

# THÈSE

présentée à

**L'UNIVERSITÉ PAUL SABATIER**

pour obtenir le grade de

**DOCTEUR-INGÉNIEUR**

*K ON/ 2312*

par

Hussein Ahmed KONBER

B.Sc.E.E. Université Ain Shams

M.Sc.E.E. Université du Caire

---

## CONTRIBUTION A LA CONCEPTION DES PROTOCOLES DE COMMUNICATION DANS LES AUTOCOMMUTATEURS ELECTRONIQUES

---

*Soutenue le 8 Décembre 1980, devant la Commission d'Examen :*

MM. A. COSTES

Président

M. DIAZ

M. GALINIER

J.B. JACOB

G. JUANOLE

G. VAUTRIN

} Examineurs

# THÈSE

présentée à

**L'UNIVERSITÉ PAUL SABATIER**

pour obtenir le grade de

**DOCTEUR-INGÉNIEUR**

par

Hussein Ahmed KONBER

B.Sc.E.E. Université Ain Shams

M.Sc.E.E. Université du Caire

---

## **CONTRIBUTION A LA CONCEPTION DES PROTOCOLES DE COMMUNICATION DANS LES AUTOCOMMUTATEURS ELECTRONIQUES**

---

*Soutenue le 8 Décembre 1980, devant la Commission d'Examen :*

**MM. A. COSTES**

**Président**

**M. DIAZ**

**M. GALINIER**

**J.B. JACOB**

**G. JUANOLE**

**G. VAUTRIN**

} **Examineurs**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الْحَمْدُ لِلَّهِ فَاطِرِ السَّمَوَاتِ وَالْأَرْضِ جَاعِلِ الْمَلَائِكَةِ  
رُسُلًا أُولَىٰ أَجْنِحَةٍ مَّثْنَىٰ وَثُلَاثَ وَرُبْعًا ۗ يَزِيدُ فِي الْخَلْقِ  
مَا يَشَاءُ ۗ إِنَّ اللَّهَ عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ ① مَا يَفْتَحُ اللَّهُ  
لِلنَّاسِ مِنْ رَحْمَةٍ فَلَا مُمْسِكَ لَهَا ۗ وَمَا يُمْسِكُ فَلَا مُرْسِلَ  
لَهُ مِنْ بَعْدِهِ ۗ وَهُوَ الْعَزِيزُ الْحَكِيمُ ② يَتَأْتِيهَا النَّاسُ  
أَذْكُرُوا نِعْمَتَ اللَّهِ عَلَيْكُمْ ۗ هَلْ مِنْ خَلْقٍ غَيْرِ اللَّهِ يَرْزُقُكُمْ  
مِنَ السَّمَاءِ وَالْأَرْضِ ۗ لَا إِلَهَ إِلَّا هُوَ ۗ فَأَنْ تُنْفَكُونَ ③

### Traduction

Au nom de Dieu le Très Miséricordieux,  
le Tout Miséricordieux.

- 1 Louange à Dieu, créateur des cieux et de la terre, qui désigne les anges comme des messagers dotés d'ailes par deux et par trois et par quatre. Il ajoute à la création ce qu'Il veut. Oui Dieu est capable de tout.
- 2 Ce que Dieu ouvre de miséricorde aux gens, il n'est personne qui le retienne. Et ce qu'Il retient, il n'est personne qui le relâche après Lui. Et c'est Lui le puissant. le sage.
- 3 Ho, les gens ! rappelez-vous le bienfait de Dieu sur vous : y a t-il, autre que Dieu, un créateur qui du ciel et de la terre vous nourrisse ? Point de Dieu, que Lui ! Comme vous voilà mis à l'envers !

# UNIVERSITE PAUL SABATIER

## PRESIDENCE

M MARTIN	Président
M LARENG	1er Vice Président
M MARPINARD	2ème Vice Président

## ORDRE DES SCIENCES

### HONORARIAT

M BEDOS	Professeur honoraire
M BLAIZOT	Doyen honoraire
M CAPDECOMME	Doyen honoraire, Recteur honoraire, Correspondant de l'Institut, Professeur honoraire
M COUCHET	Professeur honoraire
M DUPIN	Professeur honoraire
M DUPOUY	Membre de l'Institut, Doyen honoraire, Directeur honoraire du C.N.R.S., Professeur honoraire
M DURAND Emile	Doyen honoraire, Professeur honoraire
M FERT	Professeur honoraire
M GALLAIS	Professeur honoraire
M GAUSSEN	Professeur honoraire, Correspondant de l'Institut
M LESBRE	Professeur honoraire
M MARGULIS	Professeur honoraire
M MASDUPUY	Professeur honoraire
M MATHIS	Doyen honoraire
M MIGNONAC	Professeur honoraire
M MORQUER	Professeur honoraire, Correspondant de l'Institut
M PERRIER	Professeur honoraire
M SECONDAT	Professeur honoraire
M SERFATY	Professeur honoraire
M TEISSIE SOLIER	Professeur honoraire
M TRICHE	Professeur honoraire
M VANDEL	Professeur honoraire, Membre de l'Institut

### CORPS ENSEIGNANT

#### PROFESSEURS DE CLASSE EXCEPTIONNELLE ET DE 1ère CLASSE

M HURON	Mathématiques Appliquées
M LEDOUX	Zoologie Appliquée
M MATHIS	Chimie
M ORLIAC	Géochimie et Minéralogie
M LAFOURCADE	Physique
M ANGELIER	Zoologie
Mlle de FERRE	Botanique
M FARRAN	Minéralogie et Géotechnique
M LAUDET	Physique Théorique et Calcul Numérique
M LAGASSE	Electrotechnique
M BLANC	Physique Nucléaire
M LEREDDE	Botanique
M LELUBRE	Géologie
M LALAGUE	Mathématiques Générales
M BOUIGUE	Astronomie
M ASSELINEAU	Chimie Biologique
M MAURET	Chimie Systématique
M AGID	Physiologie
M MONTANT	Cryptogamie
M GAUTIER	Physique
M CRUMEYROLLE	Mathématiques
M GOURINARD	Géologie
M POLOU	Minéralogie
M CAMBOU	Physique Spatiale
M LACOSTE	Electrotechnique
M THIBAUT	Mécanique Rationnelle et Appliquée
M MASCART	Mathématiques
M MEDIONI	Psychophysiologie
M RAYNAUD P	Physiologie Animale
M ZALTA	Chimie Biologique
M SEVELY	Electrotechnique
M POMMIEZ	Mathématiques
M REY Paul	Biologie Végétale
M COULOMB	Physique
M TRINQUIER	Physique
M MARONI	Chimie
M BEETSCHEN	Biologie Générale
M DERACHE	Physiologie Animale
M SATGE	Chimie Organique
M LATTES	Chimie
M VEDRENNE	Géophysique
M DURAND DELGA	Géologie, Correspondant de l'Institut
M CARRARA	Physique
M MAHENC	Chimie
M MIROUSE	Géologie
M BITSCH	Zoologie

M DEGEILH	Physique
M MARTIN J C	Génie Electrique
M REY Gérard	Electronique, Electrotechnique, Automatique
M SICARD	Biologie Génétique
M SOUQUET	Géologie
M TOUZE	Physiologie Végétale
M FRASNAY	Mathématiques (Algebre et Combinatoire)
M CASSAGNAU	Zoologie
M CAUSSINUS	Mathématiques Appliquées (Statistiques Appliquées)
M PESCIA	Physique
M PICCA	Physique de l'Atmosphère
M BAUDIERE	Botanique Fondamentale et Pyrénéenne
M BARRANS	Chimie Physique Organique
M POILBLANC	Chimie Minérale
M PERENNOU	Informatique
M ATTEIA	Mathématiques
M CASTAN	Informatique
M COLLETTE	Physique
M REME	Mesures Physiques
M CUPPENS	Mathématiques

#### PROFESSEURS DE 2ème CLASSE

M MERIC	Mathématiques Appliquées
Mme LECAL	Zoologie
M PILOD	Physique
M LARROQUE	Physique
Mme LAUDET LAPEYRE	Mathématiques, Informatique
M BERTRAND	Chimie
M DESO	Mathématiques
M ROCARD	Electronique
M GUERIN	Mathématiques
M SCHNEIDER	Biologie Cellulaire
M de LOTH	Chimie Physique
M SAPORTE	Physique
M THENOZ	Génie Civil
M DURAND Ph	Physique
M FONTAN	Physique Nucléaire
M BAUDRAS	Chimie Biologique
M CALVET	Mécanique des Fluides
M PAGANI	Physique
M BEAUFILS	Informatique
M BERTHELEMY	Zoologie
M TERJANIAN	Mathématiques
M MORUCCI	Génie Biologique et Medical
M BONEL	Chimie
M SOTIROPOULOS	Chimie Organique
M VERDIER	Physique
M ETTINGER	Mathématiques
M BONNET Louis	Biologie
M JOSSE RAND	Mesures Physiques
M ROUTIE	Génie Chimique
M COTTU	Génie Mécanique
M HURAU X	Physique
Mme GERVAIS	Chimie Inorganique
M BANCEL	Mathématiques
M LOUARN	Génétique
M BOUDET	Physiologie Vegetale
M LETAC	Mathématiques
M HERAULT	Chimie
M GRANDET	Génie Civil
Mlle BARBANCE	Mathématiques
M GILLY	Génie Mécanique
M MARAL	Physique
M LEGRAND	Génie Civil
M ABAT	Electronique, Electrotechnique, Automatique
M MAUSS	Mécanique
M BETOURNE	Informatique
M CAMPAN	Psychophysiologie
M CLERC	Mécanique
M GRIFONE	Mathématiques
M COUOT	Mathématiques, Analyse Numérique
M NGUYEN THANH VAN	Mathématiques
M TRAVERSE	Problemes Chimiques de l'Energie
M ALRAN	Génie Chimique
M REY J	Géologie Sédimentaire et Paléontologie
M DARTIGUENAVE	Chimie Minérale Moléculaire
M PRADINES	Mathématiques
M GALINIER	Informatique
M VIGNOLLE	Informatique
M DEPARIS	Embryologie
M CAVALIE	Physiologie Vegetale

M MASSOL	Chimie des Composés Organiques et Organominéraux d'intérêt biologique
M HARTMANN	Mécanique
M ROUSSET	Chimie Appliquée (Matériaux)
M TARDY	Sciences de la Terre et Aménagement
M HOLLANDE	Biologie Cellulaire
M DUGAS	Physique des Energies Nouvelles
M BENOIT CATTIN	Physique
M COMTAT	Chimie Appliquée
M LANEELLE	Biochimie
M LUGUET	Informatique Fondamentale et Appliquée
M BONNET J J	Chimie Minérale
M PERAMI	Minéralogie et Matériaux
M AUDOUNET	Mathématiques
M PERIE	Chimie Organique

#### PROFESSEURS ASSOCIES

M GUMOWSKI	Mathématiques
M COLLINS	Génie Mécanique

#### CHERCHEURS DU C.N.R.S.

#### DIRECTEURS DE RECHERCHE

M GIRALT Georges
M LABARRE Jean
M LAURENT Jean Pierre
M LEGRIS
M MARTINOT Henri
M MAZEROLLES
M PRADAL
M WOLF Robert
M ESTEVE Daniel
M GALY Jean

#### MAITRES DE RECHERCHE

M AGUILAR MARTIN José
Mme ASSELINEAU Cécile
M AZEMA Pierre
M BUXO Jean
Mme DARTIGUENAVE M.
M HAWKES Peter
M HOUALLA Douard
M JEREBZOFF
M MALRIEU J.P.
Mme MARONI Yvette
Mme MATHIS
M MUNOZ Aurélio
M NAVECH
M PRAJOUX Roland
M SEVELY Jean
M VACQUIE Serge
Mme BENAZETH Nicole
Mme DUPRAT Anne-Marie

#### CORPS DES OBSERVATOIRES ASTRONOMIQUES ET INSTITUTS DE PHYSIQUE DU GLOBE

Mme ANDRILLAT Y	Astronome titulaire
M COUPINOT G	Astronome adjoint
M LEROY J Louis	Astronome adjoint
M MIANES	Astronome adjoint
M PEDOUSSAUT A	Astronome adjoint
M ROBLEY Robert	Physicien titulaire
M ROSCH Jean	Astronome titulaire
M SAISSAC Joseph	Physicien titulaire

#### ADMINISTRATION

M PRINEAU	Secrétaire Général de l'Université
-----------	------------------------------------

### INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

#### PRESIDENCE

M MONTEL	Président
M MARTY	Vice-Président
M ENJALBERT	Vice-Président
M ANDRE	Vice-Président
M CONSTANT	Vice-Président

#### HONORARIAT

M BIREBENT	Professeur honoraire
M CASTAGNETTO	Professeur honoraire
M DIEHL	Professeur honoraire
M ESCANDE	Professeur honoraire, Membre de l'Institut
M HAMANT	Professeur honoraire

#### PROFESSEURS DE CLASSE EXCEPTIONNELLE ET DE 1ère CLASSE

M. NOUGARO	Hydraulique Générale et Appliquée
M. GARDY	Génie Chimique
M. VOIGT	Chimie Minérale
M. MONTEL	Chimie Inorganique
M. THIRRIOT	Hydraulique
M. GRUAT	Hydraulique
M. BUGAREL	Génie Chimique
M. DAT	Hydraulique
M. MARTY	Electronique
M. ANGELINO	Génie Chimique
M. HOFFMANN	Electronique
M. TRANNOY	Electrotechnique

#### PROFESSEURS DE 2ème CLASSE

Mlle BERDUCOU	Physiologie Végétale
M FALLOT	Biotechnologie Végétale Appliquée
M ENJALBERT	Génie Chimique
M. TRUCHASSON	Hydraulique
M DABOSI	Métallurgie et Réfractaires
M LEFEUVRE	Electronique
M CALMON	Chimie Agricole
M GILOT	Génie Chimique
M MATHIEU	Chimie Appliquée
M BAUDRAND	Electronique
M BOURGEAT	Pédologie
M. MATHEAU	Electronique
M BAJON	Electronique
M COUDERC	Génie Chimique
M GOURDENNE	Chimie
M LENZI	Chimie Industrielle
M MASBERNAT	Hydraulique
M TERRON	Zoologie
M. BUIS	Biologie Quantitative
M CONSTANT	Chimie Minérale
M COSTES	Electrotechnique
M ECOCHARD	Agronomie
M CANDAU	Zootecnie
M LABAT	Ichtyologie Appliquée
M MORELIERE	Electronique
M GASET	Chimie Industrielle
M BRUEL	Informatique
M ALBERTINI	Cytologie et Pathologie Végétales
M BELLET	Mécanique - Hydraulique
M FABRE	Mécanique - Hydraulique
M FOCH	Electronique, Electrotechnique, Automatique
M MORA	Génie Chimique
M MORARD	Physiologie Végétale Appliquée
M MURATET	Génie Chimique
M NOAILLES	Mathématiques

#### ADMINISTRATION

M CRAMPES	Secrétaire Général
-----------	--------------------

### INSTITUT DES SCIENCES APPLIQUEES DE TOULOUSE

M SARAZIN	Directeur Général
-----------	-------------------

#### PROFESSEURS DE CLASSE EXCEPTIONNELLE ET DE 1ère CLASSE

M ROQUES	Chimie Appliquée, Génie Chimique
M GRATELOUP	Electronique, Electrotechnique, Automatique
M MASO	Génie Civil
M MIRA	Electronique, Electrotechnique, Automatique
M DURAND Gilbert	Biologie et Biochimie Appliquée
M ASKENAZY	Physique Atomique, Physique du Solide
M LETURCO	Electronique, Electrotechnique, Automatique
M SCHUTTLER	Electronique, Electrotechnique, Automatique

#### PROFESSEURS DE 2ème CLASSE

M FAGET	Physique Atomique, Physique du Solide
M BARTHET	Mécanique
M TITLI	Electronique, Electrotechnique, Automatique
M BROUSSEAU	Physique Atomique, Physique du Solide
M SIRIEYS	Mécanique
M BESOMBES VAILHE	Chimie Appliquée, Génie Chimique
M VERDIER	Chimie Organique, Minérale et Analytique
M BOUDET René	Génie Mécanique
M GROH	Physique Atomique, Physique du Solide
M JAVELAS	Génie Civil
M LACAZE	Mathématiques II
M LORRAIN	Génie Civil
M MARTINEZ	Electronique, Electrotechnique, Automatique

#### ADMINISTRATION

M CROS	Secrétaire Général
--------	--------------------

*A L'EGYPTE*

*A LA MEMOIRE DE MON PERE*

*A MA MERE*

*A MA FEMME SANAA*

*A TOUS LES MIENS*

*AVEC TOUTE MON AFFECTION*



## R E M E R C I E M E N T S

---

*Le travail présenté dans cette thèse a été effectué au Laboratoire d'Automatique et d'Analyse des Systèmes du C.N.R.S., dirigé par Monsieur le Professeur G.GRATELOUP, dans l'équipe Logiciel et Communication (LC), dirigé par Monsieur M.DIAZ, Chargé de recherche au C.N.R.S.*

*Je tiens à remercier pour leur participation à mon jury de thèse :*

- Monsieur A.COSTES, Professeur à l'Institut National Polytechnique de Toulouse,*
- Monsieur M.DIAZ, Chargé de recherche au C.N.R.S.,*
- Monsieur M.GALINIER, Professeur à l'Université Paul Sabatier de Toulouse,*
- Monsieur J.B.JACOB, Ingénieur en Chef à la Compagnie Industrielle de Télécommunication CIT-ALCATEL,*
- Monsieur G.JUANOLE, Maître Assistant à l'INSA de Toulouse,*
- Monsieur G.VAUTRIN, Chef du Département SCR au CNET de Lannion.*

*J'exprime mes sincères remerciements au Chef de l'Equipe Logiciel et Communication, Monsieur M.DIAZ, qui a dirigé ce travail, pour ses encouragements et sa confiance.*

*Mes remerciements vont également à tous les membres de l'équipe LC, et en particulier J.M.AYACHE, Mme B.CHEZALVIEL-PRADIN, J.P.COURTIAT, G.JUANOLE et Melle NOUBEL pour leur aide et leur soutien amical ; à R.VALETTE de l'équipe PASTELS pour les discussions profitables que j'ai eues avec lui ; ainsi qu'à S.MEGAHEDE de l'équipe LSR CAR pour son aide.*

*Je tiens à remercier MM A.COSTES, M.DIAZ et G.JUANOLE pour leurs remarques constructives qui m'ont permis d'améliorer la présentation et la qualité de ce mémoire.*

*Mes remerciements vont également aux Ingénieurs de la CIT-ALCATEL pour leurs apports nombreux et profitables et en particulier à Monsieur J.B.JACOB pour, de plus, la confiance qu'il m'a accordée.*

*J'exprime enfin toute ma reconnaissance au service SIS, à Madame J.PENAVAYRE, Secrétaire de la Division Structures des Systèmes de Commande Automatique, à Mademoiselle M.LABERGUE qui a assuré la frappe ainsi qu'à MM. D.DAURAT, R.LORTAL et R.ZITTEL qui ont assuré la réalisation matérielle de ce mémoire.*

*\$\$\$*

# INTRODUCTION





La complexité croissante des systèmes de commande en temps réel rend de plus en plus indispensable l'utilisation d'une méthodologie de conception parfaitement définie et s'appuyant, ce qui est indispensable, sur un ensemble d'outils adaptés.

Parmi les étapes nécessaires permettant d'aboutir à la réalisation du système, nous considérons ici la première d'entre elles, à savoir la phase de spécification qui, à partir de l'expression des besoins, conduit à l'établissement des spécifications formelles du système à réaliser. Cette tâche est particulièrement cruciale pour la conception du système car le concepteur doit être exhaustif sur la description des objets qu'il devra manipuler et des liens qui existent entre eux et doit aussi donner une représentation globale du système avant le début de l'implémentation. La modélisation correspondante est d'autant plus difficile à réaliser que le système est complexe et que son fonctionnement est lié à celui d'autres ensembles auxquels il est connecté. Ces deux points sont particulièrement vrais dans le cas des autocommutateurs téléphoniques, compte-tenu du nombre des abonnés gérés et des interactions existant entre ces abonnés.

Dans le cadre de ce mémoire, nous envisageons la spécification fonctionnelle d'un autocommutateur en commençant par représenter la communication vue d'un abonné.

Notre démarche a été telle que deux types de spécification, une spécification informelle et une spécification formelle, forment un couple complémentaire constituant le résultat final de la modélisation. Cette approche a été appliquée d'abord pour décrire le comportement d'un abonné isolé et ensuite pour spécifier des protocoles de communication entre les abonnés communicants.

La partie formelle du couple informalité-formalité repose sur l'utilisation des réseaux de Pétri ; les raisons de ce choix parmi les modèles que nous avons envisagés sont données dans le chapitre I.

Le chapitre II présente la méthode de spécification et ses trois phases principales. Dans une première phase, dite de définition, le texte informel est écrit en langage naturel et les composants ou objets qui interviennent sont recensés et classés.

Dans une seconde phase, dite de formalisation, les relations données en langage informel sont réécrites pour permettre une formalisation au moyen des réseaux de Pétri labellés. Les réseaux obtenus sont alors analysés par un outil interactif qui a été développé dans l'équipe, ce qui constitue la troisième phase dite de vérification. L'exemple support de cette spécification est la représentation de la communication vue d'un abonné unique.

Dans le chapitre III, la spécification de l'abonné est développée à un plus grand niveau de détails. En particulier, cette modélisation fait apparaître les liens existant entre les abonnés et conduit donc à la spécification par réseaux de Pétri des protocoles d'interaction entre les abonnés. Après avoir discuté certains problèmes de représentation, nous analysons les recommandations du CCITT qui se rapportent à ces interactions. Nous en déduisons alors des modélisations possibles qui résultent des recommandations N°6 et N°7.

Nous montrons en particulier comment modéliser un protocole de communication, comment introduire ou supprimer des messages et comment les possibilités d'analyse par un outil graphique interactif simplifient la définition et l'étude des choix des messages et des interactions lors de la proposition des différents protocoles.

Enfin, dans le chapitre IV, nous abordons les problèmes liés au passage de la représentation à l'implémentation. Il s'agit, à partir du réseau de Pétri validé et correct modélisant le protocole, d'en déduire, par une méthode systématique, un programme écrit dans un langage de programmation. Cette méthode, qui repose sur l'utilisation du langage PLM et d'un noyau de moniteur multitâches (ayant comme primitives de synchronisation ENVOI et RECEPTION de messages), a conduit à une réalisation sur un outil de développement de microcalculateurs.

# CHAPITRE I

-----

## QUELQUES METHODOLOGIES SIGNIFICATIVES DE SPECIFICATION POUR LES SYSTEMES TEMPS REEL

-----



## I.1 INTRODUCTION

Un autocommutateur est un système complexe tant au niveau matériel qu'au niveau logiciel. D'une façon générale, la conception de tels systèmes nécessite l'utilisation d'une méthodologie globale d'approche comportant plusieurs phases successives. La première est la phase de définition des besoins ou d'élaboration du cahier des charges. Très souvent, elle consiste en une description en langage naturel du problème à traiter. Il est ensuite nécessaire de passer à la phase de spécification, qui formalisera ces besoins et lèvera les ambiguïtés pouvant exister dans le cahier des charges ; à partir de la fin de cette phase, il sera possible de démarrer les phases d'implémentation et de test.

Dans la suite de ce chapitre, nous exposons en premier lieu les approches nous paraissant actuellement les plus significatives pour spécifier les systèmes fonctionnant en temps réel.

A partir des 3 possibilités retenues, diagramme d'états, réseau de Pétri, SADT [16], et afin de faire un choix dans notre étude, nous exposons alors les principaux critères qui, à notre avis, doivent être satisfaits par la représentation.

Enfin, nous motiverons, dans le contexte d'entités communicantes, le choix et l'intérêt de critères liés à la vérification et à l'analyse du modèle.

## II.2 CRITERES DE MESURE ET EVALUATION D'UN MODELE [1,2]

Tout modèle de spécification d'une mission doit satisfaire à un certain nombre de critères correspondant à deux points principaux : sa création et son utilisation ; nous donnons ci-après ces critères en les commentant :

a) La précision

La notation utilisée pour spécifier doit être non ambiguë et les règles de fonctionnement parfaitement précisées. Ce critère est indispensable si l'on désire analyser le modèle et vérifier l'adéquation entre les propriétés du modèle inhérentes au mode de représentation et les propriétés que le concepteur estime devoir conférer à la mission, par exemple l'absence de blocage.

b) La lisibilité

Ce critère très important permet aux spécifications d'être compréhensibles dans l'espace (entre le concepteur et les utilisateurs) et dans le temps (entre le moment de la conception et celui de l'utilisation).

c) L'extensibilité, la modifiabilité

On doit pouvoir facilement augmenter ou diminuer le schéma de mission, supprimer ou rajouter de nouveaux modules, appliquer la méthodologie de construction du modèle à différents niveaux fonctionnels etc...

d) La cohérence

La méthodologie de construction ne doit pas engendrer d'incohérence et nécessite donc l'établissement d'une méthode extrêmement rigoureuse.

e) La constructibilité

Elle caractérise, pour le concepteur, la possibilité et la facilité de passer de sa vision de la mission à la représentation au moyen du modèle.

REMARQUE

Un certain antagonisme existe entre ces critères. Ainsi la précision peut s'opposer à la lisibilité : une mission décrite dans une notation précise permettant l'analyse (par exemple, la notation mathématique) peut s'avérer très difficilement compréhensible pour un utilisateur peu familier avec une telle notation.

### Principe de spécification

Les spécifications peuvent avoir divers niveaux de formalité. Leur expression peut aller de la langue naturelle à la notation mathématique en passant par les diagrammes, les langages abstraits, etc...

Il y a de nombreux avantages à utiliser des spécifications formelles plutôt qu'informelles. Les spécifications formelles peuvent être étudiées mathématiquement et surtout automatiquement. Des spécifications incohérentes ou incomplètes peuvent être détectées automatiquement. De plus, l'interprétation de spécifications formelles est unique (ou devrait l'être) par l'utilisation d'une notation non ambiguë. Celle-ci devrait pousser le concepteur à plus de rigueur et donc à dévoiler des aspects qui resteraient peu clairs dans une spécification informelle. Malheureusement, ces avantages peuvent se retourner contre l'utilisateur : l'ensemble des relations, actions et interactions sur les objets devant être complet, l'attention demandée à l'utilisateur est considérable. Les erreurs sont, par là même, favorisées.

Les spécifications informelles sont, par contre, plus faciles à établir. L'utilisation d'une langue naturelle supprime l'exercice d'une notation formelle et tout l'effort est reporté sur le problème. Un avantage très important des spécifications informelles est alors la facilité de compréhension, même par des non spécialistes du problème qui peuvent ainsi participer plus activement à la spécification.

Mais, compte tenu de leurs différences et avantages respectifs, plutôt que d'opposer spécifications formelles et informelles, il semble préférable de les associer et c'est ce que nous utiliserons ici. Le produit final de notre spécification comportera [1], donc, un modèle formel commenté par un modèle informel.

Le modèle informel sera le résultat de la phase de définition ; le modèle formel sera le résultat d'une série de transformations du modèle informel. Il n'est jamais bien simple de parcourir le spectre du modèle informel au modèle formel. C'est pourquoi la méthodologie proposée visera

surtout à dégager les imprécisions et à déterminer la nature des objets utilisés dans la mission plutôt que de définir très strictement l'évolution du modèle.

## I.3 OUTILS DE MODELISATION - LES PRINCIPALES APPROCHES [1,2]

Nous discutons ici les principales approches et techniques de modélisation.

### I.3.1 LES APPROCHES

Nous distinguerons trois grandes familles que nous classerons comme étant les approches linguistique , mathématique et graphique .

#### I.3.1.1 L'approche linguistique

Cette approche est utilisée depuis longtemps et est la plus connue. Les langages naturels, les langages de programmation et aussi certains langages spécifiques sont utilisés pour décrire les fonctions d'un système donné.

La spécification par un langage naturel est informelle. L'ambiguïté inhérente et les redondances pouvant exister dans la spécification sont sources de problèmes lors de la réalisation. En revanche, c'est le moyen le plus répandu pour écrire la spécification lors de la première phase de définition des besoins (cahier des charges), ceci étant dû à sa facilité d'expression et d'utilisation. D'un autre côté, les langages de haut niveau sont utilisés pour spécifier certains systèmes : ils sont plus adaptables à la représentation des nombres, des données, etc..., mais en revanche sont moins convenables pour représenter les systèmes complexes interactifs notamment l'aspect contrôle et les aspects temporels (initialisation, synchronisation, etc...).

Notons qu'il existe une forte tendance proposant de combiner les langages évolués avec les modèles graphiques [3,4] pour spécifier les systèmes complexes.

### I.3.1.2 L'approche mathématique

L'approche mathématique paraît être la plus efficace pour permettre l'analyse des spécifications, ceci résultant de la rigueur de l'expression mathématique.

De plus, elle impose l'utilisation de nombreuses définitions très précises qui peuvent rendre leur acquisition et leur utilisation difficiles. Malgré que l'approche mathématique paraisse privilégiée en ce qui concerne la vérification des modèles, elle entraîne à l'heure actuelle une grande difficulté de transmissibilité entre les concepteurs et les utilisateurs des spécifications.

### I.3.1.3 L'approche graphique

Cette approche représente la transition entre les deux approches précédentes dans la mesure où elle peut compléter la première approche et où elle est proche par sa nature de la seconde et permet avec certaines précautions l'analyse.

Pour des raisons qui apparaîtront plus tard, c'est cette approche que nous avons choisie. Nous allons maintenant présenter les principaux modèles existant actuellement dans cette classe.

## I.3.2 LES PRINCIPAUX MODELES GRAPHIQUES

### I.3.2.1 Diagrammes d'états [4,5,6]

Le fonctionnement du système est divisé en un nombre fini d'états stables. Chaque état stable porte un nombre et un nom propres à cet état ; les états sont connectés entre eux par des arcs qui correspondent aux transitions.

Un diagramme d'états est un 5tuplet  $\langle X, I, O, M, N \rangle$  où :

$X$  : est un ensemble fini des états.

$I$  : est un ensemble fini des entrées ( $i_1, i_2, \dots, i_k$ ).

$O$  : est un ensemble fini des sorties ( $O_1, O_2, \dots, O_l$ ).

$M$  : est la fonction état suivant ( $M : I \times X \longrightarrow X$ ).

$N$  : est la fonction de sortie (actions), ( $N : X \times I \longrightarrow O$ ).

L'ensemble des entrées est composé des évènements qui provoquent la transition d'un état à un autre. Ces évènements peuvent être divers ; par exemple, des commandes (internes ou externes), des réceptions de messages ou des fins de délais.

L'ensemble des sorties comprend les mêmes types d'évènements.

### I.3.2.2 Réseaux de Pétri [11,13,14,15]

Les réseaux de Pétri qui sont une généralisation des diagrammes d'états sont très utilisés pour modéliser les interactions entre des processus ou des activités parallèles ; il est de ce fait possible de spécifier par des réseaux de Pétri des protocoles de communication [7,8,9,10,11,12].

Le réseau de Pétri est constitué de deux types de sommets, les places et les transitions ; la place est représentée par un cercle dans lequel se trouve un nombre de jetons ( $0 \rightarrow N$ ) qui représente un état (partiel) du système ; la transition est dessinée par une barre et précise l'évènement provoquant le changement d'état, c'est-à-dire le tir de la transition. Les places et les transitions sont connectées entre elles par des flèches ou arcs qui représentent les liens existant dans la mission.

#### I.3.2.2.1 Définitions

##### Définition 1 : Le Réseau de Pétri

Un réseau de Pétri est un 4tuple  $\langle P, T, \alpha, \beta \rangle$  tel que :

$P$  : est un ensemble d'objets appelés places ( $P \neq \emptyset$ ) représentés par des cercles.

$T$  : est un ensemble d'objets appelés transitions ( $T \neq \emptyset, P \cap T = \emptyset$ ) représentés par des barres.

$\alpha : P \times T \longrightarrow N$ , est la fonction d'incidence avant.

$\beta : P \times T \longrightarrow N$ , est la fonction d'incidence arrière.

