseillé à l'utilisateur de vérifier cette hypothèse en analysant les mesures chronologiques car, comme nous le verrons, une ressource quoique faiblement utilisée en moyenne peut être quand même contraignante.

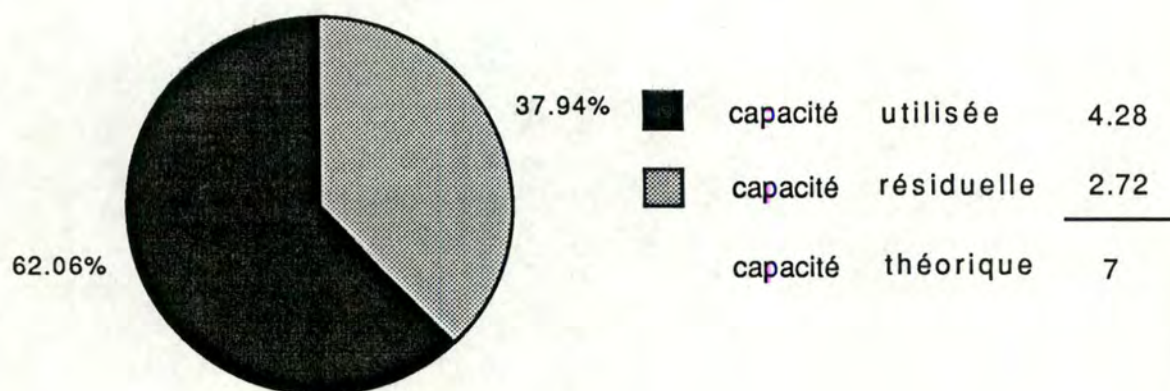
Une ressource peut également être contraignante à cause de son calendrier de disponibilité. Ce qui sera vérifié en examinant les temps de repos des processus (cf. [7]) et les calendriers de disponibilité des ressources requises par ces processus (cf [11]).

Exemple

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Chronologiques concernant la Ressource Service crédit établies durant la période 1d à 5d
- Disponible de 8h30m à 12h30m , de 13h30m à 17h30m du 0 au 5^{ème} jour de la semaine

11.8 processus en moyenne en attente devant la ressource

CAPACITÉ MOYENNE UTILISÉE



NOMBRE MOYEN DE POINTS D'ENTRÉE UTILISÉS

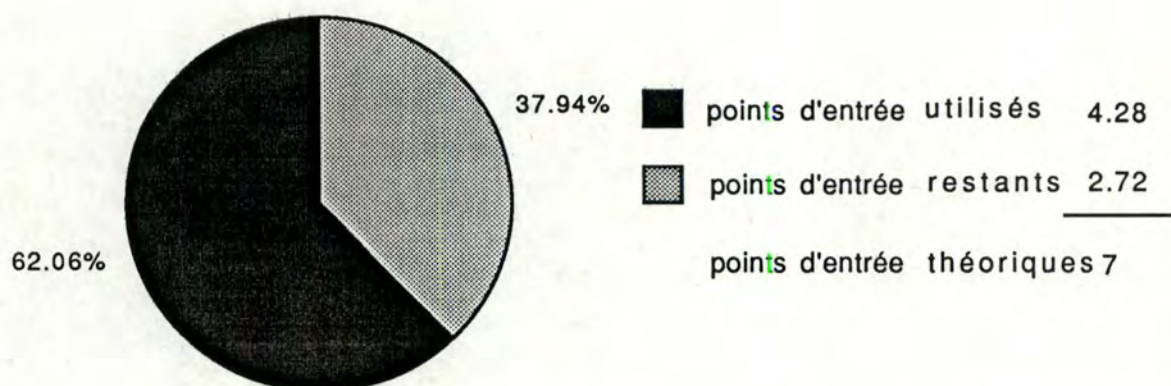


figure I.3 statistiques globales : ressource

1.2.3. Point de synchronisation

Informations

Comme illustré à la figure 1.4, cet outil [3] reprend

- la condition de synchronisation dans le formalisme avec lequel elle a été décrite (cf [DSL-SIM, 86])
- le mode de contribution (simple, multiple)

Nous avons associé à la condition de synchronisation

- le nombre de réalisations et de contributions
- le temps moyen de réalisation et de participation.

D'un point de vue graphique, cet outil présente sous la forme de deux camemberts

- la répartition des événements contraignants en fréquence (1)
- la répartition des événements contraignants en durée (1)

Utilités

Le nombre de réalisations ainsi que la durée moyenne de réalisation permettent d'estimer globalement le comportement du point de synchronisation.

Si ce comportement n'est pas satisfaisant, il convient d'analyser plus finement les autres informations disponibles pour en déterminer la cause. Cette analyse peut se faire en examinant les durées de participation ainsi que le nombre de contributions au point de synchronisation tout en rapprochant ces mesures du prédicat de réalisation et du mode de contribution.

Lors de cette analyse, il est parfois utile de vérifier le mode de mémorisation des événements. Cette information n'est pas reprise par cet outil mais est disponible par exemple dans le rapport produit par le générateur du programme de simulation.

Les deux camemberts permettent, quant à eux, une identification facile des types d'événements contraignants en fréquence et/ou en durée; ce qui nous permet de déterminer, s'il y a lieu, un chemin critique.

(1) Le calcul de cette mesure est expliqué dans [DSL-SIM,]

Exemple

- Simulation théorique exécutée le 10 août 1987 à 10h37m59s
- Statistiques Globales concernant le Point de Synchronisation Synchro-théorique établies durant la période 0d et 2d

<u>CONDITION DE SYNCHRONISATION</u>	<u>NOMBRE MOY. DE RÉALI. / CONTRL.</u>	<u>TEMPS MOY. DE RÉALI. / PARTI.</u>
Réalisé quand et	7	1h18m30s
[1] terminaison de "P1" (simple) et	7	30m10s
[2] réalisation de "S2" (simple)	7	66m45s
[3] génération de "M1" (simple)	7	1h15m30s

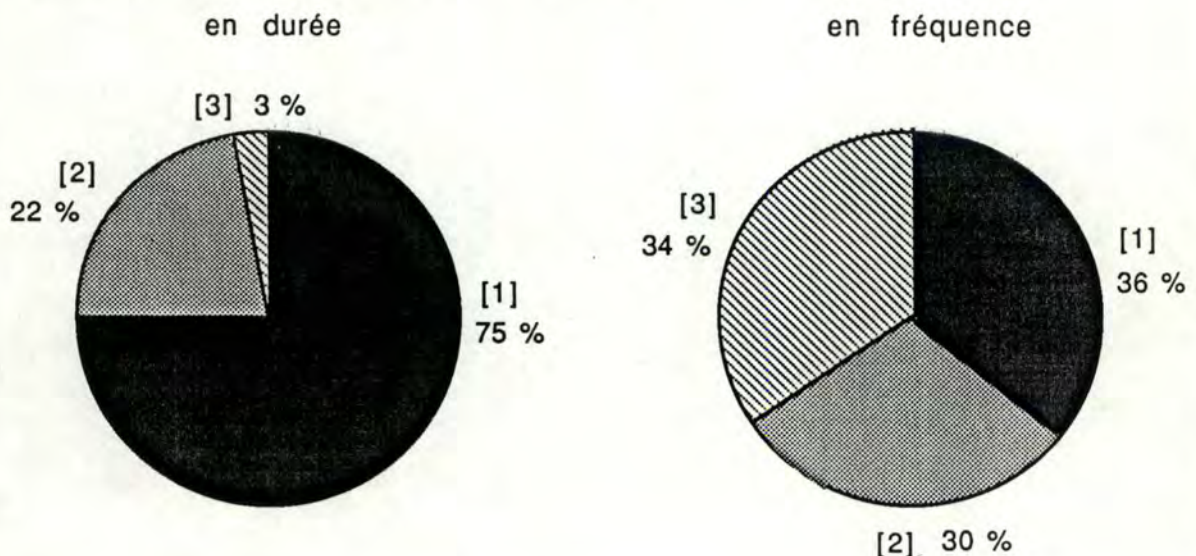
RÉPARTITION DES ÉVÉNEMENTS CONTRAGNANTS

figure I.4 statistiques globales : point de synchronisation

2. FONCTIONNALITES

2.1. Identification d'un type d'objet

La représentation du comportement global d'un type de processus d'une ressource ou d'un point de synchronisation suppose que l'utilisateur ait dans tous les cas spécifié préalablement le type d'objet au sujet duquel les informations sont données.

La sélection d'un type de processus est facilitée en montrant la liste de tous les types de processus faisant partie du modèle simulé et en faisant ressortir ceux qui requièrent une ressource déterminée si celle-ci a été sélectionnée préalablement.

Par exemple, si la ressource R1 a été sélectionnée et qu'elle est requise par les processus P1 et P3, cette manière de faire est illustrée à la figure I.5.

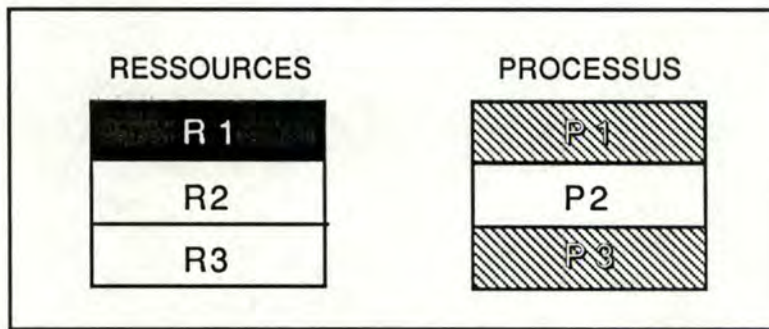


figure I.5 sélection d'un processus

De la même manière, la sélection d'une ressource est facilitée en montrant la liste de toutes les ressources faisant partie du modèle simulé et en faisant ressortir celles requises par un type de processus si ce dernier a été sélectionné auparavant.

2.2. Activation d'autres outils

Tous les outils décrits dans cette première partie doivent pouvoir être activés simultanément.

Par exemple, l'utilisateur doit pouvoir afficher sur un même écran les informations relatives à un type de processus en même temps que les informations sur une ressource requise par les processus du type considéré.

2.3. Gestion de fenêtres

Pour pouvoir juxtaposer sur un même écran des informations relatives à deux outils différents, on doit pouvoir réduire la taille des fenêtres dans lesquelles sont affichées les informations relatives à un outil.

Pour visualiser à nouveau l'entièreté des informations, on doit disposer d'autres fonctions de gestion de fenêtres, en l'occurrence le "scrolling" et le "zooming".

2.4. Afficher/effacer la partie commentaire

Lorsque l'utilisateur active l'un des outils que nous venons de décrire, sont affichées, en plus de la partie graphique, les informations de la zone commentaire de l'outil. S'il désire ensuite récupérer l'espace occupé par ces données, il a la possibilité de les faire disparaître et de les rappeler ultérieurement.

2.5. Impression

L'analyse du comportement d'un type d'objet peut se faire de manière interactive, mais également en examinant ces mêmes informations sur papier.

De plus, étant donné la vue synthétique du comportement que ces outils offrent, la présence de ces informations, sur un support matériel, constitue un moyen de communication agréable entre les personnes intéressées par la modélisation en cours.

1.2. MESURES CHRONOLOGIQUES ET PERIODIQUES

1.2.1. Objectifs

L'intérêt des outils décrits au point 1.1 réside, comme nous l'avons vu, dans la vue synthétique du comportement d'un type de processus, d'une ressource ou encore d'un point de synchronisation qu'ils permettent.

Si cette vue s'avère suffisante pour une première étude du comportement, elle présente certaines lacunes pour analyser plus finement le système modélisé.

Pour les processus, il est en effet intéressant en vue de localiser dans le temps un comportement défaillant, voire même de l'expliquer, de disposer de mesures moyennes calculées dans des périodes d'observation plus fines que la période de calcul des statistiques globales.

Ainsi, par exemple, si l'on dispose du temps moyen de repos d'un type de processus pour toutes les heures d'une journée, on est à même de localiser une durée de repos anormal. Si on constate, d'autre part, un temps de repos élevé vers 17 heures, celui-ci est probablement causé par la fin de la journée.

La même analyse effectuée au niveau des ressources peut, quant à elle, faire apparaître des ressources contraignantes qui n'avaient pas été détectées à l'aide des mesures globales.

A titre d'illustration, supposons qu'une ressource R1 ait une capacité tout juste suffisante pour satisfaire les besoins de 10 processus P1. Dans l'hypothèse où

- la ressource R1 est disponible pendant 1 mois et

- les processus P1 ont été déclenchés en début de mois et se sont terminés endéans les 15 jours,

après une simulation de 1 mois, les statistiques concernant R1 indiqueront une capacité moyenne utilisée à 50% alors que la ressource est contraignante.

A défaut de mesures périodiques et chronologiques concernant les points de synchronisation, nous proposons, au point IV.1, un outil recréant de manière animée ce mécanisme qui peut, en partie, pallier à cette lacune.

L'évolution temporelle de mesures concernant les processus et les ressources se présente sous la forme de mesures chronologiques et périodiques.

Dans les deux cas, il s'agit de décomposer la période de calcul des statistiques en cellules d'observation (cf. figure 1.6). Les statistiques périodiques se distinguent des chronologiques car "les mesures qui tombent dans une cellule d'observation sont cumulées et rapportées dans une seule occurrence de chaque cellule" [DSL-SIM, 86].

Les mesures périodiques, plus compactes, seront préférées aux mesures chronologiques lorsque les phénomènes à observer se produisent périodiquement. Il faut cependant être attentif à l'interprétation de ces mesures car elles ne sont fiables que si le régime permanent est atteint.

Ces deux types de mesures sont actuellement présentées par DSL-SIM sous la forme de tableaux de chiffres particulièrement difficiles à interpréter. Pour faciliter le compulsage de ces données, nous proposons des outils graphiques qui présentent, sous la forme de diagrammes à bâtons, l'évolution de certaines mesures.

Le choix des informations à représenter se situe à plusieurs niveaux :

- Nous n'avons repris que les mesures les plus pertinentes pour une telle analyse. Par exemple, nous n'avons pas retenu l'évolution du nombre de requêtes et d'allocations d'une ressource.
- Certaines mesures sont reprises mais présentées de manière cumulée. Par exemple, nous avons cumulé le temps d'attente avant activation avec celui avant réactivation.
- Nous avons groupé, par outil, les mesures de même nature ou dont la comparaison présente un intérêt. Par exemple, l'utilisateur a la possibilité de comparer le nombre de déclenchements, d'activations et de terminaisons de processus d'un même type pour examiner comment la charge est absorbée par le système.(1)

En ce qui concerne la représentation, nous avons choisi, par souci de simplification, une forme de présentation unique pour toutes les mesures. Cette contrainte nous a conduits à préférer le diagramme à bâtons à une présentation sous la forme de courbes.

Les courbes suscitent, en effet, une extrapolation des mesures qui n'a pas toujours de sens. A titre d'illustration, si la courbe de la figure 1.6 représente le nombre de déclenchements de processus d'un même type, on ne peut pas interpréter ce graphique en prétendant que le nombre de déclenchements à 9 heures est de 15. On peut juste affirmer qu'il y a eu 10 déclenchements de 8 heures à 9 heures et 20 déclenchements de 9 heures à 10 heures.

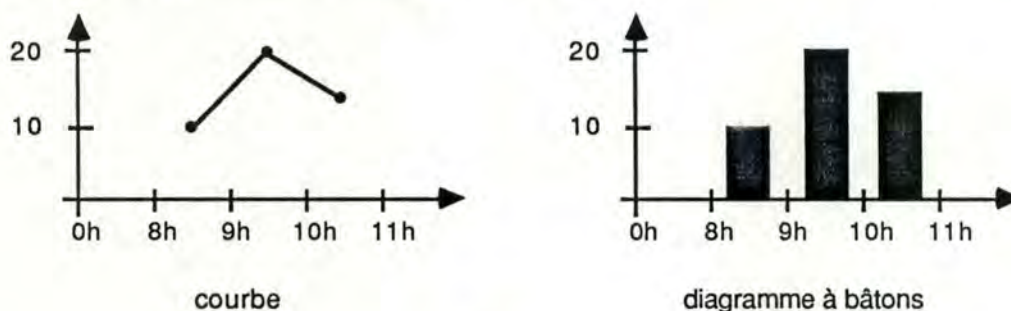


figure 1.6

Le diagramme à bâtons est une représentation plus neutre car il se contente de représenter graphiquement des valeurs sans extrapolations parfois abusives. De plus, il permet une comparaison plus facile de mesures, surtout lorsque les valeurs sont très semblables. Toutefois, si le nombre de mesures différentes juxtaposées sur un tel graphique est important, cela nuit à la lisibilité du dessin, ce qui nous a amenés à limiter le nombre de variables présentes simultanément sur un même schéma.

Dans tous les graphiques présentés dans cette partie, l'axe des abscisses est découpé en cellules d'observation. Cette partition peut se faire en périodes de durées équivalentes (par exemple toutes les heures de la journée) mais rien ne nous empêche de définir des cellules d'observation de durées différentes. Nous avons alors la possibilité de découper l'axe des abscisses en cellules de tailles variables ou préférer une découpe en intervalles constants.

La première alternative présente l'avantage de refléter la durée des cellules tandis que la deuxième permet une répartition des bâtonnets plus harmonieuse. Comme la première découpe pose également un problème de placement des bâtonnets à l'intérieur d'une cellule (cf. figure 1.7), nous avons choisi de découper l'axe des abscisses en intervalles de taille constants. Cette découpe présente, par ailleurs, l'avantage de mettre plus d'informations sur un même espace physique, ce qui n'est pas à négliger lorsque la largeur du dessin est particulièrement grande.

(1) L'utilisateur a également la possibilité de juxtaposer sur un même écran des informations relatives à deux outils différents.

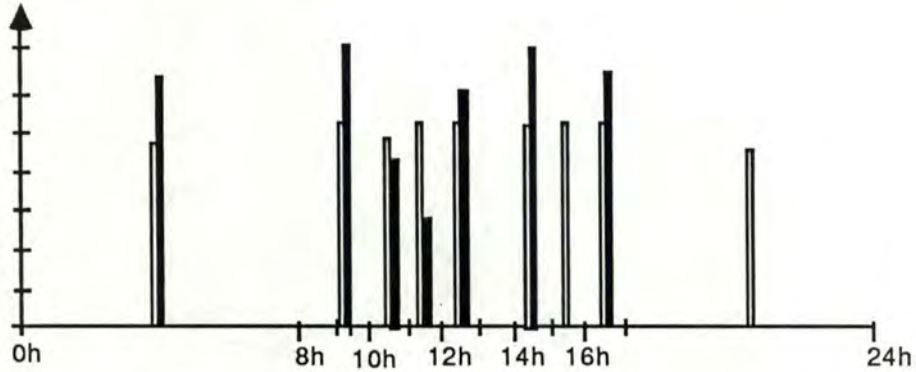


figure 1.7 cellules d'observations de taille variable

1.2.2. Description

La description des outils de représentation de mesures chronologiques et périodiques comprend le détail des informations affichées ainsi qu'une présentation des fonctionnalités qu'ils offrent à l'utilisateur.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent soit, sous la forme d'un texte dans une zone de commentaires, soit sous une forme graphique dans une seconde partie.

1.1. Partie commentaire

Tous les outils de représentation de mesures chronologiques et périodiques ont une partie commentaire qui contient des informations relatives

- à la simulation (le titre, la date d'exécution)
- aux mesures statistiques (le type de statistique, la période durant laquelle les statistiques chronologiques ou périodiques ont été calculées)
- au type d'objet considéré (nom du type de processus ou de la ressource)
- au calendrier (s'il s'agit d'un type de processus, le calendrier représente la période d'activité; dans le cas d'une ressource, il représente son calendrier global de disponibilité.

Lorsque la ressource possède des calendriers particuliers à certains processus et qu'un type de processus a été sélectionné au préalable, seul le calendrier particulier au type de processus est repris. Si, par contre, aucun type de processus n'a été sélectionné, tous les calendriers de la ressource sont affichés).

1.2. Partie graphique

Les informations présentées dans cette partie diffèrent selon qu'il s'agit d'un type de processus ou d'une ressource.

1.2.1. Type de processus

Informations

Ce tableau synoptique reprend les mesures affichées par chacun des outils sous la forme de diagrammes à bâtons.

OUTILS	MESURES REPRÉSENTÉES	TYPE DE MESURES	
		Chrono.	Pério.
[4]	nombre de déclenchements d' activations d' interruptions de terminaisons	X	
[5]	idem		X
[6]	durée moyenne d' attente	X	
[7]	durée moyenne d' attente de repos		X
[8]	nombre moyen de processus en attente en activité	X	

[4] reprend chaque fois le nombre d'événements d'un type donné qui se sont produits à l'intérieur d'une cellule d'observation. Par exemple, le nombre de déclenchements de processus "Examen du prêt" à l'intérieur d'une cellule d'observation s'étendant de 8 heures à 9 heures le premier jour de la semaine, comme illustré à la figure I.11.

[5] reprend les mêmes informations que celles de l'outil précédent mais, cumulées à l'intérieur d'une période. Par exemple, le nombre total de déclenchements des processus d'un type donné de 8 heures à 9 heures tous les lundis d'un mois.

[6] reprend le temps moyen d'attente des processus d'un type donné qui se sont (ré)activés à l'intérieur d'une cellule d'observation.

Il est dommage que DSL-SIM ne produise pas le même type de mesure pour le temps de repos. D'autant plus que cette mesure est reprise, comme nous allons le voir, dans les statistiques périodiques.

[7] reprend le temps moyen d'attente et de repos des processus d'un type donné qui se sont (ré)activés à l'intérieur de cellules d'observation se répétant dans une période plus large.

Une illustration de cet outil est reprise à la figure I.12.

[8] reprend le nombre moyen de processus d'un type donné

- en attente

- en activité

à l'intérieur d'une cellule d'observation.

Ces outils permettent aussi de situer, au moyen d'un symbole placé sur l'axe des ordonnées qui rappelle celui utilisé pour identifier les variables représentées sous la forme de diagrammes à bâtons, les moyennes des mesures que nous venons de décrire calculées sur toutes les mesures présentées dans les cellules d'observation. Par exemple à la figure I.11, le carré noir placé sur l'axe des ordonnées indique le nombre moyen de processus "Examen du prêt" déclenchés dans une cellule d'observation.

De plus, à l'aide d'une fonction décrite ultérieurement, l'utilisateur a la possibilité de situer graphiquement ces valeurs moyennes par rapport aux valeurs minimales et maximales apparaissant dans une cellule d'observation. Cette fonctionnalité est illustrée à la figure I.8 pour le nombre de déclenchements de processus "Examen du prêt".

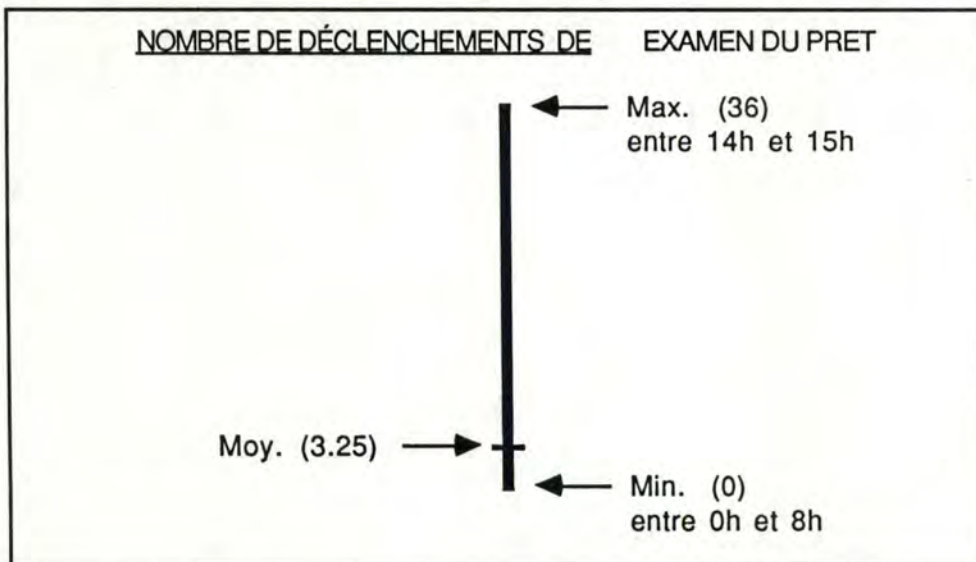


figure I.8 situer la valeur moyenne

Utilités

- [4] Les informations présentées par cet outil, à l'exception du nombre d'interruptions, permettent une analyse détaillée de l'absorption de la charge. On peut en effet vérifier non seulement que tous les processus déclenchés ont pu démarrer, s'activer et se terminer mais aussi avoir une idée de la rapidité avec laquelle la charge est absorbée et tester ainsi la vitesse de réaction du système d'information.

Ce phénomène est illustré caricaturalement à la figure I.9 où l'on voit, d'un côté, une charge totalement absorbée en 4 heures et de l'autre, la même charge résorbée en deux fois plus de temps.

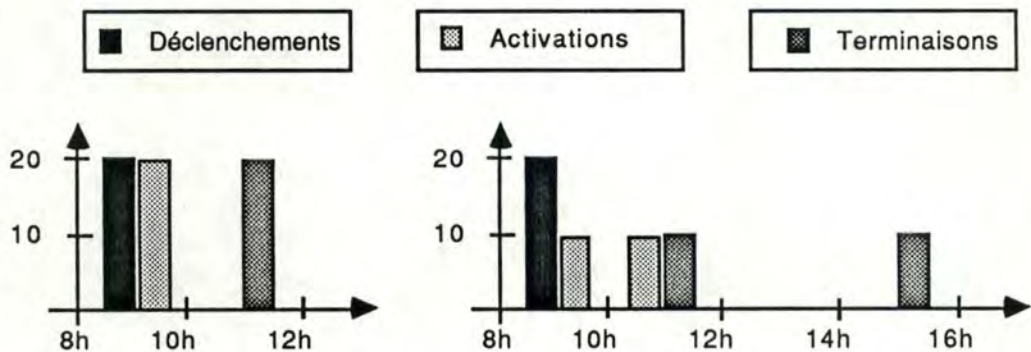


figure I.9 absorption de la charge

La répartition du nombre d'interruptions par cellules d'observation permet d'évaluer et de localiser la caractéristique contraignante des ressources requises par les processus du type considéré.

- [5] Cet outil sera préféré à l'outil [4] lorsqu'on est en présence de phénomènes périodiques.
- [6] Le temps moyen d'attente permet de montrer un problème de capacité et/ou de points d'entrée. Comme DSL-SIM ne produit pas de statistiques chronologiques concernant le temps de repos, on ne peut pas localiser, de manière précise, un problème de calendrier qui a été décelé à l'aide des statistiques globales.

Il faut être attentif à l'interprétation de ce graphique car le temps d'attente est représenté graphiquement au moment de l'activation ou de la réactivation. Il existe donc un décalage entre le moment où les processus sont en attente et la représentation de ces durées.

L'exemple de la figure I.10 montre un temps moyen d'attente de 10 heures pour les processus qui se sont (ré)activés entre 16 heures et 18 heures mais cette attente est présente également dans les cellules d'observation précédentes. Dans l'hypothèse où les processus ont été déclenchés à 8 heures, à 10 heures ils ont attendu 2 heures, à 12 heures leur attente se porte à 4 heures,...

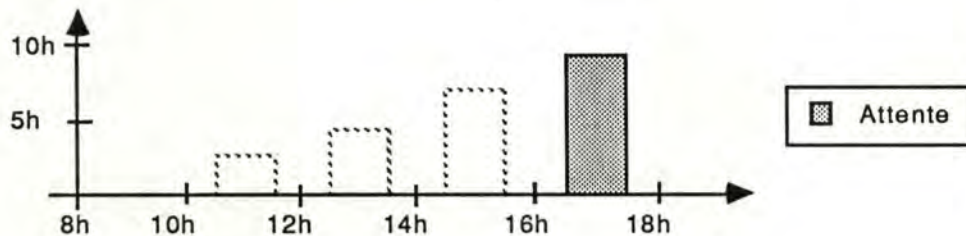


figure I.10 évolution du temps d'attente

Un temps d'attente important peut être expliqué par un nombre de déclenchements élevé. Il est donc nécessaire de pouvoir juxtaposer sur un même écran des informations relatives à deux outils différents, en l'occurrence ici : le nombre de déclenchements [4] et le temps d'attente [6]. Cette technique d'interprétation est rendue possible à l'aide d'une fonction décrite ultérieurement.

- [7] L'évolution du temps moyen d'attente permet d'identifier un problème de capacité et/ou de points d'entrée pour au moins une ressource requise par les processus.

La répartition des temps de repos facilite, quant à elle, une localisation d'un problème de calendrier pour au moins une des ressources requises par les processus.

Tout comme pour l'outil précédent, une attention particulière doit être apportée à l'interprétation de ce graphique car il existe un décalage entre le moment où les processus sont en attente ou en repos et la représentation de ces durées.

- [8] L'évolution du nombre moyen de processus en attente peut faire apparaître, de manière similaire à l'outil [6] et de façon complémentaire, un manque de capacité ou de points d'entrée pour au moins une des ressources requises par les processus du type considéré.

Toutefois, comme on ne dispose pas du nombre de processus en repos, on ne peut pas localiser un problème de calendrier de disponibilité.

L'évolution du nombre moyen de processus en activité permet, quant à lui, d'expliquer, en partie, un nombre important de processus en attente. En effet, si le nombre de processus activés est important, il est vraisemblable que le nombre de processus en attente grandisse également.

Exemples

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Chronologiques concernant les Processus Examen du prêt établies durant la période 1d à 5d
- Actifs de 8h30m à 12h30m, de 13h30m à 17h30m du 0 au 5^{ème} jour de la semaine

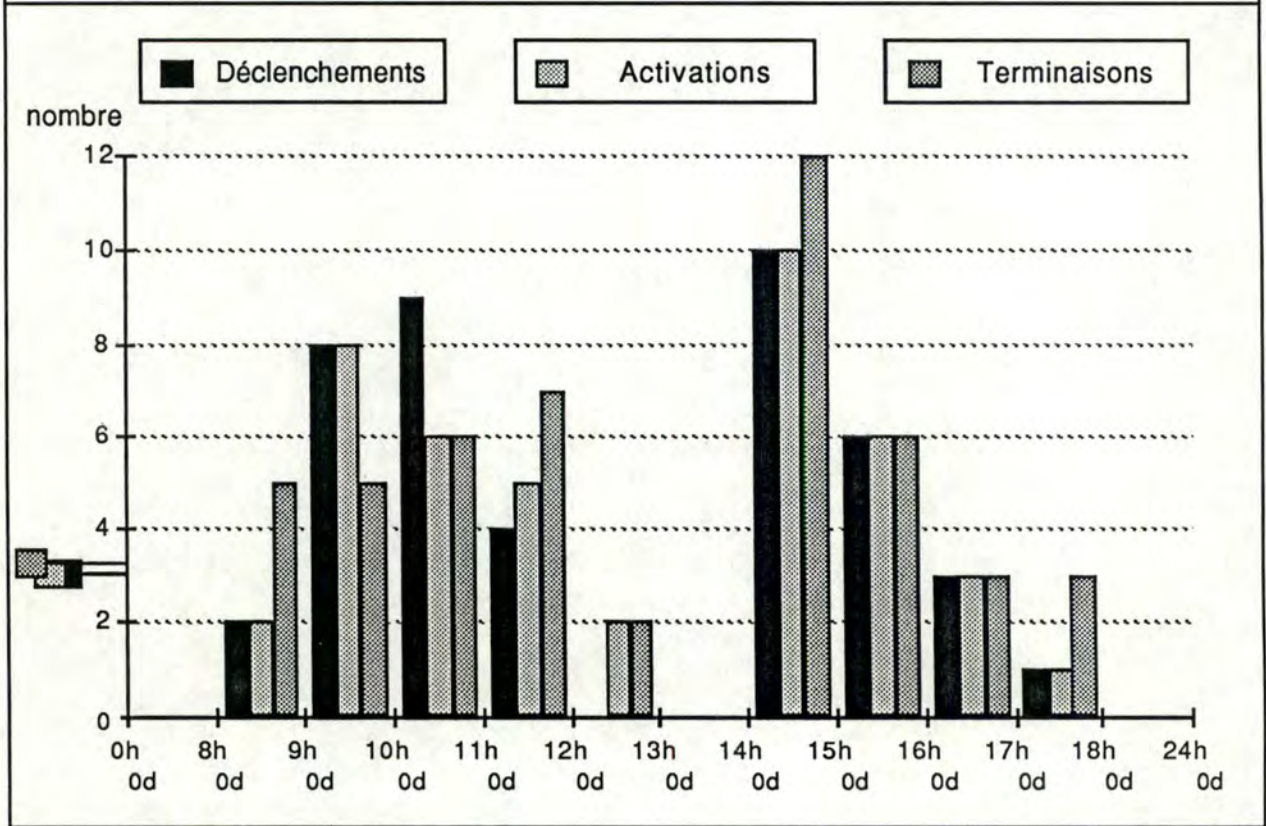


figure 1.11 statistiques chronologiques : processus

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Périodiques concernant les Processus Examen du prêt établies durant la période 1d à 5d
- Actifs de 8h30m à 12h30m , de 13h30m à 17h30m du 0 au 5^{ème} jour de la semaine

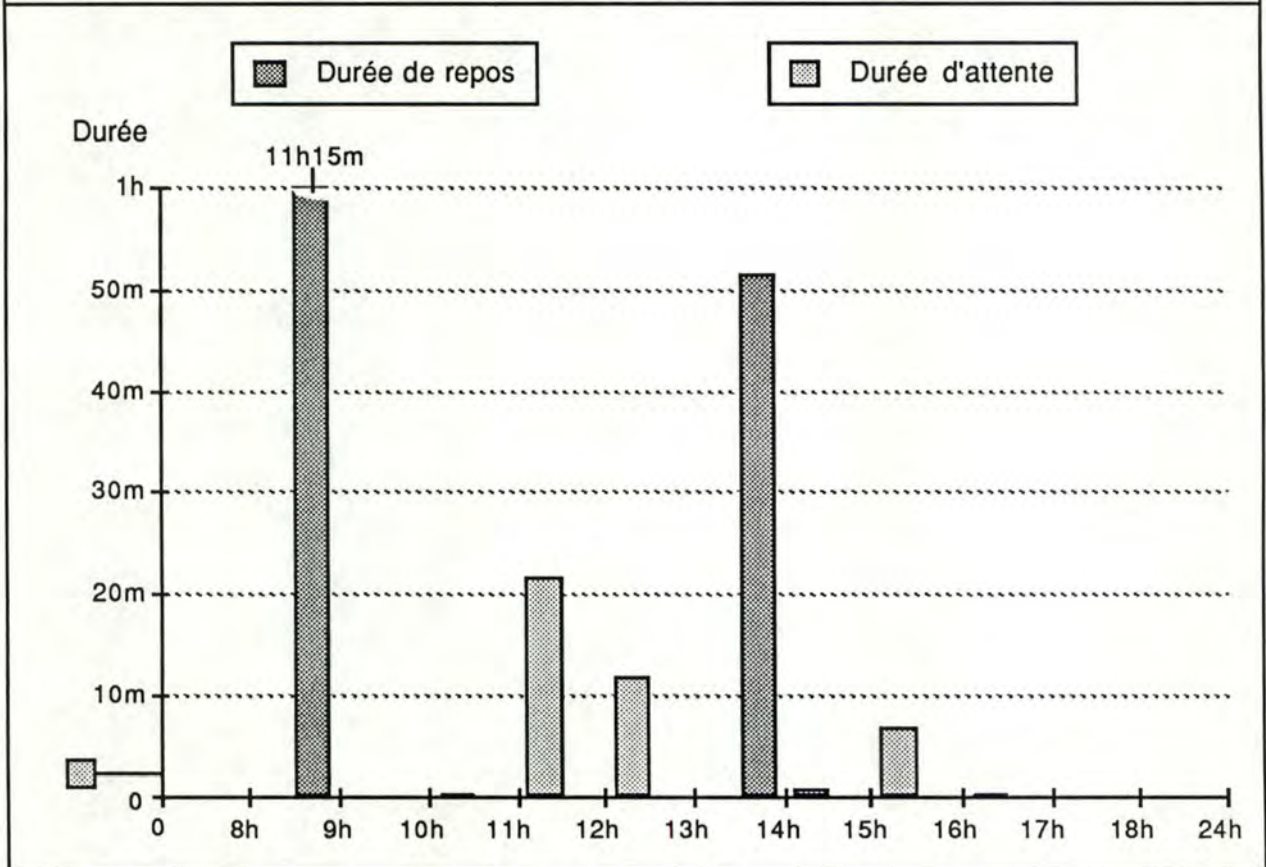


figure 1.12 statistiques périodiques : processus

1.2.2. Ressource

Informations

Un premier outil [9] représente

- la capacité et
- le nombre de points d'entrée

utilisés en moyenne dans une cellule d'observation.

Le deuxième outil [10] représente le nombre moyen de processus de tout type en attente devant la ressource.

Comme pour les processus, ces deux outils placent, sur l'axe des ordonnées, des valeurs représentant les moyennes calculées sur toutes les cellules d'observation des mesures affichées. Par exemple, à la figure 1.13, le carré noir placé sur l'axe des ordonnées désigne la capacité moyenne utilisée dans une cellule d'observation.

Il est également possible de situer ces valeurs de référence par rapport aux valeurs minimales et maximales apparaissant dans une cellule d'observation.

Utilités

On peut imaginer deux utilisations différentes des outils que nous venons de décrire suivant qu'on désire analyser le comportement du système d'information à partir des ressources du modèle simulé ou au travers des traitements effectués.

L'examen systématique de toutes les ressources, au moyen de l'outil [9], permet de détecter celles qui sont sous-utilisées.

La même démarche peut être effectuée pour détecter les ressources critiques mais en utilisant conjointement les outils [9] et [10]. Nous verrons, dans les fonctionnalités, qu'il est possible de présenter sur un même écran des informations relatives à deux outils différents. Cette juxtaposition est nécessaire car une ressource est contraignante à un moment donné si l'on constate

- une attente des processus qui la requiert, ce qui correspond à une file d'attente importante devant la ressource et
- un taux d'utilisation élevé et/ou une saturation du nombre de points d'entrée.

On peut également choisir d'analyser le comportement du système en se référant d'abord aux processus. Dans ce cas, lorsque l'on constate un temps d'attente important pour un type de processus, à l'aide des outils [6] ou [7], en examinant la capacité et le nombre de points d'entrée des ressources requises par ces processus [9], on pourra déterminer la ou les ressources contraignantes.

La première démarche sera préférée à la seconde lorsque l'on a déjà une idée, à priori, des ressources critiques.

Exemple

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Chronologiques concernant la Ressource Service crédit établies durant la période 1d à 5d
- Disponible de 8h30m à 12h30m, de 13h30m à 17h30m du 0 au 5 ^{ème} jour de la semaine

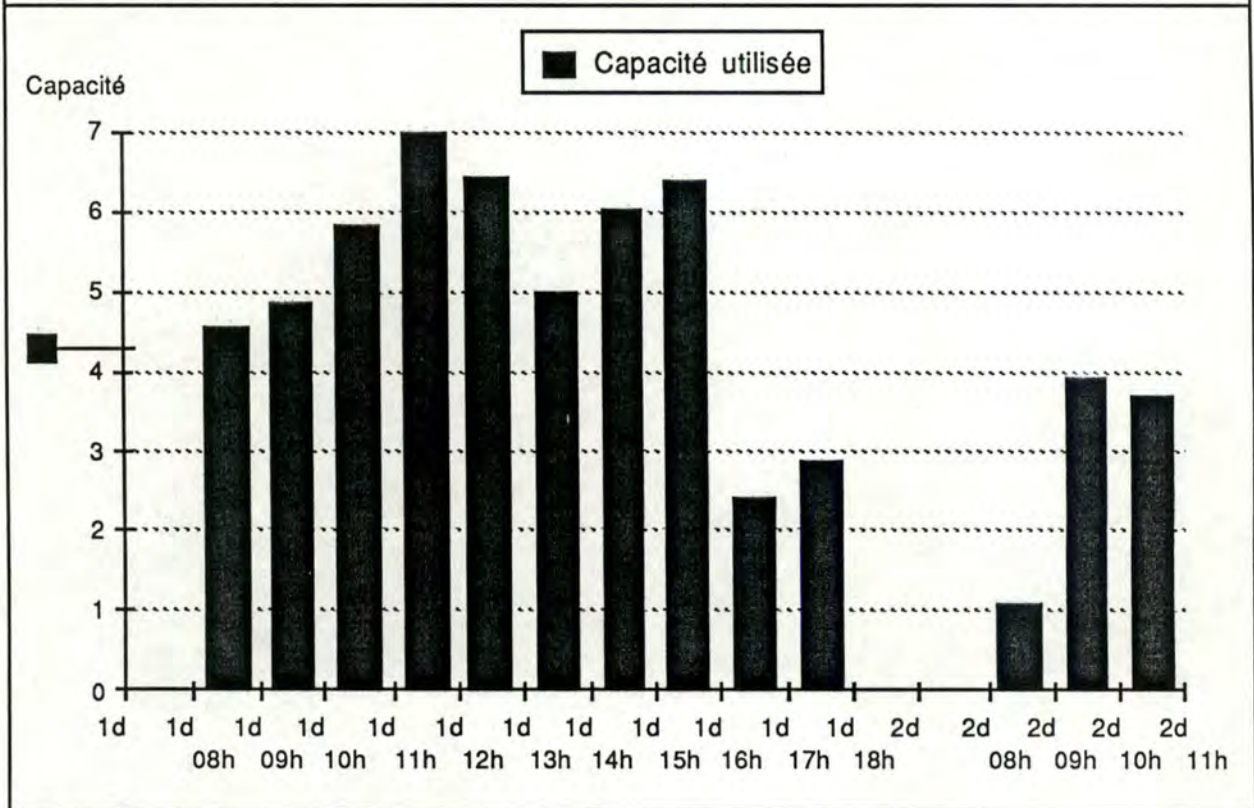


figure I.13 statistiques chronologiques : ressource

2. FONCTIONNALITES

2.1. Identification d'un type d'objet

L'identification du type de processus ou de la ressource pour lesquels on désire visualiser des mesures chronologiques ou périodiques se fait de la même manière que pour les mesures globales.

2.2. Sélection de mesures

Pour éviter de surcharger inutilement un graphique, l'utilisateur a la possibilité de sélectionner des mesures parmi celles proposées par un outil.

Ainsi, par exemple, il peut demander de ne représenter que le nombre de déclenchements de processus dans l'outil [4] ou de ne visualiser que la capacité utilisée dans l'outil [9].

2.3. Activation d'autres outils

Cf. fonctionnalité 2.2. mesures globales.

2.4. Gestion de fenêtre

La taille du graphique peut être largement supérieure à l'espace physique dont on dispose, il est donc nécessaire d'offrir à l'utilisateur la possibilité de faire du "scrolling" horizontal et lui permettre ainsi de parcourir toute la zone d'observation.

De même, pour pouvoir présenter des informations relatives à deux outils différents sur un même écran, on doit pouvoir agir sur la taille et la position de la fenêtre.

2.5. Zooming des axes

En ce qui concerne l'axe des abscisses, l'utilisateur doit pouvoir déterminer la période qu'il désire voir représentée dans une fenêtre.

Ainsi, par exemple, s'il a spécifié une période de calcul de mesures chronologiques s'étendant sur 4 jours, il peut, dans un premier temps, vouloir visualiser sur un même écran l'évolution des mesures sur toute cette durée. S'il constate un problème à un certain moment, il demandera de faire un "zoom" sur la sous-période qui l'intéresse. A titre d'illustration, dans l'exemple de la figure 1.13, nous avons fait un "zoom" sur une période de 35 heures.

La même fonctionnalité est disponible sur l'axe des ordonnées. Cette fonction est surtout utile lorsque la découpe n'est pas assez précise par rapport aux mesures affichées dans la fenêtre. Comme l'axe des ordonnées est découpé à partir de la valeur maximale des mesures affichées, cette situation se produit lorsque les mesures divergent très fort les unes des autres

Par exemple à la figure 1.14, une graduation s'étendant jusque 50 permet une analyse plus fine des données, qu'une sur 150.

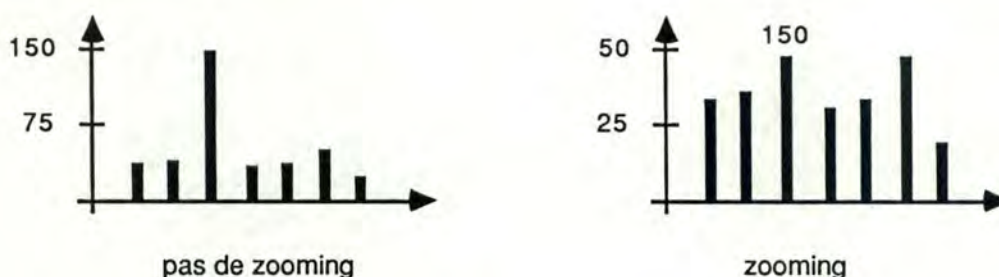


figure 1.14

Lorsqu'un bâtonnet est tronqué suite à un "zoom" sur l'axe des ordonnées, la valeur qu'il représente est affichée sur ce bâtonnet, comme illustré ci-dessus.

2.6. Situer les valeurs moyennes

Cette fonction permet de situer les valeurs moyennes présentes sur l'axe des ordonnées par rapport aux valeurs minimales et maximales de ces mesures. Cette fonction est illustrée à la figure 1.8 pour le nombre moyen de déclenchements.

2.7. Afficher/effacer la partie commentaire

Cf. fonctionnalité 2.5. mesures globales.

2.8. Impression

Les outils que nous venons de présenter se prêtent aussi bien à une utilisation interactive qu'à une analyse sur papier. Toutefois, lorsque la période à observer est particulièrement longue, une analyse sur papier est probablement plus rapide.

De plus, si l'utilisateur a déjà parcouru la période d'observation interactivement, il doit pouvoir imprimer de manière sélective la ou les sous-périodes critiques.

I.3. CALENDRIERS

I.3.1. Objectif

L'objectif de cet outil [11] est de détecter les ressources dont le calendrier de disponibilité est contraignant.

A cet effet, nous proposons de représenter graphiquement, sous forme de droites, les calendriers de disponibilité des ressources requises par les processus d'un type donné.

Chacun de ces calendriers représente le calendrier global de disponibilité de la ressource ou un calendrier particulier aux processus, s'il existe.

I.3.2. Description

La description de cet outil comprend le détail des informations affichées ainsi qu'une présentation des fonctionnalités offertes à l'utilisateur.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent soit, sous la forme d'un texte dans une zone de commentaire, soit sous une forme graphique dans une seconde partie.

1.1. Partie commentaire

Cette partie reprend des informations concernant la simulation, en l'occurrence : le titre et la date d'exécution.

1.2. Partie graphique

Informations

Comme illustré à la figure I.15, la partie graphique de cet outil reprend :

- les calendriers de disponibilité des ressources requises par un type de processus
- le calendrier d'activité du type de processus retenu.

Ces calendriers sont représentés sur toute la durée de la simulation. Cette période étant la seule intéressante au niveau de l'interprétation.

Utilité

Si l'on constate pour un type de processus des temps de repos importants, à l'aide de l'outil [1] ou [7], en consultant les calendriers de disponibilité des ressources requises par ces processus, on est à même de déterminer aisément la ressource qui a la disponibilité la plus contraignante.

Exemple

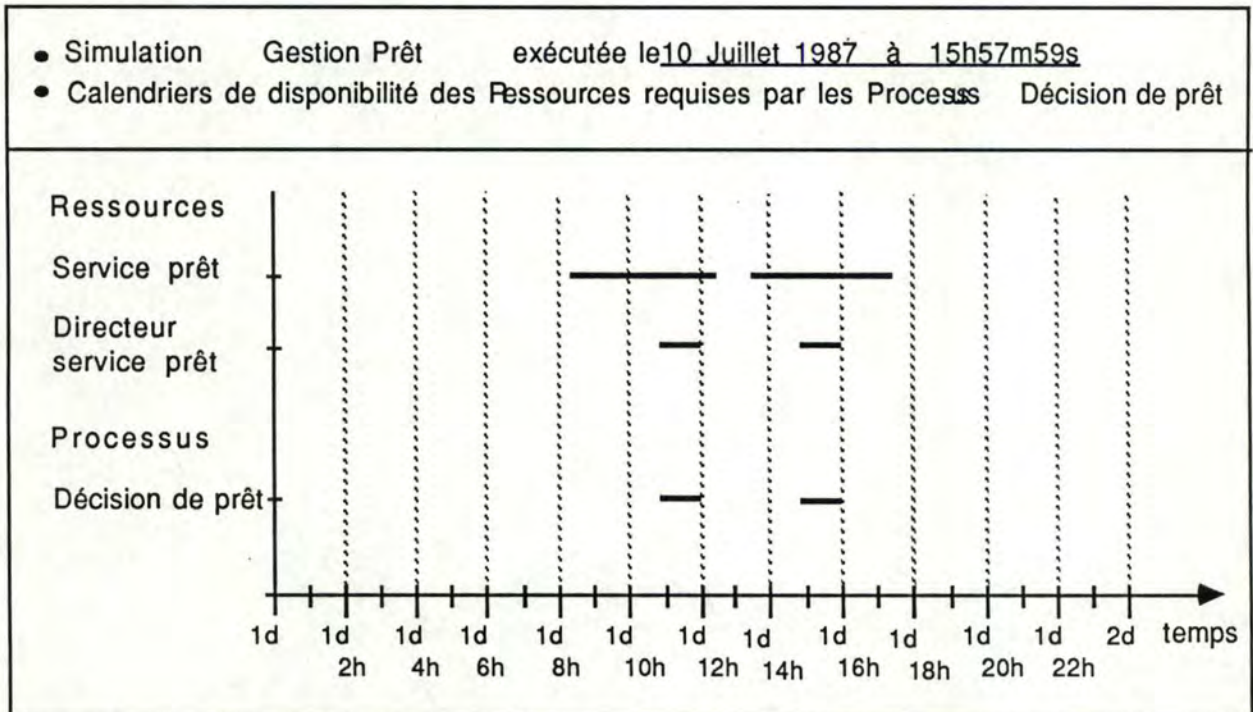


figure 1.15 calendriers

2. FONCTIONNALITES

2.1. Identification d'un type de processus

Pour afficher les calendriers des ressources requises par un type de processus, il faut que celui-ci ait été sélectionné préalablement.

2.2. Activation d'autres outils

Cf. fonctionnalité 2.2. mesures globales.

2.3. Gestion de fenêtre

Cf. fonctionnalité 2.4. mesures chronologiques ou périodiques.

2.4. Zooming de l'axe des ordonnées

Cette fonction permet à l'utilisateur de déterminer la période qu'il désire voir représentée dans une fenêtre.

2.5. Afficher/effacer la partie commentaire

Cf. fonctionnalité 2.5. mesures globales.

2.6. Impression

Cf. fonctionnalité 2.7. mesures chronologiques ou périodiques.

CONCLUSION

Les informations présentées par les outils qui ont été proposés dans ce chapitre sont reprises ci-dessous.

OUTILS	MESURES PRESENTEES	FORMES DE REPRESENTATION	TYPE DES MESURES
[1]	durée moyenne d'attente de repos d'activité nombre moyen de processus en attente en activité nombre de déclenchements d'activations d'interruptions de terminaisons	camembert camembert texte	globales
[2]	capacité moyenne nombre de points d'entrée utilisés file d'attente moyenne	camembert camembert texte	globales
[3]	nombre de réalisations d'contributions temps moyen de réalisation de participation événements contraignants en fréquence en durée	texte texte camembert camembert	globales
[4]	nombre de déclenchements d'activations d'interruptions de terminaisons	diagramme à bâtons	chronologiques
[5]	idem	diagramme à bâtons	périodiques
[6]	durée moyenne d'attente	diagramme à bâtons	chronologiques
[7]	durée moyenne d'attente de repos	diagramme à bâtons	périodiques
[8]	nombre moyen de processus en attente en activité	diagramme à bâtons	chronologiques
[9]	capacité moyenne nombre de points d'entrée utilisés	diagramme à bâtons	chronologiques
[10]	file d'attente moyenne	diagramme à bâtons	chronologiques
[11]	CALENDRIER	segments de droites	calendrier

- CHAPITRE II-

Description des bases de données

INTRODUCTION

L'objectif de ce chapitre est de décrire les structures d'informations mémorisant les données manipulées par les outils décrits au chapitre précédent.

Pour des raisons de clarté, nous n'avons pas donné le schéma conceptuel de l'ensemble de la base, nous avons préféré effectuer cette présentation en décrivant quatre sous-schémas qui correspondent :

- à des éléments de spécification DSL utiles à l'interprétation
- aux statistiques globales
- aux statistiques périodiques
- aux statistiques chronologiques

Par conséquent, il ne faudra donc pas s'étonner de la redondance des schémas les uns par rapport aux autres.

II.1. SPECIFICATIONS

II.1.1. Définition des entités

calendrier

définition

On appelle "calendrier" un calendrier de disponibilité d'une ressource "ressource"

attributs

nom-calend	nom du calendrier
périodicité	périodicité du calendrier (semaine ou mois)
intervalle-jour	suite consécutive de jours Cet attribut se répète pour chaque intervalle-jour de la période de répétitivité et il est <u>composé</u> des attributs suivants :
jour-début	délai qui sépare le début de l'intervalle-jour du début de la période de répétitivité
jour-fin	délai qui sépare la fin de l'intervalle-jour du début de la période de répétitivité
intervalle-hr	plage de temps continue à l'intérieur d'une journée faisant partie de l'intervalle jour Cet attribut se répète pour chaque plage de temps spécifiée pour l'intervalle-jour et il est <u>composé</u> des attributs suivants :
hr-début	délai qui sépare le début de l'intervalle-heure du début de la journée
hr-fin	délai qui sépare la fin de l'intervalle-heure du début de la journée

identifiée par

nom-calend

contraintes

1. $0 \leq \text{"jour-début"} \leq \text{"jour-fin"}$
2. Si la périodicité est la semaine, alors $\text{"jour-fin"} \leq 7d$
3. Si la périodicité est le mois, alors $\text{"jour-fin"} \leq 31d$
4. $0 \leq \text{"hr-début"} \leq \text{"hr-fin"} \leq 24h$

expression

définition

On appelle "expression", un opérateur complété de son opérande s'il s'agit d'un opérateur unaire ou de ses deux opérandes s'il s'agit d'un opérateur binaire faisant partie d'une condition de synchronisation.

attribut

opérateur	nom de l'opérateur de l'expression. Il existe deux types d'opérateurs, à savoir : les opérateurs unaires (COMPTER, REGROUPER) et les opérateurs binaires (ET, OU, MEME-ORIGINE)
-----------	---

identifiée par

son attribut "opérateur" et son (ses) rôle(s) "a-pour-opérande" dans l'association "op-de-l'expr".

contraintes

1. Une expression est au moins la condition d'un point de synchronisation ou l'expression d'une opérande d'une autre expression
2. Une expression participe toujours à une relation "opérande-de-l'expr" dont l'attribut "ordre" indique la première opérande de l'opérateur
3. Si l'opérateur est binaire, alors l'expression participe à une relation "opérande-de-l'expr" dont l'attribut "ordre" indique la deuxième opérande de l'opérateur.

opérande

définition

On appelle "opérande" une opérande d'une "expression". Rappelons qu'une opérande est soit la référence d'une contribution à un point de synchronisation, soit une expression.

attributs

Ces deux attributs n'existent que si l'opérande est la référence d'une contribution à un point de synchronisation

t-évé-contrib	type d'événement contribuant au point de synchronisation
mode-contrib	mode de contribution (simple, multiple)

identifiée par

ses attributs "t-évé-contrib", "mode-contrib" si ceux-ci existent ou par son rôle "a-pour-expr" dans l'association "expr-de-l'op".

point-synchro

définition

On appelle "point-synchro" tout point de synchronisation appartenant à la spécification dynamique du système simulé.

attribut

nom-synchro nom du point de synchronisation

identifiée par

nom-synchro

ressource

définition

On appelle "ressource" toute ressource requise directement ou indirectement par un ou plusieurs types de processus "type-processus"

attributs

nom-ressource nom de la ressource

capacité-th capacité maximum théorique de la ressource

nom-ressource nombre maximum théorique de requêtes pouvant être satisfaites simultanément par la ressource

identifiée par

nom-ressource

simulation

définition

On appelle "simulation" une exécution du programme "SIM"

attributs

titre-sim titre de la simulation

date-sim date d'exécution de SIM

type-processus

définition

On appelle "type-processus" tout type de processus appartenant à la spécification du système d'information simulé

attribut

nom-processus nom du type de processus

identifiée par

nom-processus

II.1.2. Définition des associations

condition-synchro

définition

"condition-synchro" représente la condition d'un point de synchronisation exprimée sous la forme d'une expression.

relie

point-synchro, expression

rôles

"a_pour_condition" pour "point-synchro"

"est_condition_de" pour "expression"

connectivités

1-1 pour "point-synchro" (car la condition d'un point de synchronisation est spécifiée par une seule expression)

o-n pour "expression" (car "expression" peut être une opérande d'une autre expression et deux points de synchronisation différents peuvent être définis avec la même condition)

expr-de-l'op

définition

"expr-de-l'op" représente l'expression d'une opérande

relie

expression, opérande

rôles

"est_l'expr_de_l'op" pour "expression"

"a_pour_expr" pour "opérande"

connectivités

o-1 pour "expression" (car une condition de synchronisation n'est généralement pas l'opérande d'une autre expression et si une expression n'est pas une condition de synchronisation, elle est la définition d'une (seule) opérande)

o-1 pour "opérande" (car une opérande peut être la référence d'une contribution à un point de synchronisation ou alors une (seule) expression)

op-de-l'expr

définition

"op-de-l'expr" représente une opérande d'une expression

attribut

ordre indique si "opérande" est la première ou la deuxième opérande de l'"expression"

relie

expression, opérande

rôles

"a_pour_op" pour "expression"

"est_l'op_de" pour "opérande"

connectivités

1-2 pour "expression" (car l'opérateur d'une expression peut être unaire ou binaire)

1-n pour "opérande" (car, de par sa définition, "opérande" est l'opérande d'au moins une expression et si un même type d'événement contribue à plusieurs points de synchronisation, alors la même référence de contribution apparaît comme opérande dans la condition de chacun de ces points de synchronisation)

réquisition

définition

"réquisition" représente la réquisition directe ou indirecte d'une ressource par un type de processus durant le calendrier "calendrier"

relie

type-processus, ressource, calendrier

rôles

"requiert" pour "type-processus"

"est_requis_par" pour "ressource"

"est_la_période_de_réquisition_du" pour "calendrier"

connectivités

0-n pour "type-processus" (car on peut définir en DSL un processus qui ne requiert aucune ressource et un autre qui requiert plusieurs ressources)

1-n pour "ressource" (car, d'une part, d'après la définition d'une ressource, toutes les ressources sont requises par au moins un type de processus et d'autre part, à chaque réquisition d'une ressource, par un type de processus est associé un calendrier de disponibilité)

1-n pour "calendrier" (car, de par sa définition, un calendrier est le calendrier de disponibilité d'au moins une ressource et on peut spécifier en DSL deux réquisitions différentes pour un même calendrier)

contrainte

Si le calendrier de disponibilité n'est pas spécifié explicitement en DSL, il l'est implicitement de la manière suivante :

```
DEF CALENDRIER nom-calend;
  ACTIF 24h-sur-24 DURANT 7d-sur7;

DEF INTERVALLE-JOUR 24h-sur-24;
  DURE DU 0 AU 7 JOUR PAR SEMAINE;

DEF INTERVALLE-HEURE 7d-sur7;
  DURE DE '0h' A '24h';
```

sim-processus

définition

"sim-processus" représente un type de processus sur lequel porte une simulation

relie

simulation, type-processus

rôles

"simule_le_modèle_contenant_pcs" pour "simulation"

"pcs_repris_dans_le_modèle_simulé_par" pour "type-processus"

connectivités

- o - n pour "simulation" (car le modèle simulé par la simulation peut ne pas contenir de type de processus et il peut en contenir plusieurs)
- 1 - 1 pour "type-processus" (car, de par sa définition, "type-processus" appartient à la spécification du système simulé et la base de données contient une seule occurrence de "simulation")

sim-synchro

définition

"sim-processus" représente un point de synchronisation sur lequel porte une simulation

relie

simulation, point-synchro

rôles

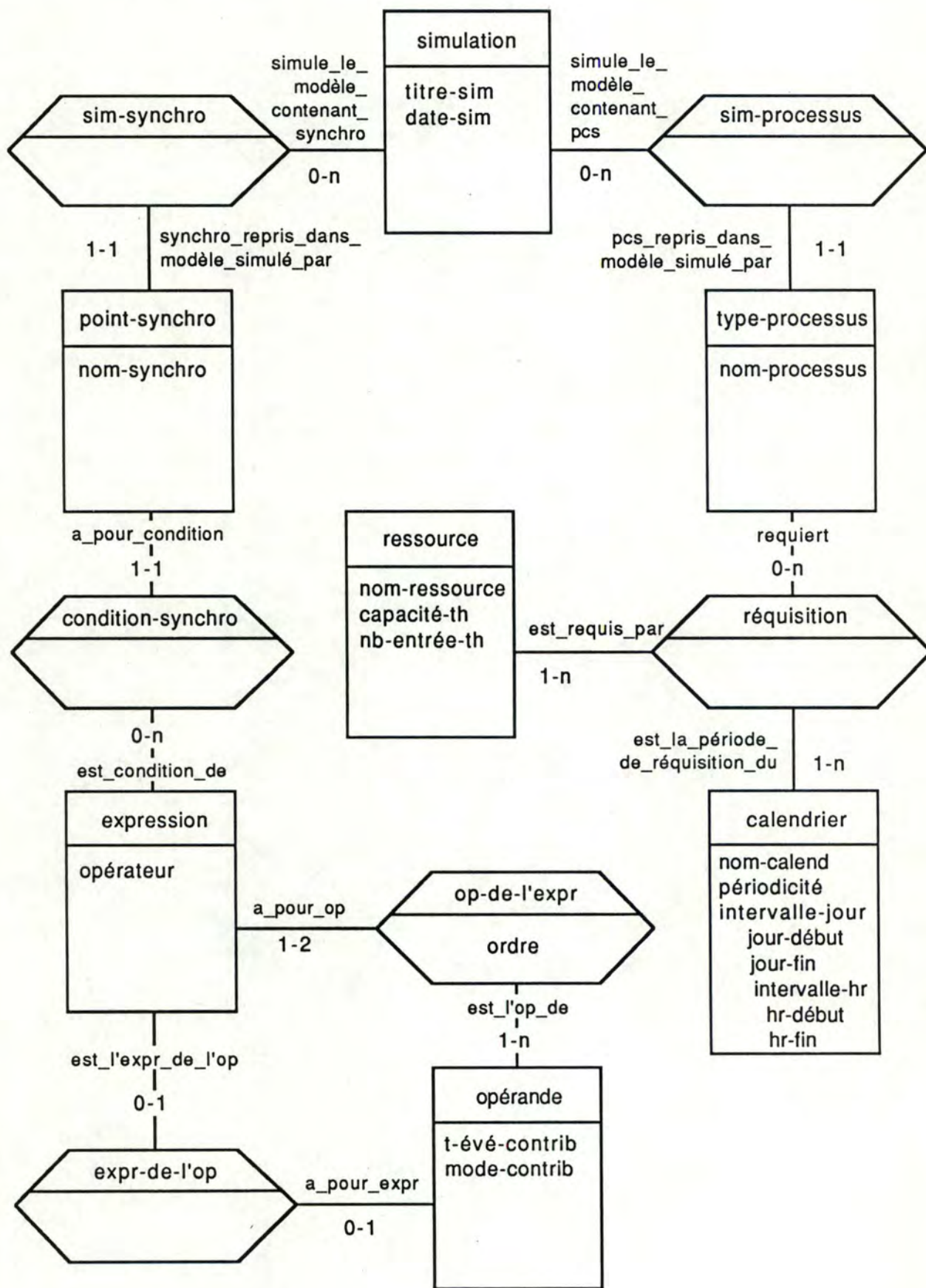
"simule_le_modèle_contenant_synchro" pour "simulation"

"synchro_repris_dans_le_modèle_simulé_par" pour "point-synchro"

connectivités

- o - n pour "simulation" (car le modèle simulé par la simulation peut ne pas contenir de point de synchronisation et il peut en contenir plusieurs)
- 1 - 1 pour "point-synchro" (car, de par sa définition, "point-synchro" appartient à la spécification du système simulé et la base de données contient une seule occurrence de "simulation")

II.1.3. Schéma de la base de données



II.2. STATISTIQUES GLOBALES

II.2.1. Définition des entités

point-synchro

Cf. entité "point-synchro" de la base de données de spécifications (II.1.1.).

ressource

Cf. entité "ressource" de la base de données de spécifications (II.1.1.).

simulation

Cf. entité "simulation" de la base de données de spécifications (II.1.1.).

stat-globales

définition

On appelle "stat-globales", l'ensemble des statistiques globales calculées par un programme BGST (1).

attributs

période-obs	plage de temps pendant laquelle les mesures de la simulation sont prises en compte pour le calcul de statistiques globales. Cet attribut est <u>composé</u> des attributs suivants :
date-début	début de la période d'observation
date-fin	fin de la période d'observation

contraintes

Une occurrence de "stat-globales" participe à au moins une des associations suivantes : "glob-synchro", "glob-ressource", "glob-processus".

type-processus

Cf. entité "type-processus" de la base de données de spécifications (II.1.1.).