Il existe un arc connectant la place  $p_k$  à la transition  $t_i$  si et seulement si  $\alpha(p_k, t_i) = n_{ki} \neq 0$  ; cet arc est étiqueté par  $n_{ki}$  qui s'appelle le poids de l'arc. De la même façon, un arc allant de la transition  $t_j$  à la place  $p_l$  existe si et seulement si  $\beta(p_l, t_j) = n_{lj} \neq 0$  ; cet arc est étiqueté par  $n_{lj}$ . Si l'arc n'est pas étiqueté, la valeur par défaut du poids est égale à un.

Définition 2 : Les Vecteurs Pre(t) et Post (t)

Dans un réseau de Pétri on définit deux vecteurs Pre(t), Post(t).

Pour une transition t dans T, il existe un vecteur Pre (t) dans lequel se trouvent les places qui précèdent t.

Pre (t) :  $\langle p_a, p_b, \dots \rangle$

Pour une transition dans T, il existe un vecteur Post (t) dans lequel se trouve les places qui suivent t.

Post (t) :  $\langle p_l, p_m, \dots \rangle$

Pre (t) : est l'ensemble de places d'entrée d'une transition t.

Post (t) : est l'ensemble de places de sortie d'une transition t.

Définition 3 : Marquage d'un Réseau

On appelle marquage une fonction M de l'ensemble P sur l'ensemble des entiers naturels N : à chaque place p est associé un nombre n qui symbolise la présence de n jetons dans la place p.

Ce marquage M peut être représenté de 2 façons :

- ♦ par la liste des nombres de jetons contenus dans chaque place,
- ♦ par l'indication du nom des places marquées et la répétition de celles qui sont marquées plusieurs fois.

Le marquage initial d'un réseau, généralement noté  $M_0$ , est la distribution de jetons représentant l'état initial du système.

Définition 4 : Activation d'une Transition

Une transition est activée pour un marquage M, si et seulement si

$$\forall p \in \text{Pre}(t), M(p) \geq \alpha(p,t).$$

Par exemple, dans la figure I1, on trouve :

$$\text{Pre}(t_1) = (p_1, p_2, p_3) ;$$

$$M(p_1) = 4, M(p_2) = 2, M(p_3) = 1 ;$$

$$\alpha(p_1, t_1) = 3, \alpha(p_2, t_1) = 2, \alpha(p_3, t_1) = 1$$

Donc  $\forall p \in \text{Pre}(t_1), M(p) \geq \alpha(p, t_1)$  ; en conséquence la transition  $t_1$  est activée.

Définition 5 : Tir d'une transition

Une transition est tirée si et seulement si elle est activée.

Le tir d'une transition entraîne l'évolution du marquage.

L'évolution du marquage est telle que :

$$\forall p \in \text{Pre}(t), M'(p) = M(p) - \alpha(p,t) \dots\dots\dots (1)$$

$$\forall p \in \text{Post}(t), M'(p) = M(p) + \beta(p,t) \dots\dots\dots (2)$$

Par exemple dans la figure I.1, on obtient :

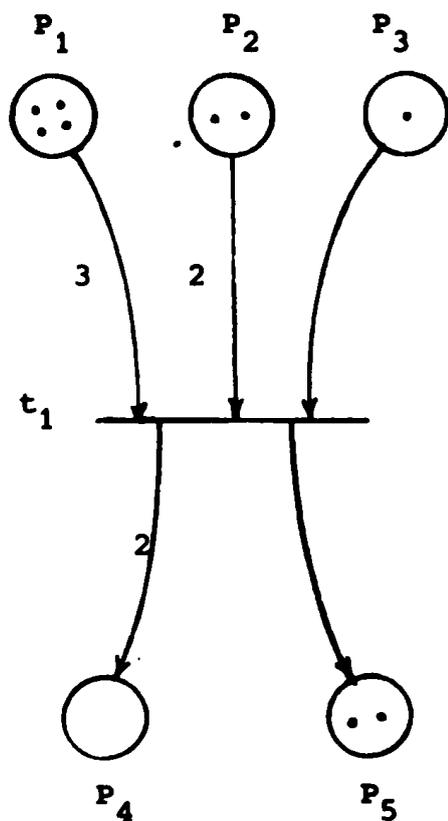
$$\text{Pre}(t_1) = (p_1, p_2, p_3), \text{ Donc :}$$

$$M'(p_1) = M(p_1) - \alpha(p_1, t_1) = 4 - 3 = 1$$

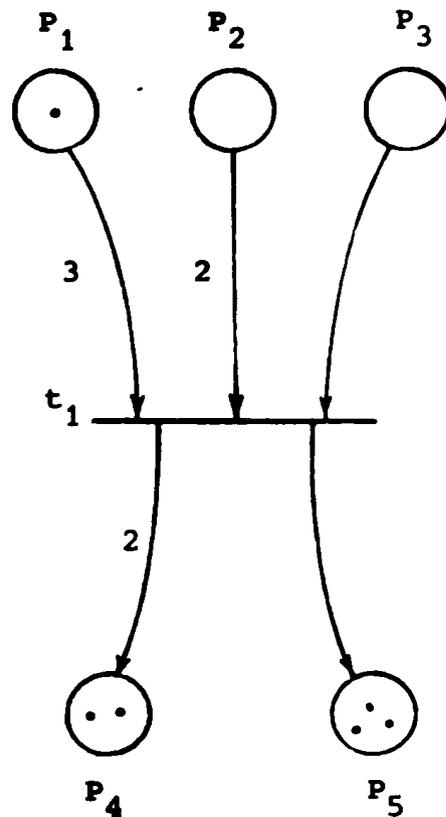
$$M'(p_2) = M(p_2) - \alpha(p_2, t_1) = 2 - 2 = 0$$

$$M'(p_3) = M(p_3) - \alpha(p_3, t_1) = 1 - 1 = 0$$

Exemple



(a) Avant le tir de  $t_1$



(b) Après le tir de  $t_1$

$$P = \{P_1, P_2, P_3, P_4, P_5\}$$

$$T = \{t_1\}$$

$$\alpha(P, t_1) = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\beta(P, t_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix}$$

$$M = (4, 2, 1, 0, 2) \quad , \quad \text{Avant le tir de } t_1.$$

$$M' = (1, 0, 0, 2, 3) \quad , \quad \text{Après le tir de } t_1.$$

Fig. I.1. Tir d'une Transition dans un Réseau de Pétri

$\text{Post}(t_1) = (p_4, p_5)$ , Donc :

$$M'(p_4) = M(p_4) + f(p_4, t_1) = 0 + 2 = 2$$

$$M'(p_5) = M(p_5) + \beta(p_5, t_1) = 2 + 1 = 3$$

En général l'évolution du marquage peut être formalisée par :

$$\forall p \in P, M'(p) = M(p) - \alpha(p, t) + \beta(p, t)$$

#### Définition 6 : Réseau de Pétri Pur

Un réseau de Pétri est pur (libre de boucle élémentaire) si et seulement si  $\forall t \in T, \text{Pre}(t) \cap \text{Post}(t) = \emptyset$ ,

c'est-à-dire si une place n'est pas à la fois place d'entrée et de sortie pour une même transition.

Un réseau de Pétri pur peut être décrit par une seule matrice C qui s'appellera la matrice d'incidence avec  $C = \beta - \alpha$ .

Dans cette situation l'évolution de marquage s'écrit :

$$M' = M + C.X \quad \text{où :}$$

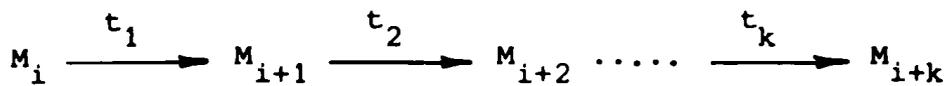
C : est la matrice d'incidence du réseau.

X : est un vecteur de n composantes (n : nombre de transitions) dans lequel toutes les composantes sont égales à zéro sauf celle qui correspond à la transition tirée et qui a la valeur 1.

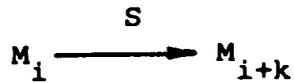
Cette forme s'avère très utile pour manipuler les réseaux et les analyser.

#### Définition 7 : Séquence de Tir de Transitions

Soit S une séquence finie de transitions  $t_1, t_2, \dots, t_k$  de l'ensemble T ; on dit que S est une séquence de tir tirable à partir de  $M_i$  si et seulement si il existe des marquages  $M_{i+1}, M_{i+2}, \dots, M_{i+k}$  du réseau de Pétri tel que :



A partir de  $M_i$  on obtient  $M_{i+k}$  par le tir de S



Définition 8 : Classe des Marquages Conséquents

La classe des marquages conséquents  $\overrightarrow{M_0}$  est l'ensemble des marquages accessibles à partir du marquage  $M_0$  par une séquence de tir S.

$$M_i \in \overrightarrow{M_0} \iff \exists S \text{ telle que } M_0 \xrightarrow{S} M_i$$

$\overrightarrow{M_0}$  indiquera en général un ensemble d'états du contrôle possibles pour un système.

Définition 9 : Graphe des Marquages Conséquents de  $M_0$

Le graphe des marquages conséquents de  $M_0$ , noté  $G(\overrightarrow{M_0})$ , est le graphe orienté où l'ensemble des sommets est  $\overrightarrow{M_0}$  et l'ensemble A des arcs  $(M_i, M_j)$  est défini par :

$$M_i \in \overrightarrow{M_0}, M_j \in \overrightarrow{M_0}, \exists t \in T \text{ tel que :}$$

$$M_i \xrightarrow{t} M_j \iff (M_i, M_j) \in A$$

chaque arc est étiqueté par la transition tirée t.

I.3.2.2.2 Propriétés du Réseau de Pétri

Nous rappelons dans ce paragraphe quelques propriétés des réseaux de Pétri qui peuvent être utilisées pour la validation des systèmes parallèles. L'analyse d'un réseau de Pétri peut montrer, au niveau du modèle, un défaut de conception. Le cahier des charges devra alors être modifié et corrigé à ce niveau là, avant que l'erreur n'ait d'autres conséquences au niveau du système.

### 1. Réseau de Pétri borné

Un réseau de Pétri est borné par  $n$  pour un marquage initial  $M_0$  si et seulement si :  $\forall M \in \overline{M_0}, \forall p \in P, M(p) \leq n$ .

L'implémentation d'un réseau non borné n'est pas nécessairement sûre car il existe des séquences fonctionnelles qui peuvent provoquer un débordement de la capacité choisie lors de l'implémentation.

Si  $n = 1$  on dit que le réseau est sauf ; si cette propriété doit être satisfaite alors une séquence amenant deux jetons ou plus dans une place montre qu'il existe un comportement possible non correct du système.

Dans tous les cas, en général, le réseau doit être borné.

### 2. Réseau de Pétri réinitialisable

Un réseau de Pétri est réinitialisable pour un marquage initial  $M_0$  si et seulement si :  $\forall M \in \overline{M_0}, \exists S$  tel que  $M \xrightarrow{S} M_0$ .

Si le modèle n'est pas réinitialisable cela veut dire que le système n'est pas répétitif.

### 3. Réseau de Pétri vivant

Un réseau de Pétri est vivant pour un marquage initial  $M_0$  si et seulement si :

$$\forall M_i \in \overline{M_0}, \forall t \in T, \exists S \text{ tel que } (M_i \xrightarrow{S} M_j \wedge S \text{ contient } t)$$

Autrement dit chaque transition reste tirable à partir de n'importe quel état et donc aucune partie ne peut devenir non tirable.

Un réseau de Pétri vivant implique que son fonctionnement est libre de blocage. Un blocage signifie en effet qu'il existe un (des) marquage(s) où certaines transitions ne sont pas tirables ; cela montre une défaillance du fonctionnement. Un modèle vivant assure donc que toute partie du système correspondant est toujours accessible.

### I.3.2.2.3 Réseau de Pétri Etiqueté

#### Définition

Un réseau de Pétri étiqueté est un réseau de Pétri ayant un marquage initial et une étiquette associée à chaque transition. Soit  $(q_1 | O_1)$  l'étiquette associée à la transition  $t_1$ , elle est telle que :

$q_1$  est une fonction booléenne (un prédicat),  $O_1$  est une liste d'opérateurs. Lorsque  $q_1$  est toujours vraie elle est omise ; si  $O_1$  est vide, elle est omise.

#### Règle de tir

Le tir d'une transition dans un réseau de Pétri étiqueté exige la satisfaction des deux conditions suivantes :

- a) La transition doit être activée (I.3.2.2.1 Définition 4) ;
- b) La transition doit être validée, c'est-à-dire que la condition logique  $q_1$  doit être vraie.

Le tir d'une transition provoque l'activation des opérateurs de la liste  $O_1$ , puis l'évolution des marquages.

### I.3.2.2.4 Schémas à Réseaux de Pétri [14]

Les schémas à réseaux de Pétri sont une extension des réseaux de Pétri constituée de :

#### 1) Un Graphe de commande

Le graphe de commande est un réseau de Pétri étiqueté. L'évolution du graphe de commande dépend donc de l'activation des transitions et de la validation des conditions logiques associées.

#### 2) Un Graphe de données

Le graphe de données est semblable au graphe de données de LOGOS et plus généralement des schémas de calcul parallèle. Le graphe de données est composé de trois sommets différents :

a) Des cellules mémoires  $M_i$  représentées par des rectangles ; certaines de ces mémoires peuvent être des cellules "d'entrée-sortie" où le monde extérieur peut écrire ou lire à tout instant ; ces cellules peuvent représenter des capteurs ou des actionneurs.

b) Des opérateurs  $O_i$  représentés par des cercles ; ils exécutent des opérations sur un ensemble de cellules appartenant au graphe de données.

c) Des cellules prédicat  $Q_k$  représentées par des losanges ; elles donnent en sortie les valeurs "vrai" ou "faux" associées aux prédicats du graphe de commande.

### 3) Interprétation d'un schéma à réseau de Pétri

Pour spécifier un système particulier, il sera nécessaire de préciser l'interprétation des sommets du graphe de données ; cette interprétation précise exactement les noms, les opérations... des sommets du graphe de données.

#### I.3.2.2.5 Propriétés des Réseaux de Pétri Non Etiqueté et Etiqueté [11]

(i) Si un réseau de Pétri étiqueté contient au moins un prédicat associé à une transition, ce ou ces prédicats peuvent seulement diminuer le nombre des marquages accessibles ; il en résulte que :

Réseau Non Etiqueté Borné  $\implies$  Réseau Etiqueté Borné

Mais :

Réseau Non Etiqueté Réinitialisable  $\not\implies$  Réseau Etiqueté Réinitialisable

Réseau Non Etiqueté Vivant  $\not\implies$  Réseau Etiqueté Vivant

(ii) Un réseau de Pétri non étiqueté peut contenir des transitions en conflit, mais si il est étiqueté le conflit doit être résolu ; à ce moment là, on dit que le réseau de Pétri est déterministe. Un réseau non déterministe peut indiquer une erreur de conception : il est donc nécessaire de signaler ceci lors de l'analyse car ce conflit devra être résolu à un niveau de conception plus bas.

### I.3.2.3 Analyse Structurelle et Technique de Conception(SADT) [16,17,18]

Le modèle SADT est basé sur la décomposition des systèmes en boîtes noires d'activités (Actigrams) et boîtes de données (datagrams). Chaque boîte pourra se décomposer de façon hiérarchique par une approche descendante en de nouvelles boîtes (6 boîtes au maximum) plus détaillées.

Des liens graphiques définissent les interfaces d'entrée-sortie et de contrôle entre les boîtes ; ces interfaces sont représentées par des flèches. Le mécanisme de réalisation de la boîte peut aussi être indiqué par une flèche connectée à la boîte (figure I.2).

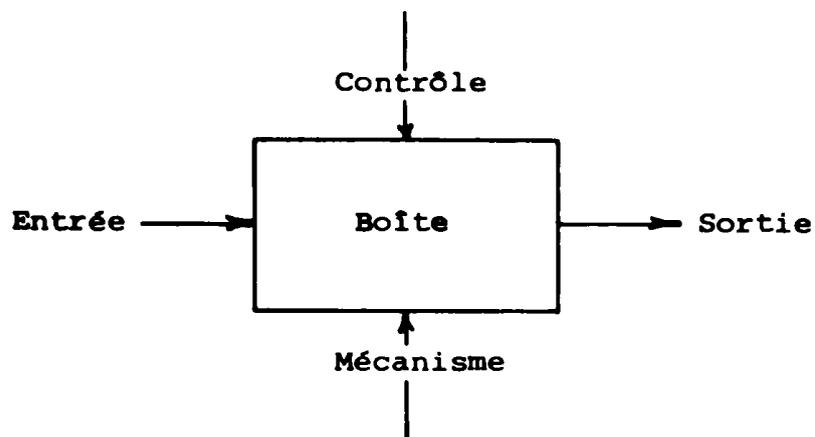


Fig. I.2 "La boîte noire de SADT"

Les primitives d'analyse de SADT précisent quel est le raisonnement permettant de créer une boîte, quelles en sont les fonctions et comment elle est construite.

SADT est une technique qui permet une modélisation lisible, structurée, facilement compréhensible, dans laquelle les communications des idées entre le concepteur et l'utilisateur du modèle est aidée par un langage, SA (Structured Analysis).

Une particularité importante de SADT provient de l'existence de trois niveaux dans la spécification ; il existe des auteurs qui écrivent les spécifications, des commentateurs qui les critiquent et des lecteurs qui ne font que comprendre. Ainsi, l'auteur doit transmettre son travail à plusieurs commentateurs selon une procédure parfaitement définie de façon à atteindre un accord sur la spécification étudiée, ceci afin de réduire les erreurs de conception.

Notons que cette technique a été utilisée par ITT Europe pour la représentation du matériel et du logiciel d'un autocommutateur [18] .

## 1.4 MOTIVATION ET CHOIX D'UN MODELE [1]

Comme nous l'avons déjà dit précédemment, les méthodologies de spécification sont diverses et chacune possède ses avantages et ses inconvénients ; en fait, pratiquement chaque système a ses besoins et caractéristiques particuliers qui peuvent favoriser une méthodologie plutôt qu'une autre.

Les systèmes temps réel en général et les autocommutateurs en particulier appartiennent à l'ensemble de ceux qui favorisent les modèles permettant d'exploiter les différents états (partiels ou totaux) du système, les interactions entre les entités communicantes et la synchronisation nécessaire afin d'accomplir une tâche donnée.

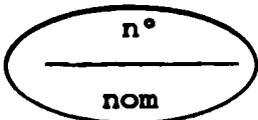
Du point de vue de la clarté du modèle, et bien qu'il représente tous les aspects fonctionnels, le modèle du diagramme d'états ne permet pas d'explicitier les interactions entre les différents processus ; au contraire, le modèle basé sur les réseaux de Pétri permet d'explicitier cet aspect important ; de plus, la modélisation par réseaux de Pétri apparaît privilégiée si l'on considère les possibilités importantes de vérification du modèle.

Il s'est avéré que nous avons pu réduire notre problème afin d'aboutir à une complexité raisonnable qui ne nécessitait pas l'utilisation d'une technique telle que celle donnée par SADT. Il était alors nécessaire de choisir entre les diagrammes d'états et les réseaux de Pétri. Nous allons maintenant montrer que, malgré leur utilisation très répandue dans certains secteurs y compris la téléphonie [19,20,21] , les diagrammes d'états s'avèrent moins adaptés que les réseaux de Pétri, surtout si l'aspect vérification est considéré comme important, ce qui apparaîtra dans la suite.

Nous donnerons ici, un exemple simple de la spécification d'un appel téléphonique classique par un diagramme d'états, fig. I.3 (a,b) et par un réseau de Pétri, fig. I.4.

En réalité, les figures I.3 (a) et I.3 (b) ne diffèrent que par leur représentation, la figure I.3 (b) étant la représentation la plus fréquemment trouvée.

La notation dans la figure I.3(a) est la suivante :

◆  signifie :  
n° : numéro d'état, nom : nom de l'état.

◆ texte 1 | texte 2 signifie :  
texte 1 : évènement d'entrée ;  
texte 2 : action par le système de commande.

La notation dans la figure I.3(b) est la suivante :

◆  signifie :  
n° numéro d'état, nom : nom d'état.

◆  signifie : évènement d'entrée.

◆  signifie : action par le système de commande.

La figure I.4 est un réseau de Pétri étiqueté dans lequel :

◆ la place porte le numéro d'un état et le nom de l'état est donné par un texte.

◆ la transition porte les évènements et l'action (séparés par une barre verticale).

A ce niveau, l'intérêt d'utiliser les réseaux de Pétri n'apparaît pas de manière fondamentale. Il faut cependant noter que, dans un grand système l'activité n'est pas toujours localisée et, en particulier, les processus

Déc : Décrochage  
Rac : Raccrochage  
DR : Demandeur  
Dé : Demandé

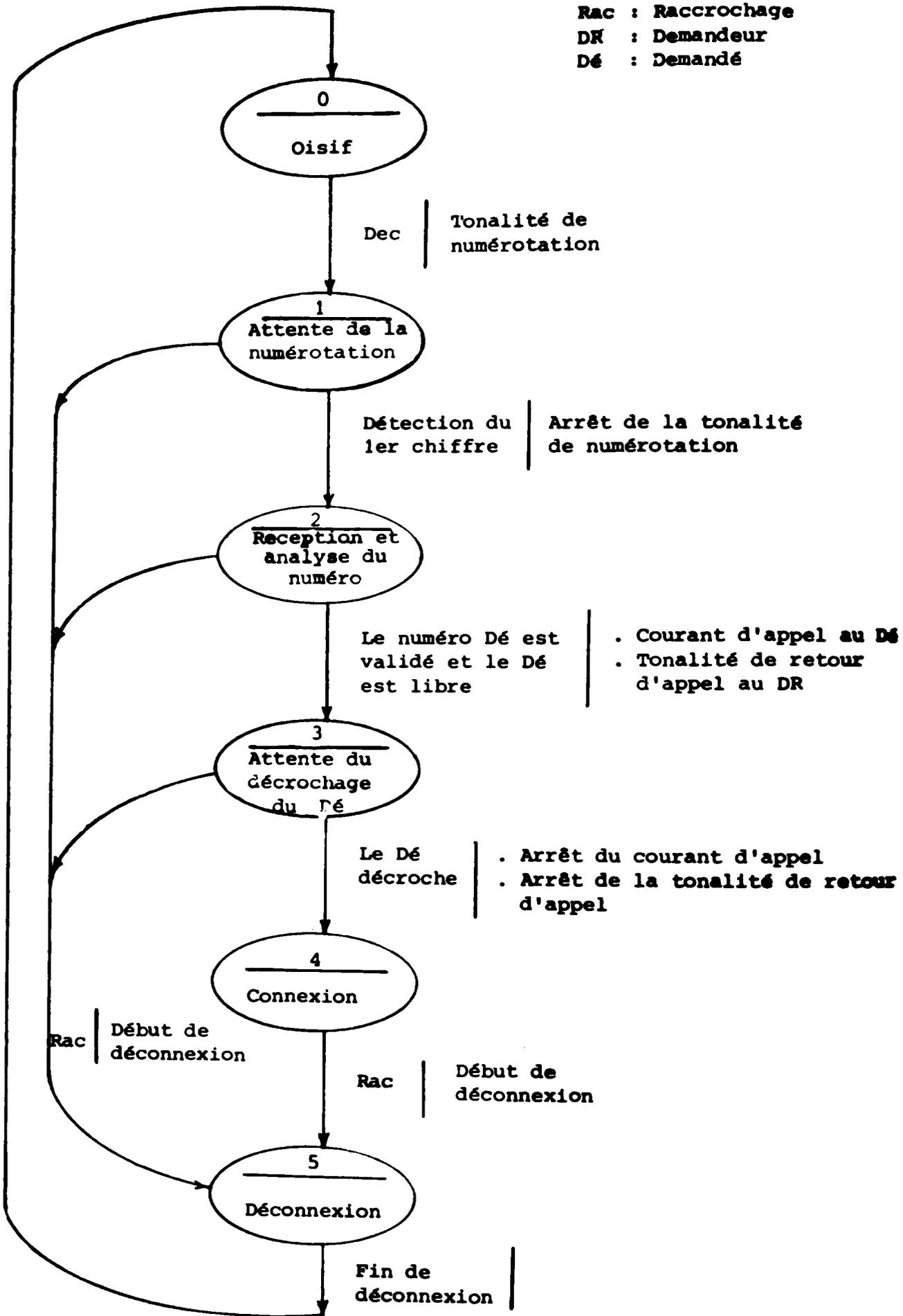


Fig I.3.a Modèle simple d'appel téléphonique par un diagramme d'états

Déc : Décrochage  
Rac : Raccrochage  
DR : Demandeur  
Dé : Demandé

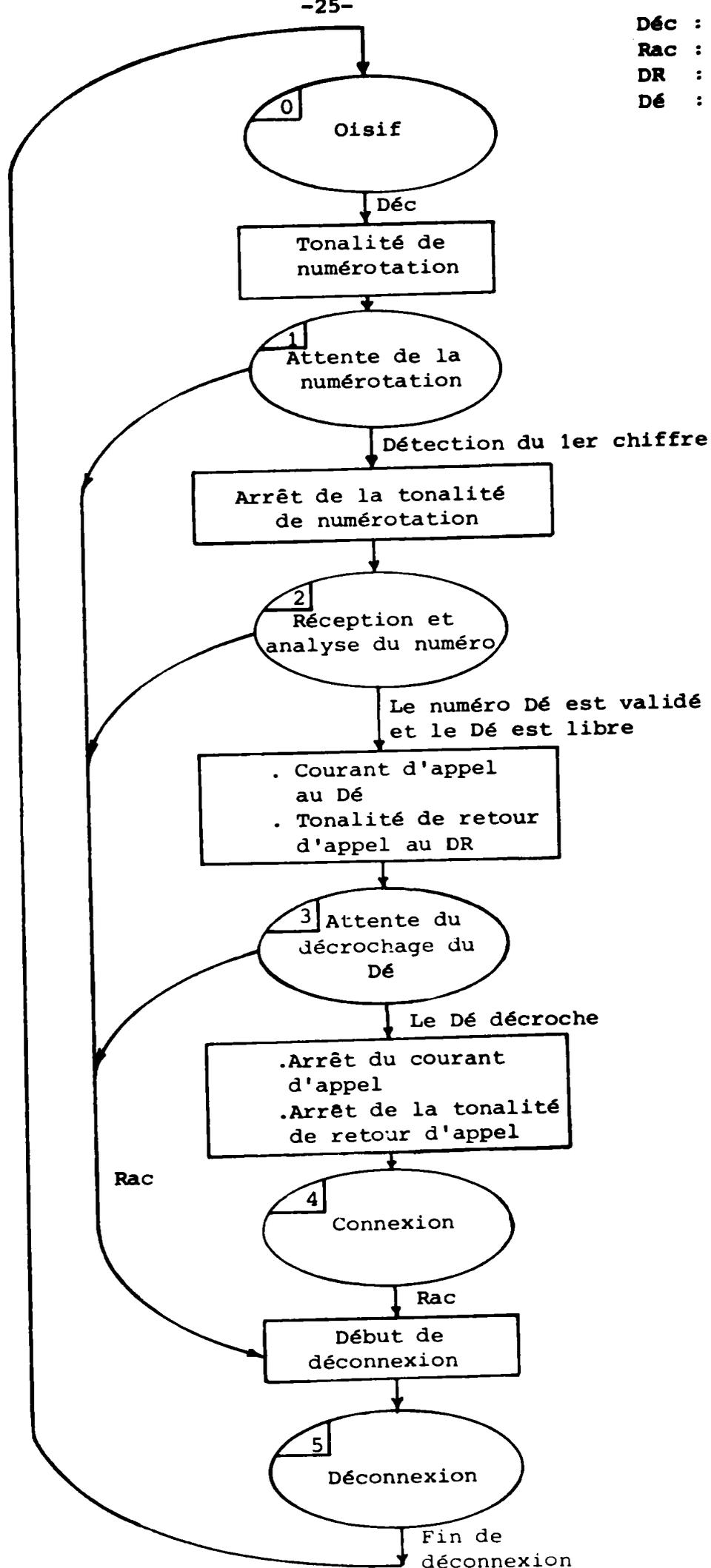


Fig. I.3.b. Modèle simple d'appel téléphonique par un diagramme d'états

Déc : Décrochage  
 Rac : Raccrochage  
 DR : Demandeur  
 Dé : Demandé

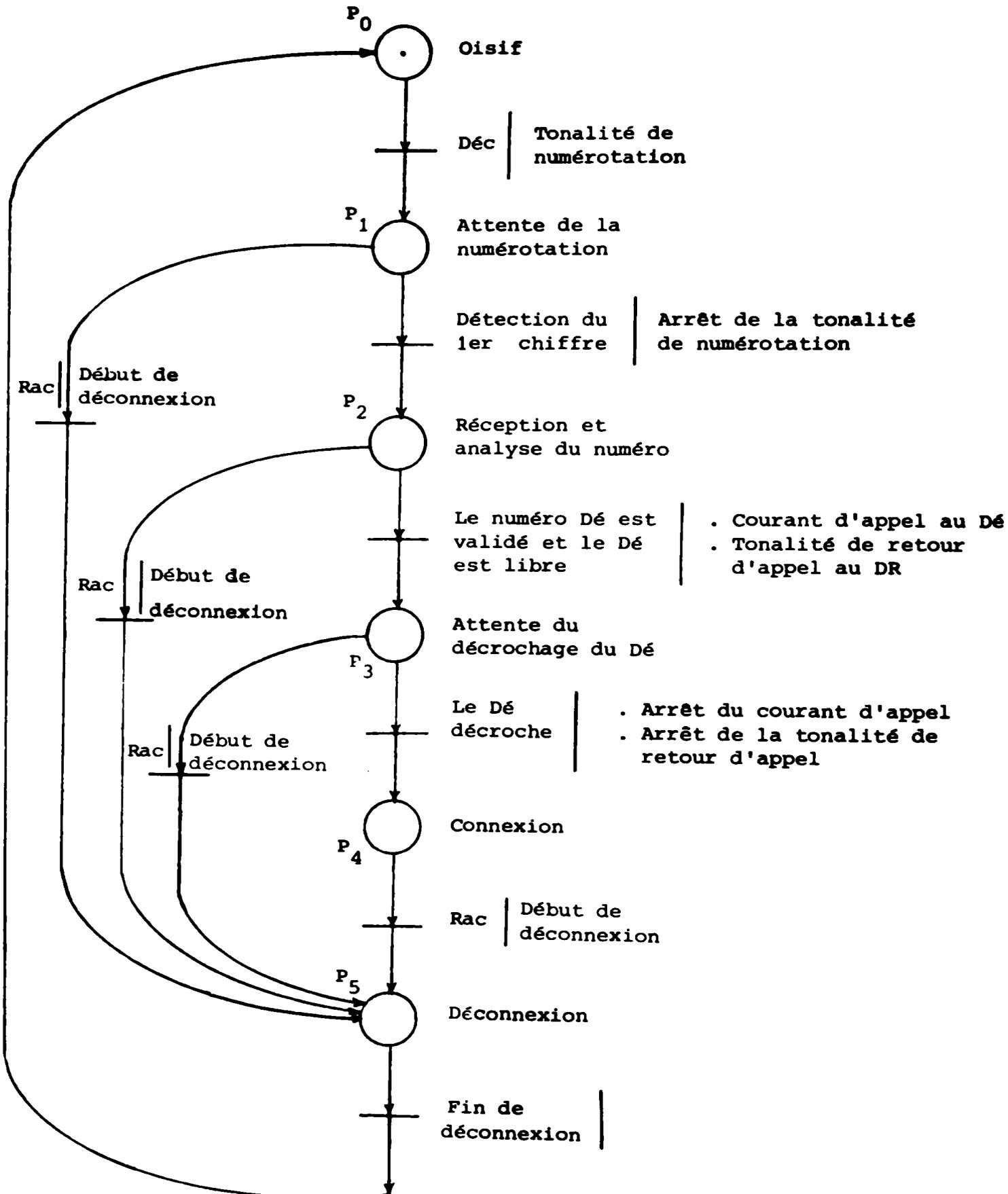


Fig. I.4 Modèle simple d'appel téléphonique par un réseau de Pétri

distants doivent coopérer pour réaliser leur tâche. Le problème de synchronisation inhérent à cette coopération doit être explicité clairement, ce que permettent les réseaux de Pétri par rapport aux diagrammes d'états.

Nous allons montrer, dans le cas de la communication entre deux abonnés, cet aspect essentiel des réseaux de Pétri.