II.2.2. Définition des associations

calcul-glob

définition

"calcul-glob" représente le calcul de statistiques globales à partir des résultats de la simulation.

relie

simulation, stat-globales

(1) Puisque une base de données de statistiques globales correspond à une seule exécution du programme BGST, elle ne contient qu'une seule occurrence de "stat-globales".

rôles

"permet_le_calcul_de" pour "simulation"

"calculées_à_partir_de" pour "stat-globales"

connectivités

1 - 1 pour "simulation" (car la base de données correspond à l'exécution d'un seul programme BGST qui produit un seul ensemble de statistiques globales à partir d'une simulation)

1 - 1 pour "stat-globales" (car la base de données correspond à l'exécution d'un seul programme BGST qui produit un ensemble de statistiques globales à partir d'une seule simulation)

glob-processus

définition

"glob-processus" représente le sous-ensemble de "stat-globales" contenant les statistiques globales qui concernent un même type de processus "type-processus". Cet ensemble est vide si aucun processus du type considéré n'a été déclenché durant la simulation.

relie

stat-globales, type-processus

rôles

"contient_stat_pcs" pour "stat-globales"

"pcs_a_pour_stat" pour "type-processus"

connectivités

0 - n pour "stat-globales" (car le programme BGST calcule les statistiques pour tous les types de processus appartenant à la spécification du système simulé. Or, ce modèle peut ne pas contenir de type de processus, et, en général, il en contient plusieurs)

1 - 1 pour "type-processus" (car le programme BGST calcule les statistiques pour tous les types de processus appartenant à la spécification du système simulé et la base de données contient une seule occurrence de "stat-globales")

attributs

Les attributs de "glob-processus" sont facultatifs, ils n'existent pas si aucun processus du type considéré n'a été déclenché durant la simulation.

nb-déclench	nombre de déclenchements de processus du type donné survenus pendant la période d'observation de "stat-globales"
nb-acti	nombre d'activations de processus du type donné survenues pendant la période d'observation de "stat-globales"
nb-inter	nombre d'interruptions de processus du type donné survenues pendant la période d'observation de "stat-globales"
nb-termin	nombre de terminaisons de processus du type donné survenues pendant la période d'observation de "stat-globales"
tps-acti-moy	durée moyenne d'activité des processus du type donné qui se sont déclenchés au cours de la période d'observation de "stat-globales". Cette statistique existe uniquement si au moins un processus du type considéré a été déclenché durant la période d'observation.

tps-att-moy	durée moyenne d'attente des processus du type donné qui se sont (ré)activés au cours de la période d'observation de "stat-globales". Cette statistique existe uniquement si au moins un processus du type considéré a été (ré)activé durant la période d'observation.
tps-repos-moy	durée moyenne de repos pour les processus du type donné qui se sont (ré)activés au cours de la période d'observation de "stat-globales". Cette statistique existe uniquement si au moins un processus du type considéré a été (ré)activé durant la période d'observation.
nb-att-moy	nombre moyen de processus du type donné en attente durant la période d'observation de "stat-globales".
nb-acti-moy	nombre moyen de processus du type donné en activité durant la période d'observation de "stat-globales"

glob-ressource

définition

"glob-ressource" représente le sous-ensemble de "stat-globales" contenant les statistiques globales qui concernent une même ressource "ressource".

relie

stat-globales, ressource

rôles

"contient_stat_res" pour "stat-globales"

"res_a_pour_stat" pour "ressource"

connectivités

0-n pour "stat-globales" (car le programme BGST calcule les statistiques pour toutes les ressources appartenant au système simulé. Or, ce modèle peut ne pas contenir de ressource et, en général, il en contient plusieurs)

1-1 pour "ressource" (car le programme BGST calcule les statistiques pour toutes les ressources appartenant au système simulé et la base de données contient une seule occurrence de "stat-globales")

attributs

cap-util-moy capacité de la ressource utilisée en moyenne durant toute la période d'observation de "stat-globales"

nb-entrée-moy nombre moyen de points d'entrée utilisés durant toute la période d'observation de "stat-globales"

file-att-moy nombre moyen de processus en attente devant la ressource durant toute la période d'observation de "stat-globales"

glob-synchro

définition

"glob-synchro" représente le sous-ensemble de "stat-globales" contenant les statistiques globales qui concernent un même point de synchronisation "point-synchro". Cet ensemble est vide si aucune réalisation du point de synchronisation ne survient durant la simulation.

relie

stat-globales, point-synchro

rôles

"contient_stat_synchro" pour "stat-globales"

"synchro_a_pour_stat" pour "point-synchro"

connectivités

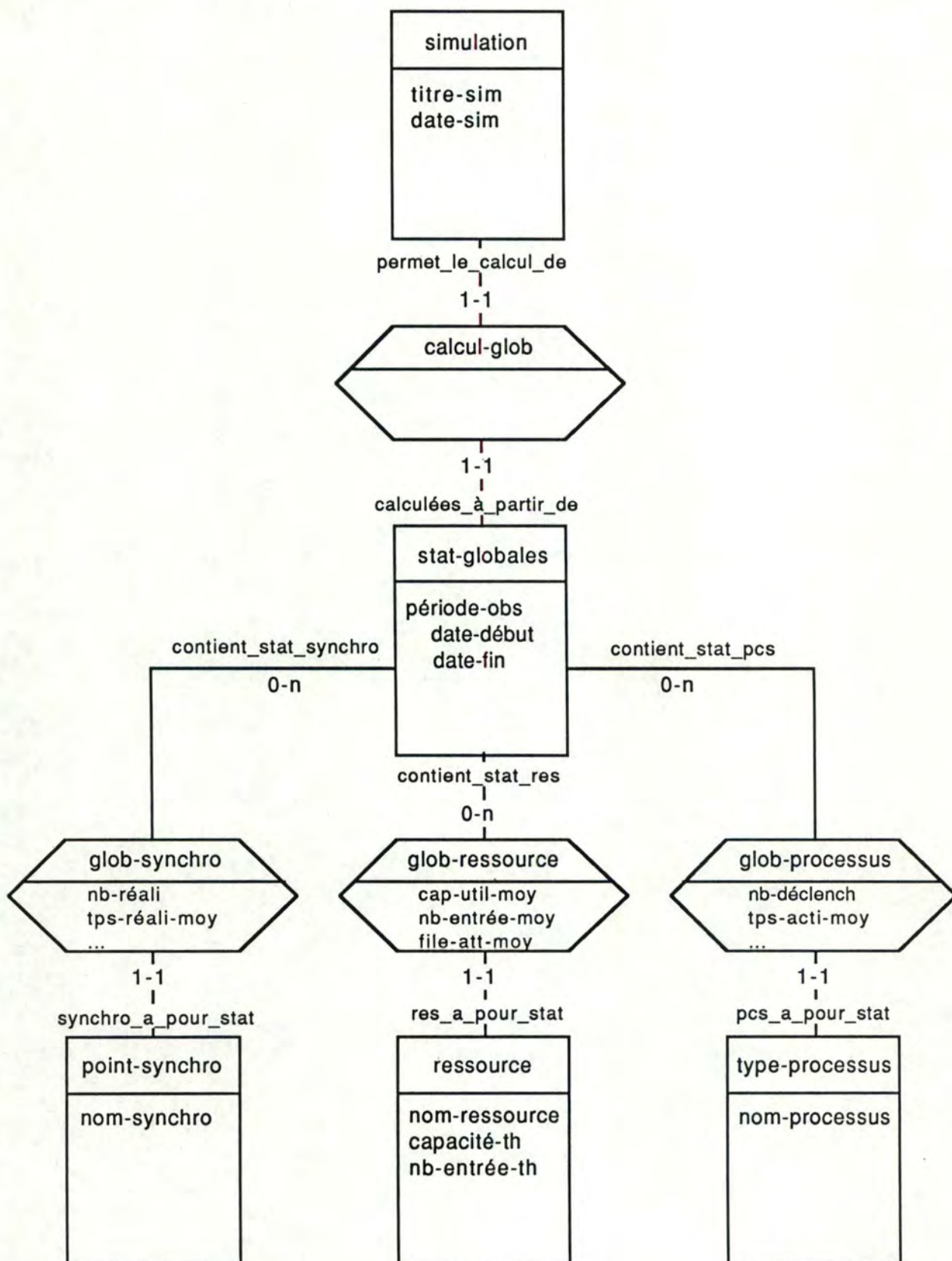
o - n pour "stat-globales" (car le programme BGST calcule les statistiques pour tous les points de synchronisation appartenant à la spécification du système simulé. Or, ce modèle peut ne pas contenir de point de synchronisation et il peut en contenir plusieurs)

1 - 1 pour "point-synchro" (car le programme BGST calcule les statistiques pour tous les points de synchronisation appartenant à la spécification du système simulé et la base de données contient une seule occurrence de "stat-globales")

attributs

nb-réali	nombre de réalisations du point de synchronisation survenues durant la période d'observation de "stat-globales"
tps-réali-moy	temps moyen des réalisations du point de synchronisation qui ont eu lieu dans la période d'observation de "stat-globales"
stat-évé	ensemble des statistiques globales concernant un type d'événement contribuant au point de synchronisation. Cet attribut se répète pour chaque type d'événement contribuant au point de synchronisation, et est <u>composé</u> des attributs suivants :
nom-typ-évé	nom du type d'événement contribuant au point de synchronisation
nb-contrib	nombre de contributions de ce type survenues entre le début de la période d'observation de "stat-globales" et la dernière réalisation du point de synchronisation.
tps-particip-moy	temps moyen de participation des événements du type considéré ayant participé à la réalisation d'une synchronisation.
contraign-en-durée	pourcentage de contraignance en durée du type d'événement pour le point de synchronisation
contraign-en-fréq	pourcentage de contraignance en fréquence du type d'événement pour le point de synchronisation

II.2.3. Schéma de la base de données



II.3. STATISTIQUES PERIODIQUES

II.3.1. Définition des entités

cellule-répét

définition

On appelle "cellule-répét" toute plage de temps à l'intérieur d'une période de répétitivité.

attributs

borne-inf	délai qui sépare le début de la plage de temps du début de la période de répétitivité
borne-sup	délai qui sépare la fin de la plage de temps du début de la période de répétitivité

identifiée par

ses attributs "borne-inf" et "borne-sup"

contraintes

1. $0 \leq \text{"borne-inf"} \leq \text{"borne-sup"}$.
2. si la période de répétitivité est :
 - le mois, alors "borne-sup" \leq 28d
 - la semaine, alors "borne-sup" \leq 7d
 - le jour, alors "borne-sup" \leq 24h
 - l'heure, alors "borne-sup" \leq 60m
 - la minute, alors "borne-sup" \leq 60s

simulation

Cf. entité "simulation" de la base de données de spécifications (II.1.1.).

stat-péριο

définition

On appelle "stat-péριο", l'ensemble des statistiques périodiques calculées par un programme BGST (1).

attributs

période-obs	plage de temps pendant laquelle les mesures de la simulation sont prises en compte pour le calcul de statistiques périodiques. Cet attribut est <u>composé</u> des attributs suivants :
date-début	début de la période d'observation
date-fin	fin de la période d'observation
période-répét	période de répétitivité sur laquelle les statistiques périodiques sont calculées (mois, semaine, jour, heure, minute)

(1) Puisque une base de données de statistiques périodiques correspond à une seule exécution du programme BGST, elle ne contient qu'une seule occurrence de "stat-péριο"

contrainte

"période-répét" est découpée en cellules "cellule-répét" qui sont disjointes et qui couvrent toute la période de répétitivité.

type-processus

Cf. entité "type-processus" de la base de données de spécifications (II.1.1.).

II.3.2. Définition des associations

calcul-péριο

définition

"calcul-péριο" représente le calcul de statistiques périodiques effectué à partir des mesures prises par la simulation.

relie

simulation, stat-péριο

rôles

"permet_le_calcul_de" pour "simulation"

"calculées_à_partir_de" pour "stat-péριο"

connectivités

- 1 - 1 pour "simulation" (car la base de données correspond à l'exécution d'un seul programme BGST qui produit un seul ensemble de statistiques périodiques à partir d'une simulation)
- 1 - 1 pour "stat-péριο" (car la base de données correspond à l'exécution d'un seul programme BGST qui produit un ensemble de statistiques périodiques à partir d'une seule simulation)

découpe-répét

définition

"découpe-répét" représente une plage de temps "cellule-répét" à l'intérieur de la période de répétitivité.

relie

cellule-répét, stat-péριο

rôles

"est_une_découpe_pour" pour "cellule-répét"

"période_répét_découpée_en" pour "stat-péριο"

connectivités

- 1 - 1 pour "cellule-répét" (car, de par sa définition, "cellule-répét" est la découpe d'au moins une période de répétitivité et la base de données contient une seule occurrence de "stat-péριο")
- 1 - 28 pour "stat-péριο" (car l'utilisateur peut spécifier de 1 à 28 cellules à l'intérieur de la période de répétitivité pour le calcul des statistiques périodiques (cf. [DSL-SIM, 86]))

périod-processus

définition

"périod-processus" représente le sous-ensemble de "stat-pério" contenant les statistiques périodiques qui concernent un type de processus "type-processus" pour une cellule de répétitivité "cellule-répét". Cet ensemble est vide si aucun processus du type considéré n'a été déclenché durant la simulation.

relie

stat-pério, type-processus, cellule-répét

rôles

"contient_stat_pcs" pour "stat-pério"

"pcs_a_pour_stat" pour "type-processus"

"cellule_pour_stat_pcs" pour "cellule-répét"

connectivités

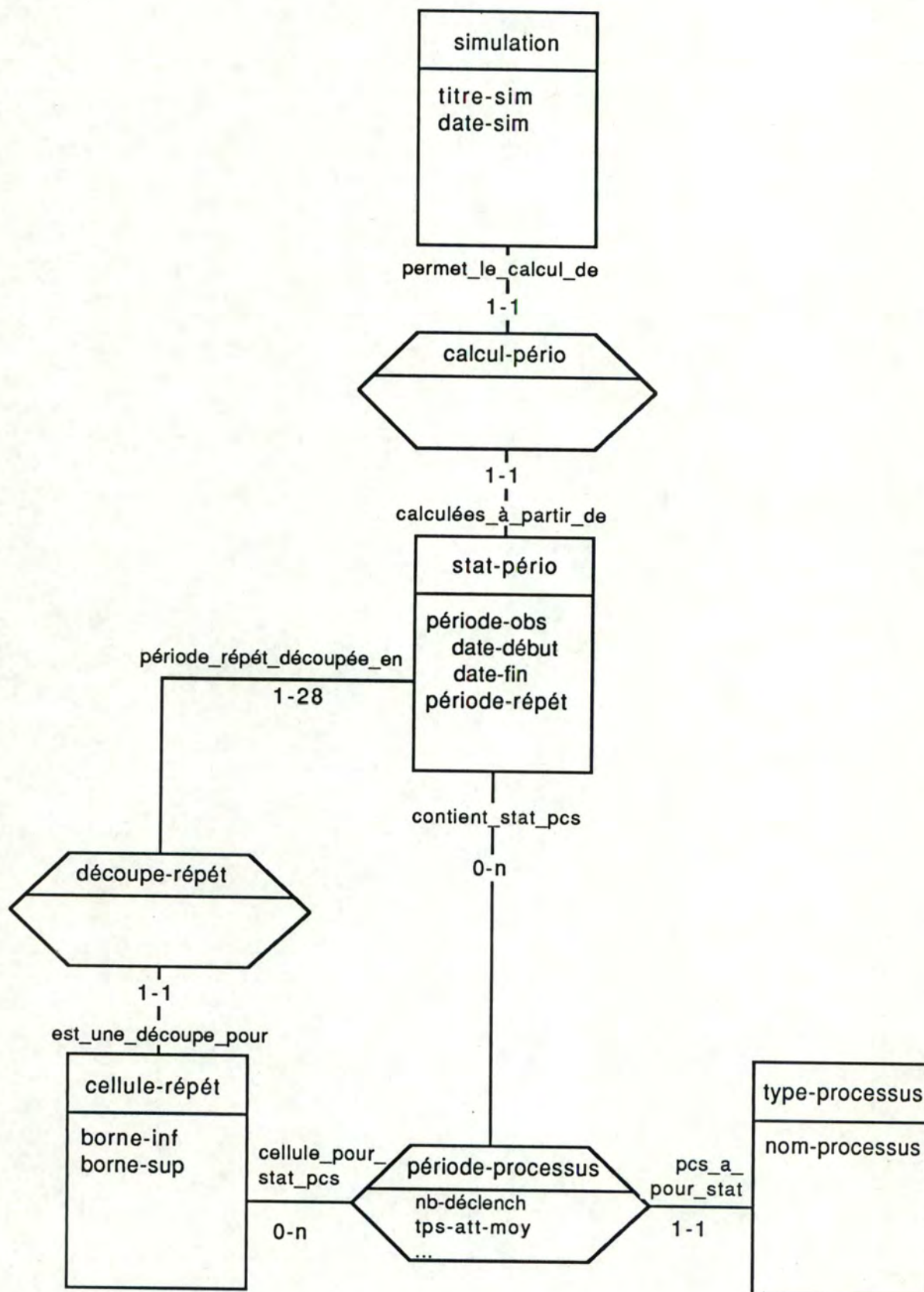
- o-n pour "stat-pério" (car le programme BGST calcule les statistiques pour tous les types de processus appartenant à la spécification du système simulé. Or, ce modèle peut ne pas contenir de type de processus et, en général, il en contient plusieurs)
- 1-1 pour "type-processus" (car le programme BGST calcule les statistiques pour tous les types de processus appartenant à la spécification du système simulé et la base de données contient une seule occurrence de "stat-pério")
- o-n pour "cellule-répét" (car si la cellule de répétitivité est hors de la période d'observation, aucune statistique périodique n'est produite pour cette cellule et normalement, BGST calcule dans une cellule un ensemble de statistiques périodiques pour chaque "type-processus")

attributs

Les attributs de "périod-processus" sont facultatifs, ils n'existent pas si aucun processus du type considéré n'a été déclenché durant la simulation.

nb-déclench	nombre de déclenchements de processus du type donné survenus dans cellule de répétitivité considérée.
nb-acti	nombre d'activations de processus du type donné survenues dans la cellule de répétitivité considérée.
nb-inter	nombre d'interruptions de processus du type donné survenues dans la cellule de répétitivité considérée.
nb-termin	nombre de terminaisons de processus du type donné survenues dans la cellule de répétitivité considérée.
tps-att-moy	durée moyenne d'attente des processus du type donné qui se sont (ré)activés au cours de la cellule de répétitivité retenue. Cette statistique existe uniquement si au moins un processus du type considéré a été (ré)activé dans la cellule de répétitivité considérée.
tps-repos-moy	durée moyenne de repos des processus du type donné qui se sont (ré)activés au cours de la cellule de répétitivité retenue. Cette statistique existe uniquement si au moins un processus du type considéré a été (ré)activé dans la cellule de répétitivité considérée.

II.3.3. Schéma de la base de données



II.4. STATISTIQUES CHRONOLOGIQUES

II.4.1. Définition des entités

cellule-obs

définition

On appelle "cellule-obs" toute plage de temps à l'intérieur d'une période d'observation.

attributs

borne-inf	délai qui sépare le début de la plage de temps du début de la période d'observation
borne-sup	délai qui sépare la fin de la plage de temps du début de la période d'observation

identifiée par

ses attributs "borne-inf" et "borne-sup"

contrainte

$0 \leq \text{"borne-inf"} \leq \text{"borne-sup"} \leq \text{fin de la période d'observation.}$

ressource

Cf. entité "ressource" de la base de données de spécifications (II.1.1.).

simulation

Cf. entité "simulation" de la base de données de spécifications (II.1.1.).

stat-chrono.

définition

On appelle "stat-chrono.", l'ensemble des statistiques chronologiques calculées par un programme BCST (1) .

attributs

période-obs	plage de temps pendant laquelle les mesures de la simulation sont prises en compte pour le calcul de statistiques chronologiques. Cet attribut est <u>composé</u> des attributs suivants :
date-début	début de la période d'observation
date-fin	fin de la période d'observation

(1) Puisque une base de données de statistiques chronologiques correspond à une seule exécution du programme BCST, elle ne contient qu'une seule occurrence de "stat-chrono".

période-répét	période de répétitivité sur laquelle les statistiques chronologiques sont calculées (mois, semaine, jour, heure, minute)
cellule-répét	plage de temps à l'intérieur de la période de répétitivité. Cet attribut se répète pour chaque cellule appartenant à la période de répétitivité et est <u>composé</u> des attributs suivants :
borne-inf	délai qui sépare le début de la cellule du début de la période de répétitivité
borne-sup	délai qui sépare la fin de la cellule du début de la période de répétitivité

contraintes

1. les "cellule-répét" sont disjointes et leur union couvre "période-répét".
2. $0 \leq \text{"borne-inf"} \leq \text{"borne-sup"}$.
3. si la période de répétitivité est :
 - le mois, alors "borne-sup" $\leq 28d$
 - la semaine, alors "borne-sup" $\leq 7d$
 - le jour, alors "borne-sup" $\leq 24h$
 - l'heure, alors "borne-sup" $\leq 60m$
 - la minute, alors "borne-sup" $\leq 60s$

type-processus

Cf. entité "type-processus" de la base de données de spécifications (II.1.1.).

II.4.2. Définition des associations

calcul-chrono

définition

"calcul-chrono" représente le calcul de statistiques chronologiques effectué à partir des mesures prises par la simulation.

relie

simulation, stat-chrono.

rôles

"permet_le_calcul_de" pour "simulation"

"calculées_à_partir_de" pour "stat-chrono."

connectivités

- 1 - 1 pour "simulation" (car la base de données correspond à l'exécution d'un seul programme BCST qui produit un seul ensemble de statistiques chronologiques à partir d'une simulation)
- 1 - 1 pour "stat-chrono." (car la base de données correspond à l'exécution d'un seul programme BCST qui produit un ensemble de statistiques chronologiques à partir d'une seule simulation)

découpe-obs

définition

"découpe-obs" représente une plage de temps "cellule-obs" à l'intérieur de la période d'observation

relie

cellule-obs, stat-chrono.

rôles

"est_une_découpe_pour" pour "cellule-obs"

"période_obs_découpée_en" pour "stat-chrono."

connectivités

- 1 - 1 pour "cellule-obs" (car, de par sa définition, "cellule-obs" fait partie d'au moins une période d'observation et la base de données contient une seule occurrence de "stat-chrono.")
- o - n pour "stat-chrono." (car, si la période d'observation est plus courte que la première cellule de répétitivité, elle ne contient aucune cellule d'observation et en général, elle est découpée en plus d'une cellule d'observation)

chrono-processus

définition

"chrono-processus" représente le sous-ensemble de "stat-chrono." contenant les statistiques chronologiques qui concernent un type de processus "type-processus" pour une cellule d'observation "cellule-obs". Cet ensemble est vide si aucune occurrence de ce type de processus n'est déclenchée durant la simulation.

relie

stat-chrono., type-processus, cellule-obs

rôles

"contient_stat_pcs" pour "stat-chrono."

"pcs_a_pour_stat" pour "type-processus"

"cellule_pour_stat_pcs" pour "cellule-obs"

connectivités

- o - n pour "stat-chrono." (car le programme BCST calcule les statistiques chronologiques pour les types de processus demandés par l'utilisateur qui peut ne pas en sélectionner ou, au contraire, en choisir plusieurs)
- o - 1 pour "type-processus" (car l'utilisateur peut demander de ne calculer aucune statistique chronologique concernant le type de processus en question et la base de données contient une seule occurrence de "stat-chrono.")
- o - n pour "cellule-obs" (car le programme BCST calcule les statistiques chronologiques pour les types de processus demandés par l'utilisateur qui peut ne pas en sélectionner ou, au contraire, en choisir plusieurs)

attributs

Les attributs de "chrono-processus" sont facultatifs, ils n'existent pas si aucun processus du type considéré n'a été déclenché durant la simulation.

nb_déclench	nombre de déclenchements de processus du type donné survenus dans la cellule d'observation considérée
nb_acti	nombre d'activations de processus du type donné survenues dans la cellule d'observation considérée
nb_inter	nombre d'interruptions de processus du type donné survenues dans la cellule d'observation considérée
nb_termin	nombre de terminaisons de processus du type donné survenus dans la cellule d'observation considérée
tps_att_moy	durée moyenne d'attente des processus du type donné qui se sont (ré)activés au cours de la cellule d'observation retenue. Cette statistique existe uniquement si au moins un processus du type considéré a été (ré)activé dans la cellule d'observation considérée
nb_att_moy	nombre moyen de processus du type donné en attente dans la cellule d'observation considérée
nb_acti_moy	nombre moyen de processus du type donné en activité dans la cellule d'observation considérée

chrono-ressource

définition

"chrono-processus" représente le sous-ensemble de "stat-chrono." contenant les statistiques chronologiques qui concernent une ressource "ressource" pour une cellule d'observation "cellule-obs".

relie

stat-chrono., ressource, cellule-obs

rôles

"contient_stat_res" pour "stat-chrono."

"res_a_pour_stat" pour "ressource"

"cellule_pour_stat_res" pour "cellule-obs"

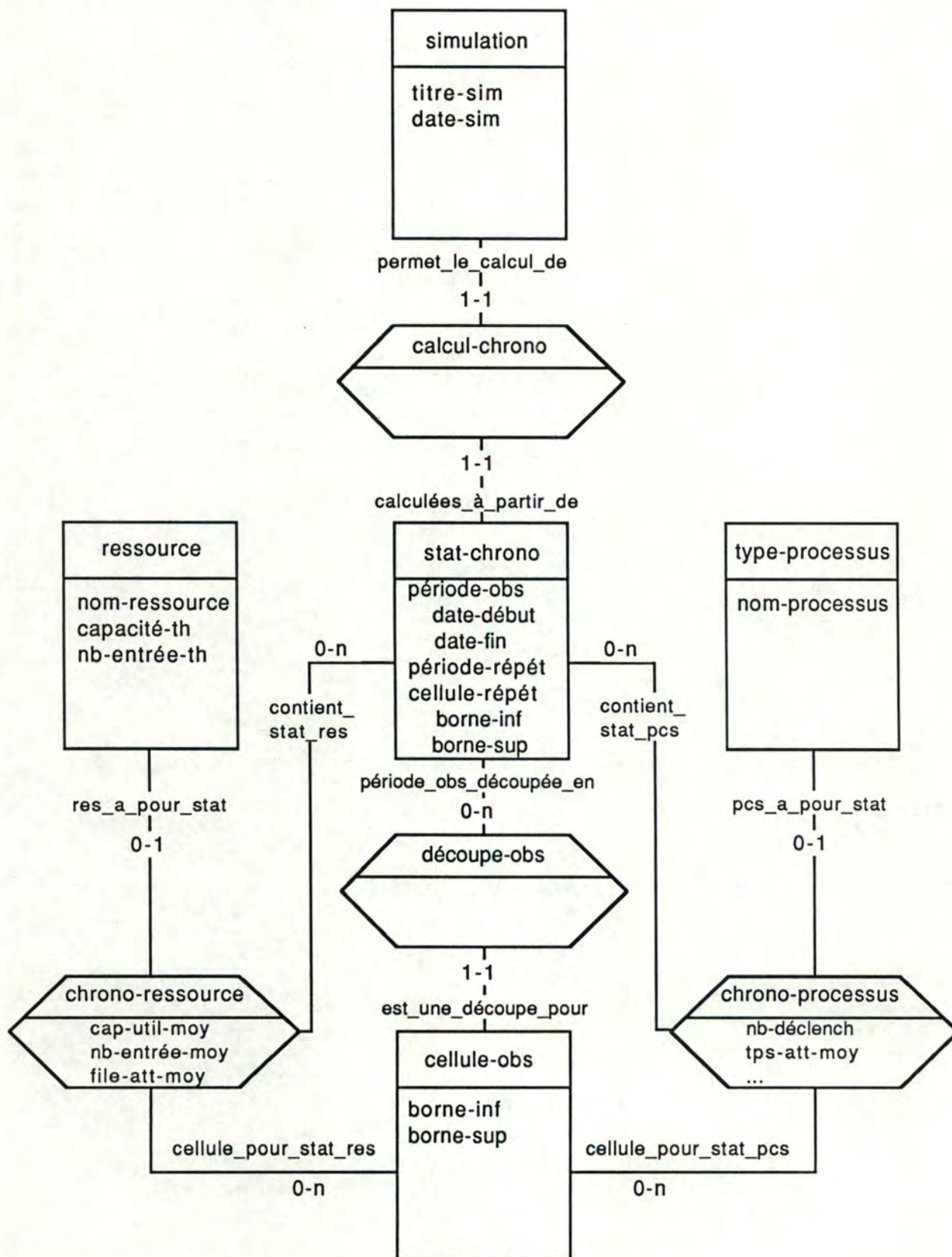
connectivités

- o-n pour "stat-chrono." (car le programme BCST calcule les statistiques chronologiques pour les ressources demandées par l'utilisateur qui peut ne pas en sélectionner ou, au contraire, en choisir plusieurs)
- o-1 pour "ressource" (car l'utilisateur peut demander de ne calculer aucune statistique chronologique concernant la ressource en question et la base de données contient une seule occurrence de "stat-chrono.")
- o-n pour "cellule-obs" (car le programme BCST calcule les statistiques chronologiques pour les types de processus demandés par l'utilisateur qui peut ne pas en sélectionner ou, au contraire, en choisir plusieurs)

attributs

- | | |
|---------------|---|
| cap_util_moy | capacité de la ressource utilisée en moyenne durant la cellule d'observation |
| nb_entrée_moy | nombre de points d'entrée de la ressource utilisés en moyenne durant la cellule d'observation |
| file_att_moy | nombre moyen de processus en attente devant la ressource dans la cellule d'observation considérée |

II.4.3. Schéma de la base de données



- PARTIE II -

Outils de représentation et d'évaluation du
comportement du système

INTRODUCTION

Dans cette deuxième partie, nous nous sommes intéressés essentiellement à des outils de représentation et d'évaluation du comportement du système ayant comme source principale d'information les mesures contenues dans le rapport produit par la simulation.

Nous avons constaté qu'actuellement DSL-SIM ne produit aucune mesure facilement exploitable sur le fonctionnement du système d'information permettant d'évaluer si une solution conceptuelle est réalisable - ou faisable - par rapport aux ressources dont dispose l'organisation.

Pour pallier à cette lacune nous avons imaginé le concept de transaction qui nous a permis de définir une mesure de la performance du système. Nous avons également associé à ce concept deux représentations statiques et une animation qui nous permettent de visualiser le bon ou le mauvais fonctionnement d'une partie du système d'information à un moment donné.

D'autres aspects du comportement du système d'information seront étudiés. Ainsi nous proposerons de recréer sous la forme d'une animation le comportement d'un mécanisme clé de la dynamique du système, en l'occurrence le point de synchronisation.

Conscients de la nécessité d'un guide dans l'évaluation de la performance d'un modèle simulé, nous proposons une démarche ayant comme support méthodologique le schéma de la dynamique.

Cette deuxième partie s'organise comme suit : le concept de transaction fera l'objet d'une définition précise dans le chapitre III. Différentes représentations graphiques du comportement du système seront envisagées au chapitre IV. Le chapitre V présente une démarche générale d'évaluation du comportement du système d'information ainsi qu'un diagramme qui montre l'évolution de la performance du système d'information à différents moments. Pour terminer, le chapitre VI décrit les bases de données qui mémorisent les informations extraites de la trace de la simulation et manipulées par les outils décrits dans les deux chapitres précédents.

- CHAPITRE III-

Concept de transaction

INTRODUCTION

Dans la partie introduction de ce chapitre, nous mettrons tout d'abord l'accent sur la nécessité du concept de transaction. Cette nouvelle notion fera ensuite l'objet d'une présentation intuitive. Après cette introduction, nous proposerons une définition précise du concept qui nous servira de base pour l'algorithme de construction d'une transaction décrit au point suivant.

III.1. INTRODUCTION

III.1.1. Problème à résoudre

L'intérêt principal de l'outil de simulation réside "...dans l'assistance qu'il est susceptible d'apporter dans la vérification de la faisabilité de la solution fonctionnelle retenue et donc dans l'évaluation des performances du futur système d'information -avec un souci d'ordre quantitatif".

"Un schéma de comportement est **faisable** si le comportement modélisé, compte tenu du niveau d'utilisation des ressources et de la charge estimée du système, garantit le respect des performances attendues de la solution retenue".

"La spécification des **performances attendues** d'un système porte essentiellement sur le délai compris entre la survenance des événements d'une classe E1 et celle des événements d'une classe E2 qui leur succèdent dans le temps" [Pigneur, 84].

Actuellement DSL-SIM ne produit aucune mesure sur le fonctionnement **général** du système d'information modélisé, permettant de vérifier aisément si les performances attendues sont effectivement respectées.

Il existe des mesures concernant la survenance d'événements dans le fichier LOG (cf Introduction) produit par la simulation, mais ces mesures sont difficilement exploitables telles que présentées à ce jour.

Par exemple, si l'utilisateur désire connaître le délai nécessaire au traitement d'une demande de prêt survenue le deuxième jour à 8h30, il est contraint de "détricoter" toute la trace, reprise partiellement à la figure III.1. afin de déterminer le moment où le prêt est accepté ou refusé.

Le parcours du fichier LOG constituant la trace de la simulation, consiste à :

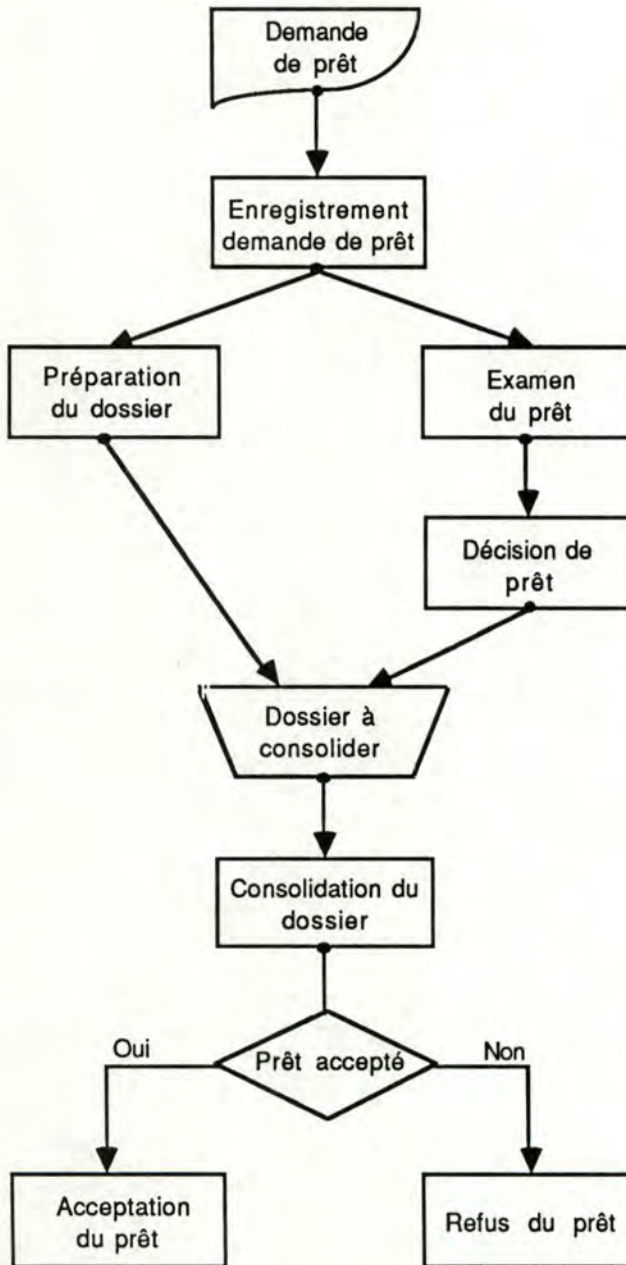
- identifier le processus "Enregistrement demande de prêt" (401) déclenché par la génération du message "Demande de prêt" (400).
- suivre les changements d'état de ce processus (401) jusqu'à sa terminaison.
- identifier le processus "Préparation du dossier" (414) déclenché par la terminaison du processus (401).

...

L'identification des processus déclenchés directement ("Enregistrement demande de prêt") ou indirectement ("Préparation du dossier") par la génération d'une "Demande de prêt" se fait sur base de la dynamique de la gestion de prêt décrite à l'aide des phrases DSL et représentée graphiquement par le schéma de la dynamique à la figure III.1.

Pour éviter cette recherche fastidieuse dans le fichier LOG, nous proposons d'extraire, à la demande de l'utilisateur, l'ensemble des événements succédant à un **événement Initial** (par exemple, la génération d'une "Demande de prêt" survenue le deuxième jour à 8h30) et auquel succèdent un ou plusieurs autres **événements terminaux** qui sont de même type (par exemple, les terminaisons des processus "Acceptation du prêt" qui succèdent à l'événement initial).

Ces événements constituent ce que nous avons appelé une **transaction** dont le **délai** (par exemple, le temps nécessaire pour accepter un prêt) constitue la mesure de performance souhaitée.



Extrait du LOG

- 1d08h30m gén demande-de-prêt #400
- 1d08h30m décl enreg-demande-prêt #401
- 1d08h30m acti enreg-demande-prêt #401
- 1d08h31m22s gén demande-de-prêt #402
- ...
- 1d08h54m47s term enreg-demande-prêt #401
- 1d08h54m47s décl préparation-dossier #414
- 1d08h54m47s acti préparation-dossier #414
- 1d08h54m47s décl examen-du-prêt #415
- 1d08h54m47s acti examen-du-prêt #415
- ...
- 1d09h07m39s term préparation-dossier #414
- 1d09h07m39s cont dossier-à-consolider
- ...
- 1d09h34m24s term examen-du-prêt #415
- 1d09h34m24s décl décision-de-prêt #451
- ...
- 1d15h03m54s réal dossier-à-consolider #728
- 1d15h03m54s acti décision-de-prêt #451
- ...
- 1d15h09m35s term décision-de-prêt #451
- 1d15h09m35s cont dossier-à-consolider
- 1d15h09m35s réal dossier-à-consolider #739
- 1d15h09m35s acti décision-de-prêt #469
- 1d15h09m35s décl consolidation-dossier#740
- 1d15h09m35s acti consolidation-dossier#740
- ...
- 1d15h34m30s term consolidation-dossier#740
- 1d15h34m30s acti enreg-demande-prêt #775
- 1d15h34m30s décl acceptation-du-prêt #797
- 1d15h34m45s term décision-de-prêt #537
- 1d15h34m45s cont dossier-à-consolider
- ...
- 1d17h10m acti acceptation-du-prêt #797
- 1d17h12m term acceptation-du-prêt #797
- 1d17h12m acti acceptation-du-prêt #806

figure III.1 Gestion de prêt

Le délai d'une transaction pourrait être obtenu sans devoir parcourir la trace de la simulation. Pour cela, il suffit d'ajouter un attribut à valeur unique au premier événement de la transaction (par exemple, la génération d'une " Demande de prêt") et de regrouper à l'aide d'un point de synchronisation, sous l'opérateur MEME ORIGINE, le (type d') événement initial et le(s) (types d') événements terminaux (par exemple, la terminaison du processus "Acceptation du prêt"). La durée d'une synchronisation correspond, dans ce cas, à la durée nécessaire pour accepter un prêt.

Compte tenu des règles de propagation des attributs, cet artifice, pour obtenir une mesure de la performance du système, n'est pas toujours possible.

Si l'on applique cette technique pour calculer la durée nécessaire à l'emballage et à l'expédition d'un colis, dans le système décrit à la figure III.2, nous devons ajouter, dans la modélisation du système, un attribut à valeur unique "ATTr-unique" à chaque déclenchement du processus "Emballage colis" et un point de synchronisation "Transaction" regroupant les types d'événements extrémaux de la transaction, comme illustré à la figure III.3.

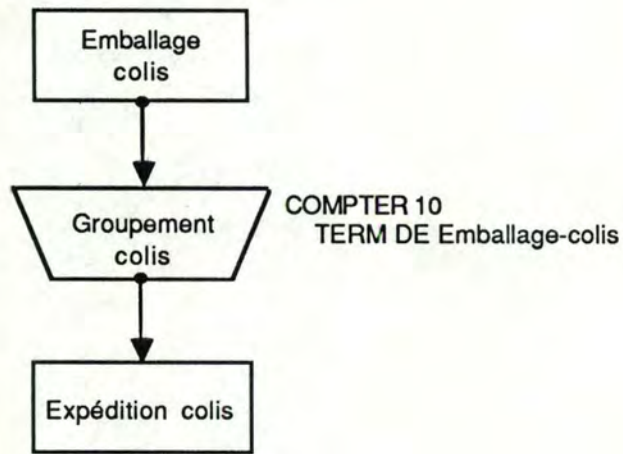


figure III.2

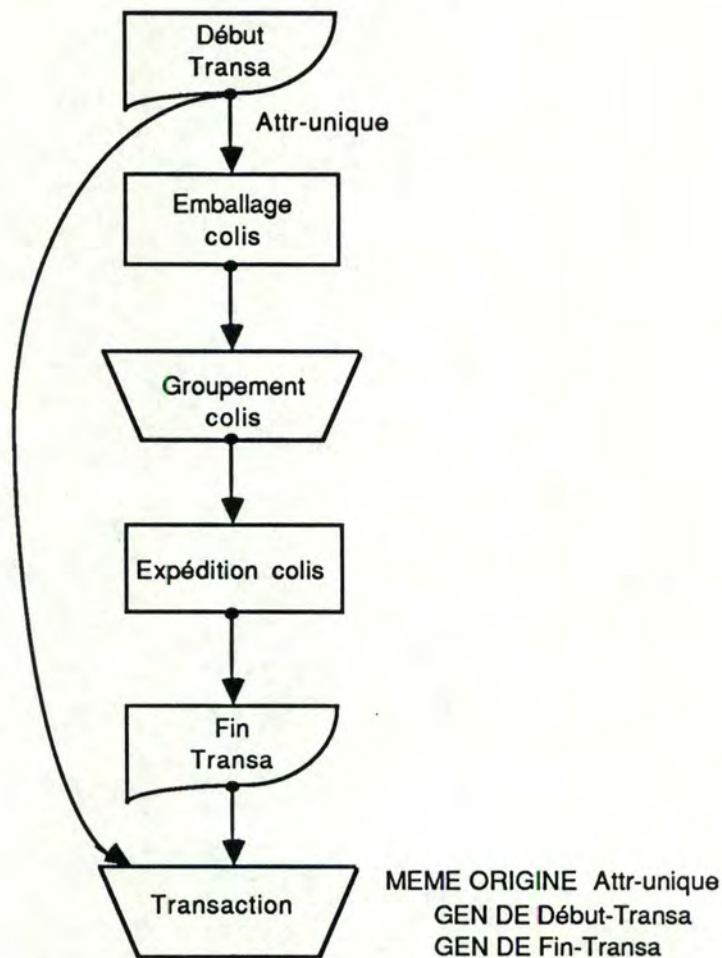


figure III.3

Comme la valeur de l'attribut "Attr-unique" transmise lors de la réalisation d'une synchronisation "Grouperment colis" est celle qui a été affectée au dernier événement contribuant à la synchronisation, la durée de réalisation d'une synchronisation "transaction" représente le délai qui sépare le déclenchement du processus "Emballage colis" qui a contribué en dernier à la synchronisation de la terminaison du processus "Expédition colis" déclenché par la réalisation en question. Ce délai ne constitue donc pas, dans ce cas, la mesure de performance souhaitée.

Ce mécanisme n'est pas général et il est difficilement praticable. Il faut par conséquent une définition du concept de transaction intrinsèque au modèle de la dynamique. Ce qui est l'objet des deux points suivants.

III.1.2. Idée intuitive

TRANSACTION

Une transaction est un ensemble d'événements qui succèdent dans le temps à un événement -initial- et auquel succèdent un ou plusieurs événements -terminaux- de même type

Par convention, une transaction est notée $TRANSA(i,T)$ où i représente l'événement initial et T l'ensemble des événements terminaux.

Un type de transaction est l'ensemble des transactions qui ont le même type d'événement initial et terminal.

Un ensemble de transactions est la classe des transactions de même type qui se sont déroulées dans une période déterminée.

A titre d'exemple, si on reprend la gestion de prêt illustrée à la figure III.1, $TRANSA$ (génération de la "Demande de prêt" survenue à 8h30m, terminaison de l'"Acceptation du prêt" survenue à 17h12m) comprend les événements marqués d'un point dans le fichier LOG représenté à la même figure.

De plus, les transactions dont l'événement initial est la génération d'une "Demande de prêt" et l'événement terminal est la terminaison de l'"Acceptation du prêt" définissent un type de transaction.

Notons qu'une transaction est définie comme un ensemble d'événements car nous voulions disposer d'un moyen de visualisation du comportement et, par-là même, d'une possibilité d'interprétation graphique du bon ou du mauvais comportement du système (cf. IV.2).

DELAI D'UNE TRANSACTION

Le délai d'une transaction est la durée séparant la survenance de l'événement initial de la survenance de l'événement terminal ou du dernier événement terminal s'il y en a plusieurs.

Comme nous allons l'illustrer par plusieurs exemples, l'objectif de cette mesure est d'estimer globalement la performance des processus déclenchés directement ou indirectement par la survenance d'un événement initial et responsable directement ou indirectement de la survenance d'un ou plusieurs événements terminaux. Par conséquent, le délai d'une transaction constitue une mesure privilégiée qui permet une évaluation de la performance d'un système.

Il est également intéressant de décomposer cette mesure en considérant le délai de chaque événement de la transaction par rapport à l'événement initial et ainsi de localiser dans le système un problème éventuel.

Pour estimer globalement la performance du système décrit à la figure III.4, on doit pouvoir quantifier la durée nécessaire pour traiter une livraison. Cette durée correspond au délai séparant la génération d'une "Livraison" et la génération du "Réassortiment" qui en découle, ce qui constitue bien le délai d'une transaction identifiée par ces deux événements.

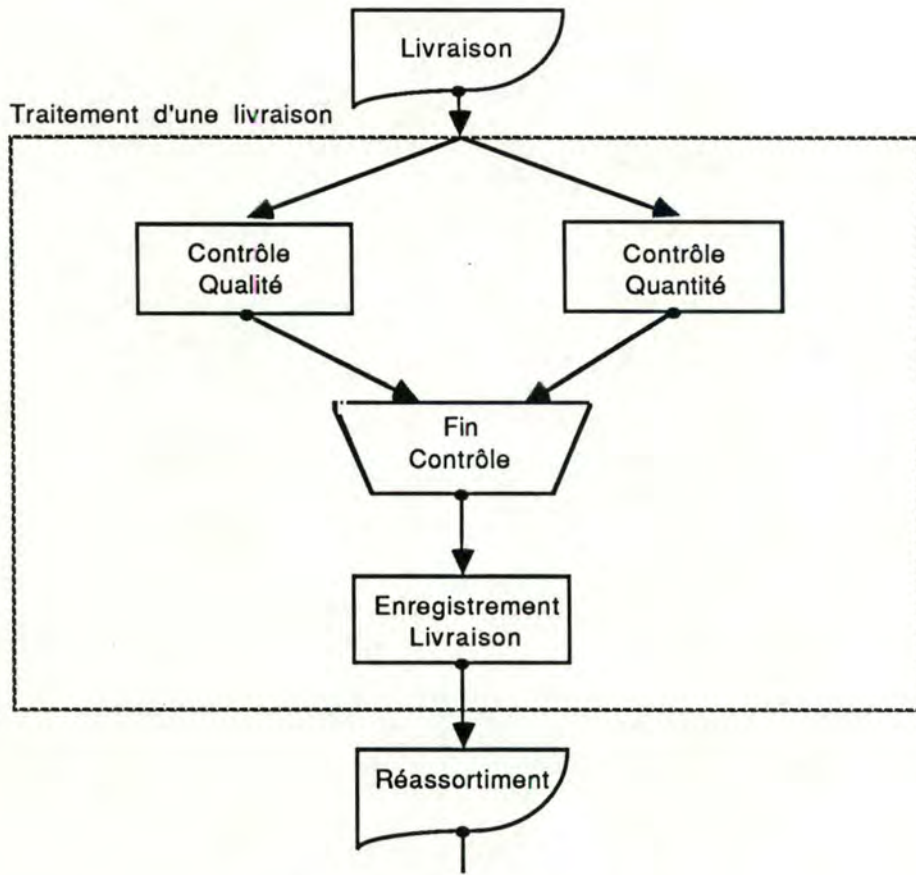


figure III.4 Traitement d'une livraison

Un schéma de la dynamique peut présenter plusieurs types d'événements terminaux. Par exemple, la terminaison de l' "acceptation du prêt" et la terminaison du "refus du prêt" dans la gestion de prêt illustré à la figure III.1. Dans ce cas, la durée de traitement d'une demande de prêt correspond à la durée de TRANSA(gén. de la "Demande de prêt", term. d'une "Acceptation du prêt"), si le prêt est accepté ou à la durée de TRANSA(gén. de la "Demande de prêt", gén. d'un "Refus du prêt") si le prêt est refusé.

On notera que l'événement initial d'une transaction n'est pas nécessairement un événement initial d'un schéma de la dynamique. Tout événement, à l'exception de l'événement terminal peut être un événement initial d'une transaction.

La figure III.5 illustre la situation selon laquelle à un événement initial correspondent plusieurs événements terminaux puisqu'à chaque génération de "Bon de Commande interne" correspondent autant de "colis expédiés" qu'il y a de "Prélèvements par parcours".

Dans ce cas, pour que le délai de TRANSA(gen. d'un "Bon de Commande interne",gén. de "Colis expédié") représente la durée du "traitement expédition", il faut que celui-ci corresponde au laps de temps nécessaire pour expédier tous les colis contenant les produits détaillés dans le bon de commande interne, ce qui explique pourquoi une transaction reprend tous les événements terminaux consécutifs à un événement initial.

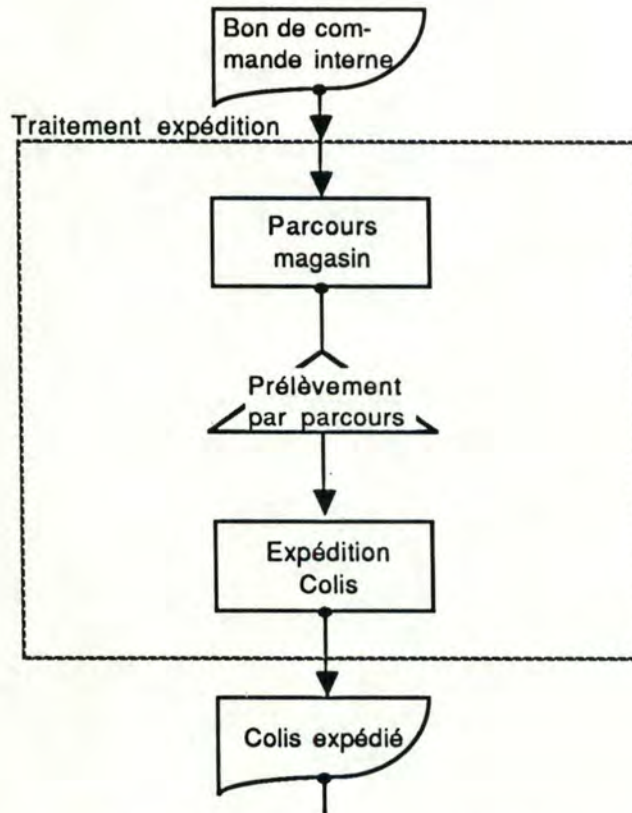


figure III.5 Traitement expédition

DELAI MOYEN D'UN ENSEMBLE DE TRANSACTIONS DE MEME TYPE

Le délai d'une transaction représente le fonctionnement exact d'un ensemble de processus. Toutefois, lors d'une première analyse du comportement d'un système, on est surtout intéressé par la performance moyenne de celui-ci.

Pour évaluer la performance moyenne d'un système, nous proposons de calculer le délai moyen d'un ensemble de transactions de même type qui se sont déroulées durant une période déterminée.

Ainsi, par exemple, nous pourrions obtenir le temps moyen de traitement d'une livraison (cf. figure III.4), les temps moyens pour décider si l'on accepte ou non une demande de prêt (cf. figure III.1), ou encore le temps moyen nécessaire pour expédier tous les colis qui contiennent les produits détaillés dans un bon de commande interne (cf. figure III.5).

DELAI MOYEN D'UN ENSEMBLE D' EVENEMENTS DE MEME TYPE

Pour localiser la mauvaise performance du système, il est intéressant de décomposer le délai moyen d'un type de transaction. Nous proposons à cet effet de calculer la moyenne des délais des événements de même type qui composent les transactions du type considéré.

Si on envisage la gestion de prêt illustrée à la figure III.1, il est par exemple intéressant de connaître le délai moyen qui sépare la demande de prêt du moment où la synchronisation a eu lieu et ainsi de déduire si la mauvaise performance du système est antérieure ou postérieure à cette synchronisation.

III.2. DEFINITION ET MESURES DU COMPORTEMENT

III.2.1. Préalable

Nous avons défini dans l'introduction une transaction comme un ensemble d'événements se succédant. Pour donner une définition plus précise de cette notion, il convient de préciser ce qu'on entend par succession d'événements. A cet effet, nous avons identifié quatre relations entre événements qui nous serviront également de base pour extraire une transaction du fichier LOG (cf III.3.) :

- <1> relation de dynamique
- <2> relation de passage
- <3> relation de précédence
- <4> relation d'ascendance

Avant d'examiner la signification exacte de ces relations, attardons-nous un instant sur la notion d'événement.

1. EVENEMENT

"Un événement représente un changement d'état qui survient à un moment donné de l'évolution du système d'information et qui correspond à un stimulus auquel ce système doit réagir ..." [Bodart, 87].

Les événements retenus par l'outil de simulation sont les suivants :

- | | |
|-----------------|------------------------|
| - génération | d'un message |
| - déclenchement | d'un processus |
| - activation | " " |
| - interruption | " " |
| - réactivation | " " |
| - terminaison | " " |
|
 | |
| - réalisation | d'une synchronisation. |

Nous avons étendu ce modèle en considérant la contribution à une synchronisation et la suppression de contribution(s) à une synchronisation comme des événements.

En effet, au niveau de la simulation, on peut considérer qu'une contribution est un stimulus auquel le système réagit en examinant, d'une part, si la condition de synchronisation est respectée, dans ce cas, il y a réalisation d'une ou plusieurs synchronisation(s) et, d'autre part, s'il n'y a pas lieu de supprimer une ou plusieurs contribution(s), par exemple, si leur durée de contribution "excède une période de rémanence fixée".

De plus, dans la représentation graphique d'une transaction (cf. IV.2) et, en particulier, dans la représentation animée (cf. IV.2.4), nous pensons qu'il est important de visualiser la contribution d'un événement à une synchronisation pour que l'utilisateur perçoive exactement le comportement du système. N'oublions pas, qu'à l'origine, le mécanisme de synchronisation a été préféré à l'approche PRE/POST condition [Pigneur, 84] car il permet de mettre en évidence le prédicat qui doit être vérifié pour qu'un processus puisse être déclenché.

"Les événements sont classés par type, dont les occurrences sont caractérisées par des survenances et des réactions équivalentes" [Bodart, 87].

2. SUCCESSION ENTRE EVENEMENTS

2.1. Relation de dynamique entre deux événements

Une relation de dynamique exprime une succession de deux événements correspondant à deux objets (message, processus, synchronisation) enchaînés suivant la dynamique spécifiée par l'utilisateur à l'aide des phrases DSL.

Selon cette définition, il peut exister une relation de dynamique :

- a> de la génération d'un message vers le déclenchement d'un processus
- b> de la génération d'un message vers la contribution à une synchronisation
- c> de l'activation d'un processus vers le déclenchement d'un processus
- d> de l'activation d'un processus vers la contribution à une synchronisation
- e> de la terminaison d'un processus vers la génération d'un message (1)
- f> de la terminaison d'un processus vers le déclenchement d'un processus
- g> de la terminaison d'un processus vers la contribution à une synchronisation
- h> de la réalisation d'une synchronisation vers le déclenchement d'un processus
- i> de la réalisation d'une synchronisation vers la contribution à une autre synchronisation

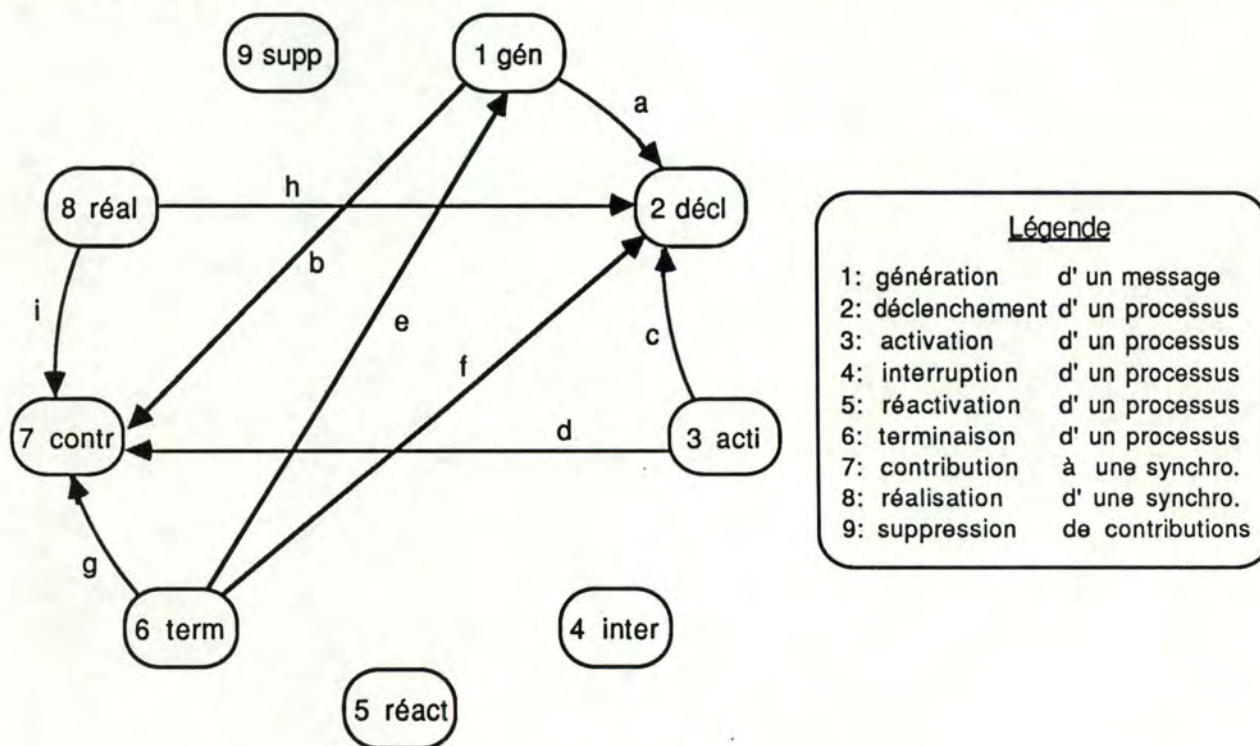
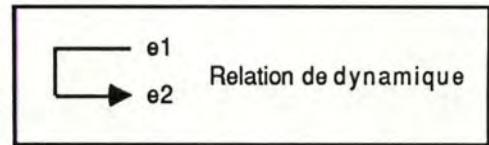


figure III.6 Relations de dynamique

(1) Le modèle de la dynamique ne spécifie pas à quel instant un processus génère un message, toutefois, le simulateur considère que c'est à la terminaison du processus que le message est généré.

Exemple

┌	1d08h30m	gén.	demande-de-prêt	#400
└─▶	1d08h30m	décl	enreg-demande-prêt	#401
	1d08h30m	acti	enreg-demande-prêt	#401
	1d08h31m22s	gén.	demande-de-prêt	#402
┌	1d08h54m47s	term	enreg-demande-prêt	#401
└─▶	1d08h54m47s	décl	préparation-dossier	#414



Propriétés

<1> Si un événement e1 est l'origine d'une relation de dynamique dont un événement e2 est la destination, alors e1 et e2 surviennent au même moment.

<2> Les événements suivants peuvent être l'origine d'une ou plusieurs relations de dynamique :

- génération d'un message
- activation d'un processus
- terminaison " "
- réalisation d'une synchronisation.

<3> Il existe au plus une relation de dynamique ayant comme destination un des événements suivants:

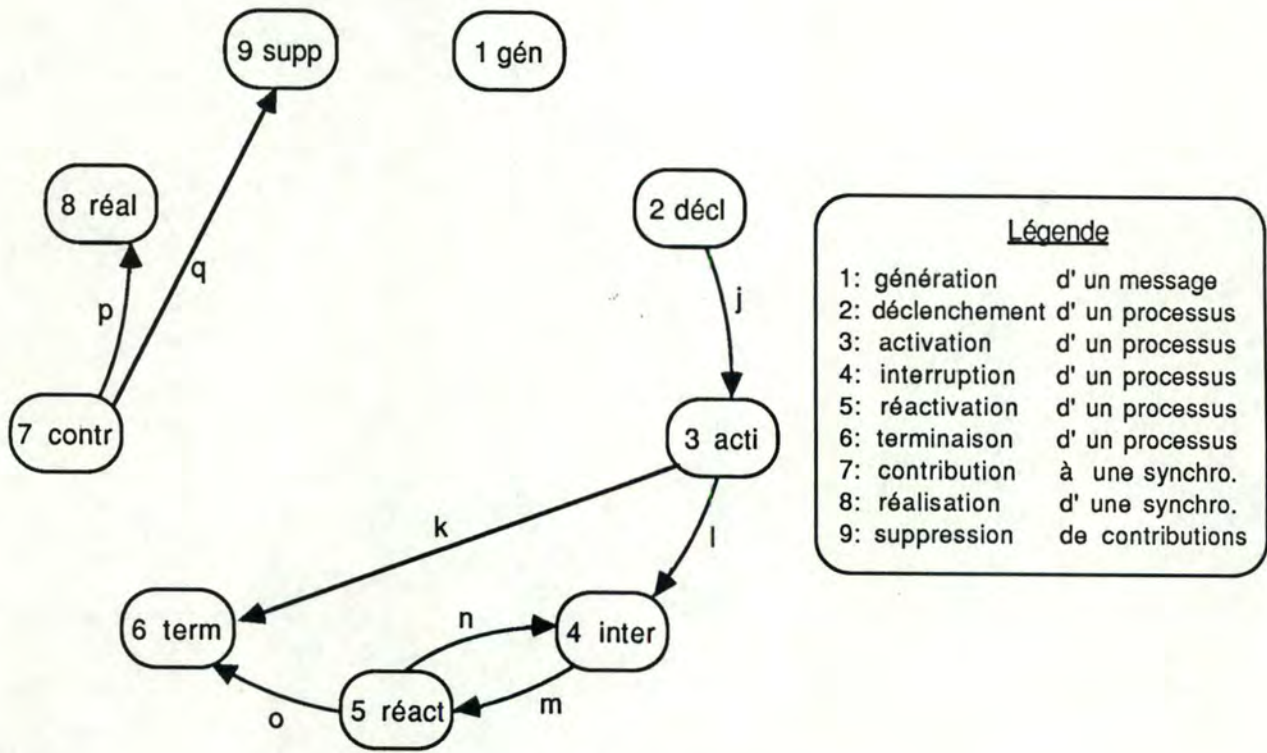
- génération d'un message
- déclenchement d'un processus
- contribution à une synchronisation.

2.2. Relation de passage entre deux événements

Une relation de passage exprime une succession de deux événements relatifs à un même objet (processus, synchronisation).

Selon cette définition, il peut exister une relation de dynamique :

j>	du déclenchement	vers l'activation	du même processus
k>	de l'activation	vers la terminaison	" "
l>	de l'activation	vers l'interruption	" "
m>	de l'interruption	vers la réactivation	" "
n>	de la réactivation	vers l'interruption	" "
o>	de la réactivation	vers la terminaison	" "
p>	de la contribution	vers la réalisation	d'une même synchronisation
q>	de la contribution	vers la suppression	de cette contribution.



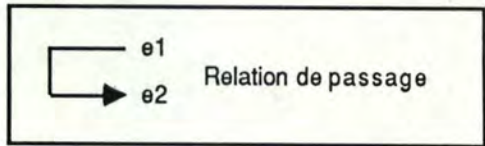
Légende

1: génération	d' un message
2: déclenchement	d' un processus
3: activation	d' un processus
4: interruption	d' un processus
5: réactivation	d' un processus
6: terminaison	d' un processus
7: contribution	à une synchro.
8: réalisation	d' une synchro.
9: suppression	de contributions

figure III.7 Relations de passage

Exemple

	1d08h30m	gén.	demande-de-prêt	#400
┌	1d08h30m	décl	enreg-demande-prêt	#401
└	1d08h30m	acti	enreg-demande-prêt	#401
┌	1d08h31m22s	gén.	demande-de-prêt	#402
└	1d08h54m47s	term	enreg-demande-prêt	#401
┌	1d08h54m47s	décl	préparation-dossier	#414



Propriétés

- <1> Si un événement e1 est l'origine d'une relation de passage dont un événement e2 est la destination, alors e1 survient avant e2 et pas nécessairement au même moment.
- <2> Il existe au plus une relation de passage ayant comme origine un des événements suivants :
 - déclenchement d'un processus
 - activation " "
 - interruption " "
 - réactivation " "
- <3> La contribution peut être l'origine d'une ou plusieurs relations de passage.
- <4> Les événements suivants sont la destination d'une relation de passage :
 - activation d'un processus
 - interruption " "
 - réactivation " "
 - terminaison " "

<5> La réalisation d'une synchronisation est la destination d'au moins deux relations de passage.

<6> La suppression de contribution(s) est la destination d'une ou plusieurs relations de passage.

2.3. Relation de précédence entre deux événements

Il existe une relation de précédence d'un événement e1 vers un événement e2 s'il existe une relation de dynamique de e1 vers e2 ou s'il existe une relation de passage de e1 vers e2.

Lorsqu'il existe une relation de précédence de e1 vers e2, nous dirons que e1 est prédécesseur de e2 et e2 est successeur de e1.

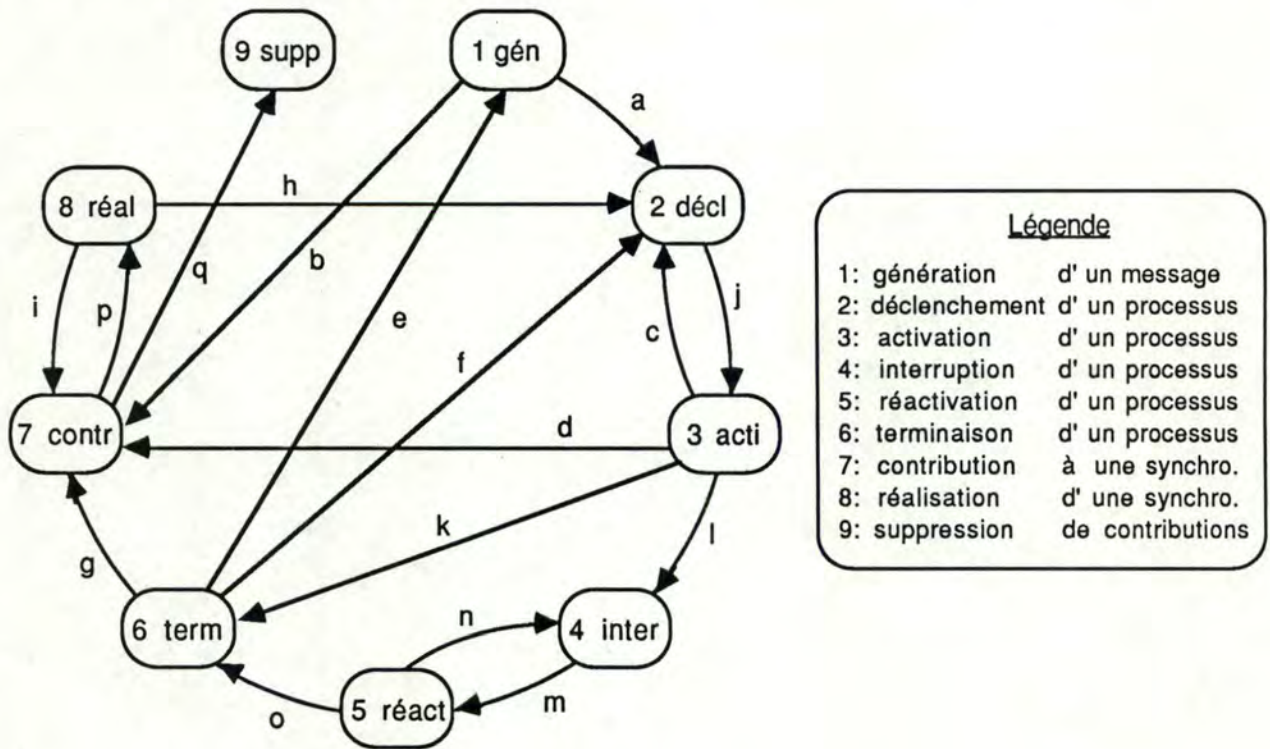
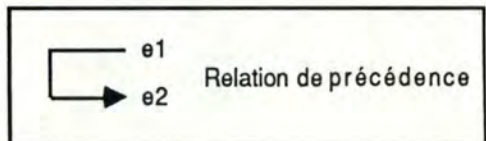


figure III.8 Relations de Précédence

Exemple

┌───┐	1d08h30m	gén.	demande-de-prêt	#400
└─┬─┘	1d08h30m	décl	enreg-demande-prêt	#401
└─┬─┘	1d08h30m	acti	enreg-demande-prêt	#401
└─┬─┘	1d08h31m22s	gén.	demande-de-prêt	#402
└─┬─┘	1d08h54m47s	term	enreg-demande-prêt	#401
└─┬─┘	1d08h54m47s	décl	préparation-dossier	#414



2.4. Relation d'ascendance entre deux événements

Il existe une relation d'ascendance d'un événement e1 vers un événement e2 si e2 est un successeur de e1 ou s'il existe un événement e3 tel qu'il existe une relation d'ascendance de e1 vers e3 et une relation d'ascendance de e3 vers e2.

Lorsqu'il existe une relation d'ascendance de e1 vers e2, nous dirons que e1 est ascendant de e2 et e2 est descendant de e1.

Notons que la suppression d'une contribution n'est l'origine d'aucune relation d'ascendance puisque cet événement n'est l'origine d'aucune relation de dynamique ou de passage.

III.2.2. Définition

Après avoir précisé la notion d'événement et défini les relations pouvant exister entre eux, nous sommes à même de donner une définition formelle du concept de transaction qui nous servira de base dans l'algorithme d'extraction décrit au point III.3.

Soit E l'ensemble des événements (1) issus d'une instanciation d'un modèle de la dynamique (telle qu'une instanciation dans le cadre d'une simulation),

i un élément de E,

T l'ensemble de tous les événements d'un même type descendant de i.

On appelle transaction, dont l'événement initial est i et l'ensemble des événements terminaux est T, l'ensemble des événements TRANSA(i,T) défini comme suit :

$$\text{TRANSA}(i,T) = \{i\} \cup T \cup \{ e \text{ appartenant à } E \text{ tel que } e \text{ est descendant de } i \text{ et ascendant d'un élément de } T \}.$$

A priori, n'importe quel événement peut être considéré comme un événement initial ou terminal; toutefois, certains événements sont plus pertinents que d'autres dans le contexte d'utilisation de la transaction.

Dès lors, les événements initiaux que nous avons retenus sont :

- la génération d'un message
- le déclenchement d'un processus

De même, nous n'avons retenu que les événements terminaux suivants :

- la génération d'un message
- la terminaison d'un processus
- la réalisation d'une synchronisation.

Exemple

Le lecteur vérifiera aisément que les événements reliés par une relation de précédence à la figure III.9. constituent la transaction TRANSA(gén. "demande-de-prêt" survenue à 1d8h30m, term. "acceptation-du-prêt" survenue le même jour à 17h12m).

(1) La contribution à une synchronisation et la suppression d'une contribution sont considérées comme des événements possibles (cf. III.2.1)

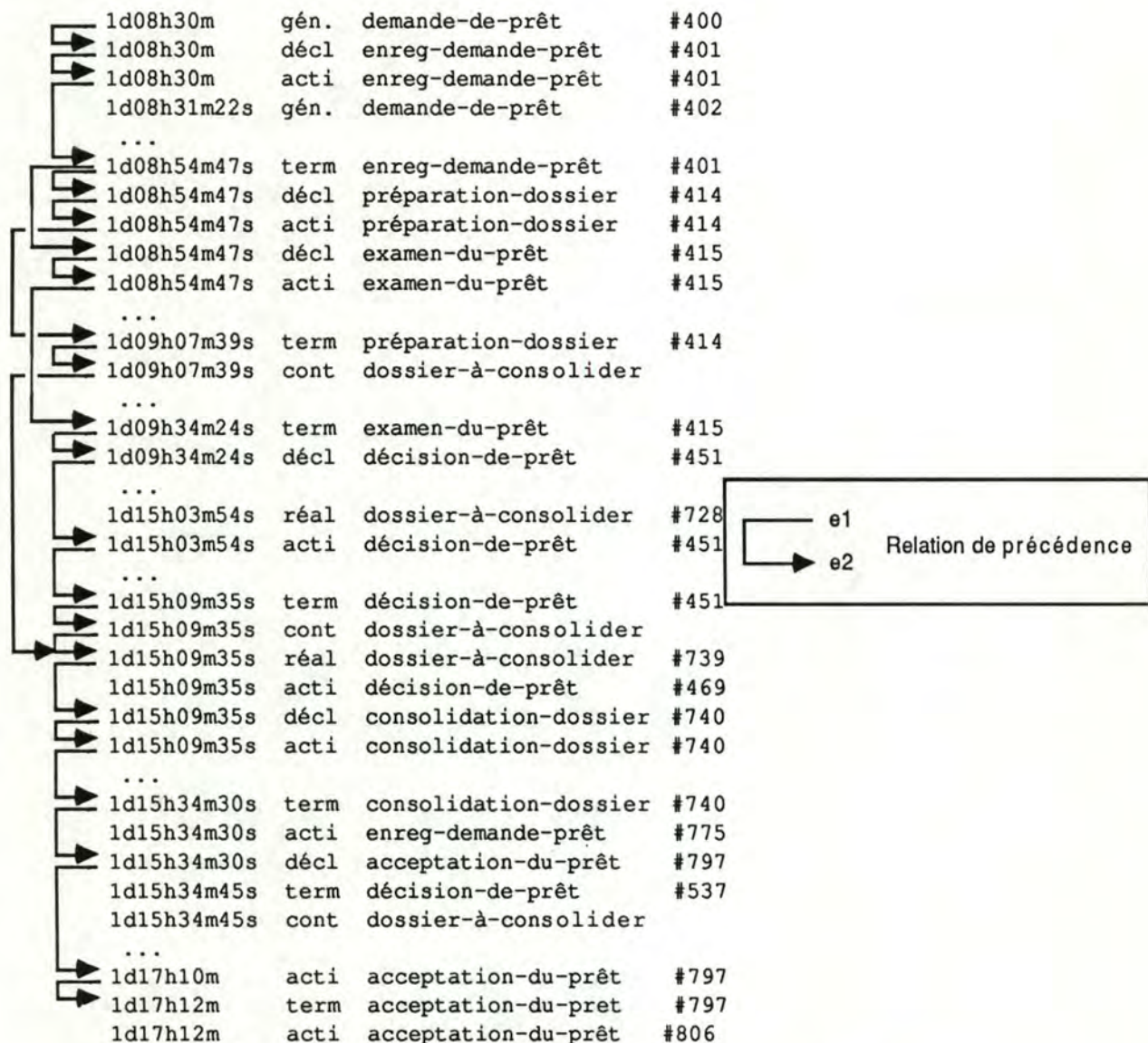


figure III.9 transaction de la "Gestion de prêt"

III.2.3. Mesures du comportement

L'objectif de la transaction est de permettre une évaluation des performances du système d'information. A cet effet, nous proposons différentes mesures de ce comportement qui seront juxtaposées sur des graphiques (cf. IV.2. et V.1.) ou représentées graphiquement (cf. V.2.).

1. COMPORTEMENT D'UNE TRANSACTION

1.1. Durée inter-événements

La durée inter-événements représente la durée séparant la survenance de deux événements e1 et e2 tels que e1 est ascendant de e2.

1.2. Délai d'un événement d'une transaction

Le délai d'un événement d'une transaction est la différence entre sa date de survenance et la date de survenance de l'événement initial de la transaction.

1.3. Délai d'un type d'événement

Le délai d'un type d'événement dont les occurrences appartiennent à une transaction est le délai maximum des événements de ce type qui appartient à cette transaction.

Ce délai correspond par exemple au délai nécessaire pour préparer le dossier d'une demande de prêt (cf. figure III.1). Dans cet exemple, le type d'événement possède une seule occurrence, ce qui n'aurait pas été le cas si on avait eu un déclenchement multiple; dans cette hypothèse, on aurait dû attendre la survenance du dernier événement du type considéré.

Ce délai n'est calculé que pour les types d'événement terminaux possibles, à savoir

- la génération d'un type de message
- la terminaison d'un type de processus
- la réalisation d'un point de synchronisation

1.4. Délai d'une transaction

Le délai d'une transaction est le délai du type d'événement terminal.

2. COMPORTEMENT D'UN TYPE DE TRANSACTION

2.1. Délai moyen d'un type de transaction

Le délai moyen d'un type de transaction est la moyenne des délais des transactions de ce type qui se sont déroulées endéans une période déterminée.

2.2. Délai minimum d'un type de transaction

Le délai minimum d'un type de transaction est le délai de la transaction la plus courte parmi les transactions de ce type qui se sont déroulées endéans une période déterminée.

2.3. Délai maximum d'un type de transaction

Le délai maximum d'un type de transaction est le délai de la transaction la plus longue parmi les transactions de ce type qui se sont déroulées endéans une période déterminée.

2.4. Délai moyen d'un type d'événement

Si on considère les transactions relatives à l'acceptation d'une demande de prêt (cf. figure III.1), ce délai correspond par exemple au délai moyen nécessaire pour préparer le dossier d'une demande de prêt le premier jour de la semaine.

Cette mesure se calcule en retenant, pour chaque transaction du type considéré qui s'est déroulée endéans une période déterminée, le délai d'un type d'événement (dans notre exemple, le délai de la terminaison du processus "Préparation dossier") et, en faisant la moyenne de ces délais.

III.3. ALGORITHME DE CONSTRUCTION D'UNE TRANSACTION

III.3.1. Préalable

L'objectif de cet algorithme est d'extraire d'une trace de la simulation la transaction dont l'événement initial est "Ev_init" et le(s) événement(s) terminaux sont de type "T_év_term".

Nous avons défini une transaction comme un ensemble d'événements reliés par des relations de précédence (cf. III.2.2.). L'extraction d'une transaction du fichier LOG suppose donc que l'on puisse retrouver ces relations entre événements.

La trace produite actuellement par la simulation permet uniquement de retrouver les relations de passage entre les changements d'état d'un processus; ce qui nous a amenés à définir une nouvelle trace de la simulation.

Une manière de la définir consiste à attribuer un numéro à chaque événement. Ce numéro est ensuite repris par chacun de ses successeurs, comme illustré à la figure III.10.

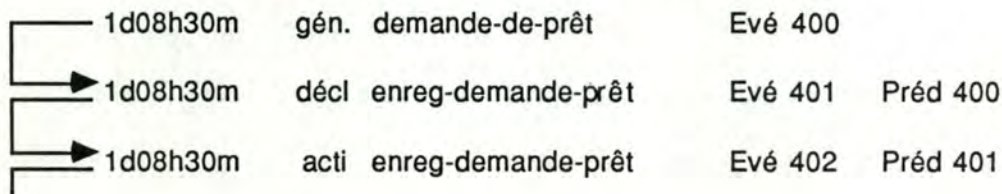


figure III.10

La production de cette trace nécessitant trop de modifications au niveau du simulateur, nous avons donc été amenés à définir une nouvelle trace moins élégante mais qui tient compte des contraintes de réalisation.

La nouvelle trace de la simulation est une suite d'événements dont un élément est composé des informations suivantes :

[Date] [Type_év] [nom_obj] [n°_objet] ([n°_obj_préd] [mult])* [Type_év_préd] [attente] [repos]

où [Date] représente la date de survenance de l'événement,

[Type_év] représente un des types d'événement suivants :

- | | | |
|----------------------|-----------------------|----------|
| <1> la génération | d'un message | (gen.) |
| <2> le déclenchement | d'un processus | (décl.) |
| <3> l'activation | " " | (acti.) |
| <4> l'interruption | " " | (inter.) |
| <5> la réactivation | " " | (réact.) |
| <6> la terminaison | " " | (term.) |
| <7> la contribution | à une synchronisation | (cont.) |
| <8> la réalisation | d'une synchronisation | (réal.) |
| <9> la suppression | de contribution(s) | (supp.) |

[nom_obj] représente le nom de l'objet (message, processus, synchronisation)

[n°_objet] représente le numéro de l'objet associé à l'événement courant. Toutefois, lors de la contribution, comme on ne connaît pas, à ce moment, la synchronisation à laquelle elle va participer, cette information représente un numéro attribué de manière unique à chaque contribution.

Lors d'une suppression de contribution(s), la synchronisation n'a jamais existé et, par conséquent, dans ce cas, le [nom_obj] est indéterminé.

[n°_obj_préd] représente le numéro de l'objet associé à un événement précédant l'événement courant (cf. III.2.1). Ce numéro est présent si l'événement courant est :

- la destination d'une relation dynamique, c'est-à-dire <1>, <2> ou <7> ,
- la réalisation d'une synchronisation,
- une suppression de contributions.

Dans ces deux derniers cas, il y aura autant de [n°_obj_préd] que de contributions qui participent à la synchronisation ou de contributions supprimées. Lors d'une réalisation, le paramètre [mult] associé à chaque contribution prend la valeur Vrai s'il s'agit d'une contribution multiple, la valeur Faux sinon.

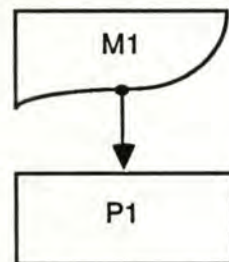
[Type_évé_préd] représente le type de l'événement prédécesseur dans le cas où [n°_obj_préd] est le numéro d'un processus.

[attente] et [repos] représentent la durée pendant laquelle le processus a été en attente ou en repos avant d'être (ré)activé lorsque l'événement courant est une activation ou une réactivation. Ces informations ne sont pas utiles pour l'extraction d'une transaction mais elles sont nécessaires au calcul des temps d'attente et de repos d'un processus qui sont juxtaposés sur le graphe des objets (cf.IV.2.3.).

Par rapport à la trace de la simulation produite actuellement, nous avons ajouté le numéro de l'objet associé à un événement précédant l'événement courant (n°_obj_préd). Cette information nous est utile pour retrouver les relations de dynamique existant entre deux événements.

En effet, si on considère l'exemple de la figure III.11, il est impossible de déterminer, sans faire d'hypothèse sur le fonctionnement de la simulation, si le message numéro 1 en génération déclenche le processus numéro 3 ou numéro 4.

Toutefois, comme l'activation ou la terminaison d'un processus peuvent être l'origine d'une relation dynamique, il convient également de préciser le type de l'événement (Type_évé_pred) pour identifier sans ambiguïté la relation de dynamique.



Extrait du LOG

```

0d08h30m gen M1 #1
0d08h30m gen M1 #2
0d08h30m décl P1 #3
0d08h30m décl P1 #4
  
```

figure III.11

Lors d'une réalisation d'une synchronisation, ou d'une suppression de contribution(s), la simulation produit, par exemple, les informations suivantes :

```

1d15h09m35s réal dossier-à-consolider #739
5d          supp 8 ctnb TERMINAISON DE préparation-dossier A dossier-à-consolider
  
```


Ces données ne nous permettent pas de retrouver les contributions ayant participé à une synchronisation ou les contributions tuées. Dès lors, nous proposons d'associer à chaque contribution un numéro repris lors de la réalisation ou de la suppression pour nous permettre d'identifier les relations de passage entre, d'une part, les contributions et, d'autre part, une synchronisation ou une suppression de contribution(s) (cf. III.2.1).

En plus du numéro de contribution, la réalisation reprend un paramètre [mult] pour chaque contribution. Cette information indique si la contribution est unique ou multiple, et donc si elle peut encore participer à une autre réalisation ou être supprimée par après.

III.3.2. Spécification

Entrées

Ev_init : Événement
 T_év_term : Type d'événement
 TRACE : Suite d'événements telle que définie au point III.3.1

Pré-conditions

- <1> TRACE est triée par ordre de survenance des événements
- <2> Ev_init est un événement initial possible (cf. III.2.2.), de même, T_év_term est le type d'un événement terminal possible (cf. III.2.2.).

Sortie

TRANSA : Suite d'événements identique à TRACE
 Transa_ok : Booléen

Post-conditions

- <1> TRANSA est trié par ordre de survenance des événements
- <2> Transa_ok est Vrai et TRANSA contient les événements de la transaction dont l'événement initial est Ev_init, et le type des événements terminaux est T_ev_term.
- ou Transa_ok est Faux et TRANSA est indéterminée

III.3.3. Algorithme

1. IDEES DE BASE

L'extraction d'une transaction se fait en deux étapes :

- <1> La première étape consiste à extraire de la trace l'ensemble de tous les événements descendant de Ev_init, y compris cet événement, et de les placer dans une suite d'événements appelée DESC.

La construction de DESC se fait progressivement, en parcourant séquentiellement la nouvelle trace de la simulation. Le traitement d'un élément quelconque peut se faire de deux manières différentes :

Une première méthode consiste à vérifier si l'événement courant est un descendant de l'événement initial en testant s'il est le successeur d'au moins un élément de DESC.

Cette méthode pose un problème de performance puisque si l'événement courant n'est pas un descendant de l'événement initial, ce qui est le cas le plus fréquent, on a parcouru en vain tous les éléments de DESC. Nous avons donc développé une autre technique pour construire DESC.

Cette deuxième technique consiste toujours à vérifier si l'événement courant est un descendant de l'événement initial mais en utilisant pour ce faire l'ensemble des événements de DESC susceptibles d'être prédécesseur de l'événement courant.

Pour tester si l'événement courant est un descendant de l'événement initial, il suffit de vérifier s'il constitue un successeur d'au moins un élément de ce sous-ensemble. Dans la négative, on aura donc parcouru un ensemble d'événements moins grand que dans la première méthode.

Ainsi, par exemple, dans la gestion de prêt (cf. figure III.1), avant de traiter le dernier événement de la transaction, DESC contient les événements suivants :

- génération de	demande de prêt,
- déclenchement, activation, terminaison de	enregistrement demande de prêt
- déclenchement, activation, terminaison de	préparation du dossier
- déclenchement, activation, terminaison de	examen du prêt
- déclenchement, activation, terminaison de	décision de prêt
- contribution au	dossier à consolider
- réalisation du	dossier à consolider
- déclenchement, activation, terminaison de	consolidation du dossier
- déclenchement, activation de	acceptation du prêt

Parmi ces événements, le seul susceptible d'être le prédécesseur de la terminaison de l'acceptation du prêt est l'activation de ce même processus.

De plus, avec la première technique, on est obligé de parcourir toute la trace de la simulation, pour être certain d'avoir tous les descendants de l'événement initial, alors qu'avec la seconde technique, on est sûr que DESC contient tous les descendants de l'événement initial lorsque l'ensemble des prédécesseurs possibles est vide. Toutefois, comme nous le verrons dans l'algorithme, l'ensemble des cas à envisager est plus important, ce qui justifie l'emploi d'invariants.

<2> La deuxième étape consiste à extraire de DESC les événements de type "T_év_term" et leurs ascendants et à les placer dans une suite d'événements, appelée TRANSA.

La construction de TRANSA se fait progressivement, en parcourant la suite d'événements DESC en sens inverse. Le traitement de l'événement courant consiste à vérifier s'il est l'ascendant d'un événement terminal. Tout comme pour la première étape, nous utilisons l'ensemble des événements susceptibles d'être un successeur de l'événement courant pour parcourir le moins possible d'événements.

A l'issue de ces deux étapes, TRANSA contient l'ensemble des événements d'une transaction en vertu de la définition donnée au point III.2.2.

2. ALGORITHME

2.0. Conventions d'écriture

S[i] désigne le ième élément de la suite S. Par exemple, TRANSA[1] désigne le premier événement de TRANSA.

S[i:j] désigne tous les éléments i, i+1, ..., j de la suite S. Par exemple, TRACE[1:10] désigne les dix premiers événements de la trace.

S.comp[i] désigne le composant "comp" du ième élément de la suite S. Par exemple, TRACE.Date[1] désigne la date de survenance du premier événement de la trace.

2.1. Procédure Calcul_DESC

2.1.1. Spécification

Entrées

Ev_init : Événement
TRACE : Suite d'événements

Pré-conditions

- <1> TRACE est triée par ordre de survenance des événements
- <2> Ev_init est un événement initial possible (cf. III.2.2.).

Sortie

DESC : Suite d'événements

Post-conditions

- <1> DESC est trié par ordre de survenance des événements
- <2> DESC contient "Ev-init" et tous les événements de TRACE descendant de Ev_init.

2.1.2. Invariant

2.1.2.1. Description

La proposition invariante qui doit être vérifiée avant chaque itération est la suivante : (P1 ET P2 ET P3 ET P4)

- <P1> DESC[1:k] contient Ev_init ainsi que les événements de TRACE[1:i-1] descendant de Ev_init. De plus, DESC[1:k] est trié par ordre de survenance des événements
- <P2> OR-DYN est l'ensemble des événements de DESC[1:k] qui peuvent être l'origine d'une relation de dynamique dont la destination n'appartient pas à DESC[1:k].
- <P3> OR-PAS est l'ensemble des événements de DESC[1:k] qui sont l'origine d'une relation de passage dont la destination n'appartient pas à DESC[1:k].
- <P4> Date_anc = TRACE.Date[i-1].

2.1.2.2. Illustration

Nous allons illustrer les différents composants de l'invariant, dans l'exemple de la figure III.12, pour l'extraction des descendants du déclenchement de P1 survenu à 8h30m.

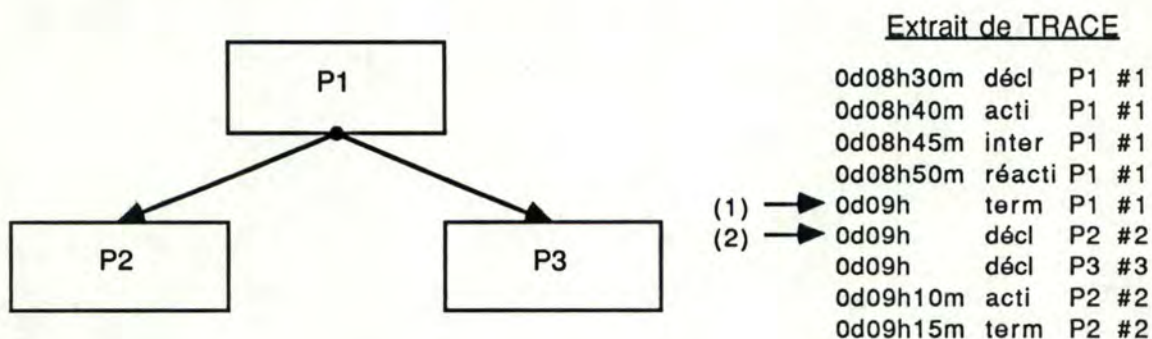


figure III.12

Après le traitement des quatre premiers événements de TRACE (cf (1), figure III.12), l'invariant est le suivant :

- <P1> DESC[1:4] contient ces quatre événements
- <P2> OR-DYN = {}
- <P3> OR-PAS = { réactivation du P1 numéro 1 }
- <P4> Date_anc = 8h50m

Après avoir traité la terminaison de P1, (cf (2), figure III.12), l'invariant devient :

- <P1> DESC[1:5] contient les cinq premiers événements de TRACE
- <P2> OR-DYN = { terminaison du P1 numéro 1 }
- <P3> OR-PAS = {}
- <P4> Date_anc = 9h

2.1.3. Algorithme

2.1.3.1. Corps du programme

Enumération des cas possibles

	A1	A2	A3	A4	A5	A6	A7	A8
C1	V	V	V	V	F	F	F	F
C2	V	V	F	F	V	V	F	F
C3	V	F	V	F	V	F	V	F

Conditions

C1 TRACE.date[i] = Date_anc

C2 L'événement courant est la destination d'une ou plusieurs relation(s) de passage dont l'origine appartient à OR-PAS.

Par exemple, à la figure III.11., si la terminaison de P1 est l'événement courant, il est la destination d'une relation de passage dont l'origine est la réactivation de P1.

Connaissant la structure de la suite TRACE (cf. III.3.1.), vérifier cette condition consiste à comparer le numéro d'objet de chaque événement appartenant à OR-PAS avec le numéro d'objet de l'événement courant si celui-ci est le changement d'état d'un processus, ou avec chaque numéro de contribution précédant l'événement courant si celui-ci est une réalisation ou une suppression de contribution(s).

C3 L'événement courant est la destination d'une relation de dynamique dont l'origine appartient à OR-DYN

Par exemple, à la figure III.11., si le déclenchement de P2 est l'événement courant, il est la destination d'une relation de passage dont l'origine est la terminaison de P1.

Pour tester cette condition, il faut vérifier s'il existe un événement dans OR-DYN dont le [n°_objet] est le même que TRACE.n°_obj_pred[i], en utilisant Type_évé_pred[i] pour déterminer éventuellement si c'est l'activation ou la terminaison du processus qui est l'origine de la relation.

Actions

A1 Ce cas est impossible, car, d'après la propriété <3> de la relation de dynamique et les propriétés <4>, <5> et <6> de la relation de passage (cf. III.2.1), un événement ne peut être la destination, à la fois d'une relation de dynamique et de passage.

A2 L'événement courant est la destination d'une, ou plusieurs, relation(s) de passage dont l'origine appartient à OR-PAS.

D'après les propriétés <4>, <5> et <6> de la relation de passage, le type de l'événement courant ne peut être que -l'activation, l'interruption, la réactivation, la terminaison- d'un processus, la réalisation d'une synchronisation, ou la suppression de contribution(s).


```

k:=k+1;
DESC[k] := TRACE[i];
case TRACE.Type_évé[i] of
  acti  : OR-PAS := OR-PAS \ {origine de la relation de passage}
         OR-PAS := OR-PAS U {événement courant}
         OR-DYN := OR-DYN U {événement courant}

  inter,
  réact : OR-PAS := OR-PAS \ {origine de la relation de passage}
         OR-PAS := OR-PAS U {événement courant}

  term  : OR-PAS := OR-PAS \ {origine de la relation de passage}
         OR-DYN := OR-DYN U {événement courant}

  réali : for each c : origine de la relation de passage do
         if NOT mult
           then OR-PAS := OR-PAS \ {c}
           OR-DYN := OR-DYN U {événement courant}
         end if
  supp  : for each c : origine de la relation de passage do
         OR-PAS := OR-PAS \ {c}
         end for
endcase
i := i+1

```

- A3 L'événement courant est la destination d'une relation de dynamique dont l'origine appartient à OR-DYN.

D'après la propriété <3> de la relation de dynamique, le type de l'événement courant ne peut être que la génération d'un message, le déclenchement d'un processus ou la contribution à une synchronisation.

```

k:=k+1;
DESC[k] := TRACE[i];
case TRACE.Type_évé[i] of
  gen  : OR-DYN := OR-DYN U {événement courant};
  décl,
  cont : OR-PAS := OR-PAS U {événement courant};
endcase
i := i+1;

```

- A4 L'événement courant n'est pas un descendant de Ev_init

```
i := i+1;
```

- A5 Ce cas est impossible, car d'après la propriété <3> de la relation de dynamique et les propriétés <4>, <5> et <6> de la relation de passage (cf. III.2.1), un événement ne peut être la destination, à la fois d'une relation de dynamique et de passage.

- A6 L'événement courant est la destination d'une ou plusieurs relation(s) de passage dont l'origine appartient à OR-PAS et l'événement courant est survenu plus tard que l'événement survenu juste avant.

D'après la propriété <1> de la relation de dynamique, les événements de DESC survenus avant le moment courant ne peuvent pas être l'origine d'une relation de dynamique.

```

OR-DYN := {};
Date_anc := TRACE.Date[i];
{ actions de A2 }

```

- A7 Ce cas est impossible car, d'après la propriété <1> des relations de dynamique, les événements qui sont l'origine et la destination d'une telle relation surviennent au même moment.

- A8 L'événement courant n'est pas un descendant de Ev_init et il est survenu plus tard que l'événement survenu juste avant.

```

OR-DYN := {};
Date_anc := TRACE.Date[i];
i := i+1;

```


2.1.3.2. Initialisation

```

i := élément de TRACE contenant Ev_init;
case TRACE.Type_évé[i] of
  gen  : OR-DYN := {Ev_init};
        OR-PAS := {};
  décl : OR-DYN := {};
        OR-PAS := {Ev_init};
endcase
k := 1;
DESC[k] := TRACE[i];
i := i+1;

```

2.1.3.3. Terminaison

Condition d'arrêt

(fin TRACE) OU (OR-PAS = {} ET OR-DYN = {})

Post-condition

La post-condition est vérifiée après la terminaison de l'algorithme car

<1> D'après la proposition P1, DESC est trié par ordre de survenance des événements

<2> Si l'algorithme se termine avec (fin TRACE) = Vrai, d'après la proposition <P1>, DESC contient "EV_init" ainsi que tous les événements de TRACE descendant de "Ev_init".

Si l'algorithme se termine avec ORPAS et OR-DYN vides, d'après les propositions <P2> et <P3>, aucun événement de DESC ne peut être l'origine d'une relation de précédence dont la destination n'appartient pas à DESC. On en déduit de <P1> que DESC contient Ev_init ainsi que tous les événements de TRACE descendant de Ev_init.

2.2. Procédure Calcul_TRANSA

2.2.1. Spécification

Entrées

T_év_term : Type d'événement
DESC : Suite d'événements

Pré-conditions

<1> DESC contient Ev_init ainsi que tous ses descendants et il est trié par ordre de survenance des événements.

<2> T-év_term est le type d'un événement terminal possible (cf. III.2.2).

Sorties

TRANSA : Suite d'événements
Transa_ok : Booléen

Post-conditions

<1> TRANSA est trié par ordre de survenance des événements

<2> Transa_ok est Vrai et TRANSA contient tous les événements de la transaction dont l'événement initial est Ev_init et le type des événements terminaux est T_év_term,
ou Transa_ok est faux et DESC ne contient aucun événement de type T_év_term.

2.2.2. Invariant

2.2.2.1. Description

La proposition invariante qui doit être vérifiée avant chaque itération est la suivante : (P1 ET P2 ET P3 ET P4)

- <P1> TRANSA[1:k] contient les événements de DESC[i:N] de type T_év_term et leurs ascendants appartenant à DESC[i:N], où N est la taille de la suite DESC. De plus, TRANSA[1:k] est trié par ordre inverse de date de survenance des événements
- <P2> DES-DYN est l'ensemble des événements de TRANSA[1:k] qui sont la destination d'une relation de dynamique dont l'origine n'appartient pas à TRANSA[1:k].
- <P3> DES-PAS est l'ensemble des événements de TRANSA[1:k] qui peuvent être la destination d'une, ou plusieurs, relation(s) de passage dont l'origine n'appartient pas à TRANSA[1:k].
- <P4> Date_anc = DESC.Date[i+1].

2.2.2.2. Illustration

Nous allons illustrer les différentes composantes de l'invariant, dans l'exemple de la figure III.13, pour l'extraction des ascendants de terminaisons de P2.

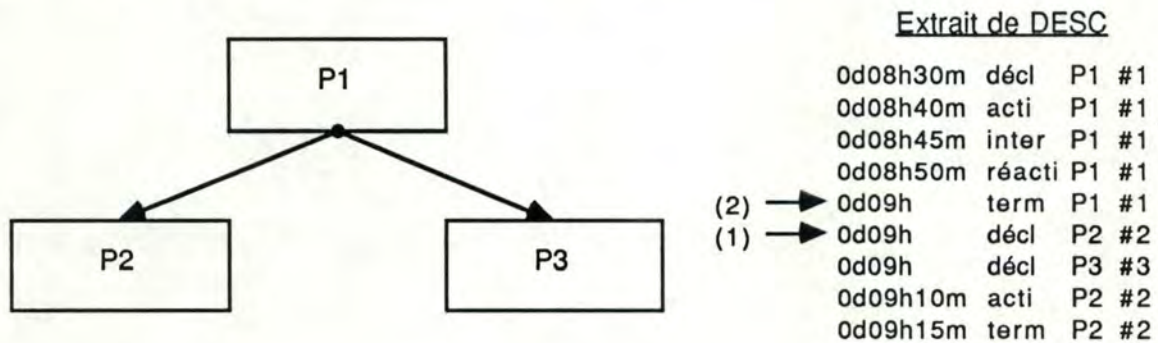


figure III.13

Après le traitement des trois dernières lignes de DESC (cf (1) figure III.13), l'invariant est le suivant :

- <P1> TRANSA[1:2] contient la terminaison et l'activation de P2
- <P2> DES-DYN = {}
- <P3> DES-PAS = { activation du P2 numéro 2 }
- <P4> Date_anc = 9h

Après avoir traité le déclenchement du P2 (cf (2) figure III.13), l'invariant devient :

- <P1> DESC[1:3] contient la terminaison, l'activation et le déclenchement de P2
- <P2> DES-DYN = { déclenchement du P2 numéro 2 }
- <P3> DES-PAS = {}
- <P4> Date_anc = 9h

2.2.3. Algorithme

2.2.3.1. Corps du programme

Enumération des cas possibles

	A1	A2	A3	A4	A5	A6	A7	A8
C1	V	V	V	V	F	F	F	F
C2	V	V	F	F	V	V	F	F
C3	V	F	V	F	V	F	V	F

Conditions

C1 DESC.date[i] = Date_anc

C2 L'événement courant est l'origine d'une relation de passage dont la destination appartient à DES-PAS.

Par exemple, à la figure III.12, si le déclenchement de P2 est l'événement courant, il est l'origine d'une relation de passage dont la destination est l'activation de P2

Si l'événement courant est le changement d'état d'un processus, la condition C2 est vraie s'il existe un événement de DES-PAS dont le numéro d'objet est identique à celui de l'événement courant.

Si l'événement courant est une contribution, la condition C2 est vraie si le numéro de la contribution est repris dans la liste des contributions participant à une synchronisation de DES-PAS, ou dans la liste des contributions tuées par une suppression appartenant à DES-PAS.

C3 L'événement courant est l'origine d'une ou plusieurs relation(s) de dynamique dont la destination appartient à DES-DYN.

Par exemple, à la figure III.12., si la terminaison de P1 est l'événement courant, il est l'origine d'une relation de dynamique dont la destination est le déclenchement de P2.

Pour tester cette condition, il faut vérifier s'il existe un événement dans DES-DYN dont le [n°_objet_pred] est le même que DESC.n°_objet[i], en utilisant Type_év_pred[i] pour déterminer éventuellement si c'est l'activation ou la terminaison du processus qui est l'origine de la relation.

Actions

A1 L'événement courant est à la fois l'origine d'une relation de passage dont la destination appartient à DES-PAS et l'origine d'une relation de dynamique dont la destination appartient à DES-DYN.

D'après la propriété <2> de la relation de dynamique, et la propriété <2> de la relation de passage (cf. III.2.1), le type de l'événement courant ne peut être que l'activation d'un processus.

```
k:=k+1;
TRANSA[k] := DESC[i];
DES-PAS := DES-PAS \ {destination de la relation de passage}
DES-PAS := DES-PAS U {événement courant}
i := i-1
```


- A2 L'événement courant est l'origine d'une relation de passage dont la destination appartient à DES-PAS.

D'après la propriété <2> et <3> de la relation de passage (cf. III.2.1), le type de l'événement courant ne peut être que -le déclenchement, l'activation, l'interruption, la réactivation- d'un processus ou la contribution à une synchronisation.

```

k:=k+1;
TRANSA[k] := DESC[i];
case DESC.Type_évé[i] of
  décl :   DES-PAS := DES-PAS \ {destination de la relation de passage}
          DES-DYN := DES-DYN U {événement courant}
  acti,
  inter,
  réact :  DES-PAS := DES-PAS \ {destination de la relation de passage}
          DES-PAS := DES-PAS U {événement courant}
  cont :   DES-DYN := DES-DYN U {événement courant}
endcase
i := i-1

```

- A3 L'événement courant est l'origine d'une relation de dynamique dont la destination appartient à DES-DYN

D'après la propriété <2> de la relation de dynamique (cf. III.2.1), le type de l'événement courant ne peut être que la génération d'un message, -l'activation, la terminaison- d'un processus et la réalisation d'une synchronisation.

```

k:=k+1;
TRANSA[k] := DESC[i];
DES-DYN := DES-DYN \ {destination de la relation de dynamique};
case DESC.Type_évé[i] of
  gen :   DES-DYN := DES-DYN U {événement courant};
  acti,
  term,
  real :  DES-PAS := DES-PAS U {événement courant};
endcase
i := i-1;

```

- A4 L'événement courant n'est pas l'origine de relation de précédence. L'événement peut toutefois être un événement terminal. Rappelons que le type de l'événement terminal est la génération d'un message, la terminaison d'un processus ou la réalisation d'une synchronisation (cf. III.2.2.)

```

if DESC.Type_évé[i] = T_ev_term
  then
  begin
    k:=k+1;
    TRANSA[k] := DESC[i];
    case T_év_term of
      gen :   DES-DYN := DES-DYN U {événement courant};
      term,
      réali : DES-PAS := DES-PAS U {événement courant};
    endcase
  end
i := i-1;

```

- A5 Ce cas est impossible car, d'après la propriété <1> de la relation de dynamique, l'origine et la destination d'une telle relation surviennent au même moment.

- A6 L'événement courant est l'origine d'une relation de passage dont la destination appartient à DES-PAS, et l'événement courant est survenu plus tôt que l'événement survenu juste après.

D'après la propriété <1> de la relation de dynamique, les événements de TRANSA survenus après le moment courant ne peuvent pas être la destination d'une relation de dynamique dont l'origine n'appartient pas à TRANSA.

```
DES-DYN := {};
Date_anc := DESC.Date[i];
{ actions de A2 }
```

- A7 Ce cas est impossible car, d'après la propriété <1> de la relation de dynamique, l'origine et la destination d'une telle relation surviennent au même moment.

- A8 L'événement courant n'est pas l'origine d'une relation de précedence et est survenu plus tôt que l'événement survenu juste après.

```
DES-DYN := {};
Date_anc := DESC.Date[i];
{ actions de A4 }
```

2.1.3.2. Initialisation

```
i := dernier élément de DESC ;
k := 0
```

2.1.3.3. Terminaison

Condition d'arrêt

(fin DESC)

Terminaison

```
if (TRANSA = {})
  then
    Transa_ok := Faux
  else
    begin
      Transa_ok := Vrai;
      Inverser l'ordre des éléments de TRANSA;
    end
```

Post-condition

La post-condition est vérifiée après la terminaison de l'algorithme car

- Si Transa_ok est Vrai, alors, d'après <P1> et la définition de DESC (cf. pré-conditions), TRANSA contient la transaction dont l'événement initial est "Ev_init" et le type des événements terminaux est T_év_term.
- Si Transa_ok est Faux, alors, d'après <P1>, DESC ne contient pas d'événement de type T_év_term. De plus, comme DESC est l'ensemble des descendants de Ev_init (cf. pré-conditions), TRACE ne contient pas de transaction dont l'événement initial est Ev_init, et le type des événements terminaux est T_év_term.

- CHAPITRE IV-

Outils de représentation du
comportement du système

INTRODUCTION

Nous allons envisager dans ce chapitre deux formes de représentation du comportement du système d'information : une forme statique tout d'abord, avec le graphe des événements et le graphe des objets. Nous décrirons ensuite une représentation animée du comportement global du système d'information, ainsi qu'une animation d'un aspect de la dynamique du système, en l'occurrence le point de synchronisation. Chacune de ces représentations sera détaillée tant au niveau des informations affichées qu'au niveau des fonctionnalités à offrir à l'utilisateur.

IV.1. OUTILS DE REPRESENTATION DU COMPORTEMENT DU SYSTEME

IV.1.1. Objectifs

La représentation du **comportement global** du système d'information pose un problème de représentation graphique étant donné le nombre important d'événements auquel le système réagit. Nous avons donc choisi de ne visualiser **qu'une partie** de ce comportement en montrant, soit, la **réaction du système** d'information à un événement initial, soit, une **transaction** du système (cf III.2.2.).

Visualiser le comportement du système d'information en réaction à un événement initial consiste à montrer, par exemple, comment le système de gestion de prêt (cf. figure III.1) réagit à la survenance d'une demande de prêt.

La visualisation d'une transaction consiste, dans le même exemple, à rendre compte du comportement du système lors du traitement d'une demande de prêt qui sera acceptée. Une transaction est donc une réaction particulière du système d'information à un événement initial.

Ces deux représentations, complémentaires, permettent une interprétation du bon ou du mauvais comportement du système en réaction à un événement initial.

Nous proposons trois outils de représentation de ce comportement :

- le graphe des événements
- le graphe des objets
- une animation

Le **graphe des événements** donne une représentation graphique du système d'information sous la forme d'événements engendrés directement ou indirectement par l'événement initial.

Le **graphe des objets** représente graphiquement le système sous la forme d'un enchaînement de processus, de messages et de synchronisations.

Ces deux graphiques diffèrent également selon les informations qu'ils font apparaître. Dans le premier, sont juxtaposées essentiellement des durées inter-événements, alors que dans le deuxième, ces durées sont agrégées par objet (par exemple, la durée d'attente d'un processus).

L'analyse de ces deux graphiques se fera essentiellement sur papier étant donné leur taille qui risque d'être importante. Pour une interprétation interactive, on utilisera plutôt la **représentation animée**, qui recrée sur la forme d'un dessin animé la réaction du système d'information à un événement initial.

IV.1.2. Graphe des événements

La description de cet outil comprend le détail des **Informations** affichées, une présentation des **fonctionnalités** offertes à l'utilisateur ainsi qu'un **exemple** à titre d'illustration.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent, soit sous la forme d'un texte, dans une **partie commentaire**, soit sous une forme semi-graphique, dans la **partie graphique** de l'outil.

1.1. Partie commentaire

Comme illustré à la figure IV.3., cet espace contient les informations relatives

- à la **simulation** (le titre, la date d'exécution)
- au **comportement** représenté (le type de l'événement initial, sa date de survenance et le type de l'événement terminal s'il s'agit d'une transaction)

1.2. Partie graphique

Le support graphique de l'outil est le dessin d'un graphe dont un **sommet** est un événement, et un **arc** est une relation de précédence (cf III.2.1.) entre deux événements.

Sur ce dessin, sont juxtaposées des informations relatives à la survenance d'événements.

Pour éviter d'agrandir inutilement le graphique lorsqu'il y a **duplication**, c'est-à-dire lorsqu'un événement est prédécesseur de plusieurs événements d'un même type (dans le cas d'un déclenchement ou d'une contribution multiple), on ne représente que son premier et son dernier successeur, en affichant cependant le nombre de successeurs.

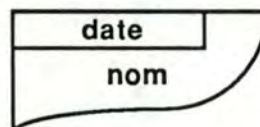
1.2.1. Représentation d'un sommet

On peut classer les sommets représentant un événement en trois ensembles, suivant qu'ils concernent

- un **message**
- un **processus**
- une **synchronisation**

Événement concernant un message

On représente la génération d'un message et sa date de survenance par :

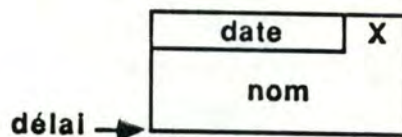


où

- le **nom** est le nom du message
- la **date** est la date de survenance de la génération

Événements concernant un processus

On représente un changement d'état d'un processus et les informations qui y sont associées par :

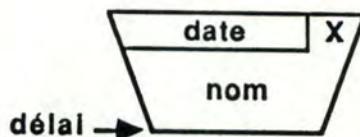


- le **nom** est le nom du processus
- la **date** est la date de survenance de l'événement
- le **délai** est le délai de l'événement par rapport à l'événement initial (cf. III.2.3)
- le **X** est une abréviation du type de l'événement, c'est-à-dire

D pour le déclenchement d'un processus
 A pour l'activation "
 I pour l'interruption "
 R pour la réactivation "
 T pour la terminaison "

Événements concernant une synchronisation

On représente un événement concernant une synchronisation et les informations qui y sont associées par :

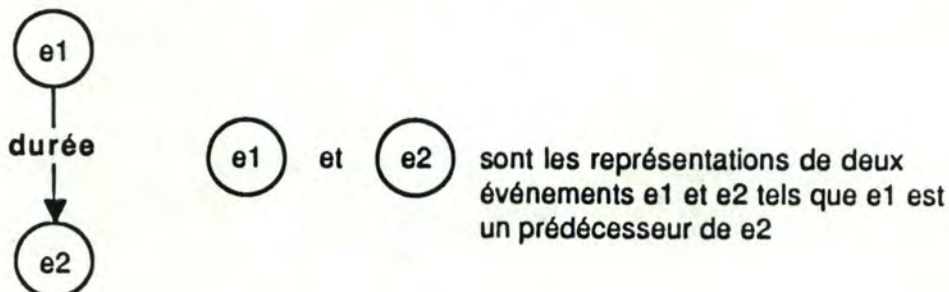


- le **nom** est le nom de la synchronisation
- la **date** est la date de survenance de l'événement
- le **délai** est le délai de l'événement par rapport à l'événement initial (cf. III.2.3)
- le **X** est une abréviation du type de l'événement, c'est-à-dire

C pour la contribution à une synchronisation
 R pour la réalisation d'une "
 S pour la suppression de contribution(s)

1.2.2. Représentation d'un arc

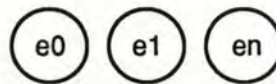
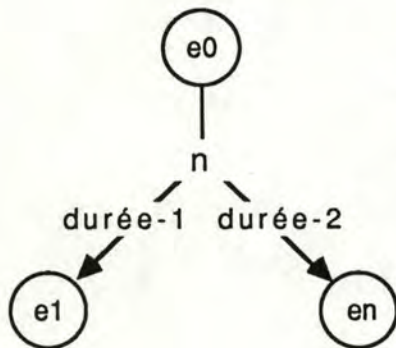
On représente une **relation de précedence** entre deux événements e1 et e2, et la **durée Inter-événements** entre ces deux événements par :



où - la **durée** est la durée inter-événements entre e1 et e2

1.2.3. Représentation d'une duplication

On représente une duplication entre un événement e_0 et ses successeurs e_1, \dots , en par :



sont les représentations des événements e_0, e_1, e_n tels que e_1 est le premier successeur de e_0 et e_n est son dernier successeur.

où

- le n indique le nombre d'objets dupliqués
- la durée-1 est la durée inter-événements entre e_0 et e_1
- la durée-2 est la durée inter-événements entre e_0 et e_n

2. FONCTIONNALITES

2.1. Identification de la partie du système d'information à représenter

L'outil décrit ci-dessus permet de représenter la réaction du système d'information à un événement initial ou une transaction.

Dans le premier cas, l'utilisateur détermine l'événement initial en spécifiant son type et en donnant une date. L'événement initial retenu est le premier événement du type choisi survenu à partir de la date donnée.

Dans le second cas, en plus de l'événement initial, l'utilisateur doit spécifier le type de l'événement terminal.

2.2. Choix d'un niveau de trace

Pour permettre à l'utilisateur d'adapter la représentation graphique du comportement au niveau d'analyse qu'il désire faire, trois niveaux de trace lui sont proposés :

-un premier niveau qui reprend les événements suivants :

- la génération d'un message
- la terminaison d'un processus
- la réalisation d'une synchronisation

Le graphe des événements avec ce niveau de trace est illustré à la figure IV.3.

-un deuxième niveau qui reprend les événements suivants :

- la génération d'un message
- le déclenchement d'un processus
- la terminaison "
- la contribution à une synchronisation
- la réalisation d'une synchronisation

-un troisième niveau qui reprend tous les événements possibles, c'est-à-dire :

- la génération d'un message
- le déclenchement d'un processus

- l'activation "
- l'interruption "
- la réactivation "
- la terminaison "
- la contribution à une synchronisation
- la réalisation d'une synchronisation
- la suppression de contribution(s)

2.3. Gestion de fenêtre

La taille d'un graphique peut être largement supérieure à la taille de l'écran. Il est donc nécessaire de disposer des fonctions classiques de gestion de fenêtres (scrolling, zooming) pour pouvoir visualiser interactivement la totalité du graphique.

2.4. Impression

Il est utile de pouvoir analyser les informations offertes par cet outil à partir d'un support sur papier, notamment lorsque le graphique est particulièrement grand.

2.5. Aide

Cette fonction affiche la signification des mesures juxtaposées sur le graphique.

2.6. Afficher/Effacer la partie commentaire

Lorsqu'on représente graphiquement le comportement du système, sont affichées également les informations de la partie commentaire. Si l'utilisateur désire récupérer l'espace occupé par ces données, il a la possibilité de les faire disparaître et de les rappeler ultérieurement.

3. EXEMPLE

- Simulation Gestion Prêt exécutée le 10 Juillet 1987 à 15h57m59s
- Typ. Evén. Initial gén. Demande de prêt Typ. Evén. Terminal term. Acceptation du prêt
- Gén. Demande de prêt survenue à 1d8h30m

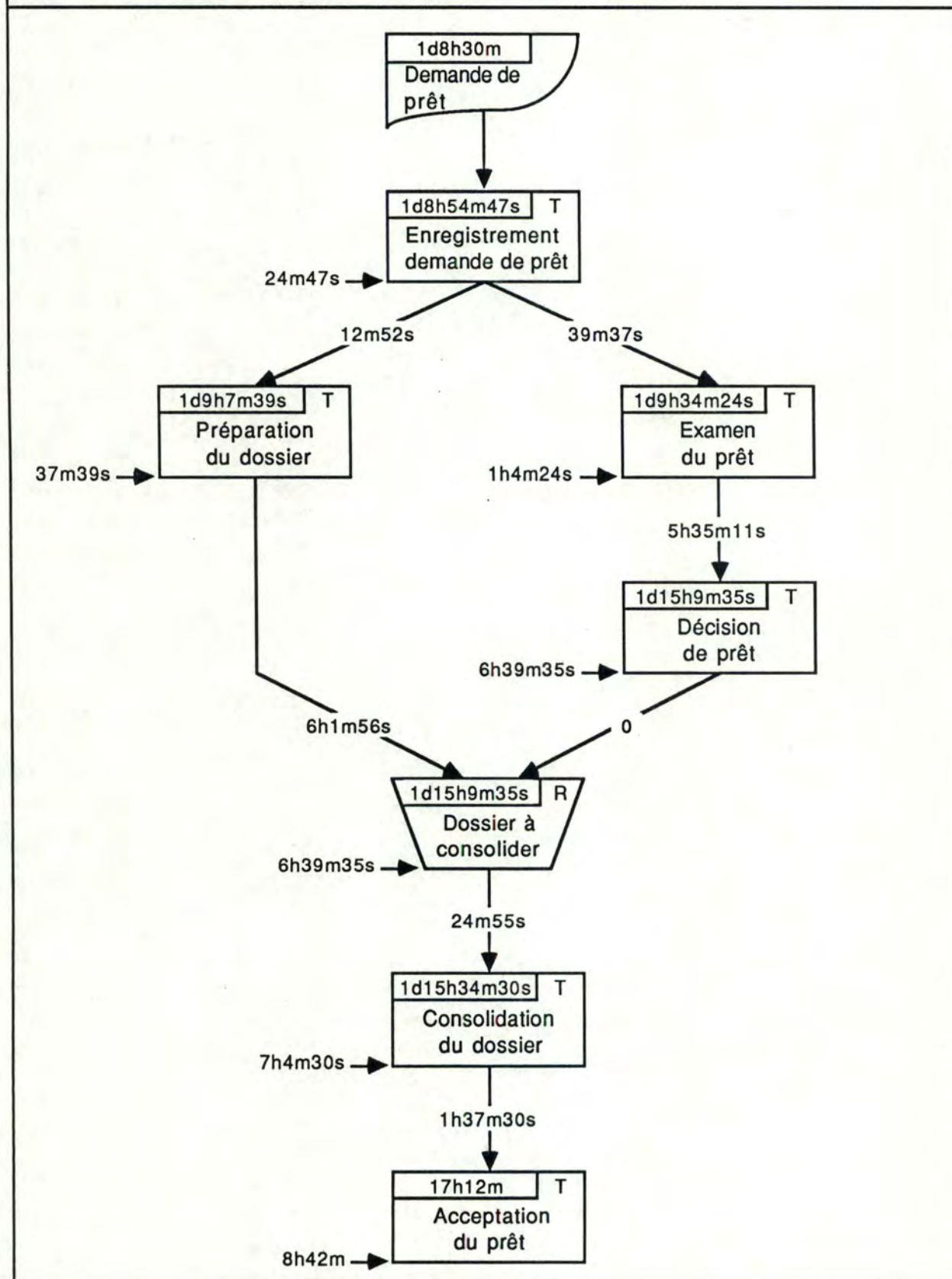


figure IV.3 Graphe des événements

IV.1.3. Graphe des objets

La description de cet outil comprend le détail des informations affichées, une présentation des fonctionnalités offertes à l'utilisateur ainsi qu'un exemple à titre d'illustration.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent soit, sous la forme d'un texte, dans une partie commentaire, soit sous une forme semi-graphique, dans la partie graphique de l'outil.

1.1. Partie commentaire

Comme illustré à la figure IV.4., cet espace contient les informations relatives

- à la simulation (le titre, la date d'exécution)
- au comportement représenté (le type de l'événement initial, sa date de survenance, et le type de l'événement terminal s'il s'agit d'une transaction).

1.2. Partie graphique

Le support graphique de l'outil est le dessin d'un graphe dont un sommet est un objet qui regroupe tous les événements le concernant et un arc exprime une relation de dynamique entre un événement d'un objet et un événement d'un autre objet.

Sur ce dessin sont juxtaposées des informations relatives au comportement des processus et des synchronisations.

Pour éviter d'agrandir inutilement le graphique lorsqu'il y a duplication, on ne représente que le premier et le dernier processus déclenché lors d'un déclenchement multiple et la première et la dernière contribution dans le cas d'une contribution multiple, en affichant cependant le nombre de ces processus ou de ces contributions.

1.2.1. Représentation d'un sommet

On peut classer les sommets en trois ensembles suivant qu'ils concernent

- un message
- un processus
- une synchronisation

Représentation d'un message

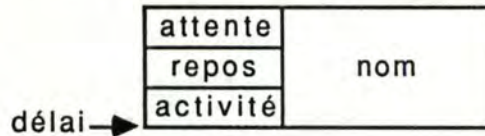
On représente un message par :



où - le nom est le nom du message

Représentation d'un processus

On représente un processus et les informations qui y sont associés par :



où

- le nom est le nom du processus
- l'attente est la durée d'attente du "
- le repos est la durée de repos du "
- l'activité est la durée d'activité du "
- le délai est le délai de la terminaison du processus par rapport à l'événement initial

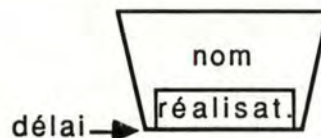
La durée d'activité, d'attente et de repos sont calculées sur base des informations se trouvant dans la trace de la simulation (cf. III.3.1.). La durée d'activité représente le total des durées entre les événements suivants :

- la terminaison et la (ré)activation
- l'interruption et la (ré)activation

La durée d'attente (resp. durée de repos) d'un processus est obtenue en faisant la somme des durées d'attente (resp. durée de repos) avant (ré)activation.

Représentation d'une synchronisation

On représente une synchronisation et les informations qui y sont associées par :



où

- le nom est le nom de la synchronisation
- la réalisation est la durée de réalisation de la synchronisation
- le délai est le délai de la réalisation de la synchronisation par rapport à l'événement initial

1.2.2. Représentation d'un arc

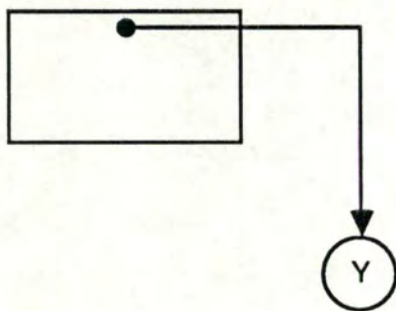
On représente une relation de dynamique entre un événement d'un objet X et un événement d'un objet Y par :



X et Y sont les représentations de deux objets X et Y

par :

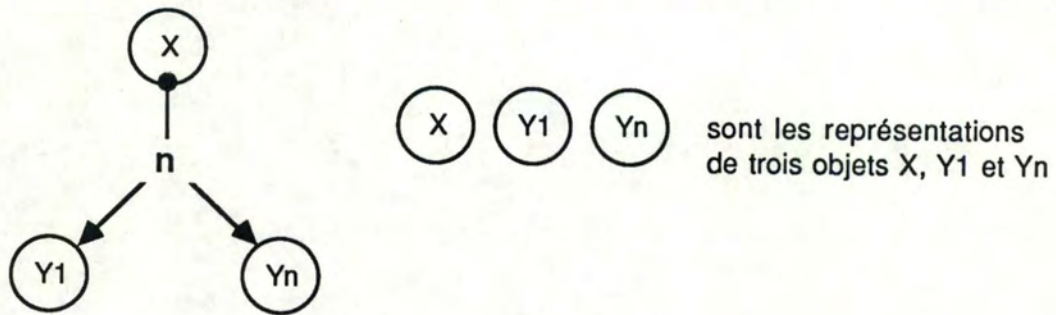
Toutefois, si l'activation est l'origine d'une relation de dynamique, celle-ci est représentée



Y est la représentation d'un objet Y

1.2.3. Représentation d'une duplication

On représente une duplication entre un objet X et les objets dupliqués Y1, ..., Yn par



où - le n indique le nombre d'objets dupliqués.

2. FONCTIONNALITES

(cf fonctionnalités du graphe des événements au point IV.2.2.)

3. **EXEMPLE**

- Simulation *Gestion Prêt* exécutée le 10 Juillet 1987 à 15h57m59s
- Typ. Evén. Initial *gén. Demande de prêt* Typ. Evén. Terminal *term. Acceptation du prêt*
- Gén. *Demande de prêt* survenue à 1d8h30m

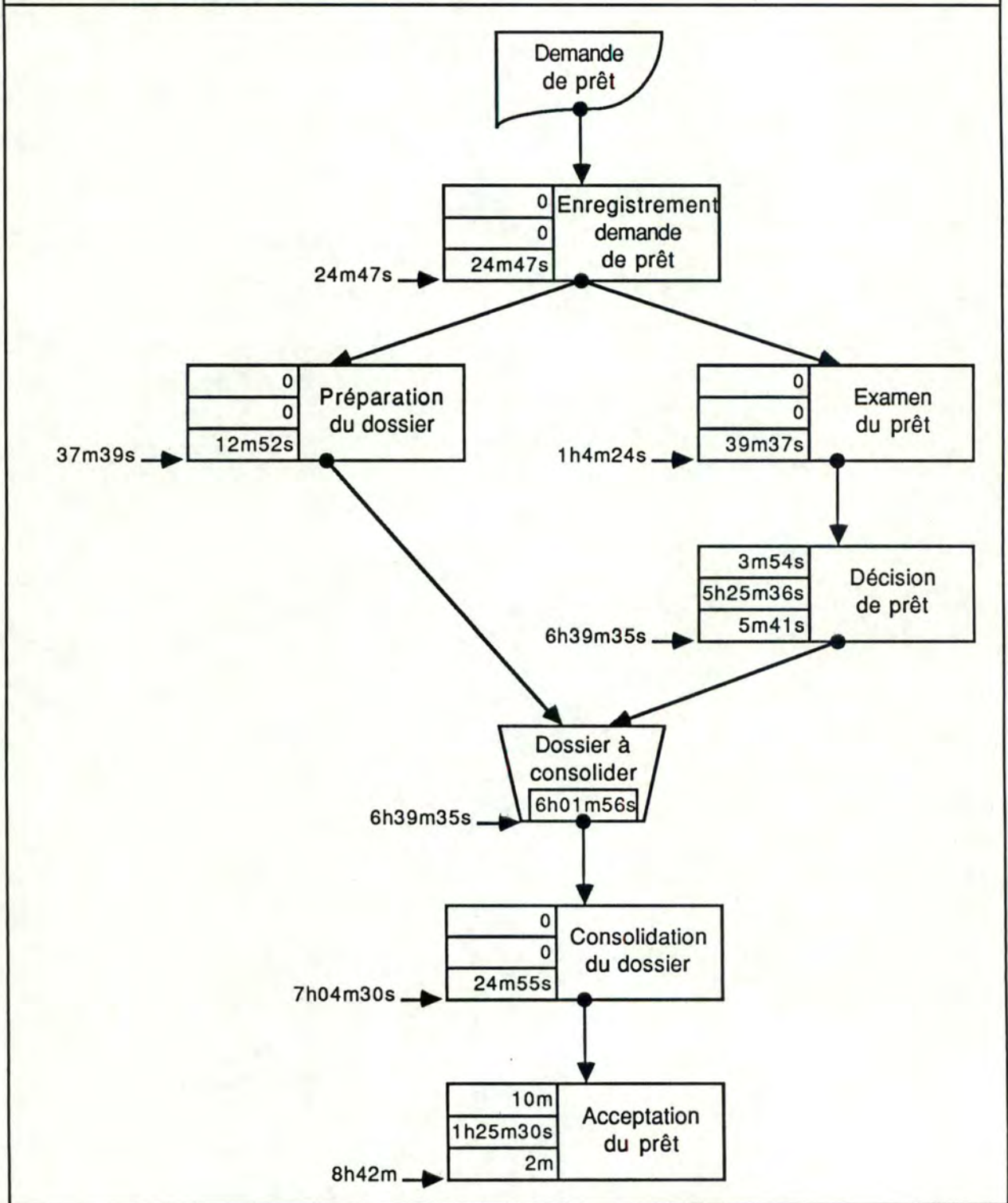


figure IV.4 Graphe des objets

IV.1.4. Animation

La description de cet outil comprend le détail de l'animation, une présentation des fonctionnalités offertes à l'utilisateur ainsi qu'un exemple à titre d'illustration.

1. ANIMATION

La représentation animée du comportement du système peut être vue comme une suite de photos prises lors de la survenance d'un événement qui est soit, un descendant de l'événement initial, si on a choisi de représenter la réaction du système suite à la survenance de cet événement, soit un événement d'une transaction.

La description de cette animation consiste à déterminer les composantes d'une photo et les changements d'une photo à une autre.

1.1. Composantes d'une photo

Comme l'illustre la figure IV.5, une photo est composée d'une partie graphique représentant l'état du système d'information à un instant donné, une horloge et une partie commentaire.

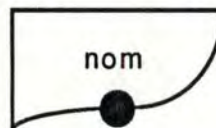
1.1.1. Représentation de l'état du système

L'état du système à un instant donné est représenté par les messages générés, l'état des processus, l'état des coordinations et les relations de dynamique entre ces objets.

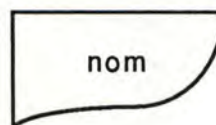
Message

Deux représentations d'un message sont envisagées suivant que la photo est prise :

-au moment de la génération



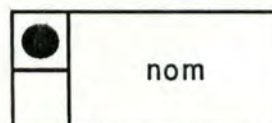
-après la génération



Processus

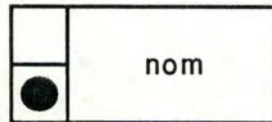
Quatre représentations d'un processus sont envisagées suivant que la photo est prise :

-au moment où le processus est en attente ou en repos

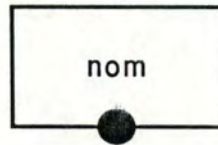


Nous n'avons pas pu faire la distinction entre ces deux états car le passage de l'un vers l'autre n'est pas considéré comme un événement par la simulation.

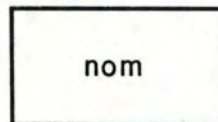
-au moment où le processus est en activité



-au moment où le processus est terminé



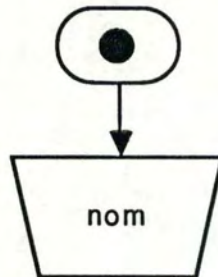
-après la terminaison du processus



Coordination

Quatre représentations d'une coordination sont envisagées suivant que la photo est prise :

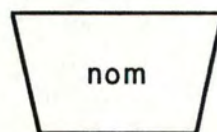
-au moment où il y a une contribution à un point de synchronisation (1)



-au moment où la synchronisation est réalisée



-après la réalisation de la synchronisation



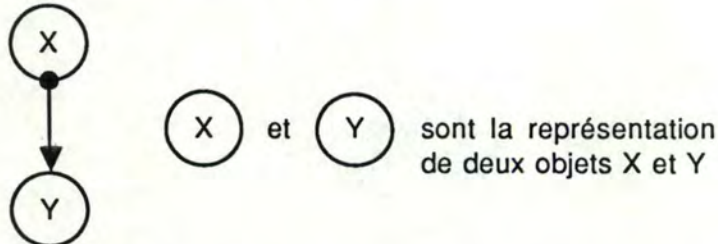
(1) On parle de contribution à une synchronisation lorsque la réalisation a eu lieu; mais au moment de la contribution, il est plus pertinent de parler de contribution à un type de synchronisation.

-après la suppression d'une contribution

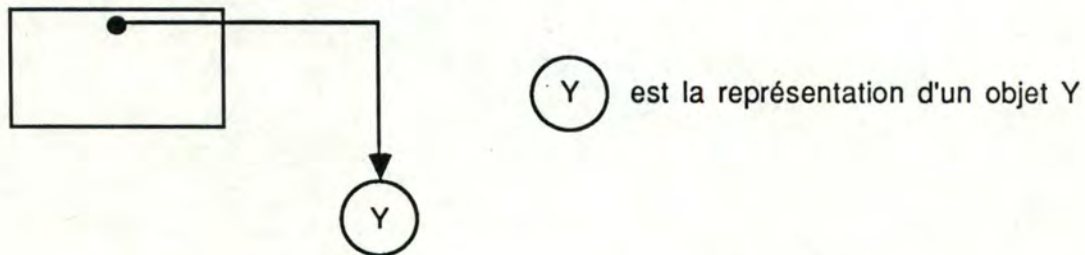


Relation de dynamique

On représente une relation de dynamique par



Toutefois, si l'activation est l'origine d'une relation de dynamique, celle-ci est représentée par :



Pour éviter d'agrandir inutilement le graphique lors d'un enchaînement multiple, une photo ne reprendra que le premier et le dernier déclenchements ou la première et la dernière réalisations auxquelles la contribution a participé.

1.1.2. Horloge

Une horloge indique l'heure à laquelle la photo a été prise.

1.1.3. Partie commentaire

Comme illustré à la figure IV.5., cet espace contient les informations relatives

- à la simulation (le titre, la date d'exécution)
- au comportement représenté (le type de l'événement initial, sa date de survenance, et le type de l'événement terminal s'il s'agit d'une transaction).

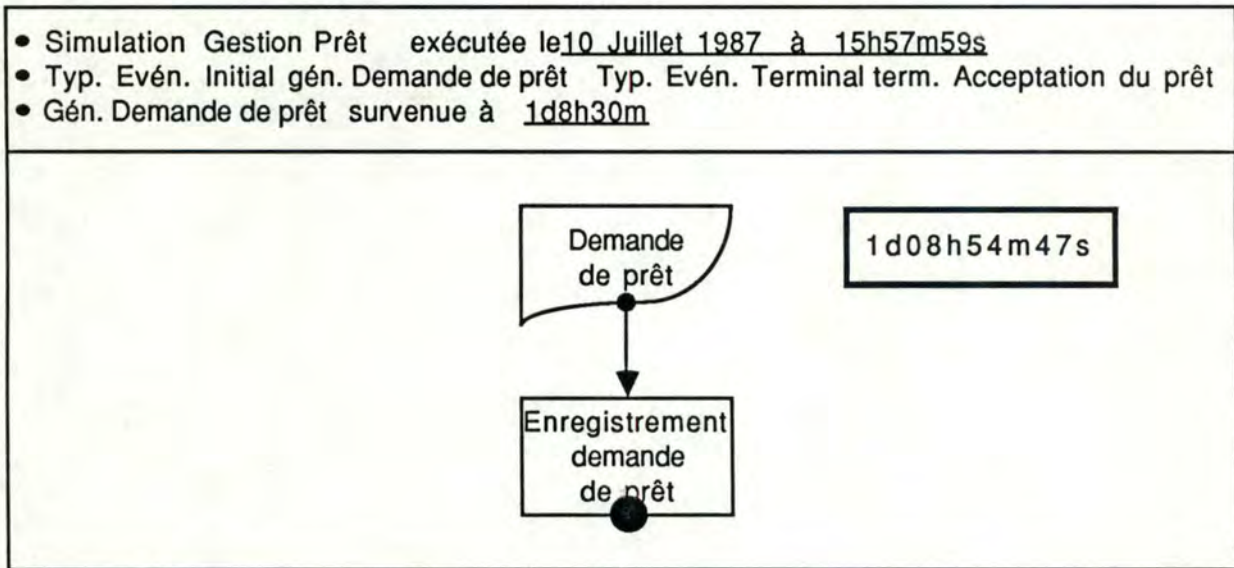


figure IV.5. Exemple de photo

1.2. Changement de photo

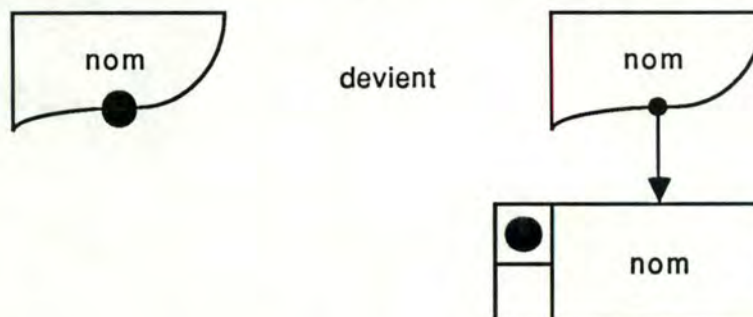
Une nouvelle photo est affichée lors de la survenance d'un événement appartenant au comportement représenté.

Créer une nouvelle photo consiste à

- changer l'horloge pour qu'elle indique le moment où la nouvelle photo a été prise. Notons que ce changement ne se fera pas brutalement mais reflétera le laps de temps séparant la survenance de deux événements consécutifs
- prendre la photo précédente et effectuer sur cette photo les changements décrits ci-dessous, qui varient suivant l'événement courant et son prédécesseur (cf.III.2.1.)