Soient deux abonnés : A le demandeur et B le demandé. La figure I.5 représente le demandeur et le demandé séparément, en montrant les événements qui transitent d'un côté à l'autre ; nous prenons pour simplifier un seul état : demandeur et demandé dans l'état "Attente du décrochage du demandé", et, pour enlever l'ambiguïté, on appellera cet état "Attente Réponse" du côté demandeur et "Elaboration Réponse" du côté demandé.

Dans la figure I.5a les deux abonnés : A demandeur et B demandé sont modélisés par diagramme d'états. Supposant A dans l'état n "Attente Réponse" et B dans l'état m "Elaboration Réponse" :

a) Si le demandé physique B décroche (dec), B envoie un signal réponse à A et passe à l'état (m+1) "Connexion" ; si le signal réponse arrive à A lorsqu'il est dans l'état n, il évolue vers l'état (n+1) "Connexion".

b) Si le demandeur physique A raccroche (rac) avant que B ne décroche, A envoie un signal Fin à B et passe à l'état oisif ; si le signal Fin arrive à B quand il est dans l'état m, il évolue vers l'état oisif.

c) Mais si le décrochage de B et le raccrochage de A arrivent simultanément il est possible que B envoie Réponse à A et passe à l'état (m+1) "Connexion" et que A envoie Fin à B et passe à l'état oisif.

Ce mauvais fonctionnement n'apparaît pas explicitement et nécessite une réflexion du concepteur.

Considérons maintenant la figure I.5b dans laquelle les deux abonnés A,B sont modélisés par le réseau de Pétri. A partir du même état initial, l'analyse du fonctionnement du réseau donne les possibles fonctionnements suivants :

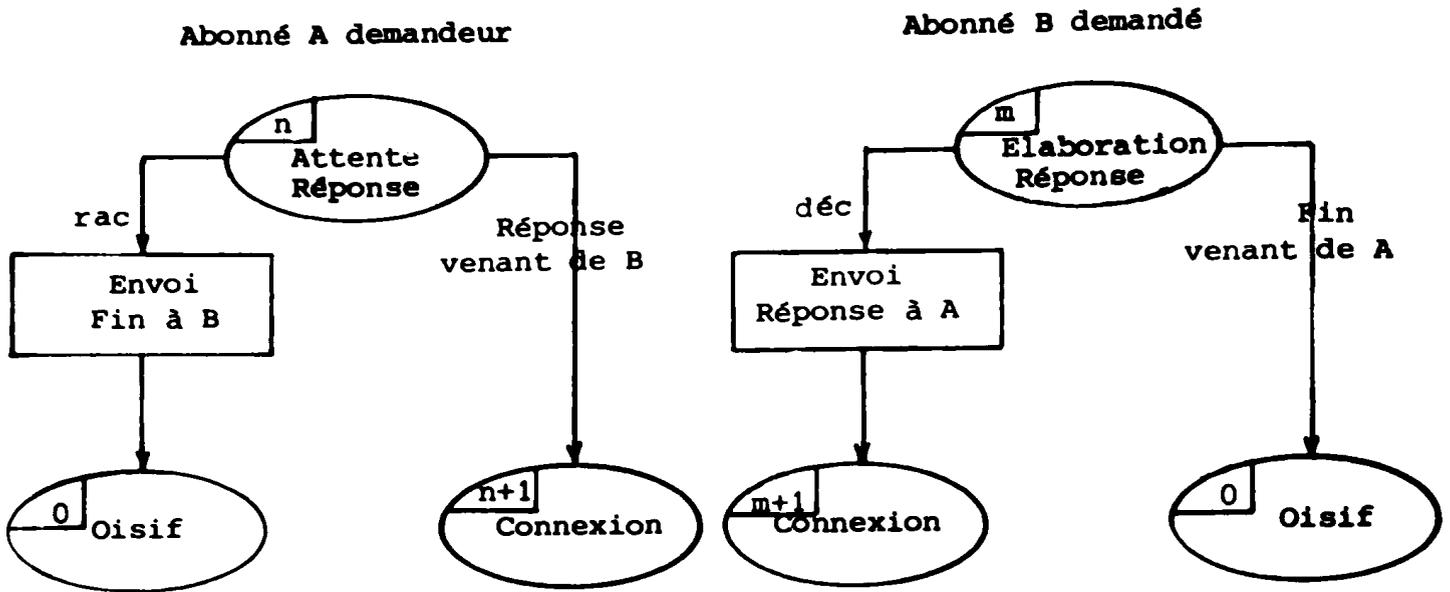


Fig. I.5. a

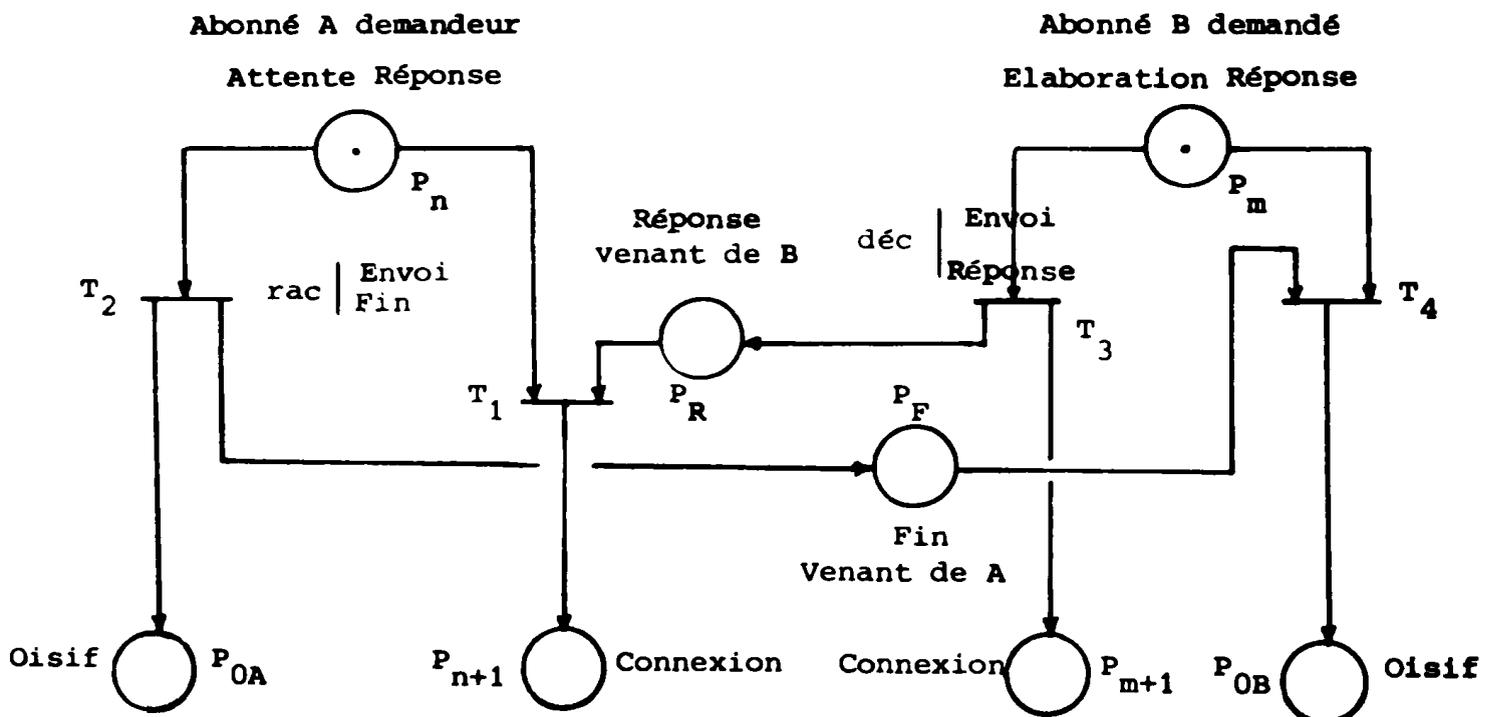


Fig. I.5.b Détection d'un blocage

a) Si B décroche (déc), la transition  $T_3$  est tirée provoquant l'envoi de Réponse à A ( $P_R$  sera marquée) et B passe à l'état  $P_{m+1}$  "Connexion" ; lorsque A, dans l'état  $P_n$ , reçoit Réponse, la transition  $T_1$  est tirée et A passe à l'état  $P_{n+1}$  "Connexion".

b) Si A raccroche (rac), la transition  $T_2$  est tirée provoquant l'envoi de Fin à B ( $P_F$  sera marquée) et A passe à l'état  $P_{OA}$  "oisif"; lorsque B dans l'état  $P_m$  reçoit Fin, la transition  $T_4$  est tirée et B passe à l'état  $P_{OB}$  "oisif").

c) Mais si le décrochage de B et le raccrochage de A arrivent simultanément alors  $T_3$  et  $T_2$  sont tirées aussi simultanément : B passe à l'état  $P_{m+1}$  "Connexion", A passe à l'état  $P_{OA}$  et les signaux Réponse et Fin sont envoyés ( $P_R$  et  $P_F$  sont marquées). Alors, un mauvais fonctionnement est détecté : le réseau est bloqué ; on voit donc l'intérêt du réseau de Pétri, puisque lors de la vérification l'existence du blocage est apparue naturellement (automatiquement).

Il apparait que, dans le cas général, la modélisation par réseaux de Pétri visualisera automatiquement lors de la vérification les situations conflictuelles, ce qui n'est pas possible en utilisant les diagrammes d'état.

## 1.5 CONCLUSION

Dans ce chapitre, nous avons exposé les principaux modèles actuellement utilisés pour spécifier les systèmes de commande et en particulier les autocommutateurs. L'étude montre qu'aucun modèle ne satisfait tous les critères nécessaires. Mais chaque système a ses besoins particuliers qui favorisent l'un ou l'autre des modèles.

Pour les systèmes temps réel dans lesquels il y a des interactions entre entités communicantes, les réseaux de Pétri paraissent être les plus

privilégiés et adaptés car ils permettent de mettre en oeuvre un important support de validation. De plus, la possibilité de les étiqueter permet d'étendre encore leur puissance de modélisation.

Nous allons dans le prochain chapitre voir comment les réseaux de Pétri interprétés peuvent être déduits des spécifications informelles afin d'accroître encore leur potentialité par rapport aux critères que doivent remplir les modèles. Ceci sera illustré, dans notre cas, par la modélisation du comportement de l'abonné dans un autocommutateur.

**SSS**

## CHAPITRE II

-----

METHODOLOGIE DE SPECIFICATION PAR UN COUPLE  
DESCRIPTION INFORMELLE - DESCRIPTION FORMELLE

ET PREMIERE

MODELISATION DU COMPORTEMENT D'UN ABONNE

-----



## II.1 INTRODUCTION

La sélection du modèle à réseaux de Pétri interprété étant effectuée, le problème est donc maintenant de créer le réseau à partir d'un cahier des charges, c'est-à-dire de passer d'une représentation informelle d'un système à une représentation formelle qui sera un réseau de Pétri.

Le concepteur va la plupart du temps directement à la formalisation (ou même à l'implémentation) d'un système. Cela signifie qu'il définit, par essais et erreurs, les liens entre tâches, l'adjonction de nouvelles tâches, les contraintes de vitesses et d'utilisation des ressources, etc... Par modifications successives, les différentes composantes du système apparaissent. Cette approche que l'on peut qualifier de naturelle n'est pas réaliste pour la conception d'un grand système car elle comporte une erreur de base : celle de considérer le système comme une entité monolithique.

Une première phase dite de définition semble donc importante. Son action sera double :

- dégager d'abord des niveaux hiérarchisés dans le système,
- constituer ensuite des classes d'objets à chaque niveau, en liaison avec le type d'application.

Son but est de mener par réécriture du cahier des charges à la phase de formalisation.

La phase de formalisation consistera alors essentiellement à traduire et à assembler l'ensemble des relations partielles entre objets.

Cette démarche peut paraître lourde et formelle (bien qu'elle parte d'un document informel). Cependant la conception d'un système de très grande échelle requiert une approche systématique, une réflexion profonde sur la nature des objets de la mission et les relations qui les lient. Il faut noter, de plus, que les choix intervenant ici ne seront pas "innocents" mais dictés par la nature du modèle final c'est-à-dire le réseau de Pétri.

## II.2 CONCEPTION HIERARCHIQUE D'UN SYSTEME COMPLEXE TEMPS REEL

[1, 2, 8, 35, 36]

Pour représenter un système complexe on peut supposer dans le cas général, qu'il est constitué de plusieurs niveaux hiérarchisés. En outre, dans un système temps réel, il existe des interactions avec le monde extérieur.

La figure II.1.a donne la représentation générale du niveau  $i$  dans un système complexe dans laquelle on représente les interactions avec le monde extérieur par  $E_i$ ,  $A_i$  où :

$E_i$  : Evènements venant du monde extérieur

$A_i$  : Actions sur le monde extérieur

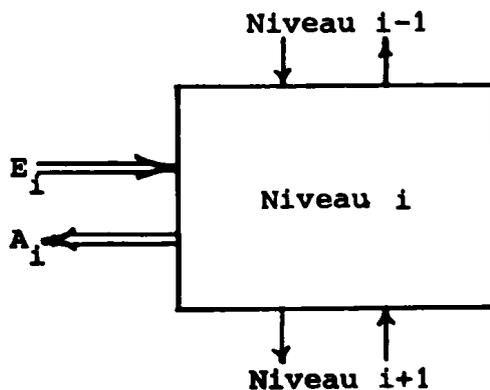


Fig. II.1a Représentation d'un niveau

Plus précisément, dans un niveau on peut également considérer plusieurs modules. La figure II.1b donne la représentation générale d'un système en tenant compte de cette considération.

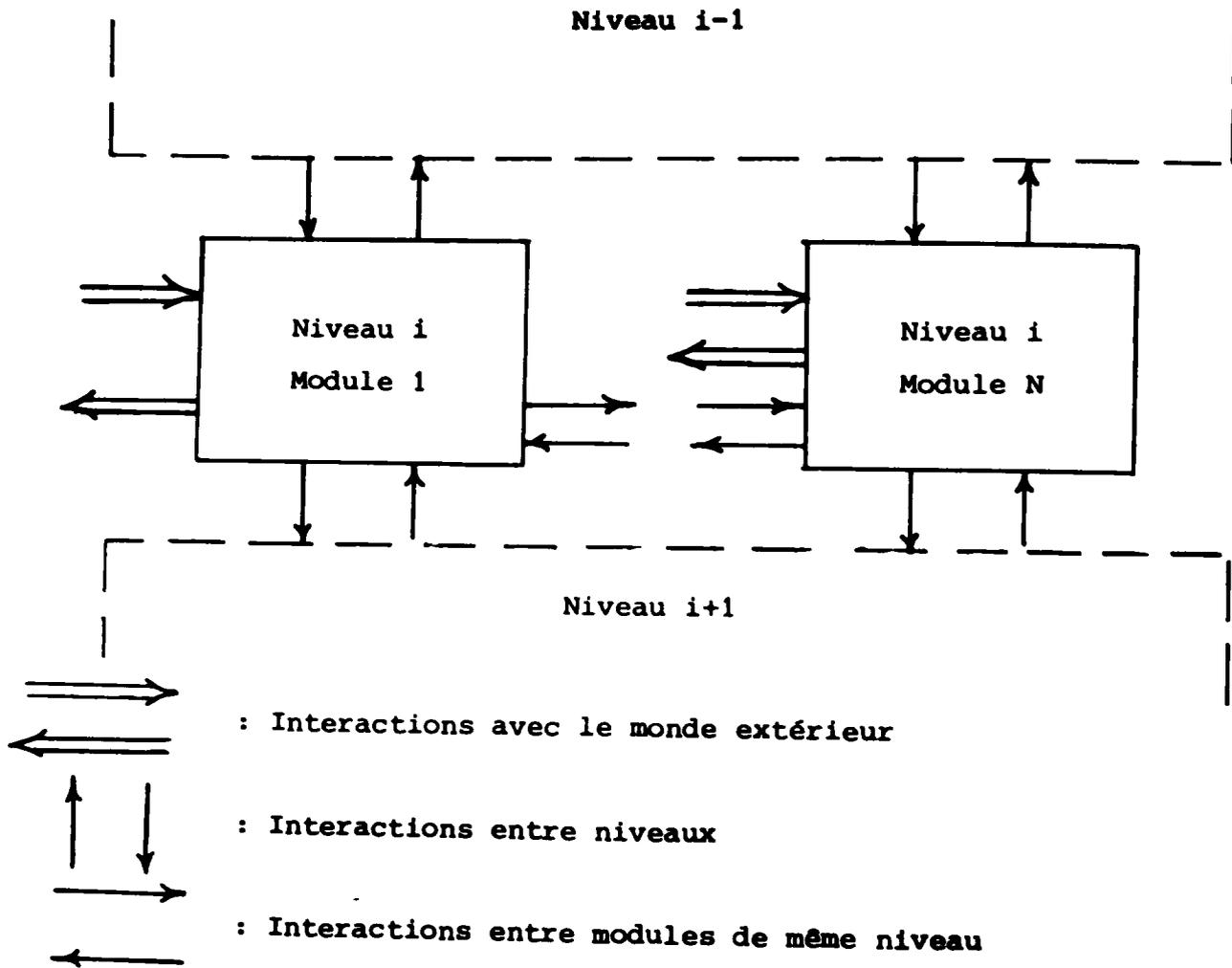


Fig. II.1b Représentation du système

## II.3 LE MODELE [1,9]

### II.3.1 LES OBJETS D'UN MODULE

Pour décrire le comportement d'un module dans un niveau du système, un ensemble  $D_i$  contenant sept classes d'objets est proposé :

$$D_i : \langle E_i, A_i, P_i, I_i, F_i, I'_i, F'_i \rangle \quad \text{où :}$$

\*  $E_i$  : ensemble des évènements venant du monde extérieur.

\*  $A_i$  : ensemble des actions sur le monde extérieur

- $P_i$  : états (partiels ou totaux) qui peuvent être un état d'une ressource ou procédure en cours ou attente d'un prédicat
- $I_i$  : évènements venant du niveau inférieur  $i+1$
- $F_i$  : actions sur le niveau inférieur  $i+1$   
 $I_i$  et  $F_i$  représentent les interactions avec le niveau inférieur  $i+1$
- $I'_i$  : évènements venant des autres modules du niveau  $i$
- $F'_i$  : actions sur les autres modules du niveau  $i$

Par la suite, nous ne considérons qu'un module par niveau et les objets  $I'_i$ ,  $F'_i$  n'interviendront pas.

Un niveau peut être modélisé par la figure II.2.

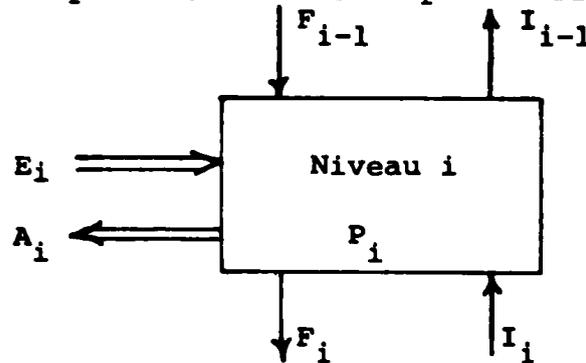


Fig. II.2 Modèle d'un niveau

### II.3.2 CORRESPONDANCE ENTRE LES SOMMETS DU RESEAU DE PETRI ET LES CLASSES D'OBJETS

Dans ce paragraphe nous allons préciser comment nous proposons de représenter les objets des classes par un réseau de Pétri. Ceci sera effectué en utilisant les conventions suivantes :

- les états  $P_i$  seront représentés par des places
- les évènements  $E_i$  et  $I_i$  seront représentés par des prédicats dans les transitions étiquetées

♦ les actions  $A_i$  et  $F_i$  seront représentées par des actions dans les transitions étiquetées.

Le problème qui apparait maintenant est le suivant : il s'agit d'établir, par une suite de transformations sur un texte informel dont les objets ont été classifiés, un réseau de Pétri qui soit déduit du modèle informel.

### II.3.3 MODELISATION PAR LE COUPLE INFORMALITE-FORMALITE

Nous présentons un module  $i$  par un couple  $\langle S_i, D_i \rangle$  où :  $S_i$  est un ensemble de phrases d'évolution du système ; c'est la grammaire du système.

En raison de la constitution des réseaux de Pétri, il est suggéré que l'évolution de l'abonné suive la règle générale suivante :

Etat + Predicat (Evènement ou Test)  $\longrightarrow$  Etat + Action

où :

Etat :  $\langle$ état d'une ressource ou procédure en cours ou attente d'un prédicat $\rangle$

Predicat (Evènement ou test) :  $\langle$ réception d'un évènement ou résultat d'un test $\rangle$

Action :  $\langle$ Action entreprise $\rangle$

## II.4 PREMIERE MODELISATION DU COMPORTEMENT D'UN ABONNE

Le modèle d'un abonné est trop complexe pour établir un cahier des charges (qui contiendrait de très nombreuses informations) en une seule étape. Nous choisissons de procéder par affinements successifs, ce qui paraît être l'approche la plus sûre et la plus efficace.

Prenons maintenant, afin d'illustrer la méthode, un exemple de base en téléphonie : la spécification du comportement d'un abonné dans un autocommutateur.

#### II.4.1 TEXTE DE SPECIFICATION HAUT NIVEAU DE L'ABONNE

Le texte suivant décrit les spécifications du comportement d'un abonné au niveau le plus élevé ( $i=0$ ). Cet ensemble de phrases exprimées en langage naturel constitue l'ensemble  $S_0$ .

$S_0$  :

1. Si l'abonné libre décroche, il devient occupé et une procédure appelant commence.
2. Si l'abonné appelant raccroche, il devient libre.
3. Si l'abonné occupé est appelé, il reste occupé et le signal appelé occupé est envoyé au demandeur.
4. Si l'abonné libre est appelé, il devient occupé et une procédure appelé débute.
5. Si l'abonné appelé raccroche, il devient libre.

A ce niveau là, les interactions entre l'autocommutateur et le monde extérieur sont facilement repérables et se traduisent par les signaux décrochage, appel et appelé occupé.

Notons en particulier que les procédures "appelant" et "appelé" ne sont pas précisées ici, et le seront donc dans une étape ultérieure.

Il est donc maintenant nécessaire de manipuler ces phrases.



Phrase 2 :  
Etat : { Abonné occupé ; Prédictat : fin Appelant →  
          { Procédure Appelant en cours  
Etat : Abonné libre

Phrase 3 :  
Etat : Abonné occupé ; Prédictat : appel →  
Etat : Abonné occupé ; Action : appelé occupé

Phrase 4 :  
Etat : Abonné libre ; Prédictat : appel →  
Etat : { Abonné occupé ; Action : début Appelé  
          { Procédure Appelé en cours

Phrase 5 :  
Etat : { Abonné occupé ; Prédictat : fin Appelé →  
          { Procédure Appelé en cours  
Etat : Abonné libre

Notons que certaines faibles ambiguïtés ont été levées : dans les phrases 2 et 5, le texte n'explique pas que l'abonné est occupé (bien que cela soit facilement déductible respectivement des phrases 1 et 4).

Nous pouvons maintenant déduire facilement un microréseau de Pétri à partir de chacune de ces phrases. La figure II.2 donne les microréseaux correspondants.

La prochaine étape sera donc l'assemblage de ces phrases entre elles, c'est-à-dire l'assemblage de microréseaux.

### II.4.3 ASSEMBLAGE DES MICRORESEaux

Chaque microréseau traduit une partie du comportement de l'abonné : pour obtenir le comportement global de l'abonné, il faut assembler tous les microréseaux obtenus.

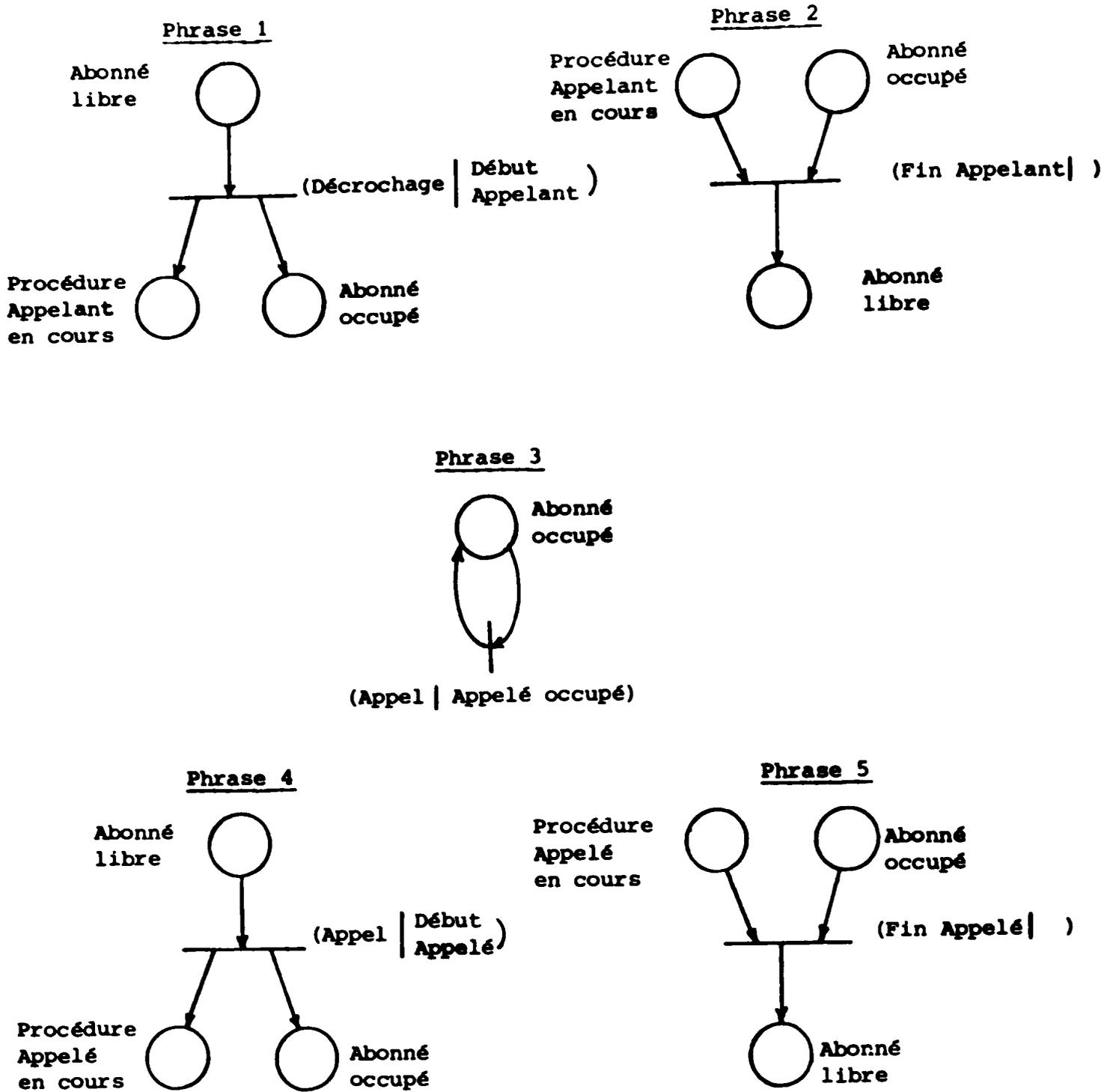


Fig II.2 Modèles des Phrases en Micro Réseaux de Pétri

La tâche d'assemblage apparait relativement délicate et se fera pour le moment manuellement.

Un microréseau sera sélectionné, puis connecté à l'un des restants, le réseau obtenu est connecté au microréseau suivant ... etc. La connexion consiste à créer les liens entre les transitions et les places, à identifier les places identiques et à les fusionner en une seule.

Les microréseaux des phrases 1 et 2 sont connectés en notant que trois états sont identiques dans les deux microréseaux, à savoir "procédure appelant en cours", "abonné libre" et "abonné occupé". Ces trois places, ayant même sémantique, sont à fusionner ; cette fusion conduit à la figure II.3.a.

En considérant maintenant le microréseau de la phrase 3, "abonné occupé" étant toujours le même état, il suffit d'ajouter dans la figure II.3.a la boucle sur la place "abonné occupé" ; cela nous donne la figure II.3.b.

La connexion du microréseau de la phrase 4 conduit au réseau de la figure II.3.c.

De la même façon, on connecte le microréseau de la phrase 5 à ce dernier réseau en fusionnant les deux places "abonné libre" et "procédure appelé en cours". L'étape d'assemblage avec réduction du nombre de places et transitions par fusion aboutit à l'obtention du modèle de l'abonné dans un autocommutateur donné par la figure II.3.d.

#### II.4.4 VERIFICATION DU MODELE

Ayant obtenu un réseau de Pétri, il est connu que nous pouvons alors l'analyser. Le point important est que la mise en évidence de ces propriétés peut se faire de façon automatique. C'est en particulier le but de l'outil graphique interactive OGIVE (Outil Graphique Interactive de VERification) [13] développé au LAAS. Il permet de vérifier qu'un réseau



est borné, sauf, vivant, réinitialisable par trois types de méthodes : énumération des marquages conséquents, réduction graphique du réseau, utilisation d'invariants (nous ferons d'ailleurs une utilisation extensive de cet outil au chapitre suivant).

Ainsi, si le réseau obtenu après la phase d'assemblage est analysé et trouvé incorrect, alors cela signifie que l'une des étapes précédentes, spécification informelle, traduction en règles ou assemblage, a été effectuée de façon erronée. Dans ce cas, la spécification doit être reprise. L'erreur révélée peut provenir d'objets mal définis, de liaisons non exprimées et laissées implicites, d'appellation du même objet par différents noms, etc...

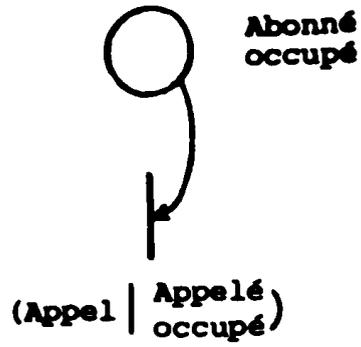
Certaines erreurs de ce type ont en fait été décelées. Par exemple, la phrase 3 précédente :

3. Si l'abonné occupé est appelé, il reste occupé et le signal appelé occupé est envoyé à l'appelant.

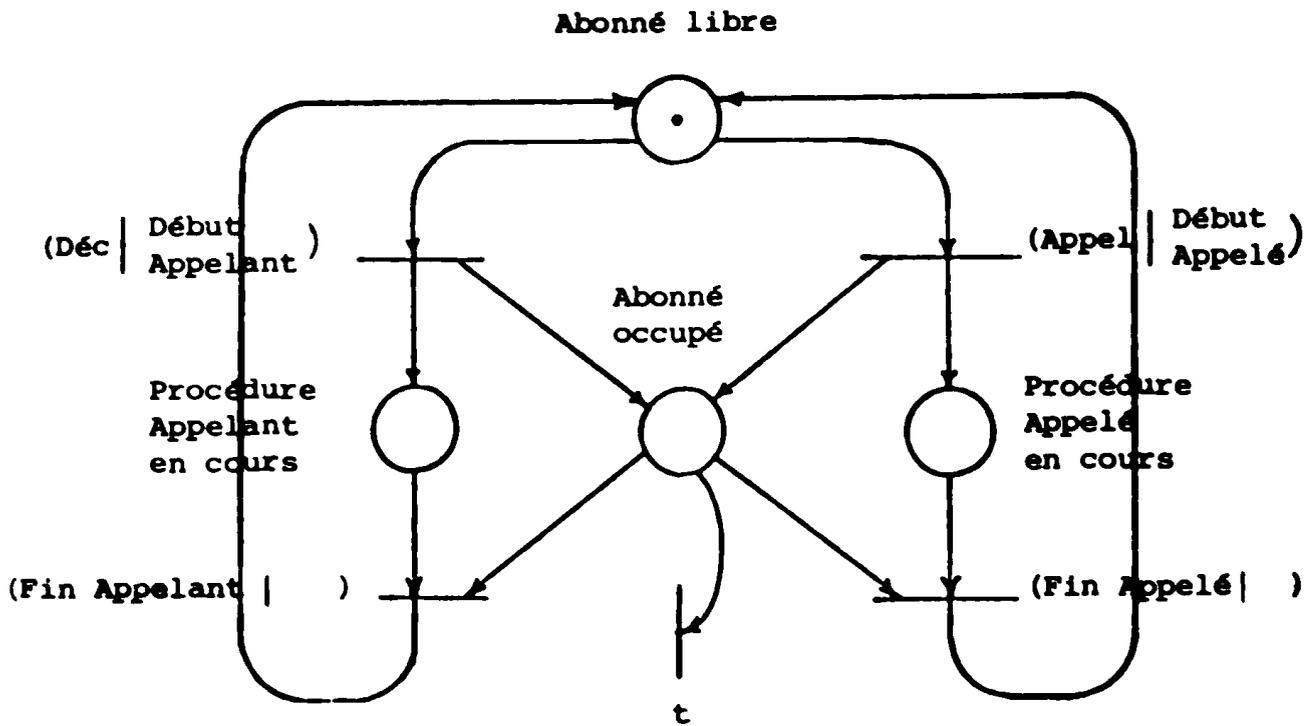
avait primitivement été écrite sous la forme suivante :

3'. Si l'abonné occupé est appelé, le signal appelé occupé est envoyé.

La traduction de cette phrase conduit au micro-réseau de la figure II.4.a et son assemblage dans  $S_0'$  conduit au réseau de la figure II.4.b. L'analyse de ce réseau montre qu'il est non vivant. En effet, après le tir de la transition  $t$ , le réseau n'a plus de jeton et aucune transition ne peut plus être tirée. Ceci provient du fait que le réseau est incomplet car la phrase 3' est elle-même incomplète : nous n'avons pas précisé le nouvel état après l'appel. La raison est que le fait que "l'abonné occupé après l'appel restait occupé" a paru tellement évident qu'il n'a pas été explicité ; mais il est indispensable de l'expliciter car ce fait n'est pas toujours vrai, par exemple, est faux si l'abonné a la possibilité de l'appel en instance. L'analyse du réseau a immédiatement révélé cette imprécision.



(a)



(b)

Fig. II.4

Quelques autres erreurs ont aussi pu être corrigées et le réseau résultant, donné par la figure II.3.d est borné, sauf vivant et réinitialisable.

Nous utiliserons ce réseau en le développant dans le chapitre suivant.

Notons enfin que cette approche a été aussi appliquée à la représentation de la conférence et de l'appel en instance [9] .

## II.5 CONCLUSION

Nous venons de montrer, déjà au niveau d'une modélisation simple du comportement d'un abonné, l'intérêt que procure l'utilisation de la méthode de spécification qui a été développée et surtout l'intérêt des réseaux de Pétri comme support formel d'un texte informel.

La caractéristique fondamentale de cette méthode est que le texte informel (le texte en langage naturel qui exprime les besoins) est le point de départ de l'étape de formalisation. Elle permet de lever les ambiguïtés du texte informel et de fournir aux équipes participant au projet une double explication, informelle (rendue non ambiguë) et formelle, du fonctionnement désiré.

Nous allons voir, dans le chapitre suivant, comment le réseau de Pétri pourra être étendu vers un niveau inférieur afin de prendre en compte, modéliser et analyser les interactions entre deux abonnés compte tenu de différents protocoles de communication déduits des recommandations CCITT N°6 et 7 [26,27] .



# CHAPITRE III

-----

CONCEPTION ET VERIFICATION  
DES PROTOCOLES DE LA COMMUNICATION  
MODELISATION ET ANALYSE  
DES INTERACTIONS ENTRE ABONNES  
BASEES SUR LES RECOMMANDATIONS CCITT N°6,7

-----



### III.1 INTRODUCTION

Dans le chapitre précédent nous avons modélisé le comportement d'un abonné au niveau qui nous paraît le plus haut ; pour expliciter les niveaux plus bas, il faut maintenant détailler avec soin la procédure appelant et la procédure appelé. Il est donc nécessaire de spécifier les protocoles de communication (Commande de la Communication entre deux abonnés (Signalisation)), c'est-à-dire l'établissement et la rupture d'une communication téléphonique. Ce problème nous permettra de mettre en évidence, comme cela a déjà été fait dans l'équipe LC du LAAS dans d'autres applications [10,22,23,24], l'intérêt de l'utilisation des réseaux de Pétri lors de la modélisation des protocoles.

Dans notre cas, la spécification du protocole de communication basée sur les recommandations CCITT N°6,7, consistera à expliciter de façon formelle et non ambiguë, en utilisant les réseaux de Pétri, les interactions nécessaires pour l'établissement et la rupture d'une communication entre un abonné appelant (procédure Appelant) et l'abonné appelé (procédure Appelé).

Cette modélisation permettra en particulier la vérification du protocole par l'analyse du réseau obtenu, analyse qui fera en particulier apparaître certaines erreurs de conception (blocages, dépassements de capacité) qui peuvent exister et qui sont représentées par les réseaux de Pétri.

Dans ce chapitre, nous spécifierons et analyserons différents protocoles.

Après leur spécification et leur vérification, nous proposons des modèles de protocoles validés qui nous paraissent intéressants car ils pourraient être applicables afin de définir des normes précises, non ambiguës et correctes. Comme il apparaîtra par la suite, ces modèles diffèrent par le nombre et la nature des messages échangés.

Néanmoins, avant de préciser ceci, nous allons d'abord donner les principales erreurs de conception qui sont habituellement prises en compte lors de la conception des protocoles. Ces erreurs nous permettront, après avoir donné différentes représentations possibles de la synchronisation entre entités communicantes, d'analyser les réseaux de Pétri proposés et d'en déduire de nouveaux protocoles possédant, eux, un fonctionnement correct. Nous pourrons ensuite utiliser ces résultats pour modéliser plusieurs protocoles déduits des recommandations CCITT N°6,7 [26, 27] .

### III.2 ERREURS DE CONCEPTION DANS UN PROTOCOLE [25]

Parmi les erreurs possibles, nous citerons ici celles qui nous paraissent les plus importantes dans les systèmes de communication.

#### 1. Etat de blocage

Un protocole est dit être dans un état de blocage lorsqu'il atteint un état à partir duquel aucune évolution n'est possible.

#### 2. Manque de spécification

Ce type d'erreurs peut survenir soit consécutivement à un oubli dans le texte des spécifications informelles, soit pendant la transformation menant des spécifications informelles aux spécifications formelles. Par exemple, dans le cas de la communication téléphonique, l'accusé de rattachage ne figure pas de façon explicite dans les systèmes de signalisation.

#### 3. Interactions non exécutables

Un concepteur peut spécifier une émission ou (et) une réception de messages qui ne peuvent pas exister dans les conditions normales d'opération. Par exemple, la réception de la réponse, pendant que l'abonné est en conversation, ne peut survenir.

#### 4. Etats ambigus

Les états ambigus apparaissent si plus d'un état dans un processus communiquant correspond à un seul état dans l'autre.

Notons que les états ambigus n'impliquent pas nécessairement une erreur, mais il faut les traiter soigneusement pour s'assurer qu'il n'y a pas de situation erronée.

### III.3 METHODOLOGIES DE CONSTRUCTION DE PROTOCOLES

Un protocole est l'ensemble des règles qui régit la coopération entre deux ou plusieurs processus qui communiquent. Dans chacun des processus apparaissent les envois et les réceptions des messages assurant la synchronisation. Ainsi, en général, une émission par l'un induira une réception dans l'autre. Après avoir spécifié ces différents séquençements, le problème qui apparaîtra concerne la connexion des réseaux représentant le processus émetteur et le processus récepteur. Pour ceci, il sera nécessaire d'explicitier la façon dont ces transitions sont liées.

Considérons maintenant les différentes modalités.

#### a) Synchronisation par fusion

Cette approche est définie par la fusion en une seule des transitions correspondantes dans les réseaux considérés : la transition représentant l'émission d'un message est fusionnée avec celle qui représente la réception de ce message. Dans ce cas, on considère que l'émission et la réception sont simultanées. En effet, cela suppose implicitement un médium de transmission nul.

A titre d'exemple, la figure III.1.a montre deux processus A et B. A envoie un message par la transition notée (!X) et B le reçoit en (?X). La figure III.1.b illustre le schéma obtenu après la fusion (!X,?X).

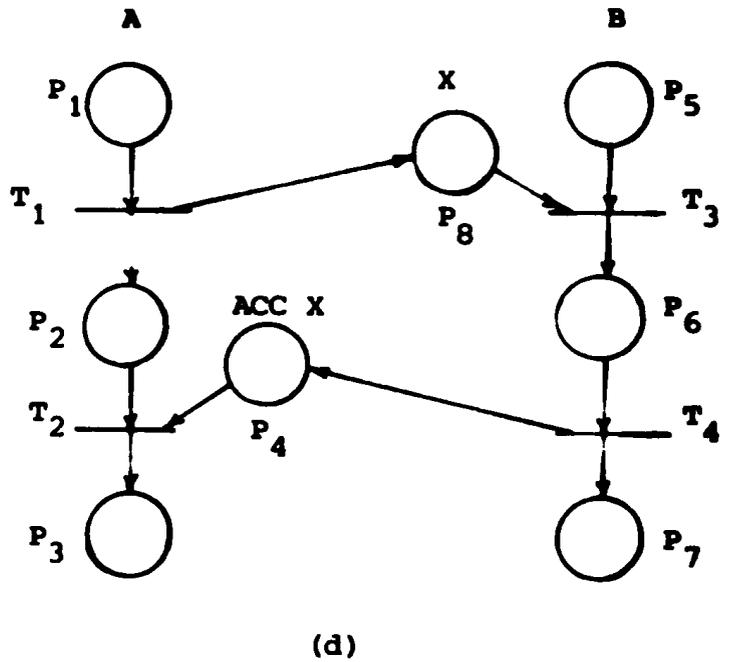
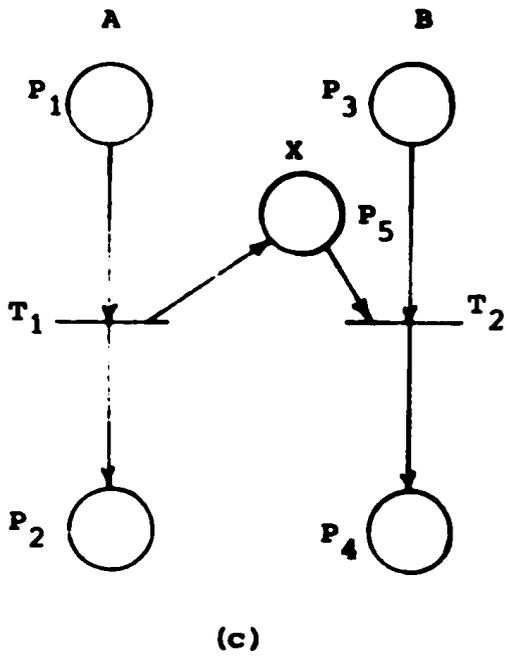
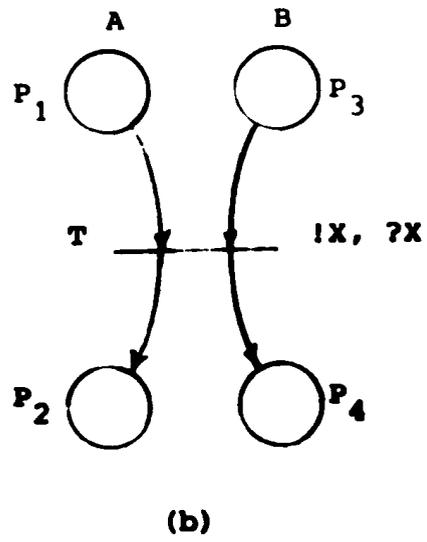
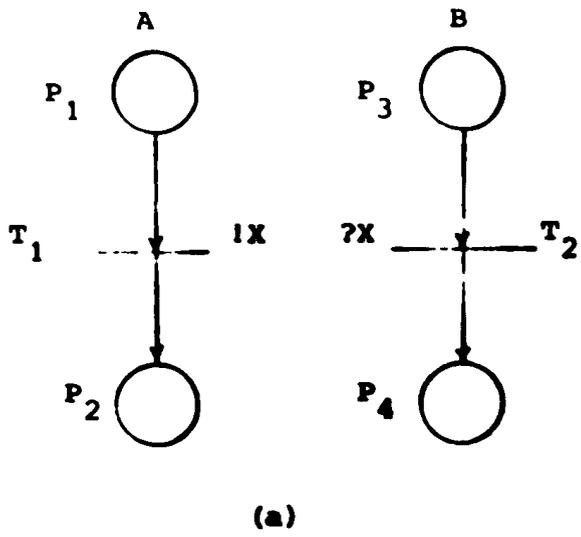


Fig. III.1 Constructions de protocoles de communication

Malgré que les protocoles basés sur l'utilisation de la fusion soient les plus facilement vérifiables car les plus simples du point de vue des interactions entre les processus, l'hypothèse de synchronisation liée à la fusion n'est pas satisfaisante car elle néglige l'existence du médium de transmission et nous ne la considérerons plus par la suite.

b) Synchronisation par envoi d'un message

Le principe de cette modalité repose sur l'envoi d'un message d'un processus à l'autre pour effectuer la synchronisation. La figure III.1.c représente les deux processus A, B reliés par le message (X). L'échange de ce message est représenté par le marquage de la place  $P_5$ .

Il apparaît ici que le médium de transmission est représenté ; le message X dans le médium de transmission est modélisé par le marquage de la place  $P_5$  : c'est donc une représentation possible mais il est bien clair que ce n'est pas la seule possibilité.

c) Synchronisation par accusé de réception

Cette modalité est aussi très connue et plus fiable que la précédente car elle nécessite l'échange de deux messages d'interactions entre les processus communicants : un processus envoie un message et attend l'accusé de réception venant de l'autre processus. Cela rend le couplage plus fort durant le déroulement de la communication et assure à l'expéditeur que le message est arrivé dans un état consommable pour le destinataire. On trouve là aussi que le médium de transmission est représenté. Dans la figure III.1.d, les deux processus A et B sont à nouveau considérés ; pour communiquer avec B, le processus A envoie une requête X à B et passe à l'état suivant (place  $P_2$ ) attendant la réponse de B ; lorsque B est prêt et reçoit la requête X, il passe à l'état suivant (place  $P_6$ ) pour élaborer la réponse à envoyer à A : B envoie la réponse (ACC X) et continue à l'état suivant (place  $P_7$ ), enfin lorsque A reçoit l'accusé X il passe à l'état suivant (place  $P_3$ ) et l'interaction est finie.

Notons que dans les protocoles, il peut exister des messages portant des informations et représentant implicitement l'accusé de réception.

Certaines conséquences pouvant en résulter ont été données en [8,10] : ainsi il apparait que, selon la modalité de synchronisation choisie, la validation peut conduire à des résultats totalement différents : correct dans un cas, non borné et vivant dans un autre, borné et non vivant dans un troisième.

Nous avons développé ces trois modalités de synchronisation car elles nous paraissent les plus utilisées. Néanmoins, ces interactions ne sont jamais utilisées seules et elles s'insèrent dans le cadre plus général de la communication. Avant d'aborder l'étude des protocoles de communication, nous allons d'abord illustrer quelles peuvent être les caractéristiques des processus se synchronisant, et en particulier en soulignant leurs états respectifs vis-à-vis de la commande de la communication, c'est-à-dire le début (l'établissement) et la fin (rupture).

### III.4 MODES DE COMMUNICATION

Afin d'explicitier les problèmes rencontrés et de nous rapprocher de l'application téléphonique, considérons deux processus A et B pouvant établir et (ou) rompre une communication. Selon les droits que ces entités vont posséder, nous pourrons distinguer plusieurs cas significatifs intéressants. Notons que nous considérons l'hypothèse d'un milieu parfait, c'est-à-dire n'introduisant pas des erreurs sur les messages échangés.

#### III.4.1 MAITRE AU DEBUT ET EN FIN DE COMMUNICATION

Dans ce mode, celui qui fait l'appel (demande l'établissement de la communication) possède seul le droit de demander la rupture. Ce mode est représenté dans la figure III.2.a, où sont utilisés les messages de base "appel", "occupé", "réponse" et "fin". Si le demandeur envoie un "appel", le demandé lui déclare en retour soit son occupation

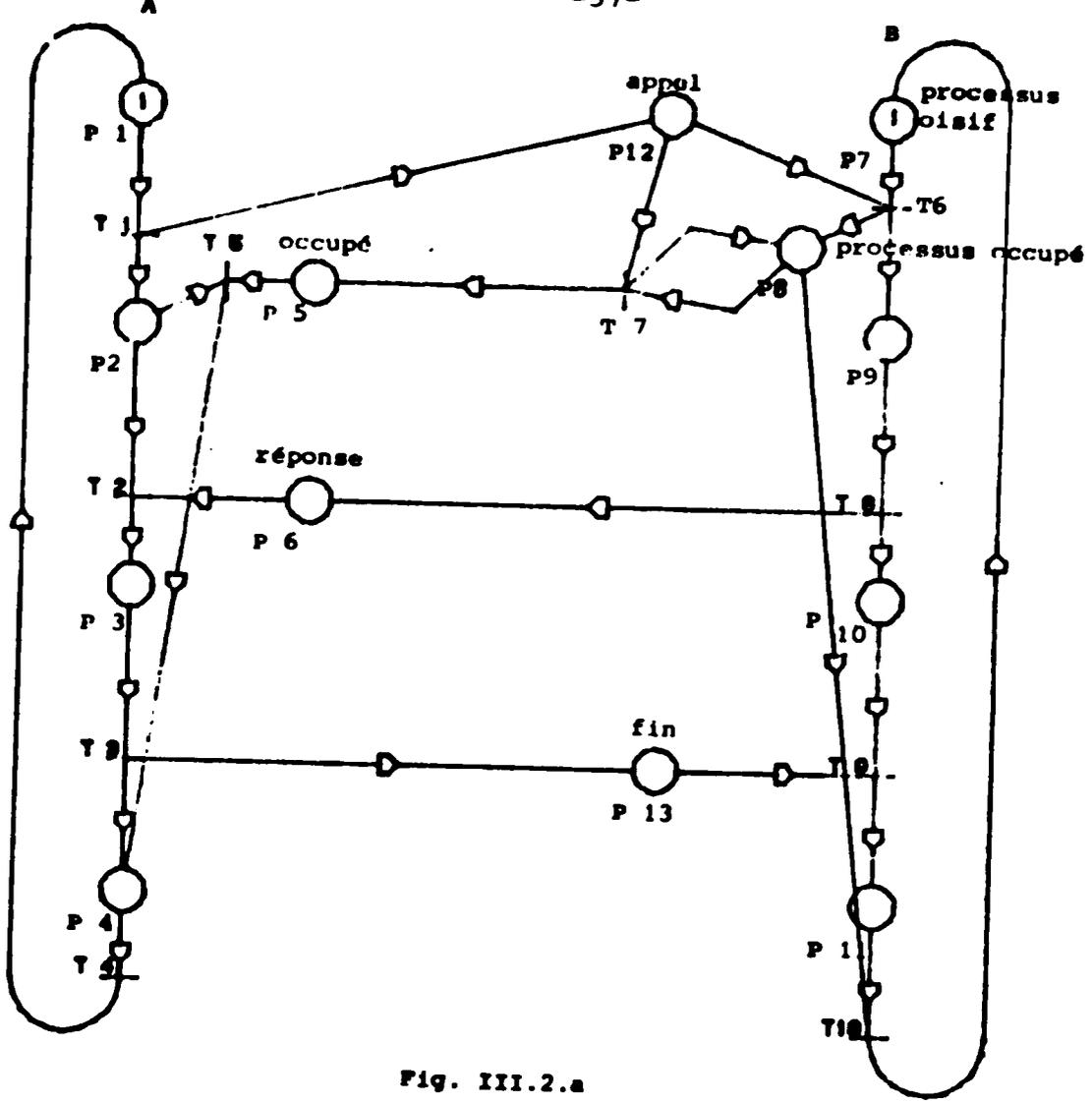


Fig. III.2.a

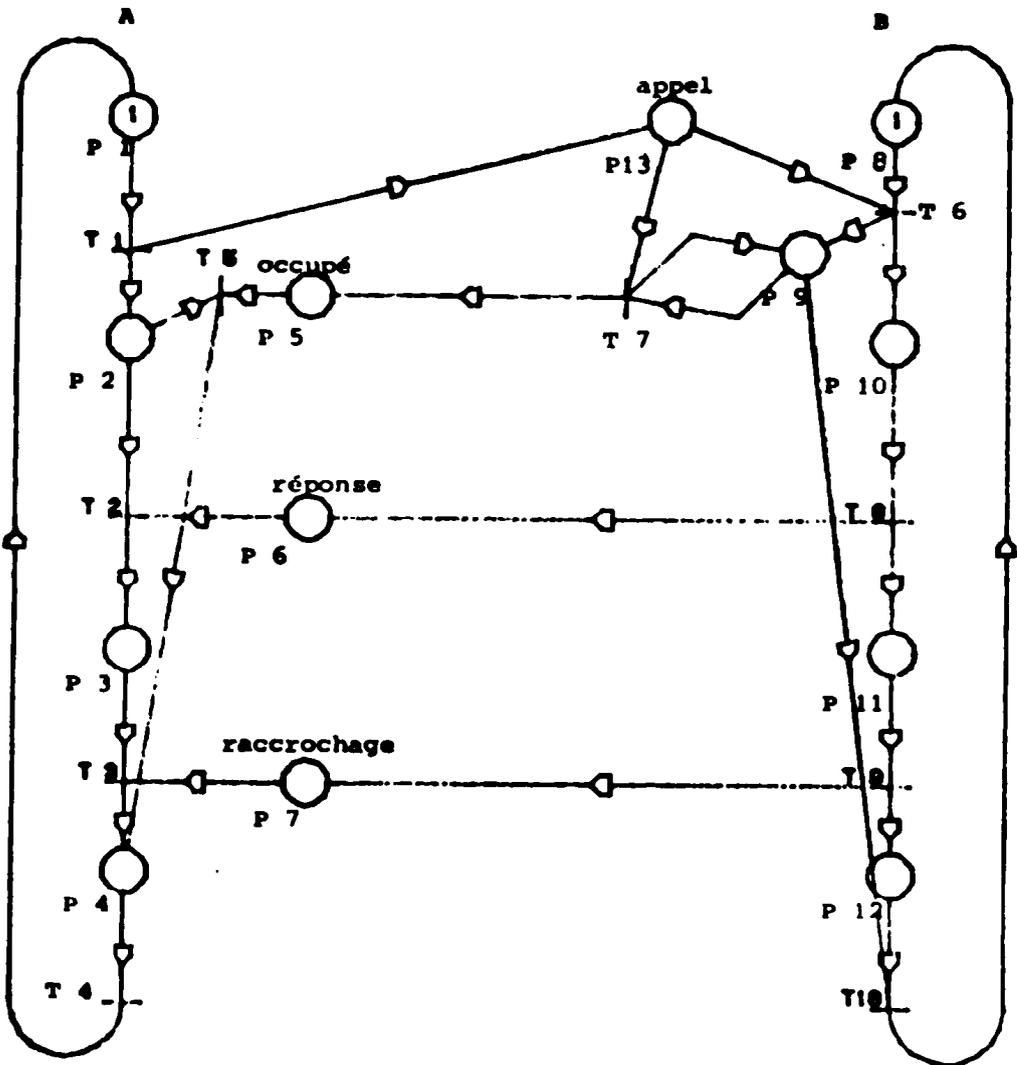


Fig. III.2.b

"occupé", soit son accord pour l'établissement de la communication "réponse". Après la réception de "réponse", les deux processus sont connectés (places P 3, P 10). Lorsqu'il le désire, le demandeur envoie "fin" et redevient oisif ; ensuite lorsque le demandé reçoit "fin", il devient oisif.

Ce mode de protocole qui est modélisé sur la figure III.2.a, peut alors être vérifié par OGIVE : le réseau qui correspond à ce mode est borné, sauf, réinitialisable et vivant.

#### III.4.2 MAITRE AU DEBUT ET ESCLAVE EN FIN DE COMMUNICATION

La différence avec le cas précédent est la suivante : le processus qui est maître à l'établissement devient esclave en ce qui concerne la rupture. Dans ce cas le message de demande de rupture qu'il reçoit du processus distant est appelé "raccrochage".

La figure III.2.b montre le réseau de Pétri qui représente ce mode. Le demandeur envoie un "appel", le demandé lui déclare soit son occupation, soit son accord ; dans ce dernier cas, il y a "réponse" et les deux processus sont connectés (P 3, P 11). Si le demandé envoie un "raccrochage", il devient oisif et à la réception de ce message, le demandeur lui aussi devient oisif.

L'analyse de ce réseau de la figure III.2.b prouve la validité de ce modèle : le réseau est borné, sauf, réinitialisable et vivant.

#### III.4.3 MODE ASYNCHRONE EN FIN DE COMMUNICATION

Ce mode de communication signifie que les deux entités communicantes peuvent toutes les deux envoyer le message de rupture. Une représentation de ce mode est donnée par la figure III.2.c ; comme dans le paragraphe III.4.1, après l'établissement de la communication la phase de conversation débute (P 3, P 11).

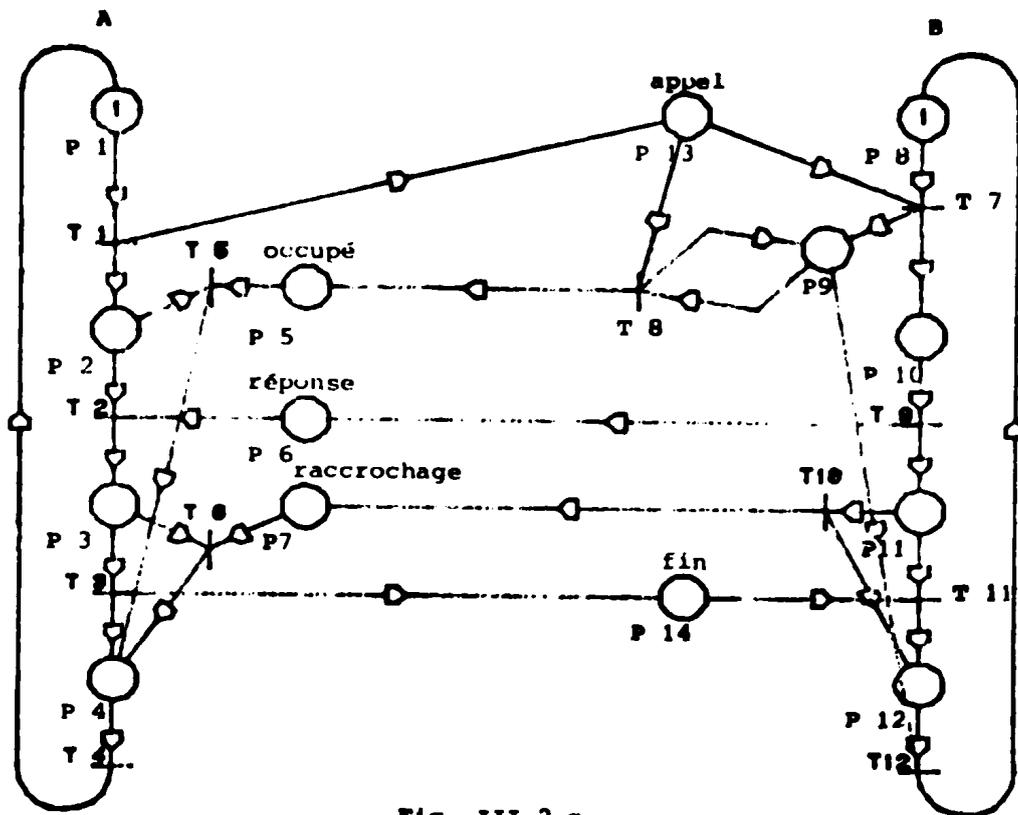


Fig. III.2.c

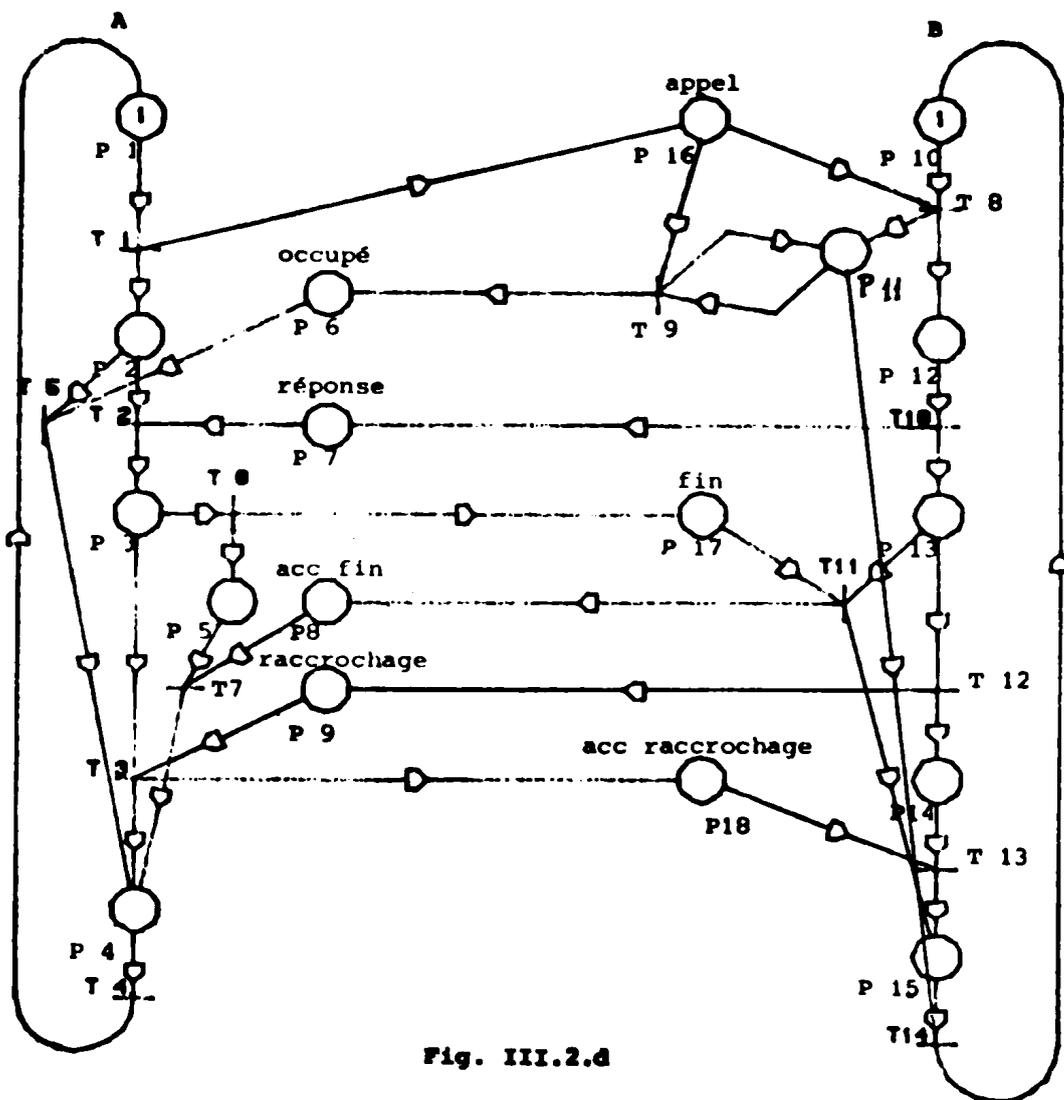


Fig. III.2.d

Il peut arriver que, simultanément, le demandeur envoie un message "fin" (P 14) et le demandé envoie un message "raccrochage" (P 7).

L'analyse montre que le réseau est non borné car le marquage (P 4, P 7, P 12, P 14) est tel que T 4 et T 12 sont tirables et il reste P 12 et P 14 marquées ; cela provient du croisement des messages "fin" et "raccrochage" transmis en parallèle : il y aura accumulation de jetons, ce qui conduira à une possibilité de mauvais fonctionnement.

Une manière de résoudre ce problème consiste à définir un message d'accusé de réception pour chaque message de demande de rupture. Ainsi nous définissons :

- "acc fin" : accusé de réception du message "fin" ;
- "acc raccrochage" : accusé de réception du message "raccrochage".