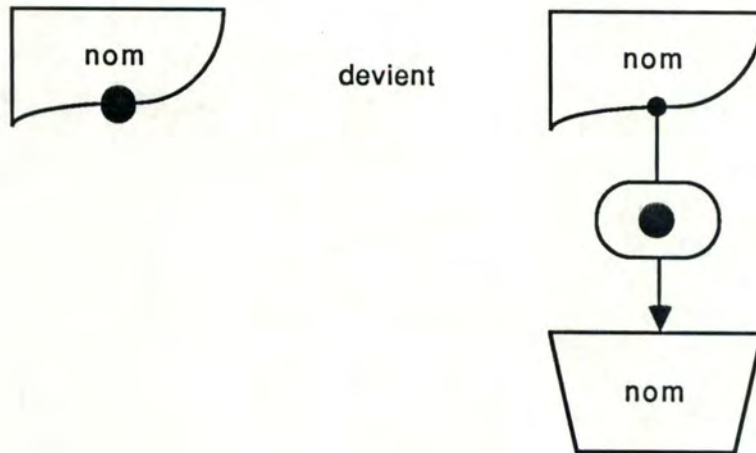
-Le déclenchement d'un processus suite à la génération d'un message

- crée un objet processus
- crée une relation de précédence du message vers le processus
- déplace le jeton du bord inférieur du message vers la case d'attente du processus



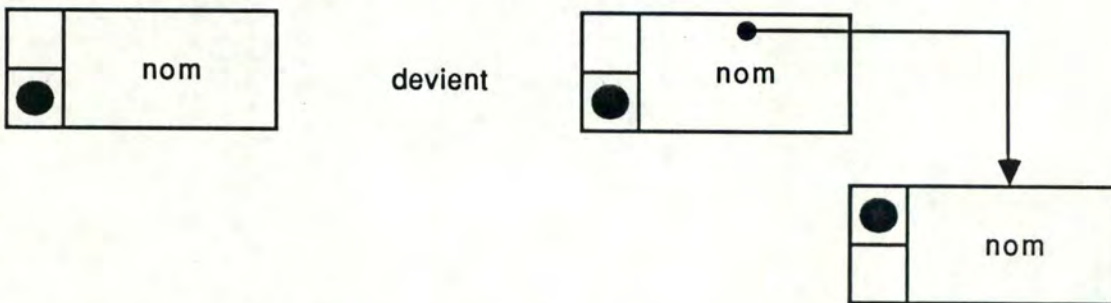
-La contribution à un point de synchronisation suite à la génération d'un message

- crée un objet contribution
- crée une relation de précédence du message vers la contribution
- déplace le jeton du bord inférieur du message vers le milieu de la contribution



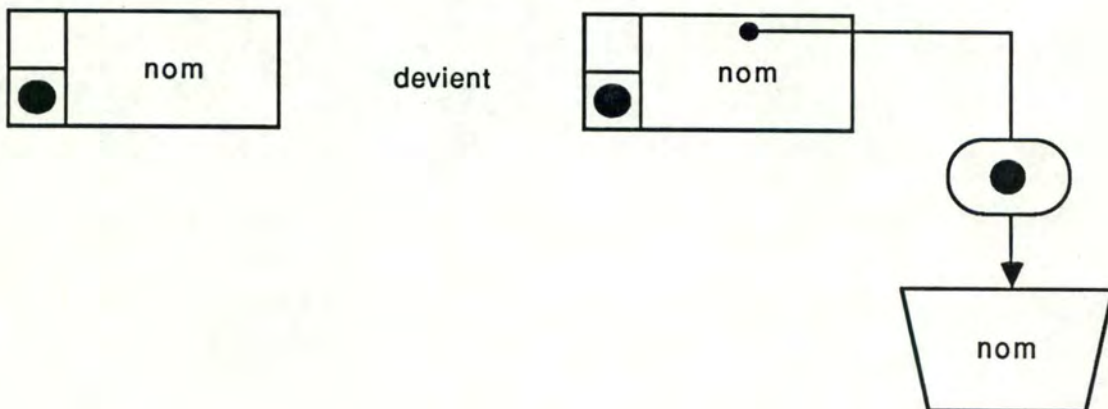
-Le déclenchement d'un processus suite à l'activation d'un autre processus

- crée un objet processus
- crée une relation de précédence du processus déclencheur vers le processus déclenché
- place un jeton dans la case d'attente du processus déclenché



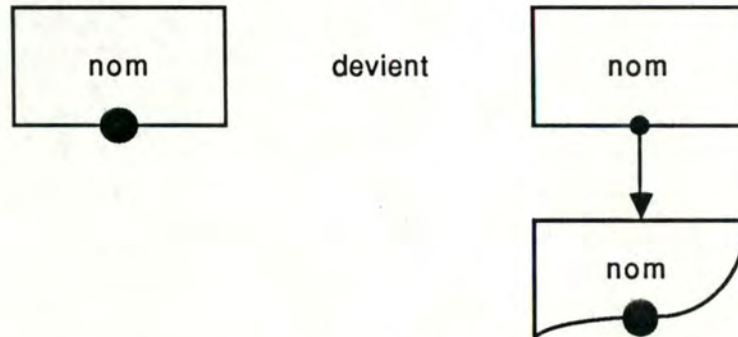
-La contribution à un point de synchronisation suite à l'activation d'un processus

- crée un objet contribution
- crée une relation de précédence du processus vers la contribution
- place un jeton au milieu de l'objet contribution

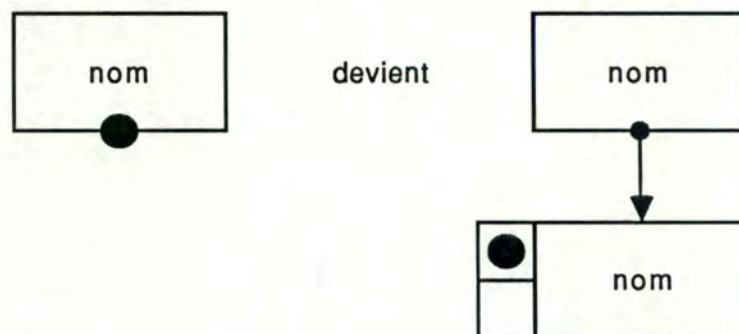


-La génération d'un message

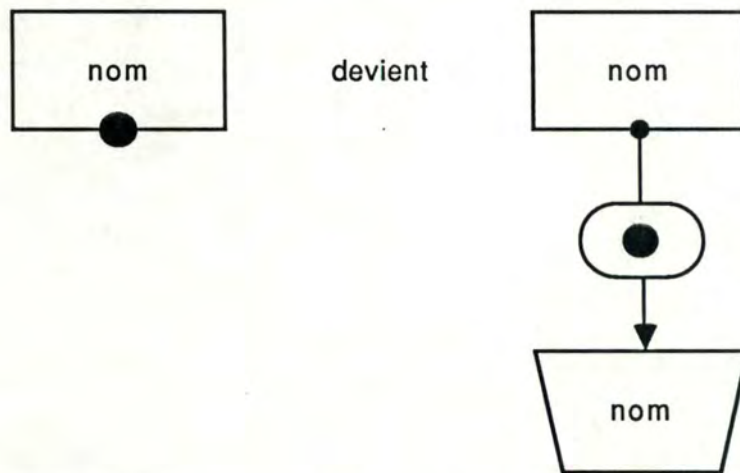
- crée un objet message
- crée une relation de précédence du processus vers le message
- déplace le jeton du bord inférieur du processus vers le bord inférieur du message

**-Le déclenchement d'un processus suite à la terminaison d'un autre processus**

- crée un objet processus
- crée une relation de précédence du processus déclencheur vers le processus déclenché
- déplace le jeton du bord inférieur du processus vers la case d'attente du processus

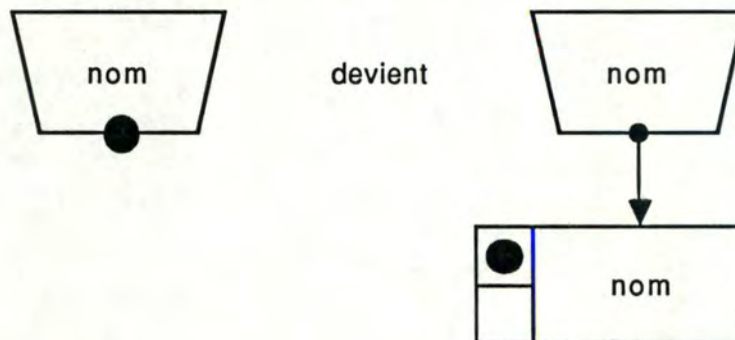
**-La contribution d'une terminaison d'un processus à un point de synchronisation**

- crée un objet contribution
- crée une relation de précédence du processus vers la contribution
- déplace le jeton du bord inférieur du processus vers le milieu de la contribution



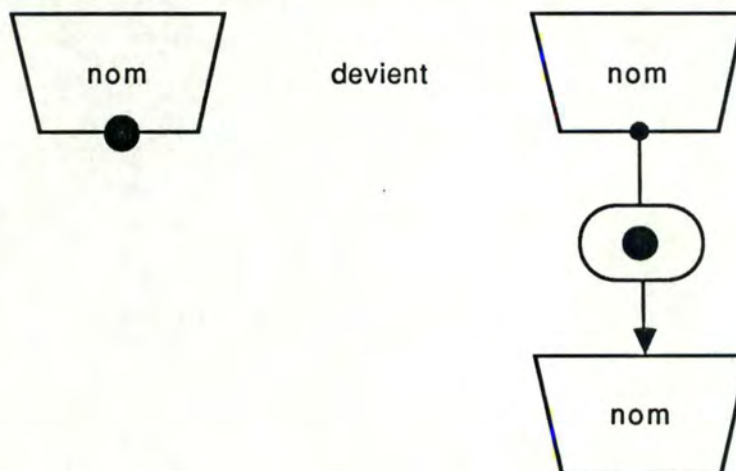
-Le déclenchement d'un processus suite à la réalisation d'une synchronisation

- crée un objet processus
- crée une relation de précédence de la synchronisation vers la processus
- déplace le jeton du bord inférieur de la synchronisation vers la case d'attente du processus



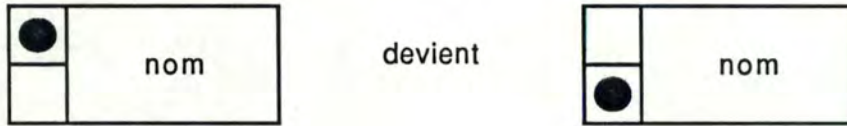
-La contribution d'une réalisation de synchronisation à une autre synchronisation

- crée un objet contribution
- crée une relation de précédence du point de synchronisation vers la contribution
- déplace le jeton du bord inférieur de la synchronisation vers le milieu de l'objet contribution



-L'activation ou la réactivation d'un processus

-déplace le jeton de la case d'attente vers la case d'activation



-La terminaison d'un processus

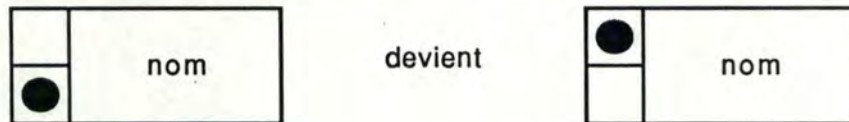
-supprime les cases d'attente et d'activité du processus

-déplace le jeton sur le bord inférieur du processus



-L'interruption d'un processus

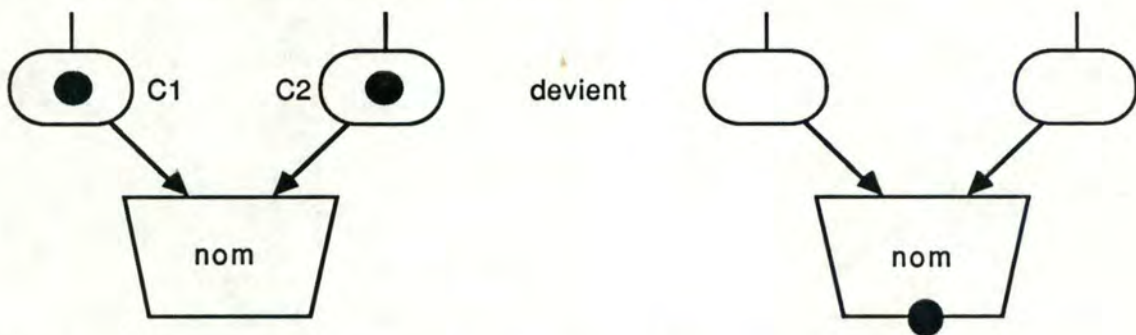
-déplace le jeton de la case d'activation vers la case d'attente du processus



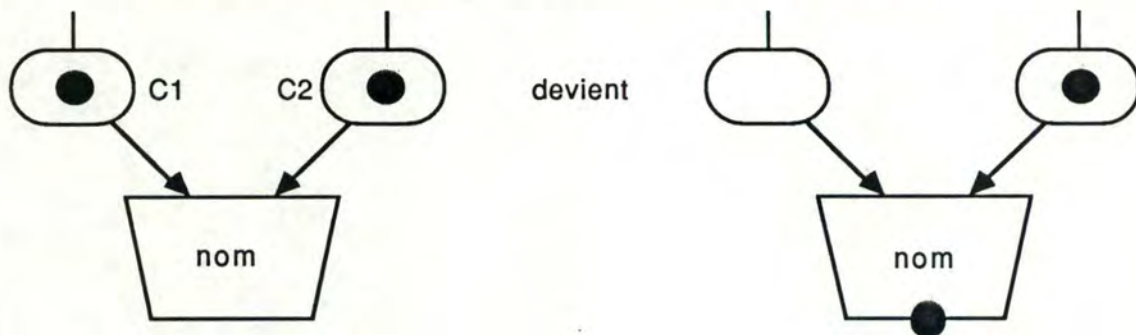
-La réalisation d'une synchronisation

-enlève le jeton des contributions simples qui ont participé à la synchronisation

-place un jeton sur le bord inférieur de la synchronisation



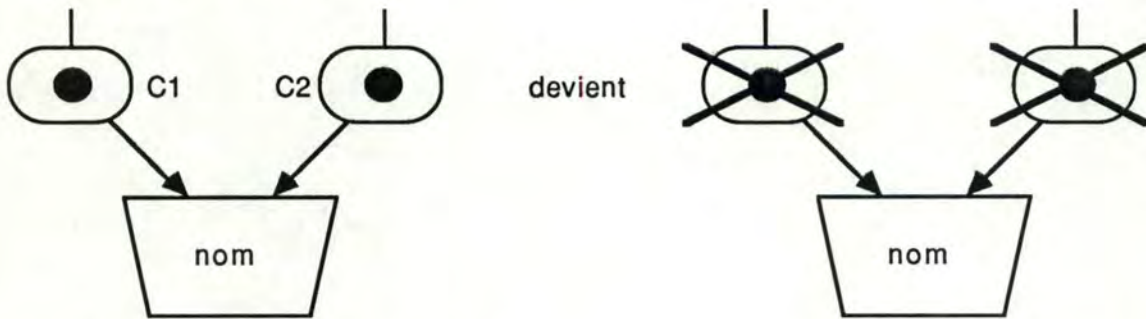
où C1 et C2 sont deux contributions simples à une même synchronisation



où C1 est une contribution simple, et C2 est une contribution multiple

-La suppression de contributions

-barre les contributions tuées



où C1 et C2 sont les contributions tuées

2. FONCTIONNALITES

2.1. Identification de la partie du système d'information à représenter

Cf. fonctionnalité 2.1. graphe des événements.

2.2. Sélection d'un jeton

Nous avons vu qu'un jeton peut être déplacé ou regroupé, mais il peut être également décomposé lors d'un enchaînement multiple, d'un éclatement, ou de l'activation d'un processus. Dans ce cas, pour garder un interface d'une complexité raisonnable, l'utilisateur devra sélectionner un successeur au dernier événement représenté.

Lors d'un déclenchement multiple, on demandera à l'utilisateur de sélectionner le premier ou le dernier déclenchement

Lors d'une contribution multiple, on demandera à l'utilisateur de sélectionner la première ou la dernière réalisation à laquelle la contribution a participé.

Dans les autres cas, on demandera à l'utilisateur de sélectionner un successeur parmi tous les successeurs possibles de l'événement courant représenté par le jeton suivi.

2.3. Contrôle du déroulement de l'animation

Il est important que l'utilisateur puisse assimiler le déplacement d'un jeton sélectionné. Dès lors, après chaque photo visualisant le déplacement du jeton choisi, c'est l'utilisateur qui décide quand il faut passer à la photo suivante.

2.4. Gestion de fenêtre

Cf. fonctionnalité 2.3. graphe des événements.

2.5. Impression

Cf. fonctionnalité 2.4. graphe des événements.

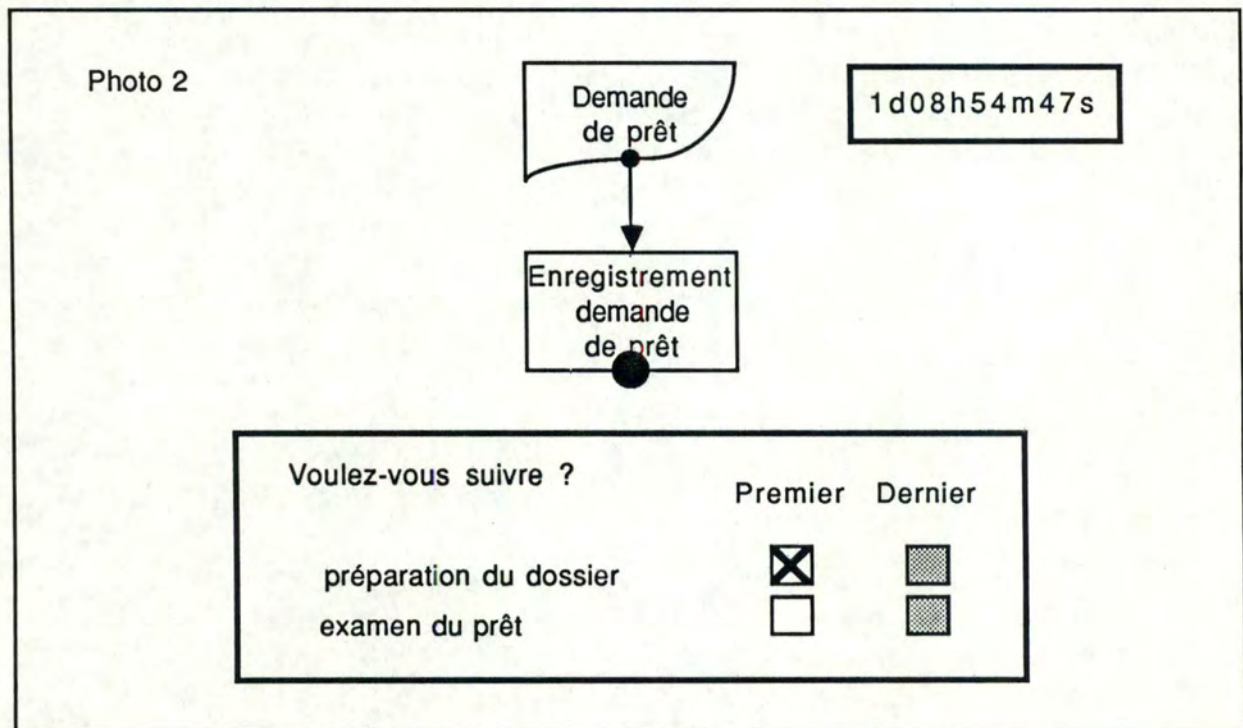
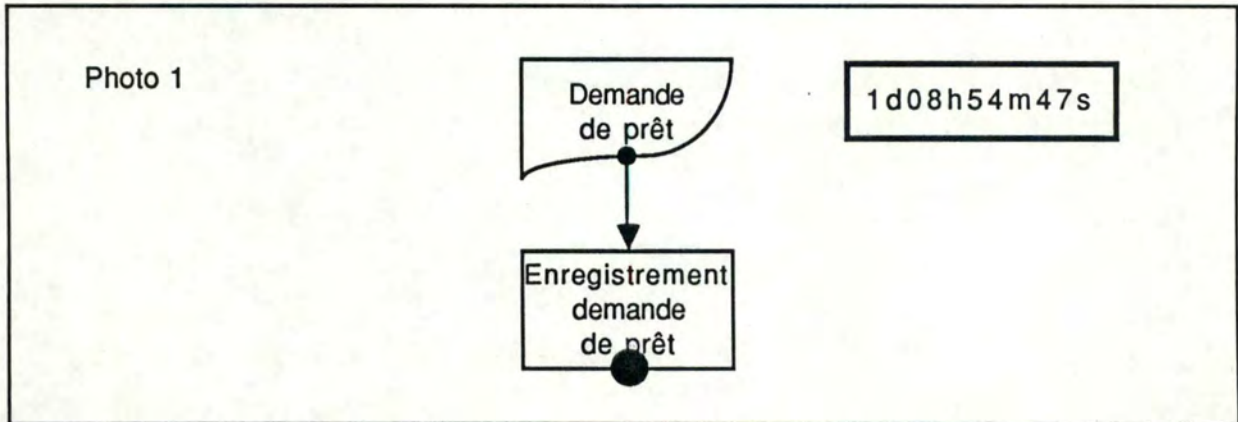
2.6. Aide

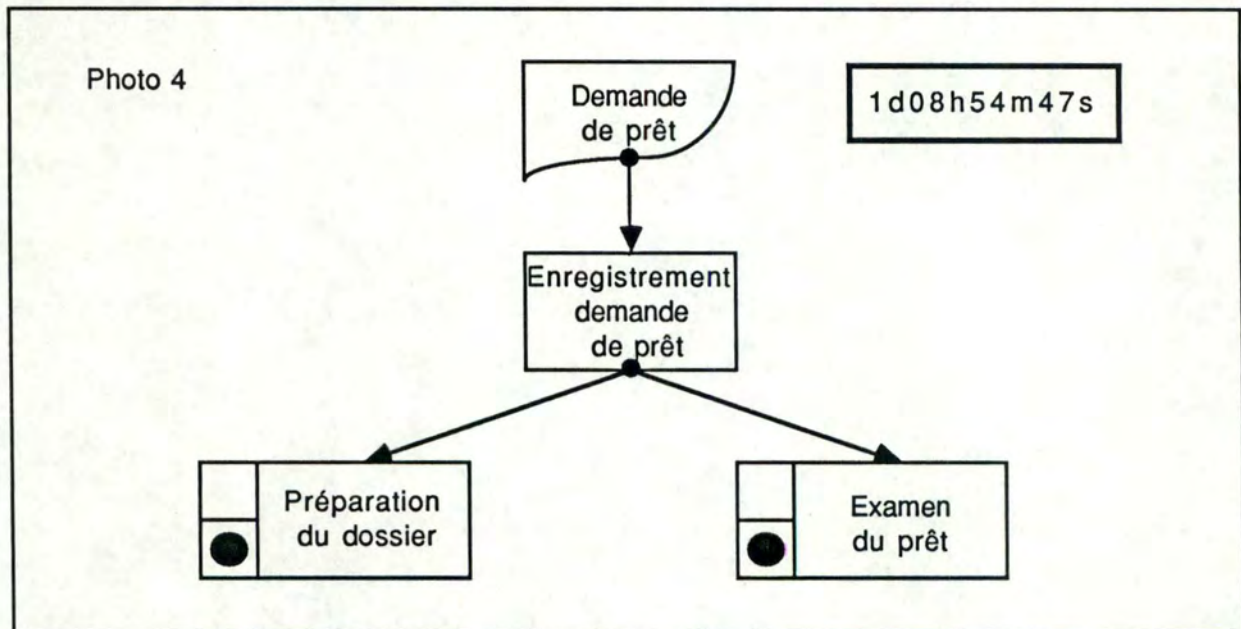
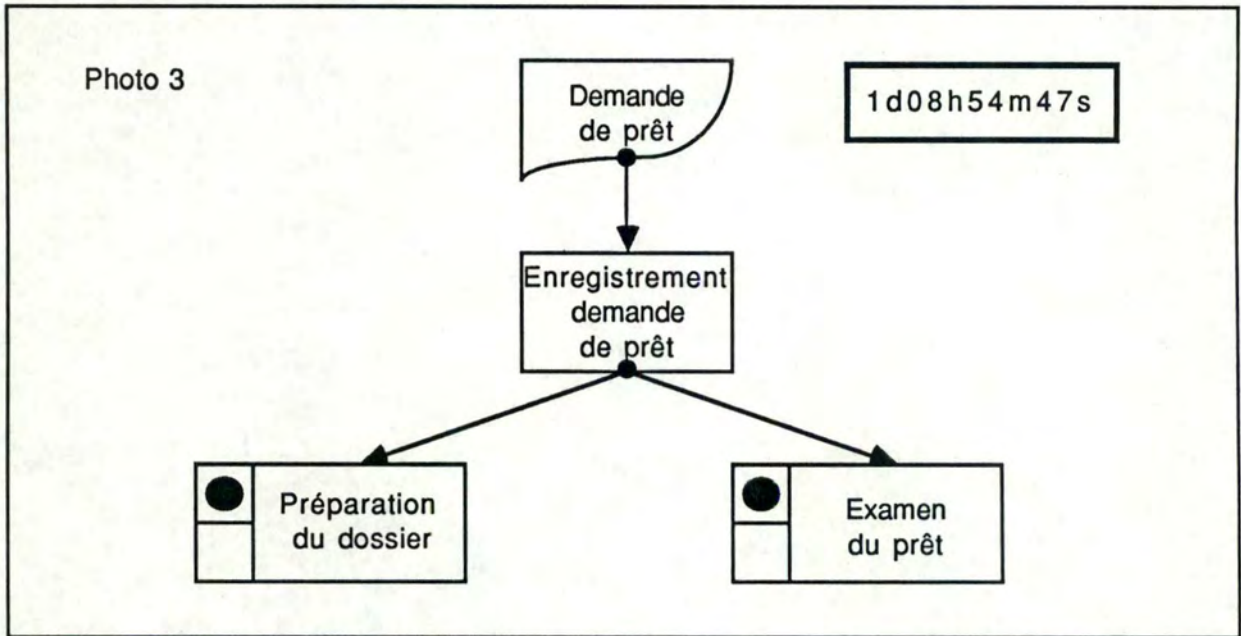
Cf. fonctionnalité 2.5. graphe des événements.

2.7. Afficher/Effacer la partie commentaire

Cf. fonctionnalité 2.6. graphe des événements.

3. EXEMPLE





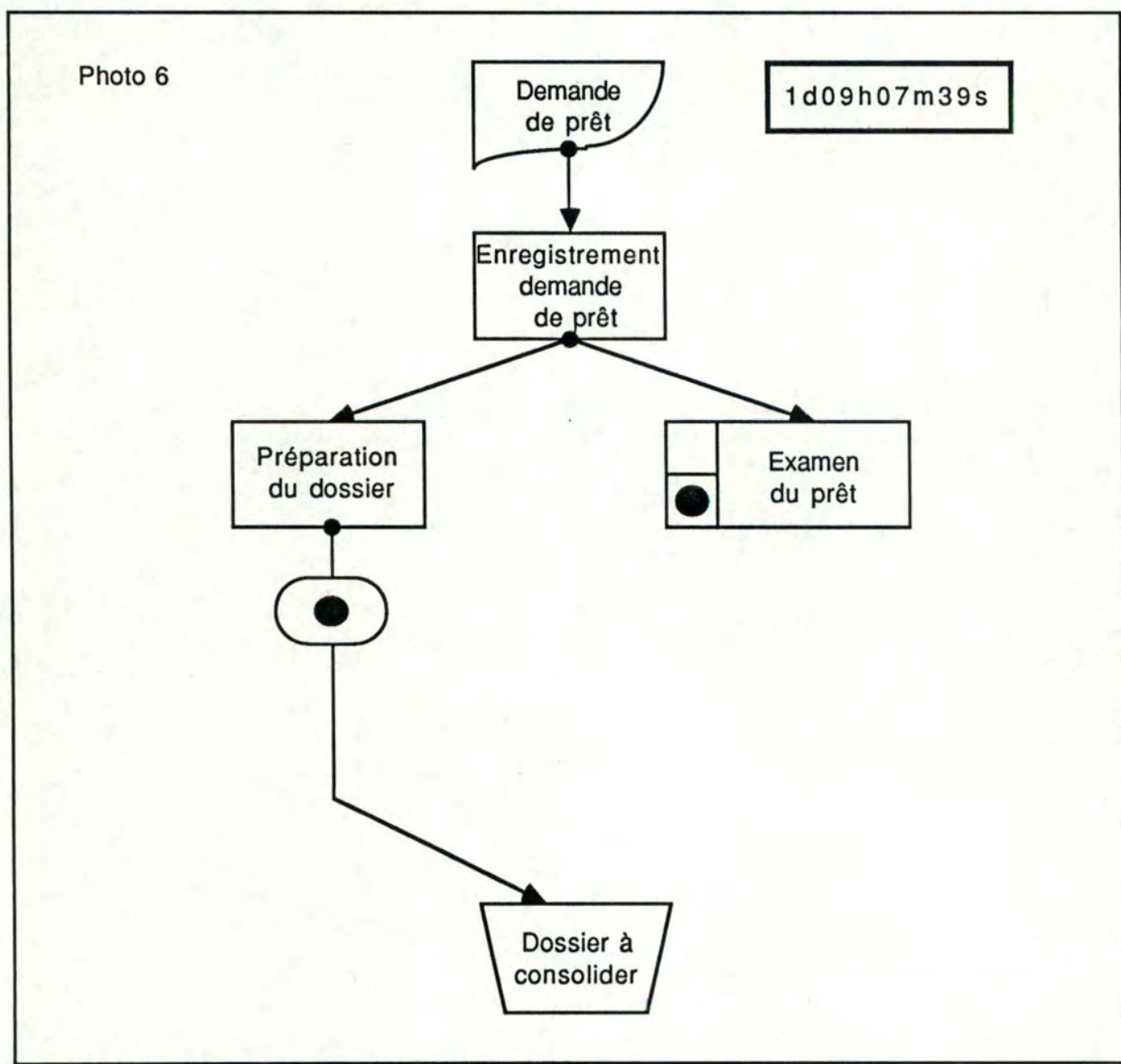
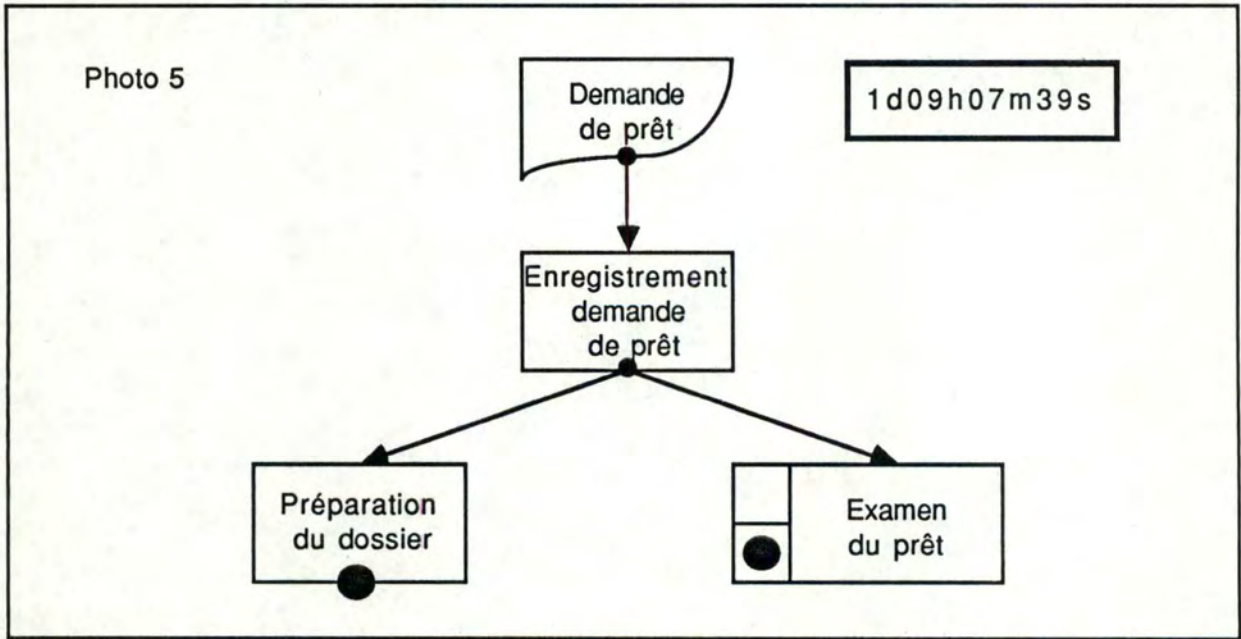
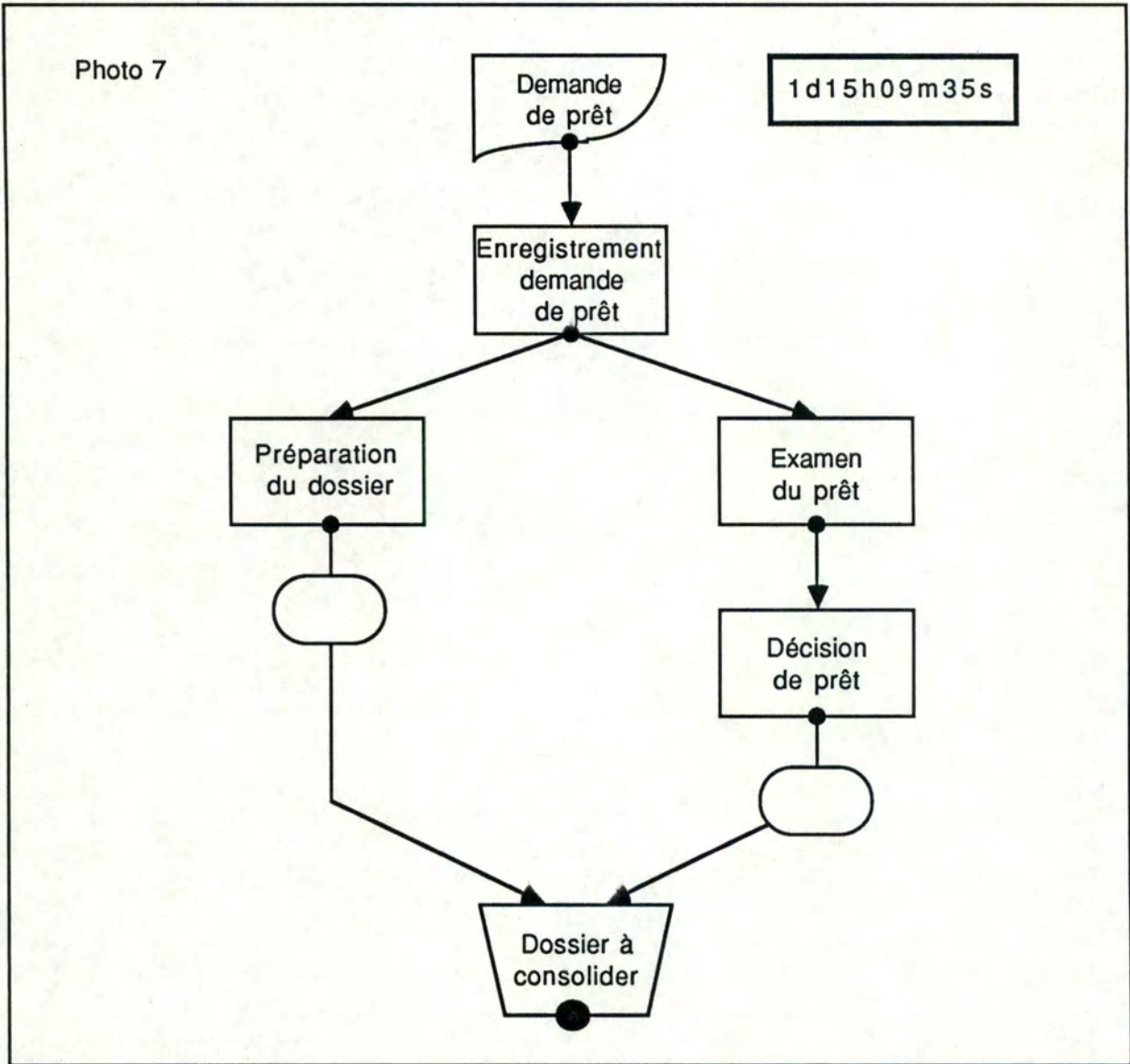


Photo 7



IV.2. COMPORTEMENT D'UN POINT DE SYNCHRONISATION

IV.2.1. Objectifs

Un point de synchronisation est "un mécanisme de coordination d'événements : l'occurrence d'un événement d'un type donné (appelé réalisation du point de synchronisation) dépend de l'occurrence de $n(\geq 2)$ événements différents; cette dépendance est exprimée par une condition de synchronisation"[DSL-SIM, 86].

Au niveau du simulateur, cette coordination est implémentée en mémorisant, pour chaque point de synchronisation défini dans le modèle, sa condition de synchronisation.

Lorsqu'un événement contribue à un point de synchronisation, il vient se placer dans la file d'attente associée au type de contribution.

Il s'agit ensuite d'évaluer la condition de synchronisation. Si elle est vérifiée, il y a synchronisation et les contributions simples participant à la réalisation sont extraites de leur file d'attente.

Une contribution à un point de synchronisation peut être supprimée de sa file d'attente sans qu'elle ne participe à une synchronisation. Dans ce cas, la contribution est dite tuée. Cette suppression de contribution survient dans trois cas :

- une contribution mémorisable qui est arrivée à son terme,
- une contribution mémorisable jusqu'à la survenance d'un événement et cet événement est survenu (1),
- une contribution non-mémorisable qui ne provoque pas de réalisation.

Suite à cette description du fonctionnement du point de synchronisation, il apparaît que la synchronisation, en tant qu'objet, n'existe qu'au moment de la réalisation. Il n'est en effet pas toujours possible de connaître, au moment de la contribution, la synchronisation à laquelle l'événement participe (par exemple dans le cas d'une contribution multiple).

L'outil que nous proposons a pour objectifs de permettre à l'utilisateur, d'une part, de mieux appréhender le mécanisme de synchronisation que nous venons de décrire brièvement et d'autre part, de lui donner la possibilité de vérifier aisément si les synchronisations se déroulent conformément à ses attentes. Cet outil est d'autant plus intéressant qu'actuellement aucune mesure chronologique (ou périodique) concernant les points de synchronisations n'est produite par DSL-SIM.

Pour atteindre ce double objectif, l'outil recrée graphiquement, sous la forme d'une animation, le fonctionnement d'un point de synchronisation tel qu'il est géré par le simulateur.

IV.2.2. Description

La description de cet outil comprend le détail de l'animation, une présentation des fonctionnalités offertes à l'utilisateur ainsi qu'un exemple à titre d'illustration.

1. ANIMATION

La représentation animée du comportement d'un point de synchronisation peut être vue comme une suite de photos prises lors de la survenance d'un événement le concernant (2) .

(1) Dans ce cas il ya suppression de tous les événements ayant ce mode de contribution.

(2) La contribution à un point de synchronisation et la suppression de contribution(s) sont considérées comme des événements du système d'information (cf. IV.2.1).

La description de cette animation consiste à déterminer les composantes d'une photo, et les changements d'une photo à une autre.

1.1. Composantes d'une photo

Comme illustré à la figure IV.1, une photo est composée d'une partie graphique représentant l'état du point de synchronisation à un instant donné, une horloge et une partie commentaire.

1.1.1. Représentation de l'état du point de synchronisation

L'état du point de synchronisation à un instant donné est visualisé en reprenant des éléments de sa spécification, en montrant l'état de ses files d'attente et en plaçant dans une zone explicative le dernier événement qui s'est produit.

Spécification du point de synchronisation

Les éléments de la spécification d'un point de synchronisation apparaissant sur une photo sont les suivants :

- le nom du point de synchronisation,
- la condition de synchronisation,
- le nombre de contributions.

La spécification d'une condition de synchronisation est exprimée au moyen d'un langage formel dont la syntaxe emboîtée est décrite dans [DSL-SIM, 86]. Les opérandes de ce langage sont des références aux contributions à ce point de synchronisation (par exemple la terminaison d'un processus P1) ou des opérateurs, qui sont de deux types : binaires (ET, OU, MEME-ORIGINE) ou unaires (COMPTER, REGROUPER).

Le nombre de contributions, qui exprime le nombre de synchronisations auquel un événement peut contribuer, est juxtaposé à chaque type d'événement. Une contribution peut être unique ou multiple.

Il aurait été intéressant de reprendre sur la photo la durée de mémorisation qui constitue la deuxième modalité de contribution d'un événement. Toutefois, pour que l'animation respecte ses objectifs, il est primordial que l'utilisateur ait une vue la plus complète possible du comportement du point de synchronisation. Cela implique un choix des informations représentées pour réduire au maximum l'espace requis par celles-ci.

Files d'attente

A chaque type d'événement contribuant à un point de synchronisation est associée une file d'attente dont un élément, visualisé par un jeton, représente la contribution d'un événement susceptible de participer, ultérieurement, à une ou plusieurs synchronisations.

Zone explicative

Cette zone contient la nature de l'événement qui s'est produit, en l'occurrence, l'un des trois événements suivants :

- une contribution au point de synchronisation (1),
- une réalisation de synchronisation,
- une suppression de contribution(s).

(1) On parle de contribution à une synchronisation lorsque la réalisation a eu lieu, mais au moment de la contribution, il est plus pertinent de parler de contribution au point de synchronisation.

1.1.2. Horloge

Une horloge indique l'heure à laquelle la photo a été prise.

1.1.3. Partie commentaire

Comme illustré à la figure IV.1., cet espace contient les informations relatives à la simulation (le titre, la date d'exécution).

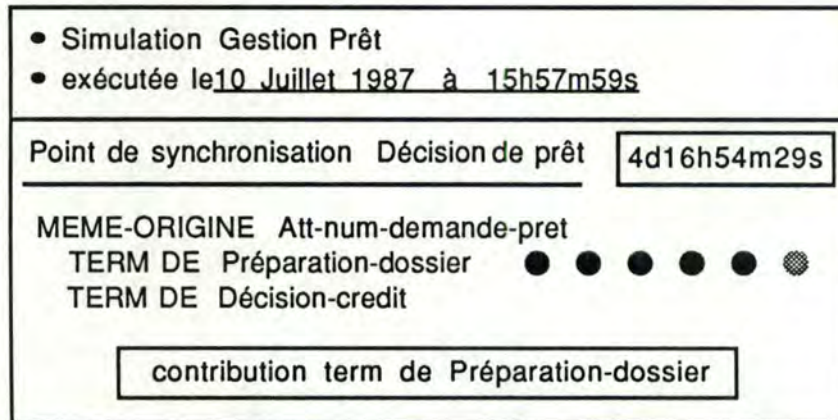


figure IV.1. Photo de l'animation d'un point de synchronisation

1.2. Changement de photo

Les photos sont prises à chaque survenance d'un événement concernant le point de synchronisation considéré.

Créer une nouvelle photo consiste à

-changer l'horloge pour qu'elle indique le moment où la nouvelle photo a été prise. Notons que ce changement ne se fera pas brutalement mais reflétera le laps de temps séparant la survenance de deux événements consécutifs.

-effectuer sur l'ancienne photo les changements, décrits ci-dessous, résultant de la survenance d'une contribution, d'une réalisation ou d'une suppression de contribution(s).

1.2.1. Contribution au point de synchronisation

La survenance d'une contribution au point de synchronisation est signifiée à l'utilisateur au moyen de la zone de dialogue et est représentée par un jeton placé dans la file d'attente qui lui est associée.

Pour dissocier cette nouvelle contribution des autres, le jeton représentant cet événement est affiché sur la photo dans une couleur particulière. Dès que l'utilisateur a décidé de passer à la photo suivante (cf. fonctionnalités), ce jeton reprend sa couleur normale.

1.2.2. Réalisation du point de synchronisation

La réalisation du point de synchronisation fait l'objet d'un commentaire dans la zone réservée à cet effet.

De plus, tous les jetons représentant les événements qui participent à cette réalisation sont affichés dans une autre couleur, avant d'être éventuellement effacés dès que l'utilisateur aura décidé de passer à la photo suivante. Cette technique permet d'identifier facilement les événements provoquant la réalisation du point de synchronisation.

Les contributions multiples ne sont pas effacées car par définition elles peuvent encore participer à d'autres synchronisations.

1.2.3. Suppression de contribution(s)

Comme pour les deux événements précédents, la suppression de contribution(s) est notifiée à l'utilisateur à l'aide de la zone de commentaire.

Le(s) jeton(s) représentant la (ou les) contribution(s) tuée(s) est barré avant d'être effacé(s) lorsque la photo suivante est affichée.

Dans la version actuelle de DSL-SIM, pour des raisons de performance, lorsqu'une contribution est mémorisable pendant une durée déterminée et que cette période est écoulée, la contribution n'est pas nécessairement tuée à cet instant. Elle est effectivement retirée de sa file d'attente, lors de la survenance du premier événement consécutif au moment où la contribution a dépassé son délai de mémorisation.

Autrement dit, ce n'est pas parce que l'on spécifie qu'une contribution est mémorisable pendant 10 minutes qu'elle l'est effectivement. Toutefois, comme elle est tuée avant toute nouvelle évaluation de la condition de synchronisation, cet artifice n'influence pas le comportement du point de synchronisation. Néanmoins, cette technique risque de perturber l'utilisateur lorsque ce comportement est visualisé graphiquement.

2. FONCTIONNALITES

2.1. Identification d'un point de synchronisation

Dans un premier temps, l'utilisateur doit pouvoir spécifier le point de synchronisation dont il désire analyser le comportement.

L'utilisateur doit également déterminer la période durant laquelle il a l'intention d'observer le mécanisme. Recréer le comportement d'un point de synchronisation durant toute la période de la simulation, risquerait en effet d'être fastidieux à observer et probablement inutile.

2.2. Contrôle du déroulement de l'animation

Pour permettre à l'utilisateur d'assimiler le comportement du point de synchronisation, il est important que ce dernier puisse contrôler la vitesse de l'animation.

Dès lors, après chaque présentation d'une photo, c'est l'utilisateur qui décide de passer à la photo suivante.

L'utilisateur a également la possibilité de revenir en arrière en changeant le sens de l'animation.

2.3. Déplacement de la fenêtre

Si la taille du graphique est supérieure à l'espace physique disponible, suite à une condition de synchronisation particulièrement longue ou à un nombre important de contributions au point de synchronisation considéré, l'utilisateur peut déplacer la fenêtre sur la partie du graphique qui l'intéresse.

2.4. Afficher/Effacer la partie commentaire

Lorsque l'outil est activé, les informations de la partie commentaire sont également affichées. Si l'utilisateur désire récupérer l'espace occupé par ces données, il a la possibilité de les faire disparaître et de les rappeler ultérieurement.

3. EXEMPLE

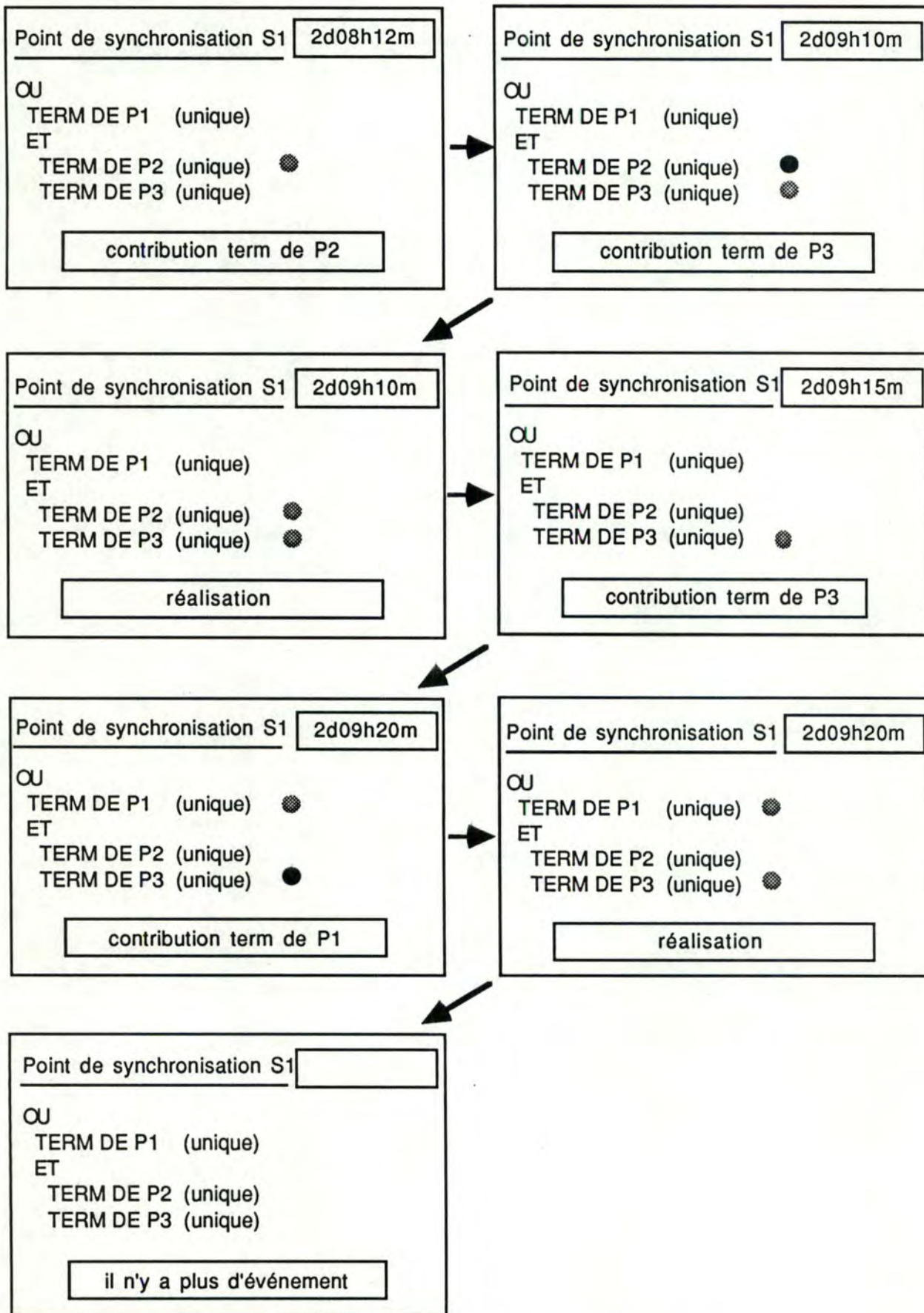


figure IV.2. Animation d'un point de synchronisation.

- CHAPITRE V-

**Outils et méthode d'évaluation du
comportement du système
d'information**

INTRODUCTION

L'objectif de ce chapitre est de proposer une démarche générale d'évaluation du comportement du système d'information ayant comme support méthodologique le schéma de la dynamique ainsi qu'un diagramme qui montre l'évolution de la performance du système d'information à différents moments.

En ce qui concerne la démarche d'interprétation et son support informationnel, nous expliquerons tout d'abord son cadre de référence, nous procéderons ensuite à la présentation de l'outil de juxtaposition d'informations sur le schéma de la dynamique, tant au niveau des informations affichées qu'au niveau des fonctionnalités offertes, et pour terminer, nous exposerons la démarche d'interprétation ayant comme support méthodologique l'outil en question.

Le deuxième outil d'évaluation du système d'information qui présente l'évolution des délais des transactions d'un type donné sera décrit au niveau des informations affichées et en exposant les fonctionnalités offertes à l'utilisateur.

V.1. OUTILS ET METHODE D'EVALUATION

V.1.1. Objectifs

Comme nous l'avons déjà souligné (cf préambule), l'objectif principal de l'outil de simulation est d'aider à la vérification de la faisabilité de la solution fonctionnelle retenue.

Rappelons également la définition que donne [Pigneur, 84] de la **faisabilité**. "Un schéma de comportement est faisable si le comportement modélisé, compte tenu du niveau d'utilisation des ressources et de la charge estimée du système, garantit le respect des performances attendues de la solution retenue".

Le contrôle de la faisabilité d'un système implique donc une vérification du niveau **d'utilisation des ressources**, ce qui peut être fait à l'aide des outils [2] et [9] décrits au chapitre I, ainsi qu'une évaluation de ses **performances**.

Nous proposons une **démarche**, inspirée de [Bodart, 85], qui permet d'**évaluer la performance** d'un système d'information, ainsi que de **localiser et d'expliquer** un comportement défaillant. De plus, cette démarche est assistée d'un **outil graphique** ayant comme support le schéma de la dynamique.

Pour **évaluer la performance** du système, nous avons juxtaposé sur ce schéma les délais moyens de type d'événement (cf. III.2.3). Nous avons également superposé des informations permettant une **localisation** de la partie du système d'information susceptible de présenter un comportement défaillant (durée moyenne d'attente, ...).

Pour **expliquer** les mauvaises performances d'une partie du système d'information, on peut faire appel aux outils décrits dans le chapitre I.

L'évaluation des performances d'un système n'ayant de sens que si la **propagation de la charge** est satisfaisante, nous juxtaposerons également sur le schéma de la dynamique des informations permettant de déterminer si la charge a été absorbée correctement.

V.1.2 Description

La description de cet outil comprend le détail des **Informations** telles qu'elles seront affichées, une présentation des **fonctionnalités** offertes à l'utilisateur ainsi qu'un **exemple** à titre d'illustration.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent soit, sous la forme d'un texte, dans une **partie commentaire**, soit sous une forme semi-graphique, dans la **partie graphique** de l'outil.

1.1. Partie commentaire

Comme illustré à la figure V.1, cet espace contient les informations relatives

- à la **simulation** (le titre, la date d'exécution)
- à la **période** durant laquelle les mesures globales ont été calculées.

Cette zone détaille éventuellement des informations concernant un **type de transaction** (le type d'événement initial et terminal, le nombre de transactions sur lequel le délai moyen du type de transaction a été calculé ainsi que la durée de la transaction la plus courte, la plus longue (avec, pour ces deux transactions, la date de survenance de l'événement initial) et, finalement, la durée moyenne des transactions du type considéré).

Ces derniers renseignements ne sont fournis à l'utilisateur que si celui-ci a identifié au préalable un type de transaction.

Notons que la période d'extraction des transactions correspond à la période de calcul des mesures globales.

1.2. Partie graphique

La partie graphique affiche le schéma de la dynamique sur lequel sont juxtaposées des informations relatives à la **propagation de la charge** ou bien des informations utiles à l'évaluation des **performances** du système.

1.2.1. Propagation de la charge

Pour les **types de processus**, les mesures les plus pertinentes en vue d'analyser la propagation de la charge sont le nombre de **déclenchements**, **d'activations** et de **terminaisons**.

Il aurait été également intéressant de juxtaposer le nombre d'interruptions et de réactivations de processus d'un même type pour vérifier si tous les processus interrompus ont pu redémarrer mais l'espace disponible était trop restreint si l'on voulait garder une lisibilité suffisante. De plus, si le nombre de processus activés et terminés sont équivalents, il y a beaucoup de chance pour que les processus interrompus aient pu être réactivés.

On représente un type de processus et les informations qui y sont associées par :

déclen.	nom
activat.	
termin.	

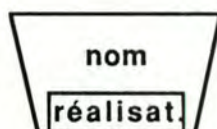
où

- le **nom** est le nom du type de processus
- le **déclen.** est le nombre de déclenchements de processus de ce type
- l'**activat.** est le nombre d'activations de processus de ce type
- la **termin.** est le nombre de terminaisons de processus de ce type

Notons qu'en général, le nombre de déclenchements d'un processus d'un type A est redondant avec le nombre de terminaisons du processus d'un type B, si B déclenche A. Toutefois, ce n'est pas toujours le cas, par exemple si B déclenche A de façon multiple, avec un paramètre de déclenchement qui varie en cours d'exécution.

Pour les **points de synchronisation**, nous retenons uniquement le **nombre de réalisations**, puisque, à priori, le nombre de types d'événements contribuant à un point de synchronisation est illimité.

On représente un point de synchronisation et son nombre de réalisations par :



où

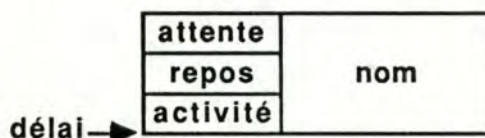
-le **nom** est le nom du point de synchronisation

-la **réalisat.** est le nombre de réalisations de ce point de synchronisation

1.1.2. Performances du système

Pour les **types de processus**, les résultats statistiques pertinents pour rendre compte des performances du système simulé sont les **délais moyens de terminaison**. De plus, pour expliquer ces délais, il est utile de connaître leur **durée moyenne** d'attente, de repos et d'activité.

On représente un processus et les informations qui y sont associées par :



où

-le **nom** est le nom du type de processus

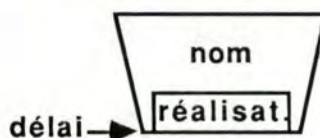
-l'**attente** est la durée moyenne d'attente du " "

-le **repos** est la durée moyenne de repos du " "

-l'**activité** est la durée moyenne d'activité du " "

-le **délai** est le délai moyen de la terminaison des processus de ce type par rapport à l'événement initial.

On représente un point de synchronisation et les informations qu'y sont associées par :



où

-le **nom** est le nom du point de synchronisation

-la **réalisat.** est la durée moyenne de réalisation du point de synchronisation

-le **délai** est le délai moyen de la réalisation de ce point de synchronisation, par rapport à l'événement initial (cf. III.2.3.).

2. FONCTIONNALITES

2.1. Identification d'un type de transaction

Si l'utilisateur désire voir juxtaposées sur le schéma de la dynamique des informations concernant un type de transaction, il doit spécifier un **type d'événement initial et terminal**

La **période** durant laquelle les occurrences du type de transaction sont extraites est la période de calcul des statistiques globales, on veut éviter de la sorte une juxtaposition d'informations non comparables.

2.2. Propagation de la charge/performances

Comme nous l'avons déjà dit au point 1, sont juxtaposées sur le schéma de la dynamique soit des informations relatives à la propagation de la charge, soit des informations concernant la performance du système.

Il est par conséquent nécessaire d'offrir à l'utilisateur une fonction qui lui permettra de sélectionner l'un de ces deux types d'information.

2.3. Activation d'autres outils

Il est parfois nécessaire pour expliquer les mauvaises performances d'un système (cf. V.I.3) d'avoir recours aux outils que nous avons décrits dans la partie I.

Il est intéressant de pouvoir activer ces outils de telle manière qu'ils disposent leurs informations conjointement aux informations affichées par l'outil décrit présentement. Ceci permettrait une consultation simultanée de diverses informations.

Par exemple, l'utilisateur devrait pouvoir visualiser des informations concernant l'évolution du temps d'attente d'un processus d'un type donné en même temps que les durées relatives aux processus en question qui sont juxtaposées sur le schéma de la dynamique. Ce qui peut se faire aisément en superposant sur un même espace physique les fenêtres relatives à ces deux outils.

2.4. Gestion de fenêtre

La taille d'un schéma de la dynamique peut être largement supérieure à la taille de l'écran. Il est donc nécessaire de disposer des fonctions classiques de gestion de fenêtres (scrolling, zooming) pour pouvoir visualiser interactivement la totalité du graphique.

2.5. Impression

Il est utile de pouvoir analyser les informations offertes par cet outil à partir d'un support sur papier ou tout simplement avoir une trace des résultats clés fournis par la simulation.

Une analyse "sur papier" est d'autant plus recommandée que la taille du schéma de la dynamique est importante. Une démarche d'interprétation exclusivement interactive serait dans ce cas très pénible.

2.6. Aide

Cette fonction affiche la signification des mesures superposées sur le schéma de la dynamique.

2.7. Afficher/effacer la partie commentaire

Lorsque l'utilisateur active l'outil décrit ici, les informations de la partie commentaire sont affichées également. S'il désire récupérer l'espace occupé par ces données, il a la possibilité de les faire disparaître et de les rappeler ultérieurement.

3. EXEMPLES

- Simulation *Gestion Prêt* exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Globales établies durant la période 1d à 5d
- Typ. Evén. Initial *gén. Demande de prêt* Typ. Evén. Terminal *term. Acceptation du prêt*
- 119 Transactions
 - Délai Moyen 10h05m29s
 - Délai Min. 2h35m47s survenue à 4d09h54m14s
 - Délai Max. 1d1h35m12s survenue à 1d15h32m48s

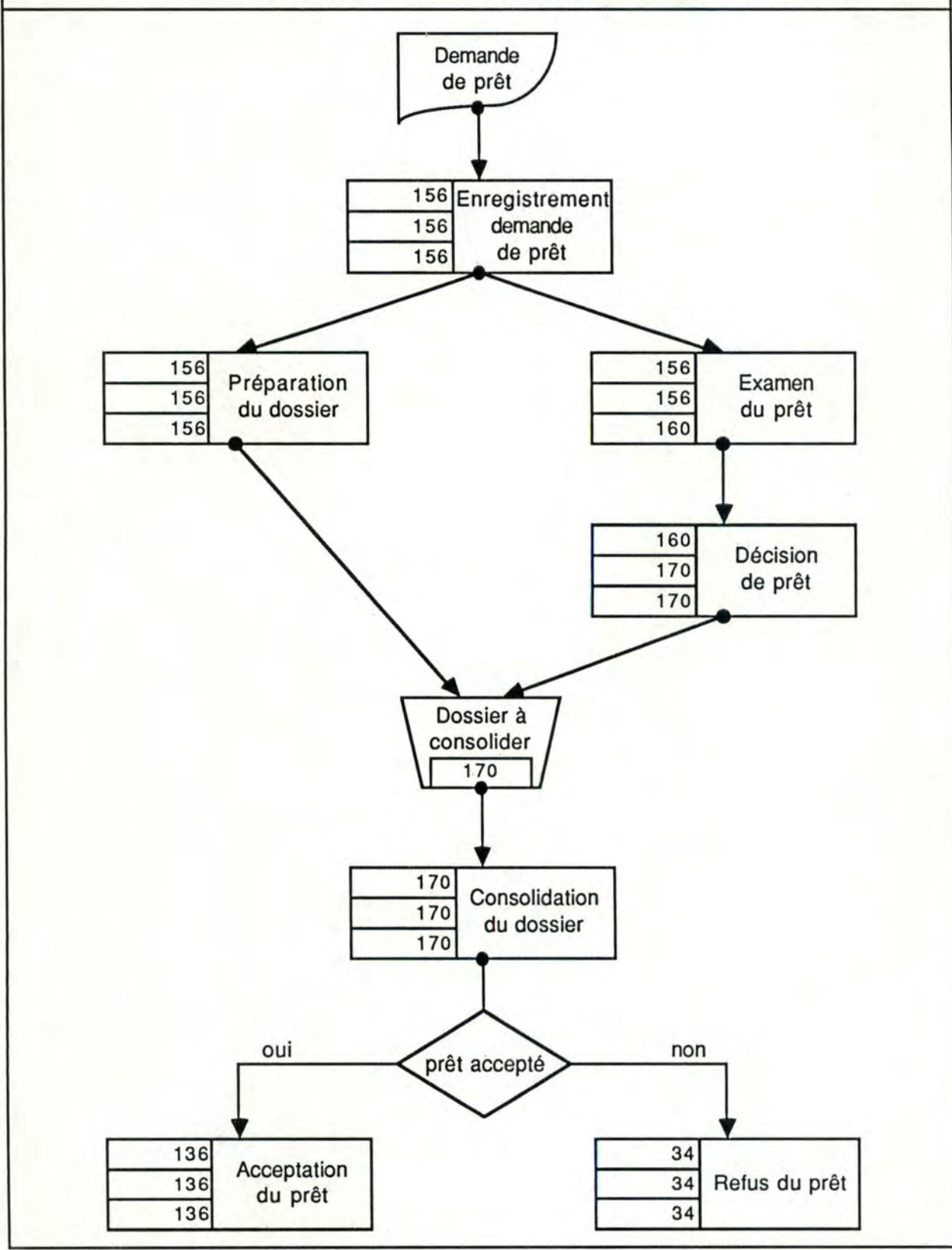


figure V.1. Propagation de la charge (après une période de chauffe de 1 jour)

- Simulation *Gestion Prêt* exécutée le 10 Juillet 1987 à 15h57m59s
- Statistiques Globales établies durant la période 1d à 5d
- Typ. Evén. Initial *gén. Demande de prêt* Typ. Evén. Terminal *term. Acceptation du prêt*
- 119 Transactions
 - Délai Moyen 10h05m29s
 - Délai Min. 2h35m47s survenue à 4d09h54m14s
 - Délai Max. 1d1h35m12s survenue à 1d15h32m48s

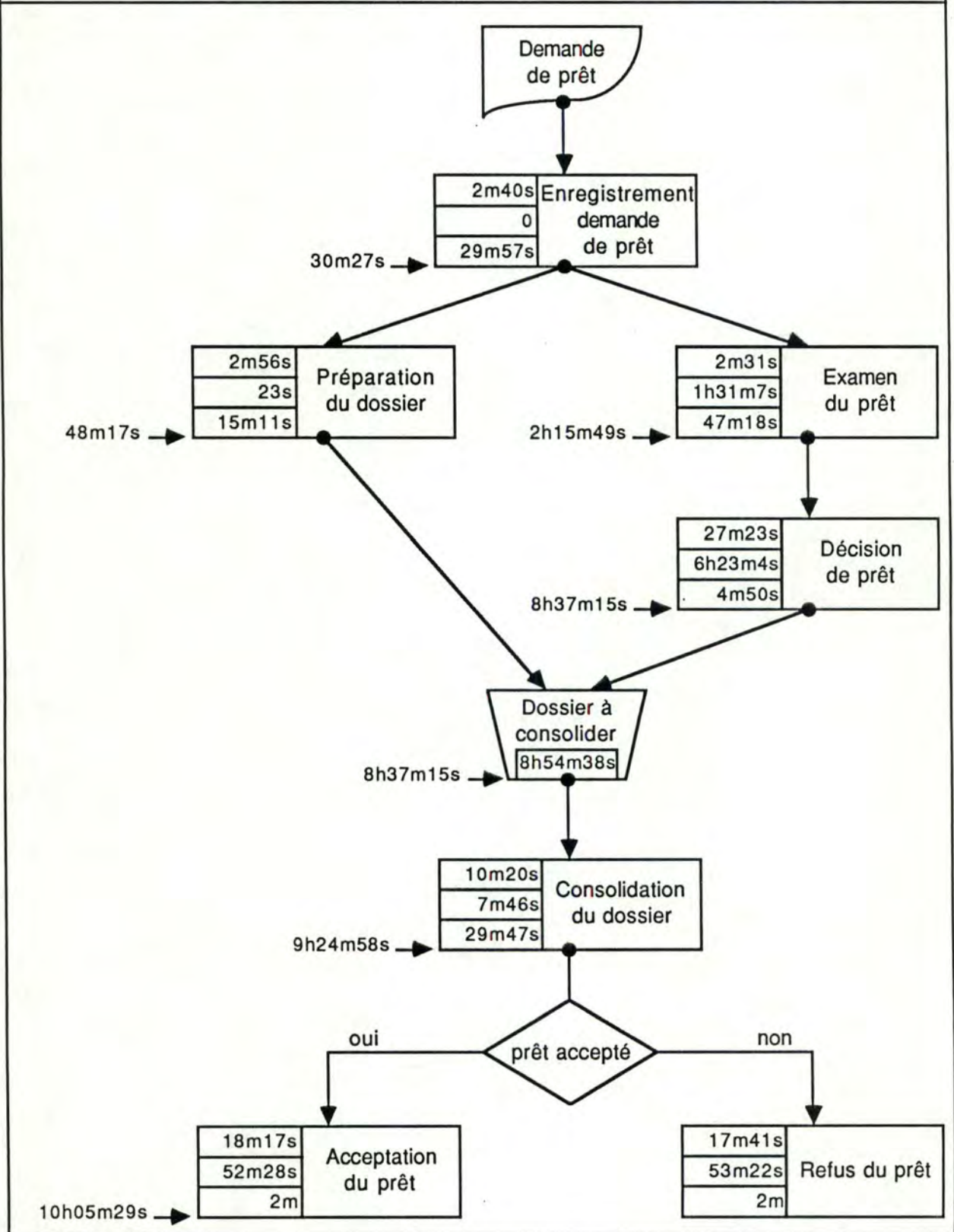


figure V.2. Performances du système

V.1.3. Méthode d'interprétation

La démarche que nous proposons pour interpréter les mesures relatives à la performance d'un système comprend deux étapes successives :

1. analyse de la propagation de la charge
2. évaluation et explication des performances du système

Suivant la précision recherchée dans la localisation et l'explication du mauvais comportement du système, l'utilisateur consultera, lors de ces étapes, les résultats statistiques globaux, périodiques ou chronologiques à l'aide de l'outil graphique approprié (cf chapitre I).

ANALYSE DE LA PROPAGATION DE LA CHARGE

Avant d'évaluer la performance d'un système, il faut s'assurer que la charge spécifiée, c'est-à-dire le volume et la répartition dans le temps des événements externes, se propage correctement. Il est en effet inutile d'analyser les délais de survenance d'événements si le volume de survenances de ces événements n'est pas satisfaisant.

Cette analyse consiste à s'assurer, pour un type de processus, que le nombre de déclenchements, d'activations et de terminaisons sont proches l'un de l'autre, ce qui peut se faire aisément en consultant ces mesures sur le schéma de la dynamique.

EVALUATION ET EXPLICATION DES PERFORMANCES DU SYSTEME

D'après [Pigneur, 84], "la spécification des **performances** attendues du comportement d'un système porte essentiellement sur le délai compris entre la survenance des événements d'une classe E1 et celle des événements d'une classe E2 qui leur succède dans le temps".

Dans le cadre de notre outil d'interprétation, il s'agira d'estimer le délai moyen pour chaque type de transaction retenu.

L'**évaluation des performances** du système consiste à **comparer** les délais moyens des types de transaction **estimés** par l'utilisateur avec ceux **établis** par la simulation.

Une identification des types de transaction qui présentent des divergences entre les performances attendues et obtenues permet de **localiser** les parties du système d'information qui ont un comportement anormal.

Chaque type de transaction défaillant doit ensuite être analysé pour découvrir les **causes** de son comportement. Pour cela, il suffit d'examiner les délais moyens des types d'événements constituant le type de transaction, leur analyse devrait en effet permettre de localiser les processus présentant une durée de vie trop importante, et, par-là même, d'**expliquer** le comportement anormal du type de transaction en identifiant les **ressources critiques**.(cf. préambule).

Une manière de procéder consiste à examiner les délais moyens des types d'événements constituant le type de transaction que l'on désire analyser en les parcourant en partant du type d'événement terminal. Lorsque la différence entre deux délais moyens de deux types d'événements successifs est trop grande, il convient d'examiner si cet écart se justifie.

Pour cela, deux cas sont à envisager :

- l'écart exprime la durée de vie moyenne de processus d'un même type
- l'écart représente une durée moyenne de synchronisation

Dans le premier cas, il faut vérifier si les processus présentent des temps d'attente et/ou des temps de repos importants ce qui se fait aisément en consultant les durées relatives au type de processus en question.

Dans le cas où le type de processus présente un **temps d'attente** important il est intéressant d'examiner la **capacité** moyenne utilisée, le nombre moyen de **points d'entrée** ainsi que la file d'attente des ressources requises par les processus du type considéré pour identifier les ressources critiques, ce qui peut se faire en consultant l'outil [2] décrit au point 1.1.2. La priorité du processus peut également intervenir dans l'explication du temps d'attente, mais on ne dispose pas de mesure facilement exploitable pour identifier cette cause.

Le type de processus peut également rendre compte d'un **temps de repos** important, ce qui nous amènera à examiner les **calendriers** de disponibilité des ressources requises par les processus pour mettre en évidence des problèmes de plage de disponibilité de ces mêmes ressources. Nous disposons à cet effet de l'outil [11] décrit au point 1.3.2.

Dans le deuxième cas, il s'agira d'identifier, à l'aide de l'outil [3] décrit au point 1.1.2, les types d'**événements contraignants** (cf. préambule). Parmi ces types d'événements, s'il y a des types d'événements internes, les **chemins** les causant seront analysés.

La démarche que nous venons de décrire est basée sur l'examen des résultats globaux. Si l'utilisateur désire faire une **analyse complémentaire**, il peut utiliser les outils décrits au point 1.2, mettant en évidence l'évolution des phénomènes en fonction du temps. Il sera alors possible à l'utilisateur de détecter la période pendant laquelle une ressource semble la plus critique.

Cette **analyse** peut encore être **affinée** au moyen des outils décrits dans le chapitre IV. Ces outils nous permettent en effet de visualiser le comportement d'un point de synchronisation défaillant, d'une transaction déterminée, ou encore de montrer graphiquement la réaction du système d'information à un événement spécifié.

Pour que la représentation d'une transaction présente un intérêt pour l'analyse de la performance du système, il est important de pouvoir sélectionner une transaction représentative d'un bon ou d'un mauvais fonctionnement du système. A cet effet, nous proposons au point V.2 un outil qui facilitera ce choix.

V.2. COMPOTEMENT D'UN TYPE DE TRANSACTION

V.2.1. Objectif

Dans la démarche d'interprétation proposée au point de V.1.2., nous avons fait remarquer, qu'une analyse plus fine du comportement moyen de transactions d'un même type, peut se faire en examinant le comportement d'une transaction particulière à l'aide des outils de représentation décrits au point de IV.2.

Le choix d'une telle transaction peut être facilité en montrant l'évolution des délais des transactions sur lesquelles le délai moyen a été calculé. L'utilisateur a ainsi la possibilité de sélectionner, sur base de ces délais, une transaction représentative du comportement moyen des transactions ou d'analyser plus finement une transaction présentant un délai particulièrement court ou anormalement long.

Il n'est pas toujours nécessaire de représenter graphiquement une transaction pour pouvoir expliquer un délai anormal. La connaissance du moment où la transaction présentant un comportement défaillant a débuté peut suffire à expliquer ce délai.

A titre d'illustration, sur l'exemple de la figure V.5., les demandes de prêt qui ont débuté le premier jour après 13 heures 30 ont duré plus longtemps que celles qui ont commencé le matin car elles n'ont pas pu être traitées le même jour.

V.2.2 Description

La description de cet outil comprend le détail des **Informations** affichées, une présentation des **fonctionnalités** offertes à l'utilisateur ainsi qu'un **exemple** à titre d'illustration.

1. INFORMATIONS

Les informations présentées à l'utilisateur apparaissent dans deux zones différentes.

1.1. Partie commentaire.

Comme illustré à la figure V.5., cet espace contient les informations relatives

- à la **simulation** (le titre, la date d'exécution)
- au **type de transaction** (le type d'événement initial et terminal, le nombre de transactions du type considéré, la période durant laquelle les transactions ont été extraites)

Une transaction appartient à une période d'extraction donnée si son événement initial et ses événements terminaux surviennent endéans la période considérée.

1.2. Partie graphique

Comme illustré à la figure V.5., la partie graphique présente sous la forme d'un **diagramme à bâtons** les délais des transactions d'un type donné.