#### III.4.4 MODE ASYNCHRONE EN FIN DE COMMUNICATION AVEC ACCUSE DE RECEPTION

Ce mode est décrit par le réseau de Pétri de la figure III.2.d. Mais dans ce cas, l'analyse montre que le réseau est non vivant (marquage P 5, P 9, P 11, P 14, P 17). Le blocage provient du croisement. Afin de débloquent le réseau, il est nécessaire de considérer que si le demandé reçoit "fin" après avoir envoyé son "raccrochage", le demandé doit consommer ce message "fin" dans l'état d'attente de l'accusé de réception du message "raccrochage" ; il en est de même du côté du demandeur. Dans la figure III.2.e, les transitions T 8 et T 15 sont utilisées dans ce but, pour le déblocage ; le réseau, donné par la figure III.2.e, est analysé et possède les caractéristiques satisfaisantes : il est borné, sauf, réinitialisable et vivant.

#### III.4.5 MODE ASYNCHRONE EN FIN DE COMMUNICATION AVEC ACCUSE DE RECEPTION UTILISANT UN NOMBRE OPTIMAL (MINIMAL) DE MESSAGES

Dans le protocole représenté dans la figure III.2.e, nous avons utilisé quatre messages pour la rupture de la communication (place P 17 :

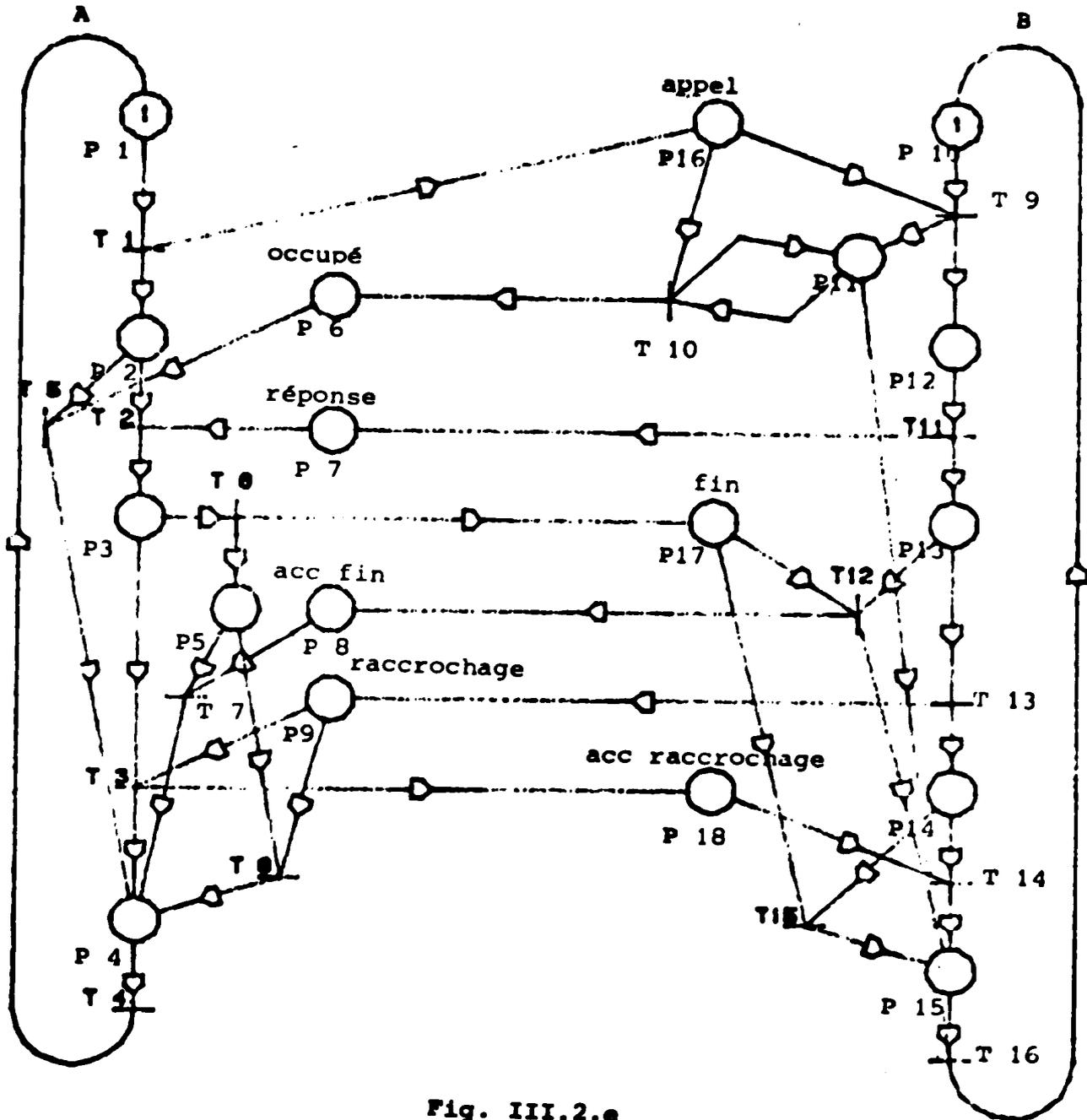


Fig. III.2.e

fin, place P 8 : acc fin, place P 9 : raccrochage, place P 18 : acc raccrochage) ; en réalité, on peut remarquer qu'il est possible de réduire d'une unité le nombre de ces places en assurant le même fonctionnement. Ces protocoles optimaux sont donnés par les figures III.2.f, III.2.g.

Dans le protocole figure III.2.f, la place P 17 qui représente la transmission du message "fin" peut être utilisée soit comme un message venant du demandeur pour rompre la communication (transition T 6), soit comme la transmission du message accusé de réception de "raccrochage" du demandé (transition T 3). Le déroulement de ce protocole est tout à fait correct (le réseau est : borné, sauf, réinitialisable et vivant).

Dans le protocole figure III.2.g, de façon similaire, la place P 8 qui représente le raccrochage peut être utilisée pour représenter soit l'émission venant du demandé pour rompre la communication (transition T 12) soit la transmission de l'accusé de réception de "fin" (transition T 11). Ce protocole a lui aussi un fonctionnement correct (le réseau est borné, sauf, réinitialisable et vivant).

Le réseau est maintenant plus difficile à lire compte tenu des places qui ont des significations multiples.

Nous avons jusqu'ici considéré le problème dû à l'asynchronisme survenant lors de la rupture de la communication. Avant d'aller plus avant dans les spécifications du comportement de l'abonné nous devons également considérer le cas de l'asynchronisme dans l'établissement de la communication.

#### III.4.6 MODE ASYNCHRONE EN DEBUT DE COMMUNICATION : APPELS CROISES

En effet, dans la réalité, les entités désirant se connecter peuvent s'appeler les unes les autres, simultanément : cette situation provoquera un croisement d'appel. Le but de ce paragraphe est d'analyser l'effet résultant.

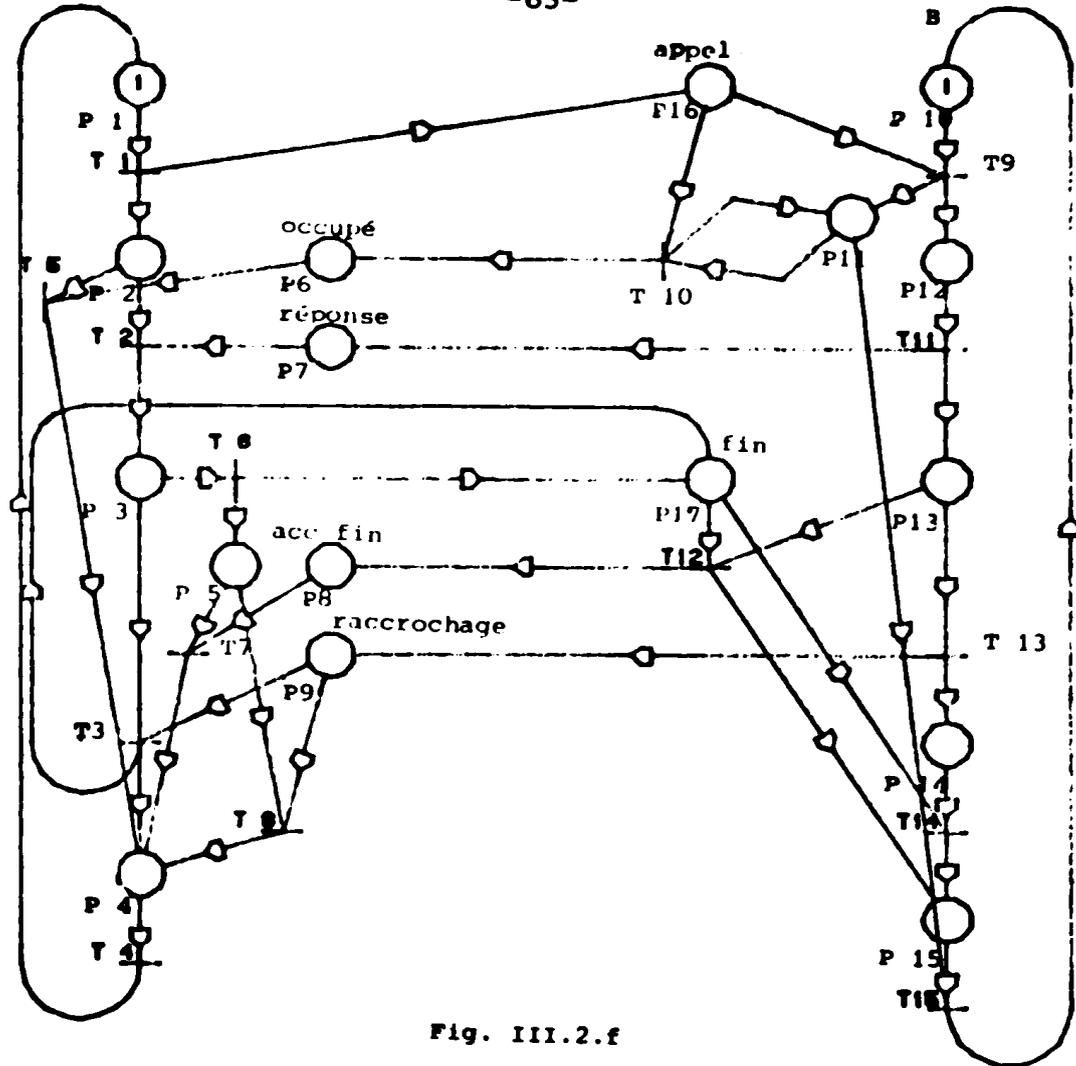


Fig. III.2.f

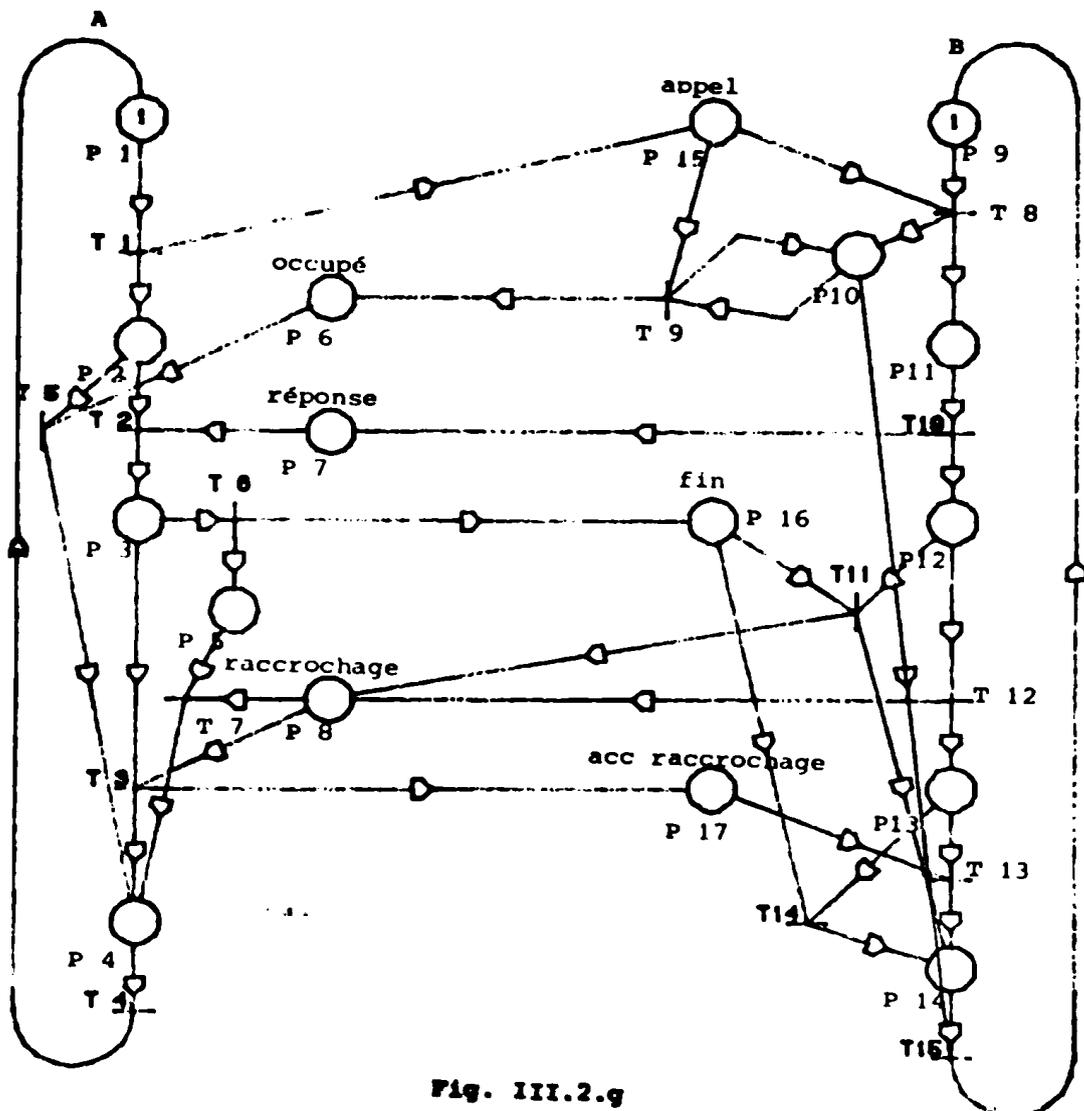


Fig. III.2.g

Le réseau qui illustre cette situation est donné par la figure III.3 ; les deux processus A, B sont reliés par des messages de communication ; ces processus peuvent être décomposés en deux sous-processus, une partie agissant en tant que partie établissant la connexion (partie gauche du processus), une partie agissant en tant que partie recevant la demande d'établissement (partie droite du processus).

Ainsi, les séquences des états des deux processus A et B peuvent se décomposer en trois phases :

1. La phase d'établissement : A (P 3, P 6) ; B (P 14, P 17) ;
2. La phase de connexion : A (P 4, P 7) ; B (P 15, P 18) ;
3. La phase de rupture : A (P 5, P 8) ; B (P 16, P 13).

Les messages émis par les abonnés ("appel", "occupé" et "réponse"), durant la phase d'établissement que nous considérons, sont représentés par les places suivantes :

- les places P 20, P 21, P 22 pour les messages émis par A vers B ;
- les places P 9, P 10, P 11 pour les messages émis par B vers A.

Ce réseau de la figure III.3 est analysé : il est montré borné, sauf, réinitialisable et vivant.

On peut en conclure que le croisement des messages d'appel n'induit pas de problèmes lors de l'analyse du réseau. Ceci est dû au fait que lorsqu'un abonné émet un appel, il devient occupé (P 2 pour A, P 13 pour B), et que si un appel arrive, il émet un message "occupé" à son correspondant. On peut ainsi noter que dans le cas d'appels croisés, la communication ne peut pas être établie. Nous pouvons également noter dans la figure III.3 que si l'appel venant d'un correspondant arrive à l'autre avant l'émission de son propre appel (c'est-à-dire dans l'état oisif P 1 pour A, P 12 pour B), le demandé émet normalement un message "réponse".

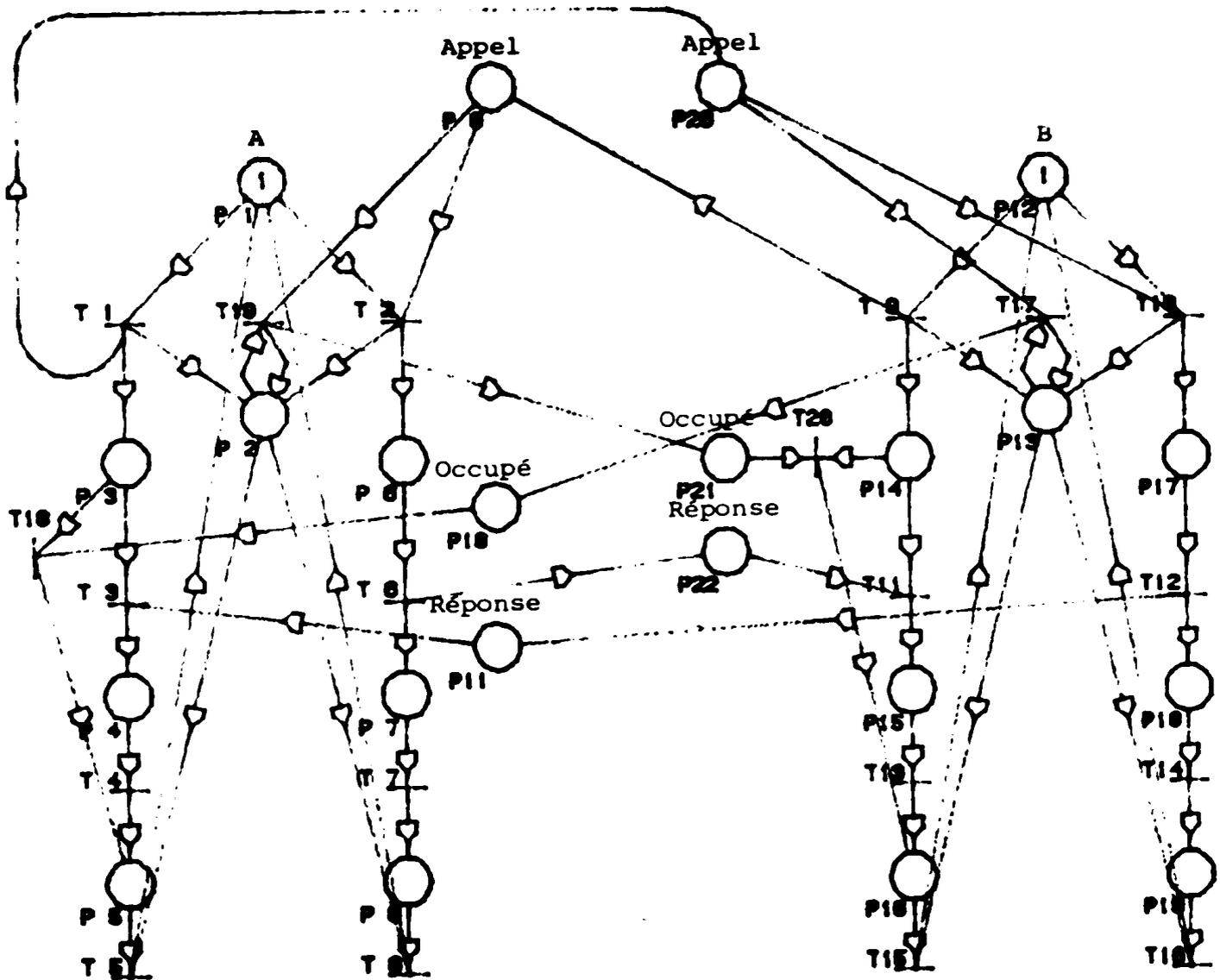


Fig. III.3 Modèle de croisement d'appel

### III.5 DEUXIEME MODELISATION DU COMPORTEMENT D'UN ABONNE

Nous allons spécifier maintenant le comportement d'un abonné à un niveau de détail plus bas que celui donné par la figure II.3.d ; ceci sera effectué en précisant la procédure Appelant et la procédure Appelé ; la spécification de ce nouveau niveau fera apparaître des messages qui doivent être échangés entre la procédure appelant et la procédure appelé pour assurer la communication. Afin de simplifier, nous ne considérons pas ici les niveaux inférieurs (2, 3, ...). Le modèle du niveau 1 est décrit par la spécification informelle suivante :

#### A. Texte de la procédure Appelant

1. Après que l'appelant ait envoyé un appel, il attend de recevoir l'état de l'appelé (l'appelé sera soit libre, soit occupé).
2. L'appelant étant en attente de l'état de l'appelé, si l'appelant physique raccroche, l'appelant envoie un message FIN à l'appelé et passe à l'état libération appelant.
3. Si l'appelant, en attente de l'état de l'appelé, reçoit l'état appelé occupé, il passe à l'état libération appelant.
4. Si l'appelant, en attente de l'état de l'appelé, reçoit l'état appelé libre, il envoie une tonalité de retour d'appel à l'appelant physique et attend de recevoir une REPONSE de l'appelé.
5. L'appelant étant en attente de la REPONSE, si l'appelant physique raccroche, l'appelant passe à l'état libération appelant et envoie un message FIN à l'appelé.
6. Si l'appelant, en attente de la REPONSE de l'appelé, la reçoit, ceci signifie que l'appelé physique a décroché, l'appelant arrête la tonalité de retour d'appel et passe dans l'état de conversation.

7. L'appelant étant en conversation, si l'appelant physique raccroche, alors l'appelant passe à l'état libération appelant et envoie le message FIN à l'appelé.
8. Si l'appelant en conversation reçoit un message RAC venant de l'appelé, ceci signifie que l'appelé physique a raccroché et l'appelant passe à l'état libération appelant. Il avertit éventuellement l'appelant physique.
9. Si l'appelant est dans l'état libération appelant, la procédure Appelant est terminée.

**B. Texte de la procédure Appelé**

1. Après réception de l'appel, l'appelé étant libre commence la procédure Appelé en passant dans l'état de réception d'appel accepté.
2. Après acceptation de la réception d'appel, l'appelé envoie le message Appelé libre à l'appelant et le courant d'appel à l'appelé physique et il passe à l'état d'attente du décrochage de l'appelé physique.
3. Après le décrochage de l'appelé physique, l'appelé envoie une REPONSE à l'appelant et passe à l'état de conversation.
4. Lorsqu'il est en attente de décrochage, l'appelé peut recevoir le message FIN venant de l'appelant ; il passe alors à l'état libération appelé.
5. L'appelé étant en conversation, si l'appelé physique raccroche, il envoie le message RAC à l'appelant et passe à l'état libération appelé.
6. Si l'appelé en conversation reçoit le message FIN de l'appelant, il passe à l'état libération appelé et éventuellement signale ceci à l'appelé physique.

7. Si l'appelé est dans l'état libération appelé, la procédure Appelé est terminée.

Si nous considérons maintenant les différentes classes d'objets pour ce niveau 1, alors :

$$D1 = \langle E1, A1, P1, I1, F1 \rangle$$

Le classement que nous proposons est le suivant :

P1 :  $\langle$  attente de l'état de l'appelé, attente de la REPONSE de l'appelé, appelant en conversation, libération appelant, état de réception d'appel accepté, attente du décrochage de l'appelé physique, appelé en conversation, libération appelé  $\rangle$

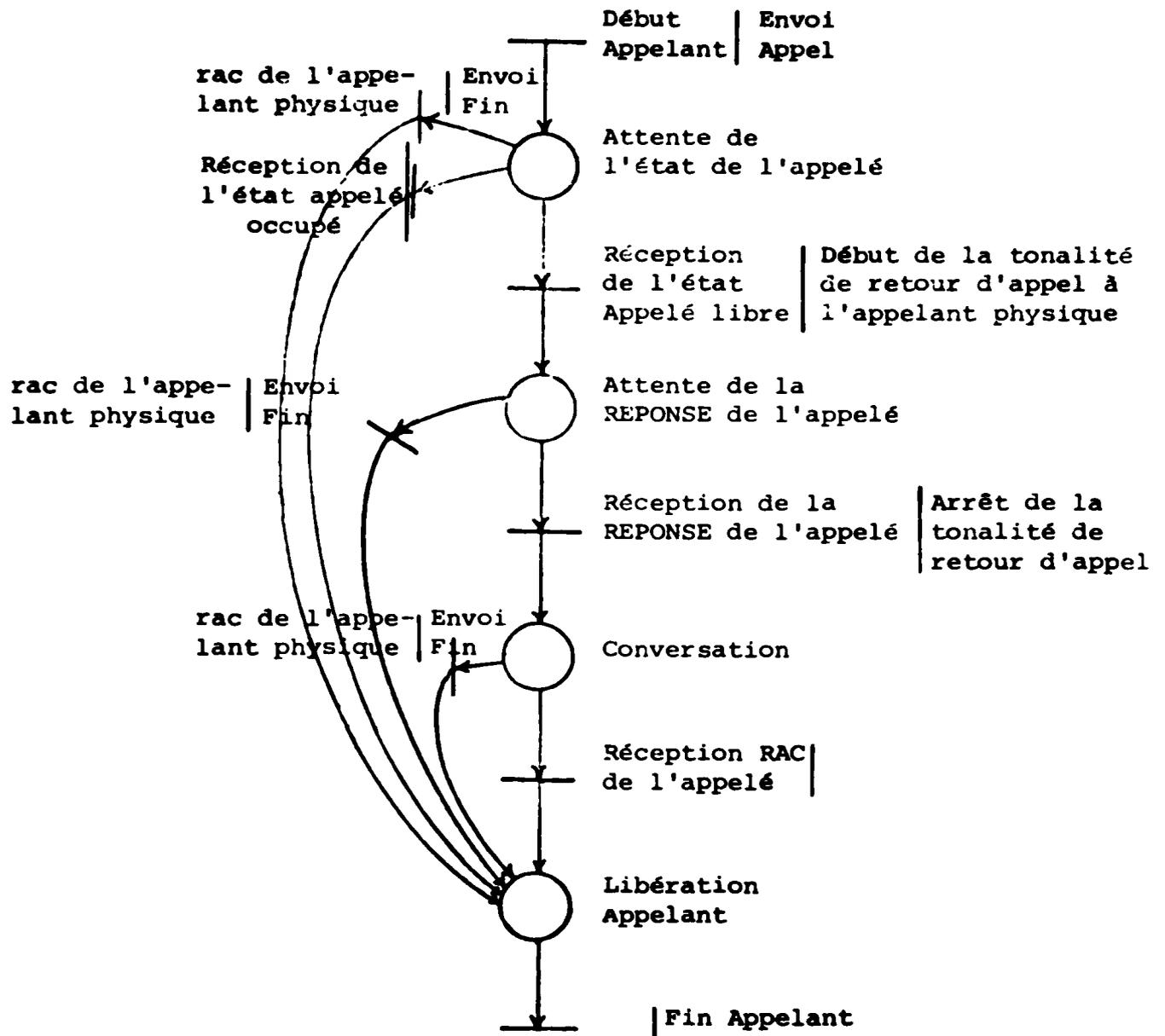
E1 :  $\langle$  raccrochage de l'appelant physique, réception de l'état appelé occupé, réception de l'état appelé libre, réception de la REPONSE de l'appelé, réception du RAC de l'appelé, décrochage de l'appelé physique, réception de FIN de l'appelant, raccrochage de l'appelé physique  $\rangle$

A1 :  $\langle$  envoi de FIN par l'appelant, début de la tonalité de retour d'appel à l'appelant physique, arrêt de la tonalité de retour d'appel, envoi du message appelé libre, début du courant d'appel à l'appelé physique, envoi de la REPONSE par l'appelé, arrêt du courant d'appel, envoi du RAC par l'appelé  $\rangle$

I1, F1 :  $\langle$  non considérés ici (contiendrait la gestion de la numérotation, la commande du réseau de connexion par exemple)  $\rangle$

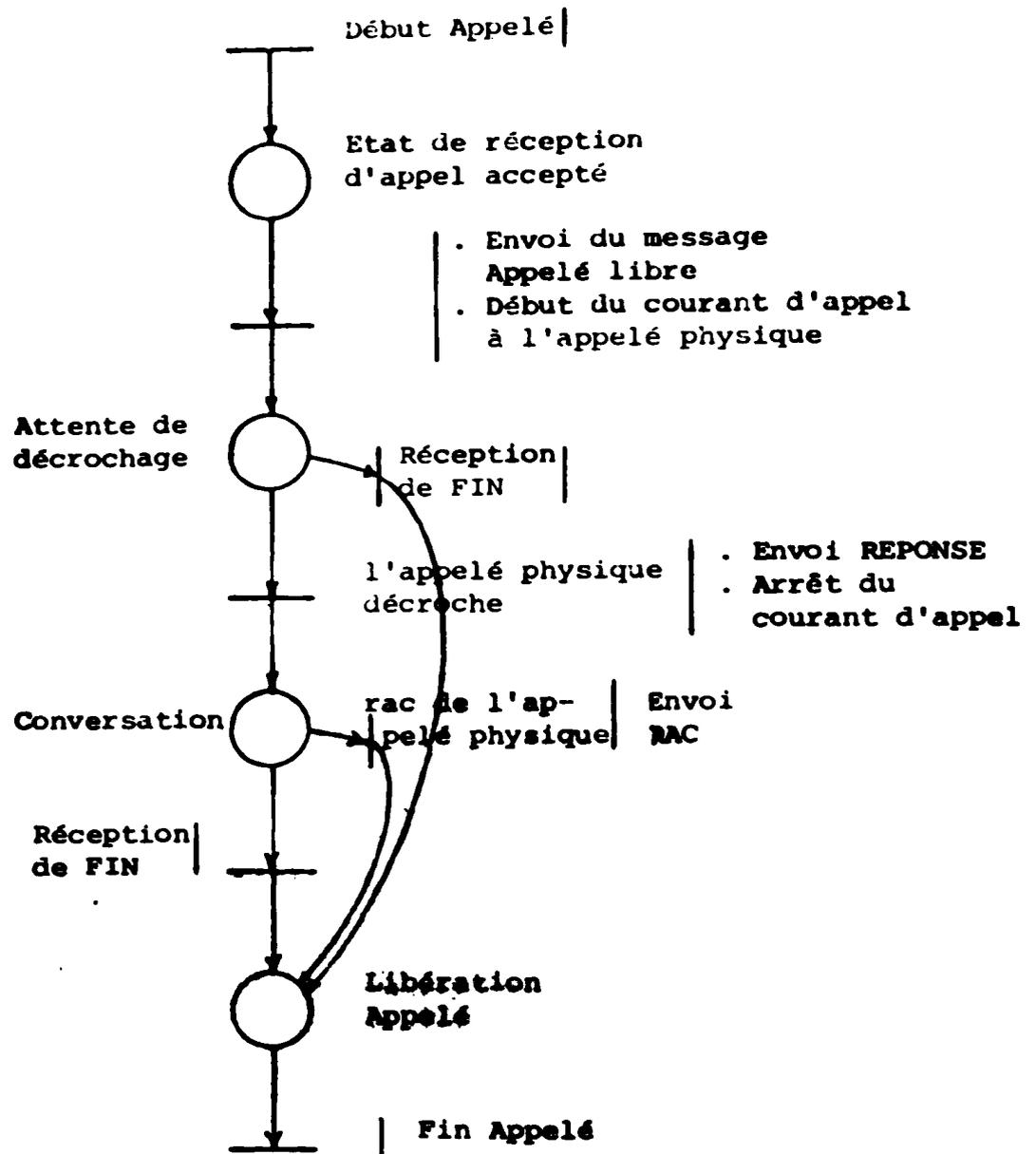
La phase de formalisation conduit aux réseaux de la figure III.4.

L'étude de ce deuxième niveau nous permet de constater que les interactions entre les procédures Appelant et Appelé sont très importantes. Donc, il apparaît la nécessité d'explicitier les protocoles existant dans ce niveau entre les abonnés. Ce problème a été abordé depuis longtemps



- rac** : raccrochage
- REPONSE** : message émis par l'appelé pour la connexion avec l'appelant
- RAC** : message émis par l'appelé pour rompre la communication
- FIN** : message émis par l'appelant pour rompre la communication

Fig. III.4a Procédure Appelant



- rac** : raccrochage
- REPONSE** : message émis par l'appelé pour la connexion avec l'appelant
- RAC** : message émis par l'appelé pour rompre la communication
- FIN** : message émis par l'appelant pour rompre la communication

Fig. III.4b Procédure Appelé

bien sûr par le CCITT et, parmi tous les systèmes de signalisation recommandés par cet organisme, nous choisirons ceux qui sont les plus récents et qui nous paraissent les plus intéressants, les recommandations CCITT N° 6,7, que nous allons maintenant expliciter puis modéliser.

### III.6 MESSAGES FONCTIONNELS DE SIGNALISATION INSPIRES DES RECOMMANDATIONS CCITT N° 6,7

Le CCITT a déjà proposé un certain nombre de systèmes de signalisation tels que, par exemple, les systèmes 5, R2, 6, 7 [26, 27, 28, 29] ; ces systèmes définissent les signaux qui peuvent être utilisés pour l'établissement ou la rupture des communications entre abonnés. Parmi ces systèmes, nous avons choisi d'utiliser ici les systèmes CCITT N° 6,7 qui sont les plus récents et les plus évolués. Ces protocoles seront pris en tant que base de travail pour définir des classes fonctionnelles représentant les messages échangés entre les centraux communicants. Cette classification est effectuée dans le but de simplifier d'une part la compréhension et d'autre part la modélisation que nous effectuons ensuite.

Nous avons sélectionné les classes suivantes pour les messages fonctionnels :

#### 1. Message appel (APPEL)

C'est un message avant (allant vers le central suivant) qui demande une connexion avec un correspondant ; il peut contenir les informations suivantes :

- (a) l'adresse (numéro demandé, ...)
- (b) l'indicatif (du pays, du département, ...)
- (c) la nature du circuit utilisé (par satellite ou non, ...)
- (d) l'insertion d'un supprimeur d'écho
- (e) la catégorie du correspondant : l'appelé est-il une opératrice ou un abonné ? Quelle est la priorité de l'appelé ? Est-ce un appel d'essai ? ...

- (f) un indicateur de fin de numérotation (ST) pour indiquer la fin de l'adresse
- (g) la nature des interventions
- (h) des indications sur la continuité de la voie de conversation dans la signalisation sémaphore

## 2. Message encombrement (ENCOMBREMENT)

C'est un message arrière (venant du central suivant et allant vers le central précédent) qui peut inclure :

- (a) des informations d'encombrement des équipements qui indiquent par exemple que l'échec de l'établissement de la communication est dû à l'encombrement de l'équipement de communication et qui donnent un certain nombre de causes
- (b) des informations sur l'encombrement du faisceau des circuits, qui indique que l'échec de l'établissement de la communication survient à la suite de l'encombrement du faisceau des circuits
- (c) des informations d'encombrement sur le réseau national indiquant que l'échec d'établissement provient d'un encombrement du réseau national

## 3. Message accept (ACCEPT)

C'est un message arrière qui, selon les cas, peut indiquer que :

- (a) le numéro est complet et avec taxation
- (b) le numéro est complet et sans taxation
- (c) le numéro est complet et correspond à un poste à prépaiement
- (d) le numéro est complet, la ligne d'abonné est libre et est avec taxation
- (e) le numéro est complet, la ligne d'abonné est libre et est sans taxation
- (f) le numéro est complet, la ligne d'abonné est libre et possède un poste à prépaiement

4. Message réponse (REPONSE)

C'est un message arrière qui peut contenir :

- (a) un signal de réponse, avec taxation, qui indique que l'appelé a répondu et que la communication doit être taxée ; ce signal est utilisé pour provoquer :
  - (i) le début de la taxation de l'appelant
  - (ii) le début de la mesure de la durée de conversation
- (b) un signal de réponse, sans taxation, qui indique que l'appelé a répondu et que la communication ne doit pas être taxée.

5. Message hors d'atteinte (HORS D'ATTEINTE)

C'est un message arrière qui peut signifier que :

- (a) le numéro est incomplet
- (b) le numéro s'avère non utilisé
- (c) la ligne est hors service ou en dérangement
- (d) l'abonné a été transféré (changement de numéro), ce qui indique que le numéro reçu a cessé d'être utilisé et que l'abonné auquel il était attribué doit être atteint avec un autre numéro
- (e) un échec d'appel a été constaté, ce qui indique l'échec d'établissement de la communication dû soit à aucun signal déterminé soit à la fin d'un délai de temporisation.

6. Message occupé (OCCUPE)

C'est un message arrière qui indique l'occupation de l'appelé.

7. Message fin (FIN)

C'est un message avant qui met fin à une communication ou à une demande de communication et qui a pour but de libérer le circuit utilisé ; ce message est normalement émis quand l'appelant raccroche.

8. Message libération de garde (LBRGARD)

C'est un message arrière qui, normalement, acquitte la réception par l'appelé du message FIN émis par l'appelant.

9. Message raccrochage (RAC)

C'est un message arrière qui indique à l'appelant que l'appelé a raccroché.

10. Message acheminement (ACHEMINEMENT)

C'est un message arrière qui accuse la réception d'un appel et indique que l'appel est en cours de propagation.

Cette liste de classes de messages fonctionnels donne les principales familles utilisées ; ces messages doivent être échangés entre deux abonnés communicants et donc en particulier pouvoir être explicités et apparaître dans la connexion de deux abonnés décrits chacun par un réseau identique à celui de la figure III.5. Ce réseau représente maintenant l'abonné complet, c'est-à-dire en incluant dans un même réseau les niveaux 0 et 1. Il faut noter que notre choix d'utiliser ces messages a été fait compte tenu des travaux déjà faits par les participants au groupe du CCITT. Notre but est de les utiliser comme base de départ pour effectuer une modélisation formelle par réseau de Pétri de différents protocoles possibles permettant d'établir et de rompre une communication entre deux abonnés. Nous verrons en particulier que cette modélisation et les analyses que nous effectuerons nous conduiront à proposer quelques aménagements améliorant, à notre avis, l'expression de la recommandation.

### III.7 MODELES DE PROTOCOLES DE COMMUNICATION

Considérons d'abord les abonnés représentés comme indiqué par la figure III.5 et les classes données précédemment pour les signaux. Il est alors possible d'établir une correspondance entre les actions de la figure III.5 et les messages des recommandations CCITT. De façon évidente, cette correspondance n'est pas bi-univoque et nous étudierons de ce fait plusieurs correspondances (dépendant du nombre de messages pris

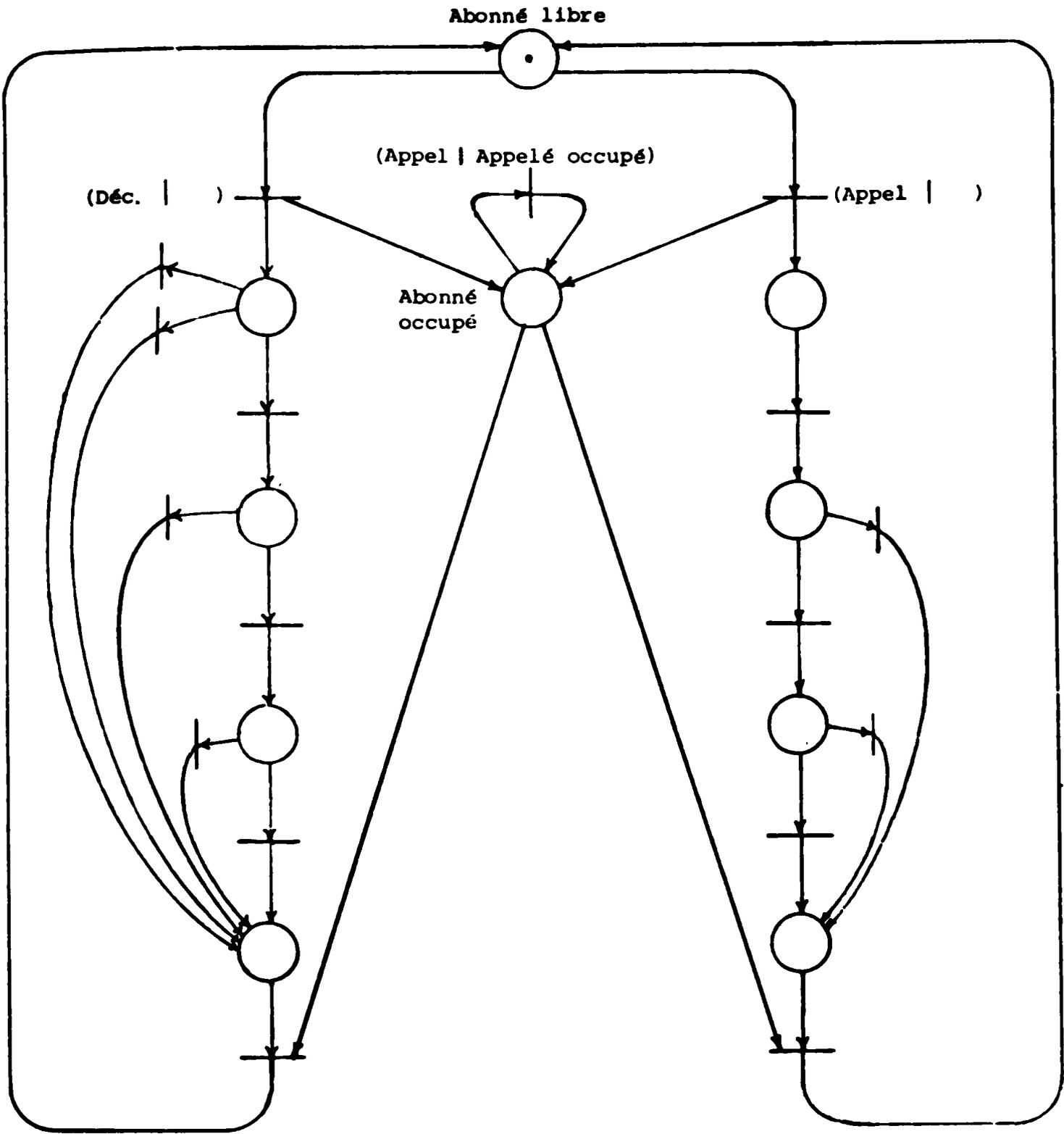


Fig. III.5 Modèle de l'abonné :  
premier et deuxième niveaux

en compte et transitant entre les abonnés...). Par exemple, nous ne tiendrons pas compte au début des classes "ENCOMBREMENT", "HORS D'ATTEINTE" et "ACHEMINEMENT" car elles ne sont pas liées à l'abonné lui-même mais au matériel existant entre les abonnés.

De plus, dans un esprit de simplification, en remarquant que dans une connexion il y a un abonné appelant et un abonné appelé, nous ne dessinerons pas entièrement l'abonné tel qu'il est représenté sur la figure III.5 mais uniquement la partie appelant (gauche) pour l'appelant et la partie appelé (droite) pour l'appelé, la justification de cette simplification a été donnée au paragraphe III.4.

Ces différentes hypothèses nous conduisent au premier réseau appelé modèle non optimal.

### III.7.1 MODELE DU PROTOCOLE NON OPTIMAL

Le protocole décrit dans la figure III.6 contient les signaux venant des abonnés et utilisés dans les systèmes de signalisation envisagés. Il est possible de remarquer que ce protocole est non optimal dans le sens que des signaux fonctionnellement identiques ont des codes (mnémoniques) différents selon l'état du système au moment de leur émission ; par exemple les messages FIN 1 et FIN 2 ont la même fonction (demande de rupture de la communication) mais FIN 1 est émis dans l'état avant la mise en conversation et FIN 2 dans l'état conversation.

Notons que :

a) L'appelant et l'appelé sont oisifs à l'instant initial

Ces états sont représentés respectivement par le marquage des places P1 et P14.

b) L'appelant envoie APPEL

Lorsque l'appelant décroche (transition T1 tirée), un message APPEL est envoyé (place P21 marquée) et l'appelant attend le retour de l'état de l'appelé (dans P2) ; bien sûr l'appelant sera

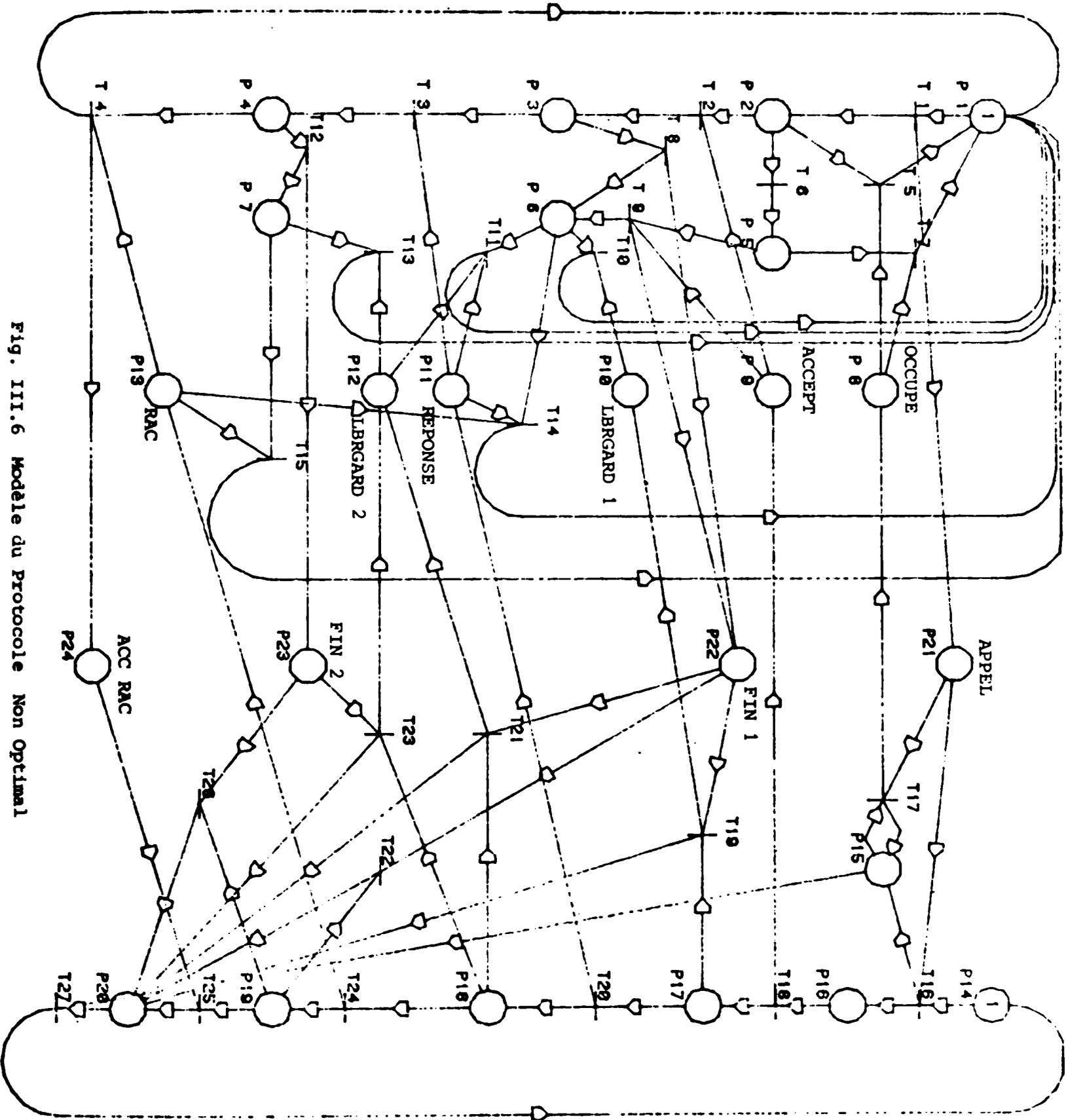


Fig. III.6 Modèle du Protocole Non Optimal

mis dans l'état occupé mais cet état n'apparaît pas dans le réseau donné ici : son rôle sera explicité dans le cas de l'appelé.

c) L'appelé reçoit APPEL

Si l'appelé, dans l'état oisif (place P14 marquée), reçoit un APPEL (T16 est tiré), il devient occupé (place P15 marquée), passe dans l'état appel accepté et la procédure Appelé débute (P16 marquée) ; l'appelé élabore le message ACCEPT et le transmet (Transition T18 tirée, place P9 marquée). Si maintenant l'appel arrive lorsque l'appelé est occupé (P15 marquée), l'appelé renvoie le message OCCUPE (Transition T17 tirée) et l'appelé reste occupé (place P15).

Notons ici que nous avons choisi pour la modélisation la représentation des interactions par échange classique de messages, c'est-à-dire par utilisation d'une place partagée.

d) L'appelant reçoit OCCUPE

Le message OCCUPE est reçu, en attente de l'état de l'appelé (P2 marquée) alors (T5 est tirée) l'appelant redevient oisif. Si avant la réception du message OCCUPE l'appelant a rompu la communication (T6 tirée), l'appelant passe en attente de retour d'appel (place P5) ; si le retour est constitué par le message OCCUPE (T7 est tirée), l'appelant devient à nouveau oisif.

e) L'appelant reçoit ACCEPT

Si le message ACCEPT est reçu, en attente de l'état de l'appelé (T2 est tirée) l'appelant passe (dans la place P3) en attente de la réponse. Au contraire, si ACCEPT est reçu pendant l'attente du retour d'appel (P5 marquée), alors (T9 est tirée) l'appelant envoie le message FIN 1 au demandé (place P22 marquée) et l'appelant ira dans l'état d'attente d'un message de libération (place P6 marquée). Par ailleurs, pendant l'attente de la réponse (P3), l'appelant peut rompre la communication avant la réception du message réponse (T8 est tirée), engendrant le message FIN 1 (place P22) et l'appelant passe dans la place P6 en attente d'un message de libération.

Envisageons maintenant le comportement du système à partir de ces deux états : l'état P6 d'attente de libération et l'état P3 d'attente de la réponse.

α) Appelant dans l'état d'attente de libération (Place P6 marquée)

Après avoir envoyé le message FIN 1 au demandé (place P22), trois possibilités peuvent exister :

1. L'appelé est dans l'état d'élaboration de la réponse (place P17 marquée)

Lorsque l'appelé reçoit FIN 1 (T19 est tirée), il envoie un message LBRGARD 1 à l'appelant et passe à l'état de libération (marques en P10 et en P20); si P10 est marquée et P6 est déjà marquée (T10 est tirée), l'appelant revient à l'état oisif.

2. L'appelé envoie la réponse (T20 est tirée, envoyant un jeton dans la place P11) et passe dans l'état conversation (place P18)

Si le message REPONSE arrive en P11, l'appelant étant en P6, nous trouvons que les transitions T11 et T14 ne sont pas tirables ; on attend soit LBRGARD 2 soit RAC .

Si l'appelé en conversation reçoit FIN 1 (T21 tirée), il envoie LBRGARD 2 (en P12) et passe en libération (P20).

L'appelant en attente de libération a reçu REPONSE puis LBRGARD 2 (places P6, P11, P12 marquées), sa rupture a été bien traitée et il devient oisif (T11 tirée). Lorsque l'abonné appelé est dans l'état de libération (P15, P20 marquées), il n'est plus occupé et devient oisif (T27 tirée).

Si l'appelé rompt la communication (T24 est tirée), il envoie un message RAC (en P13) et passe à l'état d'attente d'accusé (P19). Maintenant l'appelant en attente de libération a reçu REPONSE puis RAC (P6, P11, P13 sont marquées) ; cet appelant peut devenir oisif (T14 tirée). Si l'appelé en attente d'accusé reçoit FIN 1 (T22 est tirée), il passe en libération (P20).

**β) Appellant dans l'état d'attente de réception de la réponse**  
**(place P3 marquée)**

Lorsque l'appelé envoie REPONSE (T20 tirée) l'appelé passe en conversation (P18). Lorsque REPONSE (P11) est reçue par l'appelant en attente de réponse, il passe en conversation (la transition T3 est tirée et l'état de l'appelant passe en P4). Les places P4 et P18 marquées représentent l'état de conversation.

**f) L'appelant et l'appelé en conversation (P4, P18 marquées)**

Trois possibilités doivent être envisagées :

**1. L'appelant rompt la communication**

(T12 est tirée) et il envoie un message FIN 2 à l'appelé (place P23 marquée) et passe dans l'état (P7) d'attente de libération. Lorsque l'appelé reçoit FIN 2 (la transition T23 est tirée), le message LBRGARD 2 est envoyé (place P12) et l'appelé passe en libération. Quand l'appelant en attente de libération reçoit LBRGARD 2 (places P7 et P12 marquées), l'appelant redevient oisif (T13 tirée).

**2. L'appelé rompt la communication**

Quand l'appelé est en conversation (P18), il peut rompre la communication en raccrochant (T24 est tirée) et en envoyant un message RAC (place P13) ; l'appelé passe alors en état d'attente d'accusé ou de fin de l'appelant (place P19). A la réception du message RAC (transition T4 tirée), l'appelant envoie un accusé de raccrochage ACC RAC (place P24) à l'appelé et devient oisif. A la réception de l'accusé de raccrochage (transition T25 tirée), l'appelé passe dans l'état fin appelé (P20).

**3. L'appelant et l'appelé rompent simultanément la communication**

Il peut arriver que l'appelant envoie FIN 2 et l'appelé envoie RAC avant que chacun d'eux ne reçoive le message de l'autre. En expliquant ce cas, nous trouvons que si l'appelant envoie

FIN 2 (place P23), il attend la libération (P7). Si dans cet état RAC est reçu (place P13), l'appelant sait que l'appelé a raccroché et donc redevient oisif (T 15 tirée). Si l'appelé a envoyé RAC , il attend un accusé (P19) : lorsque FIN 2 est reçu (place P23), il sait que l'appelant a raccroché et donc passe en libération (T26 tirée).

Ce fonctionnement avec croisement de messages semble donc correct.

### Vérification du Protocole

Il est maintenant possible d'envisager l'étude des propriétés du modèle proposé.

L'analyse de ses propriétés a été effectuée en utilisant l'outil OGIVE dont nous avons déjà parlé. L'analyse montre que ce protocole possède les propriétés requises : il est borné, sauf, réinitialisable et vivant.

En conséquence, ce protocole est logiquement correct pour les analyses effectuées.

### III.7.2 MODELE DU PROTOCOLE NON OPTIMAL A NOMBRE REDUIT DE MESSAGES

Il est souvent plus intéressant de réduire le nombre de messages en maintenant le même fonctionnement. Dans le protocole donné dans la figure III.6, nous avons utilisé deux messages FIN 1, FIN 2 pour transmettre la fonction de rupture par raccrochage de l'appelant selon que la rupture est effectuée avant ou après la réception de la réponse venant de l'appelé ; de façon similaire, nous avons utilisé deux messages LBRGARD 1, LBRGARD 2 pour accuser la réception du message de rupture.

En réalité, on peut envisager de n'utiliser qu'un seul message FIN qui remplacera FIN 1, FIN 2 ; par ailleurs, on essaiera d'utiliser un seul message LBRGARD en affectant au message RAC une double fonction : d'abord sa propre fonction de raccrochage d'appelé et ensuite une fonction remplaçant celle de LBRGARD 2 ; ce protocole simplifié est possible et il est donné par la figure III.7.

Nous allons ici décrire brièvement ce protocole optimisé partiellement par réduction du nombre des messages. Nous précisons uniquement les différences qui existent par rapport au protocole précédent.

a) L'appelant est dans l'état d'attente de libération (P6)

(i) Supposons que l'appelé reçoive FIN quand il était dans l'état P16 d'élaboration de la réponse (la transition T17 est tirée), l'appelé envoie le message LBRGARD et passe dans l'état fin d'appelé.

En conséquence, (T10 est tirable), l'appelant recevant LBRGARD redevient oisif.

(ii) Supposons que l'appelé envoie REPONSE avant la réception de FIN, il est alors dans l'état conversation (P17).

Il y a alors deux possibilités pour l'appelé :

\* il reçoit le message FIN (T19 est tirable), ce qui provoque la transmission d'un accusé de réception de FIN: RAC, et la fin de la procédure Appelé. Du côté appelant, à la réception de RAC (T12 est tirable), l'appelé consomme les deux messages REPONSE et RAC, et devient oisif.

\* il raccroche (T20 est tirée), envoyant un message RAC et l'appelé passe en attente de FIN (place P18) ; comme précédemment (T12 est tirable) la consommation par l'appelant des deux messages provoque le retour de l'appelant à l'état oisif. Lorsque l'appelé, en attente de FIN, reçoit FIN (T21 est tirée), la procédure Appelé se termine.

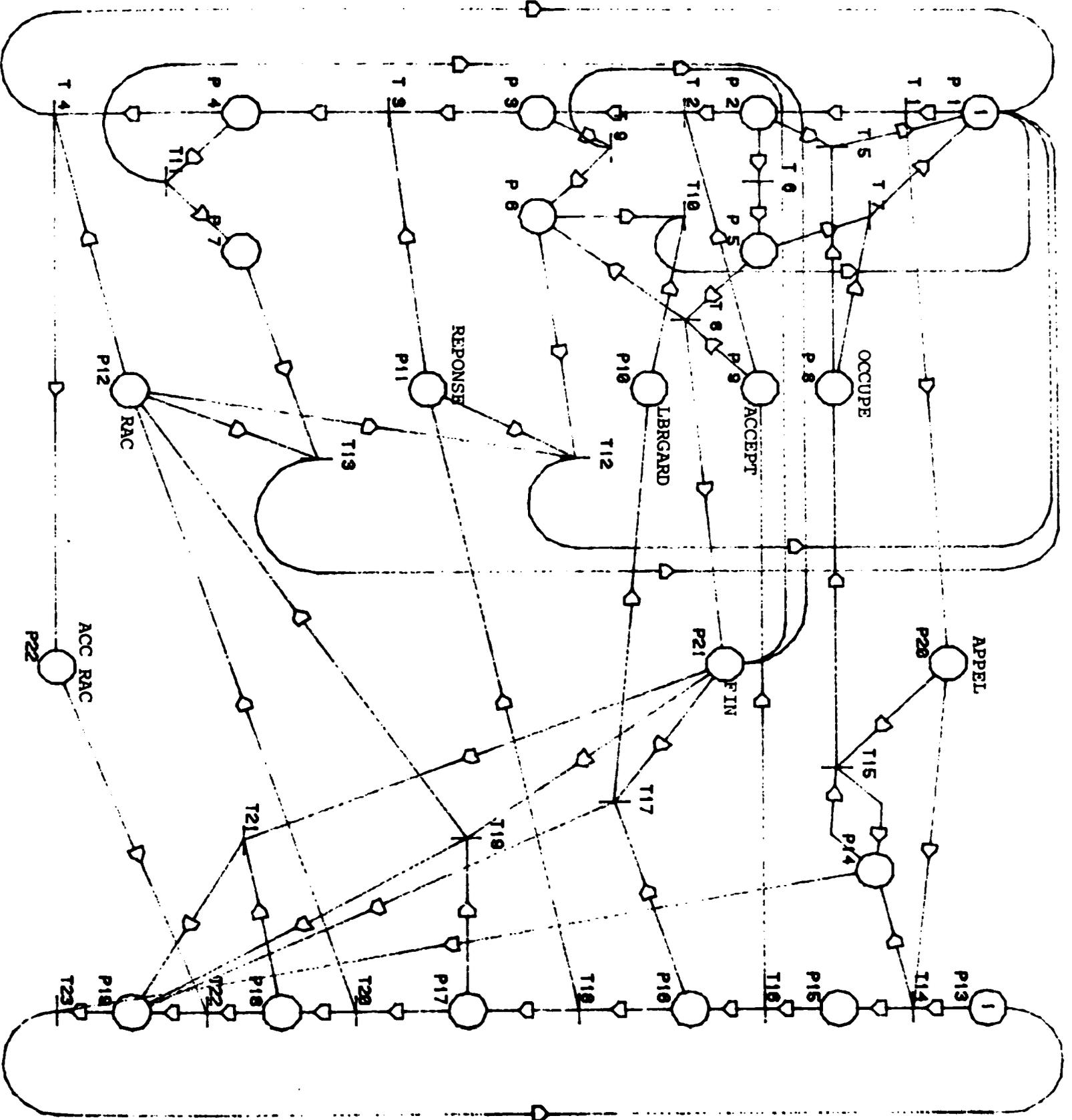


Fig. III.7 Modèle du Protocole Non Optimal à Nombre Réduit de Messages

b) L'appelant est dans l'état d'attente de libération après conversation (P7)

L'appelant attend alors le message RAC correspondant maintenant à la consommation par l'appelé du message FIN déjà émis par l'appelant ; après cette réception l'appelant redevient oisif.

Vérification du Protocole

L'analyse effectuée par OGIVE montre que ce protocole possède les propriétés requises : il est borné, sauf, réinitialisable et vivant.

En conséquence, ce protocole est logiquement correct pour les analyses effectuées.

III.7.3 MODELE DU PROTOCOLE OPTIMAL

Le message ACC RAC qui se trouve dans les deux protocoles précédents n'existe pas explicitement dans quelques systèmes de signalisation comme les systèmes récents CCITT N° 6,7 par exemple. Cet accusé de réception de raccrochage paraît nécessaire pour le bon fonctionnement des protocoles car les protocoles que nous avons pu proposer sans cet accusé se sont avérés non corrects. Nous essayerons ici de supprimer ce message du protocole donné dans la figure III.7, tout en conservant sa fonction, comme nous allons le voir.

Cela nous permettra d'obtenir le protocole le plus optimal possible (le critère étant le nombre de messages) ; ce protocole est indiqué sur la figure III.8 dans laquelle nous avons proposé d'utiliser le message FIN d'abord dans sa fonction originelle et en plus comme fonction d'accusé de réception pour le message RAC .

En réalité, le protocole donné dans la figure III.8 découle de celui de la figure III.7 en fusionnant les places P21 et P22 en une seule place (P21, figure III.8) et les transitions T21 et T22 en une seule transition (T21, figure III.8).

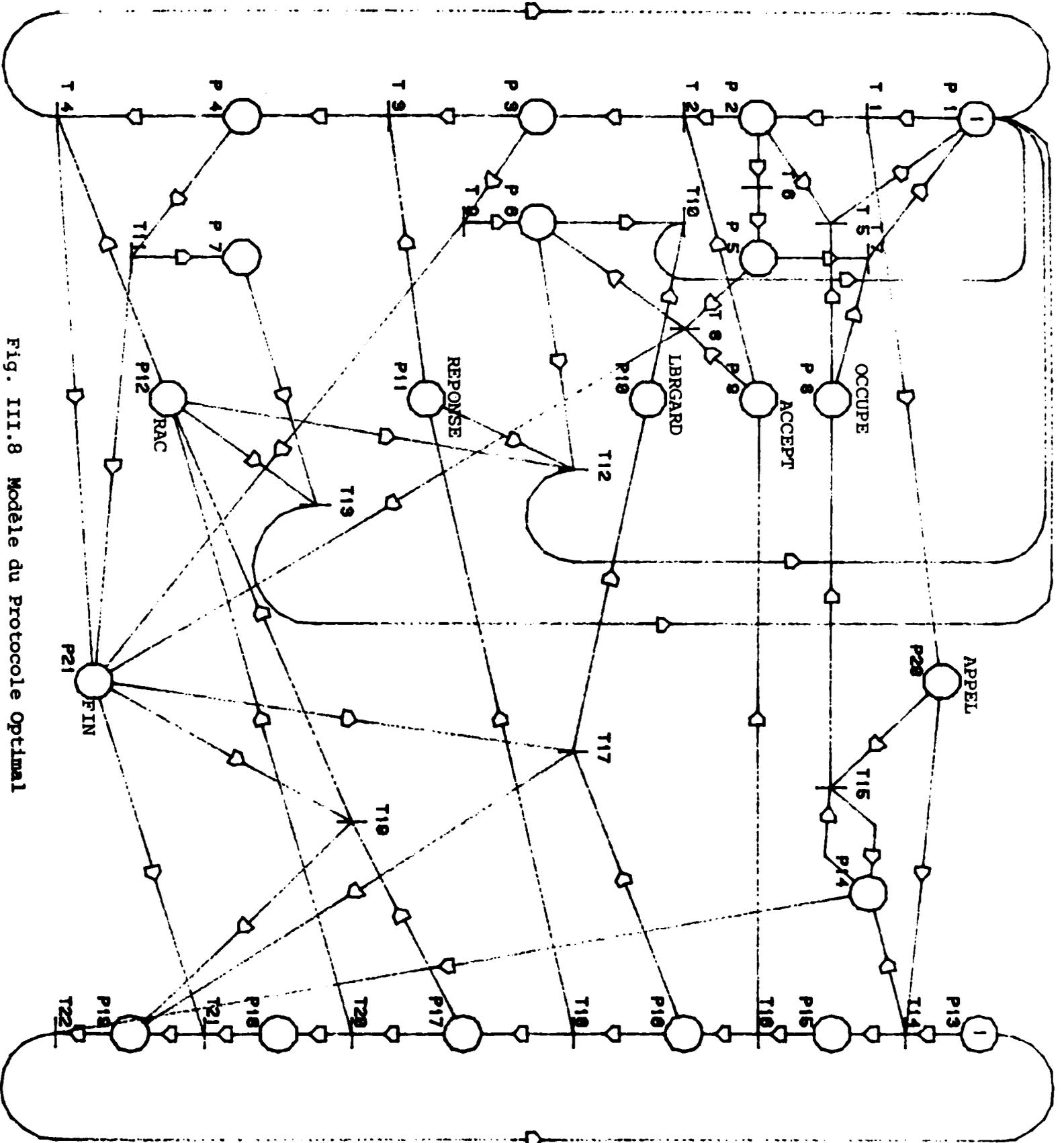


Fig. III.8 Modèle du Protocole Optimal

Supposons l'appelant et l'appelé dans l'état conversation (P4, P17 marquées) : lorsque l'appelé envoie un message RAC (P12 marquée par le tir de T20), il passe en état d'attente d'accusé (P18). Ce passage sera reçu par l'appelant (T4 tirée) qui transmettra FIN. A la réception de ce message FIN, l'appelé en attente d'accusé est libéré (T21 tirée).