La hauteur d'un bâtonnet représente le **délai d'une transaction** et son emplacement sur l'axe de abscisses détermine le moment où la transaction a débuté.

Il se peut que plusieurs transactions débutent **au même moment**. Dans ce cas, la hauteur du bâtonnet représente le délai **moyen** des transactions qui ont débuté simultanément.

L'axe des abscisses matérialise la période durant laquelle les transactions ont été extraites tandis que l'axe des ordonnées représente la durée des transactions.

La précision avec laquelle les axes sont découpés dépend du niveau de détail souhaité par l'utilisateur.

Si l'utilisateur a spécifié (cf. fonctionnalités) une période de trois heures pour l'axe des abscisses, et que l'on dispose de 18cm pour la représenter, on peut raisonnablement découper l'axe en tranches de deux minutes (cf. figure V.5.).

Cet outil permet aussi de **situer**, au moyen d'un symbole placé sur l'axe des ordonnées, les **délais** des transactions par rapport au **délai moyen** du type de transaction. De plus, à l'aide d'une fonction décrite au point 2.3., l'utilisateur a la possibilité de situer graphiquement ce délai moyen par rapport au **délai minimum** et **maximum** du type de transaction. Cette fonctionnalité est illustrée à la figure V.3.

2. FONCTIONNALITES

2.1. Identification d'un type de transaction

Pour rendre compte de l'évolution de transactions d'un même type, l'utilisateur doit identifier au préalable un type de transaction en spécifiant un **type d'événement initial** et **terminal** ainsi que la **période** durant laquelle les occurrences du type de transaction doivent être sélectionnées.

2.3. Activation d'autres outils

Ayant identifié une transaction présentant un comportement pathologique, l'utilisateur doit être à même d'activer un outil de représentation d'une transaction (Cf. IV.2.).

Si l'utilisateur ne désire pas analyser le comportement exact d'une transaction, il doit pouvoir (ré)activer l'outil de juxtaposition d'informations sur le schéma de la dynamique représentant le comportement moyen des transactions du type considéré.

2.4. Gestion de fenêtre

La taille du graphique peut être largement supérieure à l'espace physique dont on dispose, il est donc nécessaire d'offrir à l'utilisateur la possibilité de faire du "scrolling" horizontal et lui permettre ainsi de parcourir toute la zone d'observation.

De même, pour pouvoir présenter des informations relatives à deux outils différents sur un même écran, on doit pouvoir agir sur la taille de la fenêtre.

2.4. Situer la valeur moyenne

Cette fonction permet de situer le délai moyen d'un type de transaction par rapport au délai minimum et maximum (cf. figure V.3.)

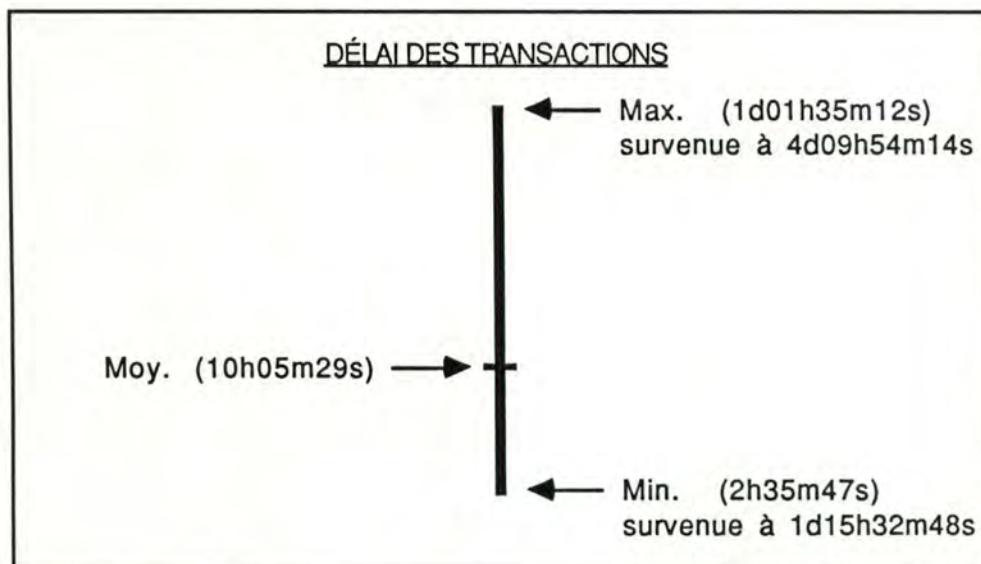


Figure V.3. situer la valeur moyenne

2.5. Zooming des axes

En ce qui concerne l'**axe des abscisses**, l'utilisateur doit pouvoir spécifier la période qu'il désire voir représentée dans une fenêtre. Si l'utilisateur a spécifié une période d'extraction s'étendant sur 4 jours, il peut se contenter dans un premier temps de visualiser dans une fenêtre l'évolution des transactions sur une période de 1 jour, la largeur totale du graphique correspondant dans ce cas à quatre fois la largeur d'une fenêtre. Si cette découpe s'avère trop grossière, pour une analyse plus détaillée, il peut restreindre cette période à quelques heures. Ainsi, par exemple, à la figure V.5. nous avons choisi de représenter dans une fenêtre une période de 3 heures.

La même fonctionnalité est disponible sur l'**axe des ordonnées**. Cette fonction est surtout utile lorsque la découpe n'est pas assez précise par rapport aux mesures affichées dans la fenêtre. Comme l'axe des ordonnées est découpé à partir de la valeur maximale des mesures affichées, cette situation se produit lorsque les mesures divergent très fort les unes des autres

Par exemple à la figure V.4., une graduation s'étendant jusque 50 permet une analyse plus fine des données, qu'une sur 150.

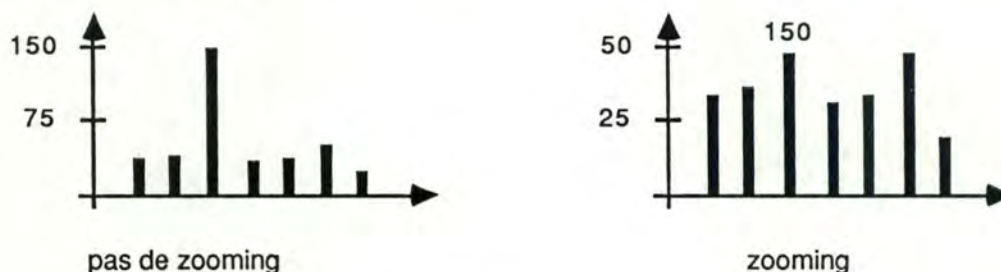


figure V.4.

Lorsqu'un bâtonnet est tronqué suite à un "zoom" sur l'axe des ordonnées, la valeur qu'il représente est affichée sur ce bâtonnet, comme illustré à la figure V.4.

2.6. Afficher/effacer la partie commentaire

Lorsque l'utilisateur active l'outil décrit ici, les informations de la partie commentaire sont affichées également. S'il désire récupérer l'espace occupé par ces données, il a la possibilité de les faire disparaître et de les rappeler ultérieurement.

2.7. Impression

L'outil que nous venons de présenter se prête aussi bien à une utilisation interactive qu'à une analyse sur papier. Toutefois, lorsque la période à observer est particulièrement longue, une analyse sur papier est probablement plus rapide.

De plus, si l'utilisateur a déjà parcouru la période d'observation interactivement, il doit pouvoir imprimer de manière sélective la ou les sous-périodes critiques.

3. EXEMPLE

- Simulation *Gestion Prêt* exécutée le 10 Juillet 1987 à 15h57m59s
- Typ. Evén. Initial *gén. Demande de prêt* Typ. Evén. Terminal *term. Acceptation du prêt*
- 119 Transactions extraites durant la période 1d à 5d

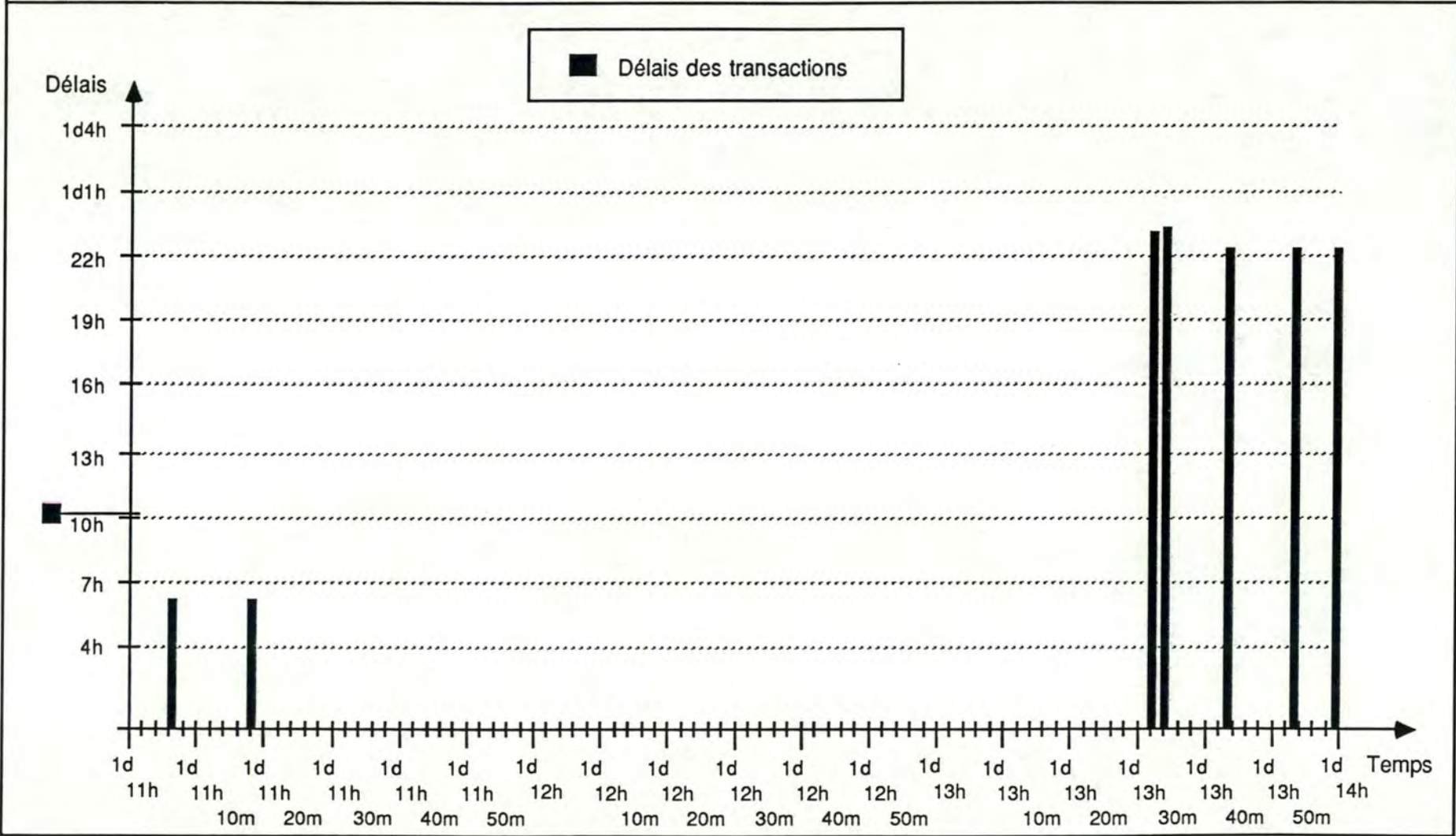


figure V.5. Comportement d'un type de transaction

- CHAPITRE VI-

Description des bases de données

INTRODUCTION

L'objectif de ce chapitre est de décrire les structures d'informations mémorisant les informations extraites de la trace de la simulation et manipulées par les outils décrits dans les deux chapitres précédents.

Pour des raisons de clarté, nous n'avons pas donné le schéma conceptuel de l'ensemble de la base, nous avons préféré effectuer cette présentation en décrivant trois sous-schémas qui correspondent :

- au comportement du système
- au comportement d'un point de synchronisation
- à l'évolution du type de transaction

VI.1. COMPORTEMENT DU SYSTEME

VI.1.1. Définition des entités

comportement

définition

On appelle "comportement" toute partie du comportement du système simulé pour laquelle l'utilisateur a demandé une représentation graphique et/ou animée. Rappelons que cette partie peut être la réaction du système à la survenance d'un événement donné ou, plus particulièrement, une transaction.

attribut

type_comport. indique si le comportement est une transaction ou la réaction du système à la survenance d'un événement particulier.

identifiée par

son attribut "**type-comport.**" son rôle "**début-par**" dans l'association "évé-initial" et ses rôles "**finl-par**" dans l'association "évé-terminal"

contrainte

Si l'attribut "type-comport" indique que le comportement considéré est une transaction, alors, "comportement" participe à au moins une association "évé-terminal, sinon, "comportement" ne participe à aucune association "évé-terminal".

événement

définition

On appelle "événement" tout événement de la trace de la simulation, décentralisé au niveau du poste de travail et faisant partie d'au moins un "comportement".

attributs

date-survenance date de survenance de l'événement.

type-évé type de l'événement.

nom-objet nom de l'objet associé à l'événement.

Les deux attributs suivants existent uniquement si l'événement est de type activation ou réactivation

temps-attente durée d'attente du processus avant sa (ré)activation

temps-repos durée de repos du processus avant sa (ré)activation

identifiée par

num_évé

contraintes

1. si un "événement" participe à une association "évé_initial", il a un des types suivants : déclenchement, génération.
2. si un "événement" participe à une association "évé_terminal", il a un des types suivants : terminaison, génération, réalisation.

VI.1.2. Définition des associations

évé-initial

définition

"évé_initial" représente l'événement auquel réagit le système ou bien l'événement initial d'une transaction.

relie

comportement, événement.

rôles

"débuté_par" pour "comportement"

"évé_initial_de" pour "événement"

connectivités

- 1-1 pour "comportement" (car comportement est soit une transaction qui, par définition, contient un et un seul événement initial, soit la réaction du système d'information à la survenance d'un événement initial).
- o-n pour "événement" (car, de par sa définition, une transaction contient généralement d'autres événements que son événement initial et un événement peut être l'événement initial de deux transactions qui diffèrent par le type d'événement terminal considéré).

évé-terminal

définition

"évé-terminal" représente un événement terminal d'une transaction.

relie

comportement, événement.

rôles

"finl_par" pour "comportement"

"évé_terminal_de" pour "événement"

connectivités

1-n pour "comportement" (car le comportement considéré peut être soit la réaction du système à la survenance d'un événement, dans ce cas il n'est pas caractérisé par un événement terminal, soit une transaction, qui, par définition, possède au moins un événement terminal).

o-n pour "événement" (car, de par sa définition, une transaction contient généralement d'autres événements que ses événements terminaux et un événement peut être l'événement terminal de deux transactions qui diffèrent par l'événement initial considéré).

précédence*définition*

"précédence" représente une relation de précédence (cf IV.2.1., point 2.4.) entre deux événements.

relie

événement, événement.

rôles

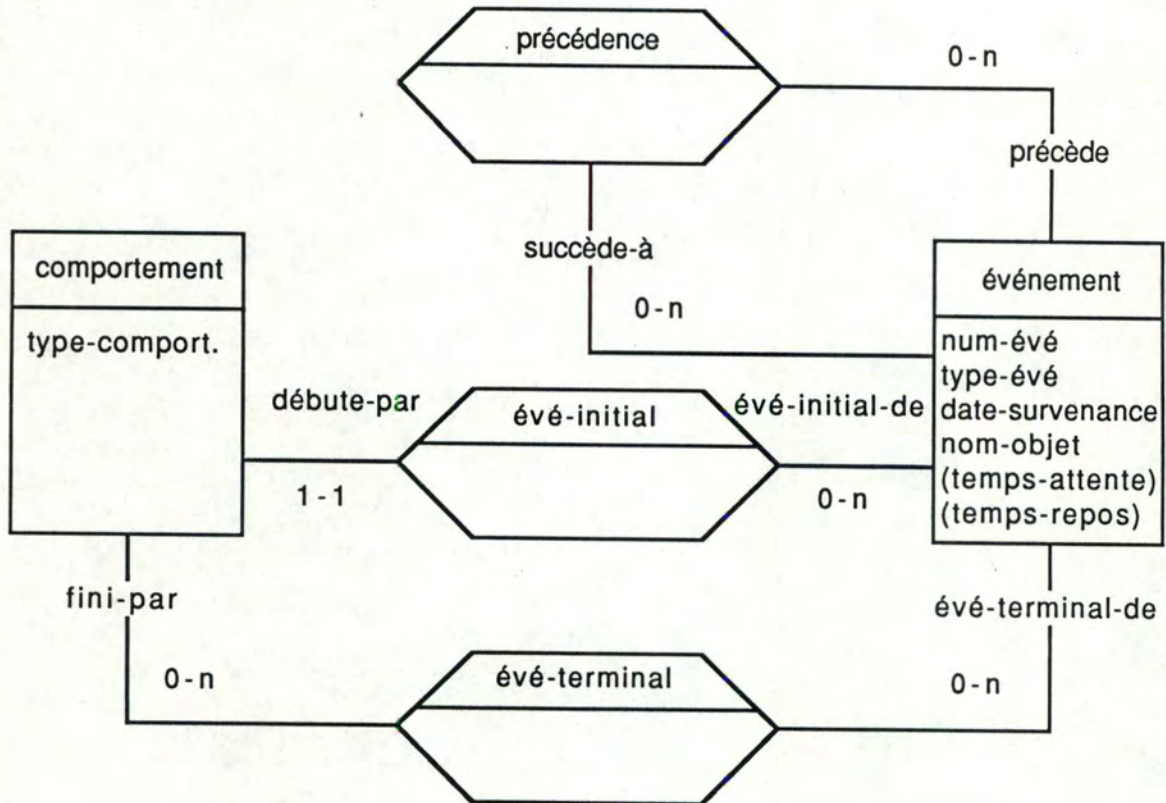
"précède" pour l'événement prédécesseur de l'autre événement.

"succède_à" pour l'événement successeur de l'autre événement.

connectivités

o-n pour "précède" (car le dernier événement de la simulation ne précède aucun événement et un événement en précède plusieurs dans le cas des enchaînements multiples ou éclatés)

o-n pour "succède_à" (car le premier événement de la simulation ne succède à aucun événement, et un événement "réalisation d'une synchronisation" succède à plusieurs contributions)

VI.1.3. Schéma de la base de données

VI.2. COMPORTEMENT D'UN POINT DE SYNCHRONISATION

VI.2.1. Définition des entités

point-synchro-animé

définition

On appelle "point-synchro-animé" tout point de synchronisation faisant partie du modèle simulé et pour lequel l'utilisateur a demandé une animation.

attribut

nom-synchro nom du point de synchronisation

identifiée par

nom-synchro

évé-synchro

définition

On appelle "évé-synchro" tout événement associé à un point de synchronisation "point-synchro-animé"

attributs

type-évé type de l'événement (contribution, réalisation, suppression de contribution(s))

date-survenance date de survenance de l'événement

identifiée par

num-évé

contraintes

1. "évé-synchro" participe à une association "contribution" si et seulement si son "type-évé" désigne une contribution.
2. Si "type-évé" désigne une réalisation, alors "évé-synchro" participe à au moins deux relations "passage" avec "est-contribué-par/supprime" comme rôle.
3. Si "type-évé" désigne une suppression, alors "évé-synchro" participe à au moins une relation "passage" avec "est-contribué-par/supprime" comme rôle.
4. Si "évé-synchro" est une occurrence d'une contribution dont le mode de contribution est unique, alors il participe au plus à une relation "passage" avec "contribue-à/est-supprimé-par" comme rôle.

expression

définition

On appelle "expression", un opérateur complété de son opérande s'il s'agit d'un opérateur unaire, ou de ses deux opérandes s'il s'agit d'un opérateur binaire faisant partie d'une condition de synchronisation

attribut

opérateur nom de l'opérateur de l'expression. Il existe deux types d'opérateurs, à savoir les opérateurs unaires (COMPTER, REGROUPER) et les opérateurs binaires (ET, OU, MEME-ORIGINE)

identifiée par

son attribut "**opérateur**" et son (ses) rôle(s) "**a-pour-opérande**" dans l'association "op-de-l'expr".

contraintes

1. Une expression est au moins la condition d'un point de synchronisation ou l'expression d'une opérande d'une autre expression.
2. Une "expression" participe toujours à une relation "opérande-de-l'expr" dont l'attribut "ordre" indique la première opérande de l'opérateur.
3. Si "opérateur" est binaire, alors l'"expression" participe à une relation "opérande-de-l'expr" dont l'attribut "ordre" indique la deuxième opérande de l'opérateur.

opérande*définition*

On appelle "opérande" une opérande d'une "expression". Rappelons qu'une opérande est soit la référence d'une contribution à un point de synchronisation, soit une expression.

attributs

Ces deux attributs n'existent que si "opérande" est la référence d'une contribution à un point de synchronisation.

t-évé-contrib type d'événement contribuant au point de synchronisation

mode-contrib mode de contribution (unique, multiple)

identifiée par

ses attributs "**t-évé-contrib**", "**mode-contrib**" si ceux-ci existent ou par son rôle "**a-pour-expr**" dans l'association "expr-de-l'op"

VI.2.2. Définition des associations

contribution

définition

"contribution" représente une contribution à un point de synchronisation.

relie

point-synchro-animé, évé-synchro.

rôles

"est-contribué-par" pour "point-synchro-animé".

"contribue-à" pour "évé-synchro".

connectivités

o-n pour "point-synchro-animé" (car une simulation peut se dérouler sans qu'il n'y ait de contribution au point de synchronisation, cependant, en général, un point de synchronisation est contribué plusieurs fois lors d'une simulation)

1-1 pour "évé-synchro" (car une contribution référence un et un seul point de synchronisation)

contrainte

Les contributions à un point de synchronisation sont des occurrences des types d'événement contribuant au point de synchronisation.

passage

définition

"passage" représente les relations de passage (cf. IV.2.1, point 2.2.) existant entre les événements associés à un point de synchronisation.

relie

évé-synchro, évé-synchro.

rôles

"participe-à/est-supprimé-par" pour l'événement prédécesseur de l'autre.

"est-contribué-par/supprime" pour l'événement successeur de l'autre.

connectivités

o-n pour "participe-à/est-supprimé-par" (car une réalisation ou une suppression ne précède pas d'événement associé au point de synchronisation et une contribution multiple peut participer à plusieurs réalisations)

o-n pour "est-contribué-par/supprime" (car une contribution ne succède à aucun événement associé au point de synchronisation et une réalisation succède à plusieurs contributions.)

condition-synchro

définition

"condition-synchro" représente la condition d'un point de synchronisation "point-synchro-animé", exprimée sous la forme d'une "expression".

relie

point-synchro-animé, expression.

rôles

"a-pour-condition" pour "point-synchro-animé".

"est-condition-de" pour "expression".

connectivités

1-1 pour "point-synchro-animé" (car la condition d'un point de synchronisation est spécifiée par une seule expression)

o-n pour "expression" (car "expression" peut être une opérande d'une autre expression et deux points de synchronisation différents peuvent être définis avec la même condition)

op-de-l'expr

définition

"op-de-l'expr" représente une opérande d'une expression

attribut

ordre indique si "opérande" est la première ou la deuxième opérande de l'"expression".

relie

expression, opérande.

rôles

"a-pour-op" pour "expression".

"est-l'op-de" pour "opérande".

connectivités

1-2 pour "expression" (car l'opérateur d'une expression est unaire ou binaire)

1-n pour "opérande" (car, de par sa définition, "opérande" est l'opérande d'au moins une expression et, si un même type d'événement contribue à plusieurs points de synchronisation, alors, la même référence de contribution apparaît comme opérande dans la condition de chacun de ces points de synchronisation)

expr-de-l'op

définition

"expr-de-l'op" représente l'expression d'une opérande

relie

expression, opérande

rôles

"est-l'expr-de-l'op" pour "expression".

"a-pour-expr" pour "opérande".

connectivités

o-1 pour "expression" (car une condition de synchronisation n'est généralement pas l'opérande d'une autre expression et si une expression n'est pas une condition de synchronisation, elle est la définition d'une (seule) opérande)

o-1 pour "opérande" (car une opérande peut être la référence d'une contribution à un point de synchronisation ou alors une (seule) expression)

type-de-contrib

définition

"type-de-contrib" représente le type de contribution à un point de synchronisation

relie

évé-synchro, opérande.

rôles

"est-une-occu-de" pour "évé-synchro".

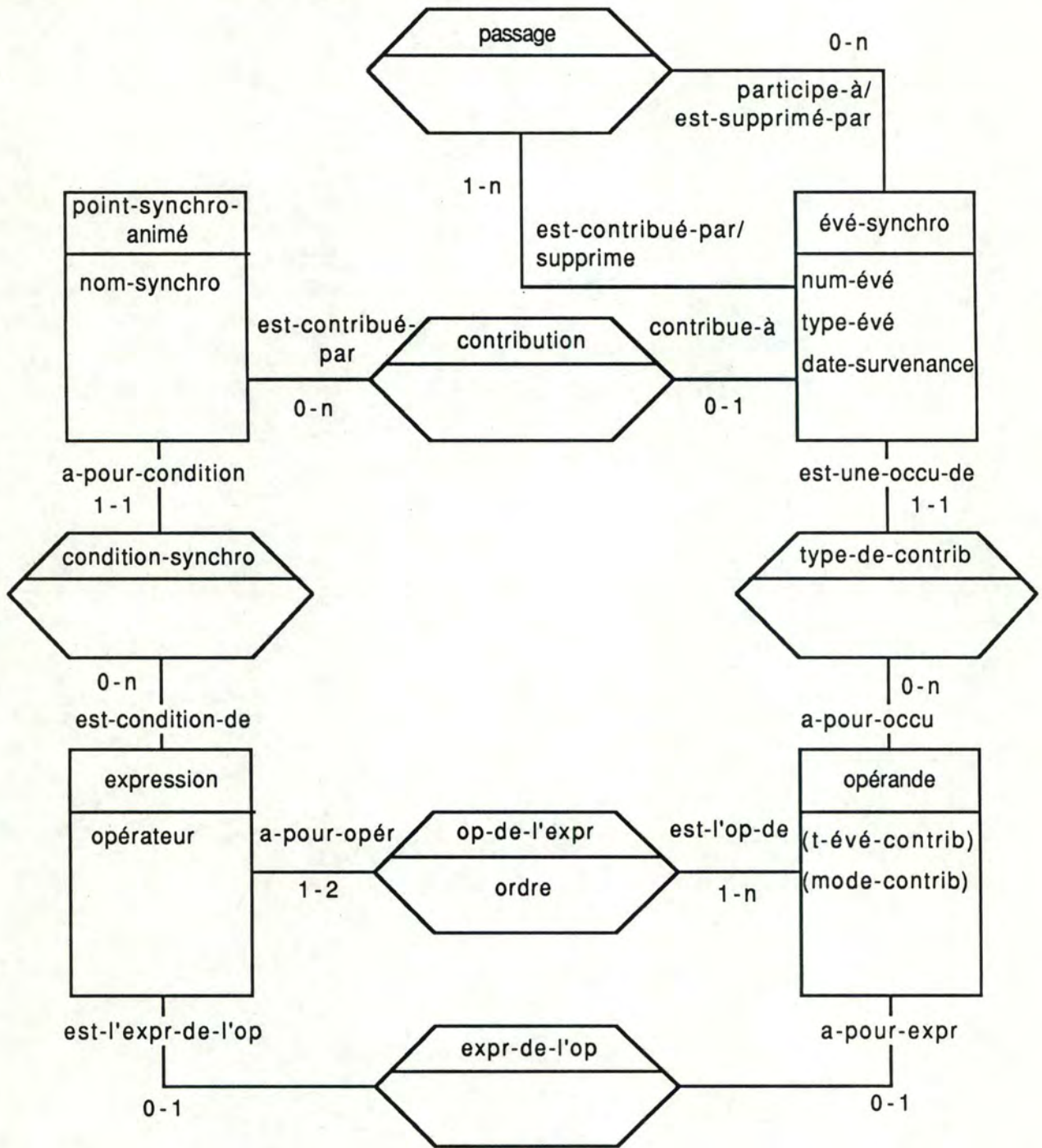
"a-pour-occu" pour "opérande".

connectivités

o-1 pour "évé-synchro" (car un événement n'est pas nécessairement une contribution et une contribution référence un et un seul point de synchronisation)

o-n pour "opérande" (car une simulation peut se dérouler sans qu'il n'y ait de contribution au point de synchronisation; cependant, en général, un point de synchronisation est contribué plusieurs fois lors d'une simulation)

VI.2.3. Schéma de la base de données



VI.3. COMPORTEMENT D'UN TYPE DE TRANSACTION

VI.3.1. Définition des entités

ens-transactions

définition

On appelle "ens-transactions" tout ensemble de transactions de même type qui se sont déroulées dans une période donnée (cf. IV.1.2). Le type de transaction et la période sont identifiés lors de l'utilisation de l'outil décrit au point VI.1.2.

attribut

période période durant laquelle les transactions de "ens-transactions" se sont déroulées

identifiée par

son attribut "**période**", ainsi que ses deux rôles "**débute-par**" dans l'association "t-évé-initial" et "**se-termine-par**" dans l'association "t-évé-terminal".

transaction

définition

On appelle "transaction" toute transaction faisant partie d'un ensemble "ens-transactions".

attributs

délai délai de la transaction

date-survenance date de survenance de l'événement initial la transaction

identifiée par

num-transa

type-évé

définition

On appelle "type-évé" tout type d'événement (génération, déclenchement, terminaison, réalisation) dont une occurrence appartient à une "transaction".

attributs

nom-objet nom du type d'objet associé au type d'événement

type type de l'événement (génération, déclenchement, terminaison, réalisation).

identifiée par

nom-objet, type

contraintes

1. Si "type-évé" participe à une relation "t-évé-initial", alors, son type est (génération), ou (déclenchement).
2. Si "type-évé" participe à une relation "t-évé-terminal", alors, son type est (génération), (terminaison), ou (réalisation).

VI.3.2. Définition des associations

appartenance

définition

"appartenance" représente une transaction appartenant à un ensemble de transactions.

relie

ens-transactions, transaction.

rôles

"contient" pour "ens-transactions"

"appartient-à" pour "transaction"

connectivités

- o-n pour "ens-transactions" (car lors de la simulation, il se peut qu'aucune transaction ne se soit déroulée durant la période considérée, ou au contraire, il se peut qu'il s'en soit déroulé plusieurs).
- 1-n pour "transaction" (car "transaction" appartient, de par sa définition, à au moins un ensemble de transactions "ens-transactions" et il se peut qu'elle appartienne à deux ensembles qui diffèrent uniquement par la période considérée pour l'extraction des transactions).

t-évé-initial

définition

"t-évé_initial" représente le type de l'événement initial des transactions d'un ensemble "ens-transactions".

relie

ens-transactions, type-évé.

rôles

"débuté_par" pour "ens-transactions"

"débuté" pour "type-évé"

connectivités

- 1-1 pour "ens-transactions" (car, par définition de "ens-transactions", les transactions qu'il contient sont de même type, elles débutent donc toutes par un événement d'un même type).
- 0-n pour "type-évé" (car une terminaison de processus n'est pas considéré comme un type d'événement initial possible et un type d'événement peut être le type d'événement initial de deux types de transaction différents).

t-évé-terminal*définition*

"t-évé-terminal" représente le type des événements terminaux des transactions d'un ensemble "ens-transactions".

relie

ens-transactions, type-évé.

rôles

"se-termine_par" pour "ens-transactions"
 "termine" pour "type-évé"

connectivités

- 1-1 pour "ens-transactions" (car, par définition de "ens-transactions", les transactions qu'il contient sont de même type, leurs événements terminaux sont donc de même type).
- 0-n pour "type-évé" (car un déclenchement de processus n'est pas considéré comme un type d'événement terminal possible et un type d'événement peut être le type d'événement terminal de deux types de transaction différents).

t-évé-inter*définition*

"t-évé-inter" représente le type d'un événement intermédiaire d'une transaction d'un ensemble donné pour lequel le délai moyen est calculé. Un événement intermédiaire est un événement d'une transaction qui n'est pas l'événement initial et qui n'est pas un événement terminal.

attribut

délai-Inter délai moyen du type d'événement.

relie

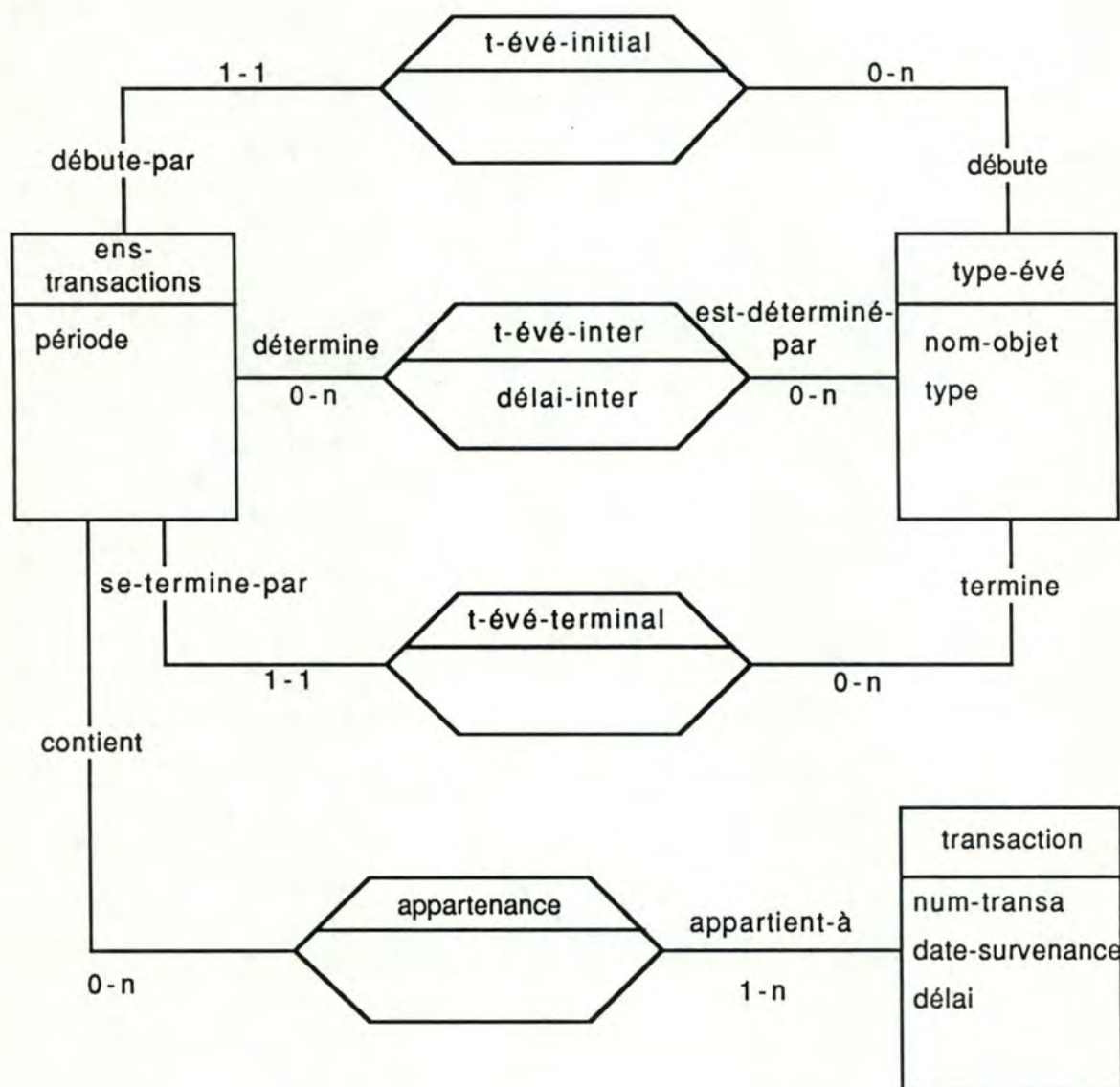
ens-transactions, type-évé.

rôles

"détermine" pour "ens-transactions"
 "est-déterminé-par" pour "type-évé"

connectivités

- o-n pour "ens-transactions" (car si les types d'événements extrémaux portent sur le même type de processus alors, le type de transaction ne détermine aucun type d'événement dit intermédiaire; dans le cas contraire, il détermine un ou plusieurs types d'événement intermédiaires).
- o-n pour "type-évé" (car un type d'événement peut être un type d'événement extrémal uniquement et un type d'événement appartient à deux types de transaction si ces types de transactions ne diffèrent que par la période durant laquelle leurs occurrences sont extraites).

VI.3.3. Schéma de la base de données

Bibliographie

- [Pigneur, 84] Y.Pigneur *Spécification et évaluation du comportement des systèmes d'information*, thèse de doctorat, Institut d'Informatique, FNDP, Namur, 1984.
- [Bodart, 85] F. Bodart, A.M.Hennebert, J.M.Leheureux, Y.Pigneur *Computer aided specification, evaluation and monitoring of information systems*, The sixth International conference on information systems, Indianapolis, 1985.
- [Bodart, 87] F.Bodart, Y.Pigneur *Conception assistée des applications informatiques, 1.-Etude d'opportunité et analyse conceptuelle*, MASSON, 1987
- [DSL-SIM, 86] DSL-SIM Manuel de référence, Institut d'Informatique, FNDP, Namur, 1986.
- [DSA, 87] DSA, version française, manuel de référence, Institut d'Informatique, FNDP, Namur, 1987.
- [DSL, 87] DSL, version française, manuel de référence, Institut d'Informatique, FNDP, Namur, 1987.

- PARTIE III -

**Production automatique du schéma
de la dynamique**

INTRODUCTION

Les méthodes de conception de systèmes d'information font largement appel aux représentations graphiques comme moyen de **documentation** et de **communication** entre le concepteur et l'utilisateur.

A titre d'illustration, nous avons repris à la figure 1 les dessins proposés par le logiciel IDA comme support graphique à l'analyse fonctionnelle. A ces schémas standards viennent s'ajouter les représentations du comportement du système d'information définies dans la partie II de notre mémoire (graphe des événements, graphe des objets, représentation animée) que l'on peut assimiler, au niveau de la forme, au schéma de la dynamique.

MODELES	REPRÉSENTATIONS GRAPHIQUES
Structuration des informations	Schéma Entité/Association
Structuration des traitements	Architecture des traitements
Dynamique des traitements	Schéma de la Dynamique
Statique des traitements	Flux des données

figure 1 représentations graphiques proposées par IDA

Si ces schémas ne visualisent pas toute la sémantique sous-jacente au modèle représenté, ils constituent néanmoins un compromis entre la formalisation et la lisibilité, aidant ainsi l'utilisateur dans sa tâche de conception.

Il existe bien entendu d'autres schémas utilisés classiquement pour la conception d'un système d'information. Par exemple, la représentation graphique d'un modèle d'accès généralisé (cf. [Hainaut, 86]), ou encore le schéma de la méthode de Jackson pour la construction de programme (cf. [Jackson, 75]).

De tels graphiques sont habituellement produits **manuellement** ou à l'aide d'un **éditeur** graphique (1) ; mais, dans les deux cas, la construction du schéma incombe totalement à l'utilisateur.

Certains éditeurs (par exemple, Information Engineering Workbench [IEW]) placent **automatiquement** les nouveaux symboles et les connections avec le reste du schéma. Toutefois, les algorithmes appliqués sont tellement naïfs que l'utilisateur doit de toute manière replacer les objets correctement par la suite.

Actuellement, sont développés des éditeurs qui offrent la possibilité de construire, à la demande de l'utilisateur un graphique automatiquement tout en respectant certains **critères esthétiques**.

Par exemple, Gincod (Graphic INTERactive CONceptual Design of Data, Transactions and Events [Batini, 85]) est un outil graphique, développé à l'Université de Rome, qui fournit un environnement intégré pour la définition de données, de transactions et d'événements.

La documentation produite par ce logiciel (modèle E/A étendu) peut être présentée sous forme d'un texte ou d'un graphique. Dans le dernier cas, une transformation est nécessaire entre la représentation interne du schéma et la représentation graphique. Pour effectuer cette génération automatique de graphique, Gincod utilise un algorithme qui tient compte de critères esthétiques adoptés par les personnes qui dessinent ce type de graphique.

(1) Un éditeur graphique signifie ici la possibilité d'éditer un schéma (E/A, Flux,...) à partir d'une description du modèle associé.

Un logiciel qui permet la production automatique de diagrammes décrivant un système d'information, est très utile pour le concepteur car il permet [Batini, 86] :

- une réduction des coûts consécutifs à la production et à la mise à jour des graphiques
- une augmentation de la lisibilité des graphiques résultant de la prise en compte de critères esthétiques qui peuvent être satisfaits automatiquement
- une intégration des phases de conception et de production des schémas
- une cohérence entre la description textuelle et la représentation graphique

Les travaux existant dans le domaine de la production automatique de graphiques qui respectent certains critères esthétiques peuvent être classés en plusieurs catégories :

- **arborescences.** [Reingold, 81] et [Wetherell, 79] définissent un ensemble de critères esthétiques pour la production automatique d'arbres binaires, tel que centrer les pères par rapport à leurs fils, minimiser la largeur de l'arbre, Comme ce problème est NP-hard, ils proposent des heuristiques pour le résoudre.
- **Graphes hiérarchisés** [Delarche, 79] propose une heuristique pour la production automatique de graphes hiérarchisés résumée dans [Carpano, 80]. Les critères esthétiques considérés sont la minimisation du nombre de croisements entre les connexions et une répartition homogène des sommets.
- **Entité/Association.** [Tamassia, 83] propose un algorithme pour la production automatique du schéma Entité/association. Cette méthode se caractérise par l'usage d'une grille pour le placement des objets et vérifie les critères esthétiques suivants, classés par ordre d'importance :
 - minimiser le nombre de croisements
 - minimiser le nombre de brisures des connexions
 - minimiser la longueur totale des connexions
 - centrer les associations par rapport aux entités qu'elles relient.
- **Flux des données** [Batini, 86] présente une méthode pour la production automatique du diagramme de flux (1). Tout comme pour le schéma précédent, une grille est définie permettant la production d'un dessin bien régulier et les critères esthétiques à vérifier sont les suivants :
 - minimiser le nombre de croisements
 - minimiser le nombre de brisures des connexions
 - minimiser la longueur totale des connexions
 - minimiser la surface du dessin
 - placement des interfaces sur les bords du schéma

Dans cette troisième partie, nous présentons une méthode originale pour la production automatique du **schéma de la dynamique** puisqu'aucun des résultats cités ne peut être utilisé à cette fin. Les **critères esthétiques** que nous avons retenus sont les suivants :

- <1> rendre le schéma le plus **hiérarchisé** possible
- <2> respecter certaines contraintes de **représentation** (certaines connexions doivent apparaître brisées)
- <3> minimiser le nombre de **croisements**
- <4> présenter un dessin **équilibré** et le plus **vertical** possible

(1) Le **diagramme de flux** produit par cet algorithme diffère de celui présenté dans [Bodart, 87] qui visualise en plus du flux des données les services gérant les processus.

La production automatique d'un schéma de la dynamique peut être assimilée à la construction de graphes hiérarchisés si le schéma ne présente pas de circuit. Comme par ailleurs beaucoup de graphiques satisfont à cette contrainte (par exemple, le graphe des événements, le graphe des objets,...), nous avons développé dans le **chapitre VII** une méthode de production automatique de tels schémas. L'algorithme que nous proposons à cet effet se distingue totalement de celui présenté par [Delarche, 77] qui, comme nous le verrons, présente des inconvénients majeurs.

Notre méthode sera ensuite appliquée à la production du schéma de la dynamique dans le **chapitre VIII**. Cette adaptation consiste essentiellement à rendre le schéma de la dynamique hiérarchisable.

- CHAPITRE VII-

**Méthode de production automatique
de graphes hiérarchisés**

INTRODUCTION

Beaucoup de diagrammes utilisés classiquement dans le cycle de vie d'un système d'information (architecture des traitements, diagramme de Jackson,...) ainsi que certains graphiques définis dans le chapitre IV de ce mémoire (graphe des événements, graphe des objets,...) peuvent être construits avec une méthode de production automatique de graphes hiérarchisés puisqu'ils se présentent sous la forme d'un arbre ou d'un graphe sans circuit.

Pour le schéma de la dynamique le problème se révèle plus complexe car ce dernier est la représentation d'un graphe qui possède éventuellement des circuits, ce qui rend toute hiérarchisation impossible. Toutefois, même lorsqu'il contient des circuits, il est représenté autant que faire se peut de manière hiérarchisée.

Par conséquent, la méthode de production automatique de diagramme proposée ici devra être particularisée pour permettre la construction du schéma de la dynamique. Cette adaptation fera l'objet du dernier chapitre.

L'exposé de la méthode de production automatique de graphes hiérarchisés comporte trois points. Tout d'abord, le problème sera spécifié. Ensuite, nous montrerons deux solutions possibles et pour terminer, nous présenterons en détail la méthode retenue.

VII.1. SPECIFICATION DU PROBLEME

Avant d'aborder la spécification du problème où l'on mettra en évidence les critères esthétiques que doit vérifier le dessin d'une hiérarchie pour être visuellement agréable, nous présentons une série de définitions concernant les graphes orientés et hiérarchisés qui fixent la terminologie employée par la suite.

VII.1.1. Définitions

Toutes les définitions concernant les graphes orientés proviennent des notes de cours de [Fichet, 84], celles concernant les graphes hiérarchisés sont extraites de la thèse de doctorat de [Delarche, 79] et d'un article de [Warfield, 77].

VII.1.1.1. Graphes orientés

Un **graphe orienté** G est un couple $G = (X,U)$ constitué par

- un ensemble $X = \{ x_1, x_2, \dots, x_n \}$ fini ou dénombrable dont les éléments sont les **sommets** du graphe.
- une famille $U = \{ u_1, u_2, \dots, u_n \}$ de couples de sommets appelés les **arcs** du graphe.

Extrémité initiale et terminale d'un arc

Soit $G = (X,U)$ un graphe avec $X = \{ x_1, x_2, \dots, x_n \}$. Si $u = (x_i, x_j)$ appartient à U , on dit que x_i est l'**extrémité initiale** et que x_j est l'**extrémité terminale** de l'arc.

Chemin, longueur d'un chemin

Un **chemin** est une suite $\mu = (u_1, u_2, \dots, u_i, u_{i+1}, \dots, u_q)$ d'arcs tels que l'extrémité terminale d'un arc u_i coïncide avec l'extrémité initiale de l'arc suivant u_{i+1} .

Le nombre q d'arcs qui composent la suite μ est la **longueur du chemin** μ

Précédent, suivant

Le sommet x_i est un **précédent** du sommet x_j si x_i est l'extrémité initiale d'un arc dont x_j est l'extrémité terminale.

Ascendant, descendant

Le sommet x_i est un **ascendant** du sommet x_j s'il existe un chemin ayant x_i et x_j comme extrémités initiale et terminale.

Le sommet x_j est un **descendant** du sommet x_i lorsque x_i est ascendant de x_j .

Circuit

Un **circuit** est un chemin dont les extrémités coïncident.

Composantes fortement connexes (C.F.C.)

Une **C.F.C.** est un ensemble de sommets X tels que pour tous sommets x_i et x_j appartenant à X , il existe un chemin allant de x_i à x_j et il existe un chemin allant de x_j à x_i .

Par conséquent, deux sommets x_i et x_j font partie d'une C.F.C. s'il existe un circuit passant par x_i et x_j ou si $x_i = x_j$.

VII.1.1.2. Graphes hiérarchisés**Hiérarchie, niveaux**

On appelle **hiérarchie** H sur un graphe $G = (X,U)$ toute application de l'ensemble des sommets $X = \{ x_i, i = 1, \dots, m \}$ dans une partie $I = \{ 1, 2, \dots, k \}$ de l'ensemble des naturels, associant à tout sommet x_i appartenant à X une valeur n_i appartenant à I .

On dit que le sommet x_i est au **niveau** n_i dans la hiérarchie H , et k est le nombre de niveaux de la hiérarchie.

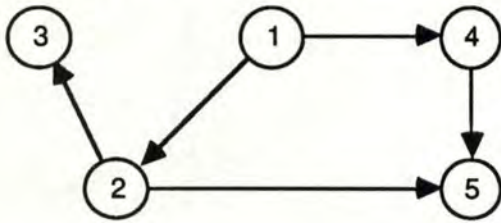
hiérarchie des rangs

Il existe une hiérarchie particulière dans le cas des graphes **sans circuit** appelée **hiérarchie des rangs** qui, à chaque sommet, fait correspondre le rang de ce dernier + 1.

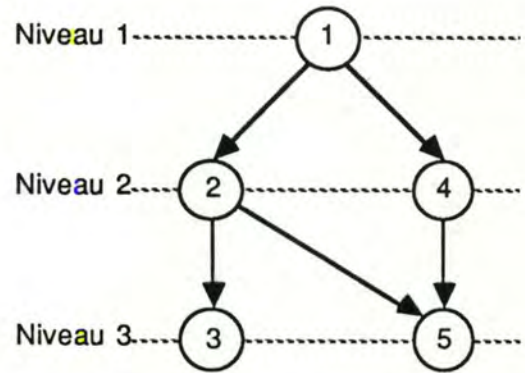
Un sommet s est dit de **rang** r , si et seulement si, r est le nombre maximal d'arcs de tout chemin d'extrémité terminale s (un sommet sans précédent est donc de rang 0).

Le dessin d'une telle hiérarchie est appelé une **carte**. Par convention, les **niveaux** sont symbolisés par des lignes horizontales parallèles entre elles et équidistantes. Les sommets sont représentés par un rond, et sont placés sur la ligne horizontale qui correspond à leur niveau, en respectant l'ordre des sommets à l'intérieur du niveau. Les **arcs** sont représentés par des droites orientées qui joignent les ronds correspondant aux extrémités de l'arc.

La figure VII.1 montre une hiérarchie des rangs composée de trois niveaux.



Graphique orienté



Hiérarchie des rangs

figure VII.1

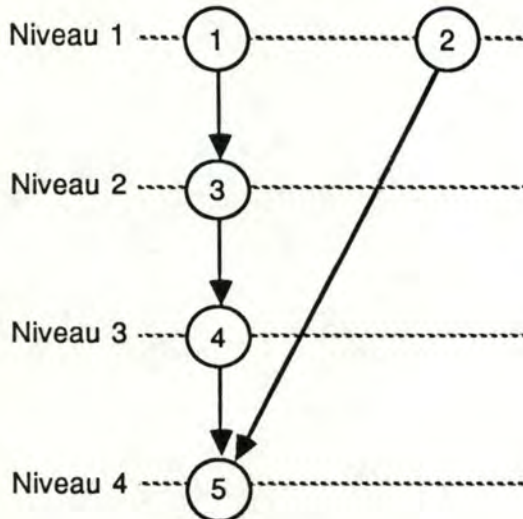
Portée d'un arc

La portée d'un arc (x_i, x_j) est la différence $n_j - n_i$ des niveaux de ses extrémités.

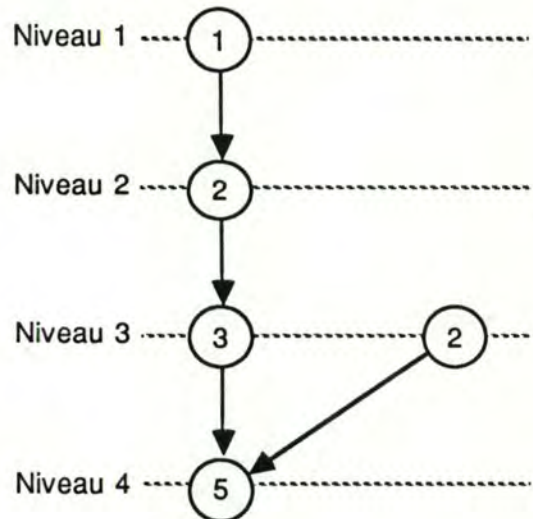
hiérarchie serrée

On appelle **hiérarchie serrée**, une hiérarchie compatible avec l'ordre des rangs (c'est-à-dire que si on prend deux sommets x_i et x_j de niveaux respectifs n_i et n_j tels que $n_i > n_j$ le rang de x_i est $>$ au rang de x_j) et qui conduit à une représentation telle que la portée de tous les arcs est minimale.

La carte de gauche de la figure VII.2 ne constitue pas une hiérarchie serrée puisque l'arc (B, E) a une portée de 3 qui peut être réduite à 1, comme illustré sur la carte de droite de la même figure.



Hiérarchie non serrée



Hiérarchie serrée

figure VII.2

hiérarchie propre

Si tous les arcs dans une hiérarchie ont une portée égale à un, on dit que la hiérarchie est propre.

Par exemple, la carte de gauche de la figure VII.2 représente une hiérarchie impropre, celle de droite par contre, représente une hiérarchie propre. Comme pourrait le laisser sous-entendre cet exemple, une hiérarchie serrée n'est pas nécessairement propre.

Longueur, largeur, mesure d'une hiérarchie

Le nombre de niveaux d'une hiérarchie est appelé la **longueur** d'une hiérarchie et le nombre maximum de sommets d'un niveau, la **largeur** d'une hiérarchie. Le nombre d'arcs est appelé la **mesure** d'une hiérarchie.

Par exemple, la hiérarchie serrée de la figure VII.2 a une longueur de 4, une largeur de 2 et une mesure de 5.

Chemin maximum

Un **chemin maximum** est un chemin contenant autant de sommets qu'il y a de niveaux dans la hiérarchie.

Par exemple, (1, 3, 4, 5) constitue un tel chemin.

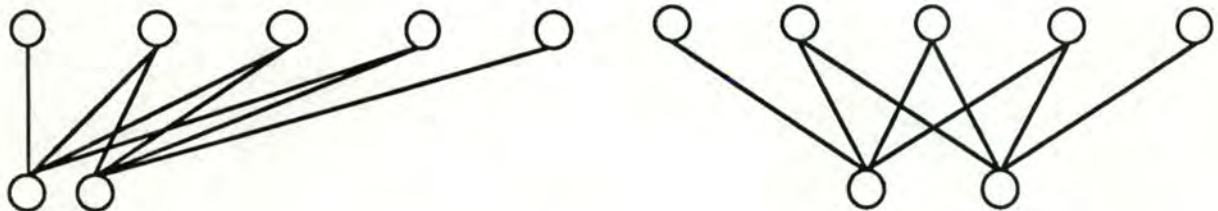
VII.1.2. Spécification du problème

Le problème consiste, à partir de la description d'un graphe orienté sans circuit, à calculer pour chaque sommet un couple (X, Y) de coordonnées cartésiennes dans le plan qui respecte certaines contraintes de dispositions inhérentes aux graphes hiérarchisés par les rangs.

Généralement, on aura un grand nombre de solutions compatibles avec les contraintes hiérarchiques. Le programme devra donc nous en fournir une qui soit visuellement agréable, c'est-à-dire

- <1> comportant peu de croisements d'arcs
- <2> assurant une répartition homogène des sommets

La minimisation du nombre de **croisements** n'est pas un critère absolu. En effet, la perception visuelle des croisements est d'autant plus néfaste pour la lisibilité du dessin que les arcs qui se croisent ont des directions voisines. Ainsi, par exemple, le dessin de gauche de la figure VII. 3 est moins lisible que celui placé à sa droite.



minimum de croisements

figure VII.3

répartition homogène

Il est donc nécessaire, en plus de la minimisation du nombre de croisements, d'imposer une **répartition homogène** des sommets. Selon nous, une répartition harmonieuse des sommets suppose un dessin **équilibré** et le plus **vertical** possible. L'application de ces critères à l'exemple qui nous intéresse conduit à la représentation de droite de la figure VII.3.

La construction d'un dessin équilibré ou symétrique consiste à centrer un sommet par rapport à ses suivants et inversement. Un dessin vertical s'obtient quant à lui en plaçant un sommet à la verticale de son suivant et/ou de son précédent.

Ces deux critères esthétiques sont illustrés à la figure VII.4. On peut remarquer sur cet exemple que les sommets 3 et 5 sont placés en dessous de leur précédent et pour garder la symétrie du dessin, le sommet 1 est centré par rapport à ses suivants.

Cet exemple met également en évidence la contradiction qui peut exister entre ces deux critères, puisque le sommet 1 ne peut pas simultanément se trouver au centre de ses suivants et être positionné à la verticale des sommets 2 et 4. Lorsqu'il y a contradiction, il convient par conséquent de privilégier un de ces critères, ce que nous avons fait en préférant la symétrie du dessin à son aspect rectiligne.

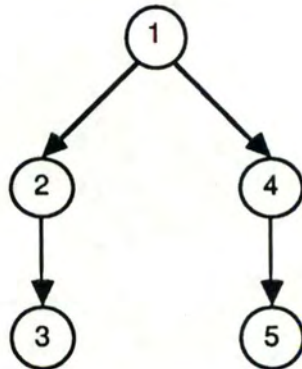


figure VII.4 répartition homogène

VII.2. PROPOSITION DE SOLUTIONS

Comme évoqué dans l'introduction de cette troisième partie [DELARCHE, 79] propose dans sa thèse de doctorat une heuristique, appelée la **méthode des degrés relatifs** pour la production automatique de graphes hiérarchisés.

Après une brève description de cette méthode, nous verrons pourquoi celle-ci n'est pas applicable à la production des diagrammes qui nous intéressent et nous présenterons ensuite une solution alternative.

VII.2.1. Méthode des degrés relatifs : principe heuristique

Le problème a d'abord été étudié pour les graphes bipartis et a ensuite été généralisé aux graphes hiérarchisés comportant plusieurs niveaux.

Parmi les graphes hiérarchisés, les graphes bipartis sont les plus simples puisqu'ils n'ont que deux niveaux.

Par convention S et S' désignent respectivement les sommets du premier niveau et du deuxième niveau, avec

$$S = \{s_i, i = 1, \dots, k\}$$

$$S' = \{s'_j, j = 1, \dots, k'\}$$

Chaque façon de dessiner un graphe est définie par un couple (p_1, p_2) de permutations respectivement sur S et S' . Ainsi, par exemple, $p_2(s_2, s_1, s_3, s_4, s_5, s_6)$ désigne l'ordre des sommets en partant de la gauche du deuxième niveau du graphe de la figure VII.7. On peut en effet constater que le sommet s_2 est en première position et que le sommet s_1 est en deuxième position, ...

Pour essayer de ne pas avoir trop de croisements d'arcs sur le dessin, une idée naturelle est de chercher à rendre les arcs les plus verticaux possibles, c'est-à-dire, étant donné une permutation p_1 des sommets du premier niveau à construire une permutation p_2 qui place les sommets du bas le plus possible en dessous des sommets du haut auxquels ils sont liés.

Pour cela, on attribue des abscisses (noté ABSC) aux sommets du premier niveau et l'on calcule des abscisses pour les sommets du deuxième niveau. Chaque abscisse est une valeur comprise entre 0 et 1 qui permet de travailler indépendamment de l'espace physique dont on dispose pour dessiner le graphique. Ainsi, par exemple, si on dispose de 30 cm en largeur pour représenter le dessin et qu'un sommet a pour abscisse 1/2, la multiplication de ces deux valeurs nous donne la coordonnée X du sommet à l'écran, en l'occurrence ici, 15 cm à partir de la gauche.

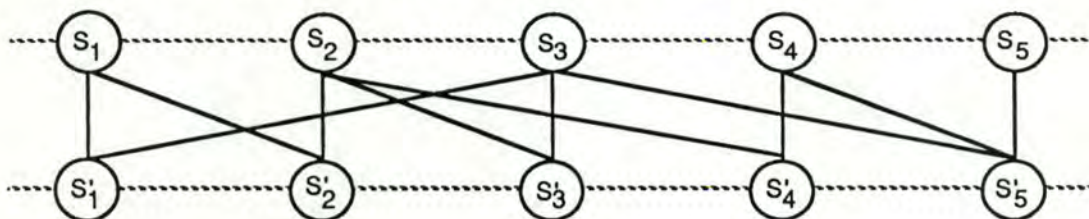


figure VII.5

Le calcul des abscisses du deuxième niveau consiste à faire la moyenne des abscisses des sommets du premier niveau auquel ils sont reliés. Par exemple, pour le graphe de la figure VII.5, le calcul des abscisses du deuxième niveau, connaissant les abscisses des sommets du premier niveau (repris à la figure VII.6), résulte des calculs suivants :

$$\text{ABSC de } s'_1 = 1/2 \cdot (\text{ABSC de } s_1 + \text{ABSC de } s_3) = 1/4$$

$$\text{ABSC de } s'_2 = 1/2 \cdot (\text{ABSC de } s_1 + \text{ABSC de } s_2) = 1/8$$

...

Les résultats de ces calculs se trouvent dans le tableau de droite de la figure VII.6 et la représentation graphique qui en découle est reprise à la figure VII.7.

S	ABSC
s_1	0
s_2	1/4
s_3	1/2
s_4	3/4
s_5	1

premier niveau

S'	ABSC
s'_1	1/4
s'_2	1/8
s'_3	3/8
s'_4	1/2
s'_5	3/4

deuxième niveau

figure VII.6

On peut constater que le passage de la figure VII.5 à la figure VII.7 améliore le dessin puisque l'on n'a plus que 5 croisements dans le second cas au lieu de 6 dans le premier.

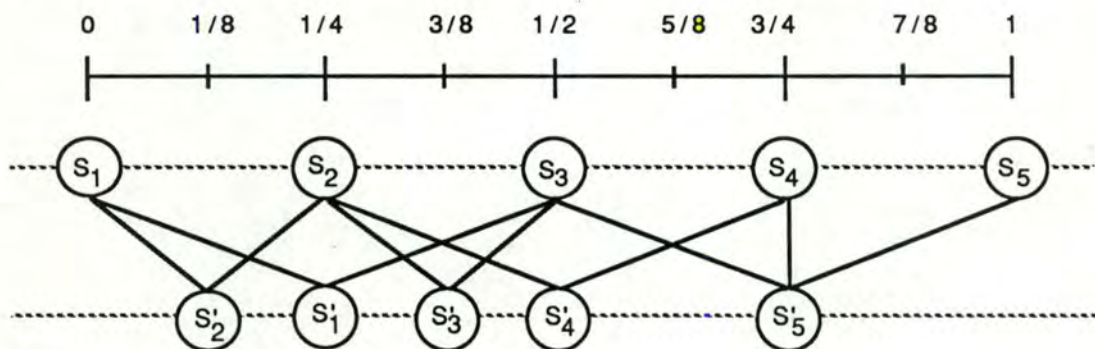


figure VII.7

La généralisation de cette méthode à plusieurs niveaux pose un certain nombre d'inconvénients :

- Il n'y a pas d'optimisation globale dans la recherche du nombre minimum de croisements. En fait, la méthode se borne à minimiser le nombre de croisements localement entre deux niveaux successifs sans tenir compte des répercussions de ce choix sur l'ensemble de la représentation.
- Cette méthode peut produire deux valeurs d'abscisses identiques pour deux sommets d'un même niveau.
- Pour certains graphes, représentables sans croisement mais qui contiennent beaucoup de sommets "suspendus" (c'est-à-dire des sommets qui apparaissent à des niveaux intermédiaires et qui sont sans précédent) et un mauvais maillage, l'heuristique peut donner de très mauvais résultats.

A ces inconvénients énoncés par Delarche, il faut ajouter que sa méthode accepte difficilement des contraintes de dispositions supplémentaires, comme, par exemple imposer qu'un sommet se trouve à côté d'un autre, ce qui la rend inapplicable pour la production automatique du schéma de la dynamique où cette contrainte existe (cf. chap VIII).

VII.2.2. Notre solution

La solution que nous proposons pour générer automatiquement un graphe hiérarchisable et par conséquent sans circuit qui respecte les critères esthétiques énoncés au point VII.2 comprend les 5 étapes suivantes :

- <1> Construction de la hiérarchie des rangs
 - <2> Serrer la hiérarchie
 - <3> Rendre la hiérarchie propre
 - <4> Minimiser le nombre de croisements
 - <5> Calcul des coordonnées (X,Y) des sommets du graphe
- <1> Disposant d'un graphe orienté sans circuit (cf. figure VII.8), cette étape consiste à attribuer à chacun des sommets un numéro de niveau qui correspond au rang de ce dernier augmenté de 1 . Le rang d'un sommet s représente la longueur maximum des chemins dont s est l'extrémité terminale .

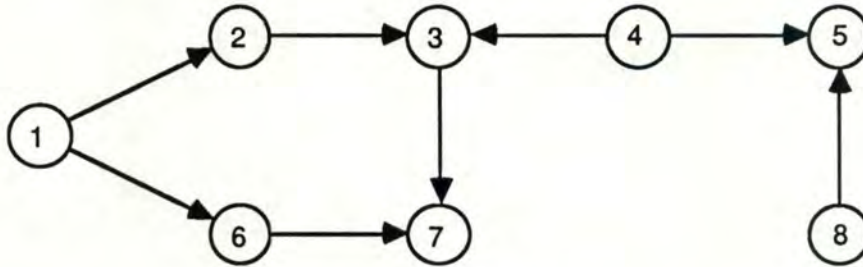


figure VII.8 graphe sans circuit

La hiérarchisation du graphe de la figure VII.8 est reprise à la figure VII.9. A titre d'illustration, on peut constater sur la première figure, que le sommet 3 est l'extrémité terminale de deux chemins : (x_1, x_2, x_3) et (x_4, x_3) qui ont une longueur maximale de 3, ce qui explique la présence du sommet x_3 au troisième niveau à la figure VII.9.

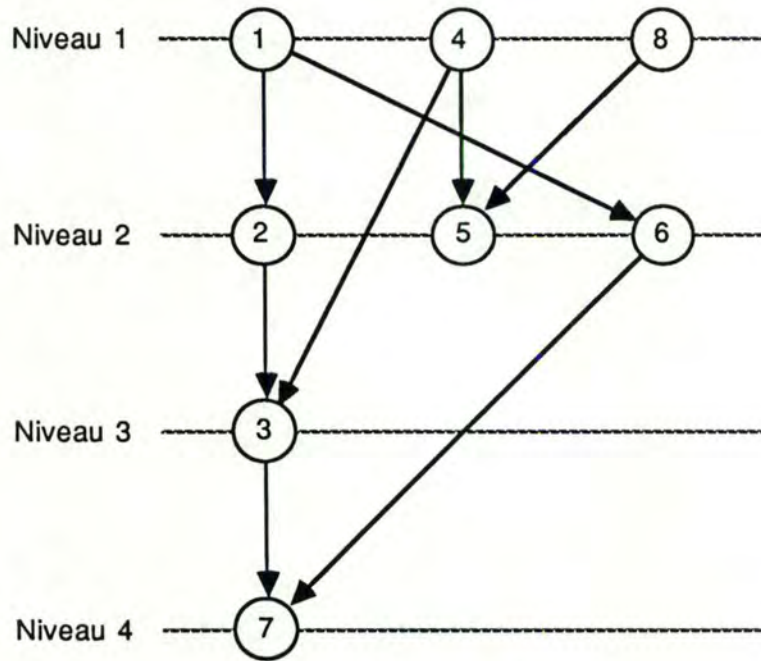


figure VII.9 graphe hiérarchisé

- <2> La deuxième étape consiste à serrer la hiérarchie en minimisant la portée de tous les arcs. La figure VII.10 représente une telle hiérarchie. On peut en effet constater sur cet exemple que la portée de l'arc (x_4, x_3) est passée de 2 pour le graphe de la figure VII.9 à 1 pour celui de la figure suivante et que la portée des autres arcs est restée inchangée pour les deux figures.

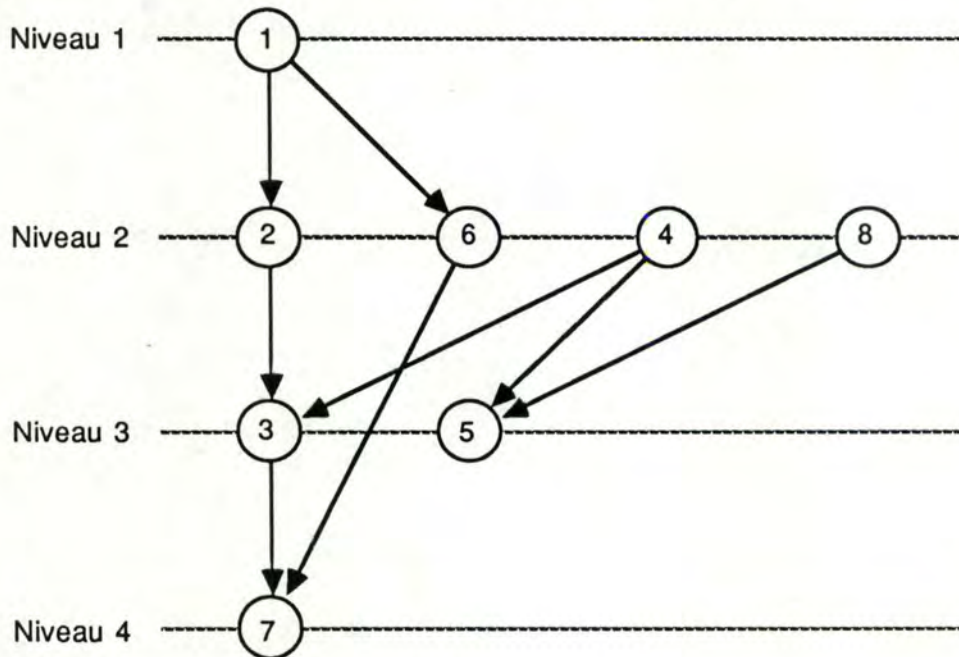


figure VII.10 hiérarchie serrée

- <3> Pour pouvoir minimiser le nombre de croisements, il faut que tous les arcs aient une portée de 1, autrement dit, il faut que la hiérarchie soit propre. La technique employée à cet effet consiste à ajouter des sommets "bidons" comme illustré à la figure VII.11. On peut en effet constater sur ce graphe l'ajout du sommet X_9 par rapport au graphe de la figure précédente ce qui rend la hiérarchie propre.