#### Vérification de protocole

Le modèle de ce protocole est vérifié par OGIVE en énumérant ses marquages ; l'analyse montre que ce modèle est borné, sauf, réinitialisable et vivant.

En conséquence, ce protocole est logiquement correct pour les analyses effectuées.

#### III.7.4 MODELE DU PROTOCOLE AVEC ACQUITTEMENT POUR LE MESSAGE "OCCUPE"

Nous allons maintenant utiliser un acquittement pour le message OCCUPE avant que l'appelant ne se libère ; pour cela on ajoute un message FIN OCC qui agit comme accusé de réception de OCCUPE.

La figure III.9 donne le modèle résultat dans lequel nous considérons deux cas (notons cependant que le reste du comportement du protocole de la figure III.9 est identique à celui du protocole donné par la figure III.8) :

##### Cas 1 : L'appelant reçoit le message OCCUPE dans l'état "attente de l'état de l'appelé "(P2)

Puisque l'appelé est supposé occupé, lorsqu'il reçoit un appel (T13 est tirée), le message OCCUPE est transmis (P8 marquée), l'appelé reste occupé (P14) et attend l'acquiescement de OCCUPE (état P22). Lorsque l'appelant (dans l'état P2) reçoit le message OCCUPE (la transition T21 est tirée), il envoie l'acquiescement d'occupation FIN OCC (P23) et l'appelant passe dans l'état d'attente de libération (P6). A la réception de FIN OCC par l'appelé en attente d'acquiescement, (T23 est tirée) il transmet LBRGARD (P10) qui libère l'appelant en attente de libération (tir de T8) et l'appelant redevient oisif.

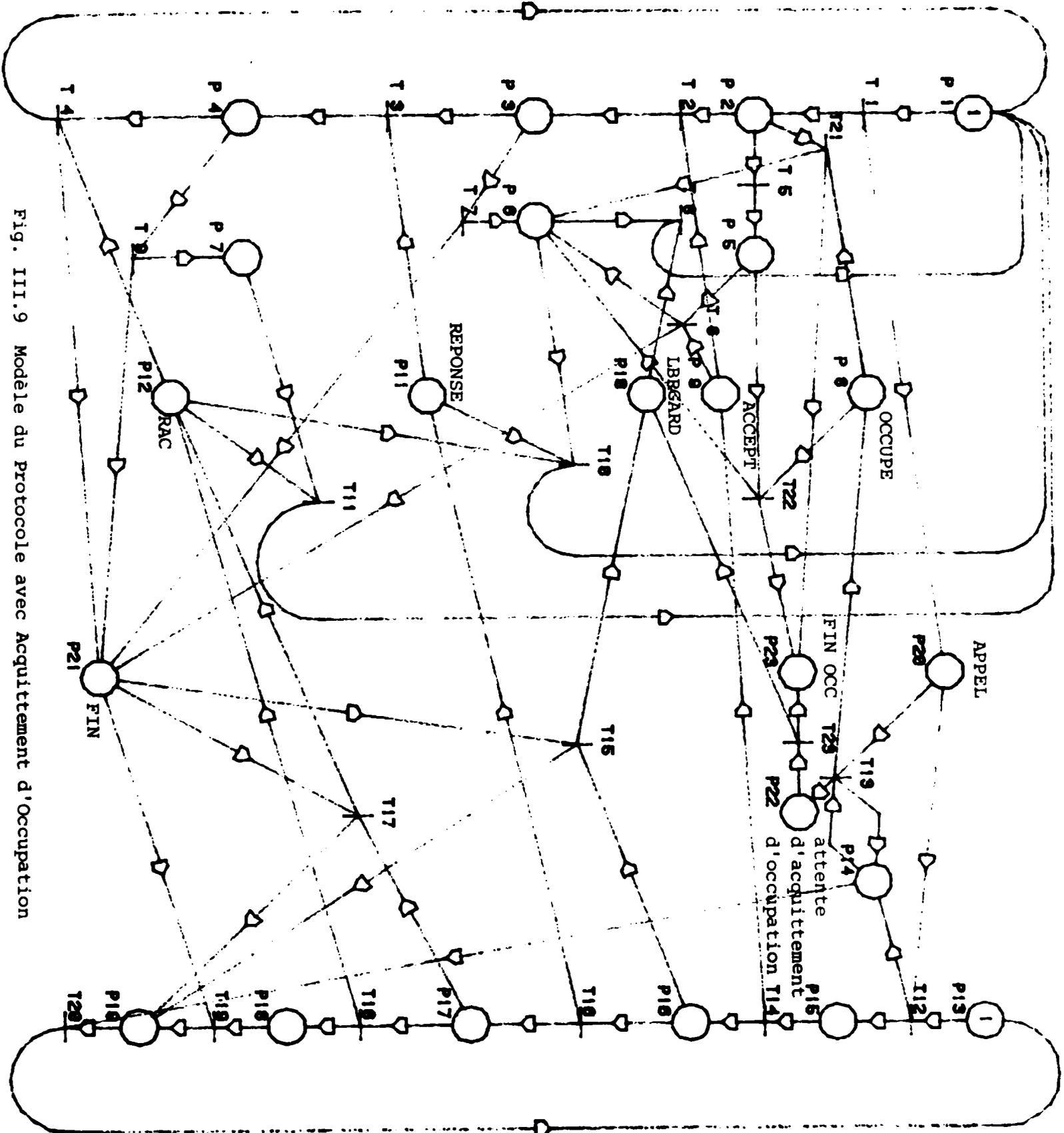


Fig. III.9 Modele du Protocole avec Acquittement d'Occupation

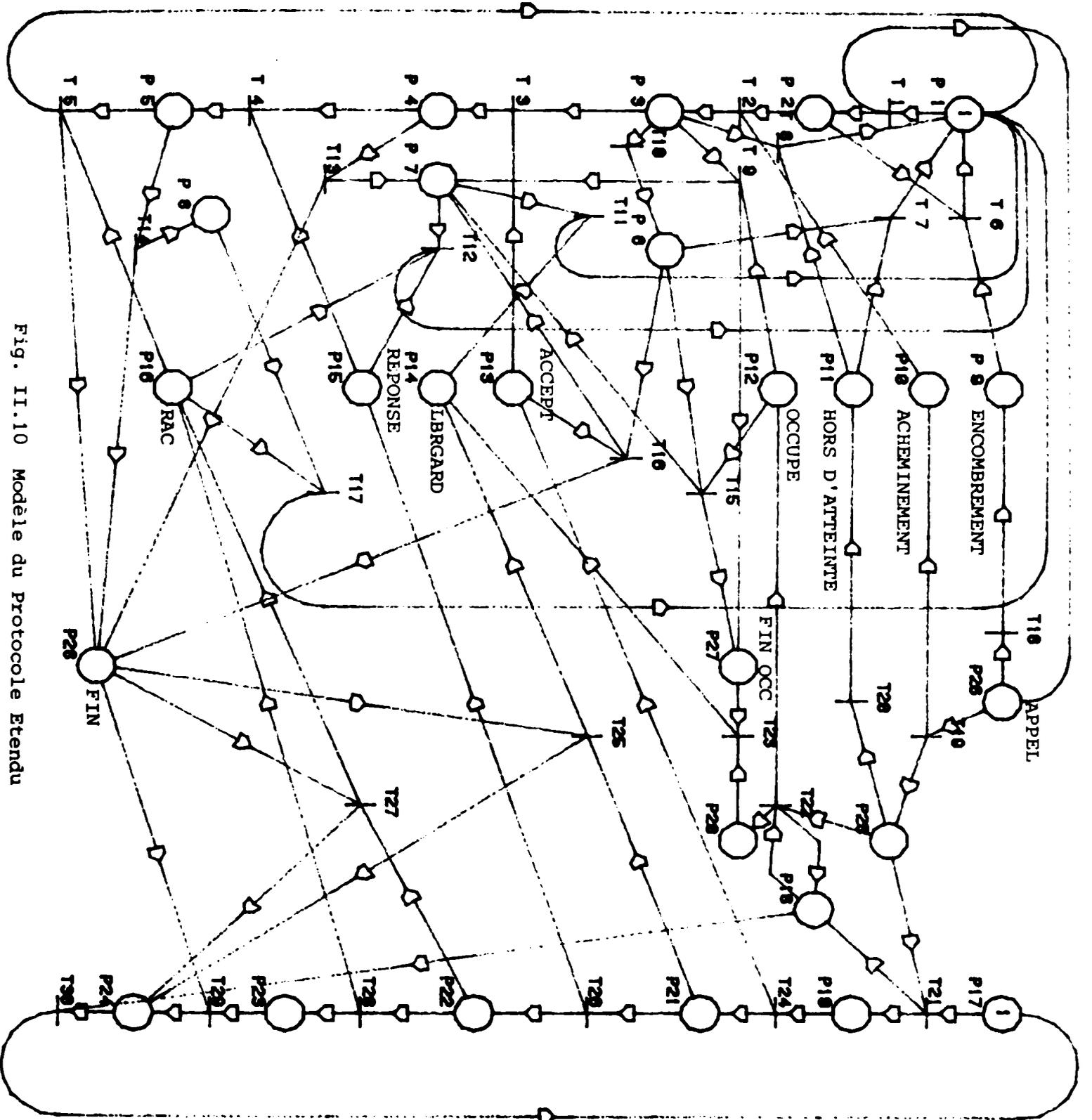


Fig. II.10 Modèle du Protocole Etendu

Lorsque l'appelant (P1) envoie l'appel (T1 tirée), il passe à l'état d'attente d'acheminement (P2) où il peut recevoir soit un message ENCOMBREMENT et donc revenir oisif (par le tir de T6) ou un message ACHEMINEMENT et il passe alors dans l'état "attente de l'état de l'appelé" (P3). Dans ce dernier état, il peut recevoir trois messages : le message HORS D'ATTEINTE qui le fait revenir oisif (par le tir de T8); le message ACCEPT qui le fait évoluer vers l'état d'attente de la réponse (P4) ; le message OCCUPE auquel il répond par le message FIN OCC avant de passer dans l'état d'attente de libération de garde (P7). Si l'appelant a raccroché (état P6) la réception du message HORS D'ATTEINTE le rend oisif (Tir de T7).

#### Vérification du Protocole

Le modèle de ce protocole est vérifié par OGIVE en énumérant ses marquages ; l'analyse montre que ce modèle est borné, sauf, réinitialisable et vivant.

En conséquence, ce protocole est logiquement correct pour les analyses effectuées.

#### III.7.6 MODELE DU PROTOCOLE TOTALEMENT ASSERVI

Les protocoles précédents possèdent un accusé de réception pour tous les messages venant de l'appelé sauf un : en effet, ils ne contiennent pas un accusé de réception de la réponse (que nous appellerons ACC REPONSE ). Nous proposons maintenant un protocole qui contient ce type de message et qui en conséquence permettra de parler de protocole totalement asservi car fonctionnant en appel-réponse, c'est-à-dire par couple de messages ; le modèle de ce protocole est donné par la figure III.11.

Comme dans les cas précédents, nous décrirons ici uniquement la partie de ce protocole qui concerne les messages REPONSE et ACC REPONSE car les autres parties du protocole peuvent facilement être comprises à partir du fonctionnement des protocoles précédents.

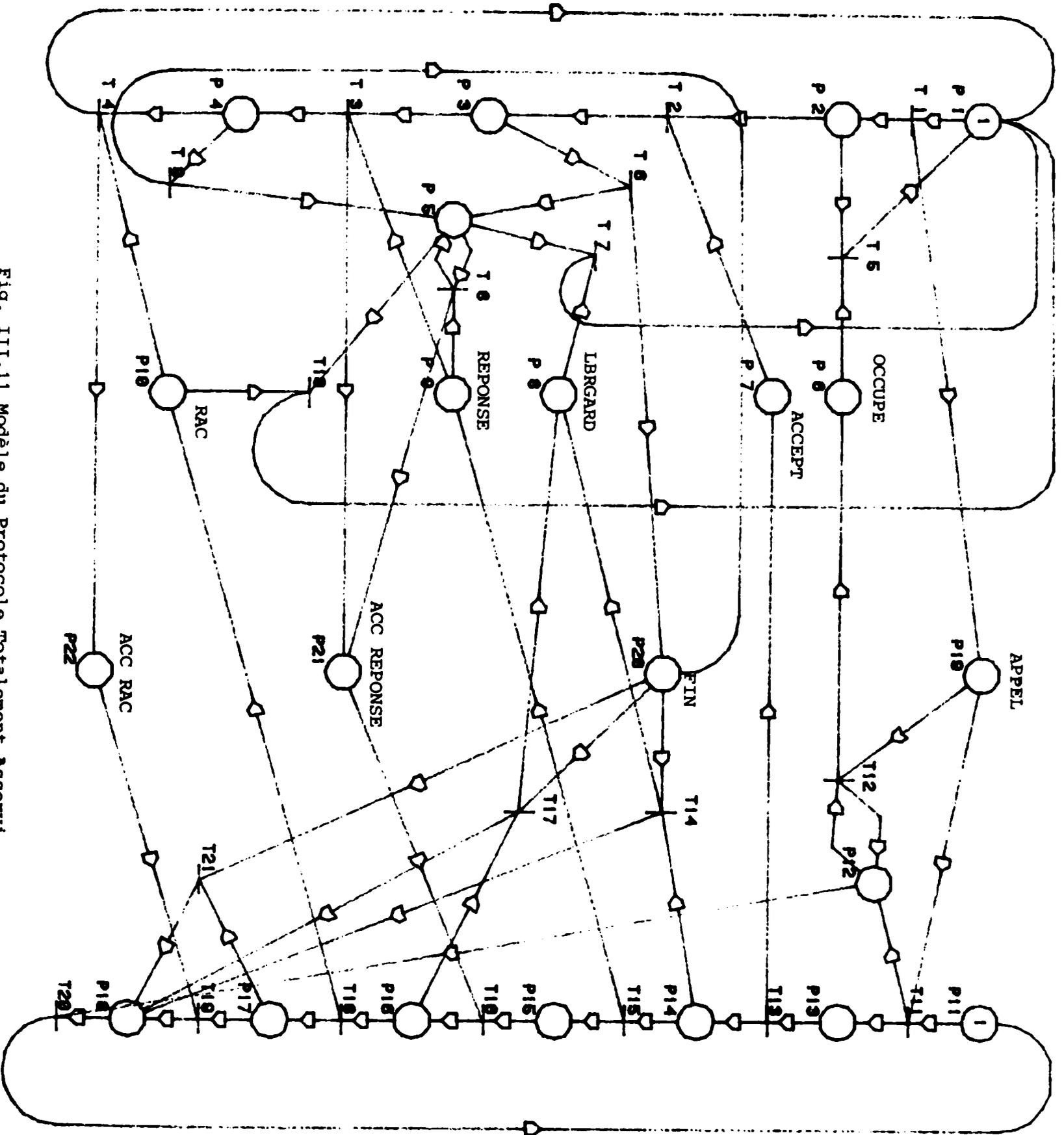


Fig. III.11 Modèle du Protocole Totalemment Asservi

Supposons l'appelant dans l'état "attente de la réponse de l'appelé" (P3) et l'appelé dans l'état "élaboration de la réponse" (P14).

Lorsque l'appelé a envoyé REPONSE (par le tir de T15) et que ce message a été reçu par l'appelant, celui-ci évolue et passe dans l'état de conversation (P4) en transmettant ACC REPONSE (par le tir T3). A la réception de ACC REPONSE, l'appelé passe dans l'état de conversation P16 (par le tir de T16).

Si le message REPONSE est envoyé mais arrive chez l'appelant après le raccrochage de cet appelant (P5) et l'envoi de FIN, alors, dans ce cas, l'appelant envoie ACC REPONSE (par le tir de T8) qui, lorsqu'il est reçu chez l'appelé, le conduit dans l'état conversation P16 (par le tir de T16); mais recevant là FIN, il envoie LBRGARD à l'appelant et finit sa procédure Appelé; à la réception de LBRGARD l'appelant redevient oisif (par le tir de T7).

#### Vérification du Protocole

Le réseau est vérifié par OGIVE en énumérant ses marquages; l'analyse montre que le réseau est borné, sauf, réinitialisable et vivant.

En conséquence, ce protocole est logiquement correct pour les analyses effectuées.

### III.8 CONCLUSION

Dans ce chapitre, nous avons proposé plusieurs protocoles de communication utilisables afin de connecter entre eux des autocommutateurs téléphoniques. Ces protocoles ont d'abord été modélisés en utilisant les réseaux de Pétri et ensuite validés, c'est-à-dire montrés bornés, saufs, réinitialisables et vivants.

Afin de réaliser ce travail, nous avons choisi comme base de départ les signaux donnés par les recommandations CCITT N° 6,7. Ces signaux ont d'abord été regroupés en classes fonctionnelles, nous avons ensuite proposé différents protocoles contenant soit une partie de ces messages, soit des messages non explicitement inclus dans la recommandation.

Ces modélisations nous ont permis de montrer :

- qu'il était possible de spécifier de façon non ambiguë plusieurs protocoles légèrement différents, plus ou moins complets, permettant de réaliser la signalisation dans les systèmes téléphoniques, malgré leur complexité.
- que la modélisation effectuée pouvait être prouvée bornée et sans blocage total ou partiel, ceci par la nature même des réseaux dès que leur applicabilité était montrée.

Nous devons néanmoins souligner que les réseaux donnés ici n'ont pas été obtenus très facilement. Il a été nécessaire de représenter de façon correcte les interactions et ceci n'a pas été effectué dès le premier essai. Nombreux sont les réseaux que nous avons dessinés et que nous n'avons pas donné ici car ils se sont avérés non bornés ou non vivants. Dans chacun de ces cas, nous avons dû avoir recours à OGIVE et à l'analyse afin de nous persuader que ces réseaux n'étaient pas corrects. En conséquence, la confiance résultant de cette possibilité d'analyse nous a permis d'envisager et de spécifier un grand nombre de variantes pour ce protocole, variantes pouvant dépendre de la présence ou non de certains messages.

Notre but n'était pas de définir le "meilleur choix" mais de donner une approche permettant de définir et proposer facilement des protocoles corrects. Le choix, pouvant dépendre de très nombreux paramètres, a été considéré comme étant en dehors de notre travail. Simple-  
ment, il semble que le protocole optimal (éventuellement le protocole étendu) soit intéressant car il implique un nombre minimal de messages échangés.

§§§

**C H A P I T R E I V**

-----

**PRINCIPES D'UNE IMPLEMENTATION**

**DE PROTOCOLES**

**DECRIIS PAR DES RESEAUX DE PETRI**

-----



## IV.1 INTRODUCTION

Nous avons spécifié et vérifié dans le chapitre précédent différents protocoles de communication déduits des systèmes de signalisation recommandés par le CCITT. La vérification de ces protocoles a été effectuée par l'analyse des réseaux de Pétri non étiquetés représentant les liens explicités entre deux abonnés communicants. Ainsi, toutes les interactions existant entre les abonnés ont été ramenées à des couples d'interactions entre l'abonné appelant et l'abonné appelé.

Le problème est donc maintenant de proposer, à partir des réseaux précédemment étudiés, une implémentation et, plus généralement, une approche systématique conduisant à l'implémentation. Cette méthode ne peut absolument pas être générale et certains choix sont nécessaires. Nous avons choisi de réaliser l'implémentation de telle façon qu'elle satisfasse les trois critères suivants qui nous paraissent, dans notre contexte, les plus importants :

(i) L'implémentation doit être la plus proche possible de la modélisation effectuée, dans notre cas les réseaux de Pétri déjà étudiés :

- le passage du réseau de Pétri à l'implémentation pourra être rendu systématique ;
- ce passage devra conduire à une implémentation conforme aux spécifications formelles, préalablement validées, ce qui apparaît extrêmement intéressant.

(ii) Le matériel utilisé sera un microcalculateur. En effet, il paraît possible d'allouer un microcalculateur à chaque abonné ou à un groupe de quelques abonnés, ceci d'autant plus que dans les prochaines années les microcalculateurs intégrés en un seul boîtier se développeront.

(iii) Cette implémentation doit de plus être compatible avec le réseau **REBUS (RECONFIGURATION sur BUS)**, réseau développé pour la commande

et le contrôle en temps réel [30] par l'équipe LC avec l'aide du service MITRI et en collaboration avec la société SCHLUMBERGER.

Ce réseau a pour but de réaliser un système de communication entre un ensemble de processeurs fonctionnels distribués.

Le système de communication est constitué de processeurs d'interface d'une part connectés à leur propre processeur fonctionnel, et d'autre part à un bus série doublé. Ainsi, il apparaît que ce réseau fournit un support intéressant pour notre implémentation.

Néanmoins, l'état d'avancement du projet REBUS au début de notre étude et les difficultés soulevées nous ont conduit à envisager non pas une implémentation sur REBUS, mais une simulation sur un système de développement INTELLEC. Ainsi, le transport sera facile entre le système de développement et les mémoires mortes des processeurs de REBUS. De plus, une telle approche permettra au concepteur, s'il le juge nécessaire, de compléter la validation effectuée avec OGIVE par une simulation plus poussée sur le système de développement.

Dans ce but, le simulateur réalisé doit permettre la mise en oeuvre de mécanismes existants dans l'abonné, en particulier l'envoi et la réception de messages. Ceci nous a conduit à sélectionner une implémentation reposant sur l'utilisation du moniteur RMX développé par INTEL [31]. De plus un moniteur proche de celui-ci a été le premier logiciel développé sur REBUS [32] et il supporte actuellement tout le système de communication. Nous pourrions alors l'utiliser afin de construire, de façon cohérente, notre implémentation.

En conséquence, ce chapitre exposera d'abord les fonctions primitives qui nous paraissent nécessaires et présentera ensuite leur réalisation en utilisant RMX qui s'avèrera particulièrement adapté pour décrire la synchronisation.

Ceci nous permettra d'aborder ensuite la simulation proprement dite, c'est-à-dire sa structure générale, la relation entre les réseaux de Pétri et les primitives de RMX et l'implémentation du processus représentant l'abonné.

## IV.2 PRESENTATION DU MONITEUR TEMPS REEL MULTITACHES

### IV.2.1 LES TACHES ET LES MESSAGES

Notre premier problème est donc d'implémenter les abonnés. Il apparaît clairement que les abonnés sont des entités indépendantes, évoluant de façons séparées. Ceci implique que cet asynchronisme doit être rendu au niveau de la simulation. Une possibilité consiste à associer à chacune des entités une tâche (ou un processus).

Il sera alors nécessaire de faire collaborer ces tâches et de les synchroniser. En effet, ces tâches doivent contrôler le monde extérieur à l'abonné et répondre aux événements provenant du monde extérieur. Ces événements extérieurs seront soit locaux (décrochage, raccrochage, ..) soit distants, c'est-à-dire pouvant provenir d'autres abonnés géographiquement éloignés (appel, occupé, fin...). Dans le premier cas, ces événements sont des événements classiques ; dans le deuxième cas, ces événements seront en fait des messages qui devront donc transiter entre les abonnés. Dans le cas des abonnés distants, celui qui nous intéresse le plus ici, les abonnés envoient un message à leur correspondant et doivent attendre leur réponse, ce qui nécessite l'utilisation de primitives d'envoi et d'attente de messages.

Nous allons voir maintenant comment dans ce cas un moniteur du type RMX constitue en fait un excellent support car il contient toutes les primitives qui seront nécessaires à l'implémentation.

#### IV.2.2 LES CONCEPTS DE BASE DU MONITEUR

Il y a trois concepts de base à manipuler dans l'utilisation d'un moniteur multitâches du type RMX.

Le premier est la tâche : portion de code, exécutée de façon autonome et qui communique avec d'autres tâches pour se synchroniser, coopérer ou partager l'accès à une ressource. Par la suite, la tâche sera représentée par un rectangle.

Le second concept est le message : unité de communication entre les tâches. Les messages contiennent de l'information et/ou des événements nécessaires à la synchronisation entre les tâches. Ce message, dans le cas qui nous intéresse, devra comporter les informations suivantes :

- le numéro de l'abonné expéditeur du message,
- le numéro de l'abonné destinataire du message,
- le type du message échangé entre les abonnés.

Le troisième concept est l'échange : sorte de boîte aux lettres où les tâches déposent les messages à destination d'autres tâches ou attendent des messages en provenance d'autres tâches.

L'échange se présente donc comme une double file d'attente FIFO (une pour les tâches et une pour les messages). Dans les figures, il sera représenté par un double cercle.

#### IV.2.3 LES PRIMITIVES IMPORTANTES : ATTENTE ET ENVOI DE MESSAGES

Les transferts de messages entre les tâches nécessitent des primitives de coordination, par exemple l'envoi d'un message et l'attente d'un message [33]. Dans le moniteur, ces messages ne sont pas envoyés directement aux tâches mais sont envoyés à des échanges qui relieront les tâches communicantes. Ceci est réalisé par les deux primitives suivantes :

### 1. Primitive d'envoi (RQSEND)

Cette primitive, à deux paramètres d'appel, provoque l'envoi d'un message spécifié à un échange déterminé. Si aucune tâche n'attend sur l'échange, le message est chaîné dans la file des messages ; si il y a une tâche ou plusieurs qui attendent sur l'échange, la première tâche en attente reçoit le message.

L'appel de cette primitive s'effectue en spécifiant deux paramètres : le premier est l'adresse de l'échange vers lequel le message est envoyé et le second spécifie l'adresse du message envoyé.

Le format de l'appel est le suivant en PLM :

Call RQSEND (adresse d'échange, adresse de message) ;

### 2. Primitive d'attente (RQWAIT)

Cette primitive, à deux paramètres d'appel, provoque l'attente d'une tâche sur un échange jusqu'à la réception d'un message ou la fin d'un temps d'attente. Si le temps d'attente est écoulé avant que le message attendu n'arrive, le moniteur multitâches engendre un réveil qui libère la tâche en la déconnectant de l'échange et l'utilisateur est averti. L'appel de la primitive RQWAIT exige deux paramètres : le premier est l'adresse de l'échange sur lequel la tâche attend, et le second spécifie le temps d'attente ; si le temps d'attente est illimité ce second paramètre vaudra "0", et la tâche attendra le message indéfiniment.

Le format d'appel est le suivant en PLM :

Adresse de Message = RQWAIT (adresse d'échange, temps d'attente) ;

Le lecteur pourra se reporter à [31,32] pour avoir des précisions supplémentaires sur le moniteur qui comporte d'autres mécanismes et primitives. Néanmoins, seuls les concepts déjà donnés seront manipulés dans la suite de notre texte.

## IV.3 BASES DE LA SIMULATION

### IV.3.1 PRINCIPES GENERAUX

Notre but est maintenant d'effectuer la simulation à partir des tâches, représentant les abonnés, qui communiquent entre elles selon le protocole à simuler.

Dans cette simulation, deux aspects principaux doivent être pris en compte :

- l'échange des informations avec le monde extérieur ;
- l'échange d'informations entre les différents abonnés.

Ces informations sont parfaitement définies pour les cas qui nous intéressent car ce sont des événements explicitement précisés au niveau de la spécification comme des actions externes au module représentant un abonné.

Nous avons donc à réaliser la tâche de l'abonné à partir des protocoles validés. Nous choisirons ici celui qui nous paraît le plus intéressant, le protocole optimal.

Afin de passer d'un réseau de Pétri à l'utilisation des primitives "envoi" et "réception" du moniteur, considérons les réseaux de la figure IV.1. Les places P 1 et P 2 représentent les états de la tâche associée à l'abonné et la place P3 un événement. Ainsi, le cas a) peut être implémenté comme l'attente d'un événement (soit l'attente d'un message) sur un certain échange, venant d'une autre tâche ; le cas b) peut être implémenté par l'envoi d'un événement, vers un certain échange, pour une autre tâche.

Comme à un moment donné, une tâche peut attendre plusieurs messages venant de plusieurs abonnés, nous associerons uniquement, pour simplifier, un seul échange par abonné ; cet échange sera  $E_i$  si l'abonné possède le numéro  $i$  et la tâche correspondante attendra tous ses messages dans cette (unique) boîte aux lettres.

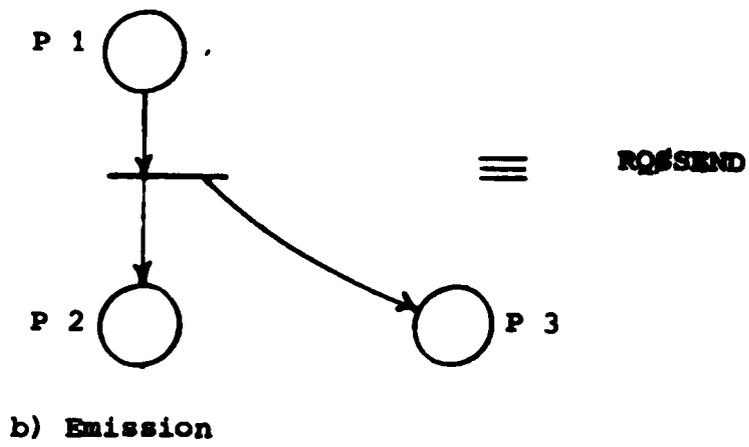
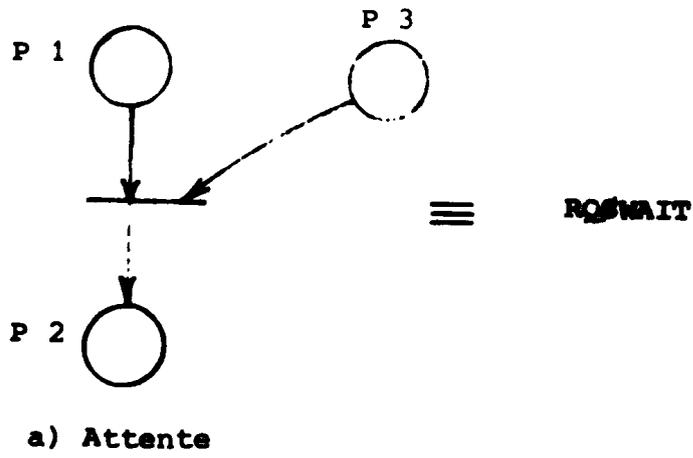


Fig IV.1

Ceci nous conduit à la structure de simulation donnée par la figure IV.2 dans laquelle chaque abonné est représenté par une tâche qui reçoit les messages venant d'autres tâches.

Les tâches "Monde Extérieur" permettent de connecter les abonnés avec le monde extérieur. La tâche "Abonnés Physiques" simule les abonnés physiques (par exemple le décrochage et le raccrochage) et visualise le comportement du système. La tâche "Medium" simule l'envoi et la réception des messages relatifs aux abonnés distants et permet éventuellement de simuler des cas de pannes.

Il apparaît ainsi que, de façon cohérente, les tâches communiquent à travers les échanges.

#### IV.3.2 REPRESENTATION DE L'ABONNE

A partir des considérations précédentes, le comportement de l'abonné est donné par le modèle de la figure IV.3 déduit du Protocole Optimal donné par la figure III.8. Dans cette figure IV.3 :

$(\alpha | \beta)$ , (prédicat/action) : est l'étiquette associée à une transition ;

$j ? \text{MES}$  : signifie que le message MES est attendu de la tâche j ;

$j ! \text{MES}$  : signifie que le message MES est envoyé à la tâche j ;

$i \text{ ext}$  : signifie que la tâche communicante est l'abonné physique ;

$X$  : signifie que la tâche communicante peut être quelconque.

Nous voyons dans cette figure que les places ont été remplacées par des émissions ou réceptions de messages et que les deux parties de l'abonné (appelant et appelé) sont représentées.

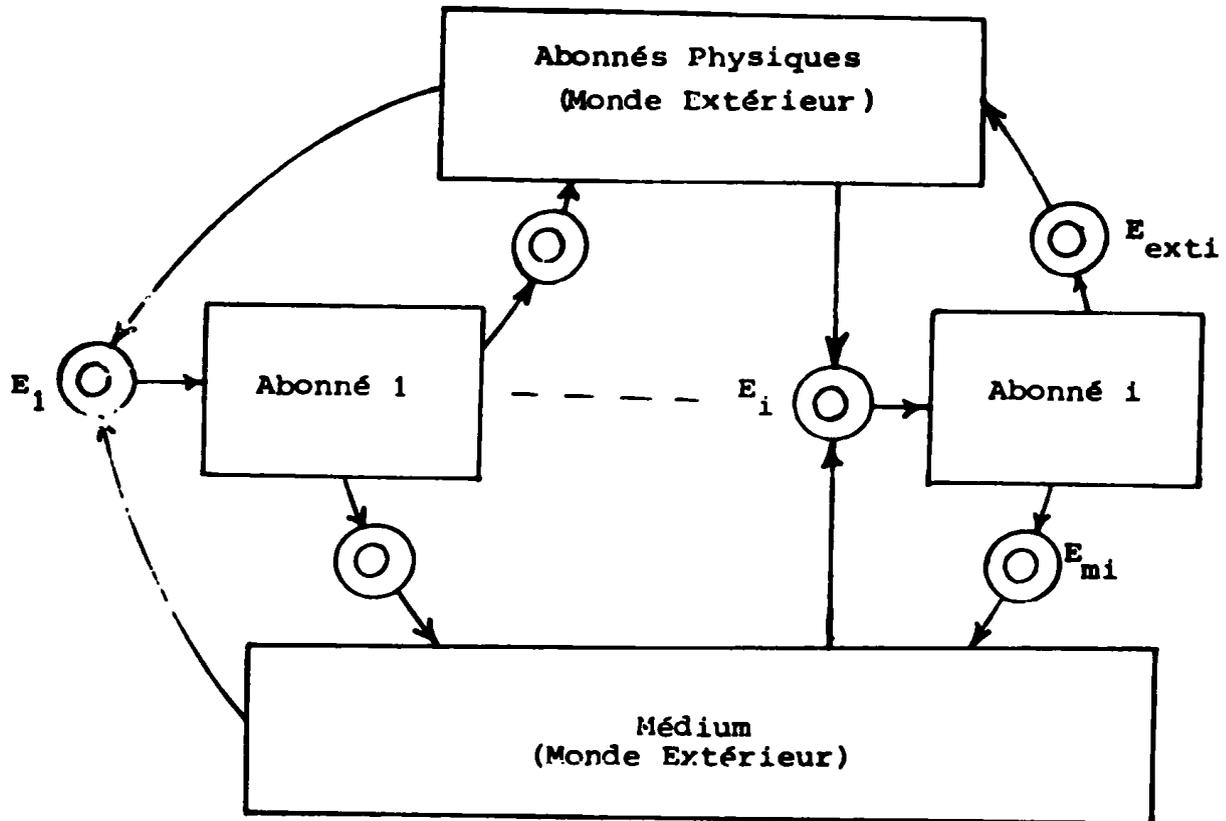
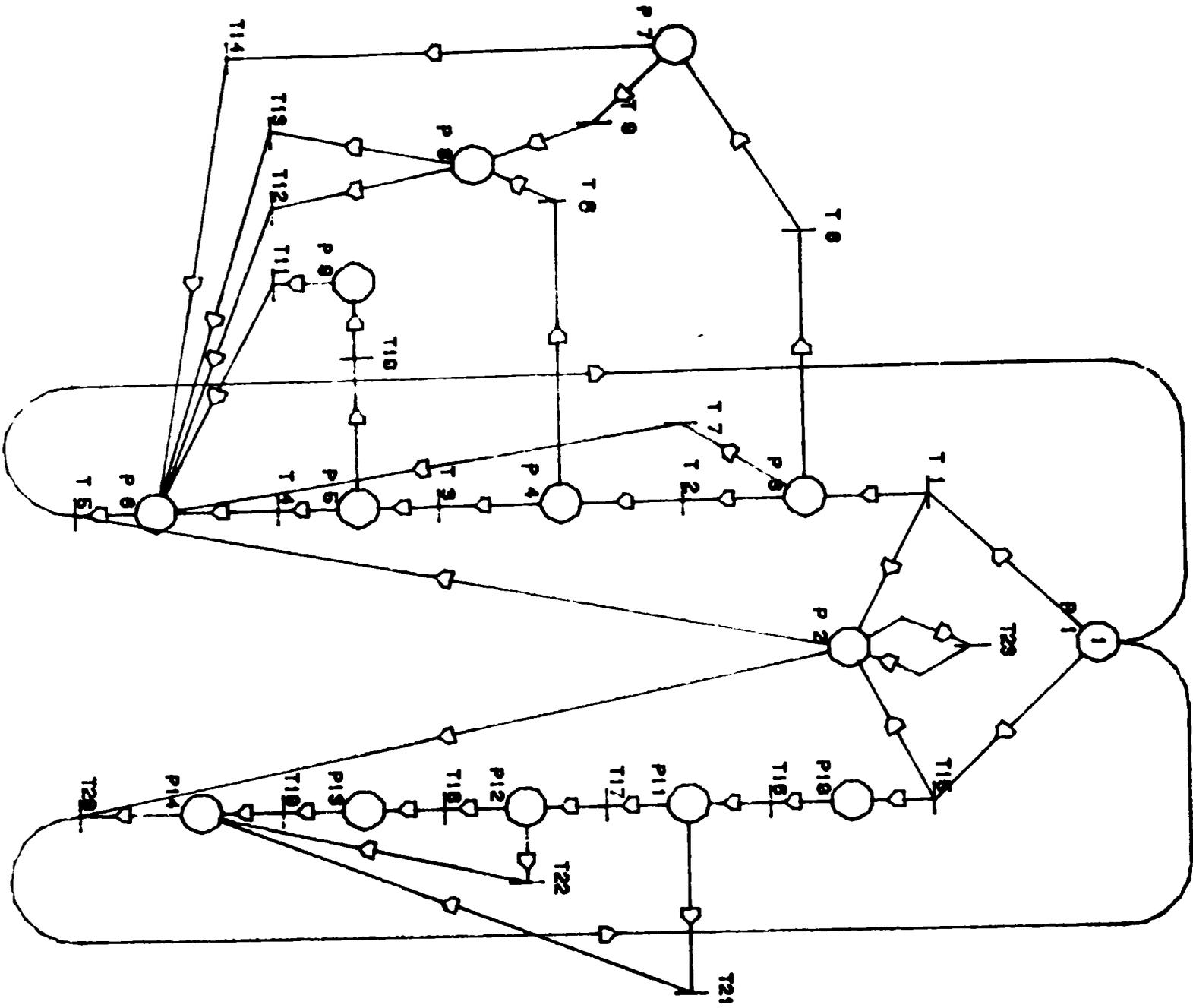


Fig. IV.2 Structure de Simulateur



ETIQUETTES

- T 1 : (text ? déc(j) | j ! APPEL
- T 2 : (j ? ACCEPT | )
- T 3 : (j ? REPONSE | )
- T 4 : (j ? RAC | j ! FIN)
- T 5 : -
- T 6 : (text ? rac | )
- T 7 : (j ? OCCUPE | )
- T 8 : (text ? rac | j ! FIN)
- T 9 : (j ? ACCEPT | j ! FIN)
- T 10 : (text ? rac | j ! FIN)
- T 11 : (j ? RAC | )
- T 12 : (j ? REPONSE \ j ? RAC |
- T 13 : (j ? LBARGARD | )
- T 14 : (j ? OCCUPE | )
- T 15 : (j ? APPEL | )
- T 16 : ( | j ! ACCEPT)
- T 17 : (text ? déc | j ! REPONSE)
- T 18 : (text ? rac | j ! RAC)
- T 19 : (j ? FIN | )
- T 20 : -
- T 21 : (j ? FIN | j ! LBARGARD)
- T 22 : (j ? FIN | j ! RAC)
- T 23 : (X ? APPEL | X ! OCCUPE)

FIGURE IV.3. Modèle de l'abonné i déduit du protocole optimal

L'exécution de ce réseau de Pétri, comme celle de tout réseau peut être réalisée de différentes façons, plus ou moins intéressantes, selon l'implémentation réelle. Nous choisirons ici une méthode directe consistant à faire évoluer le marquage du modèle de l'abonné selon les réceptions de messages, ceci pour être le plus proche possible du réseau de Pétri de départ. D'autres solutions, au moyen de nombreux RQ\$WAIT et RQ\$SEND spécifiques aux messages, sont aussi envisageables.

Dans notre cas, nous ne nous intéressons uniquement qu'au protocole, les prédicats seront des réceptions de messages et les actions des envois de messages ; l'extension à des actions et des prédicats locaux aux abonnés serait aisée et tout à fait compatible avec notre choix.

Il faut remarquer cependant qu'à la réception d'un évènement peut correspondre plusieurs évolutions possibles du modèle (c'est-à-dire plusieurs transitions) ; cette remarque nous amène à considérer deux méthodes selon que l'on poursuit l'exécution du réseau par rapport à l'état courant de l'abonné ou par rapport à l'évènement reçu.

C'est la seconde méthode que nous choisirons ici car elle implique une recherche rapide de la transition à tirer.

Appelons SIMPET la procédure implémentant cette seconde méthode : son organigramme est donné par la figure IV.4. Cette procédure sera en fait réentrante car elle devra être utilisée par les différentes tâches parallèles qui s'exécutent. En effet, les données (tables, variables) caractérisant les abonnés sont engendrées à l'initialisation. En particulier chaque tâche Abonné comporte dans son contexte une adresse spécifique désignant un échange par défaut, appelé échange implicite. Son intérêt est le suivant : si dans la procédure RQ\$WAIT, le paramètre correspondant à l'échange est nul, alors l'échange d'attente sera l'échange implicite ; ceci permettra de respecter l'hypothèse de réentrance car ce paramètre n'aura pas à être spécifié au niveau du code de la tâche.

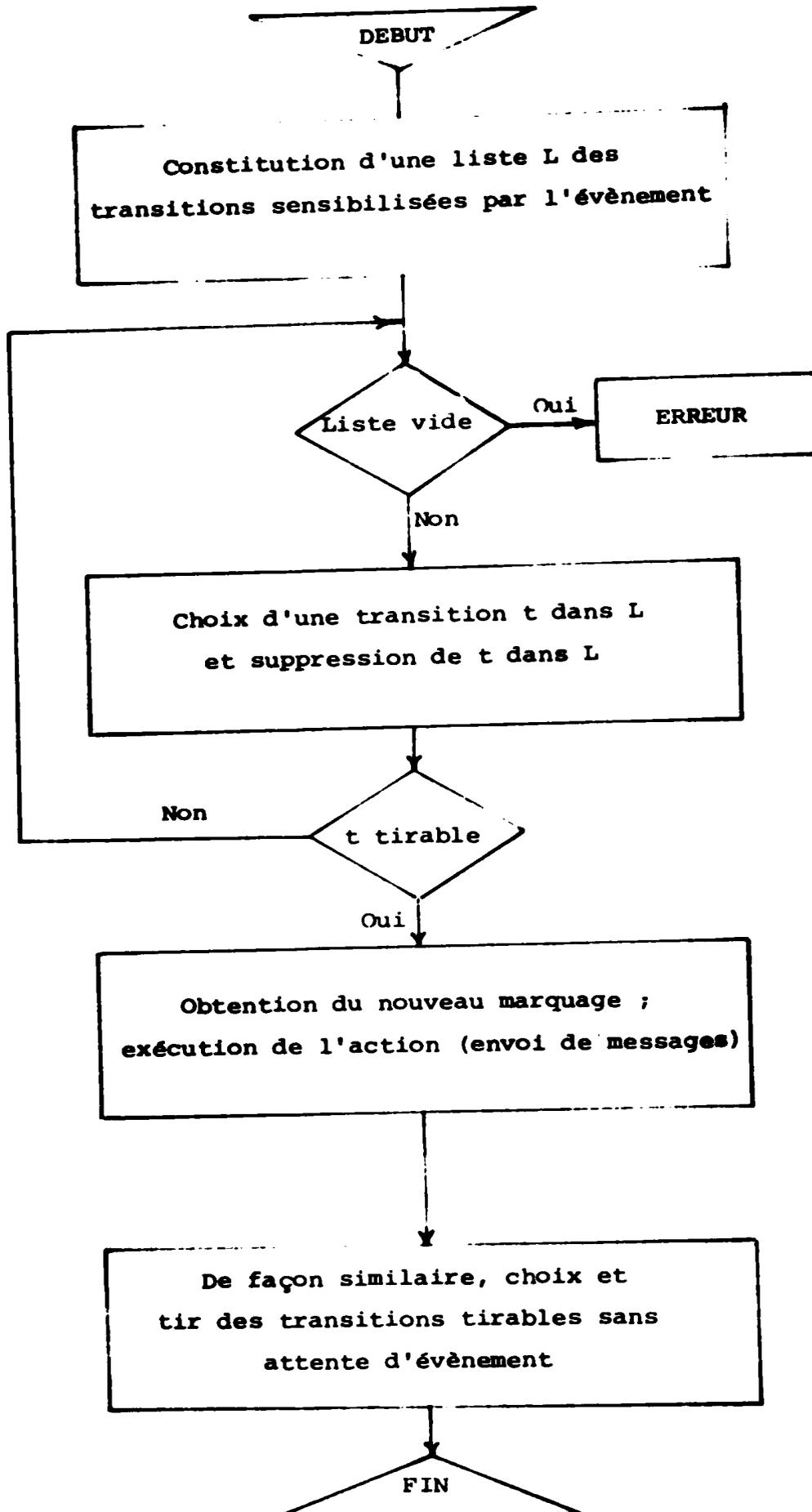


Fig. IV.4 Organigramme de SIMPET

En utilisant cette propriété, la tâche "Abonné" est alors très simple et peut s'écrire en PLM comme suit :

```
DO FOREVER ;  
/* attente illimitée sur l'échange implicite  $E_i$  de l'abonné i */  
MESSADD = RQWAIT (0,0) ;  
/* à la réception, activation de SIMPET */  
CALL SIMPET (MESSADD) ;  
END ;
```

La procédure SIMPET implémente l'algorithme donné dans la figure IV.4. Elle a comme paramètre d'appel l'adresse du message contenant l'évènement destiné à l'abonné ; elle engendre les actions éventuelles sur les variables internes et l'envoi du ou des messages.

#### IV.4 CONCLUSION

Dans ce chapitre nous avons présenté une approche possible permettant, à partir d'un réseau de Pétri, d'utiliser les primitives d'envoi et de réception de messages afin de faire coopérer des tâches concurrentes. Ceci permet d'obtenir une implémentation très proche des spécifications validées par OGIVE et donc d'utiliser la validation du réseau pour effectuer la validation de l'implémentation. Cette approche a été illustrée en l'appliquant à un protocole de communication, le protocole optimal donné dans le chapitre précédent.



# CONCLUSION

-----



Nous avons voulu, lors de ce travail, montrer l'intérêt découlant de l'utilisation des réseaux de Pétri pour la conception des protocoles de communication plus particulièrement utilisables dans le domaine téléphonique. Les enseignements résultants ont été obtenus en considérant un exemple réel, significatif et complexe, à savoir des systèmes de signalisation très proches de ceux donnés par les recommandations CCITT N°6,7.

Afin d'aboutir à ce résultat, il a été nécessaire de choisir dans une première étape le modèle de base permettant la spécification et l'analyse. A partir des possibilités existantes, nous avons choisi de spécifier les systèmes temps réels par un couple informalité-formalité, l'informalité provenant de l'utilisation du langage naturel et la formalité de l'utilisation des réseaux de Pétri.

Un point spécifique important est l'existence de règles qui permettent le passage du texte informel au réseau de Pétri. Cette association nous paraît extrêmement intéressante car elle associe l'explication (le texte écrit en langage naturel) et la non-ambiguïté (le dessin explicite du réseau de Pétri). L'approche résultante a été utilisée d'abord pour spécifier une communication téléphonique considérée du point de vue d'un abonné.

Un premier niveau de description a introduit les procédures Appelant et Appelé.

Un deuxième niveau de description a conduit à préciser ces procédures Appelant et Appelé. Lors de cette deuxième étape, les interactions existantes entre les deux abonnés sont apparues et il a été nécessaire de les modéliser de façon non ambiguë : les réseaux de Pétri supportés par l'outil graphique interactif de vérification OGIVE, se sont avérés particulièrement adaptés et efficaces lors des différentes modélisations et analyses que nous avons effectuées.

En effet, il apparaît rapidement que les choix nécessaires pour concevoir un protocole sont très nombreux, surtout si celui-ci est

complexe, et plusieurs essais sont indispensables pour obtenir une bonne spécification fonctionnelle. Dans chacun des essais, le parallélisme inhérent au problème conduit inévitablement à un grand nombre de possibilités en ce qui concerne la nature et le transfert des messages dans le réseau de communication. En conséquence, il nous paraît impossible, sans l'aide d'outils adaptés, de s'assurer du bon fonctionnement d'un protocole ; c'est pourquoi les réseaux de Pétri, malgré leurs limitations, se sont avérés très précieux. Nous devons signaler que dans notre étude, la grande efficacité obtenue a été pour la partie commande de la communication, à savoir l'établissement et la rupture. L'étude de la partie transmission des données est par ailleurs actuellement en cours de modélisation dans l'équipe Logiciel et Communication [23] .

Néanmoins, la partie commande paraît être extrêmement importante et donc sa représentation et son analyse sont fondamentales. Nous avons, quant à nous, proposé, spécifié, analysé plusieurs protocoles possibles.

Un certain nombre de ceux que nous avons modélisé lors de premiers essais se sont avérés à l'analyse non corrects et ce sont en fait ces erreurs corrigées par la suite qui, à notre avis, rendent indiscutable l'apport très positif de la spécification et de la vérification.

Enfin, nous avons dans le dernier chapitre abordé le problème de l'implémentation lié à celui de la vérification du modèle. En utilisant un moniteur temps réel multitâches contenant les primitives de synchronisation par envoi et réception de messages, il a été possible de proposer une approche permettant d'aboutir à une réalisation facilement déductible du modèle de spécification.

Un grand nombre de développements sont possibles à la suite de notre étude et les principaux seront poursuivis ou sont déjà entamés dans l'équipe. Ils concernent tous les aspects, à savoir la représentation, la spécification, l'analyse, la preuve, l'implémentation. La représentation par les réseaux de Pétri, même généralisés, est insuffisante pour modéliser les phénomènes spécifiques des systèmes de communication comme l'adressage logique, la diffusion, les effets de fautes.

La méthode de spécification au moyen du couple informalité-formalité peut être développée et un outil d'aide par ordinateur est indispensable. L'analyse possède de nombreuses limitations, dues au grand nombre de processus et de messages apparaissant dans un système distribué non trivial, ceci malgré l'approche modulaire multi-niveaux proposée en [8]. La preuve formelle en est encore à ses débuts et certains logiciels, parmi lesquels AFFIRM, ont pu être utilisés pour la première fois très récemment [34]. Enfin, l'implémentation des protocoles nous paraît encore maintenant extrêmement peu couplée à la vérification des spécifications et n'est supportée que par très peu d'outils permettant d'aider le concepteur.

Néanmoins, comme les problèmes qui ont été signalés sont très récents, nous croyons qu'un grand nombre de progrès vont être accomplis lors des prochaines années afin de permettre la conception sûre de fonctionnement des protocoles réalisant des algorithmes complexes dans un contexte distribué.



# B I B L I O G R A P H I E

-----



- 1) J.M.AYACHE, M.DIAZ  
"Méthodologie d'analyse et de synthèse de structures de commande de centraux électroniques en vue de l'amélioration de la sûreté de fonctionnement"  
"Modélisation multiniveaux des spécifications fonctionnelles d'un central de type E 10".  
Rapport du contrat d'étude préliminaire convention de recherche C.N.E.T.  
N° 789B0079009245, Avril 1979.
  
- 2) R.VALETTE, M.DIAZ  
"Méthodologie de spécification des systèmes parallèles: approche descendante"  
Rapport final de convention I.R.I.A. N° 1785 Juillet 1978.
  
- 3) G.V.BOCHMANN, C.A.SUNSHINE  
"Formal methods in communication protocol design"  
IEEE Trans.on comm., vol ; COM-28, N°4, April 1980.
  
- 4) G.V.BOCHMANN  
"Specification and verification of computer communication protocols"  
Report 294, December 1978, Département d'informatique et de recherche opérationnelle, Université de Montréal.
  
- 5) A.A.S.DANTINE  
"Protocol representation with finite-state models"  
IEEE Trans.on comm., vol ; COM-28, N°4, April 1980.
  
- 6) G.V.BOCHMANN  
"Finite state description of communication protocols"  
Computer network symposium, Liège, February 1978.
  
- 7) P.M.MERLÍN  
"A methodology for the design and implementation of communication protocols"  
IEEE Trans.on comm., vol ; COM-24, N°6, June 1976.

- 8) M.DEVY, M.DIAZ  
"Multilevel specification and validation of the control in communication systems"  
International conf. on distributed computing systems, Huntsville-Alabama, October 1979.
- 9) J.M.AYACHE, M. DIAZ, R.VALETTE  
"A methodology for specifying control in electronic switching systems"  
I.S.S. 79-Paris, May 1979.
- 10) M.DEVY  
"Modélisation et validation des protocoles multiniveaux dans les systèmes localement distribués"  
Thèse Docteur Ingénieur, Université Paul Sabatier, Septembre 1980.
- 11) M.DIAZ  
"Specification of concurrent systems : Model et methodology",  
"Validation of concurrent systems based on Petri-like models", 2nd Advanced Course "Computing Systems Reliability", September 10th, 21th 1979, Toulouse (France).
- 12) P.M.MERLIN  
"Specification and validation of protocols"  
IEEE Trans.on comm., vol ; COM-27, N°11, November 1979.
- 13) B.CHEZALVIEL , PRADIN  
"Outil graphique interactif pour la vérification des systèmes à évolution parallèle décrits par réseaux de Pétri"  
Thèse Docteur Ingénieur, Université Paul Sabatier, Décembre 1979.
- 14) R.VALETTE  
"Sur la description, l'analyse et la validation des systèmes de commande parallèles"  
Thèse Docteur D'état, Université Paul Sabatier, Novembre 1976.

- 15) B.BERTHOMIEU  
"Analyse structurelle des réseaux de Pétri : Méthodes et outils"  
Thèse Docteur Ingénieur, Université Paul Sabatier, Septembre 1979.
- 16) D.T.ROSS  
"SADT structured analysis and technique author guide"  
August 1976.
- 17) D.T.ROSS  
"Structured analysis : A language for communicating idea"  
IEEE Trans.on Software Eng., vol SE-3, N°1, January 1977.
- 18) D.T.ROSS and KENNETH E.SHOMAN.TR  
"Structured analysis for requirements definition"  
IEEE Trans.on Software Eng., vol SE-3, N°1, January 1977.
- 19) GRINSEC  
"La commutation électronique"  
Eyrolles Paris 1980.
- 20) M.T.HILLS, S.KANO  
"Programming electronic switching systems"  
IEEE telecommunication Series 3, Peter Pergrims, LTD 1976.
- 21) M.KAJIWARA, K.SUZUKI  
"Consideration on call processing function distribution to subscriber terminal"  
IEEE Trans.on comm., vol COM-27, N°8, August 1979.
- 22) P.AZEMA, B.BERTHOMIEU, P.DECITRE  
"The design and validation by Petri nets of a mechanism for the invocation of remote servers"  
Proc. of IFIP Congress, Melbourne, October 1980.

- 23) G.JUANOLE  
"Modélisation du protocole de transport ECMA (European computer manu-  
facture association)"  
Rapport N°1 contrat LAAS-S.I.N.T.R.A.  
Note interne LAAS 80.I.27, Octobre 1980.
- 24) T.JOUBERT  
"Tentative de formalisation d'une spécification téléphonique : Aspects  
contrôle des enchaînements et gestion de données"  
Note technique NT/PAA/CLC/LSC/481  
CNET Issy-les-Moulineaux.
- 25) P.ZAFIROPULO, C.H.WEST, H.RUDIN, D.D.COWAN, D.BRAND  
"Towards analyzing and synthesizing protocols"  
IEEE Trans.on comm., vol. COM-28, N°4, April 1980.
- 26) CCITT  
"Spécification du système de signalisation N°6"  
CCITT - LIVRE ORANGE tome VI.2, Genève 1977.
- 27) CCITT  
"Signalling system N°7 specification 1979"
- 28) S.WELCH  
"Signalling in telecommunication networks"  
Peter Peregrims LTD 1979.
- 29) H.KONBER, M.DIAZ  
"Etude et analyse des techniques et systèmes de signalisations dans  
les réseaux de télécommunication"  
Note interne LAAS-IC (à paraître)
- 30) M.DIAZ, B.POTIN  
"Un système distribué pour la commande et le contrôle en temps réel  
des processus industriels"  
Actes du congrès AFCET - Informatique, Nancy, Novembre 1980.

31) I N T E L

RMX/80 Manual 1977.

32) O.APPEL, J.M.AYACHE, M.DIAZ

"Etude et développement d'un moniteur temps réel multitâches pour micro-ordinateur"

Note interne LAAS-LC 79.I.53 Toulouse, Septembre 1979.

33) C.MACCHI, J.F.GUILBERT

"Téléinformatique"

Dunod Paris 1979.

34) B.BERTHOMIEU

"Proving progress properties of communication protocols in AFFIRM"

AFFIRM memos 35 and 36, USC information sciences institute, Marine del rey, cal., U.S.A., Septembre 1980.

35) N.NAFFAH

"Etude de la gestion des terminaux dans un réseau général informatique et développement d'un système microprogrammé pour la connexion directe d'un terminal intelligent sur le réseau CYCLADES"

Thèse Docteur Ingénieur, Université Paris VI, Décembre 1975.

36) I S O

"Open systems interconnection"

ISO/TC97/SC16, June 1979.



# TABLE DES MATIERES

-----



	Pages
INTRODUCTION	1
CHAPITRE I - QUELQUES METHODOLOGIES SIGNIFICATIVES DE SPECIFICATION POUR LES SYSTEMES TEMPS REEL	5
I.1 INTRODUCTION	7
I.2 CRITERES DE MESURE ET D'EVALUATION D'UN MODELE	7
I.3 OUTILS DE MODELISATION - LES PRINCIPALES APPROCHES	7
I.3.1 Les approches	10
I.3.1.1 L'approche linguistique	10
I.3.1.2 L'approche mathématique	11
I.3.1.3 L'approche graphique	11
I.3.2 Les principaux modèles graphiques	11
I.3.2.1 Diagrammes d'états	11
I.3.2.2 Réseaux de Pétri	12
I.3.2.2.1 Définitions	12
I.3.2.2.2 Propriétés des Réseaux de Pétri	17
I.3.2.2.3 Réseaux de Pétri Etiquetés	19
I.3.2.2.4 Schémas à Réseaux de Pétri	19
I.3.2.2.5 Propriétés des Réseaux de Pétri non étiquetés et étiquetés	20
I.3.2.3 Analyse Structurale et Technique de Conception (S.A.D.T.)	21
I.4 MOTIVATION ET CHOIX D'UN MODELE	22
I.5 CONCLUSION	29
CHAPITRE II - METHODOLOGIE DE SPECIFICATION PAR UN COUPLE DESCRIPTION INFORMELLE - DESCRIPTION FORMELLE ET PREMIERE MODELISATION DU COMPORTEMENT D'UN ABONNE	31
II.1 INTRODUCTION	33
II.2 CONCEPTION HIERARCHIQUE D'UN SYSTEME COMPLEXE TEMPS REEL	35
II.3 LE MODELE	36
II.3.1 Les objets d'un module	36
II.3.2 Correspondance entre les sommets du réseau de Pétri et les classes d'objets	37
II.3.3 Modélisation par le couple informalité-formalité	38

<b>II.4 PREMIERE MODELISATION DU COMPORTEMENT D'UN ABONNE</b>	<b>38</b>
II.4.1 <i>Texte de spécification haut niveau de l'abonné</i>	39
II.4.2 <i>Réécriture des phrases et formalisation</i>	40
II.4.3 <i>Assemblage des microréseaux</i>	41
II.4.4 <i>Vérification du modèle</i>	43
II.5 <b>CONCLUSION</b>	47
<b>CHAPITRE III - CONCEPTION ET VERIFICATION DES PROTOCOLES DE LA COMMUNICATION - MODELISATION ET ANALYSE DES INTERACTIONS ENTRE ABONNES BASEES SUR LES RECOMMANDATIONS CCITT N°6,7</b>	<b>49</b>
III.1 <b>INTRODUCTION</b>	51
III.2 <b>ERREUR DE CONCEPTION DANS UN PROTOCOLE</b>	52
III.3 <b>METHODOLOGIES DE CONSTRUCTION DE PROTOCOLES</b>	53
III.4 <b>MODES DE COMMUNICATION</b>	56
III.4.1 <i>Maître au début et en fin de communication</i>	56
III.4.2 <i>Maître au début et esclave en fin de communication</i>	58
III.4.3 <i>Mode Asynchrone en fin de communication</i>	58
III.4.4 <i>Mode Asynchrone en fin de communication avec                   accusé de réception</i>	60
III.4.5 <i>Mode Asynchrone en fin de communication avec                   accusé de réception, utilisant un nombre optimal                   (minimal) de messages</i>	60
III.4.6 <i>Mode Asynchrone en début de communication :                   appels croisés</i>	62
III.5 <b>DEUXIEME MODELISATION DU COMPORTEMENT D'UN ABONNE</b>	66
III.6 <b>MESSAGES FONCTIONNELS DE SIGNALISATION INSPIRES DES RECOMMANDATIONS CCITT N°6,7</b>	71
III.7 <b>MODELES DE PROTOCOLES DE COMMUNICATION</b>	74
III.7.1 <i>Modèle du protocole non optimal</i>	76
III.7.2 <i>Modèle du protocole non optimal à nombre réduit                   de messages</i>	81
III.7.3 <i>Modèle du protocole optimal</i>	84
III.7.4 <i>Modèle du protocole avec acquittement pour le                   message occupé</i>	86

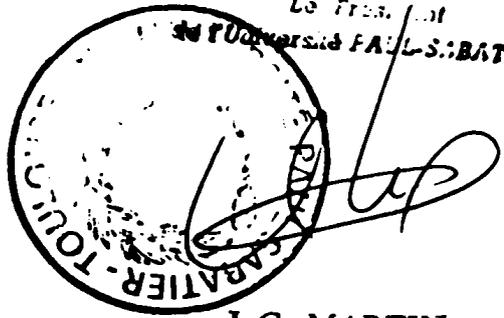
III.7.5	Modèle du protocole étendu	88
III.7.6	Modèle du protocole totalement asservi	90
III.8	CONCLUSION	93
CHAPITRE IV -	PRINCIPES D'UNE IMPLEMENTATION DE PROTOCOLES	
	DECRIIS PAR DES RESEAUX DE PETRI	95
IV.1	INTRODUCTION	97
IV.2	PRESENTATION DU MONITEUR TEMPS REEL MULTITACHES	99
IV.2.1	Les tâches et les messages	99
IV.2.2	Les concepts de base du moniteur	100
IV.2.3	Les primitives importantes : ATTENTE et ENVOI de messages	100
IV.3	BASES DE LA SIMULATION	102
IV.3.1	Principes généraux	102
IV.3.2	Représentation de l'abonné	104
IV.4	CONCLUSION	109
CONCLUSION		111
BIBLIOGRAPHIE		117



Autorisation de publication accordée  
TOULOUSE, le 26 NOV. 1980

Le Président

de l'Université PAUL-SABATIER



J.-C. MARTIN