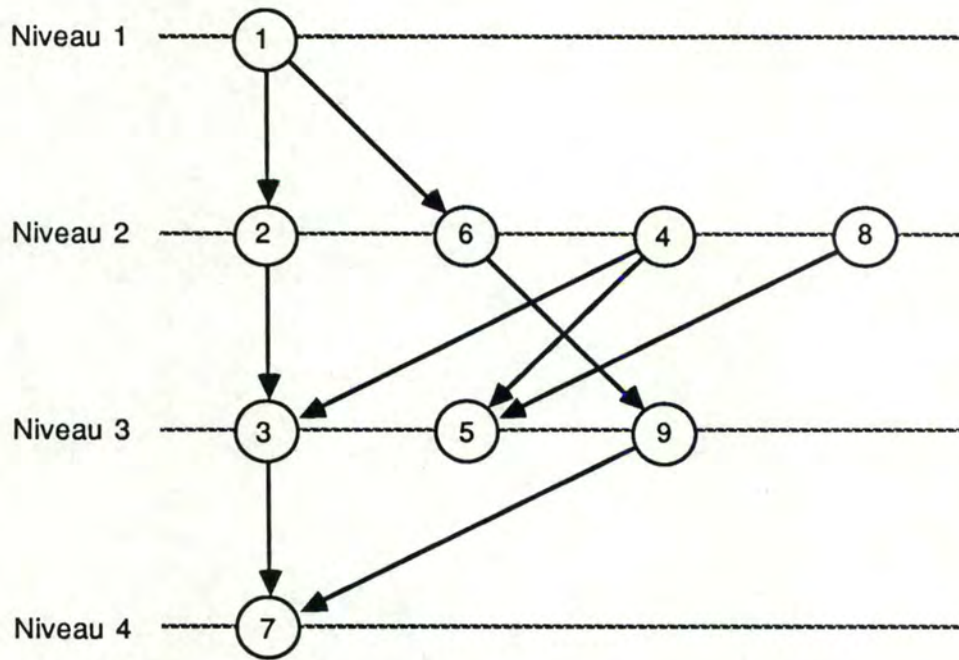


figure VII.11 hiérarchie propre

<4> Disposant d'une hiérarchie propre il est maintenant possible de minimiser le nombre de croisements. Ainsi, par exemple, la figure VII.12 présente le graphe de la figure VII.11 où l'on a éliminé tous les croisements

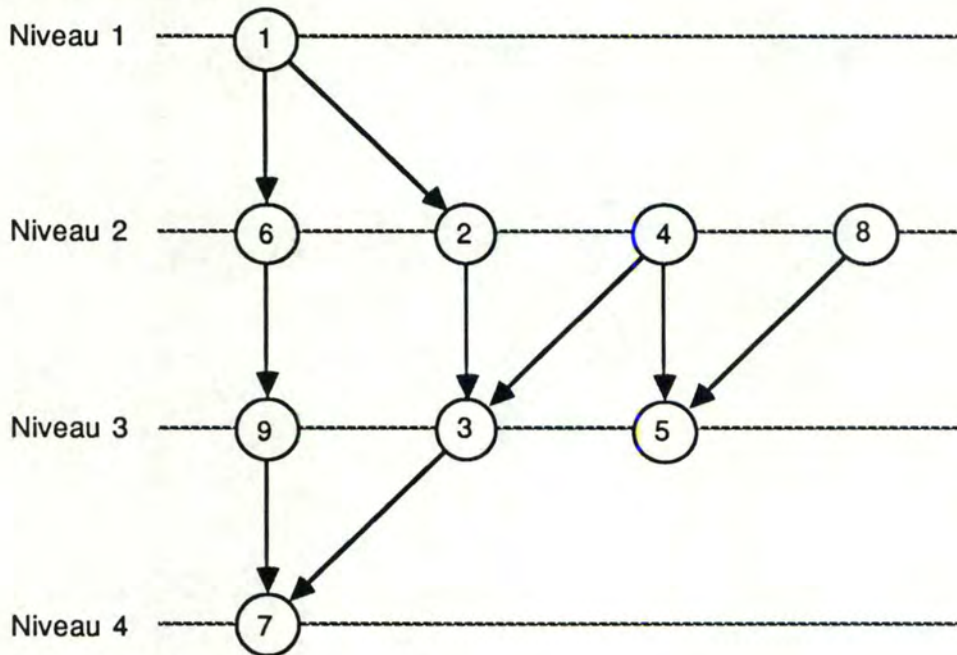


figure VII.12 graphe sans croisement

<5> La dernière étape consiste à déterminer les coordonnées (X,Y) des sommets du graphe. Les coordonnées verticales des sommets s'obtiennent facilement quand on connaît les niveaux auxquels appartiennent les sommets. Le calcul des coordonnées horizontales doit quant à lui placer les sommets de telle manière à présenter un dessin équilibré et le plus vertical possible tout en respectant l'ordre des sommets établi aux étapes précédentes.

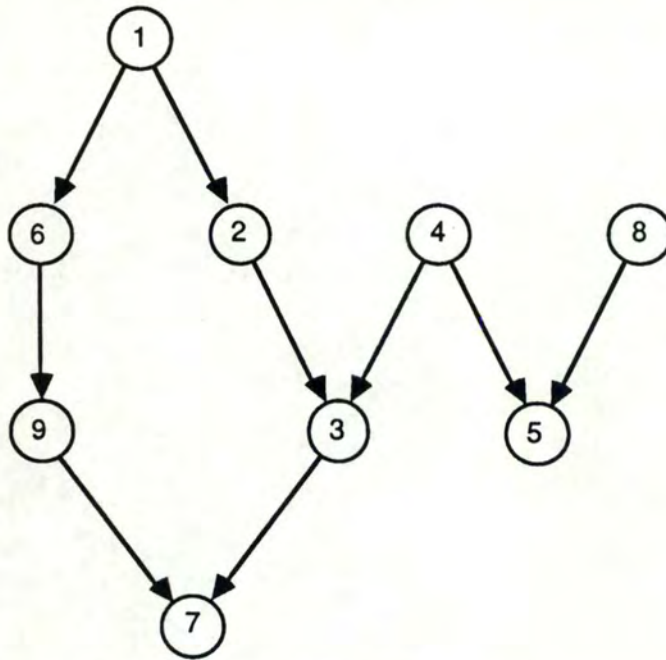


figure VII.13 placement des sommets

En conclusion, la première étape permet le dessin d'un graphe hiérarchisé. Les trois étapes suivantes ont pour objectif final la minimisation du nombre de croisements. La dernière étape assure quant à elle une répartition homogène des sommets.

VII.3 METHODE DE PRODUCTION AUTOMATIQUE DE GRAPHES HIERARCHISES

Chaque étape de la méthode de production automatique de graphes hiérarchisés présentée intuitivement au point VII.2.2. sera détaillée ici.

Le problème sera à chaque étape spécifié et nous donnerons une ou plusieurs solutions possibles. De plus, pour la réduction du nombre de croisements, nous ferons un exposé succinct de la littérature existant sur le sujet.

VII.3.1 CONSTRUCTION D'UNE HIERARCHIE DES RANGS

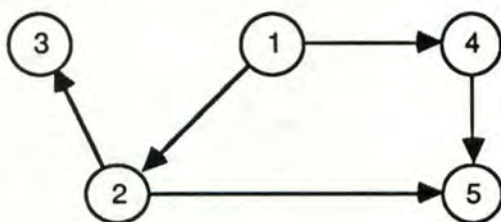
L'idée de l'algorithme que nous proposons ici pour construire une hiérarchie des rangs se trouve dans la thèse de doctorat de [Delarche, 79].

Le problème de la construction d'une telle hiérarchie sera d'abord spécifié. Nous présenterons ensuite intuitivement la méthode proposée par Delarche. Pour terminer, nous donnerons l'algorithme complet résolvant le problème.

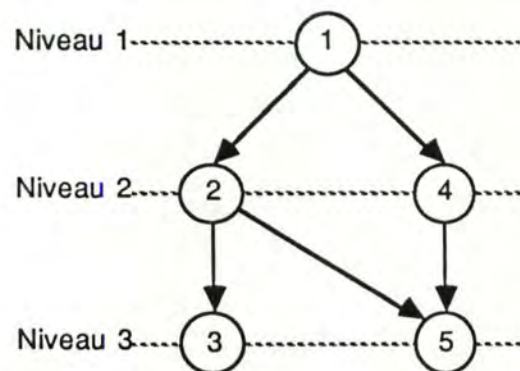
VII.3.1.1. Spécification

D'après la définition que nous avons donnée au point VII.1.1.2 d'une hiérarchie des rangs, l'exécution de cet algorithme doit avoir pour effet d'attribuer à chacun des sommets du graphe, un numéro de niveau qui correspond au rang de ce dernier augmenté de 1.

Par exemple, si l'algorithme reçoit en entrée le graphe de la figure VII.14 (a), il produit en sortie la hiérarchie de la figure VII.14 (b).



Graphe orienté



Hiérarchie des rangs

Figure VII.14

Une exécution correcte de l'algorithme nécessite que le graphe de départ ne possède **pas de circuit**. En effet, si tel était le cas on n'arriverait jamais à attribuer un numéro de rang aux sommets faisant partie d'un circuit.

VII.3.1.2. Réalisation

IDEE INTUITIVE

L'idée consiste à marquer 1 tous les sommets sans précédent (il en existe au moins un puisque le graphe est sans circuit). Ensuite, on marque 2 tous les suivants des sommets marqués 1, puis on marque 3 les suivants des sommets marqués 2, etc... On arrête le marquage lorsqu'il n'y a plus de marques nouvelles à ajouter. Comme le graphe est sans circuit et qu'il comporte un nombre fini d'arcs, la terminaison de l'algorithme est assurée.

ALGORITHMME

```
begin
  niv := 0
  L := sommets sans précédent
  while L ≠ vide do
    niv := niv + 1
    L-inter := vide
    for all s: s sommet appartenant à L do
      attribuer à s le niveau niv
      L-inter := L-inter U {s' tel que s' est un suivant de s }
    endfor
    L := L-inter
  endwhile
end
```

où L représente une liste de sommets à marquer
L-inter représente une liste intermédiaire de sommets

EVALUATION

le nombre de marquages effectués étant égal au nombre d'arcs, si n est le nombre de sommets du graphe, la complexité de l'algorithme est de l'ordre de n^2 .

VII.3.2. CONSTRUCTION D'UNE HIERARCHIE SERREE

[Delarche, 79] propose dans sa thèse de doctorat une méthode pour la construction d'une hiérarchie serrée qui nous a servi de base pour la réalisation de l'algorithme correspondant.

Le problème de la construction d'une telle hiérarchie sera d'abord spécifié. Nous présenterons ensuite intuitivement la méthode proposée par Delarche. Pour terminer, nous donnerons l'algorithme complet résolvant le problème.

VII.3.2.1. Spécification

D'après la définition que nous avons donnée au point II.2 d'une hiérarchie serrée, l'exécution de l'algorithme doit conduire à une représentation du graphe telle que la portée de tous les arcs est minimale tout en respectant l'ordre des rangs.

En guise d'illustration, on peut constater sur la carte de gauche de la figure VII.15 que la portée de tous les arcs est 1 à l'exception de l'arc qui relie le sommet 2 au sommet 5 qui a une portée de 2. Sur la carte de droite par contre la portée de tous les arcs vaut 1 et est donc minimale.

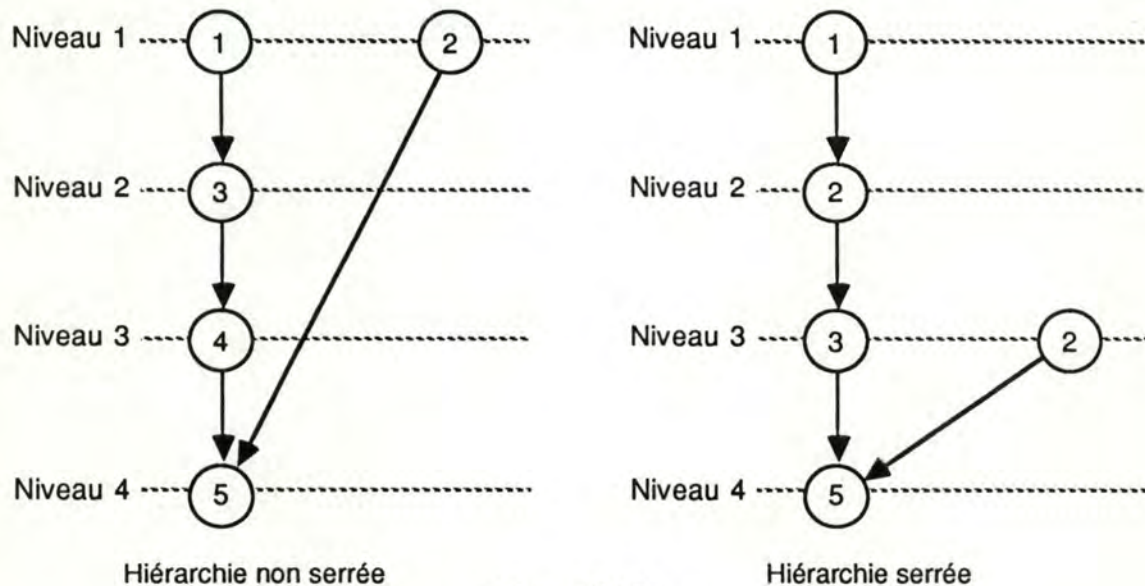


Figure VII.15

VII.3.2.2. Réalisation

IDEE INTUITIVE

L'algorithme de construction d'une hiérarchie serrée procède en trois étapes :

- <1> calcul du tableau DES tel que $DES(s)$ est le nombre maximum de niveaux dont on peut descendre le sommet s sans enfreindre la compatibilité entre les niveaux et l'orientation des arcs.
- <2> modification de DES pour ne jamais descendre un sommet plus qu'un de ses précédents, ce qui allongerait au moins un arc.
- <3> calcul du niveau définitif de chaque sommet.

<1> Première étape calcul du tableau DES

Cette étape consiste à construire le tableau DES tel que $DES(s)$ est le nombre maximum de niveaux dont le sommet s peut descendre tout en restant au-dessus de ses suivants.

Par exemple, l'application de la première étape au graphe de la figure VII.16, fournit le tableau DES présenté à la même figure. On peut constater que les sommets 1, 2, 3, 7 appartenant à un chemin maximum ne peuvent pas être descendus sans enfreindre l'ordre des rangs. Par contre, comme le sommet 5 n'a pas de suivant, il peut être descendu jusqu'au dernier niveau sans problème.

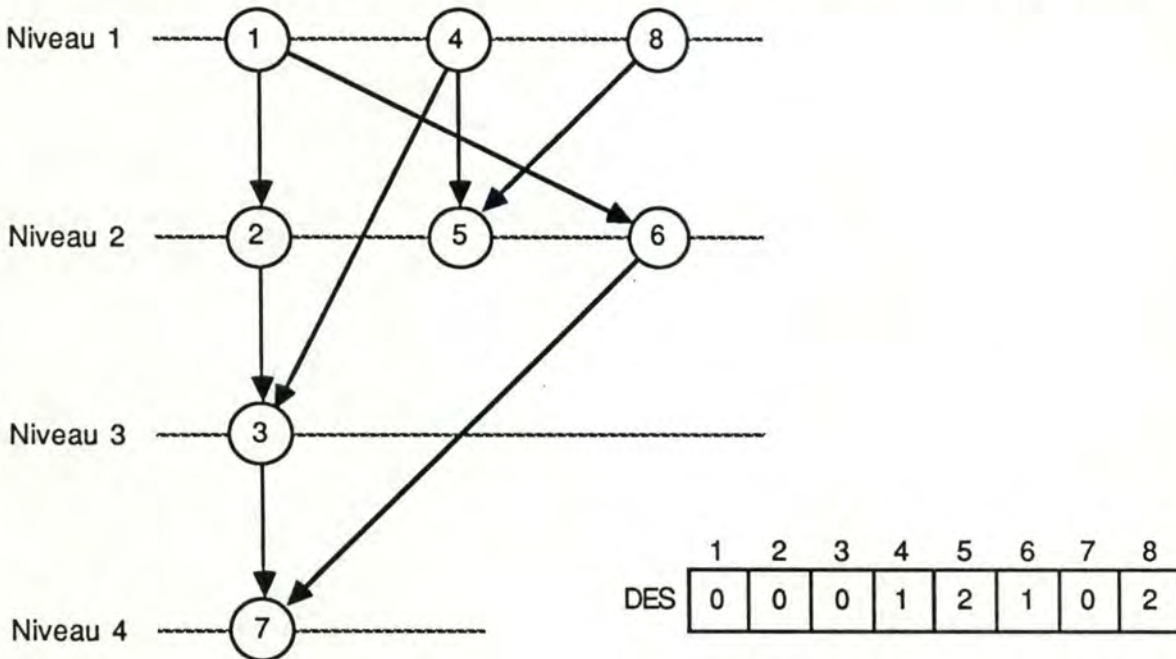


Figure VII.16 première étape

Une manière de faire consiste à initialiser dans un premier temps DES avec des valeurs très grandes ($+\infty$) et à parcourir ensuite le graphe par niveaux en partant du niveau le plus élevé.

Lors du traitement d'un sommet s de niveau niv , deux cas sont possibles :

<1> s n'a pas de suivant et peut, par conséquent, être descendu jusqu'au dernier niveau :

$$DES(s) = NIVMAX - niv \quad (E1)$$

<2> s a au moins un suivant et dans ce cas, comme nous allons le voir, $DES(s)$ est déjà fixé .

Le traitement se poursuit en ajustant $DES(p)$ pour chaque précédent p du sommet s , de manière à ne pas descendre p plus bas que s :

$$DES(p) = \text{Min} [DES(p), DES(s) + NIV(s) - NIV(p) - 1] \quad (E2)$$

Ceci explique pourquoi, lorsqu'un sommet s possédant au moins un suivant devient courant, $DES(s)$ est déjà fixé. En effet, si tel est le cas, $DES(s)$ a été ajusté lors du traitement de ses suivants afin que le sommet s ne descende pas plus bas qu'eux.

<2> Deuxième étape modification du tableau DES

Cette étape consiste à modifier DES pour que chaque sommet s ne descende pas d'un plus grand nombre de niveaux qu'un de ses précédents, ce qui allongerait des arcs.

Par exemple, en descendant le sommet 4 d'un niveau et le sommet 5 de deux niveaux, on allonge l'arc (4, 5).

Pour éviter cela, il faut modifier $DES(s)$ pour vérifier l'égalité suivante:

$$DES(s) = \text{MIN}_p \text{ appartenant aux précédents de } s [DES(p)] \quad (E3)$$

Par exemple, comme $DES(5) = \text{MIN} [1, 2] = 1$, le sommet 5 ne peut descendre que d'un niveau malgré qu'on l'avait descendu de 2 niveaux dans la première étape.

Mais cela ne suffit pas, il faut également transmettre les modifications aux précédents de s pour que la hiérarchie des rangs soit respectée. En effet, si l'on descend les sommets en respectant le tableau DES calculé à la première étape, on obtient la carte de la figure VII.17. Si on applique uniquement E3, le sommet 5 va, comme nous l'avons dit, remonter d'un niveau. On peut constater sur la figure VII.17 que, dans ce cas, la hiérarchie des rangs n'est plus respectée puisque les sommets 5 et 8 sont reliés par un arc et se trouvent au même niveau.

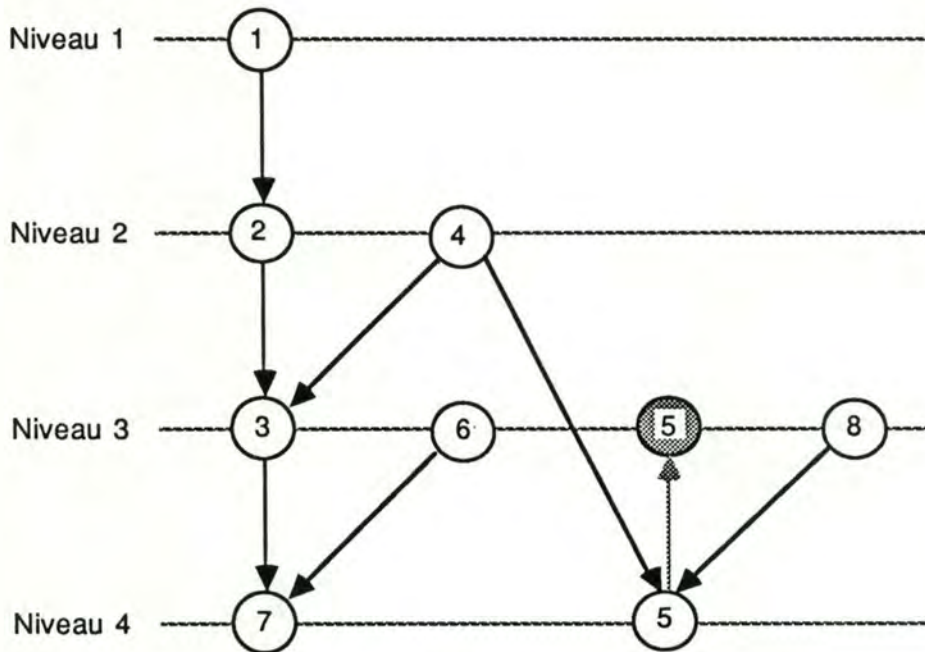


Figure VII.17 application de (E3)

Pour éviter cette situation il suffit d'appliquer (E2) à chaque précédent du sommet s . Ainsi, par exemple, en appliquant E2 au sommet 8, celui-ci est remonté d'un niveau (par rapport à la première étape), ce qui rétablit la hiérarchie des rangs.

On voit ainsi apparaître la nécessité de parcourir tous les "zigzags" du graphe pour ajuster DES au minimum

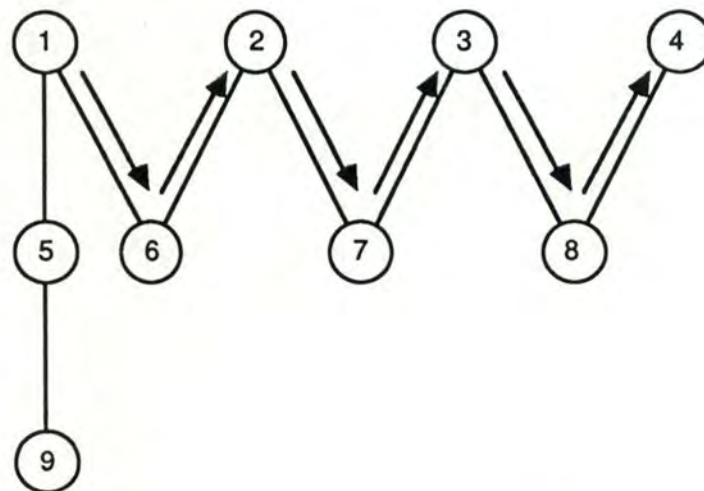


Figure VII.18 parcours en "zigzags"

En guise d'illustration, sur le graphe de la figure VII.18, si, au terme de la première étape, on a : $DES(1) = DES(5) = DES(9) = 0$ et que toutes les autres valeurs de DES sont nulles, il va falloir imposer que $DES(6)=0$ puisque $DES(1)=0$, puis transmettre la modification le long du "zigzag" : $DES(2)=0$, $DES(7)=0$ etc...jusqu'à $DES(4)=0$.

Cette modification se fera par une succession de parcours par niveaux alternativement descendants et ascendants en marquant à chaque parcours tous les sommets pour lesquels une modification de DES a eu lieu jusqu'à ce que toutes ces modifications nécessaires aient été faites.

Si un sommet est marqué D, il faut vérifier que tous ses suivants ne descendent pas d'un plus grand nombre de niveaux que lui. Si un sommet est marqué M, il faut vérifier que tous ses précédents se trouvent bien au-dessus de lui.

Parcours descendant

Le parcours descendant vérifie que les suivants s' d'un sommet s marqué D ne descendent pas plus que lui, et par conséquent

$$DES(s') = \text{MIN} [DES(s'), DES(s)]$$

Les sommets pour lesquels une modification a été nécessaire sont marqués D et M pour répercuter par la suite la modification vers le bas et vers le haut.

Par exemple, en supposant qu'initialement tous les sommets de la figure VII.19 soient marqués D, le parcours descendant effectue les changements suivants du tableau DES :

- au niveau 1 :

$DES(1) = 0 < DES(6) = 1$, par conséquent, $DES(6) := 0$, et 6 est marqué avec M et D.

$DES(4) = 1 < DES(5) = 2$, par conséquent, $DES(5) := 1$, et 5 est marqué avec M et D.

- au niveaux 2, 3, et 4 il n'y a aucune modification de DES

A la fin de ce parcours descendant, on est dans la situation décrite à la figure VII.19

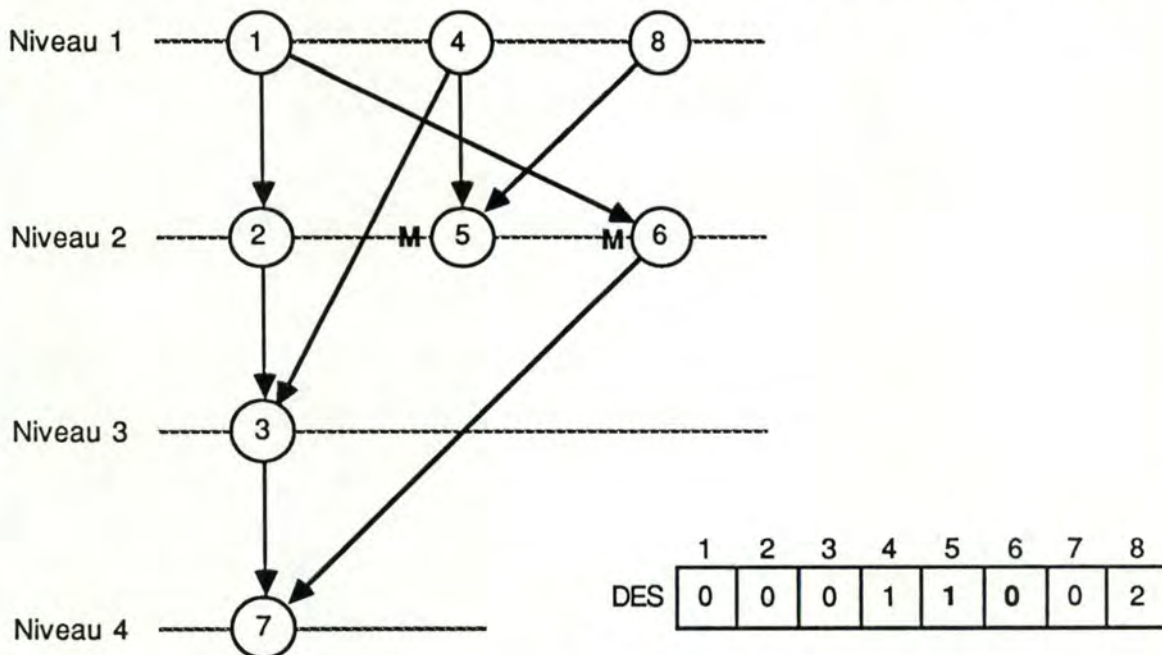


Figure VII.19 fin du parcours descendant

Parcours ascendant

Le parcours ascendant (ou montant) quant à lui vérifie que la hiérarchie des rangs est toujours respectée, c'est-à-dire que pour un sommet s marqué M, tous ses précédents p sont tels que :

$$DES(p) = \text{MIN} [DES(p), DES(s) + \text{NIV}(s) - \text{NIV}(p) - 1]$$

Les sommets pour lesquels DES a été ajusté sont marqués M et D afin de répercuter plus tard les changements vers le haut et vers le bas.

Par exemple, partant du tableau DES de la figure VII.19, le parcours ascendant effectue les changements suivants :

- aux niveaux 4 et 3 il n'y a aucun sommet marqué M

- au niveau 2 :

le sommet 5 est marqué M, on envisage donc ses précédents :

DES(4) est inchangé

DES(8) = 2 > (1+2-1-1), par conséquent, DES(8) est ajusté à 1, et le sommet 8 est marqué avec M et D.

le sommet 6 est marqué M mais n'implique aucun changement.

- au niveau 1, le sommet 8 est marqué M mais n'implique aucun changement.

A la fin de ce parcours ascendant, on est dans la situation décrite à la figure VII.20

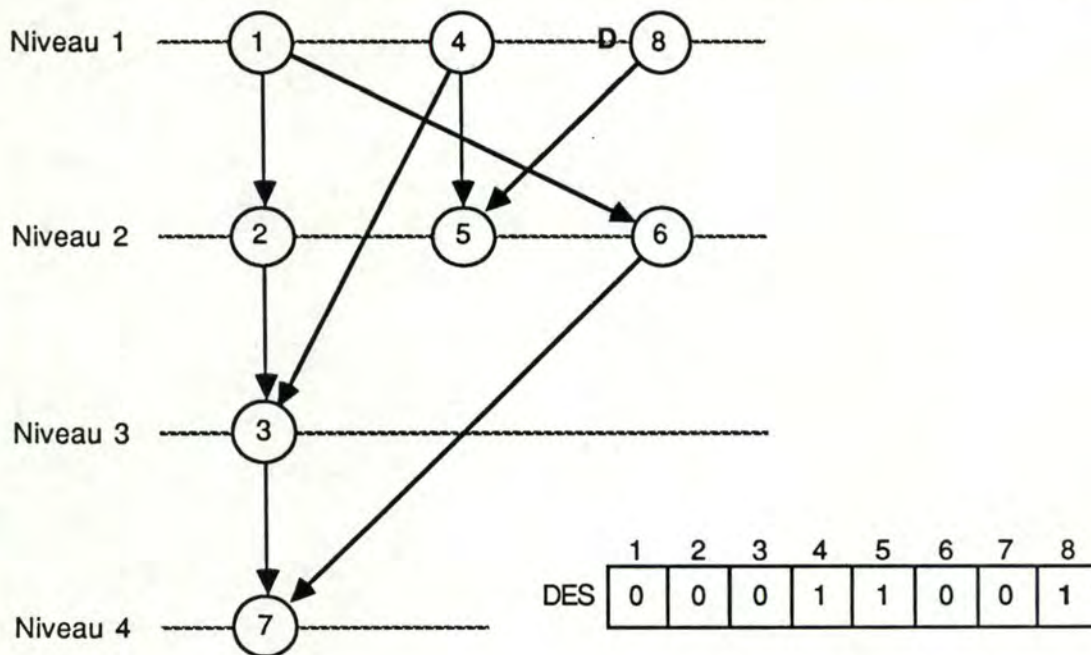


Figure VII.20 fin du parcours ascendant

Programme principal suite de parcours descendants et montants

La deuxième étape alterne les parcours descendants et montants tant qu'il y a des modifications, c'est-à-dire de nouvelles marques posées par l'un de ces parcours.

L'application de la deuxième étape sur l'exemple de la figure fournit le tableau DES ci-dessous :

	1	2	3	4	5	6	7	8
DES	0	0	0	1	1	0	0	1

<3> Troisième étape calcul des niveaux définitifs

En effectuant pour tout sommet s l'opération $NIV(s) := NIV(s) + DES(s)$, on obtient le niveau définitif de chaque sommet de sorte que, sur l'exemple de la figure VII.20, on obtient la hiérarchie de la figure VII.21.

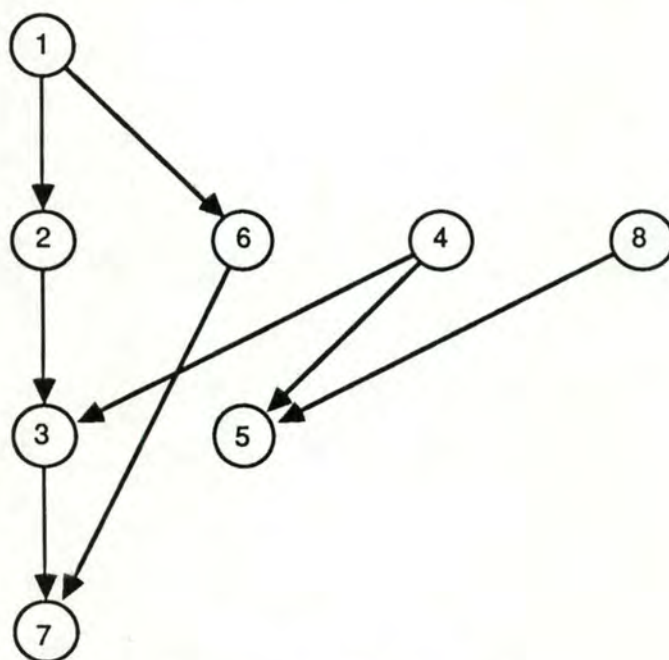


Figure VII.21 hiérarchie serrée

ALGORITHME**<1> Première étape** Calcul du tableau DES.

```

begin
  for all s: sommet do
    DES[s] := NIVMAX + 1; (+∞)
  endfor
  for niv:=NIVMAX-1 downto 1 do
    for all s: sommet de niveau niv do
      begin
        if (s n'a pas de suivant)
          then DES(s) := NIVMAX - niv
          for all p: précédent de s do
            DES(p) = Min [DES(p), DES(s) + niv - NIV(p) - 1]
          endfor
        end
      end
    endfor
  endfor
end

```


<2> Deuxième étape *Modification du tableau DES.**Parcours descendant*

```

begin
  modif := FALSE
  for niv:=1 to NIVMAX-1 do
    for all s: sommet de niveau niv et marqué D do
      enlever la marque D
      for all s': suivant de s do
        if DES[s'] > DES[s]
          then
            begin
              DES[s'] := DES[s]
              marquer s' avec M et D
              modif := TRUE
            end
          endif
        endfor
      endfor
    endfor
  end
end

```

Parcours ascendant

```

begin
  modif := FALSE
  for niv:= NIVMAX downto 2 do
    for all s: sommet de niveau niv et marqué M do
      enlever la marque M
      for all p: précédent de s do
        aux := DES[s] + niv - niv[p] - 1
        if DES[p] > aux
          then
            begin
              DES[p] := aux
              marquer p avec M et D
              modif := TRUE
            end
          endif
        endfor
      endfor
    endfor
  end
end

```

*Programme principal**suite de parcours descendant et montant*

```

begin
  for all s: sommet do marquer D le sommet s
  modif := TRUE
  while modif do
    begin
      parcours descendant
      if modif then parcours montant
    endwhile
  endfor
end

```

<3> Troisième étape *Calcul des niveaux définitifs*

```
begin
  for all s: sommet do
    NIV(s) := NIV(s) + DES(s)
  endfor
end
```

EVALUATION

L'algorithme que nous venons de décrire (si l'on ne tient pas compte des étapes 1 et 3 qui nécessitent chacune un parcours par niveau) a une complexité de l'ordre de $n \cdot (n_{niv})^2$ où n représente le nombre total de sommets et n_{niv} le nombre maximum de sommets à un niveau.

En effet, chaque parcours par niveau considère tous les sommets et envisage pour chacun de ceux-ci l'ensemble de leurs suivants. Comme la hiérarchie est propre, un sommet a au maximum n_{niv} suivants ; par conséquent, un parcours ascendant ou descendant est de l'ordre de $n \cdot (n_{niv})$. Or, on doit faire moins de n_{niv} parcours, puisque le nombre de parcours est déterminé par le plus long "zigzag" existant dans le graphe qui, de toute façon, comprend moins de n_{niv} sommets.

VII.3.3. CONSTRUCTION D'UNE HIERARCHIE PROPRE

Les méthodes proposées au point VII.3.4 pour réduire le nombre de croisements fonctionnent uniquement si la hiérarchie est propre.

Le problème de la construction d'une telle hiérarchie sera d'abord spécifié. Nous présenterons ensuite intuitivement l'algorithme que propose [Warfield, 77]. Pour terminer, nous donnerons l'algorithme complet résolvant le problème.

VII.3.3.1. Spécification

Nous avons vu au point VII.1.1.2. qu'une hiérarchie propre est une hiérarchie dans laquelle tous les arcs ont une portée égale à un. L'objet de l'algorithme est de construire une telle hiérarchie.

Par exemple, si le graphe initial correspond à la figure VII.21 (a), la hiérarchie produite est celle qui est représentée à la figure VII.21 (b) où les sommets gris constituent des sommets supplémentaires.

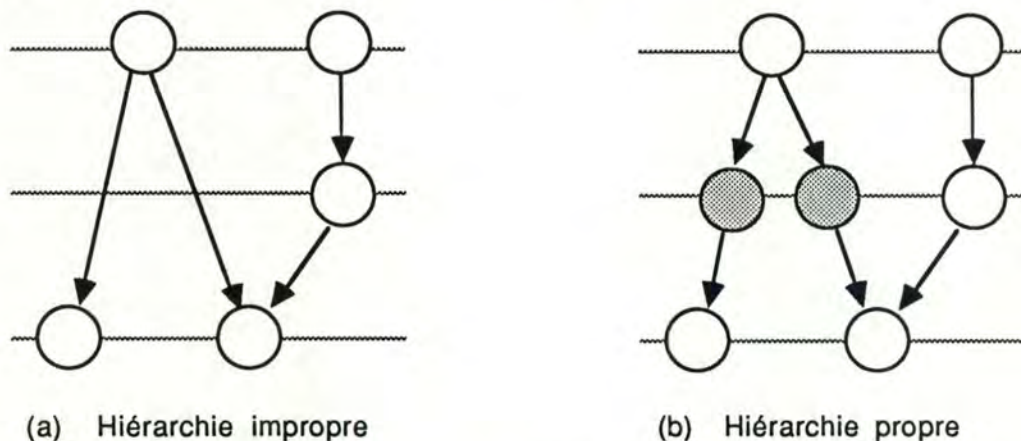


Figure VII.22

VII.3.3.2. Réalisation

IDEE INTUITIVE

Lorsqu'une hiérarchie est impropre, Warfield propose de la remplacer par une hiérarchie propre, simplement en introduisant des **sommets supplémentaires** de manière telle que la portée de tous les arcs soit égale à 1.

L'idée de l'algorithme consiste à parcourir les niveaux de la hiérarchie en partant du premier. Supposons que l'on ait examiné les $l-1$ premiers niveaux et que l'on doive traiter le niveau l .

Pour chaque sommet s du niveau l , on teste si tous ses suivants appartiennent bien au niveau $(l+1)$. S'il existe un suivant s' qui ne vérifie pas cette condition, la hiérarchie est impropre. Dans ce cas, on ajoute un sommet "bidon" b au niveau $(l+1)$ et on remplace l'arc (s,s') par les deux arcs (s,b) et (b,s') .

ALGORITHMME

```
begin
  for niv:=1 to NIVMAX-1 do
    for all s: sommet de niveau niv do
      for all s': suivant de s do
        if NIV(s)' > niv +1
          then
            { ajouter un sommet b bidon au niveau niv +1,
              enlever l'arc (s,s'), et
              créer un arc (s,b) et un arc (b,s') }
          endif
        endfor
        niv := niv + 1
      endfor
    endfor
  end
```

où NIVMAX désigne la longueur de la hiérarchie
NIV(s)' désigne le niveau du sommet s'

EVALUATION

l'algorithme parcourt une fois tous les arcs. Par conséquent, si n est le nombre de sommets du graphe, la complexité de l'algorithme est de l'ordre de n^2 .

VII.3.4. MINIMISATION DU NOMBRE DE CROISEMENTS

Le problème de la planarisation d'un graphe non orienté a fait l'objet de nombreuses recherches. Les maigres résultats obtenus ne sont cependant pas applicables aux graphes hiérarchisés. A notre connaissance, deux chercheurs se sont intéressés à ce dernier problème mais les résultats sont peu satisfaisants.

En ce qui nous concerne, nous avons réussi à formaliser le problème sous la forme d'un programme linéaire mais qui nécessite l'emploi de variables binaires, ce qui rend son utilisation pratique impossible. Nous proposons également une heuristique qui, comme nous le démontrerons, minimise le nombre de croisements lorsque le graphe est représentable sans croisement et qui donne une solution vraisemblablement non optimale dans les autres cas.

L'heuristique proposée est particulièrement intéressante pour les schémas qui seront produits par notre méthode puisqu'ils sont généralement représentables sans croisement ou tout au moins avec peu de croisements.

VII.3.4.1. Spécification

L'objet de cette étape consiste à réduire le nombre de croisements dans un graphe qui représente une hiérarchie propre.

Comme nous le verrons par la suite, ce problème revient à déterminer un ordre des sommets à chacun des niveaux.

VII.3.4.2. Existant

1. Le problème général de planarisation

Savoir si un graphe est planaire, c'est-à-dire susceptible d'être dessiné dans le plan sans que ses arêtes ne se coupent est un problème bien connu de la théorie des graphes.

[Hopcroft, 74] propose un algorithme pour tester la planarité d'un graphe. Son temps d'exécution est une fonction linéaire du nombre de sommets que contient le graphe, mais la représentation qu'il suggère si le graphe est planaire, est obtenue après plusieurs tentatives éventuellement infructueuses (recherche en profondeur plus backtracking).

[Rubin, 75] expose une amélioration de l'algorithme de Hopcroft et Tarjan afin d'éviter, autant que possible, les retours en arrière (backtracking). L'implémentation ainsi obtenue est dans la plupart des cas deux fois plus rapide que la solution précédente.

Plus récemment, [Yeh, 82] présente des algorithmes qui tentent de trouver une représentation planaire du graphe. Ces algorithmes présentent l'avantage de donner directement la représentation du graphe si celui-ci est planaire.

Dans le cas des graphes non planaires, le problème de la minimisation du nombre de croisements reste ouvert. Ce problème a toutefois été démontré comme étant NP-complet [Tamassia, 83] et par conséquent il n'est vraisemblablement pas résolvable par un algorithme de complexité linéaire [Karp, 75].

Sur le plan pratique, vu son importance, le problème a donné lieu à de nombreuses études, en particulier pour la conception des circuits électroniques, mais également dans le domaine de la production automatique de diagrammes, comme, par exemple, le diagramme de flux [Batini, 86] ou le schéma Entité/Association [Tamassia, 83].

Pour ce dernier graphique, l'heuristique proposée procède en deux temps :

- <1> Décomposition du graphe initial en un sous-graphe planaire maximal et un ensemble d'arcs irréductibles.
- <2> Introduction des arcs irréductibles dans le sous-graphe. Cette procédure qui reconstruit le graphe initial à partir du sous-graphe planaire ne garantit pas le minimum de croisements.

Toutes ces études s'avèrent cependant inapplicables à notre problème de dessin de graphes hiérarchisés étant donné qu'elles concernent exclusivement les graphes non orientés.

2. Problèmes spécifiques des graphes hiérarchisés

Comme le souligne [Delarche, 79], pour les graphes hiérarchisés, le problème est à la fois plus simple et plus compliqué.

Plus simple parce que l'obligation de respecter certaines contraintes de disposition (en particulier, les sommets du niveau i se trouvent en dessous des sommets du niveau $(i-1)$) diminue la part d'arbitraire sur le placement des sommets.

Plus compliqué car une solution optimale dans le cas d'un graphe non orienté n'est pas nécessairement une solution optimale dans le cas d'un graphe orienté car elle peut ne pas respecter les contraintes de disposition.

Par exemple, la représentation de la figure VII.23 ne respecte pas les conventions de représentation graphique d'une hiérarchie.

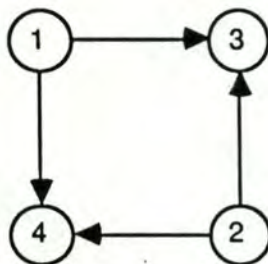


Figure VII.23

3. Solutions existantes

A notre connaissance, deux auteurs se sont intéressés au problème de la représentation d'une hiérarchie avec un minimum de croisements.

[Delarche, 79] propose une heuristique appelée la **méthode des degrés relatifs** qui a été exposée brièvement au point VII.2.1. et qui présente, comme nous l'avons souligné, des inconvénients majeurs la rendant inapplicable aux graphiques qui nous intéressent.

[Warfield, 77] a élaboré une théorie sur les croisements ainsi qu'un ensemble de techniques permettant de réduire le nombre de croisements sans pour autant donner un algorithme qui résout le problème. Cet article constitue néanmoins la base des méthodes que nous proposons.

VII.3.4.3. Réalisation

1. PREALABLE

1.1. Matrice d'interconnexion

On peut représenter un graphe à deux niveaux par une **matrice d'interconnexion** M dans laquelle

- les indices lignes représentent les sommets du premier niveau,
- les indices colonnes représentent les sommets du deuxième niveau,
- un élément $M(i,j) = 1$ s'il existe un arc reliant le sommet s_i au sommet s_j .

La matrice d'interconnexion du graphe de la figure VII.24 est représentée ci-dessous :

	4	5	6
1	1	1	1
2	1	0	0
3	1	0	0

Matrice d'interconnexion

1.2. Calcul du nombre de croisements entre deux niveaux successifs

Supposons que l'on veuille calculer le nombre de croisements sur la carte de la figure VII.24.

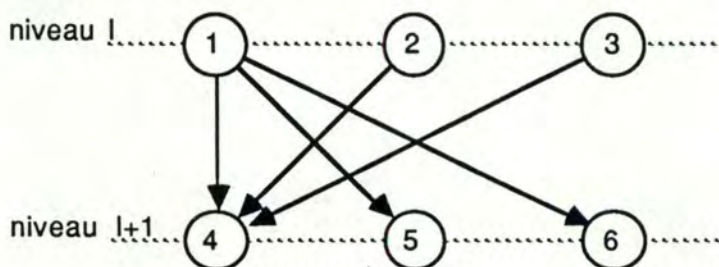


figure VII.24

L'idée proposée par [Warfield, 77] consiste à prendre tous les couples possibles de sommets du niveau l . Dans notre exemple, il s'agit des couples $(1,2)$, $(1,3)$, $(2,3)$ et pour chacun d'eux, par exemple $(1,2)$, de calculer le nombre de croisements (noté $K(1,2)$) que ces sommets induisent, c'est-à-dire dans notre exemple, le nombre de croisements entre les arcs qui partent du sommet 1 et les arcs qui partent du sommet 2, ce qui donne 2 croisements.

De la même manière, $K(1,3) = 2$, et $K(2,3) = 0$. De sorte que le nombre total de croisements vaut 4.

Le nombre total de croisements $K(M)$ d'un graphe représenté par la matrice d'interconnexion $M[n_1, n_2]$ s'exprime de la manière suivante :

$$K(M) = \sum_{i=1..n_1-1} \sum_{j=i+1..n_1} K(i,j) \quad (E1)$$

$$K(i,j) = \sum_{r=1..n_2-1} \sum_{t=r+1..n_2} N(i,r) * N(j,t) \quad (E2)$$

1.3. Calcul du nombre de croisements dans une hiérarchie

1.3.1. Hiérarchie Impropre

Si la hiérarchie est impropre, il existe des sommets de portée supérieure à 1. Par conséquent les croisements induits par ces arcs ne seront pas comptabilisés dans la formule (E.1). Par exemple, l'application de cette formule sur le graphe de la figure VII.25 aurait donné 0, alors qu'il existe un croisement.

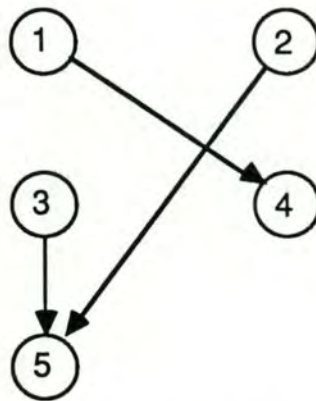


figure VII.25

De plus, l'exemple de la figure VII.25 fait apparaître que pour déterminer le nombre de croisements dans une hiérarchie impropre, l'ordre des sommets à l'intérieur d'un niveau ne suffit pas. En effet, dans cet exemple, le déplacement du sommet 4 vers la gauche supprime le croisement existant. On doit par conséquent considérer la position précise du sommet.

1.3.2. Hiérarchie propre

Si la hiérarchie est propre, le nombre de croisements d'un graphe $K(G)$ est égal à la somme du nombre de croisements entre deux niveaux successifs $K(N_l)$, ou encore :

$$K(G) = \sum_{l=1..NIVMAX-1} K(N_l) \quad (E3)$$

où $NIVMAX$ = nombre de niveaux dans le graphe G
 N_l = matrice d'interconnexion entre les niveaux l et $l+1$ du graphe G .

Comme l'ordre des sommets est suffisant pour déterminer le nombre de croisements, le problème de réduction du nombre de croisements dans le cas d'une hiérarchie propre consiste à ordonner les sommets à chaque niveau, c'est-à-dire à définir un ensemble de permutations p_l (où l représente un numéro de niveau) de sommets.

2. PROGRAMME LINEAIRE POUR MINIMISER LE NOMBRE DE CROISEMENTS

2.1. Préalable

Deux arcs sont **susceptibles** de se croiser s'ils partent tous deux du même niveau, et n'ont aucune extrémité en commun. Sur l'exemple de la figure VII.26, seuls les arcs u_1 et u_2 sont susceptibles de se croiser.

Deux arcs se croisent **effectivement** si l'origine du premier est à droite de l'origine du second et la destination du premier est à gauche de celle du second ou inversement.

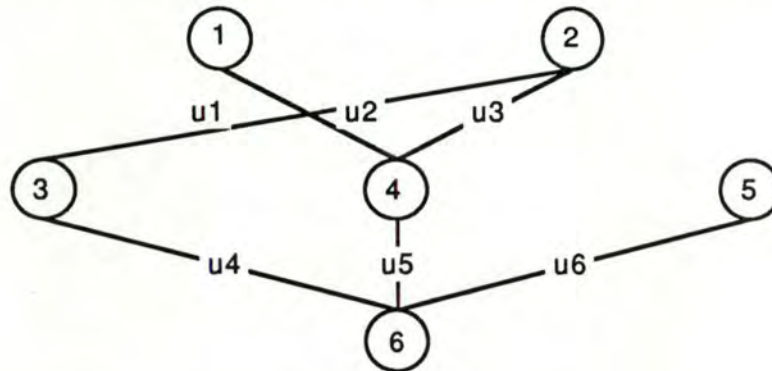


figure VII.26

2.2. Formalisation

Soit a et b deux arcs susceptibles de se croiser. Pour exprimer l'existence d'un croisement entre a et b , on utilise une variable binaire C_{ab} dont la valeur est égale à 1 si a et b se croisent, 0 dans le cas contraire.

Par conséquent, la fonction objective consiste à minimiser la somme des C_{ab} , sur tous les couples d'arcs (a,b) tels que a et b peuvent se croiser.

L'existence d'un croisement entre a et b est exprimé au moyen de l'inéquation

$$C_{ab} \geq (D_{ab} + G_{ab}) / 2$$

où

$D_{ab} = 1$ uniquement si a et b se croisent et si l'origine de a est à droite de celle de b

$G_{ab} = 1$ uniquement si a et b se croisent et si l'origine de a est à gauche de celle de b

En guise d'illustration sur l'exemple de la figure VII.26 $D_{u_1 u_2} = 1$ et $G_{u_1 u_2} = 0$ par conséquent $((D_{u_1 u_2} + G_{u_1 u_2})/2) = 0.5$ et on a bien $C_{u_1 u_2} = 1 \geq 0.5$. Pour u_1 et u_3 , $D_{u_1 u_3} = 0$ et $G_{u_1 u_3} = 0$ et on a bien $C_{u_1 u_3} = 0$.

Les variables binaires D_{ab} et C_{ab} sont elles-mêmes exprimées sous la forme d'autres variables binaires. Comme le développement de ces deux variables est analogue, nous n'expliquons ici que D_{ab} .

Les variables binaires D_{ab} où $a=(s_i, s'_i)$ et $b=(s_j, s'_j)$ sont calculées à l'aide des variables binaires d_{ij} et g_{rt} :

$d_{ij} = 1$ signifie que s_i est à droite de s_j

$g_{rt} = 1$ signifie que s'_r est à gauche de s'_t

Par sa définition, $D_{ab} = 1$ uniquement si $d_{ij} = 1$ et $g_{rt} = 1$. La contrainte suivante permet donc le calcul de D_{ab} :

$$D_{ab} \geq d_{ij} + g_{rt} - 1$$

Il reste à exprimer les variables d_{ij} et g_{rt} par rapport aux positions des sommets. La formalisation de ces deux contraintes étant similaire, nous ne détaillons ici que la première.

Un sommet s_i est à droite du sommet s_j si la différence des abscisses ($x_i - x_j$) est positive. Soit e_{max} l'écart maximal qui puisse exister entre deux sommets. La contrainte suivante définit d_{ij} :

$$d_{ij} \geq (x_i - x_j) / e_{max}$$

La modélisation complète de la minimisation du nombre de croisements est la suivante :

MIN	\sum	$C_{(a,b)}$	→	couple d'arcs susceptibles de se croiser
	(a,b)			
$C_{(a,b)}$	\geq	$(D_{(a,b)} + G_{(a,b)})$	$/$	2
$D_{(a,b)}$	\geq	$d_{i,j} + g_{r,t} - 1$		
$G_{(a,b)}$	\geq	$g_{i,j} + d_{r,t} - 1$		
$d_{i,j}$	\geq	$(X_i - X_j)$	$/$	e_{max}
$g_{r,t}$	\geq	$(X_t - X_r)$	$/$	e_{max}
$g_{i,j}$	\geq	$(X_j - X_i)$	$/$	e_{max}
$d_{r,t}$	\geq	$(X_r - X_t)$	$/$	e_{max}

2.3. Evaluation

Comme la formalisation du problème nécessite l'emploi de variables binaires, l'utilisation pratique du programme linéaire s'avère quasi impossible.

Même d'un point de vue théorique le nombre de variables utilisées est très important. On peut en effet constater l'emploi de trois variables binaires (C_{ab} , D_{ab} , et G_{ab}) pour chaque paire d'arcs susceptibles de se croiser et deux variables binaires (d_{ij} , et g_{ij}) sont associées à chaque couple de sommets d'un même niveau.

De sorte que si la hiérarchie comporte NIVMAX niveaux, n_{niv} sommets en moyenne par niveau, et m_{niv} arcs entre deux niveaux, le programme linéaire peut comporter jusqu'à $[3 * (m_{niv} * (m_{niv} - 1)) / 2] + [NIVMAX * n * (n - 1)]$ variables binaires et le même nombre de contraintes.

Par exemple, la réduction du nombre de croisements du graphe de la figure VII.11 nécessiterait l'emploi de 39 variables binaires alors que le graphe comporte 9 sommets et 9 arcs.

3. HEURISTIQUE POUR REDUIRE LE NOMBRE DE CROISEMENTS

Le rejet du programme linéaire nous a amenés à essayer de trouver une méthode alternative. L'heuristique que nous proposons donne une solution optimale lorsque le graphe peut être représenté sans croisement. Dans le cas contraire, il donne une solution qui n'est pas nécessairement la solution idéale.

Cette heuristique est particulièrement bien adaptée pour les graphes qui seront traités par l'algorithme. En effet, si on considère, par exemple, le schéma de la dynamique, il y a beaucoup de chance pour qu'il soit représentable sans croisement. L'efficacité de l'algorithme dans le cas où le graphe contient au moins un croisement sera vérifiée en pratique.

3.1. Idée de la méthode

Le nombre de croisements d'une hiérarchie propre étant uniquement déterminé par l'ordre des sommets à l'intérieur des niveaux, une méthode pour minimiser le nombre de croisements consiste à envisager toutes les permutations possibles des sommets et à retenir celles qui présentent le moins de croisements.

Comme le souligne [Warfield, 77] cette technique exhaustive étant de nature combinatoire si le nombre de sommets et/ou d'arcs augmente, on atteint rapidement la limite où elle s'avère inapplicable. L'emploi d'une heuristique semble donc vivement conseillé.

Disposant de la formule qui calcule le nombre de croisements entre deux niveaux l et $l+1$ successifs, il est possible de déterminer les couples de permutations (p_l, p_{l+1}) qui minimisent le nombre de croisements.

S'il existe une suite de permutations $\{p_1, p_2, \dots, p_{\text{nivmax}}\}$ telles que deux permutations consécutives minimisent le nombre de croisements entre leurs niveaux respectifs alors, cette suite de permutations détermine une solution optimale. En effet, le nombre de croisements induits par cette solution est égal à la somme du nombre de croisements entre chaque niveau, ce nombre de croisements étant par hypothèse minimum, la somme est également minimale.

Toutefois, cette suite de permutations n'existe pas toujours. En effet, pour le graphe de la figure VII.27, il n'existe pas de permutation au niveau 2 qui minimise le nombre de croisements à la fois entre les niveaux 1 et 2, et entre les niveaux 2 et 3.

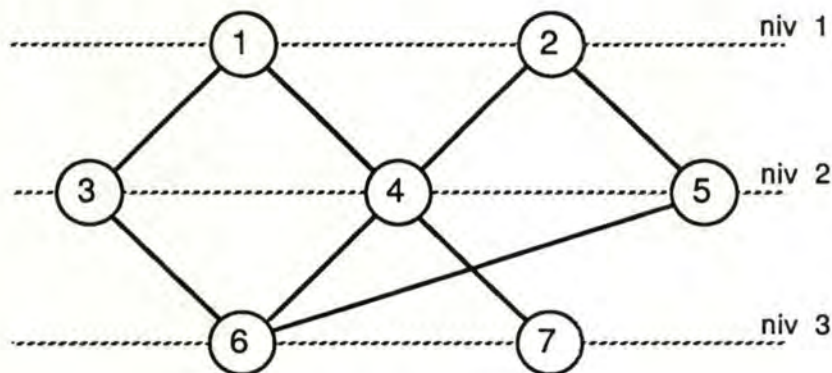


figure VII.27

L'idée de notre heuristique consiste toujours à minimiser le nombre de croisements entre deux niveaux consécutifs l et $l+1$ mais cette fois, lors du traitement d'un niveau L , plutôt que de rechercher le minimum de croisements sur tous les couples possibles de permutations, on envisage uniquement les couples dont le premier membre appartient à un ensemble P_l de permutations.

Parmi ces couples, ceux qui induisent le minimum de croisements sont rangés dans un ensemble noté CP_l et les permutations du niveau $l+1$ apparaissant dans CP_l constituent P_{l+1} .

On itère ce processus en partant du niveau 1, avec P_1 qui contient toutes les permutations des sommets du niveau 1, jusqu'au niveau $NIVMAX-1$ où $NIVMAX$ désigne la longueur de la hiérarchie.

Soit SOL, l'ensemble de toutes les solutions produites par notre méthode. Cet ensemble est constitué des suites de permutations $\{ \{ p_1, p_2, \dots, p_{n_{ivmax}} \}, \dots \}$ telles que (p_i, p_{i+1}) appartient à CP_i . Comme ces solutions déterminent chacune une représentation graphique de la hiérarchie contenant le même nombre de croisements, la solution finale produite par notre méthode est une suite quelconque de SOL.

Par exemple, sur le graphe de la figure VII.11, l'exécution de notre heuristique produit les ensembles suivants :

$$P_1 = \{ 1 \}$$

$$CP_1 = \{ [1, (2,4,6,8)], [1, (2,4,8,6)], \dots \}$$

$$P_2 = \{ \text{toutes les permutations de } (2, 4, 6, 8) \}$$

$$CP_2 = \{ [(2,4,8,6), (3,5,9)], [(6,2,4,8), (9,3,5)], \\ [(6,8,4,2), (9,5,3)], [(8,4,2,6), (5,3,9)] \}$$

$$P_3 = \{ (3,5,9), (9,3,5), (9,5,3), (5,3,9) \}$$

$$CP_3 = \{ [(3,5,9), 7], [(9,3,5), 7], \\ [(9,5,3), 7], [(5,3,9), 7] \}$$

$$SOL = \{ [1, (2,4,8,6), (3,5,9), 7], \\ [1, (6,2,4,8), (9,3,5), 7], \\ [1, (6,8,4,2), (9,5,3), 7], \\ [1, (8,4,2,6), (5,3,9), 7] \}$$

3.2. Représentation planaire

démonstration

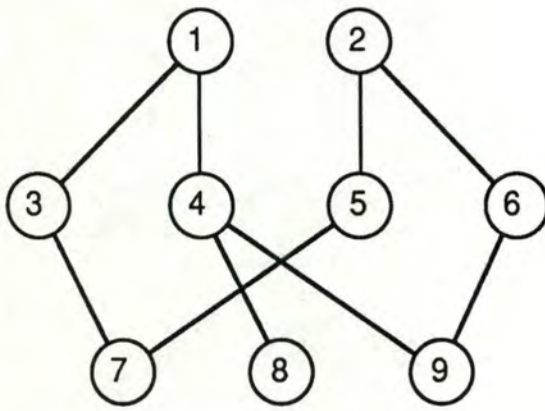
Nous démontrons ici que, si le graphe est représentable sans croisement, notre méthode donne une solution optimale :

- si le graphe est représentable sans croisement, alors, il existe une suite optimale $p_1^*, \dots, p_i^*, p_{i+1}^*, \dots, p_{n_{ivmax}}^*$ telle que $K(p_i^*, p_{i+1}^*) = 0$.
- p_1^* appartient à P_1 puisque P_1 contient toutes les permutations du niveau 1.
- (p_1^*, p_2^*) appartient à CP_1 puisque $K(p_1^*, p_2^*) = 0$.
- (p_2^*, p_3^*) appartient à CP_2 puisque $K(p_2^*, p_3^*) = 0$ et p_2^* appartient à P_1 .
- ...
- $(p_{n_{ivmax}-1}^*, p_{n_{ivmax}}^*)$ appartient à $CP_{n_{ivmax}-1}$ puisque $K(p_{n_{ivmax}-1}^*, p_{n_{ivmax}}^*) = 0$ et $p_{n_{ivmax}-1}^*$ appartient à $P_{n_{ivmax}-1}$.

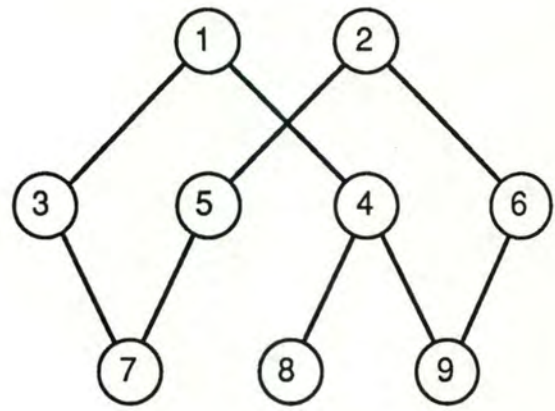
De sorte que $[p_1^*, \dots, p_i^*, p_{i+1}^*, \dots, p_{n_{ivmax}}^*]$ appartient à SOL et donc toute solution trouvée par notre méthode définit une représentation graphique sans croisement.

Par contre, si le graphe n'est pas représentable sans croisement alors, notre méthode donne une solution qui n'est pas nécessairement optimale.

A titre d'illustration, si on considère les graphiques de la figure VII.28, notre heuristique produit celui qui présente 2 croisements au lieu de la solution optimale qui ne comporte qu'un croisement.



résultat de notre heuristique



solution optimale

Figure VII.28

3.3. Algorithme

1. Programme principal

```

begin
  P[1] := ensemble des permutations du niveau 1;
  for l := 1 to NIVMAX do
    minimiser_crois(l)
    { Construction de CP[l] et de P[l+1] connaissant P[l] }
  endfor
  sol := choix_solution(CP)
  { choix d'une solution connaissant CP[1], CP[2], ..., CP[NIVMAX] }
end

```

2. Minimiser_crois(l)

{ Construction de CP[l] et de P[l+1] connaissant P[l] }

```

begin
  min-nb-crois := +∞;
  CP[l] := ∅;
  P[l+1] := ∅;
  for all perm-sup in P[l] do
    for all perm-inf : permutation des sommets du niveau l+1 do
      begin
        nb-crois := compter_crois( perm-sup, perm-inf, l );
        if nb-crois < min-nb-crois
          then
            begin
              CP[l] := { (perm-sup, perm-inf) };
              P[l+1] := { perm-inf };
              min-nb-crois := nb-crois;
            end
          else
            if nb-crois ≤ min-nb-crois
              then
                begin
                  CP[l] := CP[l] U { (perm-sup, perm-inf) };
                  P[l+1] := P[l+1] U { perm-inf };
                end
              end
            end
          end
        end
      end
    end
  end
end

```

3. Compter_crois(perm-sup, perm-inf, l)

```

begin
  nb-crois := 0;
  for all a : sommet du niveau l do
    for all b : sommet du niveau l tq a est avant b dans perm-sup do
      for all c : sommet du niveau l+1 tq (b,c) est un arc du graphe do
        for all d : sommet du niveau l+1 tq c est avant d dans perm-inf ET (a,d) est un arc do
          nb-crois := nb-crois + 1;
        endfor
      endfor
    endfor
  endfor
end

```

4. Choix_solution(CP)

{ Choix d'une solution connaissant CP[1], CP[2], ..., CP[NIVMAX] }

```

begin
  membre-2 := permutation quelconque de P[NIVMAX];
  Sol := { membre-2 };
  for i := (NIVMAX-1) downto 1 do
    begin
      Choisir perm-sup tel que (perm-sup, membre-2) appartient à CP[i];
      Sol := Sol U { perm-sup };
      membre-2 := perm-sup;
    end
  endfor
end

```

3.4. Evaluation

la complexité de notre algorithme est de l'ordre de $NIVMAX * (n_{niv})^2$ où NIVMAX est la longueur de la hiérarchie et n_{niv} est le nombre maximum de sommets à un niveau

En effet, l'algorithme recherche des permutations de sommets pour chacun des NIVMAX niveaux. Pour cela, elle compare les permutations des sommets du niveau précédent (c'est-à-dire $(n_{niv}!)$ permutations au maximum) avec toutes les permutations possibles des sommets du niveau courant (c'est-à-dire $(n_{niv}!)$ possibilités au maximum). Le traitement d'un niveau effectue donc $(n_{niv}!)^2$ comparaisons dans le plus mauvais des cas.

3.5. Optimisations

Nous allons décrire ici deux optimisations possibles de notre heuristique. Dans un premier temps, nous montrerons comment on peut facilement diviser le temps de calcul par deux. Nous verrons ensuite comment améliorer notre heuristique de manière à obtenir la solution optimale lorsque le graphe est représentable avec un croisement au plus.

[Warfield, 77] montre que le nombre de croisements induits par une suite de permutations $\{ p_1, p_2, \dots, p_{n_{nivmax}} \}$ est égal au nombre de croisements induits par la suite $\{ q_1, q_2, \dots, q_{n_{nivmax}} \}$ où q_i désigne la permutation p_i inversée. Par exemple, si $p_1 = (1,2,3)$ alors $q_1 = (3,2,1)$. De sorte que, si $P_1 = \{ p_{11}, p_{12}, \dots, p_{1x} \} \setminus \{ q_{11}, q_{12}, \dots, q_{1x} \}$, on réduit le temps de calcul de moitié.

Pour donner la solution optimale lorsque le graphe présente un croisement au plus, il suffit d'appliquer une première fois notre méthode en partant du niveau 1, une deuxième fois en partant du dernier niveau et de retenir comme solution celle qui donne le moins de croisements.

Cette deuxième technique nous donne la solution optimale dans l'exemple de la figure VII.28. Toutefois, on multiplie le temps de calcul par deux puisqu'on applique deux fois notre heuristique.

VII.3.5. CALCUL DES COORDONNEES

Nous proposons ici une méthode pour attribuer à chaque sommet d'une hiérarchie propre des coordonnées dans le plan. Nous énoncerons pour commencer les contraintes dont doit tenir compte la méthode pour assurer la lisibilité du dessin. Nous formaliserons ensuite ces contraintes sous la forme d'un programme linéaire.

VII.3.5.1. Spécification

Disposant d'une hiérarchie propre, il s'agit d'attribuer une coordonnée (x_i, y_i) à chaque sommet s_i du graphe. x_i représente la coordonnée horizontale (ou abscisse) du sommet tandis que y_i sa coordonnée verticale (ou ordonnée). La position d'un sommet doit être telle que :

- <1> l'ordre des niveaux est respecté, c'est-à-dire si le niveau d'un sommet s_i est inférieur au niveau d'un sommet s_j , alors $y_i < y_j$
- <2> l'ordre des sommets à l'intérieur d'un niveau est respecté, c'est-à-dire si un sommet s_i se trouve à gauche d'un sommet s_j , alors $x_i < x_j$
- <3> il existe une distance minimale H entre deux sommets de niveau différent
- <4> il existe une distance minimale L entre deux sommets d'un même niveau
- <5> le dessin est équilibré et le plus vertical possible. La construction d'un dessin équilibré ou symétrique, consiste à centrer un sommet par rapport à ses suivants et inversement à centrer un sommet par rapport à ses précédents. Un dessin vertical s'obtient en plaçant un sommet à la verticale de son suivant et/ou de son précédent.

VII.3.5.2. Réalisation

IDEE DE LA SOLUTION

Les seules contraintes que doivent respecter les **coordonnées verticales** sont l'ordre des niveaux <1> et la distance minimale H entre deux sommets <3>. L'attribution de ces coordonnées peut se faire simplement en multipliant le niveau du sommet par la distance minimale H. Dès lors, le calcul des ordonnées ne fera pas l'objet d'une description plus détaillée.

Le calcul des **coordonnées horizontales** est par contre beaucoup plus délicat car il faut non seulement tenir compte des contraintes sur l'ordre des sommets <2> et la distance minimale L entre deux sommets <4> mais il faut aussi prendre en considération les critères esthétiques repris au point <5> qui assurent une répartition homogène des sommets. Pour résoudre ce problème, nous allons formaliser ces contraintes sous la forme d'un programme linéaire.

Vérification des contraintes <2> et <4>

Pour que l'ordre des sommets à l'intérieur d'un niveau soit respecté <2>, il suffit que l'abscisse x_i de tout sommet s_i soit plus petite que l'abscisse x_j du premier sommet qui suit s_i sur son niveau. La condition <4> exige de plus que la distance horizontale entre ces deux sommets soit d'au moins L . Ces deux exigences conduisent à la contrainte suivante :

$$x_j - x_i \geq L \quad \text{pour tous sommets } s_i, s_j \text{ tels que}$$

$$s_i \text{ et } s_j \text{ ont le même niveau}$$

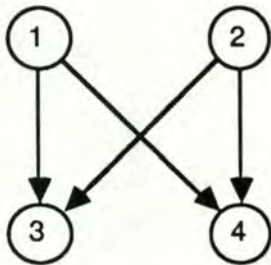
$$s_j \text{ est le premier sommet qui suit } s_i$$

Vérification de la contrainte <5>

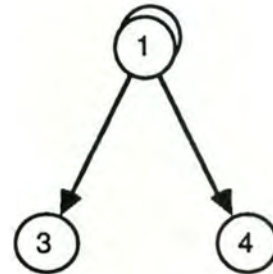
La vérification des critères esthétiques <5> qui assurent une répartition homogène des sommets est beaucoup plus complexe car on est confronté dans ce cas au problème des contraintes incompatibles dont il faudra tenir compte dans la résolution du problème.

Par exemple, à la figure VII.29 on ne peut pas imposer simultanément que :

- les sommets 1 et 2 soient placés à une distance minimale non nulle
- les sommets 1 et 2 soient au milieu des sommets 3 et 4



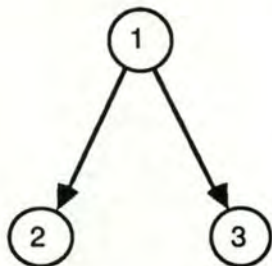
(a) Distance minimale respectée



(b) Dessin équilibré

Figure VII.29 incompatibilité des contraintes <2> et <5>

Sur l'exemple de la figure VII.30, on peut constater l'incompatibilité des contraintes de disposition des sommets sur un niveau, avec un positionnement des sommets entre deux niveaux qui assurent un dessin équilibré. Mais les critères esthétiques peuvent également être incompatibles entre eux. Par exemple, sur la figure VII.?, on ne peut pas imposer que le sommet 1 soit au milieu des sommets 2 et 3 et en plus exiger de positionner le sommet 2 à la verticale du sommet 1



(a) Dessin équilibré



(b) Dessin vertical

Figure VII.30 incompatibilité des contraintes esthétiques

Nous allons maintenant envisager successivement le problème de la construction d'un dessin équilibré et celui des droites verticales.

Dessin équilibré

Les trois contraintes à imposer pour qu'un dessin soit équilibré sont décrites ci-dessous :

<R1> Un sommet s_i est placé au milieu de ses suivants s_j :

$$x_i = \left(\sum_j x_j \right) / \text{nombre de suivants de } s_i$$

$\xrightarrow{\text{tel que } x_j \text{ est l'abscisse d'un suivant de } s_i}$

Par exemple :

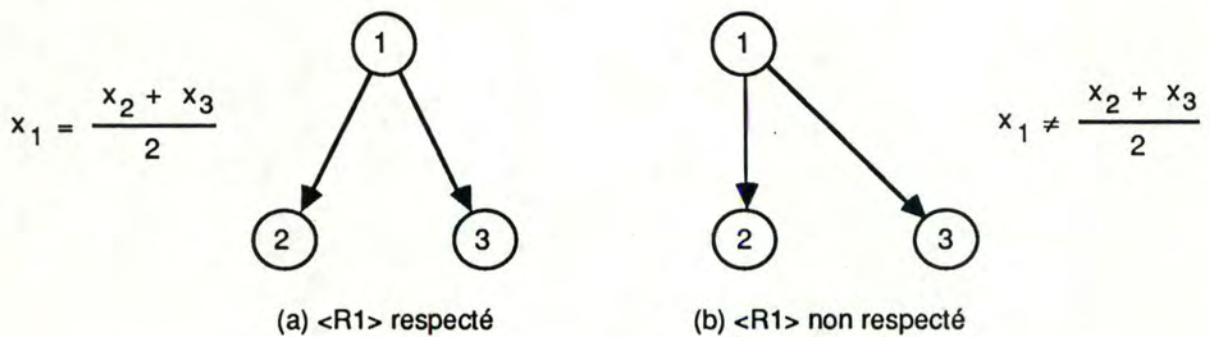


Figure VII.31 règle R1

<R2> Un sommet s_i est placé au milieu de ses précédents s_j :

$$x_i = \left(\sum_j x_j \right) / \text{nombre de précédents de } s_i$$

$\xrightarrow{\text{tel que } x_j \text{ est l'abscisse d'un précédent de } s_i}$

Par exemple :

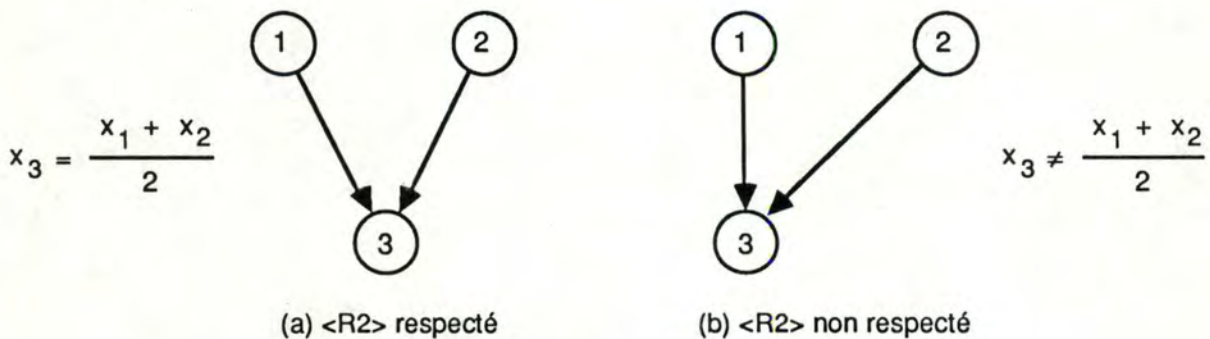


figure VII.32 règle R2

<R3> Si deux sommets ont les mêmes suivants, on ne peut pas imposer R1 puisque, comme illustré à la figure VII.?, les deux sommets occuperaient la même position, ce qui contredirait la condition <2> qui impose une distance minimale entre deux sommets. On peut toutefois imposer de minimiser l'écart horizontal entre un sommet et son suivant à l'aide des contraintes suivantes :

$$\text{MIN } \sum \text{écart}_k$$

formalisé par $e_k + e_{k+1}$

$$e_k - e_{k+1} = x_i - x_j \quad \text{pour chaque sommet } s_i \text{ et } s_j \text{ tels que } s_j \text{ est un suivant de } s_i$$

Ce qui conduirait au dessin (a) de la figure VII.33, plutôt qu'à celui de droite puisque la somme des écarts du dessin (a) vaut deux unités tandis que pour le dessin (b), cette somme vaut 3 unités.

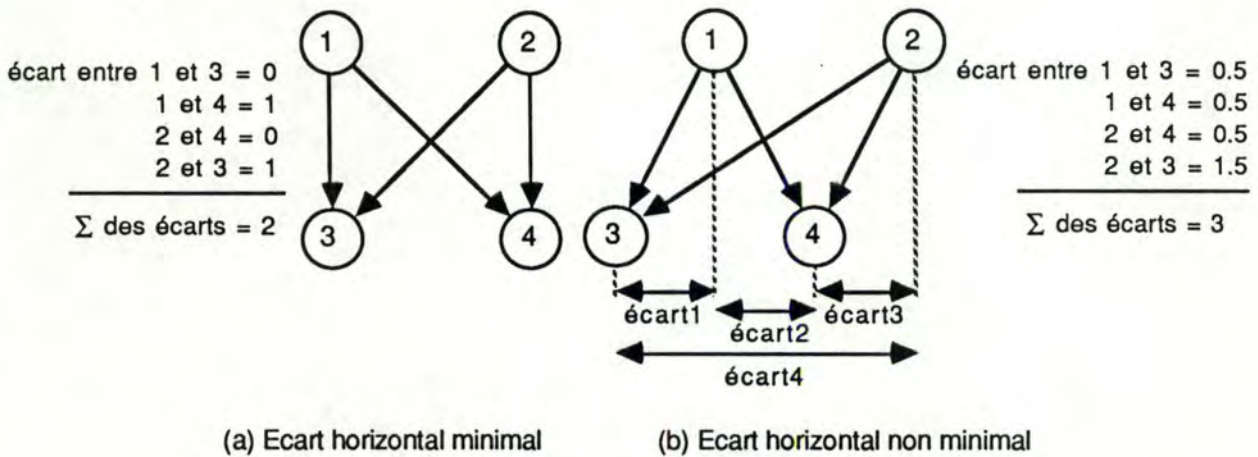


Figure VII.33 minimiser les écarts

Droites verticales

Les deux contraintes à imposer pour qu'un dessin soit vertical sont décrites ci-dessous :

<R4> un sommet s_i est placé à la verticale de son suivant s_j si s_i est le seul précédent de s_j

$$x_i = x_j \quad \text{si } s_i \text{ est le seul précédent de } s_j$$

tel que x_i est l'abscisse d'un précédent de s_j

Par exemple :

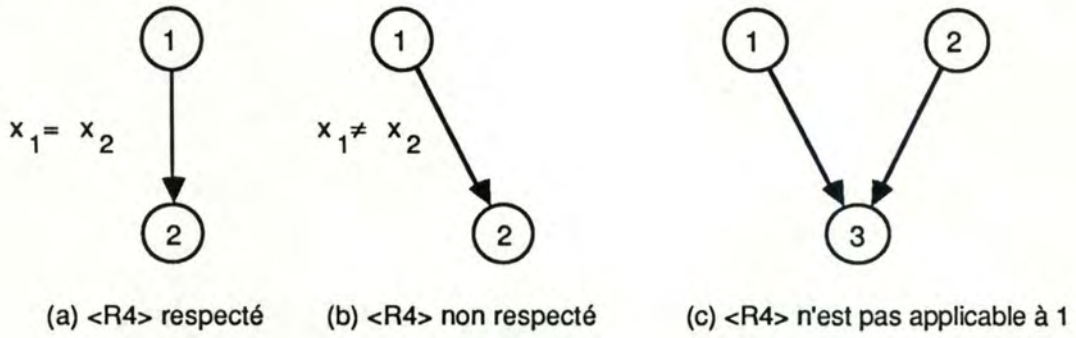
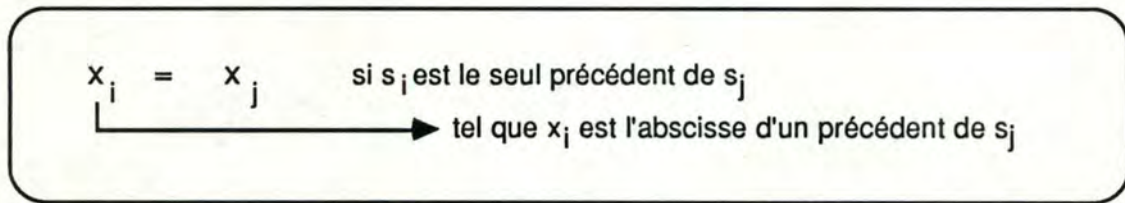


figure VII.34 règle <R4>

D'autre part, lorsque s_i n'est pas le seul précédent de s_j , <R4> ne peut pas être appliquée car elle est contradictoire avec la règle <R2> appliquée à s_j .

Par exemple, on ne peut pas imposer sur le dessin de la figure VII.34 <c> que le sommet 1 soit à la verticale du sommet 3 puisque celui-ci, en vertu de la règle <R2>, est au milieu des sommets 1 et 2.

<R5> un sommet s_i est placé à la verticale de son précédent s_j si s_i est le seul suivant de s_j .



Par exemple :

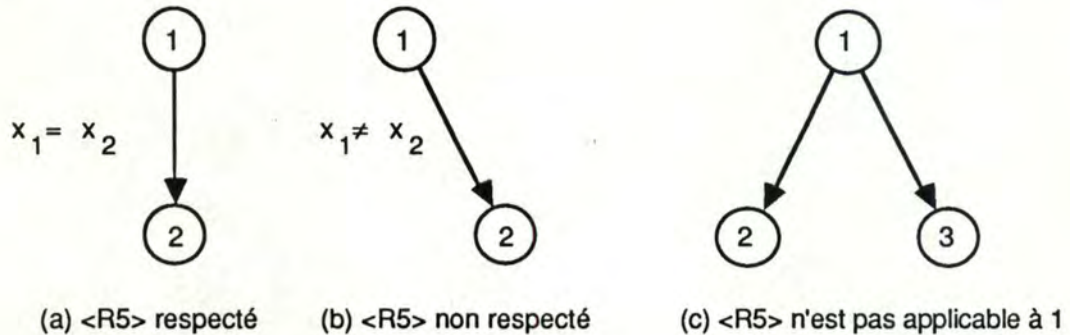


figure VII.35 règle <R5>

De plus, lorsque s_i n'est pas le seul suivant de s_j , <R5> ne peut pas être appliquée car elle est contradictoire avec la règle <R1> appliquée à s_j .

Par exemple, on ne peut pas imposer sur le dessin de la figure VII.35 <c> que le sommet 2 soit à la verticale du sommet 1 puisque celui-ci, en vertu de la règle <R1>, est au milieu des sommets 2 et 3.

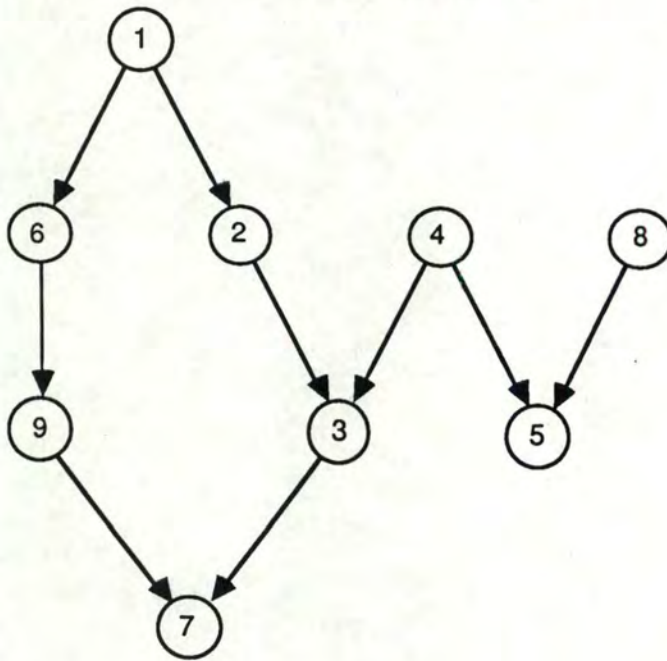
Dessin compact

Au cas où plusieurs solutions satisfont aux contraintes imposées ci-dessus, on préférera choisir un dessin le plus étroit possible. Pour ce faire, on utilise la fonction objective suivante :

$$\text{MIN } \sum_i x_i$$

Exemple

En appliquant les règles décrites ci-dessus à l'exemple de la figure VII.12., on obtient le programme linéaire et le dessin de la figure VII.36.



$$\text{MIN } (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8)$$

$$x_1 = (x_2 + x_6) / 2$$

$$x_6 = x_9$$

$$x_4 = (x_3 + x_5) / 2$$

$$x_3 = (x_2 + x_4) / 2$$

$$x_5 = (x_4 + x_8) / 2$$

$$x_7 = (x_9 + x_3) / 2$$

$$x_2 - x_6 > L$$

$$x_4 - x_2 > L$$

$$x_8 - x_4 > L$$

$$x_3 - x_9 > L$$

$$x_5 - x_3 > L$$

Figure VII.36 calcul des abscisses des sommets

CONCLUSION

En conclusion, notre méthode calcule des positions dans le plan des sommets d'un graphe sans circuit qui respectent les contraintes de dispositions inhérentes aux graphes hiérarchisés par les rangs et qui donnent une solution visuellement agréable.

Le seul critère esthétique qui n'a pas été totalement respecté est la minimisation du nombre de croisements, puisque seule une heuristique semble en pratique utilisable vu le caractère combinatoire du problème.

Etant donné la caractéristique de l'heuristique proposée, la méthode est recommandée pour les graphes représentables avec peu de croisements et idéalement sans croisement.

L'utilisation de notre méthode pour la production d'un diagramme particulier nécessite deux étapes supplémentaires :

- une étape de construction du graphe à partir de la description du schéma
- une étape de production du graphique qui à partir des coordonnées calculées par notre algorithme et de la description des symboles utilisés dans le diagramme (rectangle, flèche,...), génère automatiquement le graphique correspondant.

Comme ces deux étapes font appel à la sémantique du diagramme, elles devront être particularisées pour chacun des graphiques.

- CHAPITRE VIII-

**Production automatique du schéma
de la dynamique**

INTRODUCTION

Nous avons vu, au chapitre précédent, une méthode de production automatique d'un graphe hiérarchisé qui va nous servir de base pour résoudre le problème traité dans ce chapitre, en l'occurrence, la production automatique du schéma de la dynamique.

Tous d'abord, nous spécifierons le problème en donnant les conventions de représentation du diagramme à produire automatiquement ainsi que les contraintes esthétiques qu'il doit respecter pour assurer une bonne lisibilité du dessin.

Nous examinerons ensuite les étapes nécessaires à la production automatique du schéma de la dynamique qui assurent le respect des contraintes émises sur ce schéma.

VIII.1. SPECIFICATION DU PROBLEME

Le schéma de la dynamique est la représentation associée à la spécification du comportement d'un système d'information appelé le modèle de la dynamique. Ce comportement est spécifié dans l'atelier logiciel IDA au moyen du langage DSL (Dynamic Specification Language).

Nous présenterons d'abord l'ensemble des relations, appelées relations dynamiques, permettant de spécifier en DSL un modèle de la dynamique. Nous décrirons ensuite les représentations graphiques associées à ces relations qui définiront ainsi le schéma de la dynamique.

VIII.1.1. Modèle de la dynamique

Le modèle de la dynamique, comme les autres modèles proposés par le système logiciel IDA, est spécifié dans le langage DSL à l'aide de (type d') **objets**, de (type de) **relations** entre ces objets et de **propriétés** associées aux objets.

Par exemple, pour spécifier, en DSL, que 10 processus P2 doivent être déclenchés par la terminaison d'un processus P1 si la condition C1 est vérifiée, on utilise le formalisme suivant :

```
DEFINIR PROCESSUS P1
  EN TERMINAISON DECLENCHE 10 P2 SI C1
```

Où

- les processus P1, P2, la condition C1 et le nombre de déclenchements sont des objets de la relation
- EN TERMINAISON DECLENCHE définit la relation entre ces objets
- le nom des objets constitue une de leur propriété

De plus, chaque objet qui intervient dans une relation occupe une certaine place (un coin) qui détermine son rôle dans la relation.

Par exemple, le processus P1 dans la relation de déclenchement en question est l'objet déclencheur tandis que le processus P2 est l'objet déclenché.

Par convention nous supposerons que le premier coin contient l'objet **origine** de la relation (ici P1), le deuxième coin reprend l'objet **destination** (ici P2), le troisième coin contient éventuellement une **condition** (ici C1) et le quatrième peut contenir une **duplication** (ici le déclenchement de 10 processus P2).

La syntaxe DSL extraite de [DSL, 87] qui permet la spécification d'un modèle de la dynamique est reprise ci-dessous. Nous avons considéré la génération d'un message comme partie intégrante du comportement du système d'information pour tenir compte d'une spécification de la dynamique par les messages. De plus, les phrases DSL sont exprimées sous leur forme active et sont regroupées selon le type de l'objet origine de la relation.

DEFINIR INTERFACE <Interface>

- (I1) GENERE [<duplication>] <message> [SI <condition>]
- (I2) GENERE [<duplication>] <message> SI PAS <condition>
- (I3) CAUSE SURVENANCE DE <message> CHAQUE <répétition> PENDANT
<calendrier>
- (I4) CAUSE SURVENANCE DE <message> <nb-fois> FOIS PENDANT <calendrier>

DEFINIR MESSAGE <message>

- (M1) EN GENERATION DECLENCHE <processus>
- (M2) EN GENERATION CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation>

DEFINIR POINT-DE-SYNCHRONISATION <point-de-synchronisation>

- (S1) EN REALISATION DECLENCHE <processus> [POUR CHAQUE <duplication>] [SI <condition>]
- (S2) EN REALISATION DECLENCHE <processus> [POUR CHAQUE <duplication>] SI PAS <condition>
- (S3) EN REALISATION CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation> [SI <condition>]
- (S4) EN REALISATION CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation> SI PAS <condition>

DEFINIR PROCESSUS <processus>

- (P1) GENERE [<duplication>] <message> [SI <condition>]
- (P2) GENERE [<duplication>] <message> SI PAS <condition>
- (P3) EN ACTIVATION DECLENCHE <processus> [POUR CHAQUE <duplication>] [SI <condition>]
- (P4) EN ACTIVATION DECLENCHE <processus> [POUR CHAQUE <duplication>] SI PAS <condition>
- (P5) EN ACTIVATION CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation> [SI <condition>]
- (P6) EN ACTIVATION CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation> SI PAS <condition>
- (P7) EN TERMINAISON DECLENCHE <processus> [POUR CHAQUE <duplication>] [SI <condition>]
- (P8) EN TERMINAISON DECLENCHE <processus> [POUR CHAQUE <duplication>] SI PAS <condition>
- (P9) EN TERMINAISON CONTRIBUE A [{UN | UNE | PLUSIEURS}] <point-de-synchronisation> [SI <condition>]

(P10) EN TERMINAISON CONTRIBUE A [(UN | UNE | PLUSIEURS)] <point-de-synchronisation> SI PAS <condition>

avec :

<nb_fois> = { nom-de-PARAMETRE-DU-SYSTEME | <nombre> }
 <duplication> = { nom-de-ATTRIBUT | nom-de-PARAMETRE-DU-SYSTEME | <entier> }
 <répétition> = { nom-de-PARAMETRE-DU-SYSTEME | <chaîne de caractères> }

VIII.1.2. Schéma de la dynamique

Les représentations graphiques des relations DSL décrites au point VIII.1.1 ainsi que le dessin des objets qu'elles relient constituent un schéma de la dynamique.

Outre les conventions de représentation de ces relations et de ces objets, le diagramme produit devra également vérifier des contraintes esthétiques qui assurent une bonne lisibilité du dessin.

Nous allons envisager successivement les conventions de représentation des objets ayant un équivalent graphique, la manière de représenter les relations entre ces objets et pour terminer nous examinerons les critères esthétiques que doivent idéalement respecter les schémas de la dynamique produits automatiquement.

OBJETS

Nous avons repris ci-dessous les conventions de représentation des objets DSL ayant un équivalent graphique (ces objets sont repris en gras dans la syntaxe DSL précisée au point VIII.1.1.).

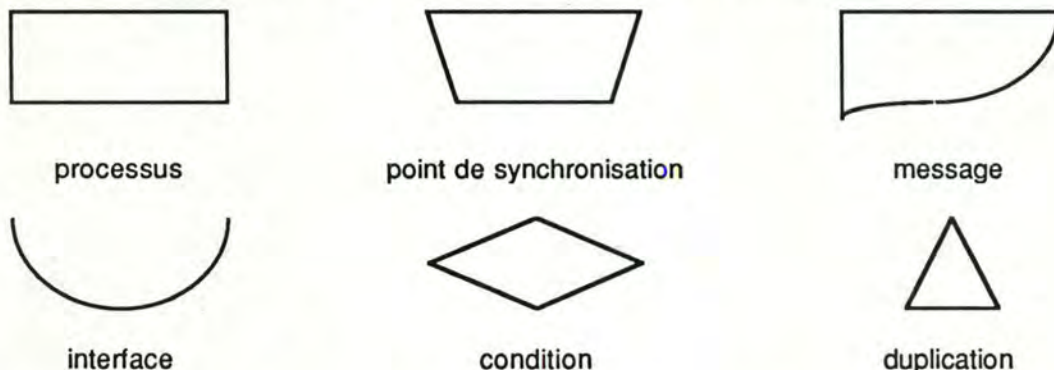


Figure VIII.1 conventions de représentation des objets

RELATIONS ENTRE CES OBJETS

On représente classiquement une relation DSL par une **droite**, éventuellement brisée, qui joint l'objet origine de la relation avec celui qui en est la destination. L'événement qui cause le déclenchement d'un processus, la génération d'un message ou qui contribue à un point de synchronisation est représenté par un **point** sur une des extrémités de cette droite. Le déclenchement, la génération ou la contribution sont quant à eux représentés à l'autre extrémité de la droite par une **flèche**.

Par souci de simplicité pour l'exposé des représentations graphiques des relations DSL, nous allons distinguer deux classes de relations :

- celles de la figure VIII.2 qui ne sont pas du type "EN ACTIVATION ...".
- celles de la figure VIII.3 qui sont pas du type "EN ACTIVATION ...".

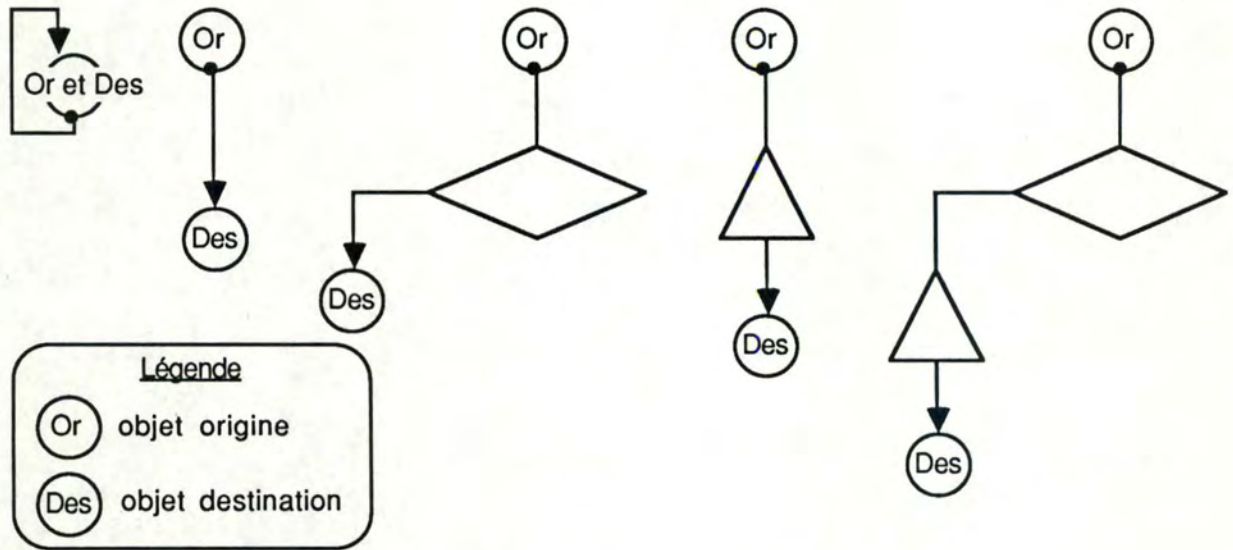


Figure VIII.2 droites qui composent une relation autre que "EN ACTIVATION ..."

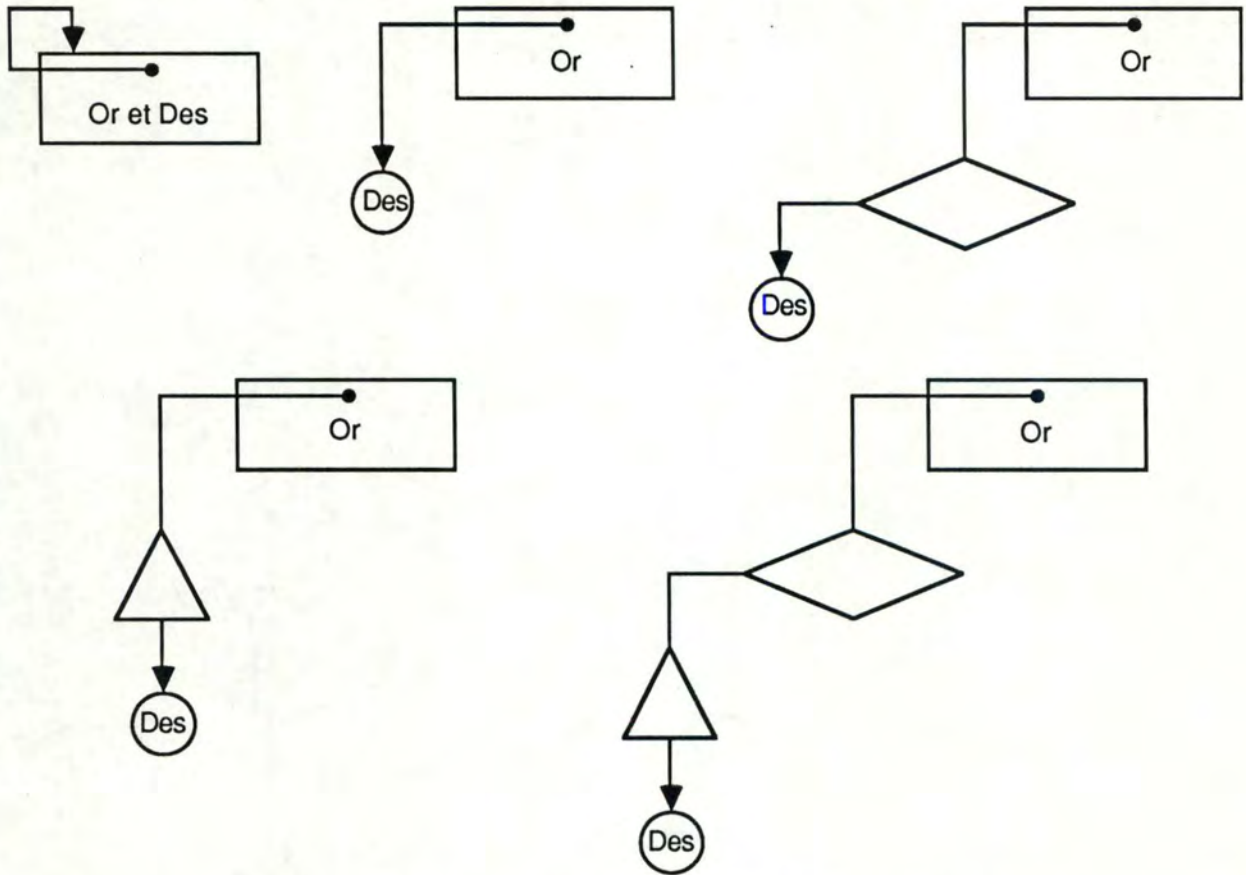


Figure VIII.3 droites qui composent une relation "EN ACTIVATION ..."

Combinaisons de relations

Lorsqu'un objet DSL apparaît dans plusieurs relations, on doit tenir compte de certaines règles particulières :

<R1> si un objet est l'origine de plusieurs relations DSL, il n'est représenté qu'une seule fois.

Suivant que la relation est du type "EN ACTIVATION ..." ou pas, nous aurons l'un des deux éclatements repris à la figure VIII.4.



Figure VIII.4 un objet est l'origine de plusieurs relations

<R2> si un objet est la destination de plusieurs relations DSL, il n'est représenté qu'une seule fois.

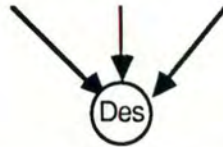


Figure VIII.5 un objet est la destination de plusieurs relations

<R3> lorsqu'une condition apparaît dans plusieurs relations DSL qui possèdent le même objet origine, la condition n'est pas répétée.

Ce qui signifie, par exemple, qu'on produira le schéma de gauche plutôt que le schéma de droite de la figure VIII.6.

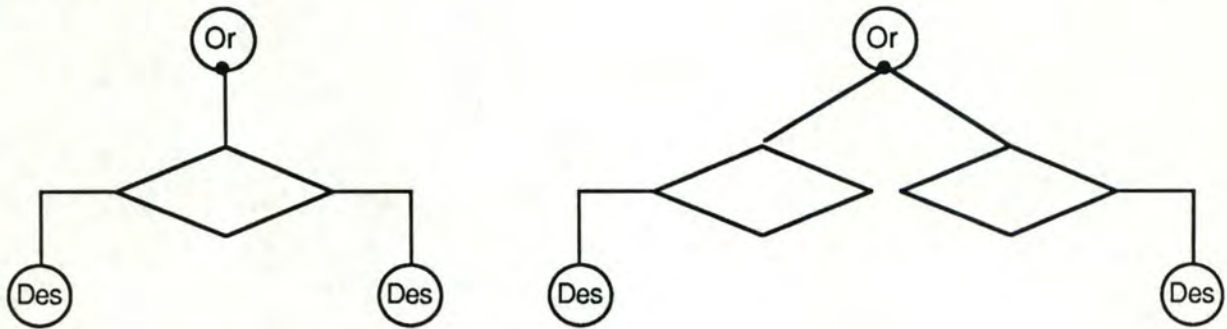


figure VIII.6 représentations de conditions

Toutefois, pour éviter la situation décrite à la figure VIII.7 (a) lorsqu'un processus est en même temps l'origine d'une relation "EN ACTIVATION" et d'une relation d'un autre type, la condition doit être répétée.



figure VIII.7

<R4> lorsqu'un mécanisme de duplication apparaît dans deux relations DSL, son équivalent graphique est répété.

Toutefois, pour éviter par exemple le dessin de la figure VIII.8 (a) lorsque les relations sont de même type et possèdent le même objet origine, les duplications ne sont pas répétées.

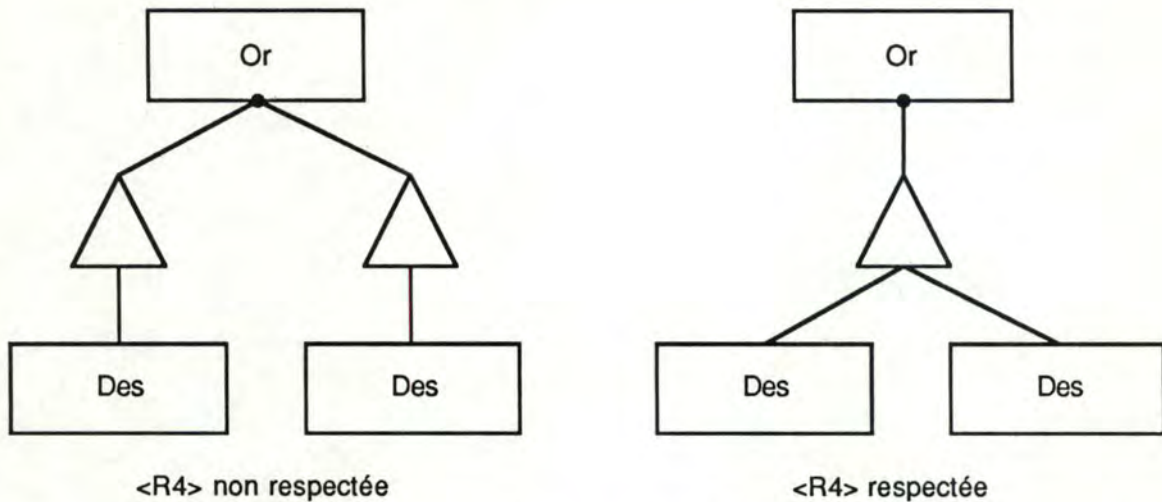
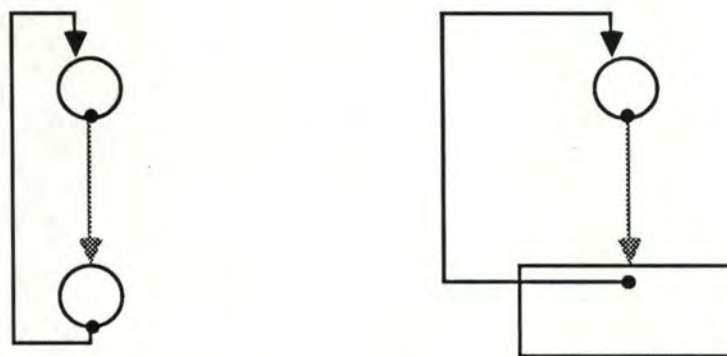


figure VIII.8 représentations de duplications

Circuit

Un deuxième type de combinaison possible de relations est celle qui conduit à un **circuit** dans le schéma de la dynamique.

Suivant que la relation est du type "EN ACTIVATION ..." ou pas, nous aurons l'une des deux boucles de la figure VIII.9.



(a) Relation autre que "EN ACTIVATION ..." (b) Relation "EN ACTIVATION ..."

Figure VIII.9

CRITERES ESTHETIQUES

Pour assurer une bonne lisibilité du schéma de la dynamique, celui-ci doit, nous semble-t-il, être dessiné en respectant les contraintes esthétiques suivantes :

- <1> rendre le schéma le plus **hiérarchisé** possible
- <2> respecter certaines contraintes de **représentation**
- <3> minimiser le nombre de **croisements**
- <4> présenter un dessin **équilibré** et le plus **vertical** possible

<1> Un schéma de la dynamique se présente classiquement de manière hiérarchisée. Ainsi, par exemple, si un processus P1 déclenche un processus P2, P1 est placé au-dessus de P2. Cette hiérarchisation est cependant impossible si le graphique présente un **circuit**. Dans ce cas, comme illustré à la figure VIII.10., le graphique devra être présenté de manière à respecter autant que possible la hiérarchie naturelle du dessin.

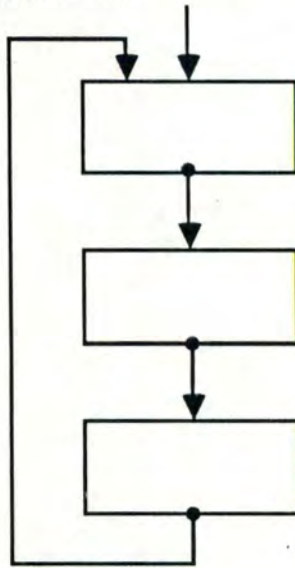


Figure VIII.10 circuit

<2> Certaines connections doivent apparaître brisées, comme illustré à la figure VIII.11.

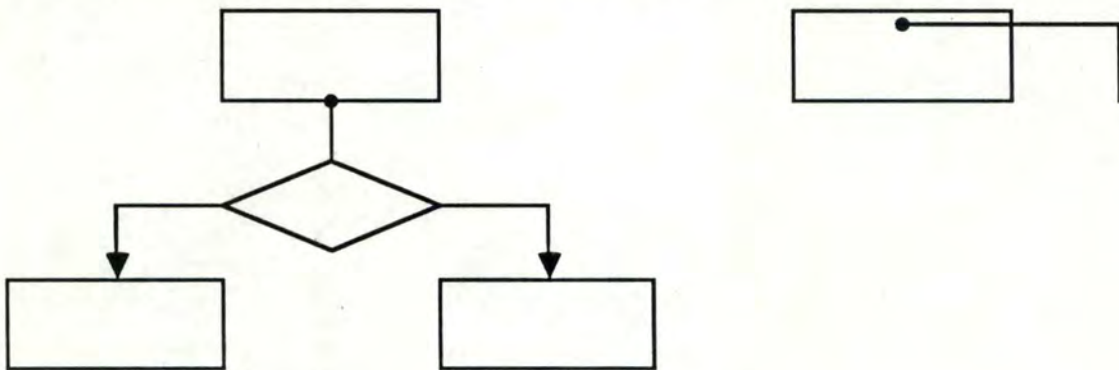


Figure VIII.11 droites brisées

<4> Centrer les pères par rapport à leurs fils et inversement, comme illustré à la figure VIII.12.

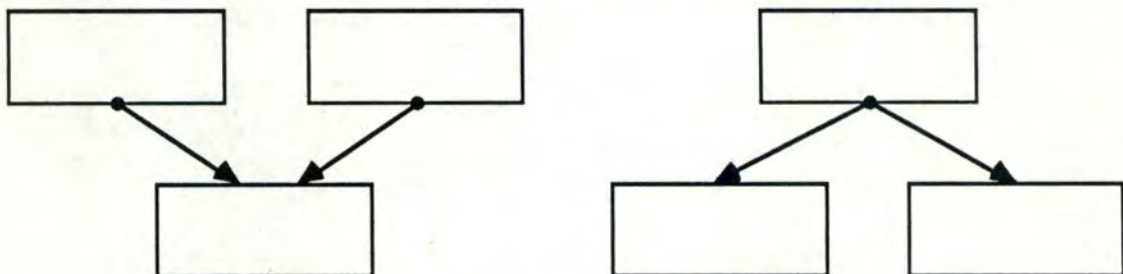


Figure VIII.12 symétrie du dessin

Tracer les connections le plus verticalement possible. Le dessin de droite de la figure VIII.13. est en effet préférable à celui de gauche de la même figure.

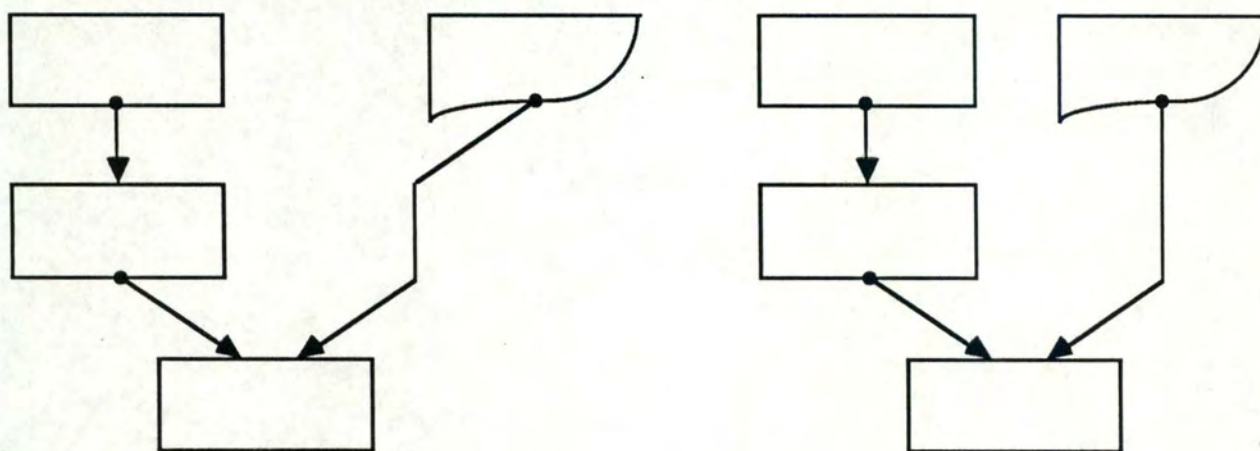


Figure VIII.13 dessin vertical

VIII.2. PRODUCTION AUTOMATIQUE DU SCHEMA DE LA DYNAMIQUE

Après avoir spécifié au point VIII.1. le problème envisagé dans ce chapitre, nous proposons une solution qui est une adaptation de la méthode de production automatique de graphes hiérarchisés présentée au chapitre VII.

Les étapes qui composent la méthode de production automatique du schéma de la dynamique feront d'abord l'objet d'une brève description où l'on mettra en évidence les difficultés que pose le dessin de ce diagramme. Nous verrons ensuite plus en détail les étapes spécifiques au schéma de la dynamique qui ne constituent donc plus une adaptation des étapes vues au chapitre précédent.

VIII.2.1. Méthode de production automatique du schéma de la dynamique

Avant d'aborder les étapes qui composent la méthode de production automatique du schéma de la dynamique, nous allons examiner les caractéristiques de ce diagramme qui font que la méthode vue au chapitre VII nécessite une adaptation pour qu'elle nous fournisse le résultat escompté.

1. CARACTERISTIQUES SPECIFIQUES AU SCHEMA DE LA DYNAMIQUE

Si l'on veut appliquer la méthode de production automatique de graphes hiérarchisés, on est confronté à deux problèmes : la présence éventuelle de circuits et certaines contraintes de représentation.

1.1. Présence de circuit

Contrairement au dessin d'une hiérarchie, un schéma de la dynamique peut contenir un circuit.

Par exemple, la terminaison d'un processus P2 peut déclencher un processus P1 qui, en terminaison, déclenche un processus P2 :

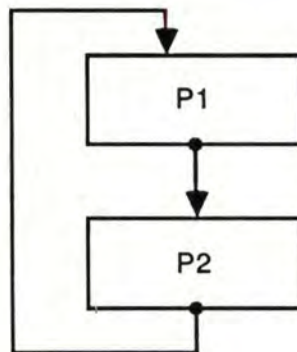


Figure VIII.14. Circuit dans un schéma de la dynamique

Or, pour pouvoir appliquer la méthode vue précédemment, le graphe qui constitue la représentation interne du diagramme à produire doit être sans circuit.

La méthode que nous proposons pour supprimer les circuits sera détaillée au point VIII.2.2. Elle consiste principalement à inverser un arc du circuit.

Nous verrons au point 1.2. que le dessin d'un circuit impose également l'introduction de deux sommets supplémentaires.

1.2. **Contraintes de représentation (droites non rectilignes)**

Si l'on se contente, lors de la construction de la représentation interne du schéma de la dynamique, d'associer un sommet à chaque objet DSL ayant un équivalent graphique, il n'est pas possible de respecter certaines contraintes de représentation émises au point VIII.1.

Par exemple, on obtiendrait la représentation de la condition reprise à la figure VIII.15.(a) au lieu de celle de la figure VIII.15.(b).

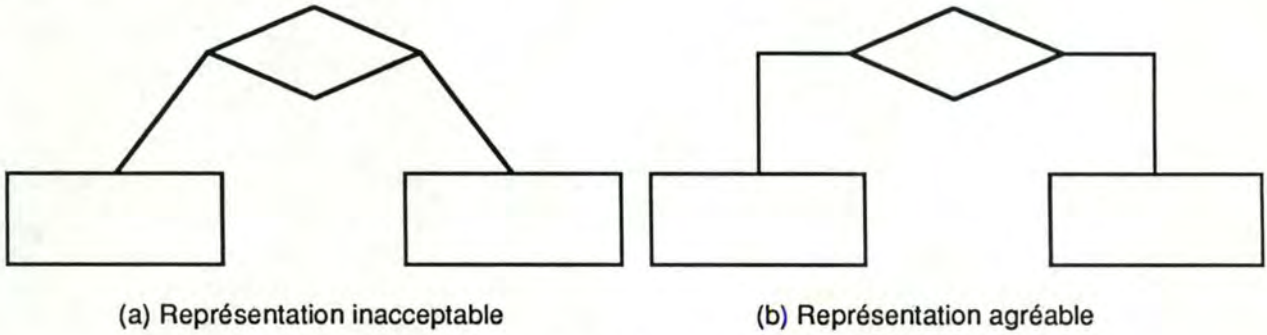


Figure VIII.15. Représentations d'une condition

Ce problème survient chaque fois qu'une droite entre deux objets doit apparaître brisée. Ce qui est le cas pour la branche vraie et fausse d'une condition, mais aussi lorsqu'on représente un circuit ou lorsqu'une relation est de type "EN ACTIVATION ...".

La solution, illustrée aux figures VIII.16. et VIII.17., que nous proposons pour résoudre ce problème consiste à ajouter un sommet supplémentaire (appelé sommet coude) chaque fois qu'une droite doit apparaître brisée. Ce sommet est relié par un arc (appelé arc horizontal) au sommet qui représente l'objet origine de la droite brisée (par exemple, le sommet qui représente la condition) et par un arc normal au sommet qui en est la destination

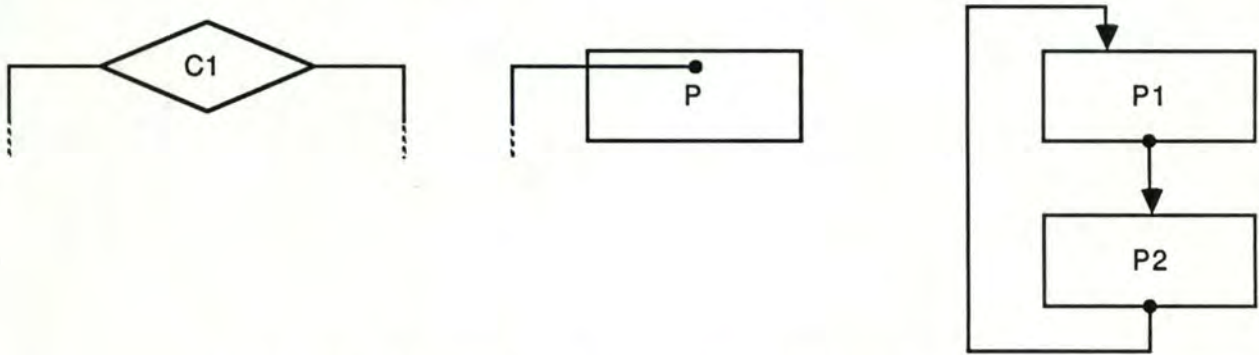


Figure VIII.16. Cas d'utilisation des coudes et des arcs horizontaux

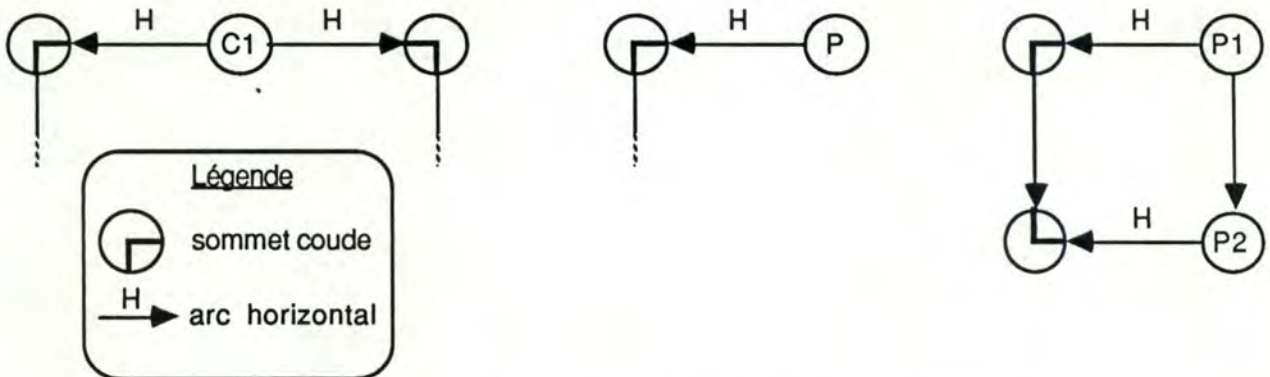


Figure VIII.17. Représentation interne des schémas de la figure précédente

De plus, les sommets reliés par des arcs horizontaux doivent vérifier les trois propriétés suivantes :

<P1> les sommets reliés par des arcs horizontaux possèdent le même niveau, ce qui permet d'éviter le dessin de la figure VIII.18.

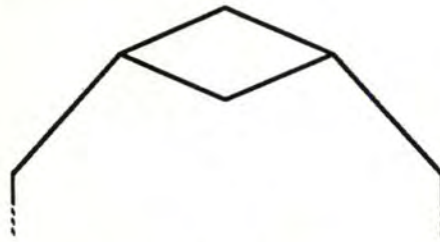


Figure VIII.18. <P1> non respectée

<P2> les sommets reliés par des arcs horizontaux forment un bloc, c'est -à-dire qu'ils occupent des positions successives sur leur niveau.

Par exemple, le dessin de la figure VIII.19. ne respecte pas cette propriété puisqu'un processus a été intercalé entre la branche vraie et la condition.

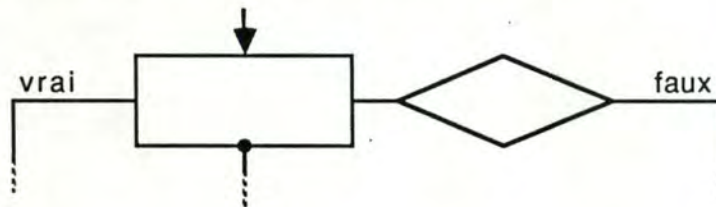


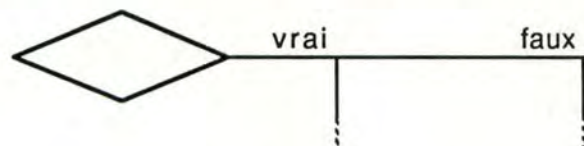
Figure VIII.19. <P2> non respectée

<P3> un sommet qui représente une condition est placé au milieu des coudes représentant les branches de la condition.

On préfère par exemple le dessin de la figure VIII.20.(a) au dessin de la figure VIII.20.(b)



(a) <P3> respectée



(b) <P3> non respectée

Figure VIII.20.

2. PRODUCTION AUTOMATIQUE DU SCHEMA DE LA DYNAMIQUE

La méthode que nous proposons pour produire automatiquement le schéma de la dynamique peut être décomposée en quatre étapes :

- construction d'un graphe orienté à partir de la spécification DSL du modèle de la dynamique à représenter graphiquement
- suppression éventuelle des circuits du graphe pour obtenir un graphe hiérarchisable
- calcul des coordonnées de chaque sommet en appliquant la méthode vue au chapitre VII
- dessin du schéma de la dynamique à partir de la position des sommets calculée à l'étape précédente et des conventions de représentation des objets et des relations du modèle de la dynamique

2.1. Première étape : construction du graphe

Cette étape a pour objet la construction d'un graphe à partir des spécifications d'un modèle de la dynamique.

La construction du graphe s'effectue comme suit :

- à chaque objet DSL ayant un équivalent graphique est associé un sommet
- une relation DSL est représentée par un ou plusieurs arcs qui relient les sommets associés aux objets de la relation

Par exemple,

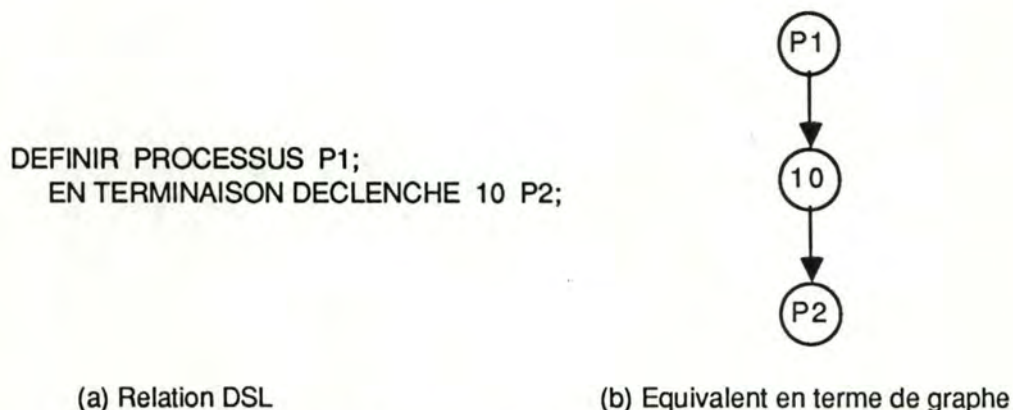


Figure VIII.21. Construction du graphe

- lorsqu'on est en présence d'une condition ou d'une relation du type "EN ACTIVATION ...", il y a création de sommets coudes et d'arcs horizontaux permettant le dessin de droites brisées.

Lorsque la représentation d'une relation conduit à une boucle dans le graphe (par exemple, lorsqu'un processus de type P1 en terminaison déclenche une autre occurrence de P1), l'arc qui provoque la boucle n'est pas repris dans le graphe. En effet, il ne s'agit pas d'un problème de placement de sommet mais plutôt d'un problème de dessin qui devra être résolu lors de la représentation graphique du schéma de la dynamique.

Par exemple, le graphe de la figure VIII.22. correspond aux spécifications du modèle de la dynamique dont la représentation graphique est illustrée à la figure VIII.32.

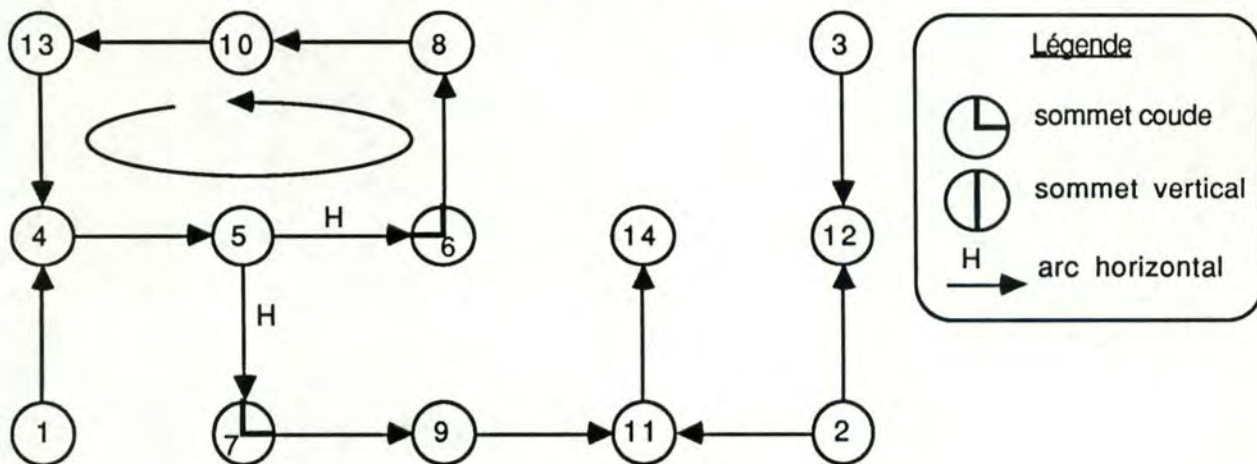


Figure VIII.22. Construction du graphe

2.2. Deuxième étape : suppression des circuits

Pour pouvoir appliquer la méthode décrite au chapitre VII, si le graphe obtenu au terme de la première étape comporte un circuit, celui-ci doit être éliminé. La technique que nous proposons à cet effet consiste à inverser un arc du circuit. Nous verrons au point VIII.2.5. que le choix de l'arc à retourner est très important pour la lisibilité du dessin et nous proposerons une méthode pour résoudre ce problème.

De plus, comme nous l'avons souligné au point 1.2., pour représenter les brisures de la droite qui remonte la hiérarchie naturelle du dessin pour constituer le circuit, il convient d'ajouter des sommets coudes et des arcs horizontaux.

La suppression des circuits appliquée à notre exemple produit le graphe de la figure VIII.23., où l'on peut constater l'ajout des sommets coudes 15 et 16, des arcs horizontaux (13,15) et (4,16) ainsi que l'arc (16,15) qui est l'équivalent de l'arc (13,4).

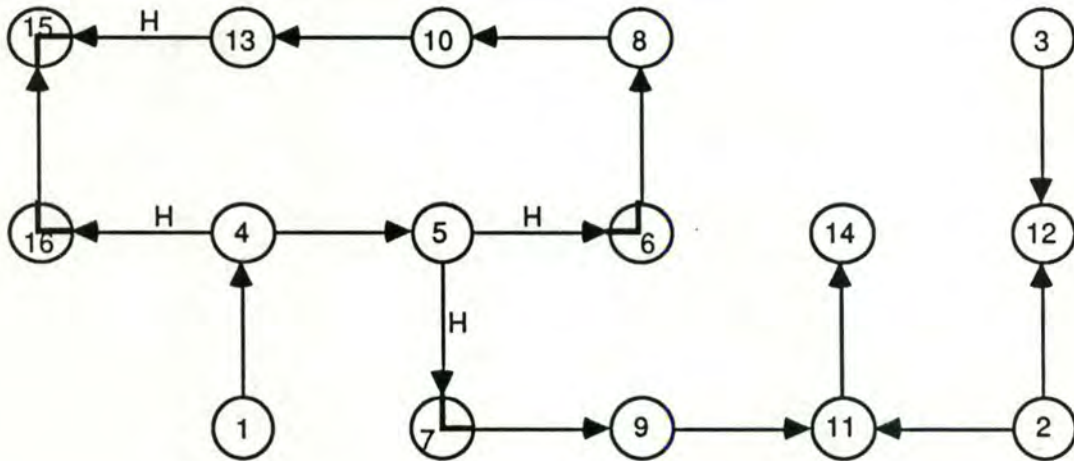


Figure VIII.23. Suppression des circuits

2.3. Troisième étape : calcul des coordonnées des sommets

Cette étape consiste à trouver une disposition dans le plan des sommets du graphe qui minimise le nombre de croisements et qui assure une répartition homogène des sommets. Elle est réalisée en appliquant la méthode de production automatique d'une hiérarchie adaptée au schéma de la dynamique pour tenir compte des propriétés des arcs horizontaux énoncées au point 1.2.

Rappelons que la production automatique d'une hiérarchie comprend les cinq étapes suivantes :

- <1> Construction de la hiérarchie des rangs
- <2> Serrer la hiérarchie
- <3> Rendre la hiérarchie propre
- <4> Minimiser le nombre de croisements
- <5> Calcul des coordonnées des sommets

Comme les trois premières étapes sont les seules concernées par le numéro de niveau des sommets, ce sont les seules qui doivent faire attention à ne pas violer la propriété <P1> qui impose que les sommets reliés par un arc horizontal se trouvent au même niveau.

Un moyen très simple pour respecter cette contrainte consiste à :

- agréger en un seul sommet tous ceux qui sont reliés par un arc horizontal
- appliquer les étapes <1>, <2>, et <3> au graphe ainsi obtenu
- désagréger le graphe et attribuer les mêmes numéros de niveau aux sommets qui ont été agrégés.

Nous allons illustrer l'étape d'agrégation, les étapes <1>, <2>, et <3> de la méthode de production automatique d'une hiérarchie ainsi que l'étape de désagrégation sur l'exemple de la figure VIII.23.

La figure VIII.24. représente le graphe obtenu après avoir agrégé les sommets de la figure VIII.23. reliés par un arc horizontal. On peut constater, par exemple le regroupement des sommets 5, 6, et 7 en un seul sommet. La hiérarchisation du graphe est reprise à la figure VIII.25.

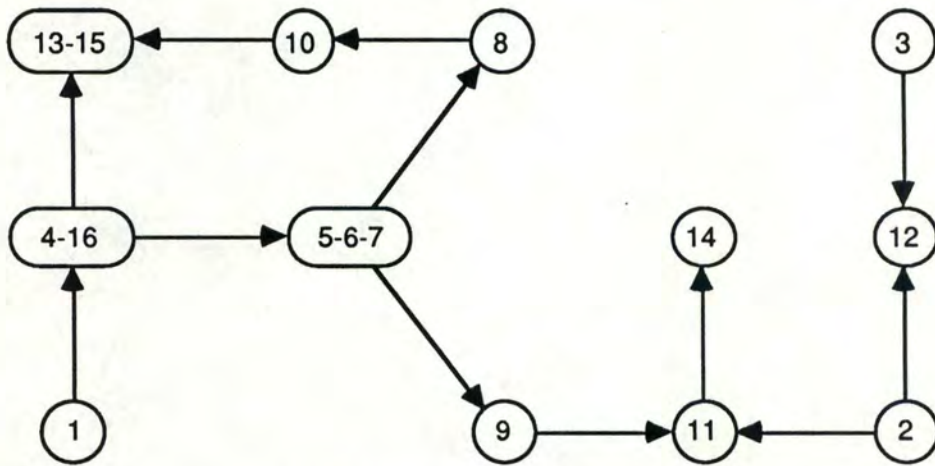


Figure VIII.24. Sommets agrégés

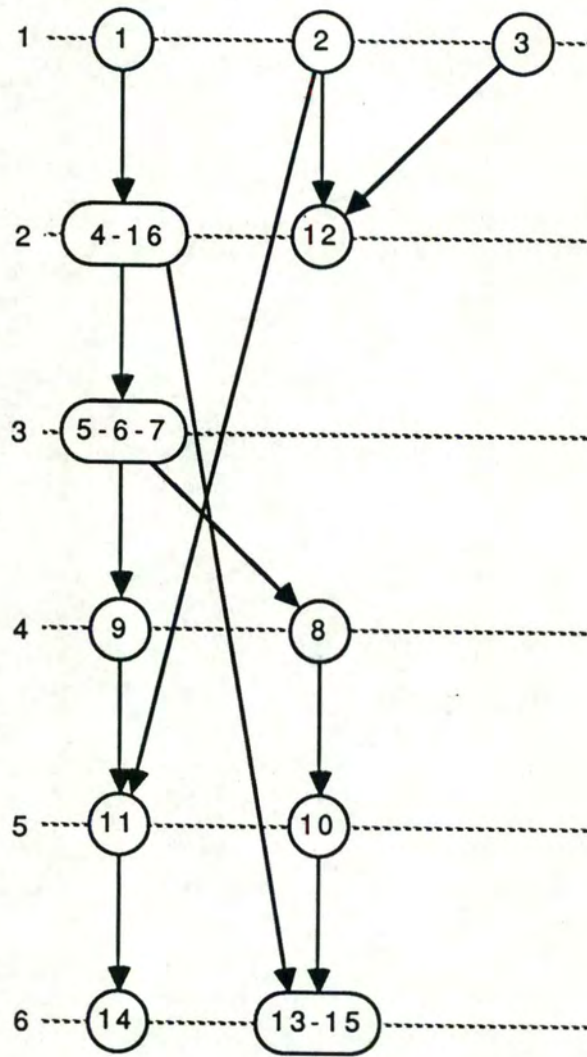


Figure VIII.25. Hiérarchisation

La construction de la hiérarchie serrée produit le graphe de la figure VIII.26. où l'on remarque le déplacement des sommets 2, 3, et 12 par rapport au graphe de la figure précédente.

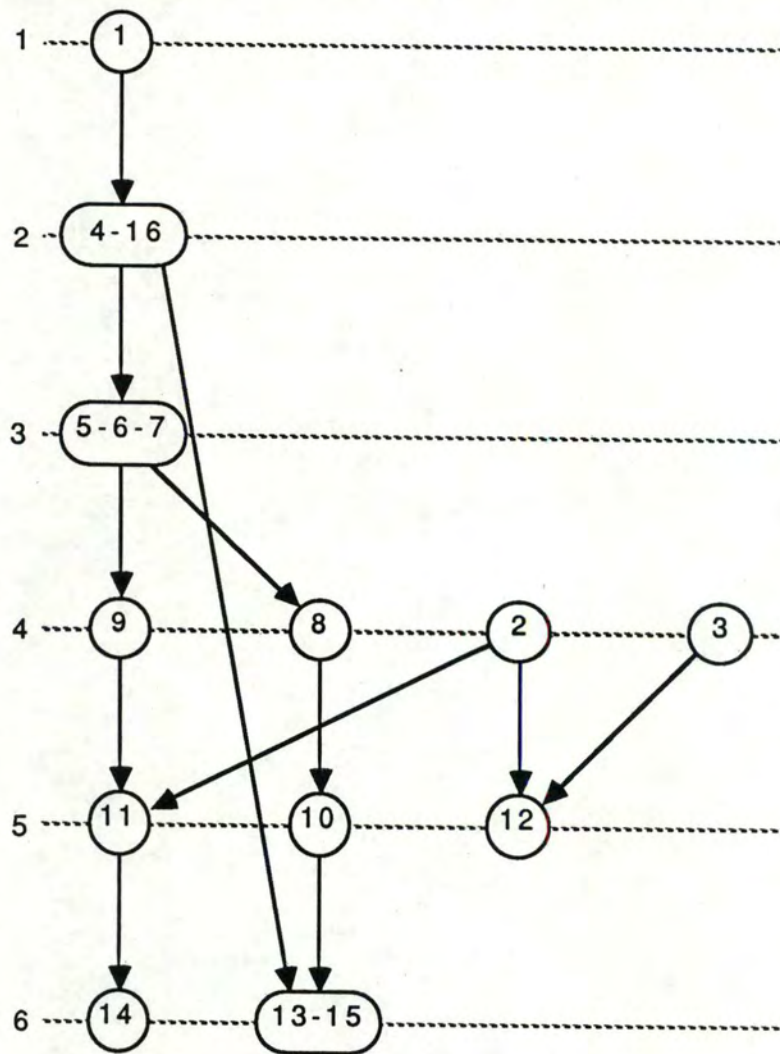


Figure VIII.26. Hiérarchie serrée

Rendre la hiérarchie de la figure VIII.26. propre consiste à remplacer l'arc (4-16,13-15) de portée égale à quatre par quatre arcs de portée égale à un en introduisant des sommets verticaux. Le graphe ainsi obtenu est celui de la figure VIII.27.

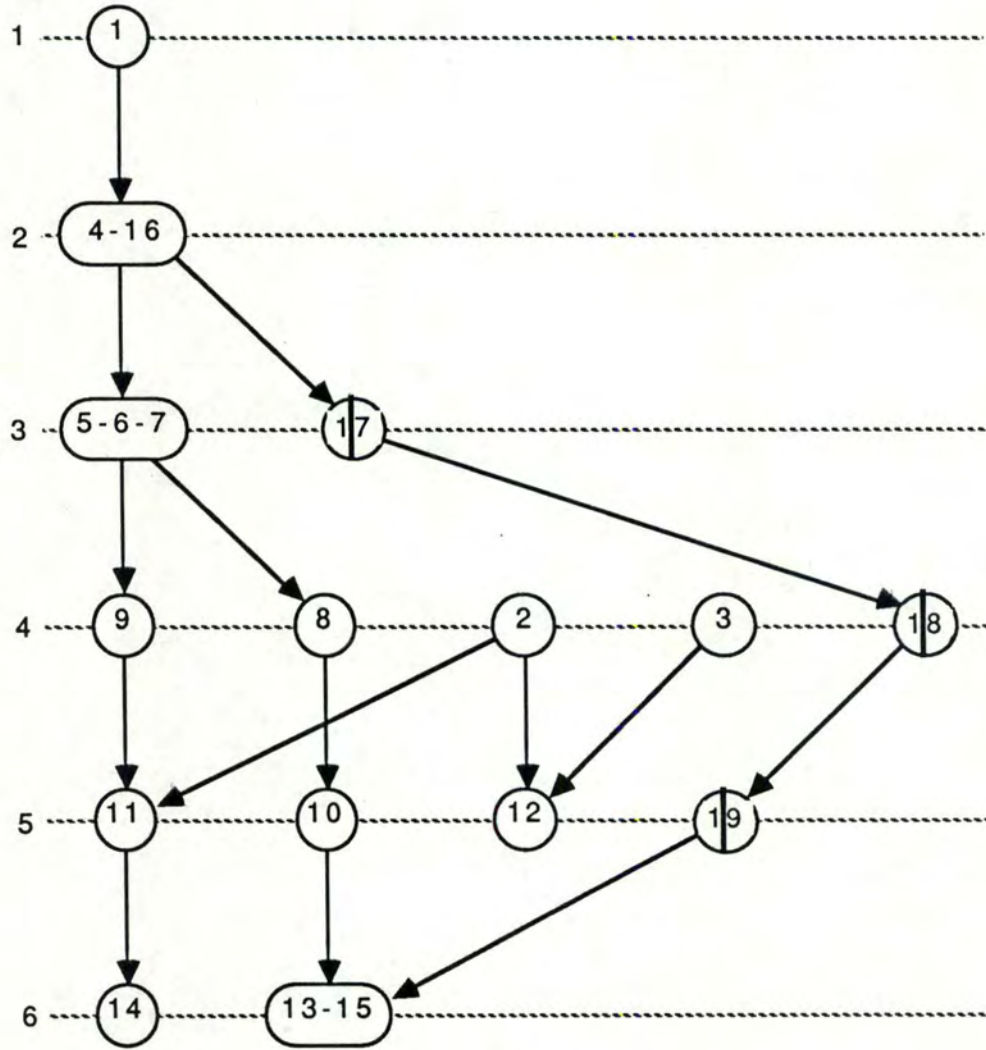


Figure VIII.27. Hiérarchie propre

La désagrégation du graphe est reprise à la figure VIII.28. On peut constater par exemple que le sommet 5-6-7 est éclaté en trois sommets qui sont placés au même niveau que le sommet 5-6-7.

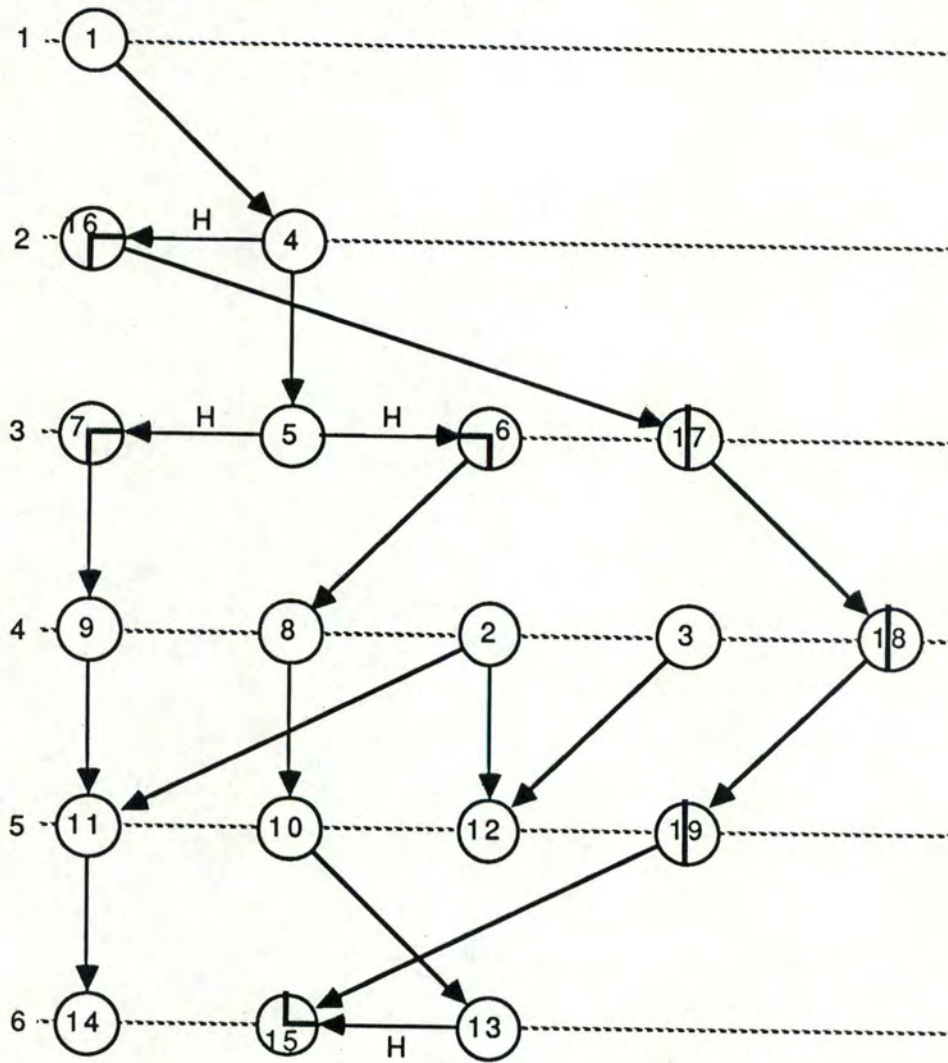


Figure VIII.28. Sommets désagrégés

Comme nous l'avons vu au chapitre VII, l'étape de minimisation du nombre de croisements consiste à ordonner les sommets à l'intérieur des niveaux. Pour la production du schéma de la dynamique, cet ordonnancement doit tenir compte de la propriété <P2> qui impose des positions successives pour des sommets reliés par des arcs horizontaux ainsi que de la propriété <P3> qui exige qu'une condition se trouve au milieu de ses branches vraie et fausse.

Pour que l'heuristique proposée au point VII.3.4 pour réduire le nombre de croisements tiennent compte de ces propriétés, il suffit qu'elle ne retienne, comme permutation possible, que celles qui respectent ces propriétés.

Autrement dit, connaissant l'ensemble P_L des permutations du niveau L, quand on détermine l'ensemble CP_L des couples de permutations qui minimisent le nombre de croisements entre les niveaux L et L+1, on ne retient que les permutations du niveau L+1 qui vérifient les propriétés en question.

Par exemple, le graphe de la figure VIII.29. montre le graphe de la figure VIII.28. après avoir minimisé le nombre de croisements en respectant les propriétés <P2> et <P3>.

Notons cependant qu'inverser l'ordre des sommets 5 et 6 n'introduit pas de croisement mais le graphe ainsi obtenu viole la propriété <P2> puisque le sommet 5 qui représente une condition n'est pas entre ses branches représentées par les sommets 6 et 7.

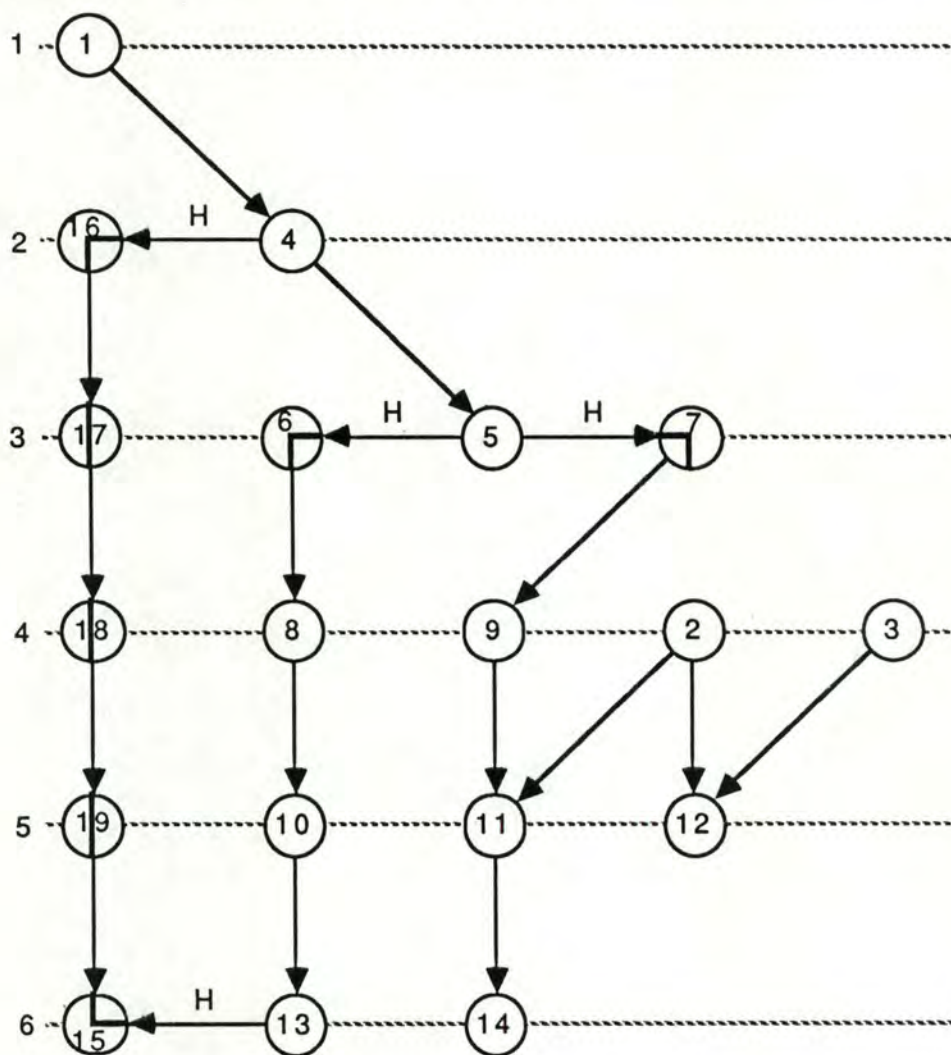
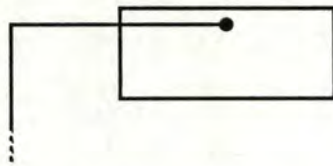


Figure VIII.29. Minimisation du nombre de croisements

Finalement, les coordonnées (X,Y) des sommets du graphe sont obtenues en appliquant le programme linéaire qui constitue la dernière étape de la méthode de production automatique de hiérarchies.

Dans la génération des contraintes du programme linéaire, il ne faut cependant pas tenir compte des arcs horizontaux car cela provoquerait un dessin incorrect.

Par exemple, on ne peut pas imposer, comme pour les arcs ordinaires, que le sommet origine d'un arc horizontal soit placé à la verticale du sommet destination on obtiendrait, dans ce cas, la figure VIII.30.(b) plutôt que la figure VIII.30.(a).



(a) dessin correct



(b) dessin incorrect

Figure VIII.30.

Le calcul des coordonnées pour notre exemple est obtenu en résolvant le programme linéaire décrit ci-dessous, ce qui produit le dessin de la figure VIII.31.

$$\text{MIN } \sum_{i=1..19} (x_i)$$

$x_1 = x_4$	$(x_{16} - x_4) \geq L$
$x_4 = x_5$	$(x_{17} - x_6) \geq L$
$x_{16} = x_{17}$	$(x_6 - x_5) \geq L$
$x_{17} = x_{18}$	$(x_5 - x_7) \geq L$
$x_6 = x_8$	$(x_{18} - x_8) \geq L$
$x_7 = x_9$	$(x_8 - x_9) \geq L$
$x_{18} = x_{19}$	$(x_9 - x_2) \geq L$
$x_8 = x_{10}$	$(x_2 - x_3) \geq L$
$x_2 = (x_{11} + x_{12}) / 0.5$	$(x_{19} - x_{10}) \geq L$
$x_{19} = x_{15}$	$(x_{10} - x_{11}) \geq L$
$x_{10} = x_{13}$	$(x_{11} - x_{12}) \geq L$
$x_{11} = (x_9 + x_2) / 0.5$	$(x_{15} - x_{13}) \geq L$
$x_{11} = x_{14}$	$(x_{13} - x_{14}) \geq L$
$x_{12} = (x_2 + x_3) / 0.5$	

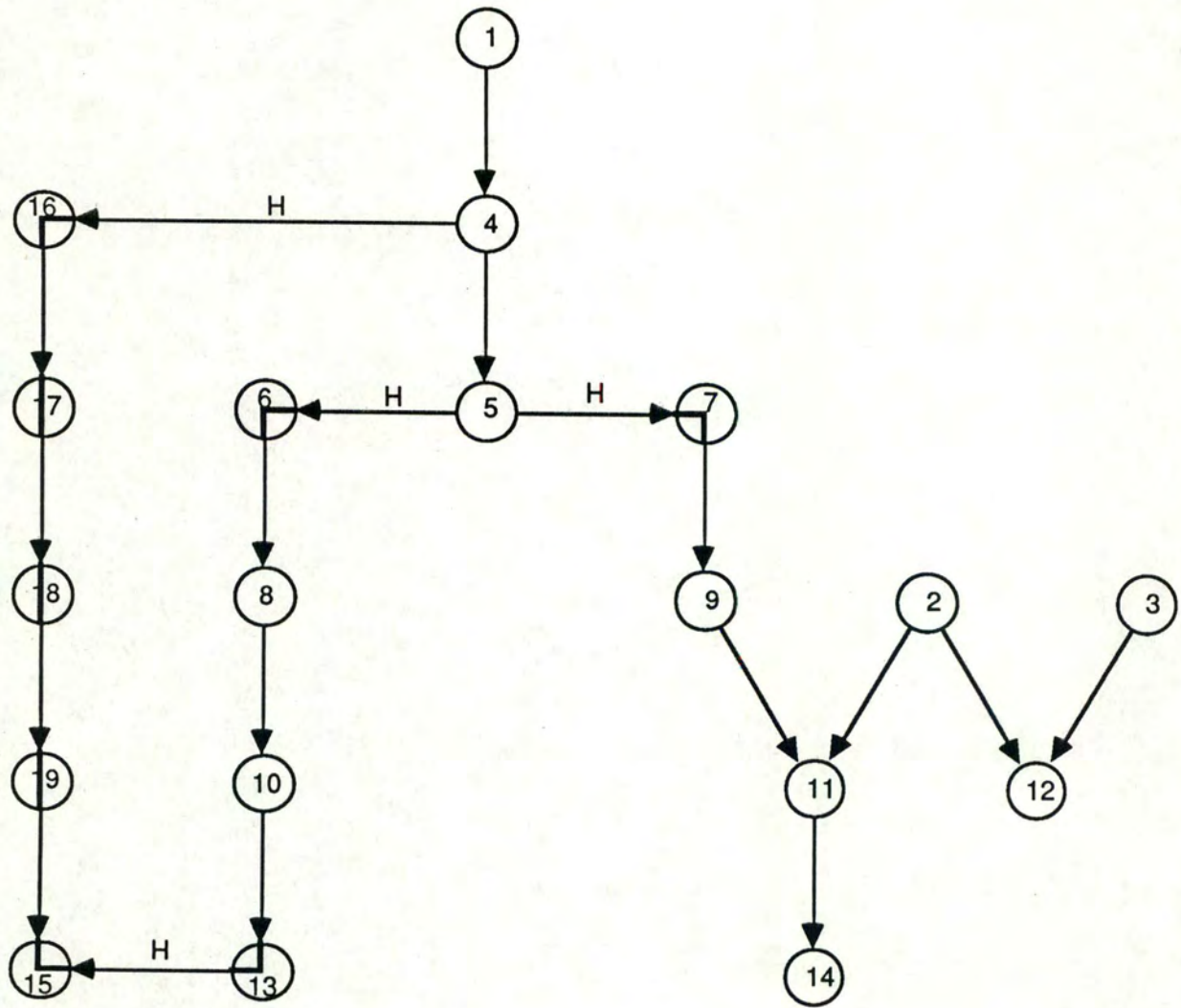


Figure VIII.31. calcul des coordonnées des sommets

2.4. Quatrième étape : Dessin du schéma de la dynamique

Nous avons défini un schéma de la dynamique comme un ensemble de représentations d'objets DSL et de relations entre ces objets.

Le dessin et le placement des objets DSL dans le plan est très facile puisque l'on connaît, d'une part, leurs conventions de représentation graphique et, d'autre part, les coordonnées des sommets associés à ces objets.

Le dessin d'une relation est plus délicat car il faut tenir compte des arcs entre les sommets, des sommets coudés et des sommets verticaux. Il convient également d'ajouter le point au début de la relation et la flèche qui termine la relation. On dispose toutefois de toutes les informations pour représenter graphiquement la relation :

- les arcs sont représentés par des droites rectilignes
- les sommets coudés sont représentés par un coude
- les sommets verticaux sont représentés par une droite verticale

Le schéma dynamique qui correspond à notre exemple est illustré à la figure VIII.32.

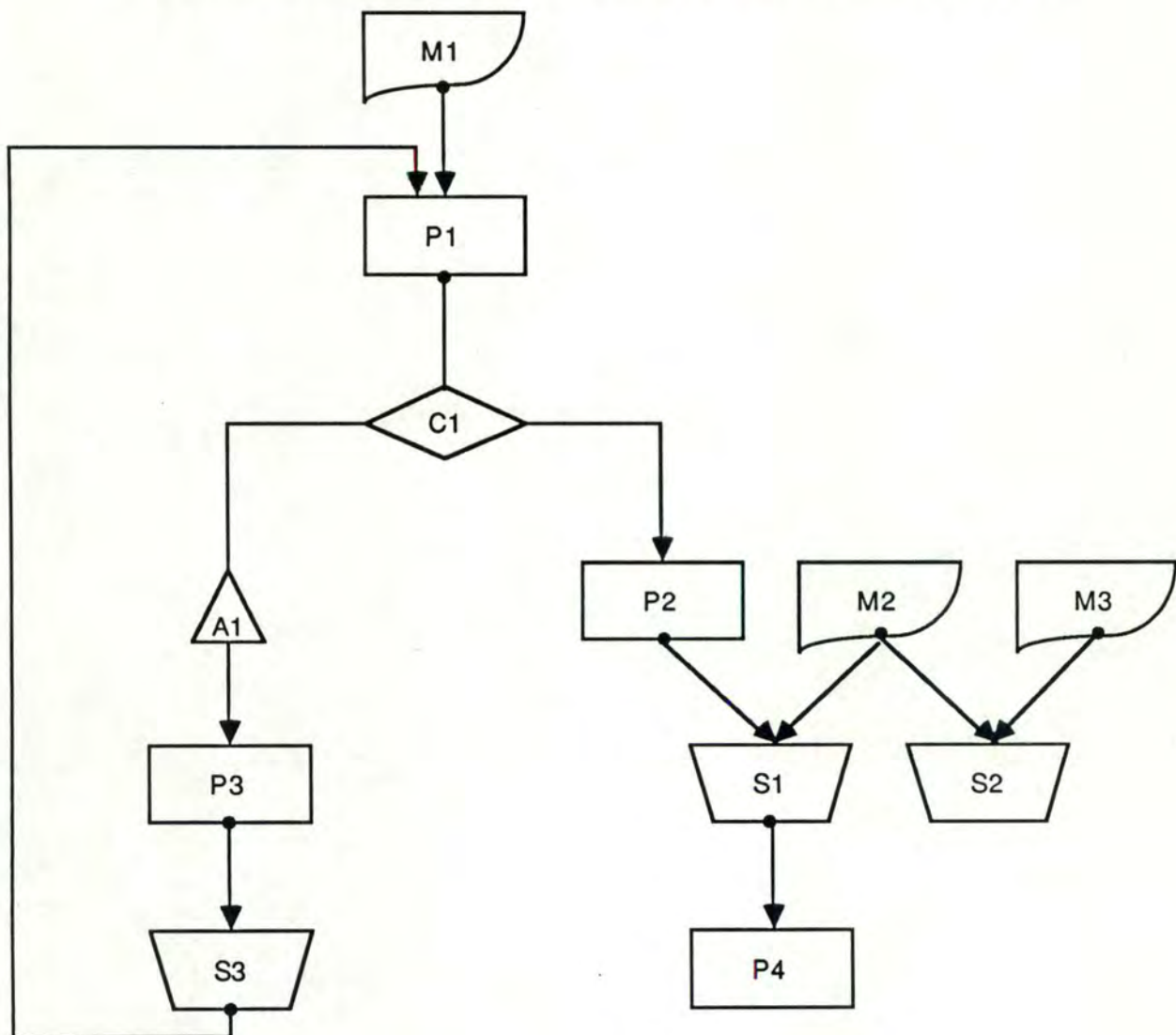


Figure VIII.32. Schéma de la dynamique

VIII.2.2. Etapes spécifiques au schéma de la dynamique

Nous avons vu au point VIII.2.1 une manière de produire automatiquement un schéma de la dynamique à partir de la spécification DSL du modèle correspondant.

Par rapport à la méthode de production automatique d'une hiérarchie, cette technique nécessite, comme nous l'avons vu, trois étapes spécifiques au schéma de la dynamique :

- construction d'un graphe orienté à partir de la spécification DSL du modèle de la dynamique à représenter graphiquement
- suppression éventuelle des circuits pour obtenir un graphe hiérarchisable
- dessin du schéma de la dynamique

VIII.2.2.1. Construction d'un graphe orienté

La construction d'un graphe orienté à partir des relations DSL qui spécifient un modèle de la dynamique pose un certain nombre de problèmes étant donné le nombre de cas à envisager pour respecter les conventions de représentation d'un schéma de la dynamique.

Les tables de décision reprises dans les annexes qui définissent la construction du graphe témoignent de la complexité du problème.

VIII.2.2.2. Suppression des circuits

Comme le graphe représentant de manière interne le schéma de la dynamique peut comporter des circuits, si on veut représenter le graphe de manière hiérarchisée, il est nécessaire de les supprimer. Nous allons détailler ici la façon dont nous comptons procéder.

METHODE POUR SUPPRIMER LES CIRCUITS

Une première idée pour résoudre le problème consiste à identifier les circuits du graphe et à retourner chaque fois un arc. Cette technique ne fonctionne cependant pas pour l'exemple de la figure VIII.33. En effet, si l'on inverse, dans le circuit composé des sommets s_1 , s_2 , et s_3 l'arc (s_2, s_3) , on introduit un nouveau circuit composé des sommets s_2 , s_3 , s_4 et l'on risque, par conséquent, de boucler indéfiniment.

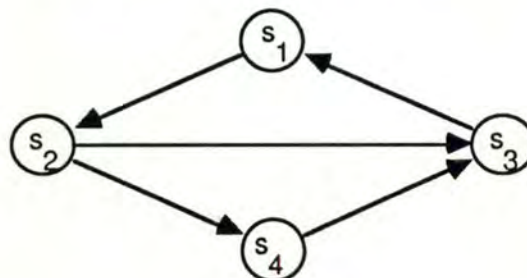


figure VIII.33. circuit

La méthode que nous avons adoptée pour supprimer les circuits dans le schéma de la dynamique se compose des **étapes** suivantes :

1. **Repérer les composantes fortement connexes** (noté C.F.C) du graphe
2. Si toutes les C.F.C identifiées au point 1 contiennent un seul élément, cela signifie que le **graphe est sans circuit** et on a terminé
 En effet, s'il existait un circuit, tous les sommets composant celui-ci feraient partie d'une C.F.C (de taille supérieure à 1) par la définition d'une C.F.C et d'un circuit.
3. Dans le cas contraire, on prend arbitrairement une C.F.C de taille supérieure à 1, dans laquelle on choisit un sommet s_i . On **inverse ensuite tous les arcs** qui ont s_i comme extrémité terminale et qui ont, pour extrémité initiale, un sommet appartenant à la C.F.C.
4. On **répète** le processus à partir de l'étape 1

L'algorithme se termine puisqu'à chaque itération, on introduit au moins une C.F.C en plus dans le graphe et le nombre de C.F.C d'un graphe est limité au nombre de sommets de celui-ci. Cette affirmation est démontrée au point suivant.

TERMINAISON DE L'ALGORITHME : DEMONSTRATION

La terminaison de l'algorithme est assurée si l'on montre qu'à chaque itération, le nombre de C.F.C dans le graphe augmente. Pour ce faire, nous allons procéder en trois étapes :

1. **Il n'existe pas de chemin qui n'appartient pas à une C.F.C et qui relie deux sommets appartenant à la C.F.C.**

Soit s un sommet d'un chemin μ tel que s n'appartient pas à la C.F.C, s_1 (resp. s_2) l'extrémité initiale (resp. terminale) de μ tel que s_1 et s_2 font partie de la C.F.C.

Pour tout s_i appartenant à la C.F.C par définition d'une C.F.C, il existe un chemin qui relie s_1 à s_i et un chemin qui relie s_i à s_2

Il existe un chemin qui relie s_1 à s puisqu'il existe un chemin qui relie s_i à s_1 et qu'il existe un chemin qui relie s_i à s .

De la même manière, il existe un chemin qui relie s à s_1 puisqu'il existe un chemin qui relie s à s_2 et qu'il existe un chemin qui relie s_2 à s_1 .

Par conséquent, s fait partie de la C.F.C, ce qui est contraire à l'hypothèse de départ.

2. **En inversant un ou plusieurs arcs d'une C.F.C, on ne crée pas de circuit qui passe par un sommet qui n'appartient pas à la C.F.C.**

Si tel était le cas, cela supposerait qu'il existe un chemin dont les extrémités appartiennent à la C.F.C et dont au moins un sommet n'appartiendrait pas à la C.F.C.

En effet, en inversant des arcs de la C.F.C on n'introduit pas d'arcs ayant pour extrémité un sommet n'appartenant pas à la C.F.C de départ.

Toutefois, un tel chemin ne peut pas exister en vertu de la première proposition.

3. **Si dans un graphe on inverse tous les arcs ayant pour extrémité terminale un sommet déterminé d'une C.F.C et pour extrémité initiale un sommet de la C.F.C, alors, on augmente le nombre de C.F.C du graphe.**

Cette proposition est démontrée si on prouve qu'en inversant tous les arcs ayant comme extrémité terminale un sommet s appartenant à une C.F.C et, pour extrémité initiale, un sommet de la C.F.C, le nombre de C.F.C du graphe augmente au moins d'une unité.

Supposons qu'on dispose d'un graphe G , en inversant les arcs de la manière décrite ci-dessus, on obtient un nouveau graphe G' .

G' contient toutes les C.F.C de G à l'exception de la C.F.C traitée. De plus,

-le sommet s constitue maintenant à lui seul une C.F.C. En effet, d'après la proposition 2, il ne peut pas appartenir à un circuit comprenant des sommets qui n'appartenaient pas à l'ancienne C.F.C et il ne peut plus être accédé par un sommet de son ancienne C.F.C.

-les sommets de l'ancienne C.F.C à l'exception de s constituent au moins une C.F.C. Rappelons que le nombre de ces sommets est ≥ 1 puisque la C.F.C traitée contenait au moins 2 sommets.

Par conséquent, G' contient au moins une C.F.C de plus que G .

CHOIX DU SOMMET DANS LA C.F.C TRAITEE

A l'étape 3 de l'algorithme que nous venons de décrire, on doit faire le choix d'un sommet dans la C.F.C retenue.

Il existe des méthodes qui déterminent l'ensemble minimum des sommets à supprimer dans un graphe pour que ce dernier soit sans circuit [SMITH, 75].

On aurait pu utiliser ces algorithmes pour choisir le sommet dans la C.F.C, ce qui nous aurait vraisemblablement conduit à inverser le moins possible d'arcs puisque le nombre d'itérations dans l'algorithme aurait été minimum.

Toutefois, dans le cas précis du schéma de la dynamique, suivant le choix du sommet dans la C.F.C considérée et donc, selon les arcs que l'on va inverser dans le graphe, on obtiendra une représentation graphique agréable ou pas.

Par exemple, dans la C.F.C illustrée à la figure VIII.34., si on choisit d'inverser l'arc (s_1, s_2) , on aura la représentation graphique décrite à la figure VIII.35. Par contre, si l'on choisit d'inverser les arcs (s_3, s_4) et (s_4, s_1) on obtiendra le graphique de la figure VIII.36.

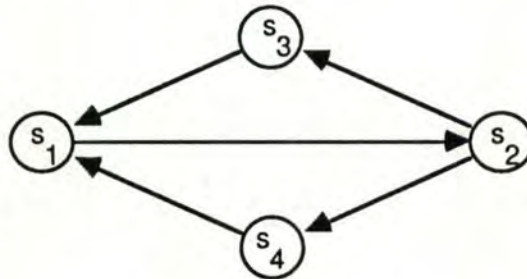


Figure VIII.34. C.F.C

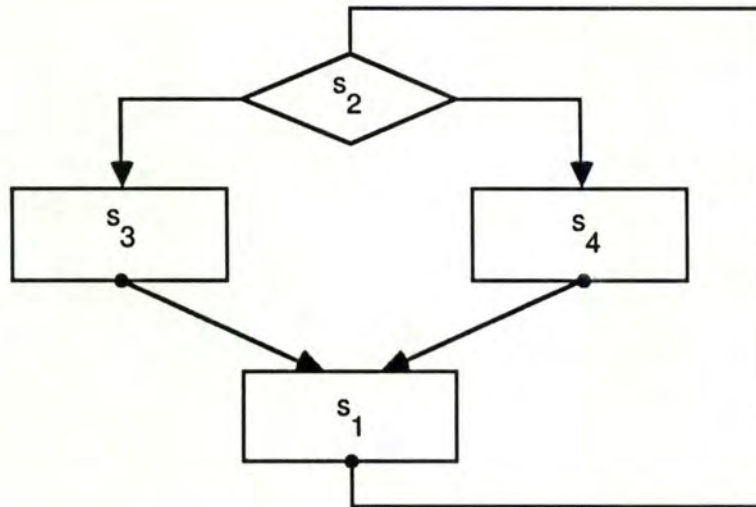


Figure VIII.35. Représentation peu agréable

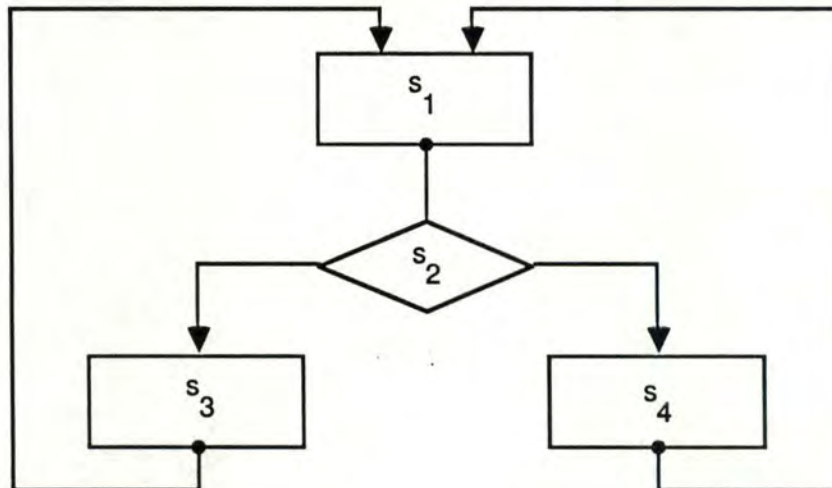


Figure VIII.36. Représentation agréable

Etant donné l'influence considérable du choix des arcs à retourner sur l'esthétique du dessin, nous avons préféré définir des critères de choix basés sur la sémantique du dessin plutôt que de privilégier le nombre minimum d'arcs à retourner.

Les **critères de choix** décrits ci-dessous sont classés par ordre d'importance:

- On choisit un sommet arbitrairement parmi les sommets correspondant aux objets DSL spécifiés par l'utilisateur pour extraire le schéma de la dynamique.

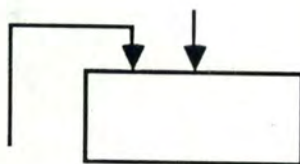
Il est logique de privilégier ces sommets car l'utilisateur les considère comme étant à l'origine de la dynamique.

- si aucun sommet de ce type n'est présent dans la C.F.C, on choisit un sommet parmi les sommets qui sont l'extrémité terminale d'un arc dont l'extrémité initiale n'appartient pas à la C.F.C considérée.

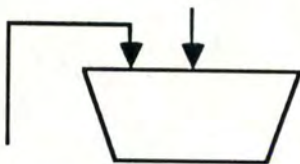
Notons qu'un tel sommet est toujours présent dans la C.F.C puisque, d'une part, aucun des sommets faisant partie de la C.F.C ne correspond à un objet de départ et d'autre part, d'après la définition du schéma de la dynamique, il existe toujours un chemin qui relie un objet de départ à un sommet de la C.F.C.

Le choix de ce sommet se fera en respectant les priorités suivantes :

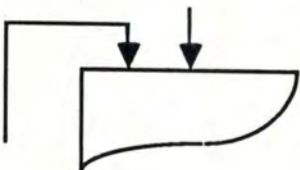
on préférera un processus



à un point de synchronisation. Et l'on préférera un point de synchronisation



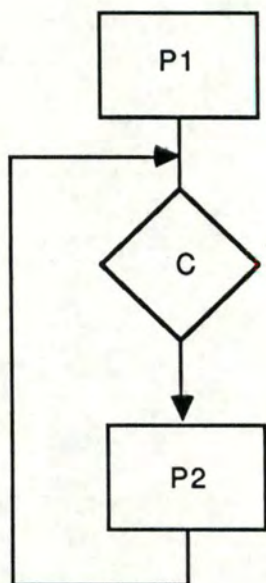
à un message



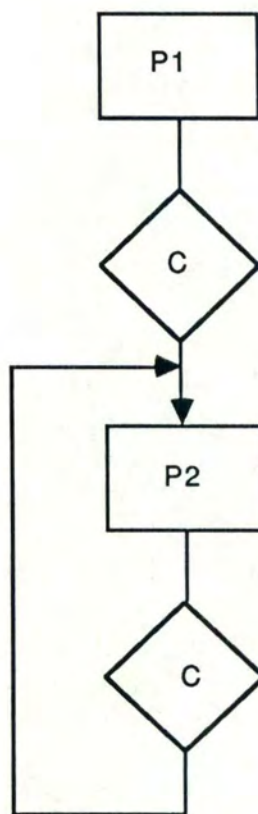
Notons qu'un sommet peut également correspondre à une condition ou à un déclenchement multiple. Mais ces sommets ne seront jamais choisis et ce, pour deux raisons :

- l'utilisateur ne peut pas les sélectionner comme objet de départ du schéma de la dynamique.
- un sommet représentant une condition ou un déclenchement multiple ne peut pas être l'extrémité terminale de deux arcs différents d'après la manière dont on construit le graphe qui représente le schéma de la dynamique.

De sorte que l'on n'aura jamais le dessin de la figure VIII.37.(a), mais plutôt le dessin de la figure VIII.37.(b)



(a) dessin correct



(b) dessin incorrect

Figure VIII.37. conditions

VIII.2.2.3. Dessin du schéma de la dynamique

Dans le cadre du poste de travail IDA, le dessin du schéma de la dynamique est réalisé par un pilote dont la spécification sort du cadre de ce mémoire.

CONCLUSION

La méthode que nous venons de présenter pour générer automatiquement le schéma de la dynamique respecte les conventions de représentation graphique de ce schéma ainsi que les critères esthétiques posés sur ce diagramme. Nous avons donc tout lieu d'être satisfaits.

Cependant, pour tenir compte de certains critères qui font intervenir la sémantique du dessin, il serait intéressant de prévoir une interaction limitée avec l'utilisateur qui permettrait à ce dernier d'imposer des contraintes de disposition propre au dessin. Une interaction serait utile :

- lors de la suppression de circuits car, comme nous l'avons vu, le choix de l'arc à retourner est important pour la représentation et les critères de choix que nous avons définis sont peut-être trop catégoriques.
- et surtout, lors de la construction de la hiérarchie serrée qui permettrait à l'utilisateur de préciser les objets qui peuvent descendre de niveau. Nous pensons notamment aux messages qui ne sont pas des messages horaires et que l'on préfère placer au premier niveau de la hiérarchie.

Cette interaction nécessiterait peu de changement au niveau des algorithmes et conduirait à un dessin personnalisé tout en étant produit automatiquement.

Bibliographie

- [Batini, 85] Batini.C., Nardelli E., Talamo M., Tamassia R. *GINCOD : A graphical tool for conceptual design of data base applications* , Albano A., De Antonellis V., and Di Leva A., 1985
- [Batini, 86] Batini.C, Nardelli E., and Tamassia.R *A layout algorithm for data flow diagrams*, IEEE Transactions on Software Engineering, Vol. SE-10, N.4, Apr. 1986.
- [Bodart, 87] Bodart.F, Pigneur.Y *Conception assistée des applications informatiques, 1.-Etude d'opportunité et analyse conceptuelle*, MASSON, 1987
- [Carpano, 80] Carpano, M-J. *Automatic display of hierarchized graphs for computer-aided decision analysis*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-10, N.11, Nov. 1980.
- [Delarche, 79] Delarche, M. *Quelques outils Infographiques pour l'analyse structurale des systèmes* , Dr. Ing. thesis, Univ. Grenoble, Jun. 1979.
- [DSL, 87] DSL, version française, manuel de référence, Institut d'Informatique, FNDP, Namur, 1987.
- [Fichet, 84] Fichet, J : *Théorie des graphes. Note de cours* .
- [Hainaut, 86] Hainaut J-L. *Conception assistée des applications informatiques, 2. Conception de la base de données*, MASSON, 1986.
- [Hopcroft, 74] Hopcroft, J. and Tarjan, R. *Efficient planarity testing*, Journal of the ACM, Vol. 21, N. 4, Oct. 1974.
- [IEW] IEW, Information Engineering Workbench, Arthur Young Proware.
- [Jackson, 75] Jackson, M. *Principles of program design*, Academic Press, 1975.
- [Karp, 75] Karp, R.M. *On the computational complexity of combinatorial problems* Networks, Vol 5, Jan. 75.
- [Reingold, 81] Reingold, E. and Tilford, J. *Tidier drawings of trees*, IEEE Transactions on Software Engineering, Vol. SE-7, N.2, Mars 1981.
- [Rubin, 75] Rubin, F. *An Improved algorithm for testing the Planarity of a Graph*, IEEE Transactions on Computers, Vol. C-24, N.2, Feb. 1975.
- [Smith, 75] Smith, W. and Walford, B. *The identification of a minimal feedback vertex set of a directed graph*, IEEE Transactions on Circuit And Systems, Vol. CAS-22, N.1, Jan. 1975.
- [Tamassia, 83] Tamassia, R. and Batini, C and Talamo, M. *An algorithm for automatic layout of entity relationship diagrams*, Entity-Relationship approach to Software Engineering, C.G. Davis, J. Jajodia, P.A., Ng and R.T. Yeh, ER Institute, 1983.
- [Warfield, 77] Warfield, J. *Crossing theory and hierarchy mapping*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, N.7, Jul. 1977.
- [Wetherel, 79] Wetherel, C. and Shannon, A. *Tidy drawings of trees* , IEEE Transactions on Software Engineering, Vol. SE-5, N.5, Sept 1979.
- [Yeth, 81] Yeth, D. *Improved planarity algorithms*, BIT, 22, Aug. 1982

CONCLUSION GENERALE

Notre objectif, au départ de ce mémoire, consistait à proposer un ensemble d'outils graphiques d'interprétation des résultats produits par DSL-SIM destinés à pallier au manque de convivialité et à l'extrême complexité d'analyse de la forme tabulaire qui existe actuellement.

Nous avons procédé à l'analyse des besoins, ce qui nous a conduits à distinguer essentiellement deux formes de représentation : une présentation classique en forme de camemberts, diagrammes à bâtons, ... pour les mesures statistiques ainsi qu'une présentation sous la forme d'enchaînements qui s'apparente au dessin de graphes pour la juxtaposition d'informations sur le schéma de la dynamique et pour visualiser le comportement du système d'information (graphe des événements, animation, ...).

Lors de cette analyse, nous avons rencontré des difficultés pour trouver des formes simples, significatives et faciles à comprendre étant donné le volume de données produites par DSL-SIM. De plus, comme le mécanisme d'interprétation de ces résultats est un processus non structuré, il est donc nécessaire de disposer d'un interface qui permet à l'utilisateur de naviguer au travers des outils proposés.

La spécification du problème pour les mesures statistiques produites par DSL-SIM n'a pas posé de difficultés majeures. Par contre, pour les enchaînements, nous avons rencontré deux types de problèmes.

Tout d'abord, au niveau conceptuel, nous avons constaté l'absence du concept de transaction pour évaluer et visualiser le comportement global du système d'information.

En ce qui concerne la réalisation de ces graphiques, nous avons dû développer différentes méthodes de construction car pour ce type de schéma, nous n'avons pas rencontré dans la littérature d'algorithme résolvant de manière satisfaisante le problème.

Compte tenu de l'ampleur de ce travail, il ne nous a pas été possible de procéder à l'implémentation. L'architecture logicielle pour la production du schéma de la dynamique a été réalisée, mais faute de temps, nous n'avons pas pu l'insérer dans ce mémoire.




La perspective à court terme de notre travail est la réalisation dans le cadre du poste de travail IDA des outils proposés. A plus long terme, il s'agira de vérifier les hypothèses et les choix que nous avons faits tout au long de ce mémoire en examinant les réactions des utilisateurs.

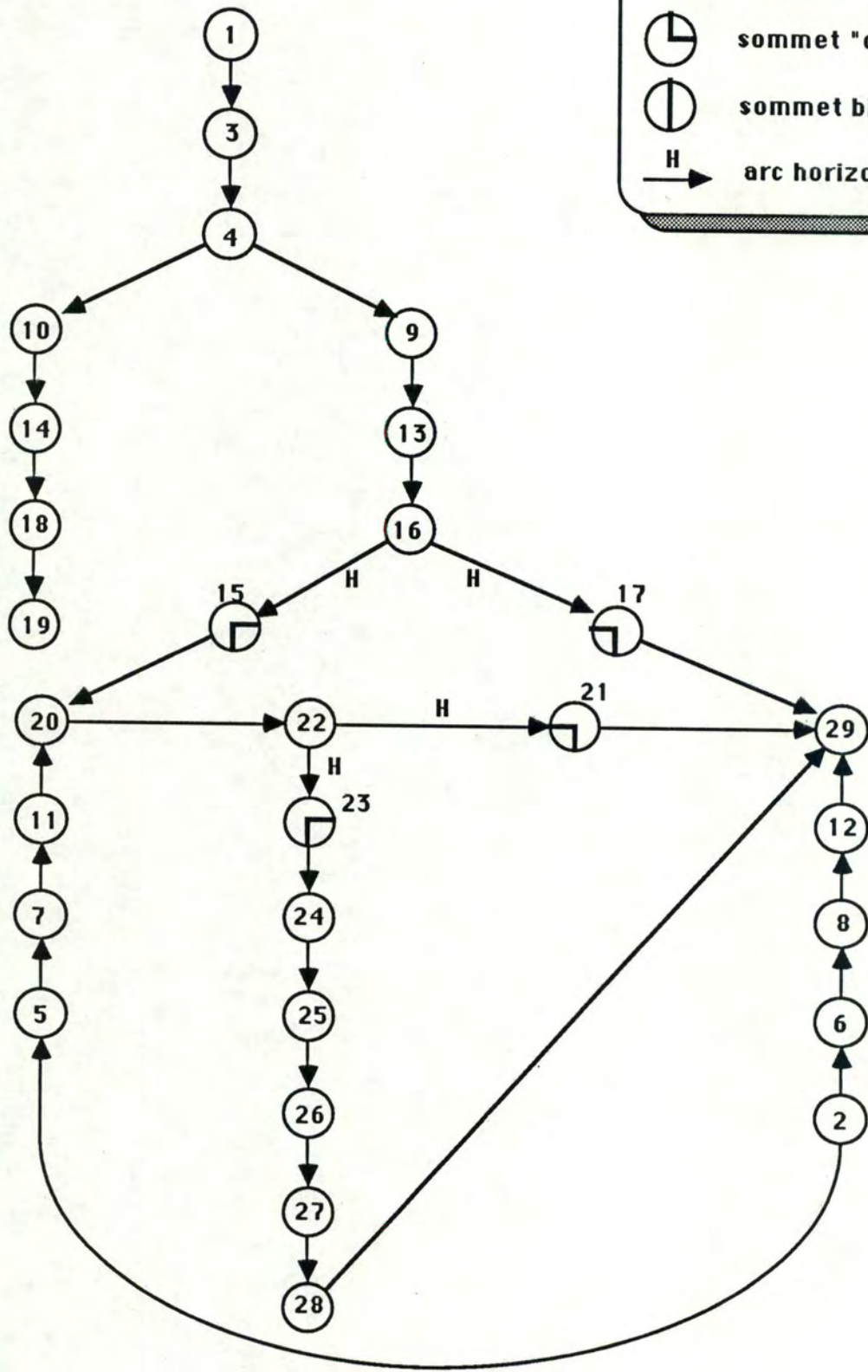
ANNEXES

ANNEXE A

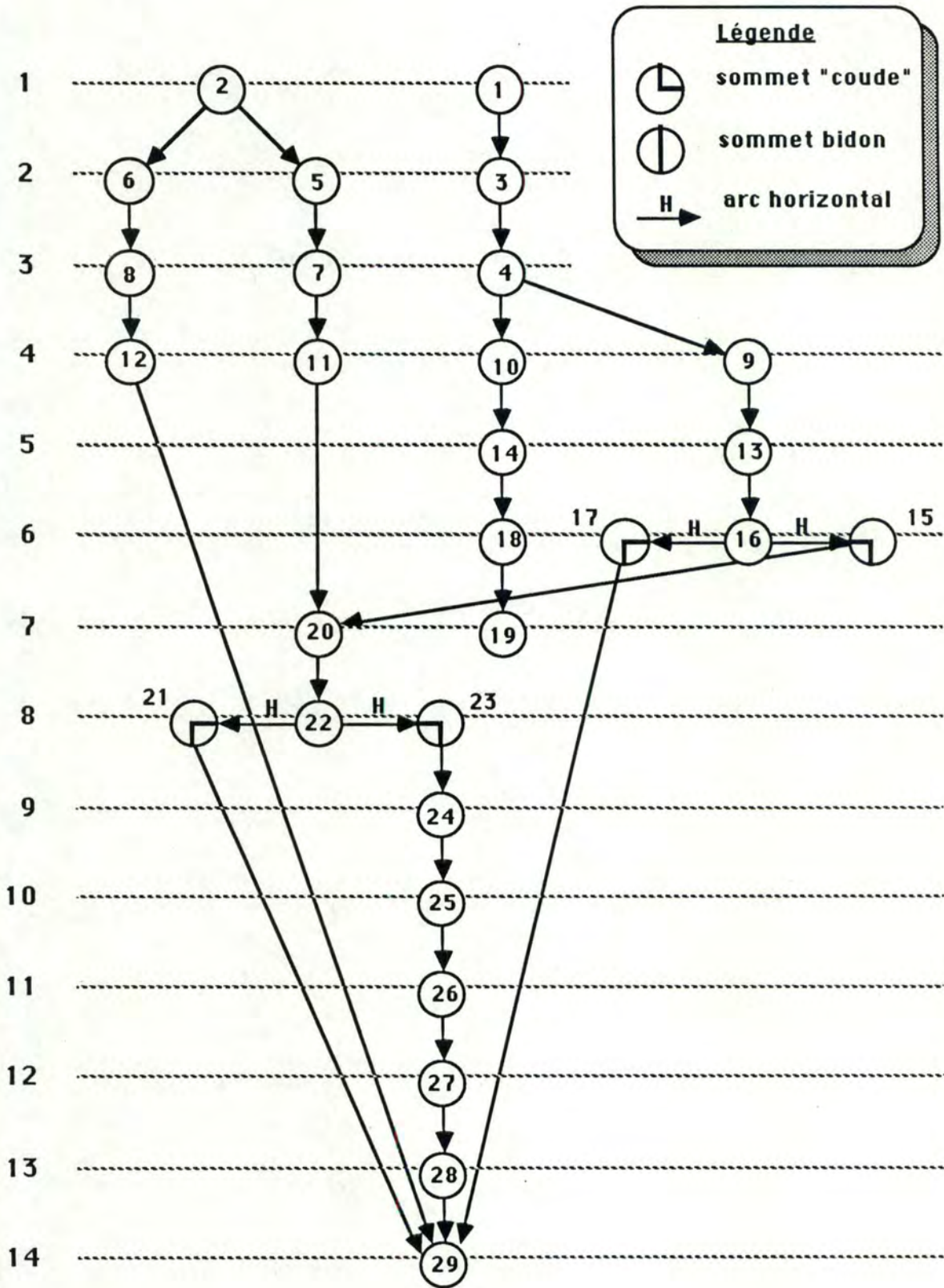
Construction du graphe

Légende

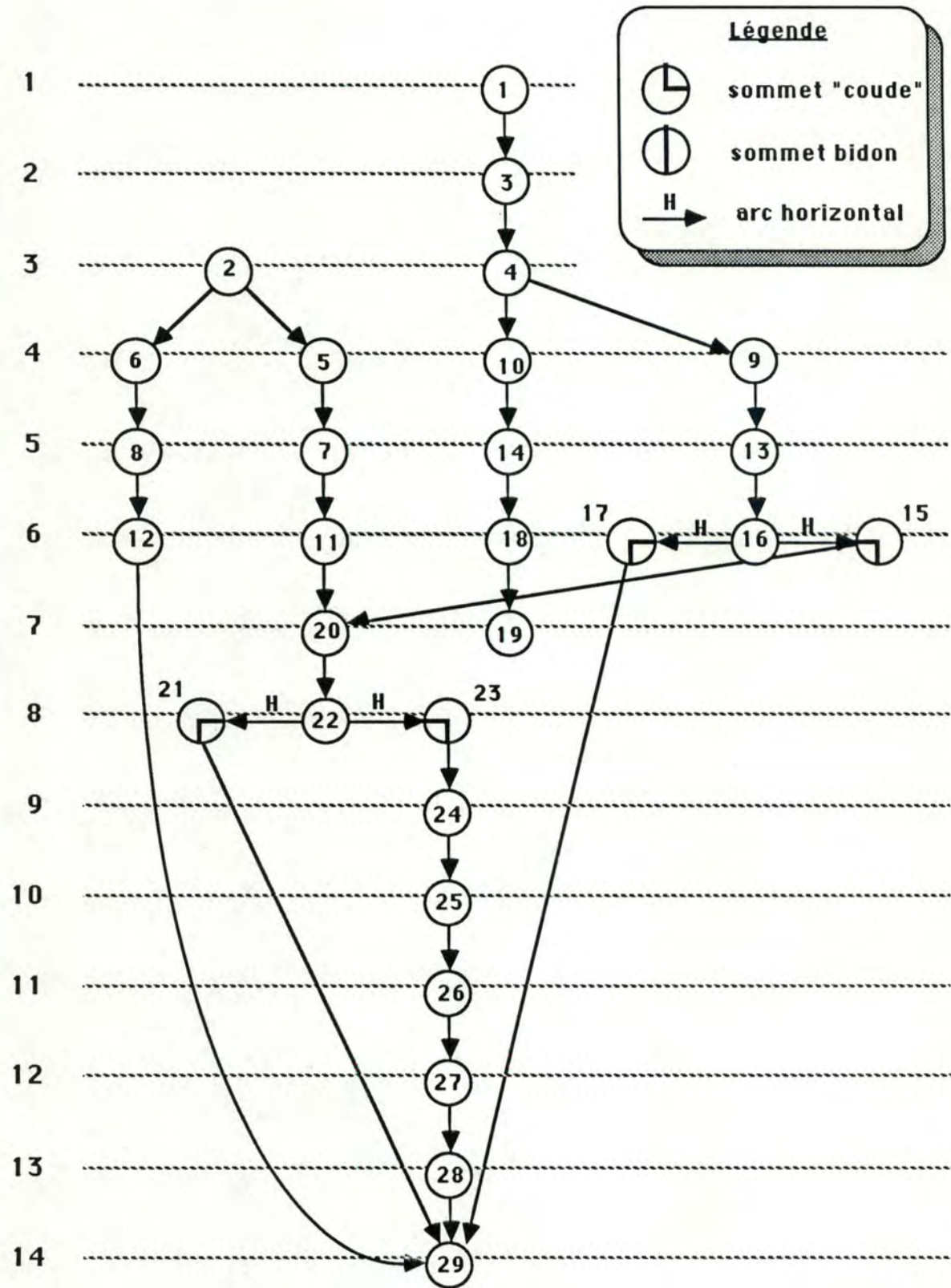
-  sommet "coude"
-  sommet bidon
-  arc horizontal



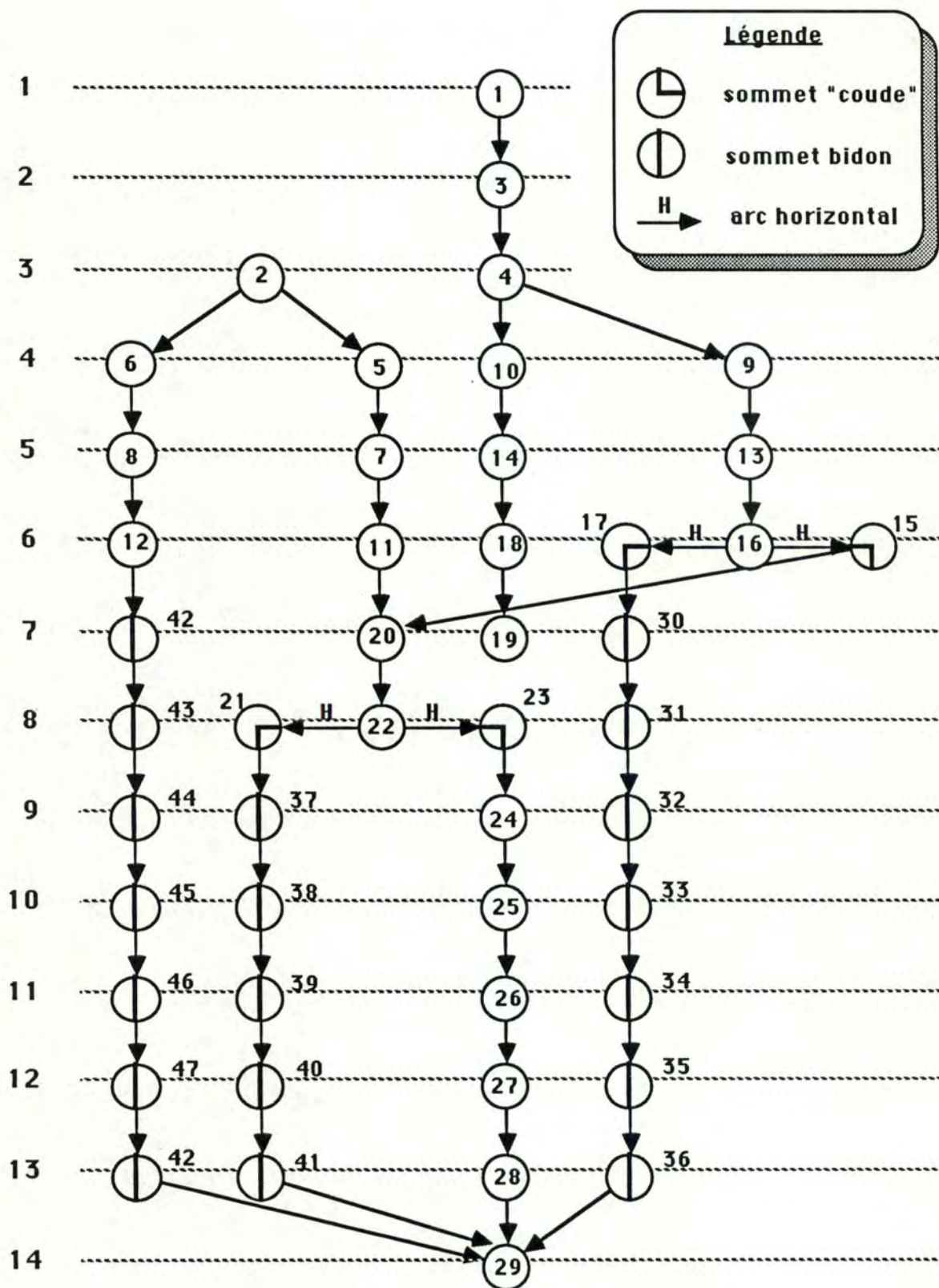
Hiérarchisation du graphe



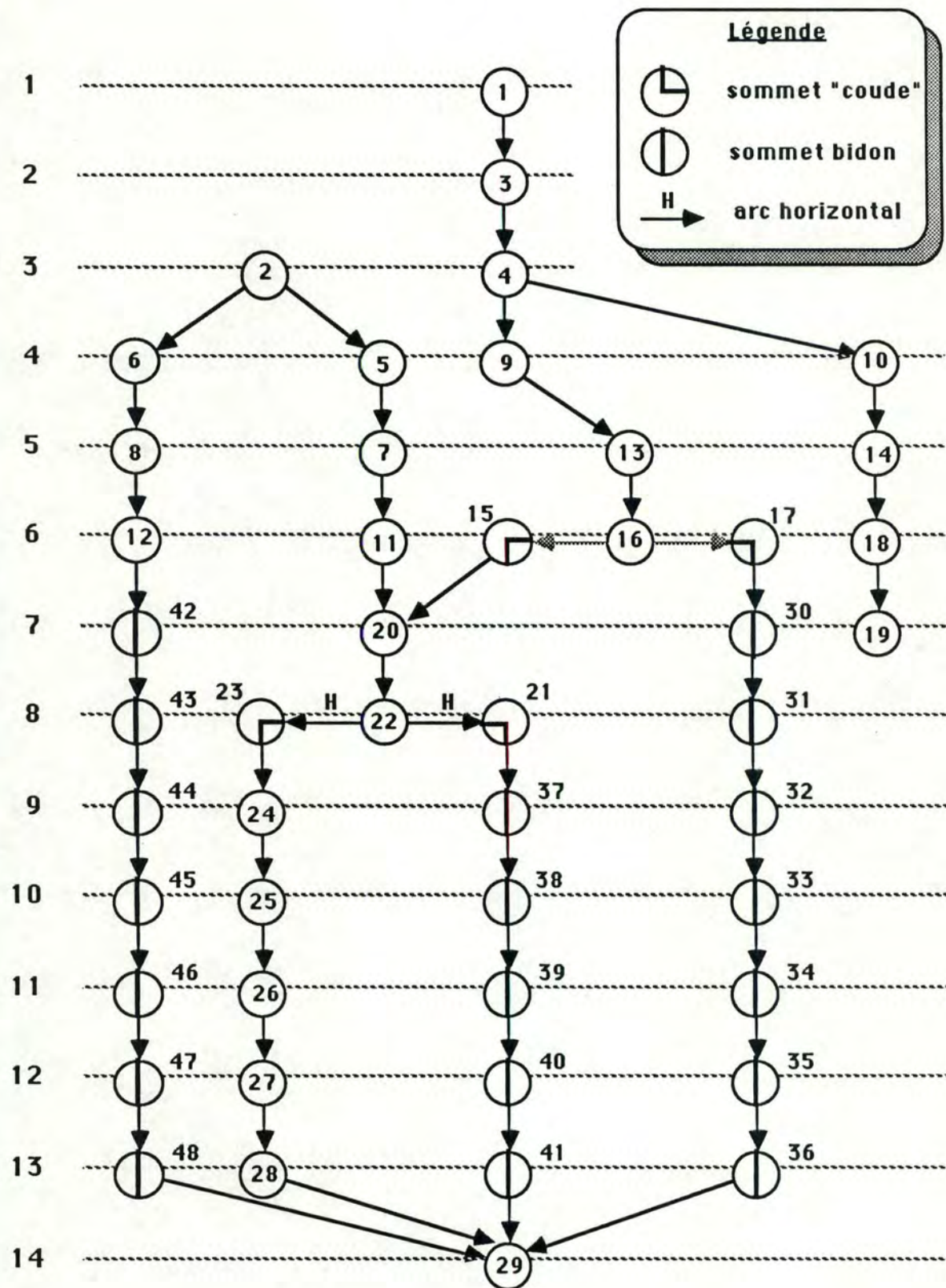
Construction de la hiérarchie serrée



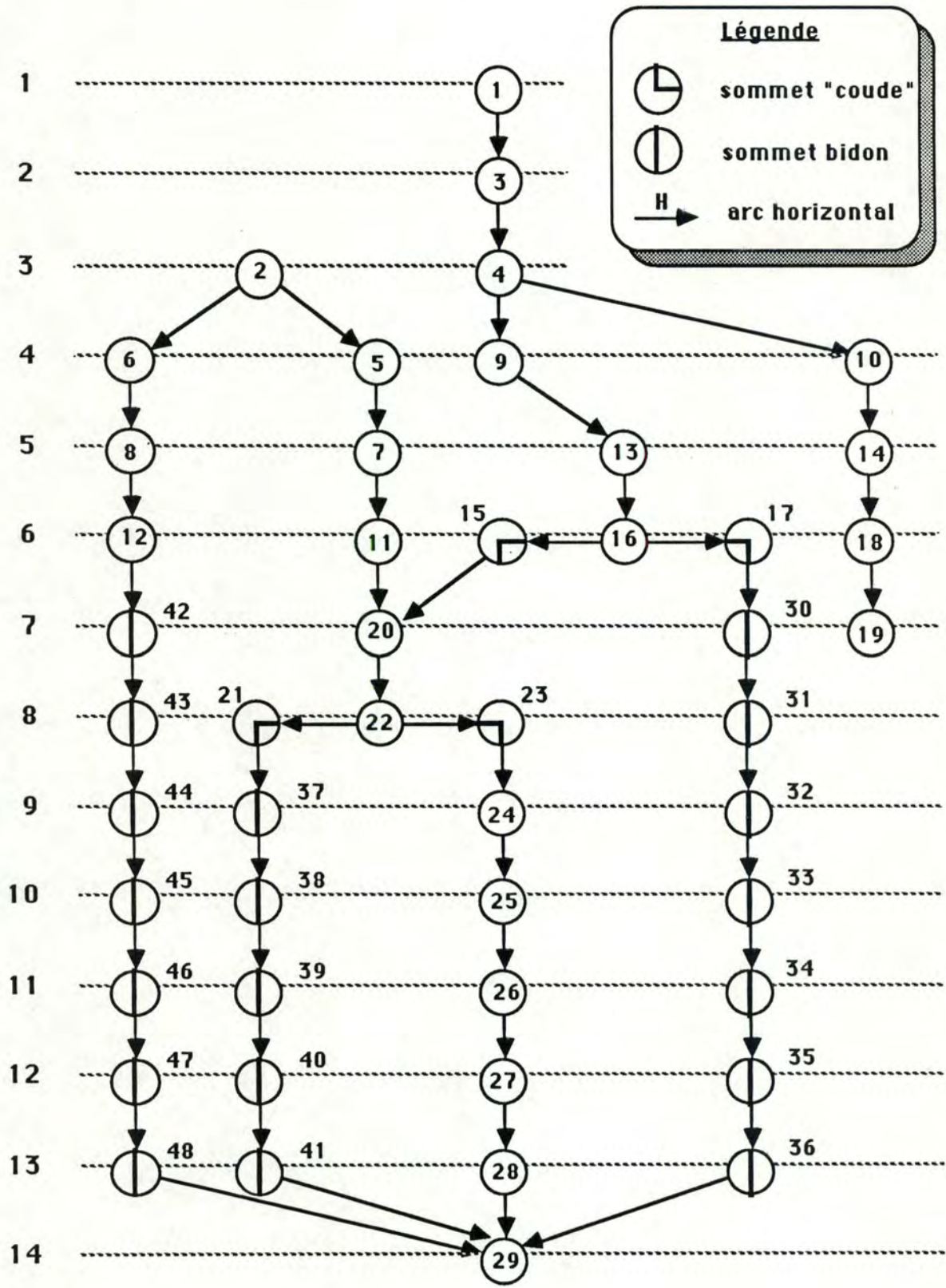
Construction de la hiérarchie propre



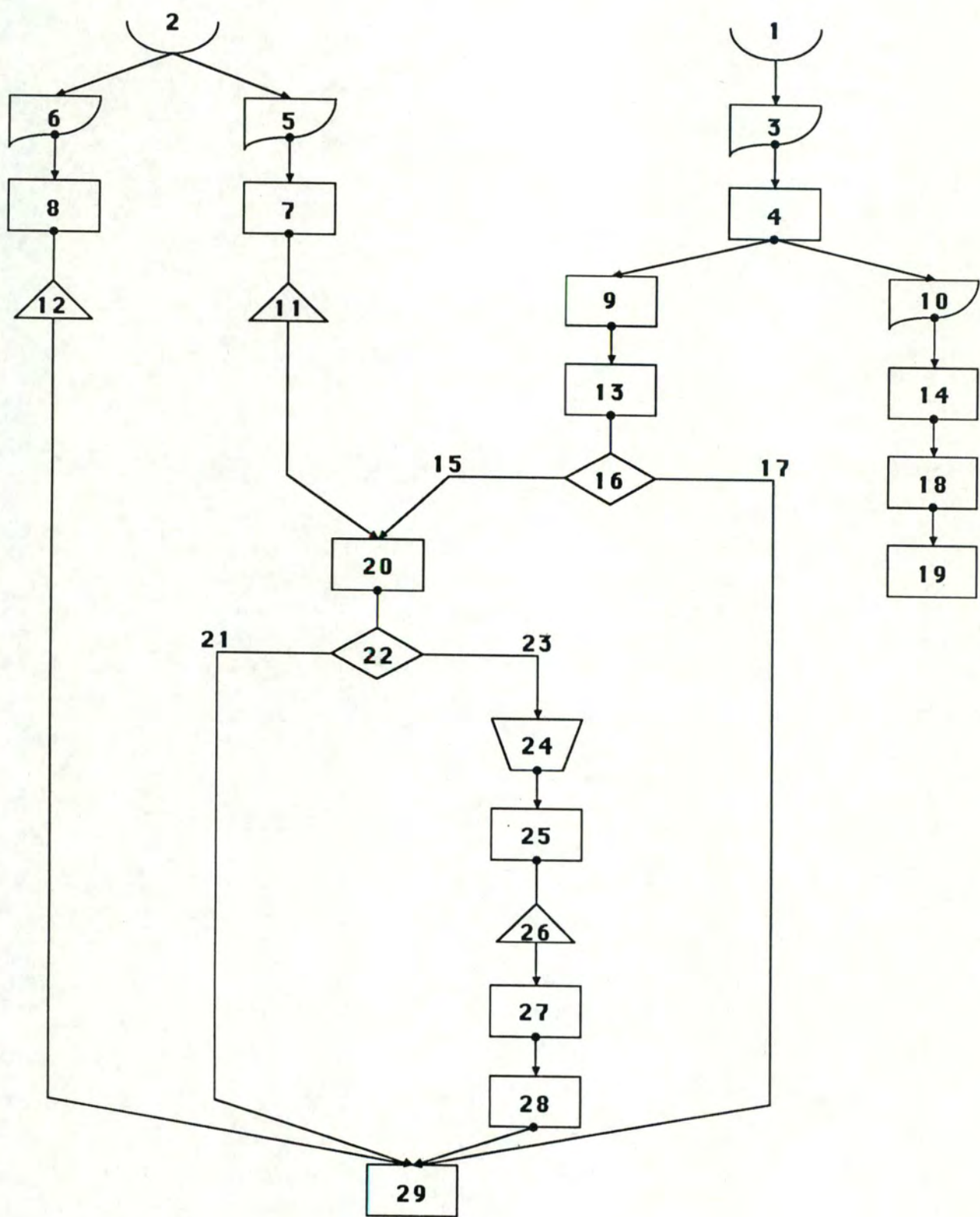
Réduction du nombre de croisements



Réduction du nombre de croisements



Calcul des coordonnées et génération du schéma.



ANNEXE B

Explications des conditions et des actions

Conditions

- 1 : la relation représente une boucle
- 2 : la relation est du type "EN ACTIVATION ...".
Suivant la valeur de cette condition, on envisage la première ou la deuxième table de
- 3 : la relation comporte une condition décision
- 4 : la condition de la relation représente une condition vraie
- 5 : la relation comporte une duplication
- 6 : le coude qui représente l'activation du processus est déjà dans le graphe
- 7 : la condition reliée à l'objet initial est déjà dans le graphe
- 8 : la branche vraie de la condition est déjà dans le graphe
- 9 : la branche fausse de la condition est déjà dans le graphe
- 10 : la duplication reliée à l'objet initial est déjà dans le graphe
- 11 : la duplication reliée à la branche vraie de la condition est déjà dans le graphe
- 12 : la duplication reliée à la branche fausse de la condition est déjà dans le graphe
- 13 : la condition reliée à l'activation du processus est déjà dans le graphe
- 14 : la branche vraie de la condition reliée à l'activation du processus est déjà dans le graphe
- 15 : la branche fausse de la condition reliée à l'activation du processus est déjà dans le graphe
- 16 : la duplication reliée à l'activation du processus est déjà dans le graphe
- 17 : la duplication reliée à la branche vraie de la condition reliée à l'activation du processus est déjà dans le graphe
- 18 : la duplication reliée à la branche fausse de la condition reliée à l'activation du processus est déjà dans le graphe

Actions

- 1 : Ajouter le coude qui représente l'activation du processus dans le graphe
- 2 : Ajouter la condition dans le graphe
- 3 : Ajouter le coude qui représente la branche vraie de la condition dans le graphe
- 4 : Ajouter le coude qui représente la branche fausse de la condition dans le graphe
- 5 : Ajouter la duplication dans le graphe
- 6 : Ajouter l'arc horizontal (processus, coude 'activation')
- 7 : Ajouter l'arc (coude 'activation', objet terminal)
- 8 : Ajouter l'arc (coude 'activation', condition)
- 9 : Ajouter l'arc (coude 'activation', duplication)
- 10 : Ajouter l'arc (objet initial, objet terminal)
- 11 : Ajouter l'arc (objet initial, condition)
- 12 : Ajouter l'arc (objet initial, duplication)
- 13 : Ajouter l'arc horizontal (condition, coude 'condition-vraie')
- 14 : Ajouter l'arc horizontal (condition, coude 'condition-fausse')
- 15 : Ajouter l'arc (coude 'condition-vraie', objet terminal)
- 16 : Ajouter l'arc (coude 'condition-fausse', objet terminal)
- 17 : Ajouter l'arc (coude 'condition-vraie', objet-terminal)
- 18 : Ajouter l'arc (coude 'condition-fausse', objet-terminal)
- 19 : Ajouter l'arc (duplication, objet-terminal)